Advanced Routing Algorithm for FUB Integration

A Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

In

Electronics & Communication Engineering

(VLSI Design)

By

PRIYA PATEL

(14MECV21)

Department of Electronics Engineering

Electronics & Communication Engineering Program

Institute of Technology, NIRMA UNIVERSITY

Ahmedabad-382 481

Advanced Routing Algorithm for FUB Integration

A Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

In

Electronics & Communication Engineering

(VLSI Design)

By

PRIYA PATEL
(14MECV21)

Under the Guidance of

Dr.N.M.Devashrayee
Professor, EC Department, Nirma University

&

Mr.Prasad Anandrao Mahajan
Section Layout Owner, INTEL India Pvt. Ltd.

Department of Electronics Engineering

Electronics & Communication Engineering Program

Institute of Technology, NIRMA UNIVERSITY

Ahmedabad-382 481

Declaration

This is to certify that

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

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

-Priya Patel(14MECV21)

Disclaimer

“The content of this paper does not represent the technology, opinions, beliefs, or positions of Intel Technologies India Pvt. Ltd. Company, its employees, vendors, customers, or associates.”

Certificate

This is to certify that the Major Project entitled **"Advanced Routing Algorithm for FUB Integration"** submitted by **Priya Patel (14MECV21)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design Engineering of Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of our knowledge, have notbeen submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Guide: Program coordinator:

Dr. N.M.Devashrayee
Professor, EC Department

Dr. N.M.Devashrayee
Professor, EC Department

HOD: Director:

Dr. P. N. Tekwani
Head of EE Dept

Dr. P. N. Tekwani
Director, ITNU

Certificate

This is to certify that the Major Project entitled **"Advanced Routing Algorithm for FUB Integration"** submitted by **Priya Patel (14MECV21)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design Engineering of Nirma University, Ahmedabad is the record of work carried out by her at Intel Technology India Pvt. Ltd., under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of our knowledge, have not been submitted to any other university or institution for award of any degree or diploma.

PROJECT MANAGER: **PROJECT GUIDE:**

Mr. Rajesh Navale **Mr. Prasad Anandrao Mahajan**

Acknowledgement

I take this opportunity to express my gratitude to my guide Dr. N. M. Devashrayee, Institute of technology, Nirma University for being a source of inspiration and for timely guidance during the project.

I take immense pleasure in thanking of my manager at Intel Mr. Rajesh Navale for his suggestions throughout the work.

I would like to express my gratitude and sincere thanks to Mr. Prasad Anandrao Mahajan and Mr. Sreenivas Subramaniam for timely suggestions and the concept clarity regarding the topic that helped me during this work.

I would like to thank Dr. P. N. Tekwani, Head of Electrical Engineering Department for allowing me to undertake this thesis work and for his guidelines during the review process.

I would also like to thank my colleagues, who in one way or the other provided me assistance and motivation whenever required.

- Priya Patel
(14MECV21)

Abstract

VLSI refers to a technology through which it is possible to implement large circuits in silicon circuits with up to a million transistors. The VLSI technology has been successfully used to build microprocessors, signal processors, systolic arrays, large capacity memories, memory controllers and interconnection networks. Chip designing consists of various design processes like architectural design, logic design, circuit design, physical design, physical verification and signoff, fabrication, packaging and testing etc. Physical design is one of the important phase of VLSI design cycle. Continuous shift towards the design in nanometer scale has been increasing complexity in physical designing of the chip. It introduces new metal layers and DRC runsets which increases complexity of the layout.

In Full chip designs, as the chip area is reduced with increased feature requirement the complexity of the interconnect design increases with great extent. On the other hand, it is important to save on interconnection cost ,since wires are far more expensive in VLSI than transistors. Among these design issues, rising RC delay on on-chip wiring, increasing noise susceptibility due to coupling, delay prediction considering inductance and noise effects, and power and reliability concerns due to increasing current density are commonly mentioned. The gap between device and wire delays is increased, especially the global interconnect delays, which do not scale well with the feature size. Interconnect routing directly impacts circuit performance, area, reliability, power, and manufacturing yield.Optimization of interconnect routing is required to achieve high performance. Full chip interconnect routing is done in three steps; global routing, timing driven routing and detailed routing. In a complex VLSI chip, interconnect routing free of DRC and LVS violation is also essential to get better power and noise performance.

In this report, detailed routing is discussed. In the process of automatic design of VLSI layouts, routing is accomplished using computer programs called routers. In this report, QEA router and problems associated with the QEA router are discussed. To overcome all those problems and increase efficiency, advanced routing algorithm for FUB integration is developed and enhanced in different ways. Result shows that it is successful in saving time as well as effort. With

the help of advanced routing algorithm for FUB integration, routing quality is improved and design cycle time is reduced.

Contents

List of Tables

List of Figures

Abbreviations

FUB	Functional Unit Block
DRC	Design Rule Check
LVS	Layout vs. Schematic
ERC	Electric Rule Check
PCB	Printed Circuit Board

Chapter 1

Introduction

1.1 VLSI chip design flow

Complexity of VLSI circuits is in the order of millions of transistors, designing a VLSI circuit is understandably a complex task. In order to reduce the complexity of design process, several intermediate levels of abstractions are introduced. Majority of the chip designing time is consumed by the physical designing stage. Physical design of an integrated circuit refers to the process of generating the final layout for the circuit.

The design is taken from specification to fabrication step by step with the help of various CAD tools. Architectural design is then created to analyze the design in terms of functionality, performance, compliance to given standards, and other specifications. Decisions made at this stage affect the cost and performance of the design significantly. Once system architecture is designed, data path and control path design is carried out. RTL description is done using HDLs. This RTL description is simulated to test functionality. RTL description is then converted to a gate-level netlist using logic synthesizer. A gate-level netlist is a description of the circuit in terms of gates and connections between them. Gate level implementation is then done based on the netlist in such a manner that it can meet the timing and power specification. Physical design converts a circuit description into a geometric description. This description is used to manufacture a chip. The physical design cycle consists of Partitioning, Floor-planning, Placement, Clock

tree synthesis, Routing and Timing closure.

Architectural design of a chip is carried out by expert human engineers. Decisions made at this stage affect the cost and performance of the design significantly.Architectural design cannot be done entirely by a computer program. However, computer programs can aid the system architect in making important decisions. For instance, an architect can tune a parameter (such as the size of the cache) through simulation. Simulators and performance prediction tools are very useful to a computer architect who is experimenting with an innovative idea. Once the system architecture is defined, it is necessary to carry out two things; data path design and control path design.The data path of a circuit includes the various functional blocks, storage elements, and hardware components to allow transfer of data among functional blocks and storage elements. The control path of a circuit generates the various control signals necessary to operate the circuit. Control signals are necessary to initialize the storage elements in the circuit, to initiate data transfers among functional blocks and storage elements, and so on.[3]

If the circuit must be implemented on a printed circuit board using off-theshelf components, then the next stage in design is to select the components so as to minimize the total cost and at the same time maximize the performance. Following the selection procedure, the IC chips are placed on one or more circuit boards and the necessary interconnections are established using one or more layers of metal deposits. A similar procedure may be used in case the circuit must be implemented on a VLSI chip using predesigned circuit components from a module library. The predesigned modules are also known as macro-cells[3]. The cells must be placed on the layout surface and wired together using metal and polysilicon (poly) interconnections.

Physical design of a circuit is the phase that precedes the fabrication of a circuit. In most general terms, physical design refers to all synthesis steps succeeding logic design and preceding fabrication. These include all or some of the following steps: logic partitioning, floorplanning, placement, and routing. The performance of the circuit, its area, its yield, and its reliability depend critically on the way the circuit is physically laid out.

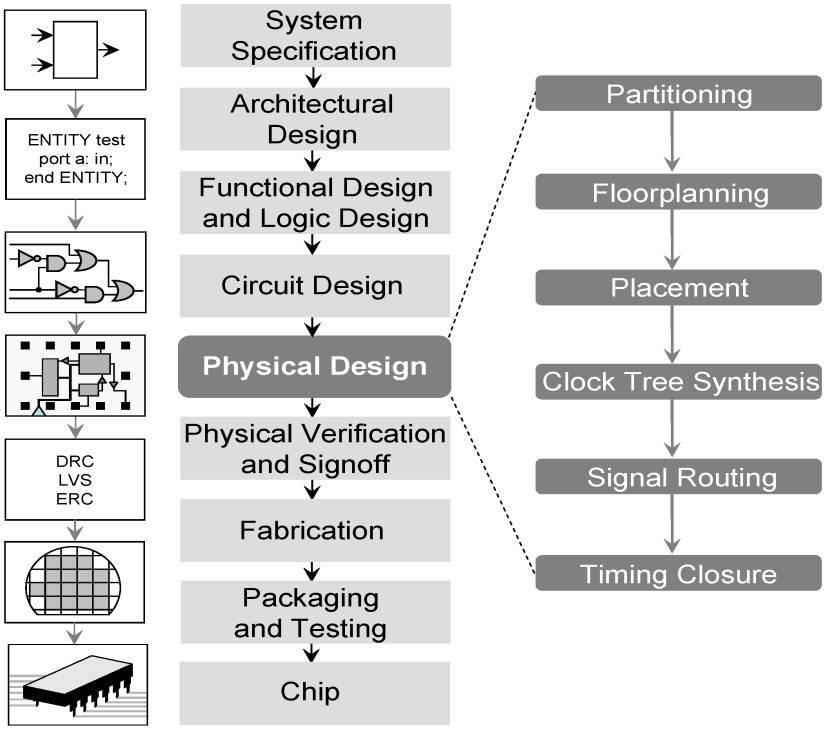


Figure 1.1: VLSI chip design flow

- Partitioning: Partitioning a big complex system into number of blocks enables structural implementation by using divide and conquer approach. Partitioning is also needed to handle engineering change orders. For huge systems, design iterations require very fast turn around time. A hierarchical partitioning methodology can localize the modifications and reduce the complexity.

- Floorplanning: It is an essential design step for hierarchical, building-module design methodology. Floorplanning provides early feedback that evaluates architectural decisions, estimates chip areas, and estimates delay and congestion caused by wiring.

- Placement: Placement is a critical step in the VLSI design flow mainly for the following four reasons. First, placement is a key factor in determining the performance of a circuit. Second, placement determines the routability of a Design. Third, placement decides the distribution of heat on a die surface. An uneven temperature profile can lead to reliability and timing problems. Fourth, power consumption is also affected by placement. A good placement solution can reduce the capacitive load because of the wires.

- Clock Tree Synthesis: Clock balancing is important for meeting the design constraints mainly in multi-clock system design. Clock tree synthesis is done after placement before signal routing to give best routing resources to clock signal. Clock trunk and spines are routed depending upon the system topology. A good clock tree network will minimize the clock skew and jitter.

- Routing:The most important objective of routing is to complete all the required connections, otherwise the chip would not function well and may even fail. Other objectives, such as (1) reducing the routing wire length and (2) ensuring each net to satisfy its required timing budget, have become essential for modern chip design.

The performance of the circuit, its area, its yield, and its reliability depend critically on the way the circuit is physically laid out. In an integrated circuit

layout, metal and polysilicon are used to connect two points that are electrically equivalent. Both metal and poly lines introduce wiring impedances. Thus a wire can impede a signal from traveling at a fast speed. The longer the wire, the larger the wiring impedance, and longer the delays introduced by the wiring impedance. When more than one metal layer is used for layout, there is another source of impedance which is via or contact. There are two components to the area of an integrated circuit - the functional area, and the wiring area. The larger the chip area, the lower the yield. A low yield would mean a high production cost, which in turn would increase the selling cost of the chip.

The reliability of the chip is also influenced by the layout. For instance, vias are sources of unreliability, and a layout which has a large number of vias is more likely to have defects. Further, the width of a metal wire must be chosen appropriately by the layout program to avoid metal migration. If a thin metal wire carries a large current, the excessive current density may cause wearing away of metal, tapering the wire slowly, resulting in an open circuit.

1.2 Overview of Routing

Routing phase is the one that determines the actual course of the wires connecting the cells that have been placed. Efficient placement of the cells can result in less resource requirement during routing. In routing different metal layers are used which have different resistive and capacitive characteristics. In a complex chip, manual routing work can be very time consuming and inefficient. To handle the high complexity different automatic router based on different routing algorithms are adopted in the industry. Most of the such algorithms adopt the two-stage approach of global routing followed by detailed routing[1]. Global routing first partitions the routing region into tiles and decides tile-to-tile paths for all nets, whereas detailed routing determines the exact tracks and vias for nets.

- Global routing: In global routing phase chip is considered to be a single layer frame. Pins of the chip component are found based on the placement of that particular block. A net can be defined as a set pins which are to be connected. The set of the nets are basically need to be routed by particular

metal layer in defined region. Actual wired connection is not done in this stage. Based on the timing constraints, metal layer resources and noise constraint, a net is assigned highest metal layer for routing. This approach is very effective for data buses having large number bits. For critical control signals which run through majority of blocks this approach is required.

Global routing flow is as mentioned below:

1. Defining the routing regions : In this step, layout area is devided into routing regions. In some cases, net can be routed over standard cell. The routing regions are formed as 2D or 3D channels, switch boxes and other region types.These routing regions, their capacities and their connections are then represented by graph[1].
2. Mapping nets to routing regions: In this step, each net of the design is tentatively assigned to one or several routing regions so as to connect all of its pins.The routing capacity of each routing region limits the number of nets traversing this region[1]. Other factors such as timing and congesion, also affect the path choosen for each net. For example routing resources can be priced differently in different regions with available routing capacity - the more congested routing region, the higher the cost for any net subsequently routed through that region.
3. Assigning crosspoints: In this step, also known as midway routing, routes are assigned to fixed locations, or crosspoints, along the edges of the routing regions. Crosspoint assignment enables the scaling of global and detailed routing to designs with millions of cells as well as distributed and parallel algorithms, since the routing regions can be handled independently in detailed routing.

- Detailed routing: In detailed routing all the routing constraints are checked for the particular net. Within a defined routing region the routing which is done in higher metal layer during global routing stage is downgraded to the metal layer of the pins of the nets based on the available metal layers and connection is completed by drawing vias and junctions between metal layers. The objective of the detailed routing is to reduce length of the wire,

number of vias, utilization of metal layers and also the total routing area. Detailed routing should also result in layout free of DRC and LVS violations.

Different types of detailed routing are as mentioned below:

1. Channel routing is a special case of detailed routing where the connections between terminal pins are routed within a routing region (channel) that has no obstacles. The pins are located on opposite sides of the channel. By convention, the channel is oriented horizontally pins are on the top and bottom of the channel. In row-based layouts, in a given block, the routing channels typically have uniform channel width. In gate-array and standard-cell circuits that use more than three layers of metal, channel height, the number of routing tracks between the top and bottom boundaries of the channel, is also uniform.
2. Switchbox routing is performed when pin locations are given on all four sides of a fixed-size routing region. This makes the detailed routing significantly more difficult than in channel routing[1].
3. OTC (over-the-cell) routing uses additional metal tracks, e.g., on Metal3 and Metal4, that are not obstructed by cells, allowing routes to cross cells and channels. OTC routing can use only the metal layers and tracks that the cell do not occupy.When the cells utilize only polysilicon and metal1 layers, routing can be done on the remaining metal layers as well as unused metal1 resources.

Apart from these two general approaches specialized routing algorithms are used for routing power lines and clock lines. Routing models can also be classified based on the usage of the metal grids.[4]

- Grid-based model: In such model grids are superimposed on the routing region. The position of these grids can vary from one metal layer to another. Wires routed in particular metal layer must follow the grid lines.
- Gridless model: In this model, there is no metal layer grid lines. Wires can be routed at any random place. There is no restriction in wire geometry in such model.

Below 0.25 μm, the performance of the chip is primary limited by interconnect delay. Optimized routing can improve the performance but it is limited to some extent. To overcome such issues repeaters are added on the nets.

1.3 Objective

The major challenge of full chip design is the physical designing. Optimized routing which can achieve timing, power and noise constraints is essentially required. Many automatic placer and router flows based on various algorithms failed to provide high quality layout. Objective is to introduce an automated router which can resolve all complex designing issues with very less amount of manual routing work.

1.4 Organization of report

The Report is organized as mentioned:

In Chapter 2, brief Literature Review regarding the back-end design tools, hierarchy of design, interconnect modelling, routing in full chip layout design flow, design rule checks, QEA router and problems related to QEA router are mentioned.

In Chapter 3, advanced routing algorithm for FUB integration is mentioned. It includes basic flow of advanced routing algorithm for FUB integration, how the problems associated with QEA router can be solved, enhancements done in this routing algorithm and results and output of the routing algorithm.

Chapter 2

Literature Review

Physical designing is very critical part of VLSI chip designing such as a processor core. The tool which is used during thesis work is PARADE. This tool is Intel specific tool, which is used for back-end design steps such as placement, routing, clock tree synthesis and timing analysis of interconnects. Partitioning a big complex system into number of blocks enables structural implementation by using divide and conquer approach. A hierarchical partitioning methodology can localize the modifications and reduce the complexity. The detailed design methodology and interconnect modelling are discussed in this chapter.

2.1 Hierarchy of design

To handle very complex and large designs, partitioning of the entire design based on some per-defined factors is required. Partitioning is also needed to handle engineering change orders.

A commonly used hierarchical partitioning model is shown in the Fig. 2.1. The lowest design hierarchy level is functional unit block (FUB). A functional unit block implements the basic functional designs such as adder, multiplier, divider, etc. A number of functional unit blocks constitute a section. Number of functional unit blocks are grouped based on their characteristics and physical location in layout. The functional unit blocks are considered to be a black box at the section level and only shared attributes are visible at section level. A cluster is made up of

different sections. The group of cluster constitute the full chip. Such methodology enables the designer to debug design issues very effectively with less amount of turn around time.

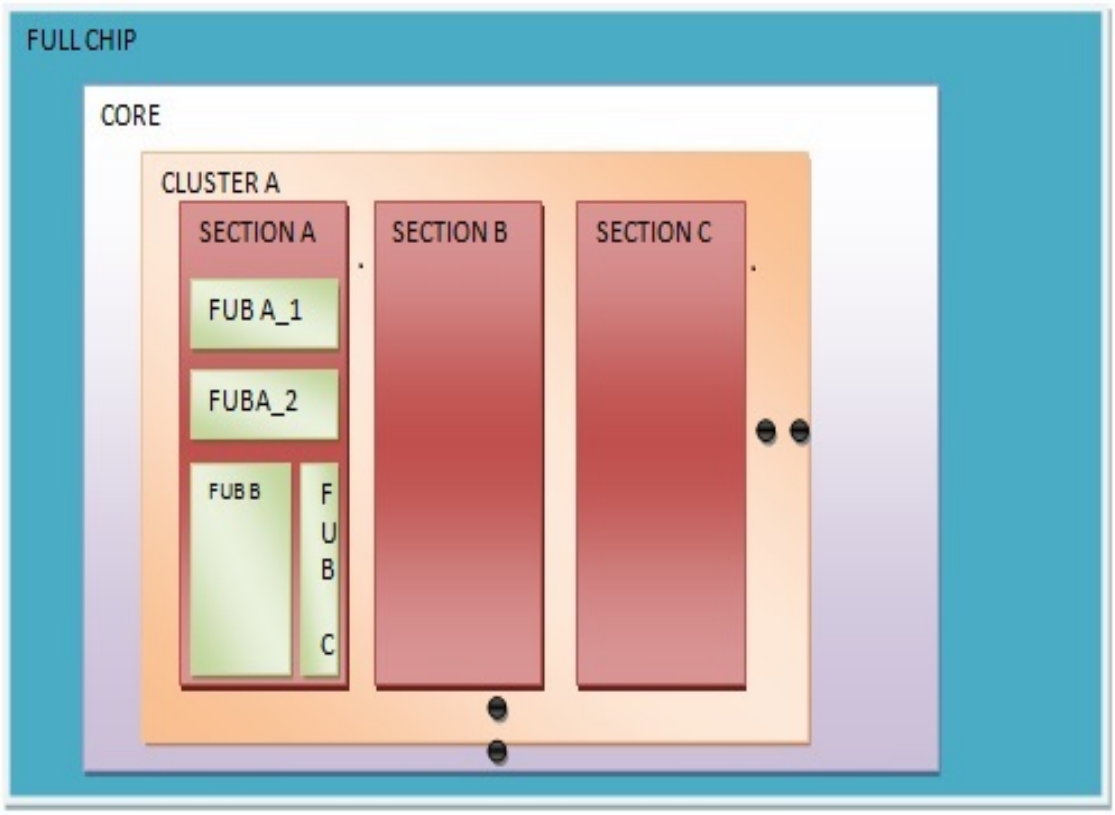


Figure 2.1: Hierarchy of design

2.2 Interconnect modelling

Interconnect is a metal drawn for connecting the transistors together. As the technology advances, more number of features are added in reduced chip area. This results in increased interconnects. The gap between device and wire delays is also increased, especially the global interconnect delays, which do not scale well with the feature size. Advancement in the design process allows the use of

more number of metal layers. Based on the different design process, different metal materials can be used. The basic differences between such metal layers can be width of the wire, thickness of the wire, spacing between two wires, material characteristics, etc.

As shown in Fig.2.2 for a particular metal layer and process thickness is fixed while length and width of the wire have fixed minimum value. Minimum spacing between two adjacent wires is also fixed which is helpful in reducing cross-talk and interference. Interconnect resistance and capacitance are used for delay and slope calculation. Resistance of any wire only depends on the wire geometry and it doesnt depend on the neighbour wire. Wire capacitance depends not only on its parameter such as metal type, width and space, but also on its neighbour from all direction. A interconnect has capacitance to the ground which is also known as self capacitance and capacitance to other interconnects which is known as cross capacitance. The total effective capacitance is the parallel combination of the self capacitance and cross capacitance.

Different metal layers in a process have minimum space and width values. Lower metal layers have lesser value of width and space as compared to higher metal layers. Hence lower metal layers have high resistance value and less capacitance value in comparison of less resistance and high capacitance value of higher metal layers. To reduce the fabrication complexity and to ease the metal routing horizontal and vertical metal grids are used for respective metal layer. For example, all even metal layers are routed in horizontal grids and all odd metal layers are routed in vertical grids.

Two different metal layers can be connected together by creating via. Via can be modelled as a resistor.

In most of the global interconnects, more than one metal layers are used to complete the routing between driver and receiver/receivers. In such cases total resistance and capacitance can be calculated by using Elmores delay model. A simple example is shown in the above Fig.2.3. Three different metal layers are used to connect pin terminals driver and receiver. Resistance of the via are included in wire resistance of respective metal layer.

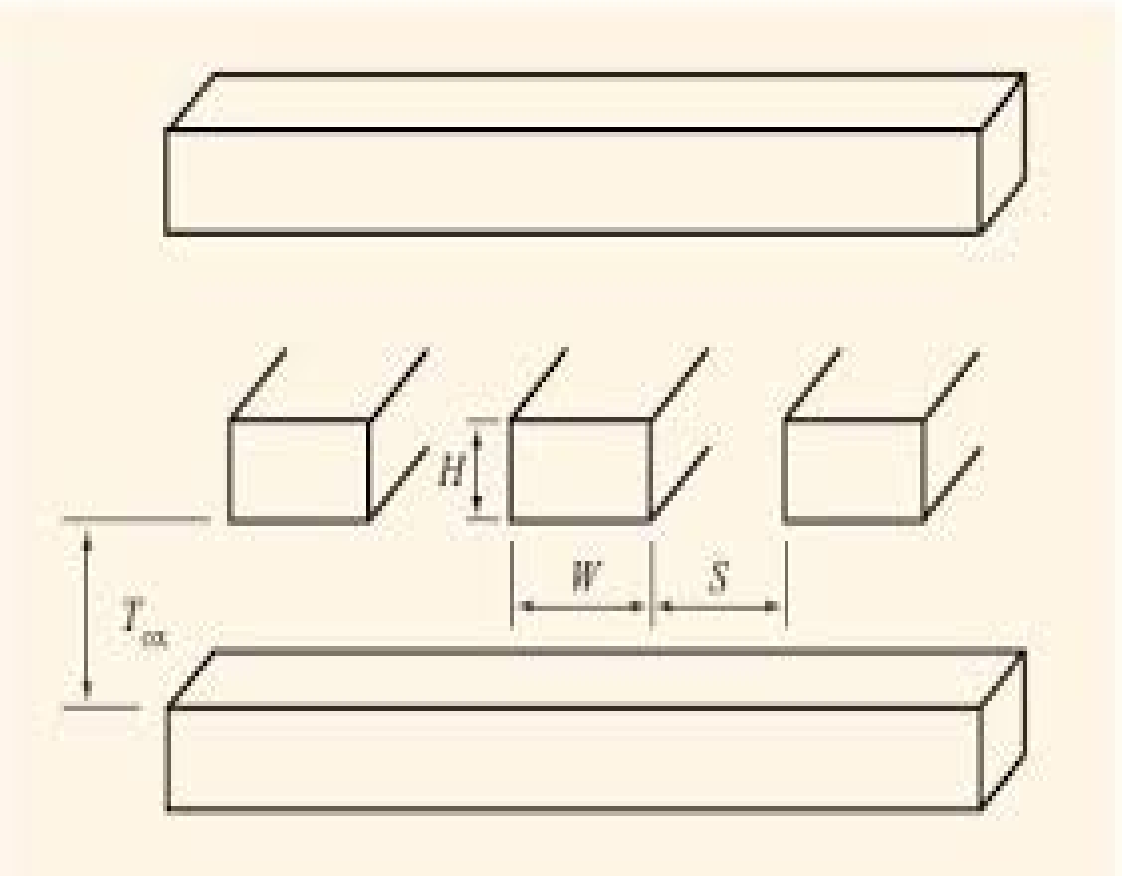


Figure 2.2: Interconnect geomatrical parameters

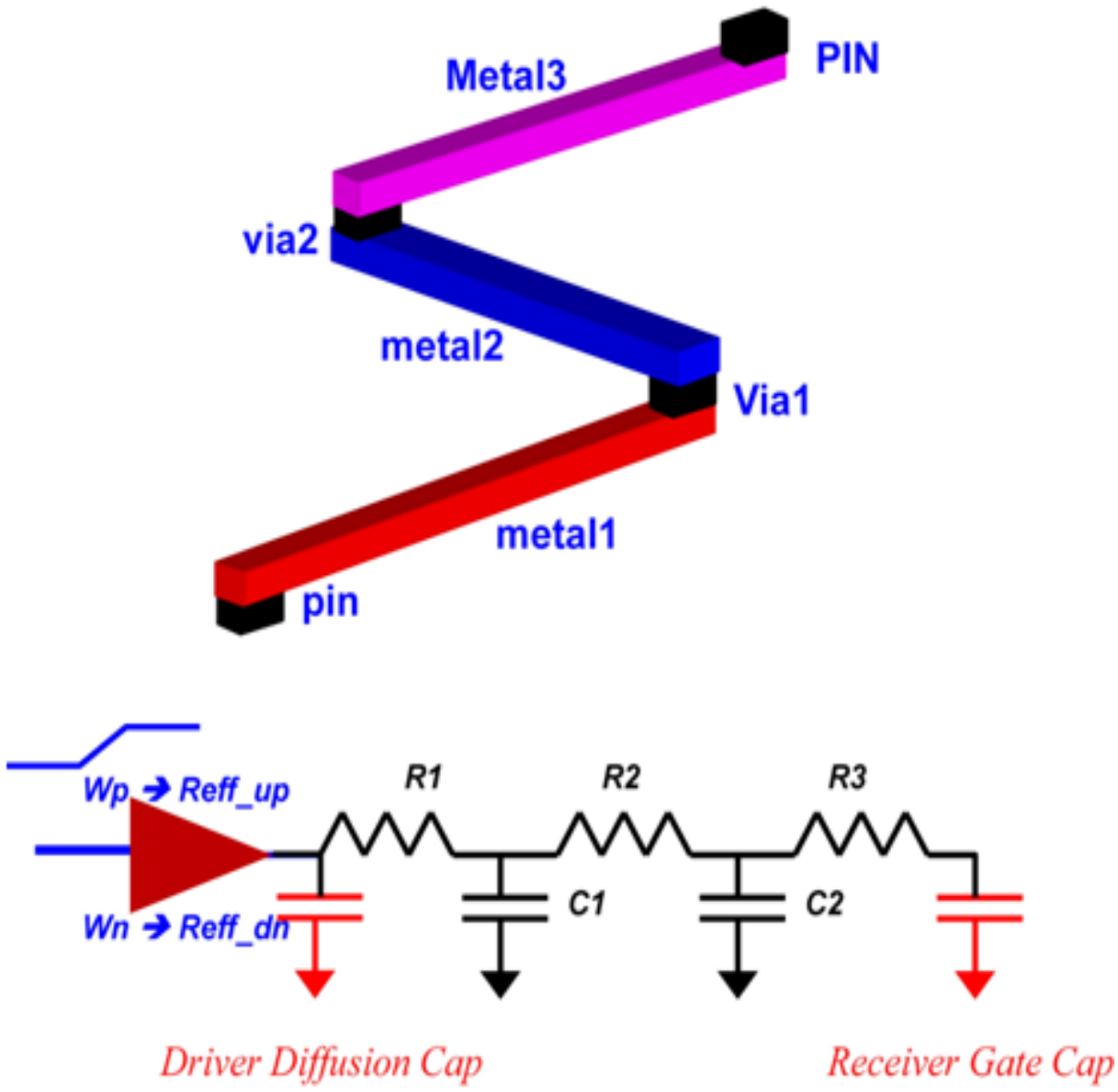


Figure 2.3: Global interconnect and its RC model

2.3 Routing in full chip layout design flow

Routing is one of the important stage in full chip layout design flow. Physical design directly impacts circuit performance, area, reliability, power, and manufacturing yield. Examples of these impacts are discussed below.

- Performance: long routes have significantly longer signal delays.
- Area: placing connected modules far apart results in larger and slower chips.
- Reliability: large number of vias can significantly reduce the reliability of the circuit.
- Power: transistors with smaller gate lengths achieve greater switching speeds at the cost of higher leakage current and manufacturing variability; larger transistors and longer wires result in greater dynamic power dissipation.
- Yield: wires routed too close together may decrease yield due to electrical shorts occurring during manufacturing, but spreading gates too far apart may also undermine yield due to longer wires and a higher probability of opens.

Routing affects all the mentioned impacts. Full chip design routing is done in three steps mentioned below:

- Global routing
- Timing driven routing
- Detailed routing

Performance driven physical layout design flow is used in all the industries which is as shown in Fig 2.4.

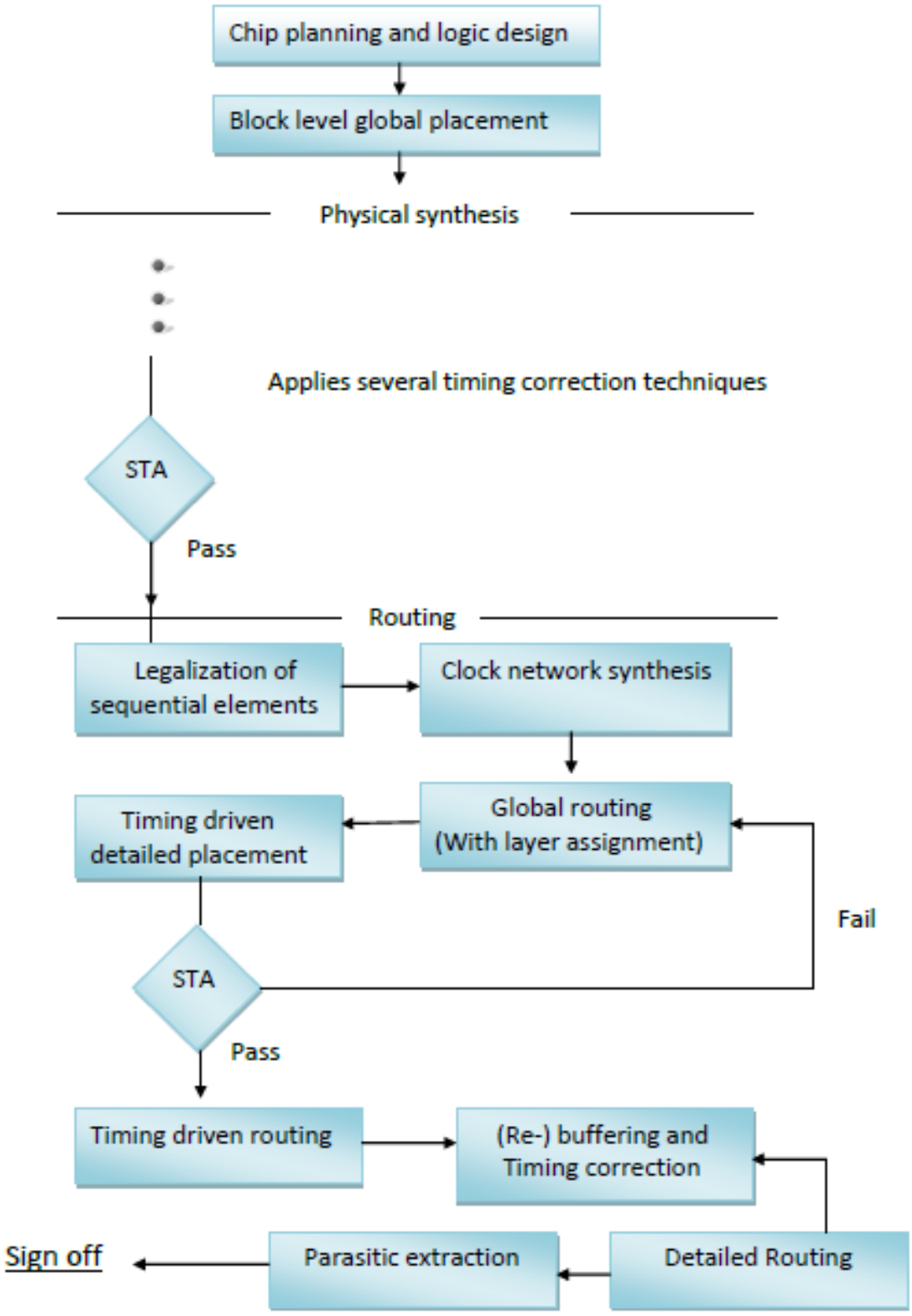


Figure 2.4: Performance driven physical design layout flow

After physical synthesis, all combinational and sequential elements in the design are connected during global and clock routing, respectively. First, the sequential elements of the design, e.g., flip-flop and latches, are legalized. Then, clock network synthesis generates the clock tree or mesh to connect all sequential elements to the clock source. Modern clock networks require a number of large clock buffers;performing clock-network design before detailed placement allows these buffers to be placed appropriately. Given the clock network, the design can be checked for hold-time (short path) constraints, since the clock skews are now known, whereas only setup (long path) constraints could be checked before[1].

Layer assignment : After clock-network synthesis, global routing assigns global route topologies to connect the combinational elements. Then, layer assignment matches each global route to a specific metal layer. This step improves the accuracy of delay estimation because it allows the use of appropriate resistance-capacitance (RC) parasitics for each net[1]. Note that clock routing is performed before signal-net routing when the two share the same metal layers clock routes take precedence and should not detour around signal nets.

Timing-driven detailed placement : The results of global routing and layer assignment provide accurate estimates of wire congestion, which is then used by a congestion-driven detailed placer. The cells are (1) spread to remove overlap among objects and decrease routing congestion, (2) snapped to standard-cell rows and legal cell sites, and then (3) optimized by swaps, shifts and other local changes. To incorporate timing optimizations, either perform (1) non-timing-driven legalization followed by timing-driven detailed placement, or (2) perform timing-driven legalization followed by non-timing-driven detailed placement. After detailed placement, another timing check is performed. If timing fails, the design could be globally re-routed or, in severe cases, globally re-placed[1].

To give higher priority to the clock network, the sequential elements can be legalized first, and then followed by global and detailed routing. With this approach, signal nets must route around the clock network. This is advantageous for large-scale designs, as clock trees are increasingly becoming a performance bottleneck.

Another variant performs detailed placement before clock network synthesis, and then is followed by legalization and several optimization steps. After the

clock network has been synthesized, another pass of setup optimization is performed. Hold violations may be addressed at this time or, optionally, after routing and initial STA.

Timing-driven routing : After detailed placement, clock network synthesis and post-clock network optimization, the timing-driven routing phase aims to fix the remaining timing violations.

If there are still outstanding timing violations, further optimizations such as re-buffering and late timing corrections are applied. An alternative is to have designers manually tune or fix the design by relaxing some design constraints, using additional logic libraries, or exploiting design structure neglected by automated tools. After this time-consuming process, another timing check is performed. If timing is met, then the design is sent to detailed routing, where each signal net is assigned to specific routing tracks. Typically, incremental STA-driven Engineering Change Orders (ECOs) are applied to fix timing violations after detailed placement; this is followed by ECO placement and routing. Then, parasitic extraction determines the electromagnetic impact on timing based on the routes shapes and lengths, and other technology-dependent parameters.

Signoff : The last few steps of the design flow validate the layout and timing, as well as fix any outstanding errors. If a timing check fails, ECO minimally modifies the placement and routing such that the violation is fixed and no new errors are introduced.

After completing timing closure, manufacturability, reliability and electrical verification ensure that the design can be successfully fabricated and will function correctly under various environmental conditions[1]. The four main components are equally important and can be performed in parallel to improve runtime.

- Design Rule Checking (DRC) ensures that the placed-and-routed layout meets all technology-specified design rules e.g., minimum wire spacing and width.

- Layout vs. Schematic (LVS) checking ensures the placed-and-routed layout matches the original netlist.

- Antenna Checks seek to detect undesirable antenna effects, which may damage a transistor during plasma-etching steps of manufacturing by collecting

excess charge on metal wires that are connected to PN-junction nodes. This can occur when a route consists of multiple metal layers and a charge is induced on a metal layer during fabrication.

- Electric Rule Checking (ERC) finds all potentially dangerous electric connections, such as floating inputs and shorted outputs.

2.4 Design Rule Check

Design Rules are a series of parameters provided by semiconductor manufacturers that enable the designer to verify the correctness of a mask set. Design rules are specific to a particular semiconductor process. A design rule set specifies certain geometric and connectivity restrictions to ensure sufficient margins to account for variability in semiconductor manufacturing processes, so as to ensure that most of the parts work correctly.

During the routing phase, most of the design rules are checked and the corresponding violations are resolved. A common set of design rules employed in industry is explained here.

- Short: When two different nets are routed on same metal grid, metal short is created. Short can also be between signal routing, via, power lines, etc.

- Space: Space violation can further classified into two categories. Min space is the minimum allowed spaced between tow wires in same metal layer. Max space rule provides maximum allowed separation between two wires in same metal layer. This rule is introduced to obtain uniform metal fill.

- Min length: Min length rule provides the minimum length of the wire required in particular metal layer.

- Max width: Max width provides maximum allowed width of the wire in given metal layer

- Corner to corner: Corner to corner specifies the minimum separation required between corners/ends of two wires in same metal layer.

- Min jog: Min jog violation is observed when the length of the jog is less than the threshold values. This is observed when two segments of the same net in same metal layer have different width.
- Valid width: Valid width specifies the set of the allowed metal layer grid widths.
- Via to via: Via to via rule provides the required minimum separation between two vias of the same metal layer. The vias are between same two metal layers.
- Via to via n-1: Via to via n-1 is similar to the via to via rule. The only difference is consideration of vias between one higher and one lower metal layer with respect of given metal layer.
- Cut size: Via cut size rule provides valid via size for a given width of the metal layer.
- Via coverage: There are many other rules are classified under this rule. This rule provides the minimum wire length required to cover up the via. If the wire is shorting with other wire or it is not on the proper grid or if the dimension of the via is not proper.
- Non fill: Non-fill violation is observed when the wire in given metal layer is not sitting exactly on the metal grid. During metal the region which have off-grid wires is left without filling.
- Grid rules: Grid rules can be classified into several categories. Off grid issue is observed when the wire of given metal layer is not sitting on the grid or wire width is not valid.

Some of the examples of the DRC violations are shown in Fig.2.5.To overcome all such violation, precise routing with the optimum use of metal source is required. Manual routing will take many iteration to obtain violation free optimum routing. QEA router is a industry level automatic router which can resolve such routing

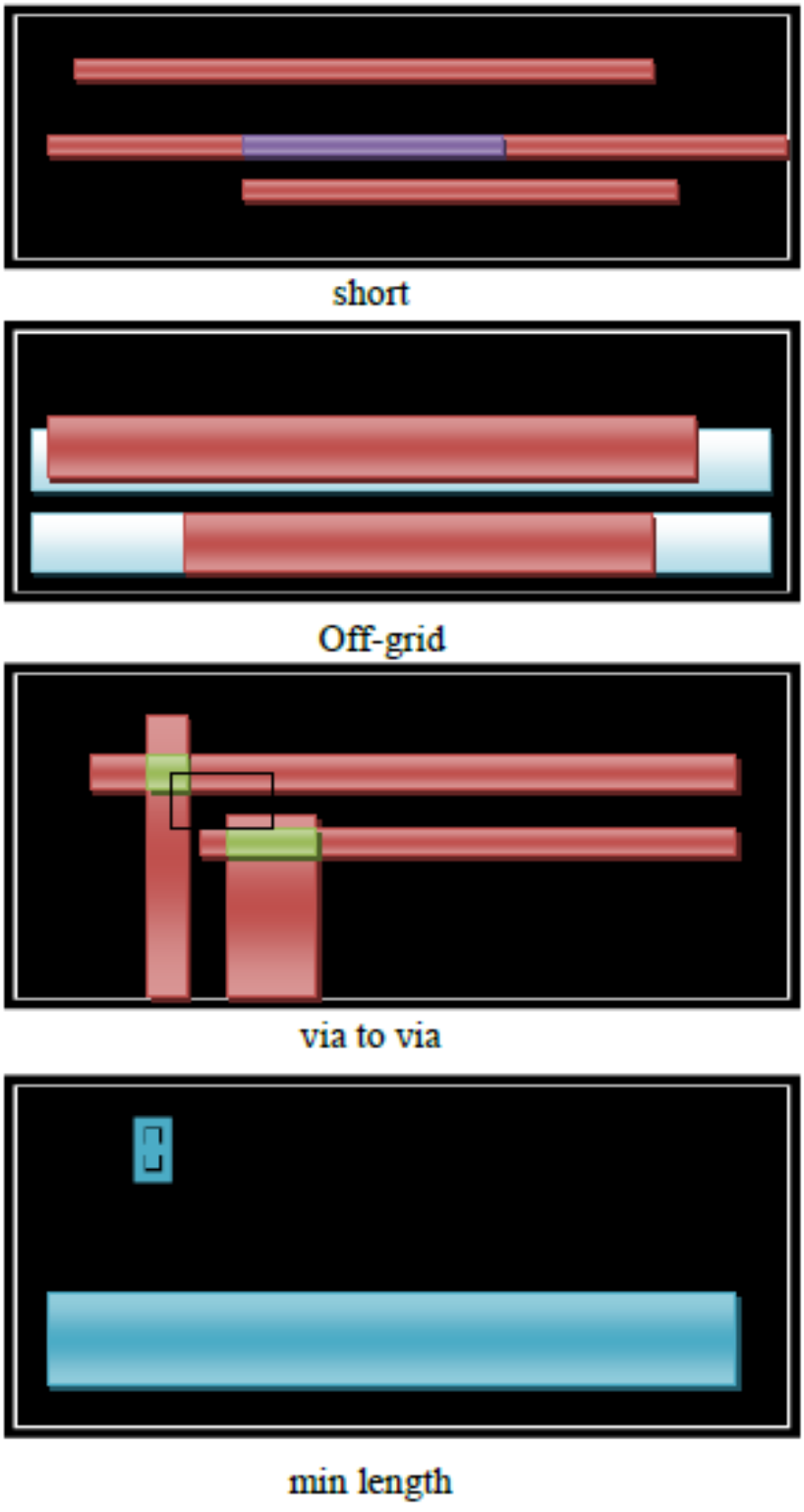


Figure 2.5: Examples of DRC violations

issues. Many other automatic router based on different algorithms can be used based on the application such as Mixed signal circuit design, SOC based design, PCB based design, etc.

2.5 Intoduction to QEA router

In the process of automatic design of VLSI layouts and printed circuit boards (PCBs), the phase following cell placement is routing. The number of cells on a VLSI chip, or IC chips on PCBs to be interconnected is generally very large. For this reason, routing is accomplished using computer programs called routers[3]. The task of router consists of precisely defining paths on the layout surface, on which conductors that carry electrical signals are run. These conductors (also called wiring segments) interconnect all the pins that are electrically equivalent.

Routing takes up almost 30 percent of design time and a large percentage of layout area. Automatic routing algorithms were first applied to design of PCBs. In the beginning PCBs were not very dense. None the less, basic ideas of automatic routing had been developed and some are still valid, and are being adapted to new emerging problems and larger PCBs. Recently, the main application of automatic routers has been in the automated design of VLSI circuits.

One very common technique that is used to connect two points belonging to the same net considers the layout as a maze. Finding a path to connect any two points is similar to finding a path in the maze. Such routers that connect two points by finding a path in a maze are called maze routers[3]. Maze routers assume the layout floor to be made up of a rectangular array of grid cells. Functional cells to be interconnected fill up some slots in this grid and constitute the obstacles of the maze. Maze routers such as those based on the Lee algorithm connect a pair of points at a time. The main objective of a routing algorithm is to achieve complete automatic routing with minimal need for any manual intervention. As mentioned earlier, the total area taken up by a circuit is the sum of functional area and the wiring area. In order to implement a circuit in minimal possible area, it is essential that the wiring area is reduced. Individual connections lengths must also be kept small in order to satisfy performance criteria.

QEA (Quasar Enhanced Algorithm) router is industry(Intel) level routing

tool, used for routing signals. It is used for the purpose of detailed routing. QEA router can be used in different modes based on the requirement of type of routing. The basic flow of QEA router is shown in the Fig.2.6.

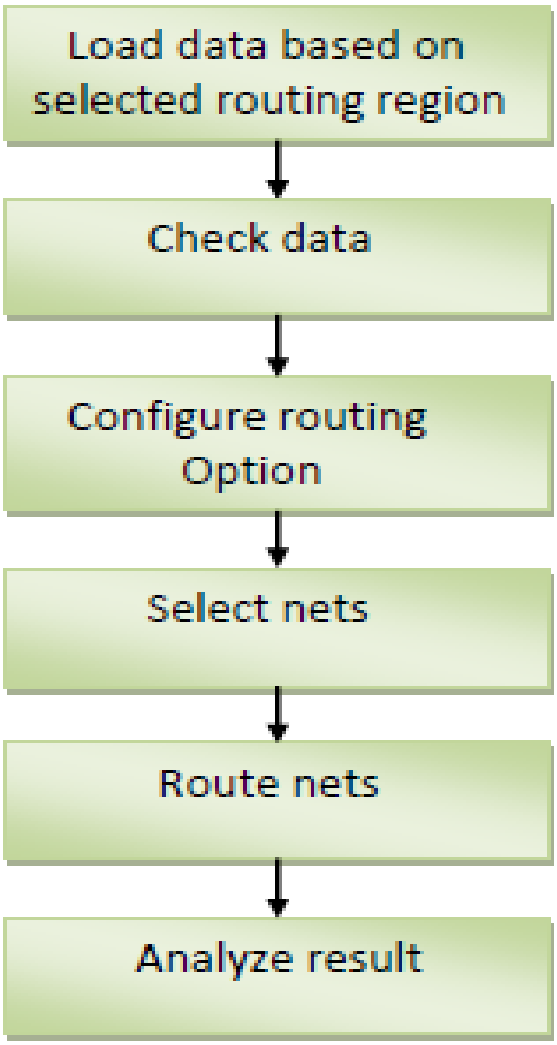


Figure 2.6: Flow diagram of QEA router

The QEA router takes input constraints from the user which can be maximum number of metal layer allowed for routing, highest metal layer, lowest metal layer, no routing region coming from the lower hierarchy, metal grid pattern for selected region, etc. This data is verified and wrong argument value is reported to user. In the following stage QEA router checks connectivity between the pin terminals and interconnect present in the given chip area. It also checks the free area for metal layers in which the routing can be done. Based on the free metal areas, the QEA router routes more than one virtual wires simultaneously. For each virtual wires, router checks all the design rules and length of the wire. At the end, QEA selects the wires having no design rule violations and minimum length. QEA router can be used in different mode based on the user settings as explained in Table.2.1.

Table 2.1: Operation modes of QEA router

Mode	Description
Route	Performs QEA routing on selected nets.
iRoute	Interactive router(QEA virtual router) to route the selected nets interactively.
Routebox	Perform QEA routing on nets in a user-selected window.If no nets are selected in the window, user will be prompted to authorize an auto run on all open nets within window.
Trunk	Performs QEA trunk generation on selected nets with no actual routing being performed.

From the above modes QEA router is mainly used in Routebox mode. As the size of the box increases the quality of routing reduces. Some of the practical examples are shown in the Fig.2.7. This examples show that QEA router is not fully optimized. QEA router has to deal with the number of nets given as an input and the congesion in the area on which we are performing the routing. It is not able to route the interconnect if metal layer congestion is very high as shown in part (c). To achieve required timing constraints, it will avoid to create long routes in lower

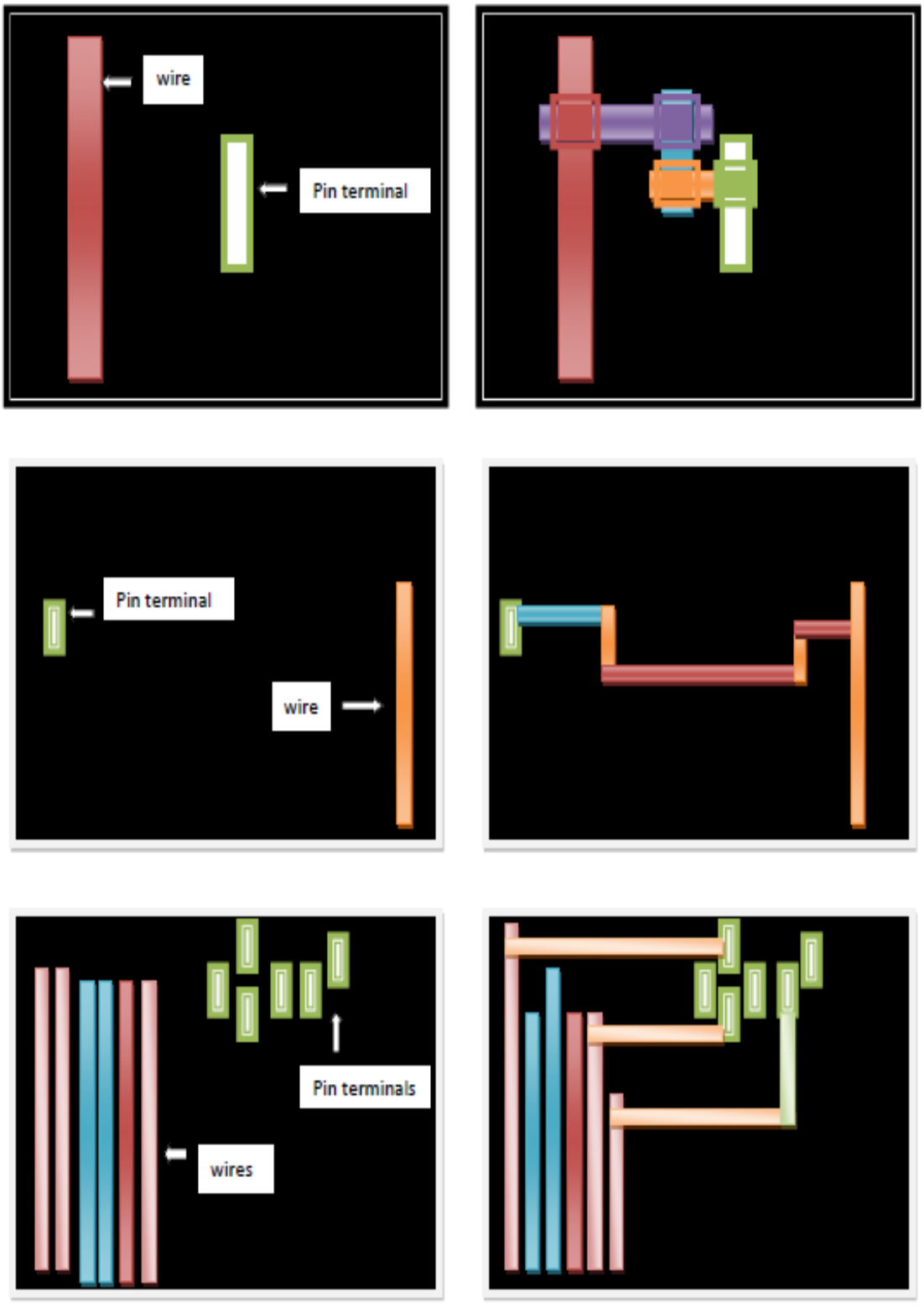


Figure 2.7: Examples of interconnect routing by QEA router.(a) Small open length, (b)Large open length, (c) Multiple interconnects

metal layers and creates jogs in higher metal layer as shown in part (b). It will affect the timing slope of that net and occupy routing resources. Part (a) shown most optimum result. To overcome all these issues advanced routing algorithm for FUB integration is designed. It does not only overcome these issues but also provides many advantages over normal QEA router.

2.6 Problems with QEA router

Problems related to QEA router are as mentioned below

- If congesion is very high then QEA router is not able to route all the nets in selected region. Generally we do not want to route clock signlas, power signlas and the nets having some other DRCs with QEA router but all these nets will be also routed by QEA router. Due to this, tracks will get occupied in selected region and there will be reduction in routing resources, hence sometimes the signals which we really want to be routed by router will be remained unrouted due to routing congesion and we will end up with manual routing. In this case rip up of unwanted routing is also required.

- Openlength between pin terminal and net to be routed is hardcoded. Due to this first some manual extention of all nets are required to make open-length within threshold value and only after that the nets will be sent to QEA router. Hence, some manual work is required before performing QEA routing which can not be preffered every time.

- The way in which currently QEA router works is not compatible with the FUB Integration Flow. In FUB Integration Flow entire FUB is given as an input while QEA router works on region selection based criteria. So while performing routing using QEA router, first we will select small region on cavas and nets present in that routing window will be sent to QEA router. This process has to be continued until we are done with the routing of entire FUB. Hence, manual involvement is continuously required while performing routing. Some routing automation is required which can take care of all

the pins inside FUB in one iteration only. In this way we can save time as well as effort.

- It is time consuming due to the way in which it works. We have to go for some iteration of routing with QEA router due to region selection based criteria.

Chapter 3

Advanced routing algorithm for FUB integration

We have seen different issues related with QEA router. When the open length of interconnect is large, QEA router is not able to produce efficient results though it is DRC violations free. It creates unnecessary jogs in the routing which is not acceptable. Apart from that the number of interconnects routed in the selected area also reduces as the area is increased. Routing all the interconnects of one FUB in the given hierarchy is necessary to reduce physical designing time. An algorithm is introduced to improve QEA router and to run QEA router over the entire FUB which is Advanced Routing Algorithm for FUB Integration. With the help of Advanced routing algorithm for FUB integration, these all issues can be overcome. It helps a lot with FUB integration. FUB integration flow includes various stages like load FUB, hook power, schvslay, gclk status, global driver shadow, opens, pin wire coverage, pin no coverage, metal fill, FUB area collisions, via density and floating power pin terminals. Advanced routing algorithm can be used after running opens stage in FUB integration flow to hook up all pins in one iteration only. In this chapter we will understand the basic flow of the algorithm, how thw problems related to QEA router can be solved, enhancements done over the algorithm and results of the algorithm.

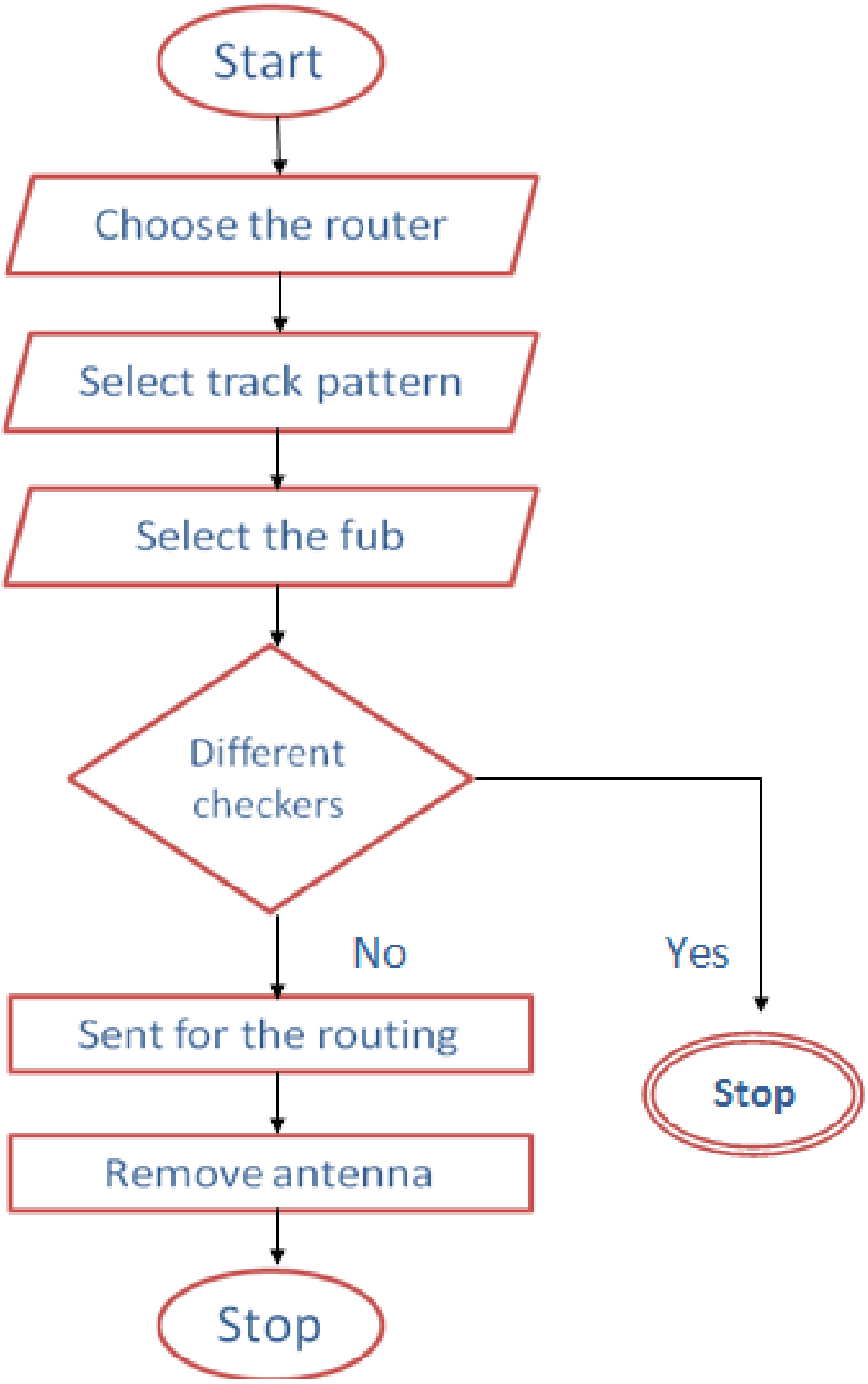


Figure 3.1: Advanced routing algorithm for FUB integration

3.1 Basic flow of Advanced Routing Algorithm for FUB Integration

Before going into details it is better to understand the basic flow on which advanced routing algorithm for FUB integration works.Consider the flow chart shown in fig 3.1 in order to have better understanding regarding Advanced Routing Algorithm for FUB Integration. Steps in which the algorithm works are as mentioned below:

- Change router settings such as mode, maximum distance to trunk, minimal metal, maximal metal, auto max metal, antenna removal, etc. based on the allowed FUB metal usage, position in layout of the chip, at the section level.
- Select track pattern appropriate to the FUB
- Select the FUB on which you want to perform integration(Routing)
- After this run Advance Routing Algorithm For FUB Integration
- It will check for the conditions mentioned below:
    1. pin is off-grid
    2. pin is power pin
    3. pin is clock pin
    4. open length is grater than the threshold value
    5. shorts present on the net needs to be routed
    6. pin has pin terminal and routing of different width
    7. pin has no terminal
- If any of the above mentioned condition is true then that pin will not be sent for routing, otherwise it will be sent for routing
- After performing the routing, antennas will be removed
- This process is continued until all the pins of the FUB are routed

3.2 How the problems associated with QEA router can be solved

The problems related to QEA router can be solved with the help of Advanced routing algorithm for FUB integration as mentioned below:

- QEA router has to deal with mainly two things; input data (number of nets we are giving as an input to be routed by router) and congesion (availability of routing resources). In this routing algorithm first we are performing different checks on the pin terminal and based on the result of a perticular check either we will send that pin terminal for the routing or eliminate it from the routing. Hence we are further filtering out the nets that should be sent to router. As a result of this, amount of input data given to the router will get reduced and routing probability of a perticular pin terminal will get increased. Hence there will be improvement in results.

- In this routing algorithm, openlenth between pin terminal and net to be routed is not hardcoded. We can perform routing on pin terminals having openlength value of our choice. This eliminates manual work required before routing.

- In this algorithm, we are giving entire FUB as an input. Hence, it is compatible with FUB Integration Flow which also uses entire FUB as an input. Even this algorithm can be integrated with FUB Integration Flow after the ”open” stage.

- As FUB itself can be given as an input, we are saving time as well as effort.

3.3 Enhancements done in Advanced Routing Algorithm for FUB Integration

We have already seen how the problems related to QEA router can be solved with the help of Advanced Routing Algorithm for FUB Integration. It is even enhanced

further more in order to save human efforts and time. The enhancements done are as mentioned below:

- With the help of this routing algorithm, routing over any number of FUBs can be done at a time. It will run routing in serial mode over all the FUBs which we have selected. This entire routing run can be launched in parallel with the current PARADE session. Once routing is done, it can be copied to current session. With the help of this utility overnight run is possible which can save lot of time. Consider the figure 3.2 for better understanding. If we want to run routing algorithm on three FUBs (FUB A, FUB B, FUB C) then those all FUBs will be selected together at a time and then algorithm will be invoked. It will perform routing over all the FUBs and once it is done, routing can be copied to current session. With the help of this utility we are saving time.

- This routing algorithm is integrated with power router. To understand the work of power router please consider the figure shown. As shown in figure 3.3, one FUB is instantiated for four times in a section. Among these four instances, a particular pin a is having logical connection only in one instance while in rest of the instances it is floating. Power router will connect those floating pins to vss. This utility will help in filtering out real opens over the section without running actual integration flow over the all the FUBs of the section. Whenever we are running the Advanced Routing Algorithm for FUB Integration, we have to passed the value of the argument. Based on the value of the argument, either signal routing will be performed with power routing or it will be performed without power routing. Hence it is user defined.

- Quality matrix is developed. It will be displayed once the routing over the FUB is done. It will show the division of openlength between pin terminal and net before invoking the routing algorithm and the total wirelength which algorithm has put in between pin terminal and net to close the open. From this value we can know the quality of routing. If routing is bad for a particular pin then it can be rip up and route again before checking in the data. With the help of this utility we can end up with quality routing.

With the help of these enhncements, we are saving time and human effoert. We will end up with good quality of the routing which is very important because at the end we are not only concerned about the number of nets which are routed but we are even more concerned about the quality of the routing. With the routing of good quality we saving resouces as well as cost. Hence performance can be improved.

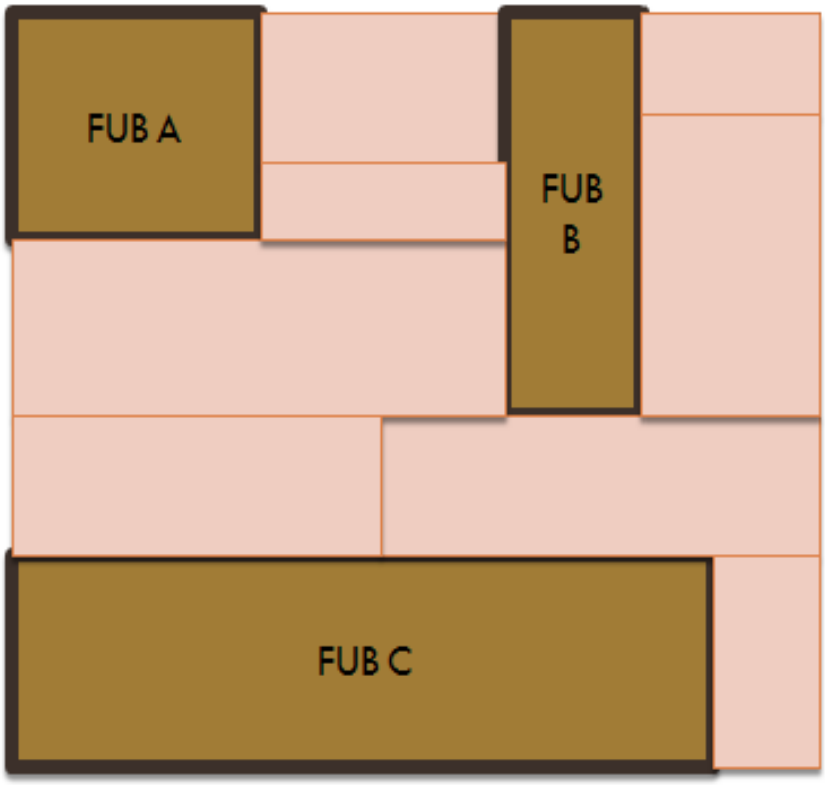


Figure 3.2: How to invoke Advanced Routing Algorithm over more than one FUB

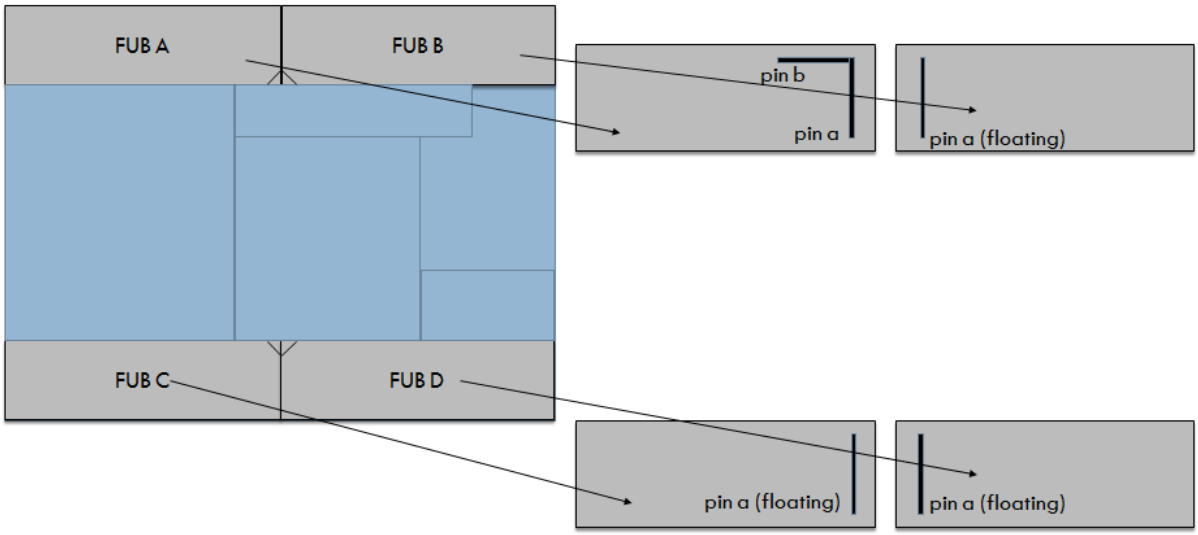


Figure 3.3: Need of power router

3.4 Results and output of the routing algorithm

Based on the algorithm proposed TCL script is built. Inputs given to the algorithm are mode of router, track patterns and FUB.

This algorithm can be invoked using command line. We need to pass two arguments; one is the threshold value of openlength between pin termianl and net to be routed and the another one is for running power router. If the openlength is less than the threshold value then only that perticular pin terminal will be sent for routing. Once routing is done, as a result of the routing algorithm three G-Tables will be displayed over the screen. It is explained in details as mentioned below:

- First G-Table is as shown in Figure 3.4. It contains list of all the pin terminals present in the fub, net associated to respective pin terminal and the result of the prechecckers performed on the perticular pin terminal. All the pin terminals inside the FUB can be sorted based on the prechecckers. We can filter out all the pin terminals which are sent for routing, pin terminals which are associated to clock nets, pin terminals which are associated to power nets, pin terminals having shorts present on the associated nets, pin

terminals having openlength grater than the threshold value, pin terminals having different width than the width of associated nets etc. If the width of pin terminal is not same as the width of associated net and it is sent for routing then some extra jogs will be created which is not really required. It will impact the timing slope of that net as well as the available routing resources. That is why generally it is avoided to send such kind of pin terminals for routing. Generally for the routing of power pins we are using different algorithm and routing of clock nets is done manually with some routing iterations to meet arrival time requirements. If any short is present on the net then it will not be sent for routing to avoid rip up and rerouting. Hence, this G-Table will give the basic idea regarding status of all the pin terminals inside the FUB after performing routing.We can know exactly how many and which pins are sent for routing.

- Second G-table which will be poped up as an output is as shown in Figure 3.5. It contains list of all the nets which were sent for routing, type of the perticular net (whether that net is input, output or local), status of the net once routing is done (whether that net is routed or still it is having an open) and laddering status of the perticular net. Hence, this G-Table gives detailed information about all the nets which were sent for routing.

- Third G-Table which will be poped up as a result of routing algorithm is as shown in Figure 3.6. It gives the openlength before performing routing, respective wirelength added by the router and division of these two value for each and every net which was sent to routing algorithm. Some threshold is there on the value of divison. If the value is grater than the threshold value then that routing quality is very bad. In such cases routing of that perticular net will be ripped up and that net will be routed again. Hence, with the help of this G-Table we can know the routing quality of all the nets which were sent for routing.

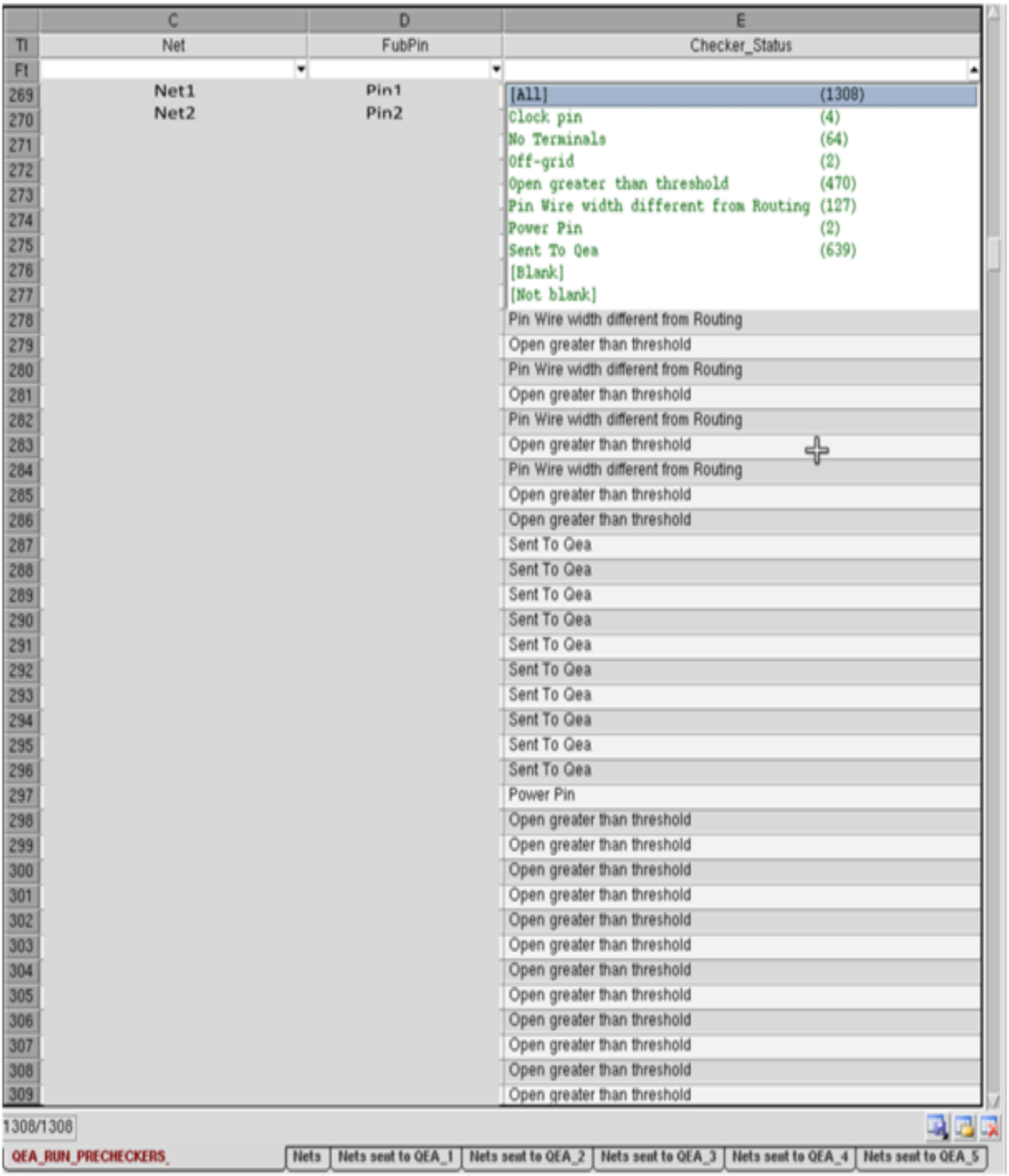


Figure 3.4: G-Table having net name, FUB pin and pre-checker status

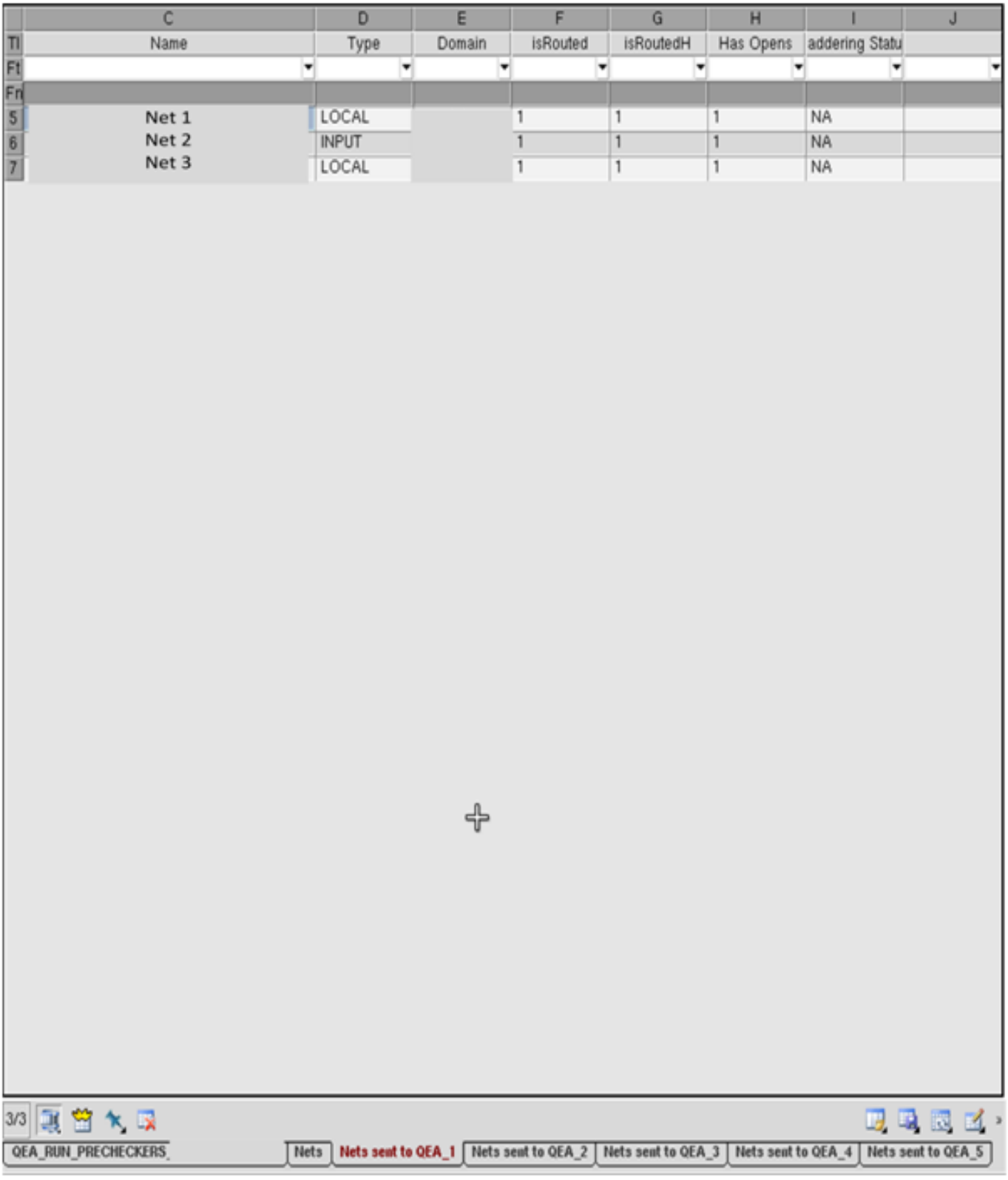
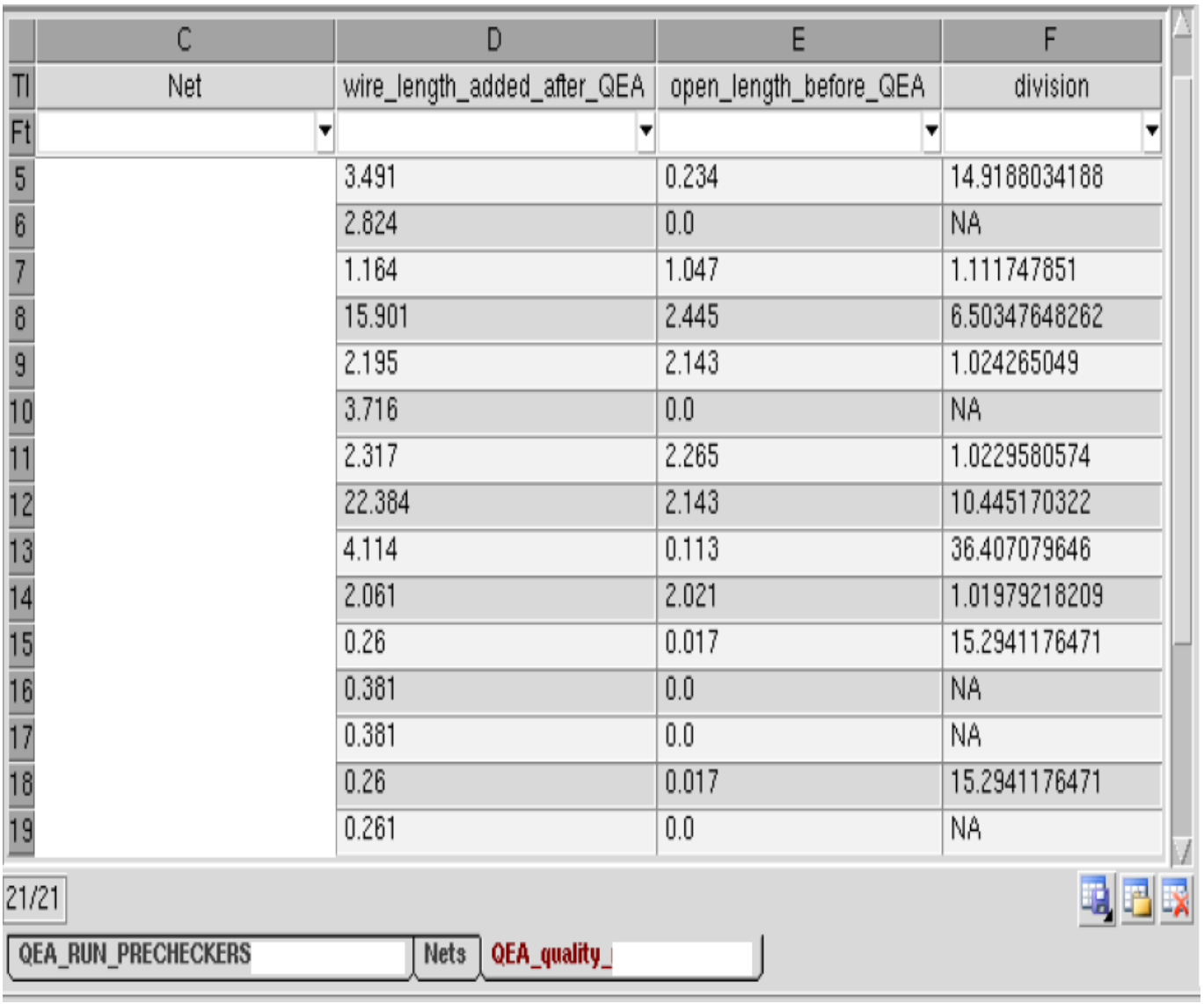


Figure 3.5: G-Table having list of nets which are sent to QEA route



	C	D	E	F
TI	Net	wire_length_added_after_QEA	open_length_before_QEA	division
Ft				
5		3.491	0.234	14.9188034188
6		2.824	0.0	NA
7		1.164	1.047	1.111747851
8		15.901	2.445	6.50347648262
9		2.195	2.143	1.024265049
10		3.716	0.0	NA
11		2.317	2.265	1.0229580574
12		22.384	2.143	10.445170322
13		4.114	0.113	36.407079646
14		2.061	2.021	1.01979218209
15		0.26	0.017	15.2941176471
16		0.381	0.0	NA
17		0.381	0.0	NA
18		0.26	0.017	15.2941176471
19		0.261	0.0	NA

Figure 3.6: Quality matrix

Conclusion

Hence, advanced routing algorithm for FUB integration reduces the overall run time for the detailed routing. It also gives us flexibility to choose the open-lenth for the pin terminal and net to be routed which eliminates the manual work required before sending the nets for routing. It is also compatible with the flow which is used by industry for FUB integration. QEA router is not able to route all the nets sent to it if congesion is very high but the way in which advanced routing algorithm for FUB integration works, routing probability of a perticular pin terminal increases even in the area having congesion. At the end of routing, this algorithm gives the quality matrix from which we can know the routing quality of each and every net. Hence, we will end up with quality routing. This routing algorithm reduces run time as well as efforts required and improves the quality of the detailed routing.

References

[1] Andrew B. Kahng, Jens Lienig, Igor L. Markov, Jin Hu, VLSI Physical Design: From Graph Partitioning To Timing Closure. Springer Dordrecht Heidelberg London, New York,2011, vol. 1

[2] N. A. Sherwani, Algorithms for VLSI Physical Design Automation. Kluwer Academic Publishers, Norwell, MA, USA, 2005, vol. 2.

[3] Sadiq M. Sait, Habib Youssef, VLSI Physical Design Automation: Theory and Practice. World Scientific Publishing, Singapore, 1999, vol. 6

[4] C.H. Hsu, H.Y. Chen, and Y.W. Chang, Multilayer global routing with via and wire capacity considerations, IEEE Trans. on Computer- Aided Design of Integrated Circuits and Systems, vol. 29, no. 5, pp. 685 696, May 2010.

[5] M. Tang and J.-F. Mao, Optimization of global interconnects in high performance VLSI circuits, 19th International Conference on VLSI Design, 2006. Held jointly with 5th International Conference on Embedded Systems and Design, pp. 611, Jan 2006.

[6] https://dts.intel.com

[7] https://sharepoint.amr.ith.intel.com