UNIVERSITY OF CALIFORNIA, SAN DIEGO

Off-Chip Wire Distribution and Signal Analysis

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Computer Science

by

Rui Shi

Committee in charge:

Professor Chung-Kuan Cheng, Chair
Professor Peter M. Asbeck
Professor Fan Chung Graham
Professor Ernest Kuh
Professor Tajana Simunic Rosing

2008

.

The dissertation of Rui Shi is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2008

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First of all, I would like to express my deep and sincere gratitude to my advisor, Professor Chung-Kuan Cheng for his continuous support. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating. His wide knowledge and his logic way of thinking have been of great value for me. His understanding, encouraging and personal guidance have provided a good basis for the present thesis. I could not have imagined having a better advisor for my PhD, and without his common-sense, knowledge, perceptiveness and cracking-of-the-whip I would never have finished.

I wish to express my sincere thanks to Professor Ernest Kuh of University of California, Berkeley. His supervision and constructive advice on my research work is extremely important to me.

My thanks and appreciation goes to my dissertation committee members, Professor Fan Chung Graham, Professor Peter M. Asbeck and Professor Tajana Simunic Rosing for technical discussions and their advices and reviews of this dissertation.

I am indebted to my many student colleagues for providing a stimulating and fun environment in which to learn and grow. I am especially grateful to Haikun Zhu, Peng He, Yi Zhu, Ling Zhang, Zhengyong Zhu, Hongyu Chen, Bo Yao, Shuo Zhou, Jianhua Liu, Renshen Wang, Wanping Zhang and many others.

I am grateful to Dan Beckman and Dawei Huang of SUN Microsystems. They provided me valuable technical discussions and support in escape routing project.

I wish to thank my entire family for providing a loving environment for me. Especially, I would like to give my special thanks to my husband Dawei Sun, whose patient love enabled me to complete this work, my thanks to my child, Kevin W. Sun for giving me happiness and joy.

Lastly, and most importantly, I wish to thank my parents, Junheng Shi and Jirong Wu. They raised me, taught me, and loved me. To them I dedicate this thesis.

Chapter 2, in part, is a reprint of the paper "Layer Count Reduction for Area Array Escape Routing" co-authored with Hongyu Chen, Chung-Kuan Cheng, Dan Beckman and Dawei Huang in the proceedings of 2005 International Conference and Exhibition on Device Packaging. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of the paper "Efficient Escape Routing for Hexagonal Array with high I/Os density" co-authored with Chung-Kuan Cheng in the proceedings of 43rd ACM/IEEE Design Automation Conference 2006. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the paper "Efficient and Accurate Eye Diagram Prediction for High Speed Signaling" co-authored with Wenjian Yu, Yi Zhu, Chung-Kuan Cheng and Ernest S. Kuh in the proceedings of IEEE/ACM International Conference on Computer-Aided Design 2008. The dissertation author was the primary investigator and author of this paper.

VITA

1999	B.E. in Electronic Engineering Tsinghua University, Beijing, China
2002	M.S. in Computer Science and Technology Tsinghua University, Beijing, China
2008	Ph.D. in Computer Science University of California, San Diego

PUBLICATIONS

R. Shi and C.K. Cheng, Efficient Escape Routing for Hexagonal Array with high I/Os density, DAC 2006, pp1003-1008

R. Shi, H. Chen, C.K. Cheng, D. Beckman, and D. Huang, Layer Count Reduction for Area Array Escape Routing, International Conference and Exhibition on Device Packaging 2005

R. Wang, R. Shi, and C.K. Cheng, Layer Minimization of Escape Routing in Area Array Packaging, ICCAD 2006, pp.815-819

Z. Zhu, R. Shi, and C.K. Cheng, An Unconditional Stable General Operator Splitting Method for Transistor Level Transient Analysis, ASP-DAC 2006, pp.428-433

H. Zhu, R. Shi, C.K. Cheng, and H. Chen, Approaching Speed-of-light Distortionless Communication for On-Chip Interconnect, ASP-DAC 2007, pp.684-689

H. Zhu, R. Shi, H. Chen, C.K. Cheng, A. Deutsch, and G. Katopis, Distortion Minimization for Packaging Level Interconnects, EPEP 2006, pp.175-178

H. Chen, R. Shi, C.K. Cheng, and D.M. Harris, Surfliner: A Distortionless Electrical Signaling Scheme for Speed of Light On-Chip Communications, ICCD 2005, pp.497-502

W. Zhang, L. Zhang, R. Shi, H. Peng, Z. Zhu, C.E. Lew, M. Rajeev, S. Toshiyuki, I. Noriyuki and C.K. Cheng, Fast Power Network Analysis with Multiple Clock Domains, ICCD 2007, pp.456-463

FIELDS OF STUDY

Major Field: Computer Engineering
Studies in VLSI CAD
Professor Chung-Kuan Cheng

ABSTRACT OF THE DISSERTATION

Off-Chip Wire Distribution and Signal Analysis

by

Rui Shi

Doctor of Philosophy in Computer Science

University of California, San Diego, 2008

Professor Chung-Kuan Cheng, Chair

With the steady progress of high performance electronic systems, the complexity of the electronic systems grows continuously and new features like high-speed, low power and low cost, become the key issues. The interconnection in the electronic systems distributes the power, clock and transfers the electrical signals among numerous components. As the feature size of microelectronic technology becomes smaller, the design and analysis of the interconnection are more important for the performance and cost of the system.

The escape routing design is a new important problem in the interconnection design because the number of chip/package input/output (I/O) pins has continuously been growing for high-speed high-performance system. The traditional escape routing strategy and several improvements, pins/pads special placement and routing resource exploration, could not handle large number of I/Os efficiently and usually require increased cost. We introduce a novel concept, escape sequence, develop efficient escape routing strategies based on the new concept and model the escaping wires using 3D extraction.

(1) For square grid array, we formulate and solve a maximum flow problem to analyze the escape bottleneck in area array. The layer count can be decreased dramatically by

optimizing the I/O escape sequence. Two new escape routing strategies, central triangular escape routing and two-sided escape routing, are proposed and the number of escape routing layers could be reduced dramatically. We implement an automatic escape routing program to verify our analysis and to compare different escape sequence strategies. The escaping wires could be modeled as frequency dependent RLGC circuit for signal performance analysis.

(2) For hexagonal area array, we analyze its preponderant properties that the hexagonal array could increase the density of I/Os in the array remarkably. We propose three escape routing strategies for the hexagonal array: column-by-column horizontal escape routing, two-sided horizontal/vertical escape routing, and multi-direction hybrid channel escape routing. We can escape I/Os in the hexagonal array in the same or less number of routing layers compared with square grid array. Therefore, we could reduce the number of escape routing layers as well as increase the density of I/Os.

The eye diagram prediction is an important problem in the interconnection analysis because an eye diagram provides the most fundamental and intuitive view to evaluate the signal quality of high-speed communication. The traditional method to obtain the eye diagram involves performing a time domain simulation. It requires a very long simulation time and usually could not accurately characterize the communication systems because the limited length pseudorandom bit sequence (PRBS) as the input stimulus. Several eye diagram prediction methods, analytical techniques, analysis based on unit pulse response and analysis based on step response, have their application constraints because of their assumptions and usually could not handle general systems. We introduce an efficient and accurate method, the accumulative prediction method, to predict the eye diagram for high-speed signaling systems. We use the step response of the signaling system to extract the worst-case eye diagram, including the eye opening and timing jitter. Furthermore, this method generates the input data

patterns which produce the worst-case inter-symbol interference. The main advantage is that this general-purpose method can handle signals with asymmetric as well as symmetric rise/fall time. Furthermore, the complexity of the proposed method is linear O(n), where n is the number of time points of step response. This method can be applied to analyze the signal performance in escape routing. It is very useful to analyze various high speed signaling systems.

Chapter 1

Interconnection

in High-speed Electronic System

Electronic system permeates our daily life as well as the advanced technology applications. As the microelectronic technology develops, the complexity of electronic systems grows rapidly. Typically, a system consists of a hierarchical structure and the interconnection plays a very important role in the system. The routing stage in the design process implements the interconnection with specific objectives. New features like high-speed, low power and low cost, become the key issues.

1.1 Overall View of Electronic System

The electronic systems are usually built in a hierarchical structure. The integrated circuits (ICs) are at the heart of electronic systems, a package is the housing of the ICs and high level boards assemble the packaged components together to form a system.

What is integrated circuit? An integrated circuit is a thin chip consisting of semiconductor devices, mainly transistors, as well as passive components like resistors. The integrated circuit was made possible by min-20th-century technology advancements in

semiconductor device fabrication. Only a half century after their development was initiated, integrated circuits have become ubiquitous. Computers, cellular phones, and other digital appliances are now inextricable parts of the structure of modern societies. That is, modern computing, communications, manufacturing and transport systems, including the Internet, all depend on the existence of these circuits.

As of 2004, typical chips are of size $1cm^2$ or smaller, and contain millions of interconnected devices. The growth of complexity of integrated circuits follows a trend called "Moore's Law". Moore's Law in its modern interpretation states that the number of transistors in an integrated circuit doubles every two years.

The packages cover the ICs to assure environmental protection for the IC chips and to provide complete testing and high-yield assembly to the next level of interconnection. The electronic packaging establishes the interconnection between the chip and package and package to board, etc. and maintains a suitable operating environment for the ICs to function effectively and efficiently. There are several reasons why an IC is packaged, the primary functions are to provide electrical interconnection (both power and signal distribution), allow for efficient heat dissipation (cooling), and physically protect the IC.

Higher-level assemblies (e.g. boards, cards, or flexible substrates) are needed to connect packaged electronic devices because of the complexity of current electronic systems. Usually, the IC packages will be connected to each other through printed circuit board (PCB). And PCBs provide mechanical support as well as electrical interconnect. A PCB interconnects electronic components using flat conductive traces laminated onto a non-conductive substrate. Usually a PCB is built as a stack of layer pairs. Each pair starts as an insulating sheet with copper deposited on one or both sides. The copper sides of the sheet are first etched with different wiring patterns. Then the sheets are stacked into a sandwich separated by insulating

material. Small holes are drilled into the board and are plated with metal, so that electrical contact is made with each layer that has copper left at the hole-location.

The typical assembly hierarchy of the electronic system is shown in Figure I. 1 [15]. Chip level (zero level) packaging includes chip metallization, and provisions for chip-package interconnection. The first level packaging is the assembly of chip and substrate (organic or ceramic package) to form a single or a multiple chip module. In some cases chips may be directly attached to the card (chip-to-board, COB). The second level packaging in the assembly of chip modules and other components on PCBs. The third level packaging differs; depending on the sophistication of the system it may involve several PCBs plugged into a motherboard.

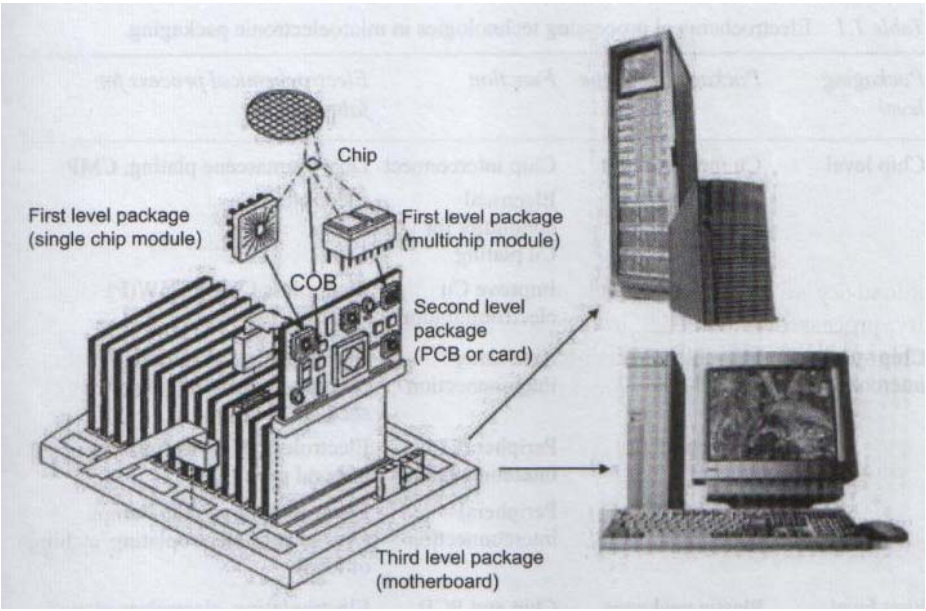


Figure I. 1 Typical assembly hierarchy of the electronic system.

1.2 Interconnection in Electronic System

The interconnection connects many kinds of electronic components to form a complicated system, as shown in Figure I. 2. They distribute the power and clock for all the

components and transfer the electrical signals among those components simultaneously. They work like the interleaved roads in our real world. IC chips are connected through connections and traces in the packages and boards. And inside IC chips, basic functional devices are implemented in the semiconductor substrate (silicon) and then are connected through multiple routing layers on top of the substrate.

The interconnection design directly influences the performance and cost for the whole electronic system. Along with the scaling of feature size in IC chips, the performance of on chip interconnection becomes more and more important for the circuit. The wiring delay becomes comparable with gate delay and start to dominate the delay of the circuit. The interconnection distributing clock signals consume roughly one fourth of the total power consumption of the IC. And the crosstalk (coupling effects) between signal wires becomes increasingly critical to on-chip timing and even functionality. For the package and board design, chip-to-chip interconnection is the fundamental objective. Obviously, the performance of the interconnection inside the packages and on the board will constrain the performance of the whole system. Also for the packages and boards, the number of interconnection layers and the technology for different interconnection style are directly related to the cost of the system.

The routing problem is one of the most important phases in the VLSI and PCB design. In the routing stage, the interconnection is implemented under specific objectives considering performance and cost. The electronic design automation tools help the designer to implement the interconnection efficiently and effectively. For the routing stage, the given information includes the placement of all components with physical locations, the net-list with physical locations for all terminals and related technical parameters, such as the number of routing layers, the wire width, the wire spacing and so on. Here, “terminal” means the endpoint of the interconnection, “net” represents a collection of terminals all of which are, or must be,

connected to each other electrically and “net-list” is the list of names of all nets and their terminals. After routing stage, all the interconnection will be routed using wires on different layers and the physical layout for all the interconnection will be output finally.

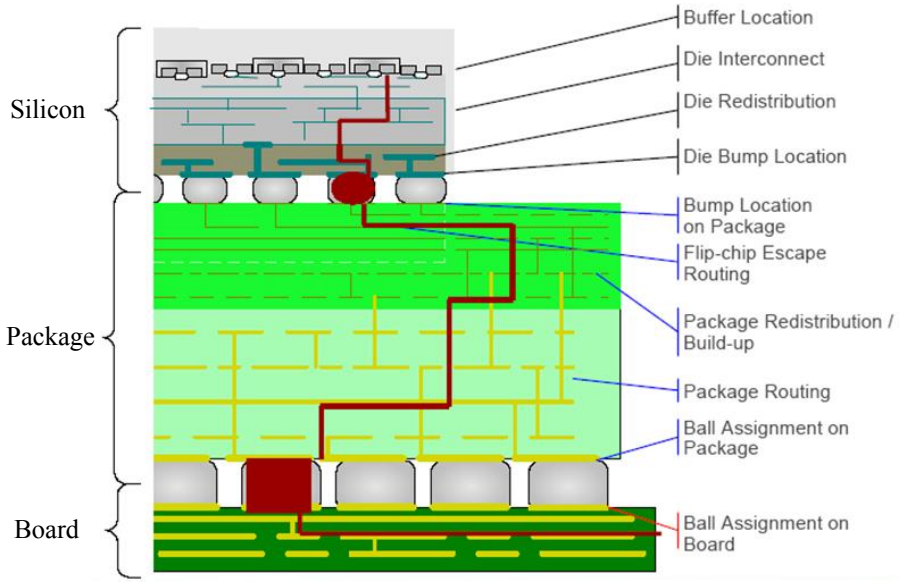


Figure I. 2 Interconnection in the electronic system.

1.3 Escape Routing

1.3.1 Background

The complicated practical electronic systems usually contain a great number of circuits and interconnections. A number of chips will be interconnected on organic or ceramic first-level packages. The electrical connections between the chip and the package are referred to as chip-level/first-level interconnections. Currently, there are three principal technologies in use: wire-bonding (WB), tape automated bounding (TAB) and flip-chip solders connection (C4). Because the complexity of current systems, the packaged electronic devices need to be connected to higher-level assemblies (e.g. boards, cards, or flexible substrates). These interconnections are referred to as package-to-board/second-level interconnections.

IC packaging technology is undergoing rapid change with the move to nanometer-scale ICs, becoming the critical link in the system interconnect flow. Without this key element of a comprehensive flow (silicon to package to board) the end result can easily be silicon that is difficult or expensive to implement into a system. Package and board design are facing many new challenges, such as high speed interconnections, signal integrity, cost efficiency and so on.

As the feature size of microelectronic technology becomes smaller, the complexity of electronic systems grows proportionally. With the steady progress of high performance electronic systems, the number of chip/package input/output (I/O) pins has continuously been growing. According to Rent's rule [3], the number of I/O signals of a module is a function of the number of gates in it:

$$N_p = K_p N_g^{\beta} \tag{I.1}$$

where, N_p is the number of external signal connections, N_g is the number of logic gates, K_p is a constant, and β is the Rent's rule constant which depends significantly on the kind of module considered. Today's high-performance ICs exhibit upwards of 2,000 I/O pins and require packages that sometimes exceed 100 layers and will go onto boards with more than 50 layers. Increasing demand for high I/O pins count prompts the packaging industry to explore new technologies, such as area array interconnection [18][25][34][37], CSP (chip scale packing), BGA (ball grid array) [38], and so on. Area array packaging contains an array of pads, pins, or solder-balls located directly underneath a unit of package area. In order to connect the area array I/Os to the next level assembly, all the I/O pins need to break out to the outside. As shown in Figure I. 3, the wires for breaking out I/O pins are referred to as Escape Routing [18][25][34][37].

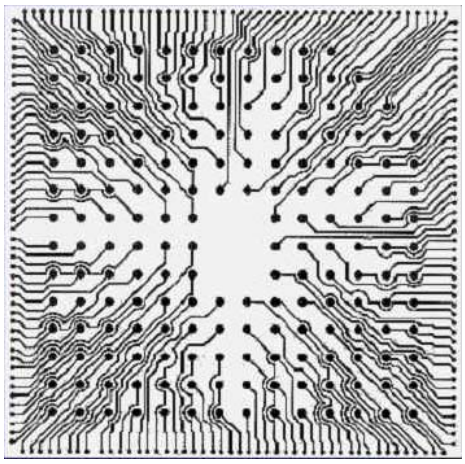


Figure I. 3 Escape routing connects I/O pins to outside.

Currently, the high count and density of I/Os require multilayer for escape routing. These interconnections directly affect the cost and performance of the whole electronic systems. The increase in the number of escape routing layers will result in a higher manufacturing cost of substrates used as a package or a board. Thus an efficient and effective escape routing strategy which achieve high performance with low cost will greatly benefit the electronic product.

1.3.2 Problem Description

In the escape routing problem, I/Os inside the area array are the objects; the corresponding pads are the obstacles and the spaces scattering among the pads in the area array are the routing resources. Fundamentally there are three basic guidelines which are described blow.

(i) Spacing between two wires and the width of the wire. As shown in Figure I. 4, the wire width is W and the pitch is $(W+S_w)$, where S_w is the spacing between the two consecutive edges of the wires.

(ii) Minimum distance S_p between the edge of the pad and the edge of any metal, as shown in Figure I. 4.

(iii) Diameter of the pad, D, and pitch, P, between two consecutive pads, as shown in Figure I. 4.

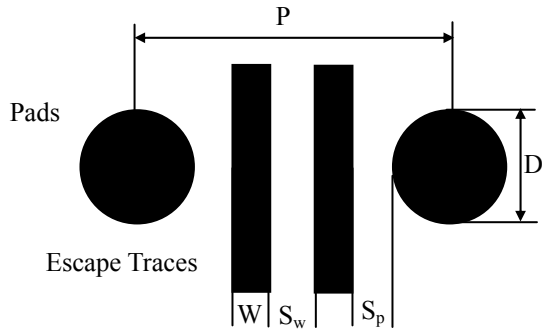


Figure I. 4 Fundamental parameters of package design.

Usually the spacing between two wires S_w and the spacing between pad and wire S_p are identical, we can use S to represent the spacing.

1.3.3 Strategies for Improving Escape Routing Performance

1.3.3.1 Traditional Method

Connecting the I/O pins in the area array to the outside, the traditional method is very straightforward. It breaks out the pins located in the outside rows/columns first as shown in Figure I. 5. For one routing layer, the escape routing wires for pins located inside will go through outside pins/pads. The spacing between two consecutive pads constrains the number of wires going through, thus limit the number of pins escaped for one layer. In this conventional geometry of pad and wire, increasing the number of wires between pads can only be realized by decreasing wire width and spacing. While these dimensional changes expose the substrate to cost, yield and reliability issue. Guinn et al. [17] have shown the trends in required area for packaging various devices depending on their number of I/Os, and pointed out that the progress in feature dimensional reduction technology of PWB (printed wiring board) is too slow compared with that of ICs.

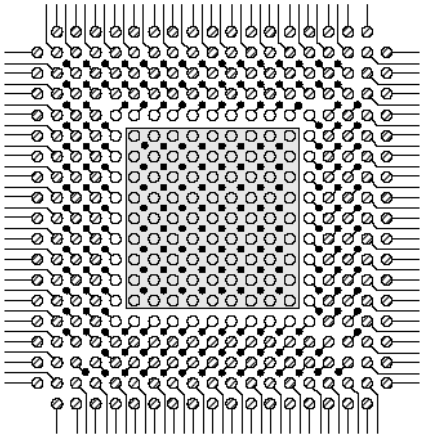


Figure I. 5 Traditional escape routing.

1.3.3.2 Special Placement of the Pins/Pads

The usual area array is a regular grid array, while this kind of methods changes the placement of pins/pads in the array to improve the escape routing performance [18][20][31].

Hirt et al. [20] analyzed the escape routing with multilayer structure according to the conventional procedure from a concentrated I/Os of area flip chip and BGA. The escape routing wires are modeled to connect the pins/pads with a VIA to an inner layer and then go out. The VIAs for pins in the outmost rows are arranged in staggered rows and the VIAs for pins inside are located inside the array. This VIAs arrangement decreases the area overhead for escape routing and makes as many rows as possible escaped on each layer to minimize the number of layers needed.

Gasparini et al. [18] suggested a specific placement of bumps for C4 packages to minimize the number of package layers. The pins/pads are placed in staggered form to maximize the number of pins that can be routed in a single routing layer.

Assume the distance between two consecutive bumps in the outmost row allows n number of wires going through, and then n bumps will be placed inside in staggered form. Figure I. 6 shows the bumps placement and escape routing wires for three different values of n.

The placement with large n will increase the value of δ, which means the depth of the bumps group from edge to inside. No bumps will be placed inside the region surrounded by the bumps group and the array edge. This region will be wasted and the larger the value of n, the bigger the wasted area. So for a given package design, there is an optimum value of n that can increase the number of bumps routed in one layer with reasonable group depth δ.

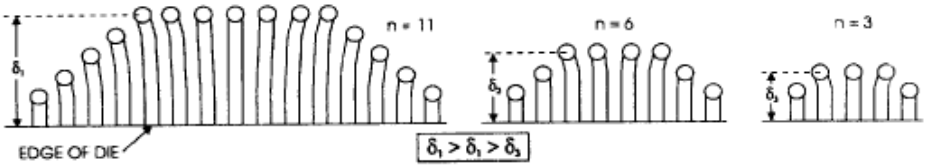


Figure I. 6 Formation of groups of pins for three different values of n.

Each group can be routed in a single layer.

By incorporating this design method into the placement of the signal I/O, Andrews et al. [1] B. Analui, J. Buckwalter, and A. Hajimiri, "Data-dependent jitter in serial communications," *IEEE Trans. Microwave Theory Tech.*, Vol. 53, No. 11, pp. 1841-1844, Nov. 2005.

[2] have succeeded to reduce the number of layers of an organic substrate for flip chip packaging.

However, this special placement of bumps requires some changes in pad allocation, not only on the substrate, but also on the corresponding IC. To realize a universal method applicable covering package to board interconnection, in which footprint is generally standardized, it is necessary to achieve the reduction of the layer counts without changing the pad allocation on the interface.

1.3.3.3 Routing Resource Exploration

Horiuchi et al. [21] studied the effects of routing manner on the layer count and

suggested a preferential routing which creates specific pad geometry resulting in a high wiring efficiency. Since nothing changed in both manufacturing processes and materials, this method is very practical for assembly of the high I/O flip chips, CSPs, and BGAs.

The preferential routing breaks out pins following a triangular array outline as shown in Figure I. 7. The number of pins escaped on one layer is increased because of the "hybrid channel" with high routing capacity.

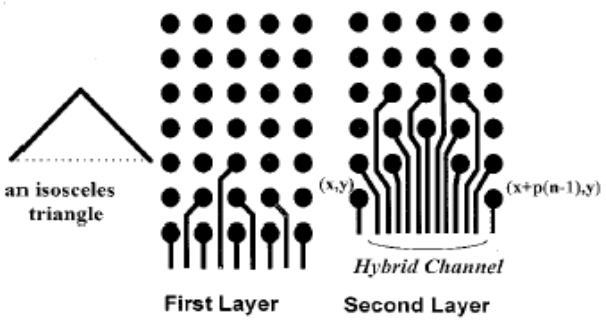


Figure I. 7 The preferential routing structure.

The size of hybrid channel is 5.

Obviously, the number of wires going through the "hybrid channel" will be increased as n increases. While on the other hand, the number of "hybrid channels" will be decreased for a certain size area array. So for a given package design, there is an optimum value of n that can achieve the best improvement compared with traditional method.

1.4 Signaling Analysis

1.4.1 Background

Signals are communicated through interconnection in the electronic systems. They are propagated among function cells in the chip and among chips in the boards. The signal performance [7][10][30][33], e.g. delay, bandwidth and power, is used to evaluate the

interconnection design. Propagation delay is a measurement of the time for a signal to reach its destination. As the feature size of microelectronic technology becomes smaller, the interconnection delay becomes comparable with the gate delay and start to dominate the delay of the circuit. Reducing the signal delays in digital circuits allows them to process data at a faster rate and improve overall performance. Bandwidth is the difference between the upper and lower cutoff frequencies of the communication channel and is typically measured in hertz. It often refers to a data rate measured in bits/s in digital system because the digital data rate limit (or channel capacity) of a physical communication link is related to its bandwidth in hertz. Power consumption is the energy consumed by the digital circuits. It is limited primarily by power dissipation concerns. So low power consumption is a big issue in high speed integrated circuit design. Moreover, the distortion, noise and interference on signal waveforms will also constrain the system performance in high speed signaling system.

An eye diagram [4] provides the most fundamental and intuitive view to evaluate the signal quality of high speed communication. The eye diagram is an oscilloscope display in which a digital signal at the receiver side is repetitively sampled to get a good representation. It's created by taking the time domain signal and overlapping the waveform for a certain time window of multiple symbol periods, as shown in Figure I. 8. The eye diagram is a useful and intuitive technique for the qualitative analysis of signal performance in digital transmission. Eye diagrams show parametric information about the signal – effects deriving from physics. Several system performance measures can be derived from the eye diagram, e.g. the rise times, fall times, jitter, overshoot, eye opening and so on [5][29].

The traditional method to obtain the eye diagram involves performing a time domain simulation [26]. Due to the long simulation times, designers usually use a limited length pseudorandom bit sequence (PRBS) as the input stimulus. The eye diagram measurement

derived from a limited length PRBS usually is better than the worst-case for many high speed signaling systems. Fast and accurate eye diagram analysis methods are very helpful to analyze the system performance.

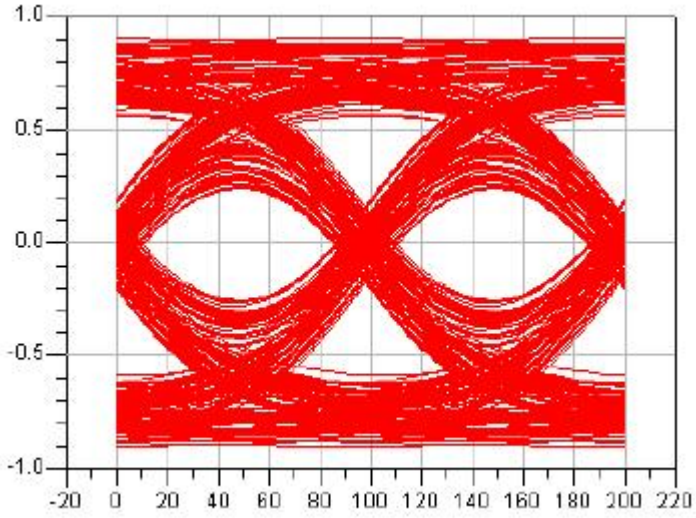


Figure I. 8 Eye diagram.

1.4.2 Eye Diagram Prediction

1.4.2.1 Analytical Technique

The analytical techniques estimate the eye diagram using analytical formula based on certain assumptions. Usually, it can analyze the eye diagram very fast because the analytical formula is used. However, the estimation accuracy is dependent on the approximation of the analytical formula and its applications are limited due to the assumptions for analytical formula derivation.

In [32], Tsuchiya et al. proposed an analytical formula to estimate the maximum eye-opening voltage. The circuit model of resistive terminated transmission-lines they investigate is shown in Figure I. 9. They model the waveform at the receiver side of transmission-lines by the PWL waveform model shown in Figure I. 10. They assume that the

driver of the interconnect achieves impedance matching and the voltage at the receiver side reaches V_{max} when the time $2t_{tof}$ passed after rising. For simplicity, the supply voltage V_{dd} is assumed as 1V.

The maximum eye-opening voltage V_{eye} is expressed as

$$V_{eye} = \begin{cases} 2V_r + (V_{\max} - V_r)\dfrac{T - t_r}{t_{tof}} - V_{\max} = \left(\dfrac{Z_n}{1 - 2\log n + Z_n} - \dfrac{nZ_n}{Z_n + 1}\right)\left(\dfrac{T - t_r}{t_{tof}} - 1\right) + \dfrac{nZ_n}{Z_n + 1} & T\text{-}t_r < 2t_{tof} \\ V_{\max} & T\text{-}t_r > 2t_{tof} \end{cases} \quad \text{(I.2)}$$

The voltage V_r is rise voltage that is determined from the attenuation and the termination of the interconnect. The voltage V_{max} is the voltage level when the continuous "1" is input to the interconnect. The time t_r is the transition time of input pulse and period T is the minimum width of input pulse. The time t_{tof} is the signal time-of-flight. The parameter Z_n is the normalized impedance defined as $Z_n = R_t/Z_0$, R_t is the termination resistance and Z_0 is the characteristic impedance of transmission-line. The parameter n is the attenuation parameter defined as $n = \exp(-\alpha l)$, α is the attenuation constant of the interconnect and l is the length of the transmission-line.

The maximum estimation error in the eye opening voltage is around 10% compared with circuit simulation results. This estimation method can only be used for the resistive termination structure and the PWL approximation model will constrain it applications very much.

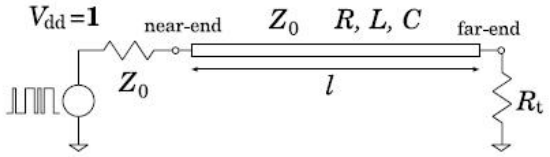


Figure I. 9 Circuit model of a transmission-line with resistive termination.

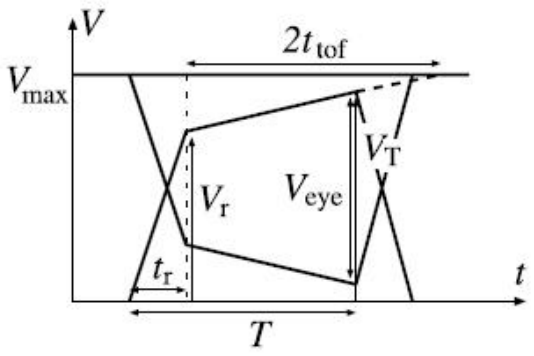


Figure I. 10 PWL waveform model.

1.4.2.2 Analysis Based on Unit Pulse Response

The unit pulse response is the output waveform when the input is unit pulse function. For digital signals with symmetric rise/fall time, the input binary data patterns can be simply modeled as combination of unit pulse function. For any linear time-invariant high speed signaling system, the signal performance can be efficiently analyzed based on the unit pulse response.

In [1], Analui et al. presented a method for predicting data dependent jitter introduced by a general linear time-invariant system based on the system's unit step response. Analytical formulas are derived for first-order system and approximated perturbation technique can be applied for general systems. The prediction has 7.5% error compared with measurement results.

In [7], Casper et al. developed the peak distortion analysis method to extract a worst-case eye diagram considering all interference sources. This method calculates the worst-case voltage or timing margin from the unit pulse response of the system and avoids tedious simulation. The eye edge due to the worst-case '1' is given by

$$s_1(t) = y(t) + \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} y(t-kT)\,|_{y(t-kT)<0} \tag{I.3}$$

and the eye edge due to the worst-case '0' is given by

$$s_0(t) = \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} y(t-kT)\,|_{y(t-kT)>0} \tag{I.4}$$

Therefore, the worst-case eye opening, e(t), is defined as

$$s_1(t) > e(t) > s_0(t) \tag{I.5}$$

However, these methods can analyze digital signals only with symmetric rise/fall time. It's inapplicable to analyze digital signals with asymmetric rise/fall time because the input data patterns cannot be modeled as combination of unit pulse input.

1.4.2.3 Analysis Based on Step Response

The step response of a general system is the time behavior of the outputs when its inputs change from zero to one in a very short time. For a signaling system, its step response implicates certain characteristics of the system. For example, the step response reveals the time-of-flight delay, the reflection and the saturation voltage for a transmission line communication.

In [39], Zhu et al. predicted the eye diagram using the derived analytical formula based on step response. They assume the step response is bitonic which means the step response monotonically increases to its peak voltage and then monotonically decreases to its saturation voltage. A generic bitonic step response is shown in Figure I. 11 and its characterization is shown in Table I. 1.

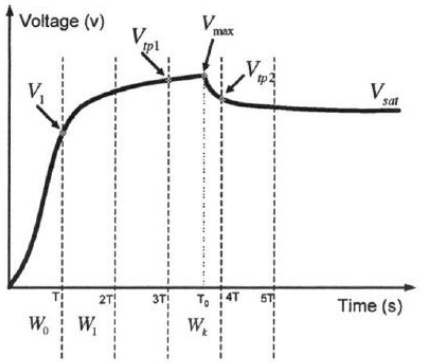


Figure I. 11 A generic bitonic step response.

Table I. 1 Characterization of a generic bitonic response.

$V_1 = V(T)$	Voltage at time point T.
$V_{max} = V(T_0)$	Maximum voltage of the step response.
$V_{sat} = V(\infty)$	Final saturation voltage of the step response.
$V_{tp1} = V(kT)$	$kT \le T_0 \le (k+1)T$
$V_{tp2} = V((k+1)T)$	

The worst-case eye opening is given by

$$V_{eye} = V_{\min}^{top} - V_{\max}^{bottom} = V_{sat} - 2(\max\{V_{tp1}, V_{tp2}\} - V_1) \qquad \text{(I.3)}$$

The worst-case jitter is obtained from the fastest rising edge to the slowest rising edge.

The predicted jitter and eye opening are 10% deviation form HSPICE simulation results. However, the bitonic step response assumption does not hold for general signaling systems.

Chapter 2

Escape Routing

for Square Area Array

With the steady progress of high performance electronic systems, the number of chip/package input/output (I/O) pins has continuously been growing. Increasing demand for high I/O pins count prompts the packaging industry to explore new technologies, such as area array interconnection, CSP (chip scale packing) and BGA (ball grid array). Escape routing connects the I/O pins to the next level assembly. It directly affects the cost and performance of the high-speed electronic systems. An efficient and effective escape routing strategy which achieve high performance with low cost will greatly benefit the electronic product.

The increase in the number of escape routing layers will result in a higher manufacturing cost of substrates used as a package or a board. In order to reduce the number of layers, intuitively some dimensional changes could be introduced, such as decreasing the width of wires, and the space between wires. However in these cases, the yield and reliability become concerned issues and new process technology is necessary as well. Methods to reduce the layer count without making any change in manufacturing process and footprint standard are accordingly desired.

In this chapter, we introduce a novel concept, escape sequence, which plays an important role in escape routing. We formulate and solve a maximum flow problem to analyze the escape bottleneck in area array. Based on the bottleneck analysis, we present the escape sequence concept and summarize several guidelines for escape routing design. Two new escape routing strategies are proposed and the number of escape routing layers can be reduced dramatically. We also design and implement an automatic escape routing program to facilitate investigating different escape routing strategies. The escaping wires could be modeled as frequency dependent RLGC circuit for signal performance analysis.

The area array considered in this chapter is a regular array, which is a fully populated array and is a typical square grid matrix. Totally, there are n^2 I/Os in an area array of size n. We also assume the blind via technology is used, which means that the connected via will disappear in the following routing layers after that I/O is escaped.

One of the main criteria of escape routing is the number of routing layers used for breaking out all I/Os in the area array, which will affect the manufacturing cost directly. Hence the reduction of the escape routing layer count is our emphasized objective.

Escape routing breaks out I/Os in the array to the outside and the objective is to minimize the number of routing layers and to break out I/Os as many as possible. In real applications, only signal IOs inside the array require breakout routing. But for analysis, all IOs are taken into consideration.

2.1 Escape Bottleneck and Escape Sequence Analysis

We explore the escape routing design from a bran-new point of view, escape sequence. We first analyze the escape routing bottleneck in the array, then present our escape sequence concept and summarize the critical guidelines for escape routing design.

2.1.1 Escape Bottleneck

We define a routing graph G = (V, E) to extract the escape routing resources in an array and formulate a maximum flow problem on the routing graph for arrays with different I/O patterns. The maximum flow solution will reveal the bottleneck of escape routing for the array.

2.1.2 Maximum Flow Problem Formulation

A routing graph is defined on the area array. We map the area array into grid with I/Os on the crossing points of the grid, as shown in Figure II. 1. Each grid cell corresponds to the routing space surrounded by four I/Os on its corners and is represented by a vertex in the routing graph. The edge in the routing graph connecting the vertices represents the routing channel between the adjacent grid cells. For a regular I/Os area array, the routing graph looks like a mesh. We append sink vertices, which are located on the boundary of the array, to represent the destinations of escape routing and all sinks will be connected together by a hyper-sink-vertex (Figure II. 2). I/Os in the array are referred as source vertices and similarly there's a hyper-source-vertex connecting all sources (Figure II. 2). Sink vertices will only connect to the vertex which corresponds to the nearest grid cell and source vertices will connect to four vertices which correspond to the four grid cells surrounding the I/O.

The vertices and edges are attached with a capacity, which is the number of available routing wires. The capacity of an edge is defined as the maximum number of wires that can route through two adjacent I/Os. The capacity of a vertex is defined as the maximum number of wires that can route through the diagonal of the grid cell. If some pins do not exist, which means some grid cells do not have exactly four I/Os in their corners, then we will assume that one routing wire locates at that corner. Thus the edges' capacity will constrain the number of

escape routing wires that go through horizontal or vertical channels, while the nodes' capacity will constrain the number of escape routing wires that go through the grid cells. The capacity of the edge between sink vertex and the vertex inside array is defined as the maximum number of wires that can route through the corresponding horizontal or vertical channel. The capacity of the edge between source vertex and the vertex inside array is defined as "1". Moreover the source vertex capacity is equal to "1" and the hyper-source-vertex capacity is equal to the total number of I/O pins in the area array. The capacity of sink vertex and hyper-sink-vertex is assigned infinite because they only represent the destinations and there is no routing resource constraint on them.

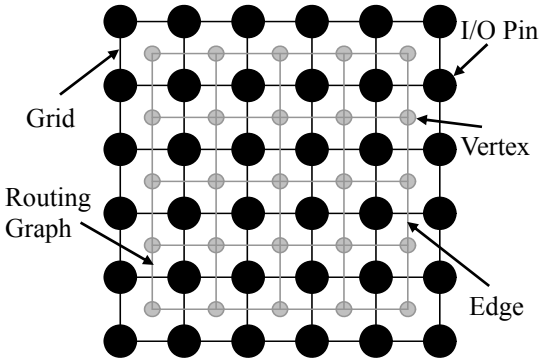


Figure II. 1 Routing graph.

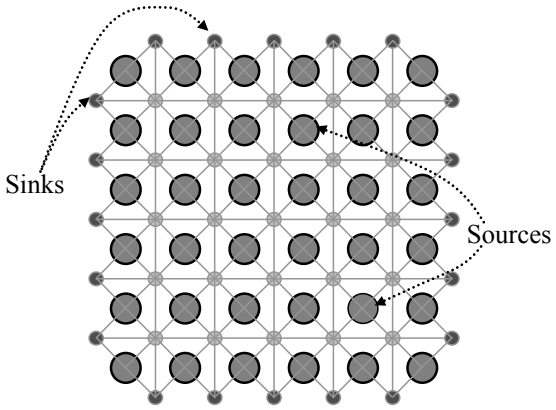


Figure II. 2 Sink vertices and source vertices.

The maximum flow formulation from the hyper-source-vertex to the

hyper-sink-vertex in the routing graph can be summarized as follows and the maximum flow solution provides the upper bound of the number of I/Os that can be escaped in one layer for the area array.

Given:	A routing graph	G = (V, E)
	Edge capacity	C_e: E→R+
	Vertex capacity	C_v: V→R+
	Hyper-source-vertex	s
	Hyper-sink-vertex	t

Find a maximum s-t flow.

2.1.3 Maximum Flow Algorithm for Escape Bottleneck Analysis

We modify the typical maximum flow algorithm, Ford-Fulkerson algorithm [23], to solve our formulated problem. Ford-Fulkerson algorithm is a widely used method for solving maximum flow problem, while it can only handle edge capacity constraints. We expand this algorithm to consider both edge and vertex capacity constraints simultaneously. Figure II. 3 shows our modified Ford-Fulkerson algorithm.

Finding an augmenting path iteratively is an important process in Ford-Fulkerson algorithm. Intuitively, the I/Os inside area array will choose shortest route to the outside because longer wires will occupy more routing resources, will block more wires and will increase the possibility of crosstalk. Therefore we use Dijkstra's shortest path algorithm [23] to find augmenting path in the residual network. The length of every edge is defined as "1". We modify Dijkstra's algorithm similarly to consider the vertex capacity constraints as well as the edge capacity constraints. The augmenting path problem can be summarized as follows and the modified Dijkstra's algorithm for finding augmenting path is shown in Figure II. 4.

Given:	A routing graph	G = (V, E)
	Edge capacity	C_e: E→R+
	Vertex capacity	C_v: V→R+

Flow in the graph	f	f(e): E→R, f(v): V→R+
Edge length		l(e) = 1, ∀ e∈E
Hyper-source-vertex	s	
Hyper-sink-vertex	t	

Find an augmenting path P from s to t.

MaxFlow_EB(G, C, s, t)
Step1: Initialization
Let f be an initial feasible flow,
f(e) = 0 (∀ e∈E) and f(v) = 0 (∀ v∈V)
Step2: Flow Augmentation
Find an augmenting path P from s to t on the residual network.
If there is no augmenting path, go to Step5.
Step3: Flow Update
Check the minimum value, delta, of residual capacity on current augmenting path P.
Update the flow of edges and vertices along path P.
f(e) = f(e) + delta if e is a forward arc on P
f(e) = f(e) – delta if e is a backward arc on P
f(v) = f(v) + delta if the direction of flow on P is identical with previous flow
f(v) = diff(f(v), delta) if the flow on P is opposite to previous flow.
Step4: Go to Step2, repeat flow augmentation.
Step5: The present f is a maximum flow solution.
Step6: Stop.

Figure II. 3 Modified Ford-Fulkerson algorithm

```
AugPath_EB(G,C,f,l,s,t)
Step1: Initialization
        Let set S = {s}, N = {neighbors of u | ∀ u∈S}
        Label vertex s as "Done" and all other vertices as "No_Path"
Step2: Path Finding
        Let set T = ϕ.
        For each vertex n in N
          If some vertex u in S satisfy:
            the capacity of the edge from u to n is not saturated,
            the capacity of u and n is not saturated either
          Then T = T ∪ {n}.
        Label all the vertices in T as "Done".
        Let S = T, N = {neighbors of u, with "No_Path" label | ∀ u∈T }
        If t∈T, go to Step4.
        If T = ϕ, go to Step5.
Step3: Go to Step2.
Step4: An augmenting path found, go to Step6.
Step5: There is no more augmenting path in the graph, go to Step6.
Step6: Stop.
```

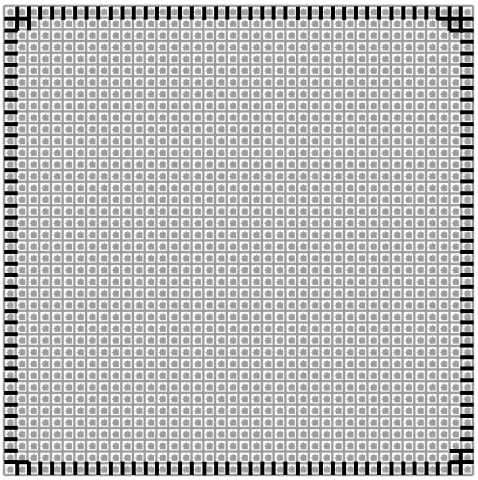
Figure II. 4 Modified Dijkstra's algorithm.

2.1.4 Escape Bottleneck Analysis

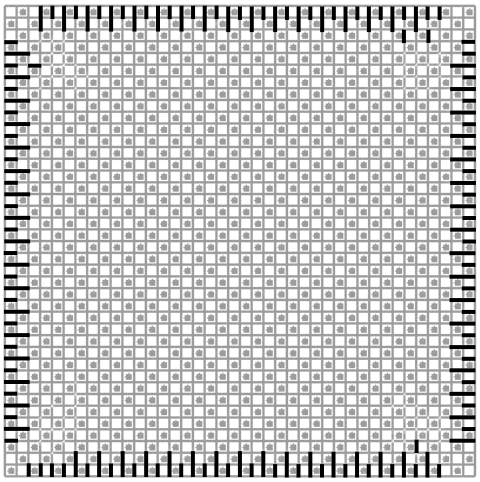
We can identify the escape bottleneck in the area array from the maximum flow solution. Furthermore we can get guidelines for designing efficient escape routing strategies. In the maximum flow solution on the routing graph, we define bottleneck edge as the edge whose flow is equal to its capacity and all bottleneck edges form a bottleneck contour for the area array. This bottleneck contour constrains the maximum number of routing wires which can be routed through in one layer. The maximum flow value is the upper bound of the number of I/Os that can be escaped from the area array in one layer.

We do experiments on area array with different I/Os' patterns and identify the bottleneck edges and bottleneck contour. The size n of the area array in our experiments is 20. The pad pitch, pad diameter, line width, and spacing are 150μm, 75μm, 20μm, and 20μm respectively. Figure II. 6 shows the escape routing bottleneck and maximum flow values for five different I/Os' patterns. The black bold edges in the figures represent the bottleneck edges.

These experimental results reveal that the bottleneck contour of the area array always follows the outline of the array for usual patterns. We can say the number of I/Os that can be broken out in one routing layer will be constrained by the capacity of the array outline.

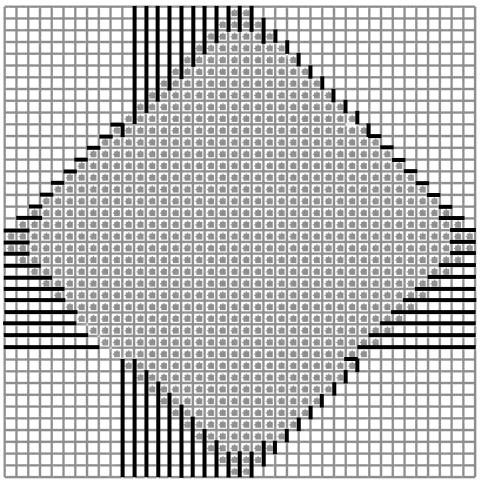


(a) Full populated area array, Maximum flow = 370.5

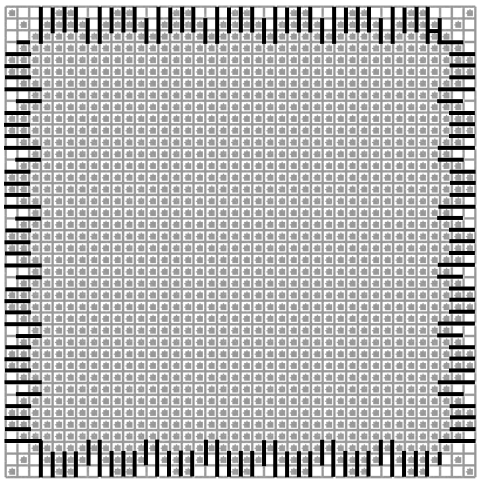


(b) Staggered distribution, Maximum flow = 446.23

Figure II. 5 Escape routing bottleneck.

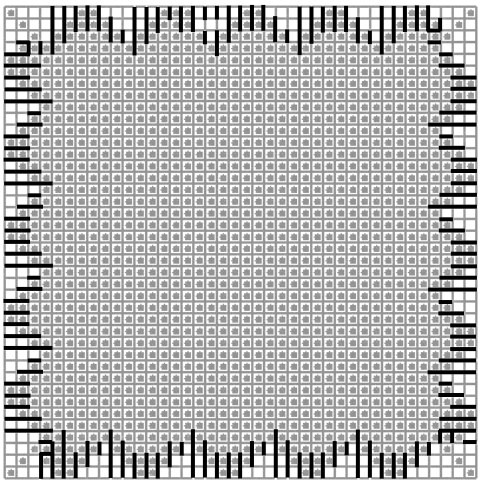


(c) Rhombic distribution, Maximum flow = 271.62



(d) Zigzag outline (size of indention = 5), Maximum flow = 438.93

Figure II. 5 Escape routing bottleneck. (Cont.)



(e) Zigzag outline (size of indention = 7), Maximum flow = 447.26

Figure II. 5 Escape routing bottleneck. (Cont.)

2.2 Escape Sequence and Guidelines for Escape Routing

We introduce a new concept “escape sequence” in escape routing design. It is defined as the overall sequence of I/Os in the area array that are escaped layer by layer. The escape sequence determines the transformation of the array outline throughout the multi-layer escape routing. Based on our escape bottleneck analysis, the number of I/Os that can be broken out in one routing layer is usually constrained by the capacity of the array outline. After we escape I/Os for one routing layer, the pins left in the array will form a new pattern and the outline of the array with this new pattern constrains the number of I/Os that can be escaped in the next routing layer. Hence which I/Os escaped in current layer directly affects the escaping efficiency in the following layers. Furthermore, the pins located at the outmost positions are escaped definitely in every layer and the outline of the array inevitably shrinks to the center layer by layer. Therefore the transformation of the array outline in the escaping process is an important factor for escape routing efficiency and the escape sequence plays a critical role.

We summarize the guidelines for designing efficient escape routing as follows using

the escape sequence concept.

(i) Maintain an appropriate shape of the array outline with large routing capacity layer by layer;

(ii) Control the shrinking rate of the array outline in escaping process;

(iii) Keep similar shape of the array outline for all escape routing layers for scalable consideration.

2.3 Escape Routing Strategies with Different Escape Sequence

We investigate different escape routing strategies from escape sequence point of view and propose two new methods which can reduce the number of escape routing layers dramatically. Those previous works can be classified as two escape sequence strategies. The escape sequence for the traditional method is straightforward sequential from outside to inside and we call it row-by-row sequence strategy. The escape sequence for the preferential escape routing method is to form zigzag array outline and we call it parallel triangular sequence strategy. According to the guidelines discussed in previous section, we propose two new escape strategies called central triangular sequence and two-sided sequence.

2.3.1 Row-by-row Escape Sequence

It is the conventional approach, as shown in Figure II. 6, which breaks out pins row by row (column by column) from outside to inside. If we assume the number of wires that can go through the vertical (horizontal) routing channel, i.e. the channel between adjacent I/Os with minimum pitch, is m.

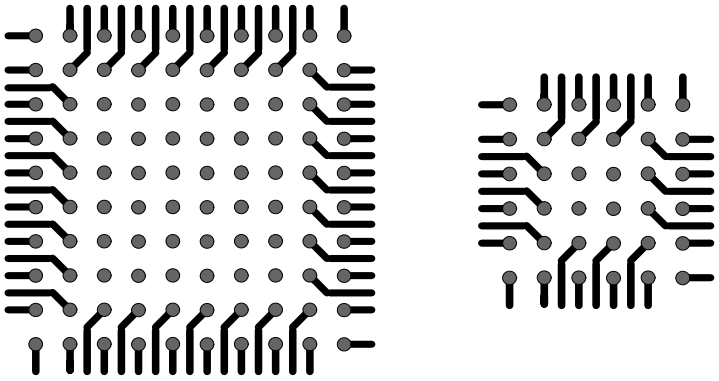
$$m \le \frac{P - D - 2S_p + S_w}{W + S_w} < m + 1 \qquad \text{(II.1)}$$

Thus for the area array, m+1 rows (columns) can be escaped in one routing layer and the total

number of routing layers for breaking out an array with size n is

$$Num(layers)\_rbr = \left\lceil \frac{n}{2(m+1)} \right\rceil \quad \text{(II.2)}$$

This strategy is very straightforward but the array outline shrinks very rapidly layer by layer. Hence usually it needs many layers for escape routing for large size array.



(a) First layer (b) Second layer

Figure II. 6 Row-by-row escape routing.

2.3.2 Parallel Triangular Escape Sequence

This strategy divides I/Os in the array into four parts by its two diagonals and I/Os are escaped to four directions, north, south, east and west respectively. I/Os in each part are further divided into small groups and each group is broken out following a triangular outline, as shown in Figure II. 7 (only two escape directions are shown for clarity). I/Os within the bottom triangle of each group are escaped in the first routing layer. The size of the triangle, k, defined as the number of I/Os on the bottom edge of the triangle, is decided by the vertical (horizontal) routing channel capacity m and we usually choose k as an odd number.

$$\frac{(k+1)^2}{4} \le (k-1)m + k$$

$$\text{i.e.} \quad k \le 4m+1 \quad \text{(II.3)}$$

In the second and following routing layers, each group has a triangular outline which supplies large routing capacity. We assume the number of wires that can go through the diagonal routing channel, i.e. the channel between diagonal adjacent I/Os, is q.

$$q \le \frac{\sqrt{2}P - D - 2S_p + S_w}{W + S_w} < q + 1 \qquad \text{(II.4)}$$

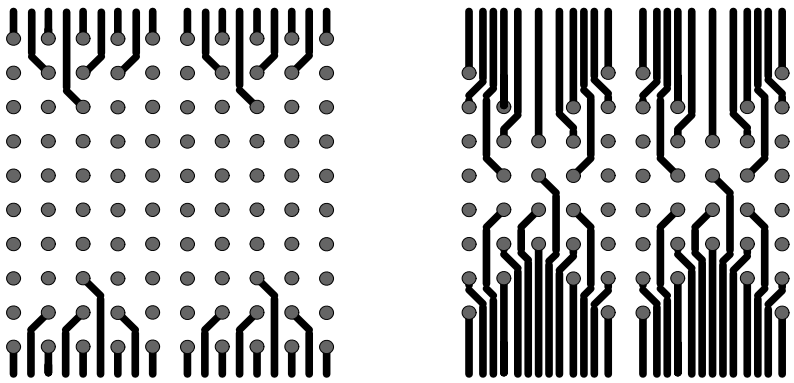
As long as the condition (II.5) is satisfied, q+1 rows (columns) can be escaped in one routing layer following the triangular outline except (q-m) I/Os left. I/Os left in every layer are scattering in the array and can be escaped in the spare space.

$$q \le \frac{p - D/2 - W/2 - S_p}{W + S_w} \qquad \text{(II.5)}$$

The total number of routing layers for breaking out an array with size n is

$$Num(layers)\_pt = 1 + \left\lceil \frac{n - k - 1}{2(q+1)} \right\rceil \qquad \text{(II.6)}$$

Compared with the traditional method, this escape sequence allows more pins broken out for a certain layer because of the zigzag array outline. However the array outline still shrinks layer by layer and the escape routing efficiency is greatly reduced for the last layers.



(a) First layer (b) Second layer

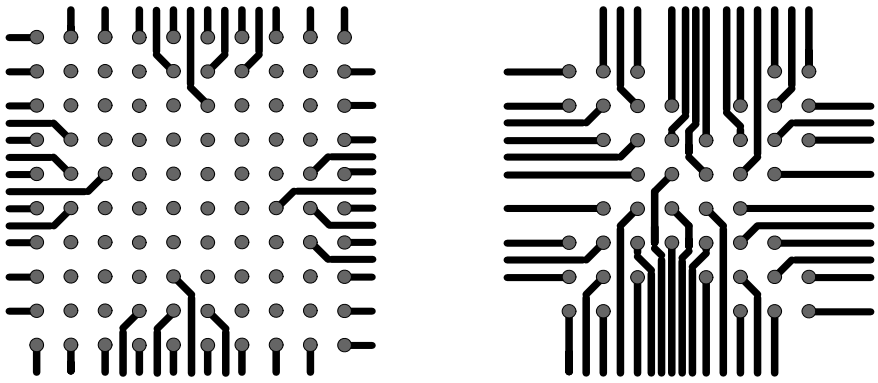
Figure II. 7 Parallel triangular escape routing.

2.3.3 Central Triangular Escape Sequence

This strategy also divides I/Os in the array into four parts by two diagonals and I/Os are escaped to four directions respectively. Different from the parallel triangular escape sequence, this strategy breaks out I/Os in each part from the center of the outside row and a single triangular indented outline is expanded layer by layer, as shown in Figure II. 8. I/Os within the bottom center triangle are escaped in the first routing layer. The size of the triangle, k, is defined in (II.3). In the second and following routing layers, one triangular outline is formed in each part and becomes larger layer by layer. We assume the routing capacity of the vertical (horizontal) channel is m and the routing capacity of the diagonal channel is q. As long as the condition (II.5) is satisfied, q+1 rows (columns) can be escaped in one routing layer following the triangular outline. Thus, the total number of routing layers for breaking out an array with size n is

$$Num(layers)\_ct = 1 + \left\lceil \frac{n-k-1}{2(q+1)} \right\rceil \qquad \text{(II.7)}$$

Using this method, we achieve continuously increasing array outline, i.e. the capacity of the array outline is increased continuously layer by layer. However the capacity for the first several layers is smaller than previous two strategies.



(a) First layer (b) Second layer

Figure II. 8 Central triangular escape routing.

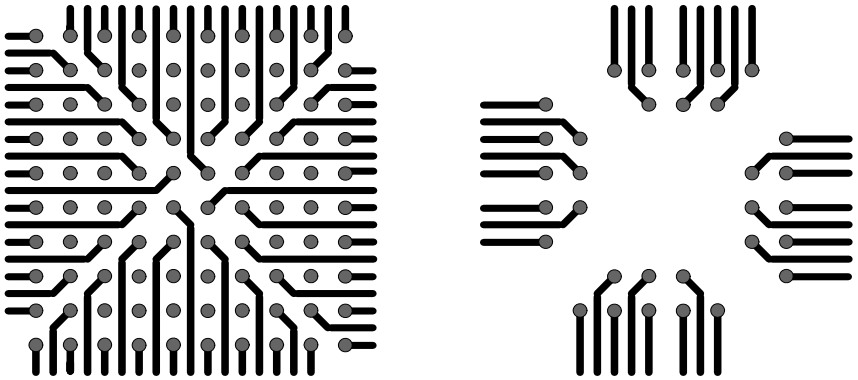
2.3.4 Two-sided Escape Sequence

Same as previous two strategies, this approach divides I/Os in the array into four parts by two diagonals and I/Os are escaped to four directions respectively. It breaks out I/Os from the inside as well as from the outside, as shown in Figure II. 9. Both the outside and inside outline maintains the zigzag shape and the array outline shrinks very slowly.

This two-sided escape method applies the zigzag outline used in the parallel triangular escape method and overcomes its shortcoming to improve the escape routing efficiency. In the parallel triangular escape method, I/Os are divided into small groups in every part of the array and are escaped to form a triangular outline in each group. But the heights of those groups in each part, i.e. the number of rows, are different. The group in the middle has larger height which constrains the number of necessary routing layers, however the group in the corner has smaller height which can be broken out easily. Therefore, we can escape I/Os inside the array through the routing channels in the corner to decrease the largest height and fully utilize the routing channels in the corner. We slow down the shrinking rate of the array outline in this way. Then the total number of routing layers for breaking out an array with size n is

$$Num(layers)\_ts \le 1 + \left\lceil \frac{n-k-1}{2(q+1)} \right\rceil \qquad \text{(II.8)}$$

The number of I/Os broken out in each routing layer is very even. So this strategy can efficiently decrease the number of layers for escape routing and it has more advantage for large size array.



(a) First layer (b) Second layer

Figure II. 9 Two-sided escape routing.

2.3.5 Four Escape Sequence Strategies Comparison

According to our analysis and guidelines for escape routing proposed in section 2.3, the advantages and disadvantages of those four escape sequence strategies are obvious. Figure II. 10 shows the transformation of the array outline in the escaping process for the four strategies we have discussed. Because the array outline of the traditional row-by-row method shrinks quickly to the center layer by layer, the escape routing for the latter layers becomes very inefficiently. The parallel triangular approach improves the escape efficiency by maintaining zigzag array outline. The escape routing of central triangular approach becomes more and more efficient layer by layer while the number of I/Os escaped in the first several layers is too small. The two-sided escape sequence controls the array outline perfectly.

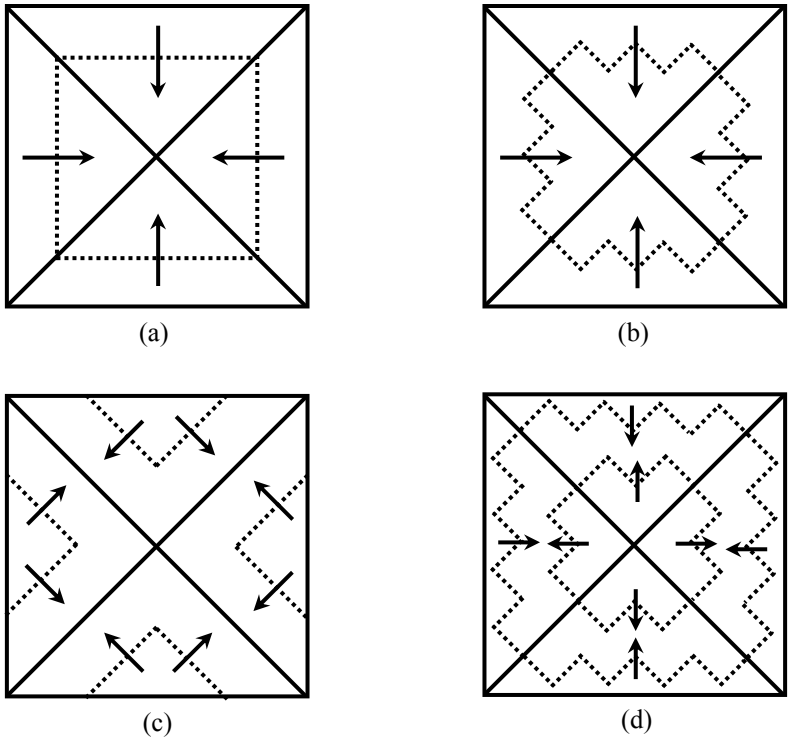


Figure II. 10 The transformation of array outline.

2.4 Automatic Escape Routing Algorithm

We need an automatic escape routing program to facilitate investigating different escape sequence strategies. We design and implement an escape routing algorithm because the escape routing is usually done manually in industry. This automatic escape routing program helps us to compare different escape strategies for real experiments and verify our analysis. The practical escape routing results will also benefit the signal integrity analysis and further researches.

2.4.1 The Characteristics of Escape Routing

The escape routing has several characteristics different from the routing in ICs

(Integrated Chip) and the routing on board.

(i) The escape routing wires connect I/Os in the array with the destinations on the four boundaries surrounding the array and each wire has only two terminals.

(ii) The wire breaking out one I/O is routed in only one layer.

(ii) The escape routing wires are composed of horizontal, vertical and diagonal segments with 0°, 90°, 45° and 135° respectively.

(iv) I/Os that haven't been escaped are obstacles for current routing wires.

I/Os are routed layer by layer according to the given escape sequence and for each layer the routing is some kind of area routing. To design the automatic escape routing algorithm, we utilize the idea of river routing algorithm [22] and devise special operations aiming at those characteristics of escape routing.

2.4.2 Automatic Escape Routing Algorithm

The main differences between escape routing and river routing are the location of net terminals, internal blockages and the direction of routing segments. The two terminals of each escape routing wire are located at some point in the array and some point on the array boundary respectively, while the two terminals of signal net in river routing are both located on the boundary of routing region. In escape routing problem, I/Os that haven't been escaped are obstacles, while the routing region for river routing problem cannot have internal blockage. The routing segments for escape routing can be horizontal, vertical and diagonal, while river routing can only support horizontal and vertical routing segments. Considering the characteristics of escape routing, we modify the river routing operations and devise our automatic escape routing algorithm. Figure II. 11 shows the main flow of the algorithm.

The algorithm proceeds layer by layer for a given escape sequence. We construct routing graph and calculate the maximum flow value using the same algorithm as escape

bottleneck analysis described in section 3 for every layer. If the input sequence is reasonable compared with the maximum flow value, I/Os are routed for current layer. For each routing layer, the algorithm proceeds I/O by I/O anticlockwise starting from left-bottom corner. The set of routing segments for each I/O is generated in three steps similar as river routing. First a continuous list of segments (S1) with current wire width by licking along the edge of previous wire just routed with the minimum wire spacing is generated. Then, current I/O is connected to S1 with shortest path using horizontal, vertical and diagonal segments; the crossing point of S1 and the array boundary is the other terminal for the wire; segments list (S2) including two terminals is generated in this way. Finally, if S2 crosses some obstacle, the wire will surround the obstacle clockwise at the crossing points. In order to exploit the diagonal routing segments, we use octagon to approximate the shape of I/Os. As output, all the routing wires, the number of routing layers and the number of I/Os escaped on each layer are reported.

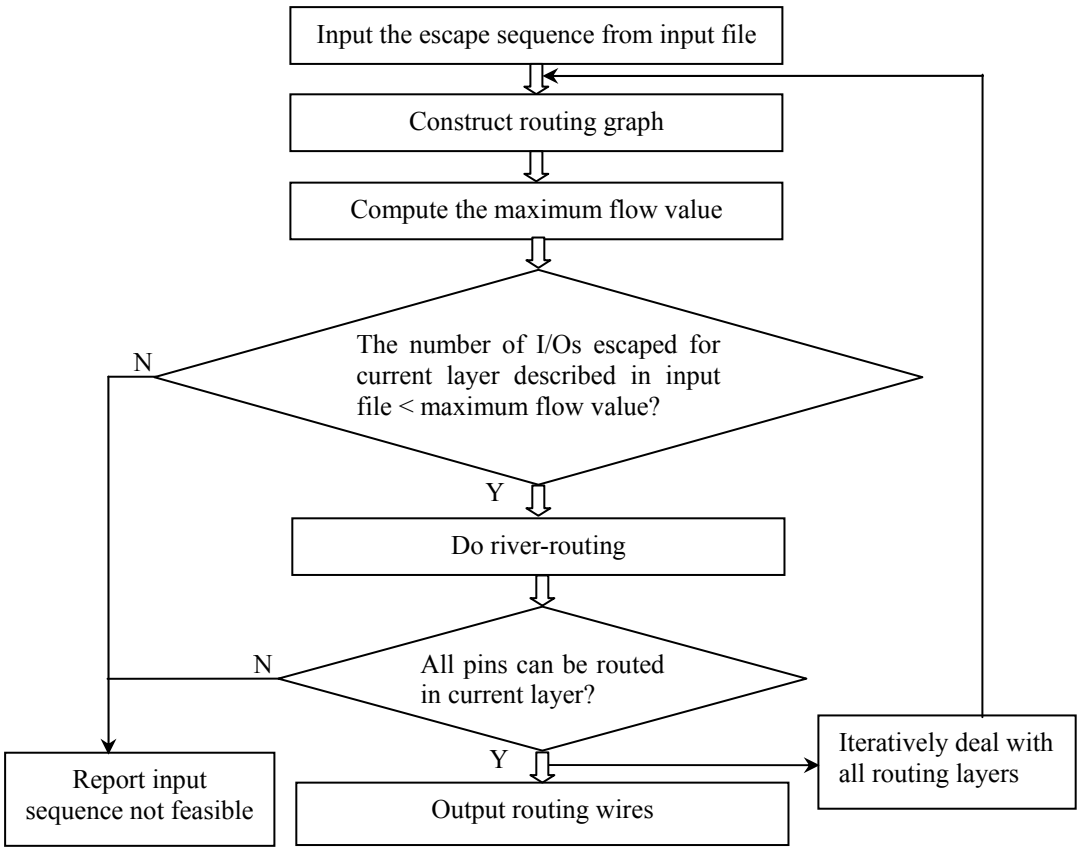


Figure II. 11 Escape routing algorithm.

We implement this automatic escape routing algorithm and use it to do routing for real experiments using those four escape sequence strategies described in section 2.4.

2.5 Experimental Results

We break out a regular I/Os area array with size n=20 and n=40 using four different escape sequence strategies discussed in section 2.4. The pad pitch, pad diameter, line width, line spacing and spacing between pad and wire are 150μm, 75μm, 20μm, 20μm and 20μm respectively. Table II. 1 and Table II. 2 list the number of I/Os escaped on every layer for those four escape strategies for 40x40 and 20x20 array respectively. Figure II. 12 shows 20x20 area array escape routing results of the first two layers for four escape strategies.

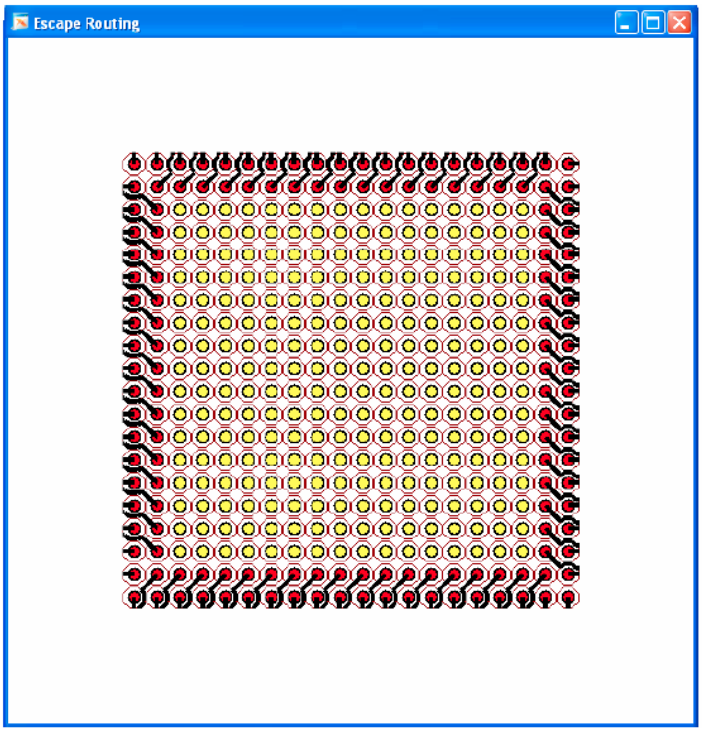
The experimental results demonstrate our analysis and show that the two new escape routing strategies we proposed are very efficient, especially the two-sided escape sequence approach reduces the number of escape routing layers dramatically. We can clearly observe the advantages and disadvantages of those strategies from the results for 40x40 area array. For the traditional row-by-row method, the number of breakout I/Os decreases rapidly layer by layer. It requires 10 total layers. For the parallel triangular method, the number of breakout pins on one-layer peaks at the second layer and drops slowly afterwards. This method reduces the required number of layers from 10 to 7. For the central triangular method, the number of breakout I/Os starts from small value but keeps on increasing and thus reduces the number of layers to 6. The two-sided method is the most efficient one, which achieves the result in only 5 routing layers.

Table II. 1 The number of breakout I/Os escaped on each layer for four strategies. 20x20 area array

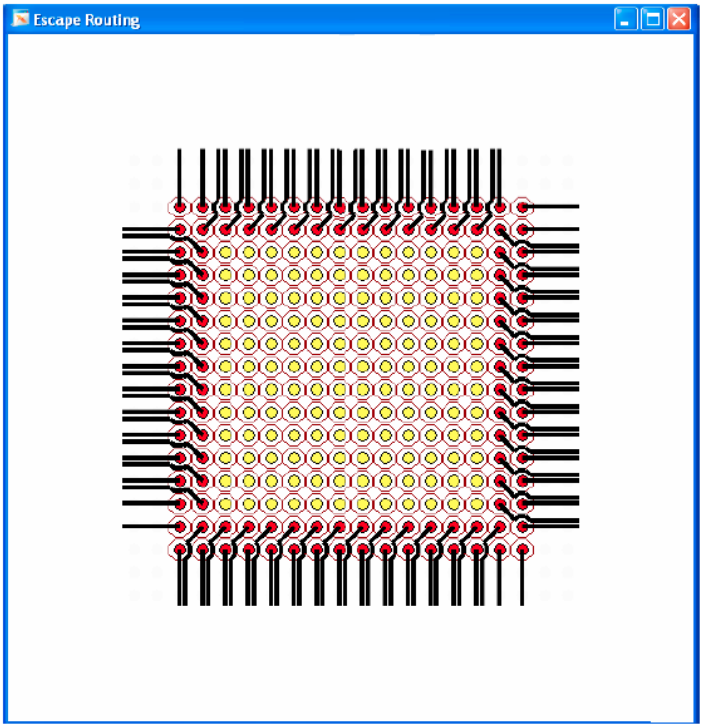
Layer	Row by row	Parallel triangular	Central triangular	Two sided
1	144	132	92	140
2	112	144	116	160
3	80	96	140	100
4	48	28	52	-
5	16	-	-	-

Table II. 2 The number of breakout I/Os escaped on each layer for four strategies. 40x40 area array

Layer	Row by row	Parallel triangular	Central triangular	Two sided
1	304	276	172	312
2	272	340	196	328
3	240	292	220	308
4	208	240	244	324
5	176	164	268	328
6	144	124	292	-
7	112	164	208	-
8	80	-	-	-
9	48	-	-	-
10	16	-	-	-

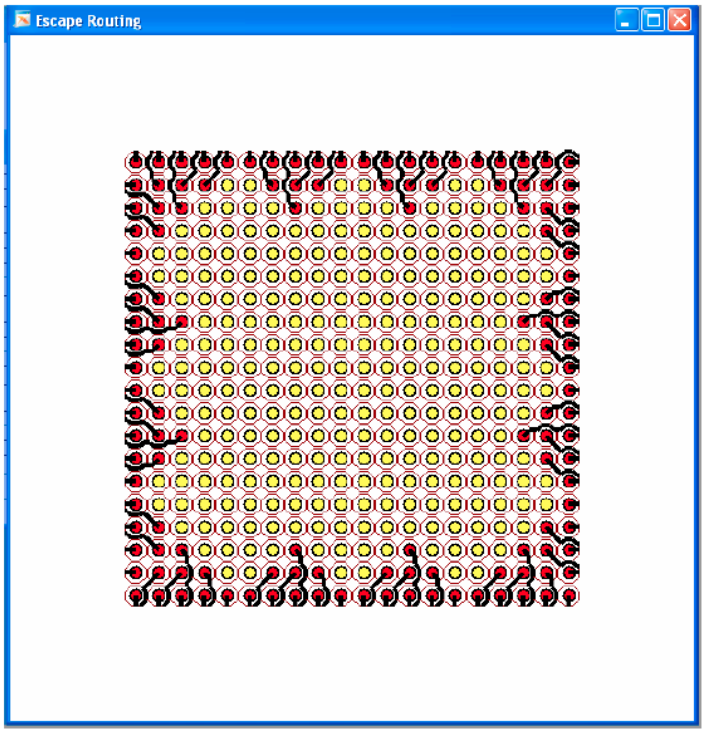


(a) row-by-row sequence, first layer

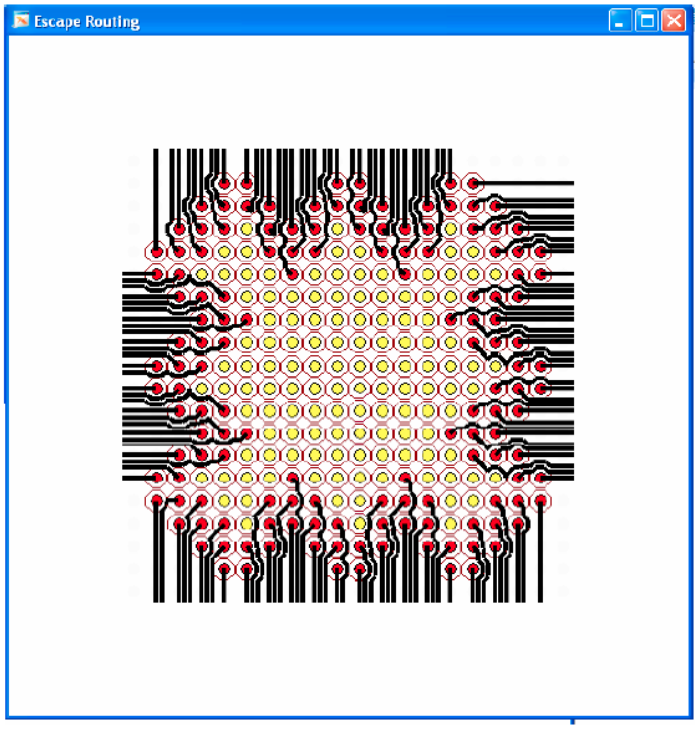


(b) row-by-row sequence, second layer

Figure II. 12 Escape routing results.

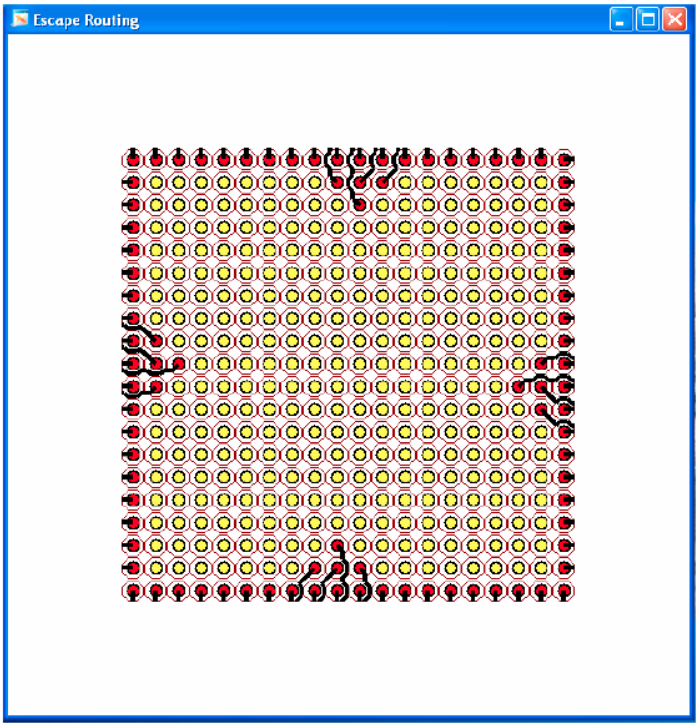


(c) parallel triangular sequence, first layer

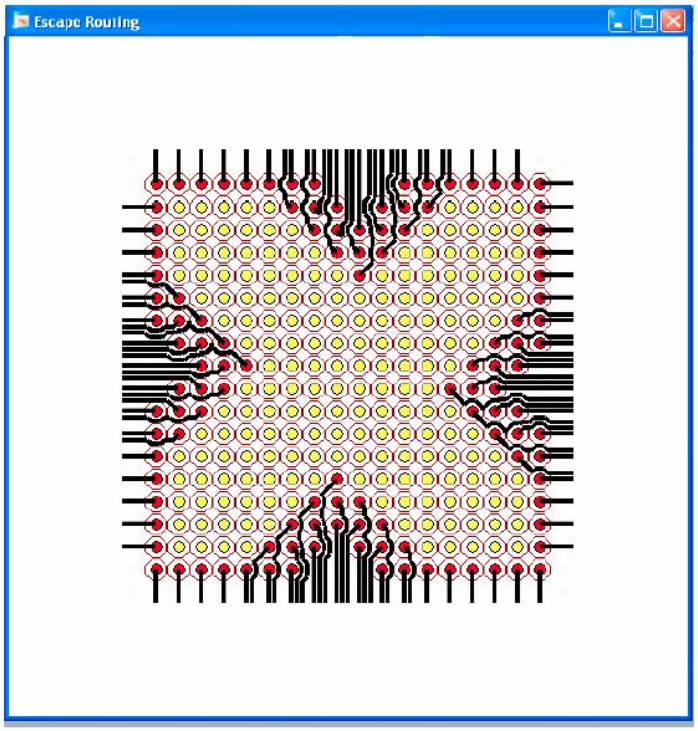


(d) parallel triangular sequence, second layer

Figure II.12 Escape routing results. (Cont.)

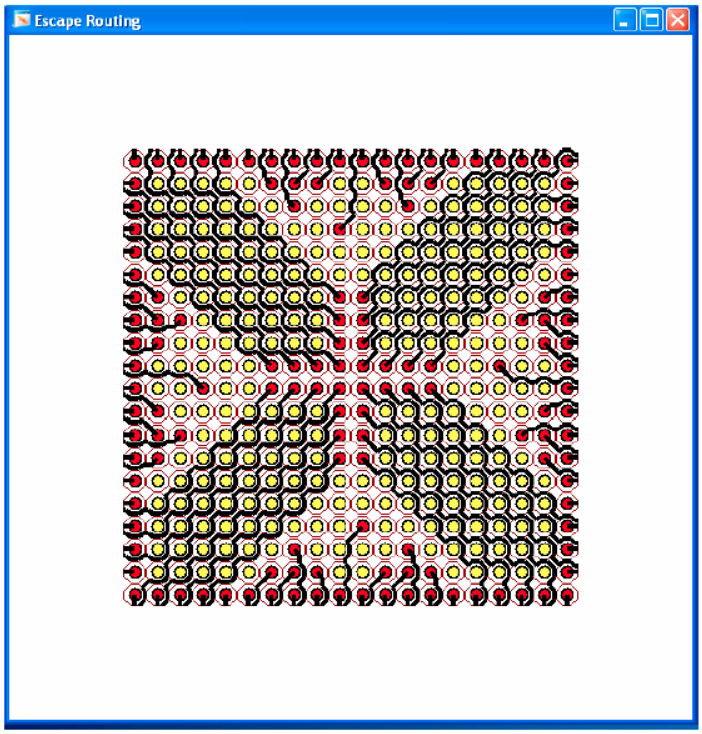


(e) central triangular sequence, first layer

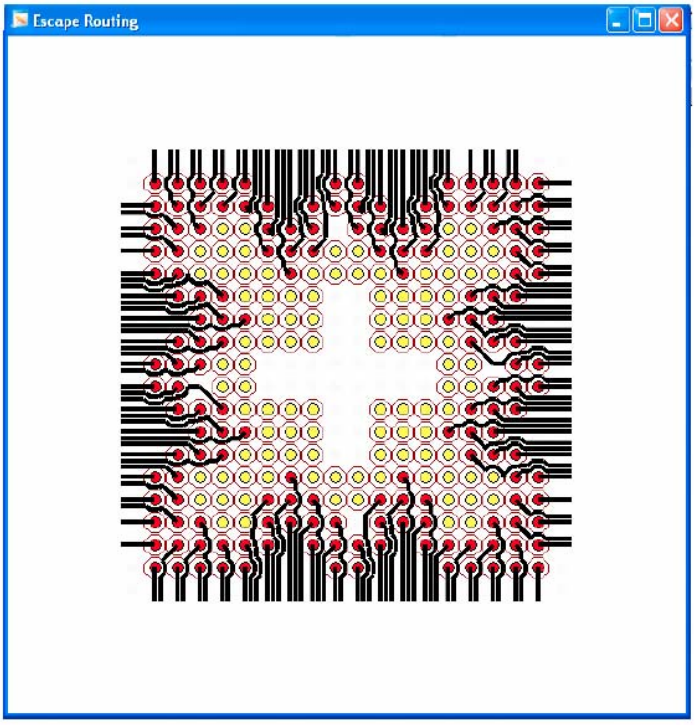


(f) central triangular sequence, second layer

Figure II.12 Escape routing results. (Cont.)



(g) two-sided sequence, first layer



(h) two-sided sequence, second layer

Figure II.12 Escape routing results. (Cont.)

2.6 Analysis Model

The escape routing connects I/Os to the next level assembly. The signal performance is another important consideration in escape routing design. The signal integrity is a measure of the quality of the electrical signal. In high-speed electronic systems, various effects could degrade the electrical signal performance to the point where errors occur. In escape routing, many signal integrity issues should be considered, noise induced by high density wires in escape routing area, crosstalk between wires and vias, the signal skew consideration and the impedance mismatch problem. The escape routing strategy with low cost and high performance is the ultimate goal for the practical electronic systems.

Appropriate modeling for the escape routing is expected in order to analyze the signal quality. We could utilize 3D electromagnetic simulation tool to extract the characteristics of the escaping wires. Further circuit simulation could be applied to analyze the escape routing performance based on the extracted model.

2.6.1 Extraction Tool Raphael

We use Raphael Interconnect Analysis Program from Synopsys to do the extraction. Raphael is a simulator for parasitic modeling in the semiconductor industry provided by Synopsys. It simulates the electrical and thermal analysis of 2D and 3D structures, such as on-chip interconnect structures, packaging, multi-chip modules, and printed circuit boards. Figure II. 13 and Figure II. 14 show two examples which could be calculated by the tool. Analysis includes the following electrical areas: resistance, capacitance, inductance, current flow densities, thermal and potential distribution.

We use RC3 (3D resistance, capacitance, and thermal resistance) and RI3 (3D resistance and inductance with skin effect) programs to extract the RLGC data for the

escaping wires.

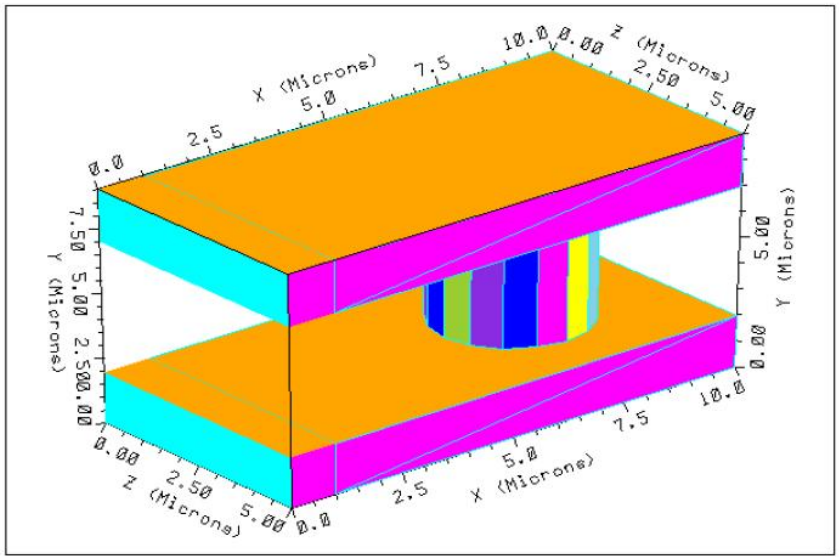


Figure II. 13 Extraction example, via structure

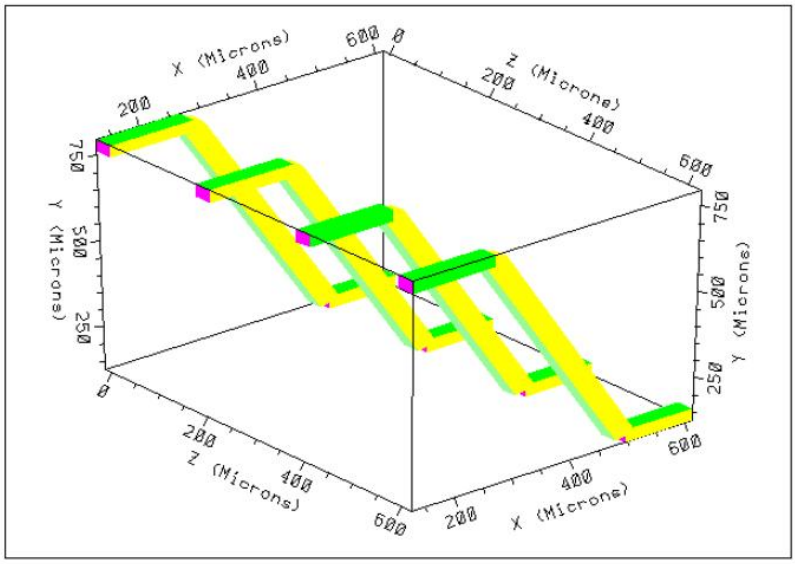


Figure II. 14 Extraction example, bond leads structure

2.6.2 Extraction Demonstration

In escape routing area, several complicated 3D structures involve, the escaping wires, the vias and the pads. Figure II. 15 shows a basic escaping wire structure which consists of

orthogonal horizontal and vertical wires and 45 degree wires. Figure II. 16 shows the model we extract, where RLGC are frequency dependent. This model could be applied into circuit simulation to analyze the signal performance.

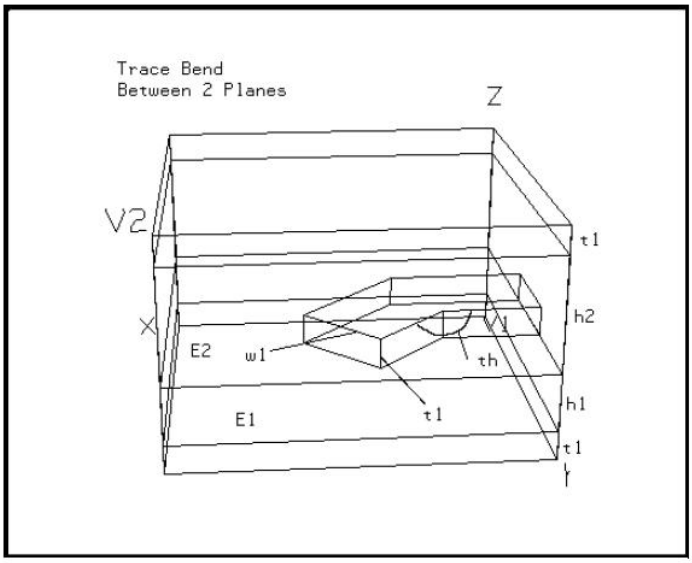


Figure II. 15 Escaping wire structure

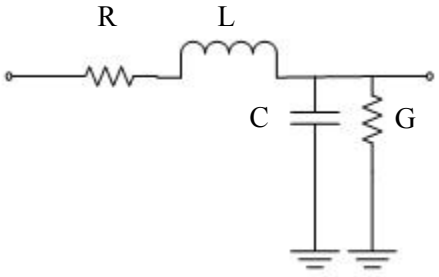


Figure II. 16 Escaping wire modeling

Chapter 2, in part, is a reprint of the paper “Layer Count Reduction for Area Array Escape Routing” co-authored with Hongyu Chen, Chung-Kuan Cheng, Dan Beckman and Dawei Huang in the proceedings of 2005 International Conference and Exhibition on Device Packaging. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Escape Routing

for Hexagonal Area Array

In this chapter, we analyze the properties of hexagonal area array. Traditionally, the area array is a square grid which we have discussed in chapter 2. The hexagonal array increases the density of I/Os in the array remarkably. We propose three escape routing strategies for the hexagonal array: column-by-column horizontal escape routing, two-sided horizontal/vertical escape routing, and multi-direction hybrid channel escape routing. The examples using practical parameters show that our strategies can escape the hexagonal array very efficiently. We can reduce the number of escape routing layers as well as increase the density of I/Os.

3.1 Hexagonal Area Array

The typical I/Os array is a square grid matrix and I/Os are located at the crossing points of the horizontal and vertical mesh. The neighboring four I/Os form a square grid unit. For an n×n square grid array, there are n I/Os in each row and column, n^2 I/Os in the array totally and the area of the array is $(nP+D)^2$. I/Os in the typical square grid array can be

shifted and packed further to form a hexagonal pattern. The neighboring six I/Os form a hexagonal unit and one more I/O locates at the center. In a hexagonal array, the angle between the lines joining any adjacent two I/Os is always a multiple of 60°. Figure III. 1 and Figure III. 2 show the square grid array and the hexagonal array respectively. We observe that the square grid array is symmetric in horizontal and vertical directions, i.e. if the square grid array is rotated 90° clockwise; it is superposed on the original array. However the hexagonal array doesn't have this properties, it is symmetric in 0°, 60°, and 120° directions, i.e. if the hexagonal array is rotated 60° or 120° clockwise; it is superposed on the original array.

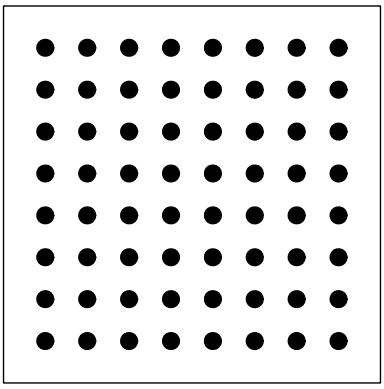


Figure III. 1 Square grid array.

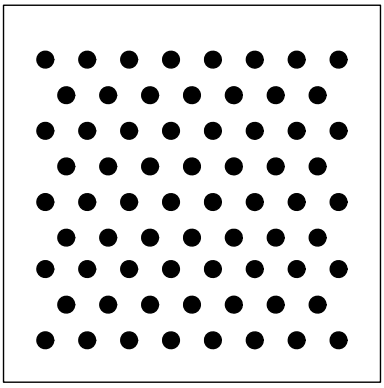


Figure III. 2 Hexagonal area array.

3.1.1 Feature 1: Increasing the Number of I/Os

The hexagonal array can hold more I/Os under the same area usage and same I/O pitch constraints, compared with the square grid array. The area of an n×n square grid array is $(nP+D)^2$. Using the same area and same I/O pitch P, the hexagonal array will have n and n-1 I/Os in each row alternately. The number of rows in the hexagonal array is

$$m = \left\lfloor \frac{2}{\sqrt{3}}(n-1) \right\rfloor + 1 \qquad \text{(III.1)}$$

The number of I/Os in the hexagonal array is

$$TotalNum(hexagon) = \begin{cases} mn - \dfrac{m}{2} & m = even \\ mn - \dfrac{m-1}{2} & m = odd \end{cases} \qquad \text{(III.2)}$$

Plug the formula (III.1) into (III.2), we can prove that the number of I/Os in the hexagonal array is larger than the number of I/Os in the square grid array using same area and same I/O pitch for any n>=8. The increased percentage of the number of I/Os can be approximated as

$$\begin{aligned} &\frac{TotalNum(hexagon) - TotalNum(square)}{TotalNum(square)} \\ &\geq \frac{mn - m/2 - n^2}{n^2} \\ &> \frac{\frac{2}{\sqrt{3}}n^2 - n^2 - \sqrt{3}n - \frac{1}{\sqrt{3}}}{n^2} \\ &= \frac{2-\sqrt{3}}{\sqrt{3}} - \frac{\sqrt{3}}{n} - \frac{1}{\sqrt{3}n^2} \\ &> 15.47\% - \frac{\sqrt{3}}{n} - \frac{1}{\sqrt{3}n^2} \end{aligned}$$

For array with large size, the hexagonal array has more advantage. Table III. 1 shows their comparison for different size of arrays.

Table III. 1 Square grid array vs. Hexagonal array.

Using same area, same pitch.

Size n	Square grid array		Hexagonal array		Increase
	# rows	# I/Os	# rows	# I/Os	
10	10	100	11	105	5%
15	15	225	17	247	9.78%
20	20	400	22	429	7.25%
25	25	625	28	686	9.76%
30	30	900	34	1003	11.44%
35	35	1225	40	1380	12.65%
40	40	1600	46	1817	13.56%

3.1.2 Feature 2: Increasing the Spacing between I/Os

The hexagonal array can increase the average I/Os pitch under the similar number of I/Os and same area constraints, compared with square grid array. In an n×n square grid array, the pitch of the adjacent horizontal or vertical I/Os is the minimum pitch P. Holding the same number of I/Os in same area, i.e. n^2 I/Os in area $(nP+D)^2$, the corresponding hexagonal array can separate the I/Os loosely.

We assume the hexagonal array has k and k-1 I/Os in each row alternately, and then according to (III.1) the number of rows in this hexagonal array is

$$l = \left\lfloor \frac{2}{\sqrt{3}}(k-1) \right\rfloor + 1 \tag{III.3}$$

The number of I/Os in the corresponding hexagonal array is

$$TotalNum(hexagon) = \begin{cases} lk - \frac{l}{2} & l = even \\ lk - \frac{l-1}{2} & l = odd \end{cases} \tag{III.4}$$

Therefore the I/Os pitch P′ in the hexagonal array can be solved from the following inequations

$$\begin{cases} TotalNum(hexagon) \geq n^2 \\ kP' \leq nP \end{cases} \quad \text{(III.5)}$$

Plug (III.3) and (III.4) into (III.5), we can prove that the pitch P′ in the hexagonal array is larger than the minimum pitch P for large n. Table III. 2 shows their comparison for different size of arrays.

Table III. 2 Square grid array vs. Hexagonal array.

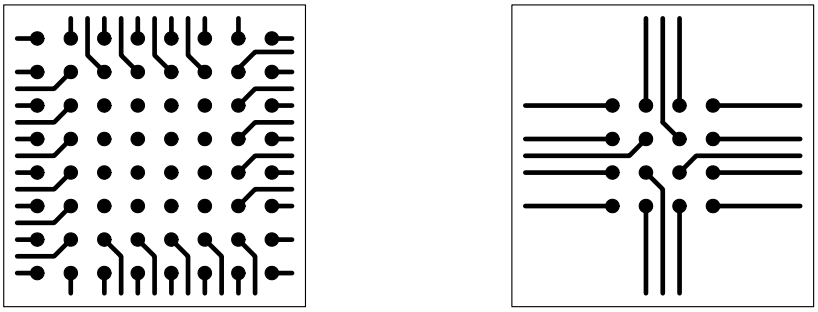
Using same area, holding similar number of I/Os

Square grid array			Hexagonal array			Increase
n	# I/Os	Pitch	k	# I/Os	Pitch	
25	625	P	24	635	1.04P	4.17%
30	900	P	29	941	1.03P	3.45%
35	1225	P	34	1307	1.03P	2.94%
40	1600	P	38	1613	1.05P	5.26%
45	2025	P	43	2083	1.05P	4.65%
50	2500	P	47	2511	1.06P	6.38%

3.2 Escape Routing Strategies for Hexagonal Area Array

3.2.1 Column-by-Column Horizontal Escape Routing

The traditional escape routing strategy for the square grid array is to break out the I/Os row-by-row/column-by-column from outside to inside as shown in Figure III. 3. The spacing between two consecutive I/Os constrains the number of wires going through and limits the number of I/Os escaped for one layer.



(a) First layer (b) Second layer

Figure III. 3 Traditional escape routing for square grid array.

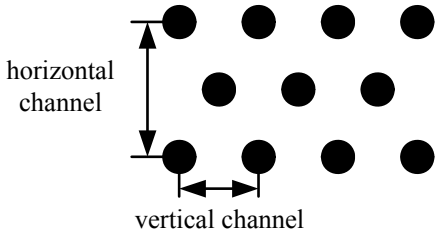


Figure III. 4 Vertical/horizontal channels.

The hexagonal array is not symmetric in horizontal and vertical directions. In the compact hexagonal array, in which the pitch of adjacent I/Os is the minimum pitch, the vertical routing channel, as shown in Figure III. 4, is the edge of the hexagonal unit and the number of wires that can be escaped through is

$$Num(wires)\_verticalChannel = \left\lfloor \frac{P - D - S}{S + W} \right\rfloor \quad \text{(III.6)}$$

However the horizontal routing channel, as shown in Figure III.5, is a diagonal of the hexagonal unit and has larger capacity:

$$Num(wires)\_horizontalChannel = \left\lfloor \frac{\sqrt{3}P - D - S}{S + W} \right\rfloor \quad \text{(III.7)}$$

Therefore the horizontal escape routing can be more efficient than the vertical escape routing. Similarly as the traditional escape routing method for the square grid array, the hexagonal

array can be treated as zigzag column array and I/Os can be escaped column-by-column through the horizontal routing channels as shown in Figure III. 5 and Figure III. 6.

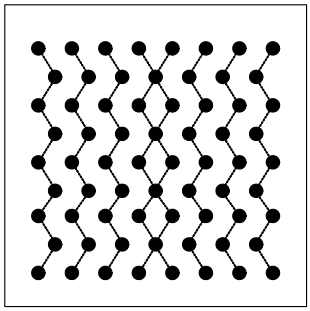


Figure III. 5 Zigzag columns in hexagonal array.

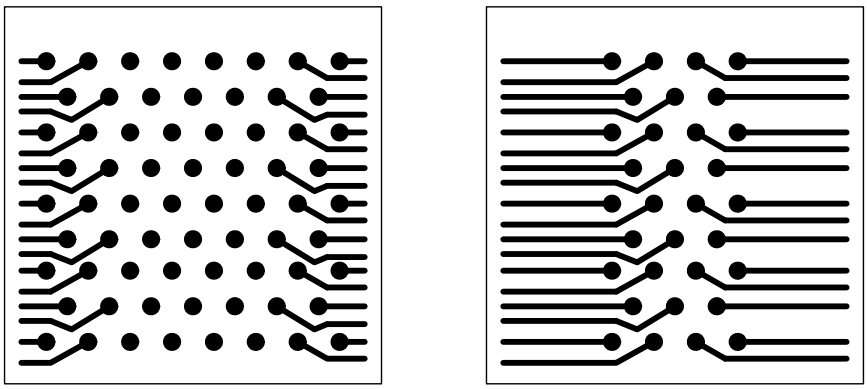


Figure III. 6 Column-by-column horizontal escape routing.

For hexagonal array.

Using column-by-column horizontal escape routing method, under some conditions, we can accomplish the escape routing for the hexagonal array with the same number of layers as the square grid array although the hexagonal array holds more I/Os.

We assume the number of wires that can go through the vertical routing channel, i.e. the channel between adjacent I/Os with minimum pitch, is A. Thus for the square grid array, A+1 rows and columns can be escaped in one routing layer and the number of routing layers for breaking out an n×n square grid array using the tradition method is

$$Num(layers)\_square = \left\lceil \frac{n}{2(A+1)} \right\rceil \quad \text{(III.8)}$$

The hexagonal array with same area and same pitch also has n zigzag rows. If we assume to use the same number of routing layers as the square grid array, the number of wires that can go through one horizontal channel in hexagonal array should satisfy

$$Num(wires)\_horizontalChannel \geq 2A+1 \quad \text{(III.9)}$$

Combine (III.6) (III.7) with the assumptions, we can derive the condition is

$$\begin{cases} Num(wires)\_verticalChannel = \left\lfloor \frac{P-D-S}{S+W} \right\rfloor = A \\ Num(wires)\_horizontalChannel = \left\lfloor \frac{\sqrt{3}P-D-S}{S+W} \right\rfloor \geq 2A+1 \end{cases}$$

$$\Rightarrow D \geq (2-\sqrt{3})P+W \quad \text{(III.10)}$$

As long as the array's parameters satisfy the condition (III.10), I/Os in the hexagonal array can be escaped within the same number of routing layers as square grid array and the hexagonal array has the same area and same pitch as the square grid array but holds more I/Os. Table III. 3 shows the practical parameters for flip chip interconnect and fine pitch BGA/CSP, which is provided by ITRS (International Technology Roadmap for Semiconductors) [42]. The condition is usually satisfied.

Table III. 3 Practical parameters for condition, unit: μm**.**

	Flip Chip		FBGA/CSP	
Year	2006	2018	2006	2018
Pitch	130	70	300	100
Pad Diameter	65	35	120	40
Line Width	27.8	15	36	12
Line Spacing	27.9	15	36	12
Condition (III.10)	✓	✓	✓	✓

We take an 8×8 square grid array as a simple example, as shown in Figure III.2. The pitch of every two adjacent horizontal/vertical I/Os is the minimum pitch. This square grid array has 64 I/Os totally. The hexagonal array, which uses same area, can have 68 I/Os as

shown in Figure III.3 and the pitch of any two adjacent I/Os is also the minimum pitch.

The following values for feature sizes are used in this example and they satisfy the condition (III.10).

1) Minimum pitch (Pmin) = 240μm

2) Diameter of I/O (D) = 110μm

3) Wire width (W) = 43μm

4) Wire spacing (S) = 43μm

The vertical routing channel can route 1 wire and the horizontal routing channel can route 3 wires. Therefore we can use two layers to break out I/Os in that square grid array as shown in Figure III.4. We can also use two layers to escape I/Os in the hexagonal array as shown in Figure III.7. The number of routing layers for the square grid array and the hexagonal array are identical although the hexagonal array has more I/Os.

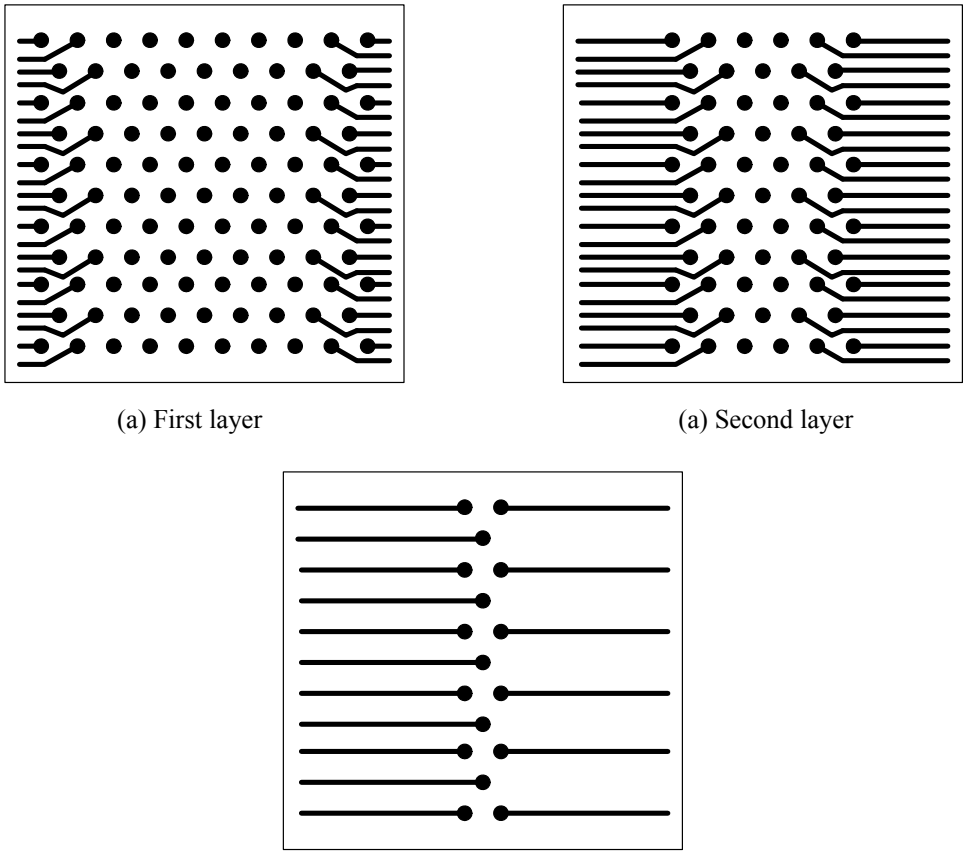
3.2.2 Two-sided Horizontal/Vertical Escape Routing

We have proposed the two-sided escape routing approach for square grid array in [6]. This approach breaks out I/Os in the outside and inside rows/columns simultaneously. It can maintain the outline of the array in a good shape and I/Os are escaped on different routing layers equably. This idea can be exploited in escape routing for hexagonal array to reduce the number of routing layers further.

In the column-by-column horizontal escape routing strategy for the hexagonal array, the number of zigzag columns in the array constrains the number of routing layers. Using the two-sided idea, I/Os in the middle zigzag columns can be escaped through vertical routing channels at the same time as I/Os in the outside zigzag columns are escaped through horizontal routing channels.

We take a 10×11 hexagonal array as an example, which has the same area as a 10×10 square grid array using same pitch. The pitch of any two adjacent I/Os is the minimum pitch and there are 105 I/Os totally. The values for feature sizes are same as the example in section 3.3.1.

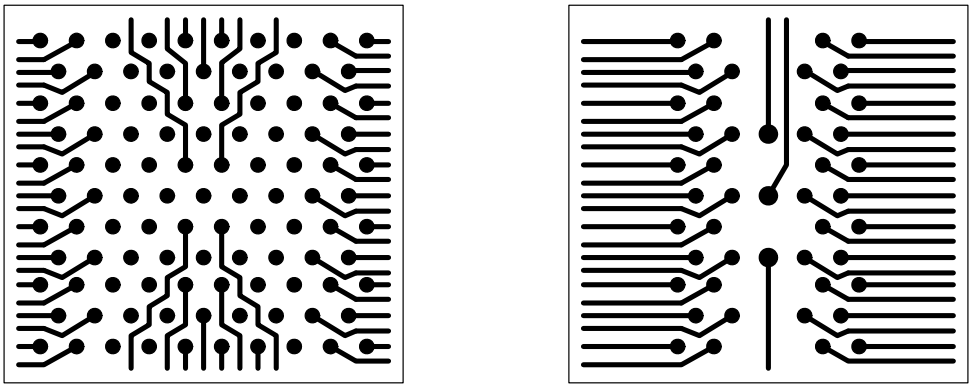
Using the column-by-column horizontal escape routing method, we need three layers as shown in Figure III. 7. However we only use two layers in two-sided horizontal/vertical escape routing approach as shown in Figure III. 8.



(a) First layer (a) Second layer

(a) Third layer

Figure III. 7 Column-by-column horizontal escape routing.



(a) First layer (b) Second layer

Figure III. 8 Two-sided horizontal/vertical escape routing.

The two-sided approach utilizes the routing channels sufficiently and increases the number of I/Os escaped in every routing layer because it breaks out I/Os inside and outside simultaneously. Compared with the column-by-column horizontal escape routing, it can decrease the number of routing layers efficiently.

3.2.3 Multi-direction Hybrid Channel Escape Routing

3.2.3.1 Array Partition and Hybrid Channel

The hexagonal array is symmetric in 0°, 60°, and 120° directions. It can be treated as many nested hexagons as shown in Figure III. 9. The array can be divided into six partitions and I/Os can be escaped to the outside following six directions. Figure III. 10 shows this escape routing style.

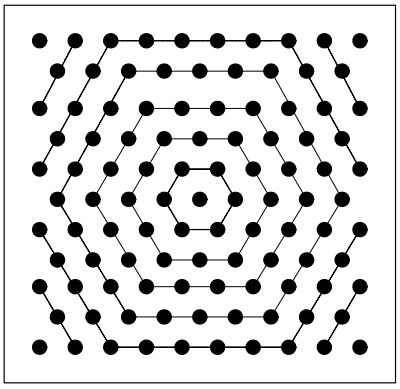


Figure III. 9 Nested hexagons.

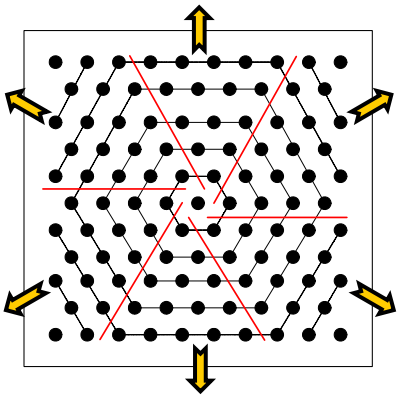


Figure III. 10 Hexagonal array partition.

For each partition, the adjacent I/Os in the same row have the minimum pitch. Instead of breaking out I/Os row by row, we can escaped the I/Os selectively to form indented outline and hybrid routing channels as shown in Figure III. 11.

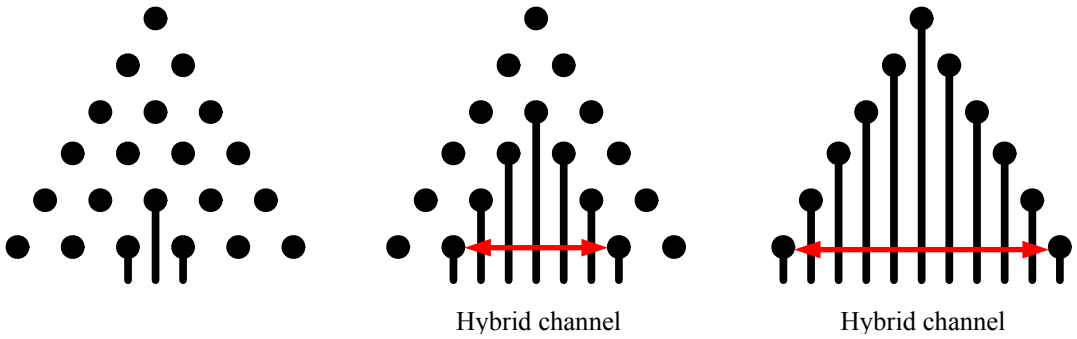


Figure III. 11 Indented outline and hybrid channel.

Because the hybrid routing channel has larger capacity, i.e. it can allow more wires going through, this escape strategy can increase the number of I/Os escaped for one routing layer and potentially reduce the number of routing layers compared with the column-by-column horizontal method. The number of wires going through the hybrid routing channel, which consists of k vertical channels, is

$$Num(wires)\_hybridChannel = \left\lfloor \frac{kP - D - S}{S + W} \right\rfloor \quad \text{(III.11)}$$

3.2.3.2 Escape Routing through Hybrid Channel

Using this multi-direction hybrid channel escape routing strategy, the hexagonal array can be treated as consisting of many indented rows as shown in Figure III. 12.

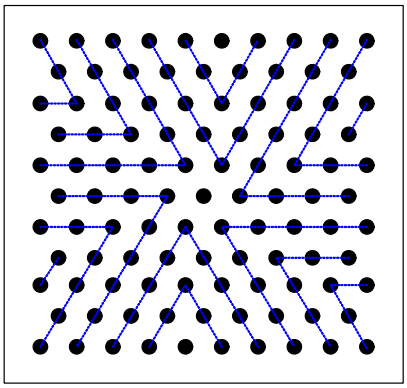


Figure III. 12 Indented rows in hexagonal array.

For an n×m hexagonal array in a square area, which means m and n satisfy the formula (III. 1), the number of indented rows in each partition is

$$I \le \left\lfloor \frac{n-2}{3} \right\rfloor + 1 \quad \text{(III.12)}$$

We assume the number of wires that can go through the vertical routing channel is A. Thus

$$Num(wires)\_verticalChannel = \left\lfloor \frac{P - D - S}{S + W} \right\rfloor = A \quad \text{(III.13)}$$

$$\Rightarrow P - D - S \ge A(S + W)$$

The condition for routing I/Os in A indented rows through the hybrid routing channel, which consists of k vertical channels, is

$$Num(wires)\_hybridChannel = \left\lfloor \frac{kP - D - S}{S + W} \right\rfloor \geq A(2k-1)$$
$$\Rightarrow kP - D - S \geq A(2k-1)(S+W) \qquad \text{(III.14)}$$

Plug (III.13) into (III.14), the condition can be simplified as

$$2(D+S) \geq P \qquad \text{(III.15)}$$

As shown in Table III. 4, the condition is generally satisfied.

Table III. 4 Practical parameters for condition, unit: μm**.**

	Flip Chip		FBGA/CSP	
Year	2006	2018	2006	2018
Pitch	130	70	300	100
Pad Diameter	65	35	120	40
Line Width	27.8	15	36	12
Line Spacing	27.9	15	36	12
Condition (III.15)	✓	✓	✓	✓

Using the multi-direction hybrid channel escape routing method, we break out I/Os at the center of each partition in the first layer to form an indented outline as large as possible and escape at least A indented rows in every following layer as long as the condition (III.15) is satisfied. I/Os in the array can be escaped very efficiently though the hybrid channel.

3.2.3.3 Automatic Escape Routing Procedure

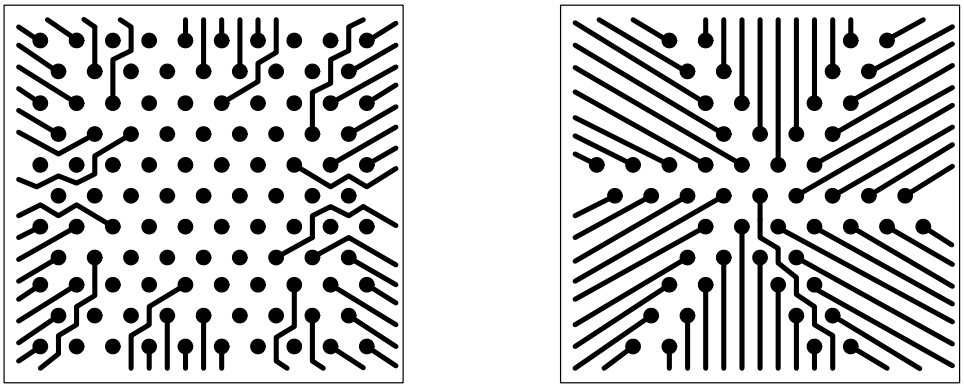
The hybrid escape routing approach organizes I/Os in the array regularly. The escape routing rules for each partition are identical and the routing for every partition is independent of each other. Furthermore the wires breaking out I/Os go through the hybrid channel orderly. Thus escape routing program can be designed straightforwardly to accomplish this kind of escape routing for any given hexagonal array automatically. The automatic procedure of the multi-direction hybrid channel escape routing is illustrated in Figure III. 13.

Partition:
Divide the hexagonal array into six partitions.
Labeling:
Identify the indented rows in each partition.
Label I/Os using partition ID and indented rows ID.
Assignment:
For each partition {
Determine the number of indented rows that can be escaped in the first routing layer according to the vertical channel capacity.
Determine the number of indented rows that can be escaped in each following layer according to the hybrid channel capacity.
}
Routing:
For each partition {
Escape I/Os in the first layer symmetrically through vertical channel.
Escape I/Os in each following layer directly through the hybrid channel.
}
Post processing refinement:
Refine the escape routing for I/Os located at special positions, e.g. center, corner, partition boundary, etc., to further reduce the number of routing layers.

Figure III. 13 Automatic escape routing.

3.2.3.4 Escape Routing Example

For the 10×11 hexagonal array example in section 3.3.2, we only need two layers using this multi-direction hybrid channel escape routing strategy as shown in Figure III. 14.



(a) First layer (b) Second layer

Figure III. 14 Multi-direction hybrid channel escape routing.

3.3 Analysis of Escape Routing for Hexagonal Array

The hexagonal array can hold more I/Os than the traditional square grid array using same area and same pitch, i.e. the hexagonal array has larger I/Os density. For complicated ICs and packages, the hexagonal array can provide more external interconnection. The approximate increase of I/Os count for large array is 15%. The hexagonal array has larger I/Os' pitch if it uses same area as the square grid array and holds similar number of I/Os. The large pitch will increase the escape routing capacities and can potentially reduce the number of layers for escape routing. Furthermore it can benefit the alleviation of crosstalk.

The column-by-column horizontal escape routing strategy is very straightforward for the hexagonal array. Using this strategy, I/Os in hexagonal array can be escaped in the same number of routing layers as the square grid array which has same area and same pitch although more I/Os are packed inside. However, the vertical routing channels are wasted in this strategy.

The two-sided horizontal/vertical escape routing strategy overcomes the shortcoming of that straightforward strategy. I/Os in the outside zigzag columns are escaped through horizontal channels and simultaneously I/Os in the middle zigzag columns are escaped through vertical routing channels. All the routing channels are utilized sufficiently and the number of escape routing layers is reduced. Nevertheless, there is no simple routing rule for the wires breaking out I/Os in the middle zigzag columns, so it's hard to implement this strategy in automatic program and those wires need to go through many other I/Os, thus crosstalk is an important issue to be considered.

The multi-direction hybrid channel escape routing strategy uses the symmetric property of hexagonal array to divide it into six partitions and exploits hybrid channels to increase the escape efficiency. The hybrid channels increase the number of escape routing

wires on every layer and consequentially decrease the number of layers. I/Os in each partition are escaped independently which makes the problem simpler and the wires routing through hybrid channels are very ordered, so this strategy is easy to be implemented in automatic program.

In summary, hexagonal array can be escaped very efficiently as well as providing high density of I/Os.

3.4 Analysis Model

The modeling for the escape routing in hexagonal area array is similar as that for square grid array. The direction of the escaping wires is more unrestricted. We could still utilize 3D electromagnetic simulation tool to extract the characteristics of the escaping wires. Further the circuit simulation could be applied to analyze the escape routing performance based on the extracted model.

Chapter 3, in part, is a reprint of the paper “Efficient Escape Routing for Hexagonal Array with high I/Os density” co-authored with Chung-Kuan Cheng in the proceedings of 43rd ACM/IEEE Design Automation Conference 2006. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Eye Diagram Prediction for High Speed Signaling

In high speed signaling system, the distortion, noise and interference on signal waveforms will constrain the system performance. An eye diagram provides the most fundamental and intuitive view to evaluate the signal quality of high speed communication.

The traditional method to obtain the eye diagram involves performing a time domain simulation. Due to the long simulation times, designers usually use a limited length pseudorandom bit sequence (PRBS) as the input stimulus. The eye diagram measurement derived from a limited length PRBS usually is better than the worst-case for many high speed signaling systems.

For various high speed signaling systems, the digital signals may have symmetric or asymmetric rise/fall time due to variations of the transmitter circuit designs [13][35]. An efficient and general method will be very helpful to analyze the system performance.

In this chapter, we introduce an efficient and accurate method, the accumulative prediction method, to predict the signal distortion from inter-symbol interference. This method predicts the worst-case eye diagram based on step response of the signaling system. It extracts

the eye opening and timing jitter and generates the input data patterns which produce the worst-case inter-symbol interference. The main advantage is that this general-purpose method can handle signals with asymmetric as well as symmetric rise/fall time. Furthermore, the complexity of the proposed method is linear O(n), where n is the number of time points of step response. Experimental results demonstrate the accuracy and efficiency of the proposed method. As one application, this method could be applied to analyze the signal performance of escaping wires.

4.1 Preliminaries

For linear time-invariant signaling system, the transmitted digital signal can be modeled as a linear combination of time shifted step responses. Hence, we can analyze the eye diagram based on the step response of the signaling system.

4.1.1 Step Response and Digital Signal Communication

The step response of a general system is the time behavior of the outputs when its inputs change from zero to one in a very short time. For a signaling system, its step response implicates certain characteristics of the system. For example, the step response reveals the time-of-flight delay, the reflection and the saturation voltage for a transmission line communication.

The input signal for a digital signal system is a bit stream which consists of a series of binary bits, zeros and ones. We can model the input signal x(t) as a linear combination of a series of transitions, zero to one transition alternating with zero to minus one transition. We represent x(t) as

$$\begin{aligned} x(t) = {} & x_r(t - k^r{}_1 T) + x_f(t - k^f{}_1 T) + \ldots \\ & + x_r(t - k^r{}_i T) + x_f(t - k^f{}_i T) + \ldots \end{aligned} \tag{IV.1}$$

where $x_r(t)$ is the zero to one transition waveform and the transition time is equal to input symbol rise time, $x_f(t)$ is the zero to minus one transition waveform and the transition time is equal to input symbol fall time, and T is the input symbol period. The coefficients k^r_i and k^f_i represent the transition time; they are non-negative integers and satisfy the following relationship because the zero to one transition must be alternating with zero to minus one transition

$$k^r_1 > k^f_1 > k^r_2 > k^f_2 > \ldots > k^r_i > k^f_i > \ldots \quad \text{(IV.2)}$$

Therefore, for linear time-invariant signaling system, the output signal y(t) can be modeled as a linear combination of time shifted step responses

$$y(t) = s_r(t - k^r{}_1 T) - s_f(t - k^f{}_1 T) + \ldots + s_r(t - k^r{}_i T) - s_f(t - k^f{}_i T) + \ldots \quad \text{(IV.3)}$$

where $s_r(t)$ is the step response for input transition time equal to symbol rise time and $s_f(t)$ is the step response for input transition time equal to symbol fall time.

Figure IV. 1 shows the relationship between the step response and the digital signal. The input signal x(t) is the combination of transitions and the output signal y(t) is the combination of step responses.

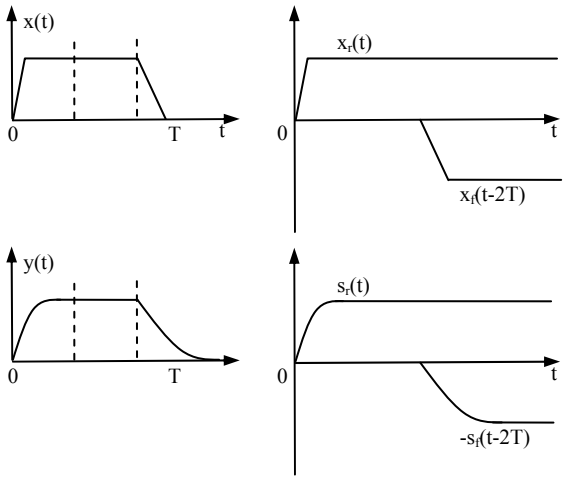


Figure IV. 1 Step response and digital signals.

4.1.2 Eye Diagram

The eye diagram is an oscilloscope display in which a digital signal at the receiver side is repetitively sampled to get a good representation. It's created by taking the time domain signal and overlapping the waveform for a certain time window of multiple symbol periods, as shown in Figure IV. 2. The eye diagram is a useful and intuitive technique for the qualitative analysis of signal performance in digital transmission. Several system performance measures can be derived from the eye diagram. In general, the following features of the eye diagram are defined

1) Eye opening (height, peak to peak), measure of the additive noise in the signal;
2) Eye overshoot/undershoot, measure of the peak distortion;
3) Eye width, measure of timing jitter effects.

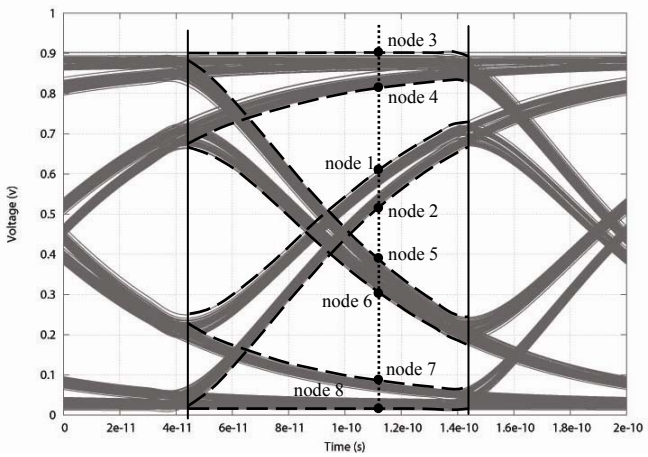


Figure IV. 2 Eye Diagram.

Eight voltage bounds for any time points.

The accumulative prediction method we developed calculates the signal distortion from inter-symbol interference and predicts worst-case eye diagram. Furthermore, two important evaluation features of eye diagram, eye opening and timing jitter are extracted. In our analysis, the eye diagram is created by overlapping the signal waveform in the time

window of one symbol period T. There are eight voltage bounds for every time point in the eye diagram, as shown in Figure IV. 2:

1) ‘1’ edge rising upper and lower bound, representing the maximum and minimum voltage for zero to one transition, as shown in Figure IV. 2 node 1 and node 2;
2) ‘1’ edge holding upper and lower bound, representing the maximum and minimum voltage for continuous one bits, as shown in Figure IV. 2 node 3 and node 4;
3) ‘0’ edge falling upper and lower bound, representing the maximum and minimum voltage for one to zero transition, as shown in Figure IV. 2 node 5 and node 6;
4) ‘0’ edge holding upper and lower bound, representing the maximum and minimum voltage for continuous zero bits, as shown in Figure IV. 2 node 7 and node 8.

Here, ‘1’ edge represents the transmitted symbol is one. ‘1’ edge rising means there’s a zero to one transition right before this symbol while ‘1’ edge holding means the previous symbol is also one. The notations for ‘0’ are similar.

These eight voltage bounds for all the time points in the overlapping time window will capture all the valuable features of the eye diagram.

4.2 Worst-case Eye Diagram Prediction

The accumulative prediction method we proposed predicts the worst-case eye diagram shape based on step responses. The eye opening and timing jitter are extracted for the worst-case eye diagram. The input data patterns, which produce the worst-case inter-symbol interference, can also be generated.

4.2.1 Voltage Bounds for Worst-case Eye Diagram

The eye diagram is created by overlapping the signal waveform in the time window of one symbol period T. The waveform in the eye diagram can be given by

$$e(t) = \begin{cases} y(t) \\ y(t+T) \\ \dots \\ y(t+k_i T) \\ \dots \end{cases} \quad t \in [0,T] \tag{IV.4}$$

The eight voltage bounds for any time point, e(t) t∈[0,T], can be derived from the output signal y(t) at the receiver side. As we have discussed, for linear time-invariant signaling system, the output signal y(t) can be modeled as a linear combination of time shifted step responses, equation (IV.3). We can shift the time axis to the observing bit, then the residue memory of previous data bits acts as the voltage offset. The maximum and minimum value of $e(t_o)$, t_o is the observing time point $t_o \in [0,T]$, will give the voltage bounds for the worst-case eye diagram.

Theorem 1: Given the step response $s_r(t)$ and $s_f(t)$, the eight voltage bounds are determined by the equations in Table IV. 1.

Table IV. 1 Voltage bounds equations.

Voltage bounds		Optimize	f(t)
'1' edge rising upper bound	$e_{upper01}(t_o)$	max(f(t))	$s_r(t_o) + \sum_i (-s_f(t_o + k^f{}_i T) + s_r(t_o + k^r{}_i T))$
'1' edge rising lower bound	$e_{lower01}(t_o)$	min(f(t))	
'1' edge holding upper bound	$e_{upper11}(t_o)$	max(f(t))	$s_r(t_o + k^o{}_r T) + \sum_i (-s_f(t_o + k^f{}_i T) + s_r(t_o + k^r{}_i T))$
'1' edge holding lower bound	$e_{lower11}(t_o)$	min(f(t))	
'0' edge rising upper bound	$e_{upper10}(t_o)$	max(f(t))	$-s_f(t_o) + s_r(t_o + k^o{}_r T) + \sum_i (-s_f(t_o + k^f{}_i T) + s_r(t_o + k^r{}_i T))$
'0' edge rising lower bound	$e_{lower10}(t_o)$	min(f(t))	
'0' edge holding upper bound	$e_{upper00}(t_o)$	max(f(t))	$-s_f(t_o + k^o{}_f T) + s_r(t_o + k^o{}_r T) + \sum_i (-s_f(t_o + k^f{}_i T) + s_r(t_o + k^r{}_i T))$
'0' edge holding lower bound	$e_{lower00}(t_o)$	min(f(t))	

The coefficients k^r_i and k^f_i represents the transition time; they are positive integers and satisfy that

$$k^f_1 < k^r_1 < k^f_2 < k^r_2 < \ldots < k^f_i < k^r_i < \ldots \quad \text{(IV.5)}$$

Theorem 2: The eight voltage bounds are the worst-case eye diagram bounds when the given step response $s_r(t)$ and $s_f(t)$ reach saturation voltage.

In order to obtain these eight voltage bounds, we formulate them as a uniform optimization problem and devise a dynamic programming algorithm to solve it. Make a comprehensive view of the equations in theorem 1, we can formulate the optimization problem as

> ***Given:*** two arrays A and B
> A=$\{s_r(t_o+T), s_r(t_o+2T), \ldots, s_r(t_o+k_mT)\}$
> B =$\{s_f(t_o+T), s_f(t_o+2T), \ldots, s_f(t_o+k_mT)\}$
> where $s_r(t)$ and $s_f(t)$ have reached saturation voltage at time (t_o+k_mT)
>
> ***Objective:*** pick elements from A and B alternatively to minimize/maximize
> $$\sum_i A[i] - \sum_j B[j]$$
> ***Constraint:*** the ending element must be selected from array A, but the starting element (from A or B) is different for different voltage bounds.

Dynamic programming can be applied to solve this formulated optimization problem efficiently. If we assume the optimal solution for A[1,…,m-1] and B[1,…,m-1] is known, the optimal solution for A[1,…,m] and B[1,..,m] can be obtained by simple arithmetical operation and comparison straightforwardly. Also the optimal solution for arrays with one or two elements can be obtained trivially. So we can use dynamic programming to solve the problem.

The detailed algorithm for '1' edge rising lower bound is shown in Figure IV. 3. For this bound, the starting element is selected from B. The obtained optimal solution, called minDeltaV, is just for the voltage offset and the '1' edge rising lower bound is given by

$$y_{lower01}(t_o) = s_r(t_o) + \text{minDeltaV} \quad \text{(IV.6)}$$

```
calBound(A, B)
   A_min[1] ← no meaning
   B_min[1] ← -B[1]
   A_min[2] ← -B[1]+A[2]
   B_min[2] ← -B[2]
   min_A ← A_min[2]
   min_B ← min(B_min[1], B_min[2])
   for i = 3 to k_m
      do A_min[i] ← min_B+A[i]
         B_min[i] ← min(min_A-B[i], -B[i])
         min_A ← min(min_A, A_min[i]);
         min_B ← min(min_B, B_min[i]);
      minDeltaV = min_A
   return(minDeltaV)
```

Figure IV. 3 Algorithm for '1' edge rising lower bound.

The algorithms for other voltage bounds are similar.

Figure IV. 4 and Table IV. 2 show a simple example of calculating the voltage offset for '1' edge rising lower bound using the proposed algorithm. The step response $s_r(t)$ and $s_f(t)$ are shown in Figure IV. 4 and the sampling points starting from the observing time point t_o are marked. Table IV. 2 shows the dynamic programming step by step. The values in the table are the optimal solution if the corresponding array element is selected. The values with gray background are the optimal solution for current step. Finally, we found the optimal solution for this example is -0.14 and it's obtained by selecting B[2], A[3], B[4] and A[5].

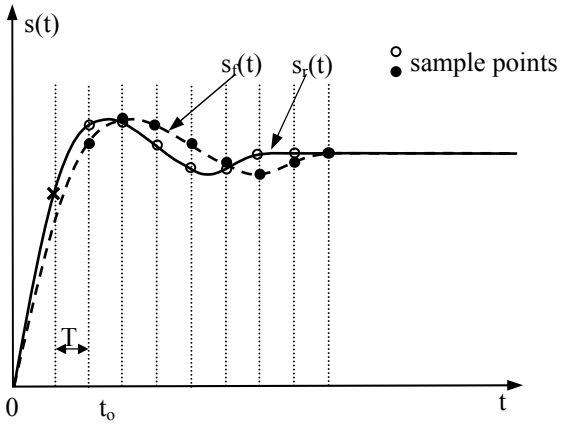


Figure IV. 4 Step response example.

Table IV. 2 Example: find minimum value starting from array B.

t	t_o+T	t_o+2T	t_o+3T	t_o+4T	t_o+5T	t_o+6T	t_o+7T	t_o+8T
A $s_r(t)$ v	0.96	0.97	0.90	0.85	0.85	0.88	0.89	0.89
B $s_f(t)$ v	0.91	0.98	0.96	0.91	0.86	0.83	0.86	0.89
step 1	-	0.06						
	-0.91	-0.98						
step 2	-	0.06	-0.08					
	-0.91	-0.98	-0.96					
step 3	-	0.06	-0.08	-0.13				
	-0.91	-0.98	-0.96	-0.99				
step 4	-	0.06	-0.08	-0.13	-0.14			
	-0.91	-0.98	-0.96	-0.99	-0.99			
step 5	-	0.06	-0.08	-0.13	-0.14	-0.11		
	-0.91	-0.98	-0.96	-0.99	-0.99	-0.97		
step 6	-	0.06	-0.08	-0.13	-0.14	-0.11	-0.10	
	-0.91	-0.98	-0.96	-0.99	-0.99	-0.97	-1.00	
step 7	-	0.06	-0.08	-0.13	-0.14	-0.11	-0.10	-0.11
	-0.91	-0.98	-0.96	-0.99	-0.99	-0.97	-1.00	-1.03

4.2.2 Worst-case Eye Opening

The eye opening is the difference between the minimum value of samples related to a logical ‘1’ and the maximum value of samples related to a logical ‘0’, measured at the sampling instant. Usually the eye opening can be captured at the symbol transition point. Therefore, we set the observing time point at T to extract the eye opening. Based on the voltage bounds equations, the worst-case eye opening is give by

$$V_{eye} = \min(y_{lower01}(T), y_{lower11}(T)) - \max(y_{upper10}(T), y_{upper00}(T)) \quad \text{(IV.7)}$$

4.2.3 Worst-case Timing Jitter

The timing jitter measures the variance in the actual transition time from the ideal transition time. We define the timing jitter as the deviation of transitions crossing the half of saturation voltage. As shown in Figure IV. 5, for an open eye, the timing jitter is given by

$$T_{jitter} = T_{jright} - T_{jleft} \quad \text{(IV.8)}$$

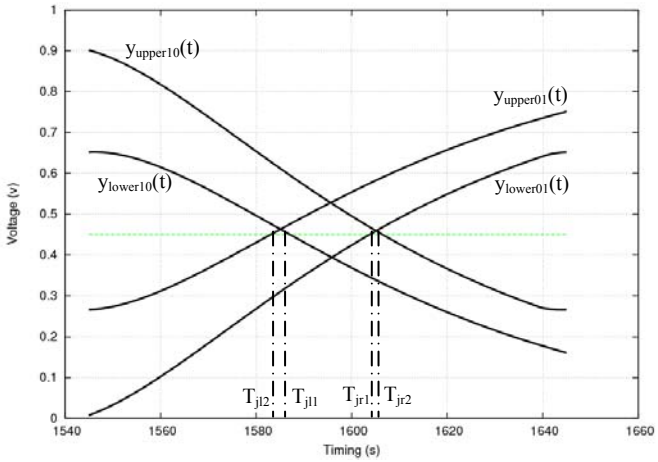


Figure IV. 5 Timing jitter.

The right and left time boundaries are the time points where certain voltage bound crosses the half saturation voltage. They are determined by the following equations

$$T_{jright} = \max(T_{jr1}, T_{jr2}) \qquad T_{jr1} \in (0,T] \text{ and } T_{jr2} \in (0,T] \tag{IV.9}$$

$$y_{lower01}(T_{jr1}) = V_{sat}/2$$

$$y_{upper10}(T_{jr2}) = V_{sat}/2$$

$$T_{jleft} = \min(T_{jl1}, T_{jl2}) \qquad T_{jl1} \in (0,T] \text{ and } T_{jl2} \in (0,T] \tag{IV.10}$$

$$y_{lower10}(T_{jl1}) = V_{sat}/2$$

$$y_{upper01}(T_{jl2}) = V_{sat}/2$$

We can use binary search method on the time region (0,T] to find T_{jr1}, T_{jr2} and T_{jl1}, T_{jl2} efficiently.

4.2.4 Worst-case Input Data Pattern

The accumulative prediction method can also generate the input data patterns which produce the worst-case intersymbol interference. This method calculates the voltage bounds for certain time point by determining the bit transitions. Thus the input data patterns, which will make the worst-case eye diagram happen, can be generated by backtracking the bit transitions.

4.3 Analysis of Prediction

The accumulative prediction method can predict the worst-case eye efficiently and accurately. It has linear complexity and its error sources limit the prediction error.

4.3.1 Prediction Complexity

We assume the number of sampled time points in the step response is n. We assume the time step is uniform and we should have an enough amount (T_N) of sample points in one symbol period for accuracy. Then in the accumulative prediction method, the array size m for each observe time point t_o is given by

$$m = \left\lceil \frac{n}{T_N} \right\rceil \tag{IV.11}$$

The complexity of the dynamic programming algorithm discussed in section 4.3 is O(m) obviously. Thus, the total complexity of calculating the bounds for all time points in one symbol period is given by

$$T_N \times O(m) = O(n) \tag{IV.12}$$

The complexity for calculating worst-case eye opening is O(m) because only four bounds for time point T are needed. The worst-case timing jitter applies binary search, so the complexity is O(mlgn).

4.3.2 Prediction Error

There are two potential error sources for our accumulative prediction method. The first error source is the sampling time points. Because we only consider discrete sampling time points, inevitable error will be introduced. More sampling time points with smaller time step will achieve better accuracy with the cost of complexity. The other error source is the saturation voltage. The step response voltage will fluctuate in a very small range around

saturation voltage after certain simulation time. It's hard to obtain the exact saturation voltage value. Also the effective voltage in the input bit pattern is actually determined by the specific elapsed time. This small variation will introduce certain prediction error.

4.3.3 Digital Signal with Symmetric and Asymmetric Rise/Fall Time

The accumulative prediction method we proposed deals with zero to one transition and one to zero transition separately. Therefore, it is general-purpose and it can handle the digital signal with symmetric or asymmetric rise/fall time in the same way.

The peak distortion analysis [3] developed by Casper et al. calculates the worst-case voltage or timing margin based on the unit pulse response. It can be only applied for the digital signal with symmetric rise/fall time because the data patterns are not linear combination of pulse input if the rise/fall time is asymmetric. As shown in Figure IV. 6 bold lines, the rise and fall edges cannot be combined to get the digital data.

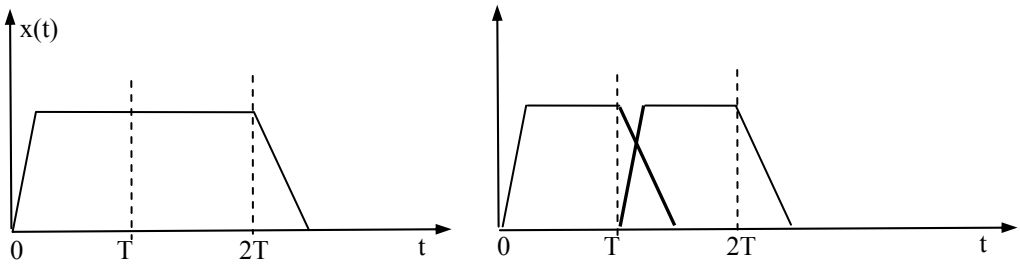


Figure IV. 6 Unit pulse response doesn't work

for signal with asymmetric rise/fall time

For symmetric rise/fall time digital signal, our accumulative prediction method can be proved to be equivalent to the peak distortion analysis. Also both of them have linear complexity O(n), where n is the number of time points in step response or unit pulse response.

Theorem 3: the accumulative prediction is equivalent to the peak distortion analysis for signals with symmetric rise/fall time.

Sketch of proof: according to our formulation, the unit pulse response can be give by

$$p(t) = s_r(t) - s_f(t-T) \qquad \text{(IV.13)}$$

In voltage bounds analysis, the unit pulse response corresponds to

$$- s_f(t_o) + s_r(t_o+T) \qquad \text{(IV.14)}$$

Because the rise/fall time are equal, the step response $s_r(t)$ and $s_f(t)$ will be the same. In our analysis, the corresponding elements in array A and B will also be equal. Hence, any feasible solution for the formulated optimization problem can be rewritten as the combination of pulse response

$$\begin{aligned}
&\sum_i A[i] - \sum_j B[j] \\
&= \sum_i (-s_f(t_o + k_{fi}T) + s_r(t_o + k_{ri}T)) \\
&= \sum_i [-s_f(t_o + k_{fi}T) + s_r(t_o + (k_{fi}+1)T)) \\
&\quad -s_f(t_o + (k_{fi}+1)T) + s_r(t_o + (k_{fi}+2)T) \\
&\quad \cdots \\
&\quad -s_f(t_o + (k_{ri}-1)T) + s_r(t_o + k_{ri}T)]
\end{aligned} \qquad \text{(IV.15)}$$

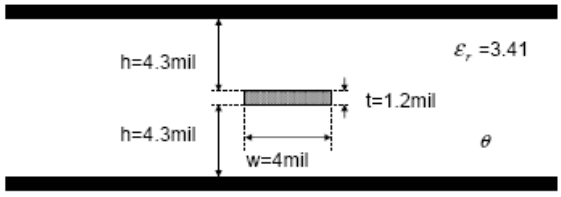
Obviously, the solution for minimization will pick all the negative pulse response and the solution for maximization will pick all the positive pulse response. Our accumulative prediction method is completely equivalent to the peak distortion analysis for symmetric rise/fall time signal.

4.4 Verification Experiments

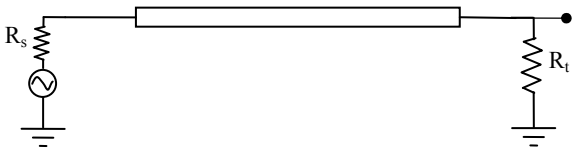
We implemented the accumulative prediction method in C program, which reads in step response, calculates the worst-case eye diagram shape, extracts eye opening and timing jitter and generates the worst-case input data patterns. With a transmission-line case, the accuracy and efficiency of the proposed accumulative prediction method is demonstrated.

4.4.1 Experiment Setup

The single-end transmission line is on printed circuit board (PCB) with FR4 materials, transmitting signals among processors and memory. The structure and parameters are shown in Figure IV. 7 (a). The length of the transmission is 25cm and the characteristic impedance is 50Ω. As shown in Figure IV. 7 (b), a voltage source with Rs=4Ω produces the input signal at the driver side and the voltage Vdd is 1V. At the receiver side, a resistor is connected as the termination. We use IBM PowerSPICE [40] to do the transient simulation and in the simulation, frequency-dependent RLCG table model is applied, which is extracted by IBM CZ2D package [41].



(a) The cross section of the single-end transmission line



(b) The signaling scheme

Figure IV. 7 The single-end transmission line structure.

We test two sets experiments, one for symmetric rise/fall time and another for asymmetric rise/fall time. For each set of experiments, three simulations are executed

1) Run step response simulation and predict the worst-case eye diagram using our accumulative prediction method;
2) Run transient simulation with worst-case input patterns generated;
3) Run transient simulation with pseudorandom bit sequence.

The accuracy and CPU time will be compared.

4.4.2 Experimental Results

We assume the tested symbol period is 100ps, the rise/fall time is 10ps for symmetric signal and the rise/fall time is 10ps/15ps respectively for asymmetric signal. The experimental results for the signal with symmetric rise/fall time are shown in Table IV.3, IV.4and IV.5 and the experimental results for the signal with asymmetric rise/fall time are shown in Table IV.6, IV.7 and IV.8. We tested 10 cases with different termination impedance, from 32Ω to 68Ω.

Figure IV. 8 show the eye opening and timing jitter curves for the symmetric rise/fall time simulations which correspond to the data in Table IV. 3 and Table IV. 4. Figure IV. 9 show the eye opening and timing jitter curves for the asymmetric rise/fall time simulations which correspond to the data in Table IV. 6 and Table IV. 7.

The experimental results demonstrate that the prediction results match the simulation results with worst-case input patterns very well. However, the simulation results using PRBS as stimulus cannot capture the worst-case eye diagram. The longer is the input bit sequence, the better result the PRBS simulation can achieve. The average errors compared with the results form worst-case input patterns simulation are given in the last row of every table. Figure IV. 10 and Figure IV. 11 show the eye diagrams from the simulation with different stimulus.

Table IV. 5 and Table IV. 8 show the average CPU times for the simulations. Compared with PRBS simulation, our prediction method is very efficient.

Table IV. 3 Eye opening, symmetric rise/fall time.

Rt	Eye Opening (V)			
	100 PRBS	10000 PRBS	Worst-case Input	Predict
32	0.24162999	0.19348612	0.14754779	0.146586
36	0.28892423	0.25287776	0.20618604	0.205273
40	0.33142498	0.30341259	0.26091507	0.260058
44	0.36606853	0.34187927	0.31197535	0.311138
48	0.39882782	0.37755125	0.35955214	0.358757
52	0.42980694	0.41067793	0.388297	0.387563
56	0.4519329	0.431006	0.37934625	0.378743
60	0.45400906	0.43434301	0.36499717	0.364447
64	0.44600999	0.42910893	0.34778497	0.347413
68	0.4370765	0.42199397	0.32841799	0.328084
Average relative error	27.47%	17.49%	0	-0.26%

Table IV. 4 Timing jitter, symmetric rise/fall time.

Rt	Timing Jitter (ps)			
	100 PRBS	10000 PRBS	Worst-case Input	Predict
32	44.0	44.0	51.3	51.4
36	32.7	32.7	41.4	41.5
40	25.2	25.2	33.6	33.7
44	19.6	19.6	27.0	27.4
48	17.5	17.3	21.9	22.1
52	15.6	14.9	20.1	20.0
56	15.8	16.7	23.3	23.4
60	16.1	19.2	27.2	27.2
64	20.6	21.6	31.1	31.1
68	23.3	24.9	35.0	35.1
Average relative error	-27.03%	-25.17%	0	0.33%

Table IV. 5 Average CPU time, symmetric rise/fall time.

100 PRBS	10000 PRBS	Worst-case Input	Step Response	Predict
5.8 s	14 m 55.4 s	2 m 1 s	26.9 s	0.345 s

Table IV. 6 Eye opening, asymmetric rise/fall time.

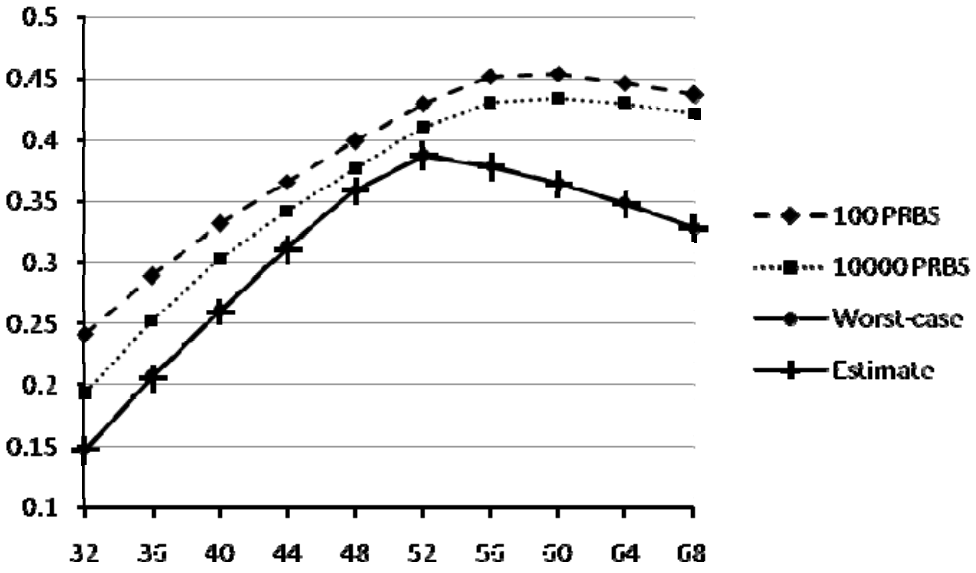
Rt	Eye Opening (V)			
	100 PRBS	5000 PRBS	Worst-case Input	Predict
32	0.24085751	0.19462714	0.14594329	0.144866
36	0.28806198	0.25378747	0.20443388	0.203427
40	0.33045842	0.30383077	0.25906041	0.258093
44	0.36473602	0.341818	0.30998847	0.309069
48	0.39714371	0.37704639	0.35748097	0.356586
52	0.42778562	0.40976195	0.38611021	0.385301
56	0.44792679	0.42957339	0.37709847	0.376400
60	0.44738613	0.43173967	0.36266126	0.362026
64	0.43849474	0.42617642	0.34534802	0.344920
68	0.42877489	0.41812539	0.32593423	0.325526
Average relative error	27.37%	18.13%	0	-0.30%

Table IV. 7 Timing jitter, asymmetric rise/fall time.

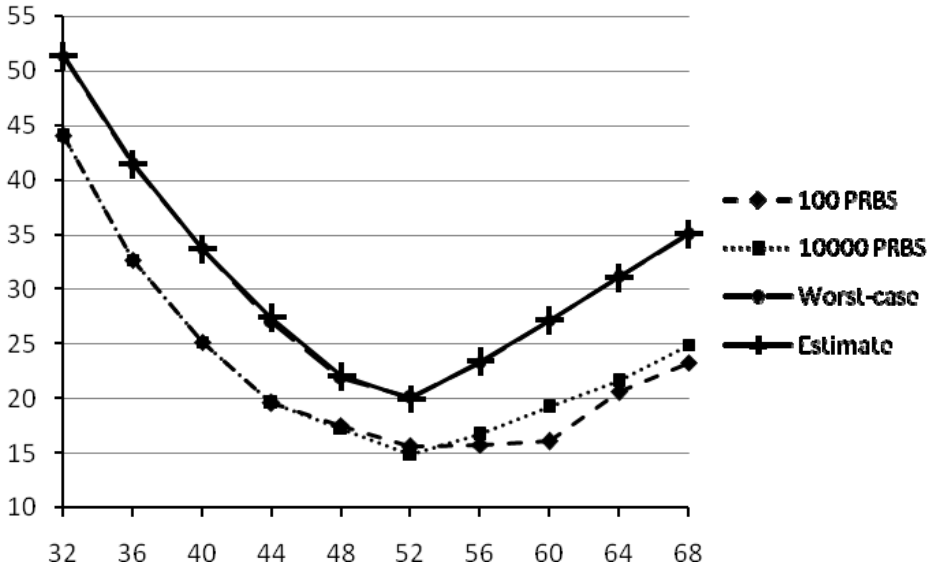
Rt	Timing Jitter (ps)			
	100 PRBS	5000 PRBS	Worst-case Input	Predict
32	55.0	55.0	54.9	55.0
36	45.0	45.0	44.9	45.0
40	37.2	37.2	36.9	37.1
44	30.8	30.8	30.6	30.7
48	25.5	25.5	25.4	25.4
52	21.2	21.2	23.3	23.2
56	19.3	20.1	26.7	26.6
60	20.4	22.6	30.6	30.5
64	22.8	25.1	34.4	34.4
68	25.7	27.6	38.5	38.4
Average relative error	-13.8%	-11.30%	0	-0.01%

Table IV. 8 Average CPU time, asymmetric rise/fall time.

100 PRBS	5000 PRBS	Worst-case Input	Step Response	Predict
5.9 s	5 m 53.2 s	3 m 3 s	54 s	0.344 s

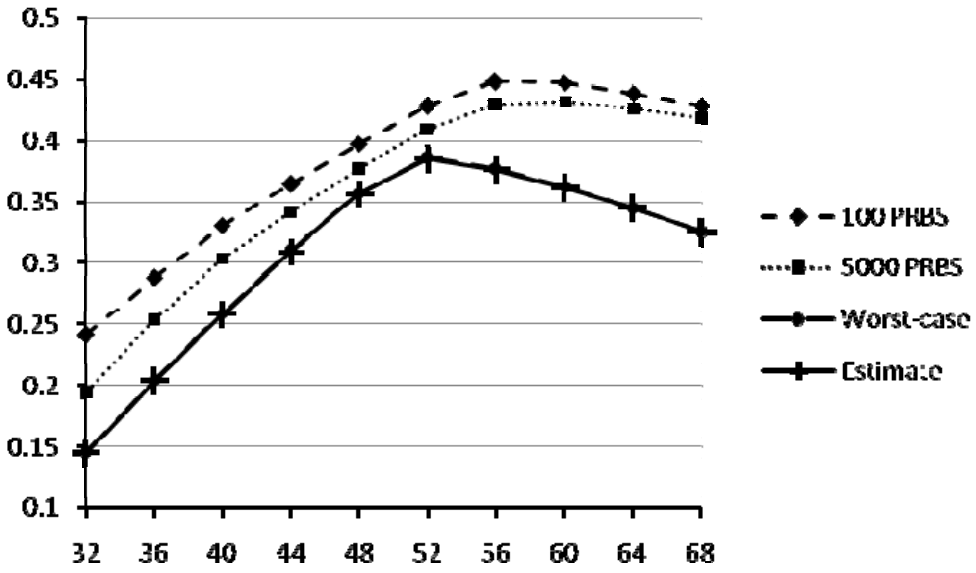


(a) Eye opening (v) vs. Rt (Ω)

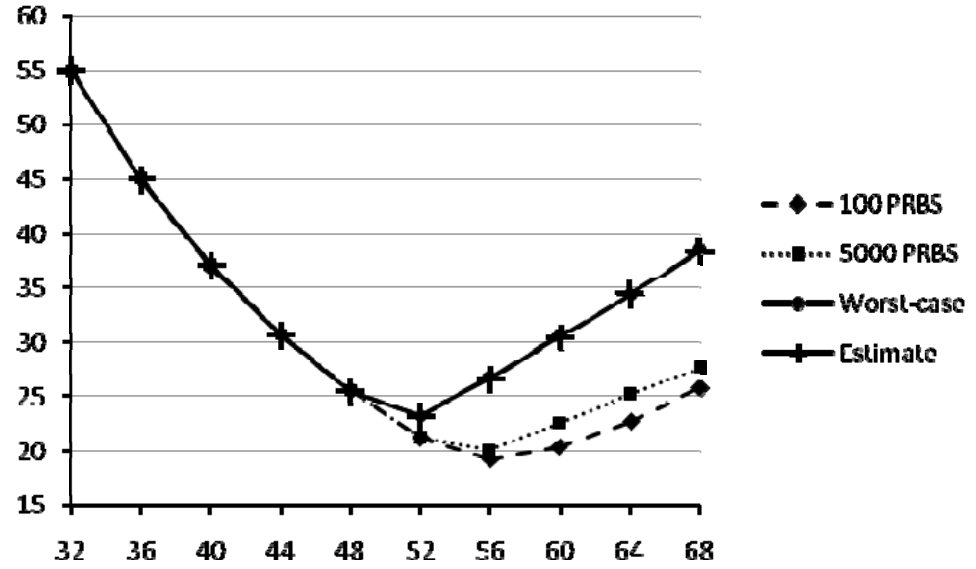


(b) Timing Jitter (ps) vs. Rt (Ω)

Figure IV. 8 Results for signal with symmetric rise/fall time.

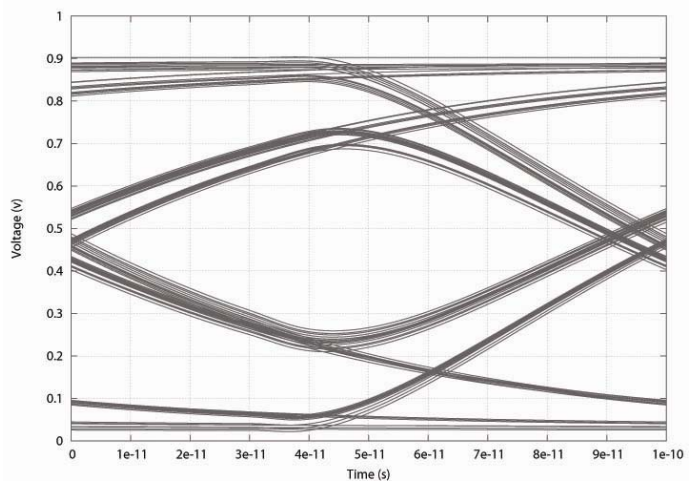


(a) Eye opening (v) vs. Rt (Ω)

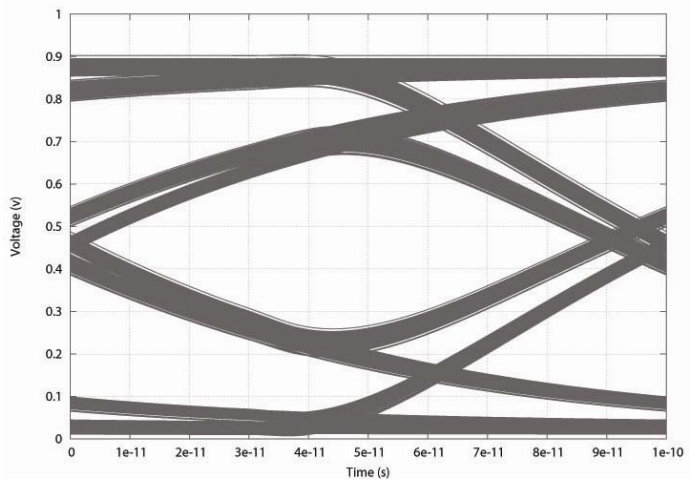


(b) Timing Jitter (ps) vs. Rt (Ω)

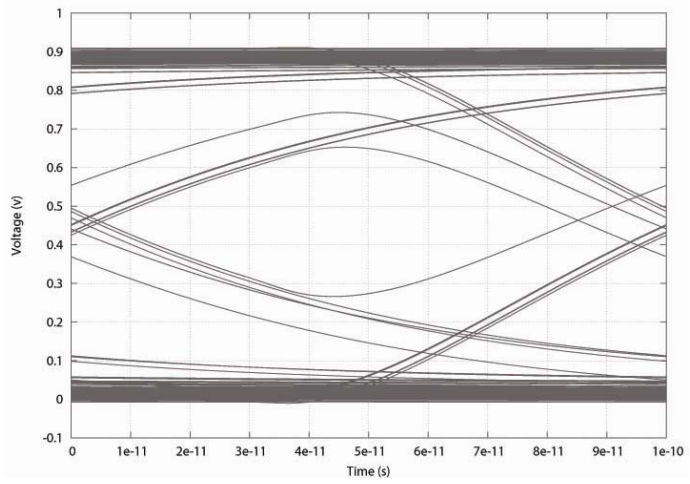
Figure IV. 9 Results for signal with asymmetric rise/fall time.



(a) 100 PRBS

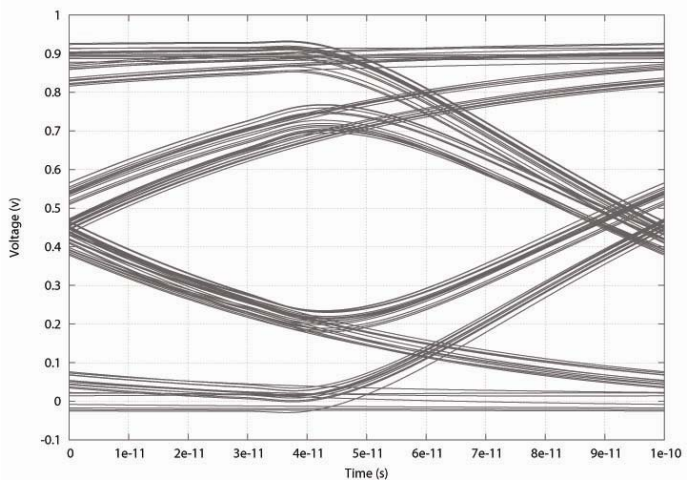


(b) 5000 PRBS

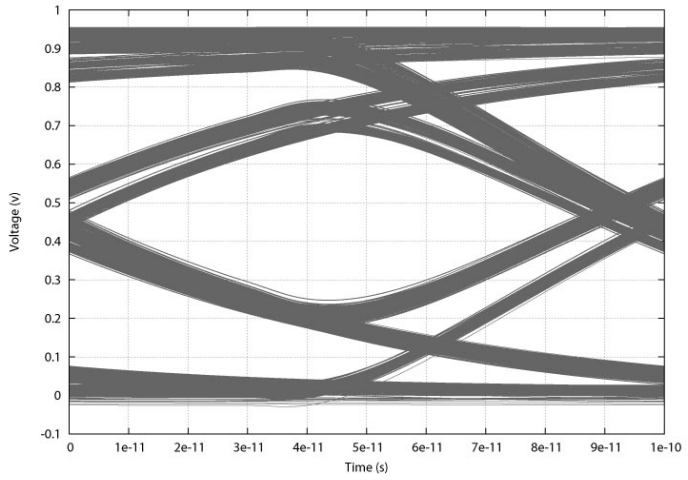


(c) worst-case input

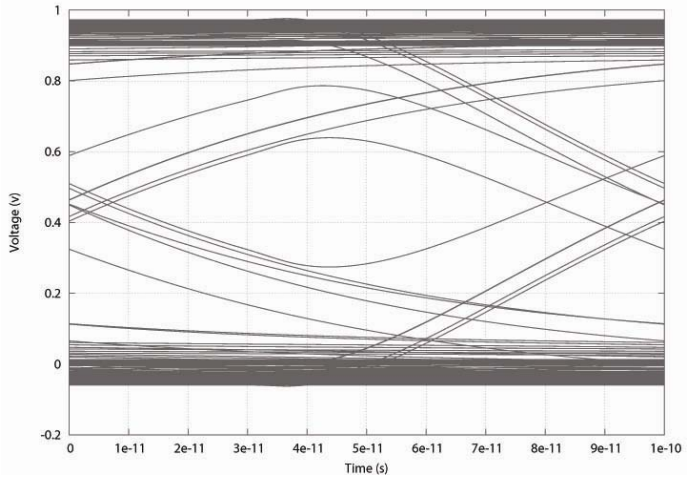
Figure IV. 10 Eye diagram for signal with asymmetric rise/fall time, Rt=52 Ω.



(a) 100 PRBS



(b) 10000 PRBS



(c) worst-case input

Figure IV. 11 Eye diagram for signal with symmetric rise/fall time, Rt=60 Ω.

4.5 Escaping Wires Analysis

The eye diagram prediction could be applied to escape routing signal analysis based on the analysis model discussed in chapter 2 and chapter 3. We extract and simulate the basic escaping wire structure shown in Figure II. 15. The parameters are the same as the verification case shown in Figure IV. 7 (a). The width of the wire is 4mil and the height of the wire is 1.2mil. The height of the dielectric is 4mil from wire to the top and bottom ground plan. The length is 100mil 45 degree wire and 900mil horizontal wire. A voltage source with Rs=4Ω produces the input signal at the driver side and the voltage Vdd is 1V. At the receiver side, a resistor 50Ω is connected as the termination.

We assume the symbol period is 1000ps and the rise/fall time is symmetric of 100ps. The step response of the escaping wire is shown in Figure IV. 12. The eye-opening and timing jitter are 0.545129V and 240.97ps respectively predicted by our accumulative prediction method. They are perfectly matched the simulation results with worst-case input patterns, 0.544915V and 241.3ps. The eye diagram of the worst-case simulation is shown in Figure IV. 13.

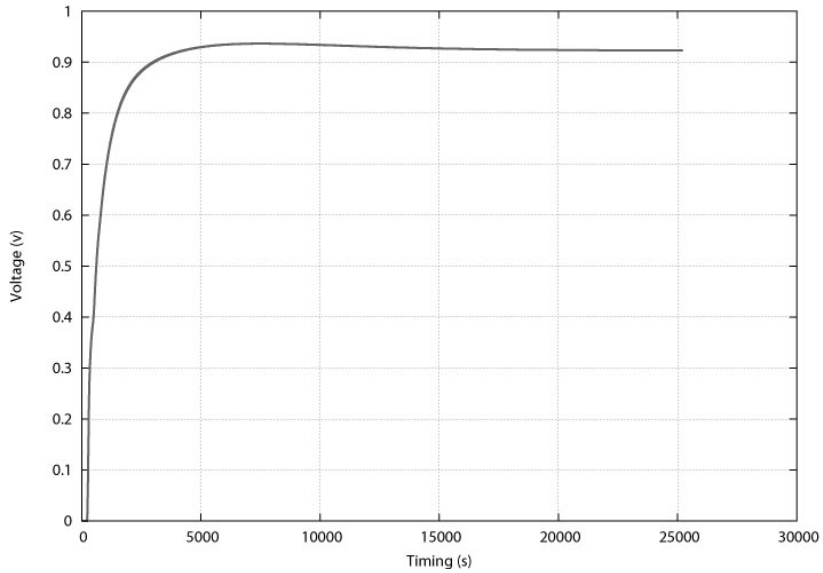


Figure IV. 12 Step response of escaping wire

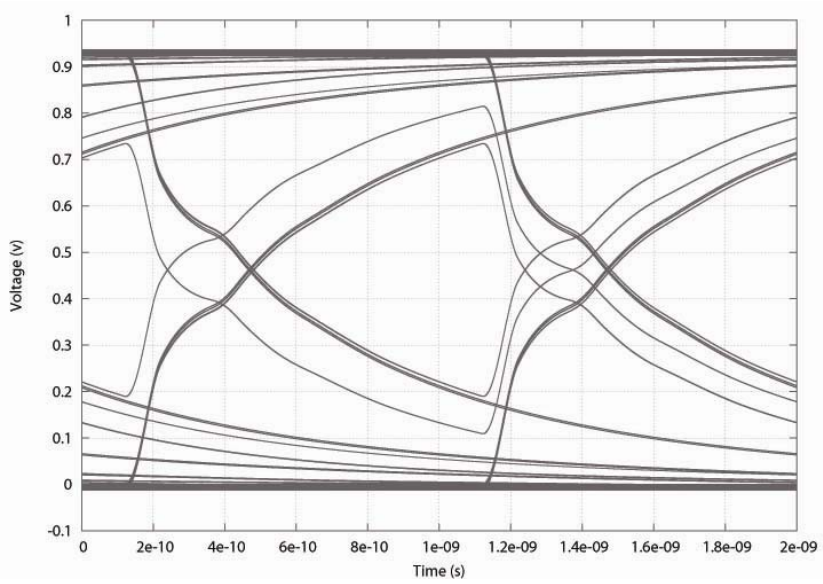


Figure IV. 13 Eye diagram of escaping wire

Chapter 4, in part, is a reprint of the paper “Efficient and Accurate Eye Diagram Prediction for High Speed Signaling” co-authored with Wenjian Yu, Yi Zhu, Chung-Kuan Cheng and Ernest S. Kuh in the proceedings of IEEE/ACM International Conference on

Computer-Aided Design 2008. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Conclusion

5.1 Dissertation Contribution

As the feature size of microelectronic technology becomes smaller, the complexity of the electronic systems grows proportionally and new features like high-speed, low power and low cost, become the key issues. The design and analysis of the interconnection in the electronic systems are more and more important for the performance and cost of the whole system. The escape routing design is a new important problem in the interconnection design because the number of chip/package input/output (I/O) pins has continuously been growing for high-speed high-performance system. The eye diagram prediction is an important problem in the interconnection analysis because an eye diagram provides the most fundamental and intuitive view to evaluate the signal quality of high-speed communication. Inspired by these new challenges in the design and analysis of the interconnection, we propose new strategies for escape routing and develop new prediction method for the eye diagram analysis.

In chapter 2, we introduce a novel concept, escape sequence, which plays an important role in escape routing. The layer count for escape routing can be decreased by optimizing I/Os escape sequence in breakout process. For square grid array, we formulate a maximum flow problem to analyze the escape bottleneck in area array. Based on the

bottleneck analysis, we

present the escape sequence concept and summarize several guidelines for escape routing design. The central triangular escape and two-sided escape methods we suggest provide efficient ways on reducing the layer count. The escaping wires could be modeled as frequency dependent RLGC circuit for signal performance analysis.

Chapter 3 addresses the escape routing for the hexagonal area array. We analyze the properties of hexagonal area array. The hexagonal array could increase the density of I/Os in the array remarkably. We propose three escape routing strategies for the hexagonal array: column-by-column horizontal escape routing, two-sided horizontal/vertical escape routing, and multi-direction hybrid channel escape routing. Using same area and same pitch, hexagonal array holds about 15% more I/Os compared with square grid array and our strategies can escape the I/Os in the hexagonal array in the same or less number of routing layers. Therefore, we can reduce the number of escape routing layers as well as increase the density of I/Os.

In chapter 4, we introduce a new efficient and accurate method, the accumulative prediction method, to predict the signal distortion from inter-symbol interference. This method predicts the worst-case eye diagram based on step response of the signaling system. It extracts the eye opening and timing jitter and generates the input data patterns which produce the worst-case inter-symbol interference. The main advantage is that this general-purpose method can handle signals with asymmetric as well as symmetric rise/fall time. Furthermore, the complexity of the proposed method is linear O(n), where n is the number of time points of step response. As one application, it could be applied to analyze the signal performance of escaping wires.

5.2 Future Work

5.2.1 Escape Routing

In the escape routing problem, the objective in our research is to reduce the number of routing layers used for breaking out all I/Os in the area array which is mainly related to the cost of the systems. There are other performance issues that should be considered in real escape routing design. Furthermore, in real applications, I/Os in the area array are used for different signals and they may have various requirements for escape routing.

Signal Integrity The signal integrity is a measure of the quality of the electrical signal. In high-speed electronic systems, various effects could degrade the electrical signal performance to the point where errors occur. Signal integrity consideration is an important task at all levels of the system, from internal interconnections of an integrated circuit, through the package, the printed circuit board, the backplane, and inter-system interconnections. The main issues of concern are ringing, crosstalk, ground bounce, and power supply noise. The escape routing connects the I/Os to the next level assembly. Many signal integrity issues should be considered, noise induced by high density wires in escape routing area, crosstalk between wires and vias, the signal skew consideration and the impedance mismatch problem. The escape routing strategy with low cost and high performance is the ultimate goal for the practical electronic systems.

Different Types of I/O Signals I/Os communicate different types of signals, e.g. power supply, clock signal and differential signals. Different types of signals may have various requirements for escape routing. Most power supply I/Os are located inside the area array. They are usually directly connected to power plane by through hole via and no escape routing is needed in this case. However, there could be some power I/Os distributed among signal I/Os and they are needed to be connected to the outside with the signals as shielding. Differential signals involve two complementary signals sent on two separate wires. They should be routed

together and may have certain symmetry requirements. These special requirements should be considered in the escape routing design.

5.2.2 Eye Diagram Prediction

The accumulative method we present analyze the signal distortion from inter-symbol interference and predicts the worst-case eye diagram based on step response of the signaling system. Farther analysis of the signal and the applications of this method could be explored in the future. , such as facilitating circuit design using step response information, considering noise margin, and so on, could be explored in the future.

Noise and Cross-talk The cross-talk and echos as well as circuit related effects such as thermal noise, power supply noise and receiver jitter are also important factors for eye diagram analysis. To predict the eye diagram considering these factors would give more practical and meaningful evaluation of the signaling systems.

Interconnection Design The fast eye diagram prediction method could be applied to optimize the interconnection design. For a signaling system, the step response implicates certain characteristics of the system, e.g. the step response reveals the time-of-flight delay, the reflection and the saturation voltage for a transmission line communication. We could create the design guidelines from the step response and the predicted eye diagram.

References

[1] B. Analui, J. Buckwalter, and A. Hajimiri, "Data-dependent jitter in serial communications," *IEEE Trans. Microwave Theory Tech.*, Vol. 53, No. 11, pp. 1841-1844, Nov. 2005.

[2] M. Andrews and J. DeRosa, "Building a domestic high density flip chip organic substrate capability: ITRI/SEMATIC chip carrier project," MCM98 Proceedings, 1994, pp.695-699.

[3] H.B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, Reading, MA, 1990.

[4] G. Breed, "Analyzing signals using the eye diagram," High Frequency Electornics, Summit Technical Media, Nov. 2005.

[5] J. Buckwalter, B. Analui, and A. Hajimiri, "Data-dependent jitter and crosstalk-induced bounded uncorrelated jitter in copper interconnects," in IEEE MTT-S Int. Microwave Symp. Dig., vol.3, Jun. 2004, pp.1627-1630.

[6] J. Buckwalter, B. Analui, and A. Hajimiri, "Predicting data-dependent jitter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol.51, no.9, pp.453-457, Sep. 2004.

[7] B.K. Casper, G. Balamurugan, J.E. Jaussi, J. Kennedy and M. Mansuri, "Future microprocessor interfaces: analysis, design and optimization," in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 479-486, Sept. 2007.

[8] B.K. Casper, M. Haycock, and R. Mooney, "An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes," in *Proc. IEEE Symposium on VLSI Circuits*, pp. 54-57, 2002.

[9] W.T. Chan, F.Y.L. Chin, and H.F. Ting, "Escaping a grid by edge-disjoint paths," in Proc. 11th Annu. ACM-SIAM Symp. Discr. Algorithms, 2000, pp. 726-734.

[10] C.K. Cheng, J. Lillis, S. Lin, and N.H. Chang, Interconnect Analysis and Synthesis, Wiley-Interscience Publication, 2000.

[11] T.C. Chung, F. Ghahghahi, B. Oberlin, D. Carey, D. Nelson, "Area Array Packaging Technologies for High-Performance Computer Workstations and Multiprocessors," The proceedings of 46th Electronic Components and Technology Conference, 1996, pp.902-10.

[12] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, Introduction to Algorithms, Cambridge, MA: MIT Press, 1992.

[13] W.J. Dally and J.W. Poulton, *Digital systems engineering*, Cambridge University Press, 1998.

[14] J. Darnauer and W. Dai, “Fast pad redistribution from periphery-IO to area-IO,” in Proc. 1994 IEEE Multichip Module Conf., 1994, pp.38-43.

[15] M. Datta, T. Osaka, J.W. Schultze, “Microelectronic Packaging”, CRC press, 2005.

[16] P.P. Gelsinger, “Microprocessors for the new millennium: Challenges, opportunities, and new frontiers,” ISSCC 2001 Digest of Tech. Papers, pp.22-25, Feb. 2001.

[17] K.V. Guinn, R.C. Frye, “Flip-Chip and Chip-Scale I/O Density Requirements and Printed Wiring Board Capabilities,” 47th Electronic Components and Technology Conference Proceedings, 1997, pp.649-55.

[18] N.M. Gasparini, B.K. Bhattacharyya, “A Method of Designing a Group of Bumps for C4 Packages to Maximize the Number of Bumps and Minimize the Number of Package Layers,” 44th Electronic Components and Technology Conference Proceedings, 1994, pp.695-9.

[19] J. Hershberger and S. Suri, “Efficient breakout routing in printed circuit boards,” in Proc. 10th Annu. Symp. Comput. Geom., 1997, pp.460-462.

[20] E. Hirt and G. Troster, “Early footprint comparison for area I/O packages and first level interconnect,” 49th Electronic Components and Technology Conference Proceedings, 1999, pp.1210-1216.

[21] M. Horiuchi, E. Yoda, Y. Takeuchi, “Escape Routing Design to Reduce the Number of Layers in Area Array Packaging,” IEEE Transactions on Advanced Packaging, vol.23, no.4, Nov. 2000, pp.686-91.

[22] C.P. Hsu, “General River Routing Algorithm,” ACM IEEE 20th Design Automation Conference Proceedings, 1983, pp.578-83.

[23] T.C. Hu and M.T. Shing, “Combinatorial Algorihms,” Dover Publications, 2002.

[24] J.C.Y. Huang, K. Feher, and M. Gendron, “Techniques to generate ISI and jitter-free bandlimited Nyquist signals and a method to analyze jitter effects,” IEEE Trans. Commun., vol. COM-27, no.11, pp.1700-1711, Nov. 1979.

[25] G.A. Katopis, T. Zhou, “Applications of Wiring Escape Estimation,” Proceedings of 7th IEEE workshop on signal propagation on interconnects, 2003.

[26] D.B. Kuznetsov and J.E. Schutt-Aine, “Optimal transient simulation of transmission lines,” IEEE Trans. Circuits Syst., vol.43, no.2, pp.110-121, Feb. 1996.

[27] M.M. Ozdal and M.D.F. Wong, “Simultaneous escape routing and layer assignment for dense PCBs,” in Proc. IEEE/ACM ICCAD, Nov. 2004, pp.822-829.

[28] M.M. Ozdal, M.D.F. Wong, and P.S. Honsinger, “Simultaneous escape-routing algorithms for via minimization of high-speed boards,” IEEE Trans. on Computer-Aided Design of

Integrated Circuits and Systems, vol.27, no.1, Jan. 2008.

[29] M. Shimanouchi, "An approach to consistent jitter modeling for various aspects and measurement methods," in Proc. IEEE Int. Test Conf., Oct.-Nov. 2001, pp.848-857.

[30] V. Stojanovic and M. Horowitz, "Modeling and analysis of high-speed links," in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 589-594, Sept. 2003.

[31] A. Titus, B. Jaiswal, J. Dishongh and A.N. Cartwright, "Innovative Circuit Board Level Routing Designs for BGA Packages," IEEE Transactions on Advanced Packaging, vol.27, no.4, Nov. 2004, 630-639.

[32] A. Tsuchiya, M. Hashimoto, and H. Onodera, "Optimal termination of on-chip transmission-lines for high-speed signaling," *IEICE Trans. Electron.*, Vol. E90-C, No.6, pp. 1267-1273, June 2007.

[33] A. Tsuchiya, M. Hashimoto, and H. Onodera, "Performance limitation of on-chip global interconnects for high-speed signaling," IEICE Trans. Fundamentals, vol.E88-A, no.4, pp.885-891, April 2005.

[34] B. Voss, "BGA Escape Routing," printed circuit design, 2002, pp.34-35.

[35] R.D. Washburn and R.F. McClanahan, "Enhanced Timing margin memory interface," *United States Patent*, 2007.

[36] J. Wilstrup, "A method of serial data jitter analysis using one-shot time interval measurements," in Proc. IEEE Int. Test Conf., Oct. 1998, pp. 819-823.

[37] E. Winkler, "Escape Routing from Chip Scale Packages," 9th IEEE/CPMT International Electronics Manufacturing Technology Symposium, 1996, pp.393-401.

[38] M.F. Yu, W.M. Dai, "Single-Layer Fanout Routing and Routability Analysis for Ball Grid Arrays," IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers, 1995, pp.581-6.

[39] H. Zhu, C.K. Cheng, A. Deutsch, and G. Katopis, "Predicting and optimizing jitter and eye-opening based on bitonic step response," in *Proc. IEEE EPEP'2007*, pp. 155-158, Oct. 2007.

[40] IBM Corp., *PowerSPICE: User's Guide*, Version 2.0, Nov. 2005

[41] IBM Corp., "IBM electromagnetic field solver suite of tools," available at http://www.alphaworks.ibm.com/tech/eip

[42] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 2003 edition.