UNIVERSITY OF CALIFORNIA, SAN DIEGO

On-Chip Interconnect Architectures: Perspectives of Layout, Circuits, and Systems

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

by

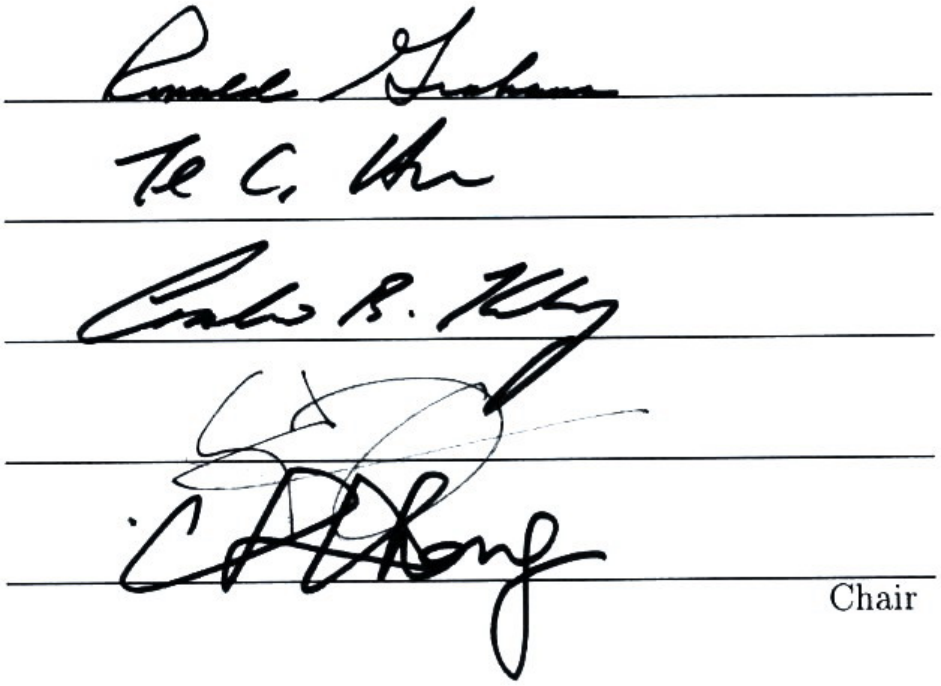
Hongyu Chen

Committee in charge:

Professor Chung-Kuan Cheng, Chair
Professor Ronald Graham
Professor Te C. Hu
Professor Andrew B. Kahng
Professor Gabriel Rebeiz

2006

The dissertation of Hongyu Chen is approved, and it is acceptable in quality and form for publication on microfilm:



Chair

University of California, San Diego

2006

To the memory of my father

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

VITA

2000	B.S. in Computer Science and Technology Tsinghua University, Beijing, China
2002	M.S. in Computer Science University of California, San Diego
2000–2005	Research Assistant, University of California, San Diego
2006	PhD in Computer Science and Engineering University of California, San Diego

PUBLICATIONS

B. Yao, H. Chen, C. K. Cheng, and R. Graham, "Floorplan Representations: Complexities and Connections," in *ACM Trans. on Design Automation of Electronic Systems,* vol.8(1), pp.55-80, February, 2003.

H. Chen, C. K. Cheng, A. B. Kahng, I. Măndoiu, Q. Wang, and B. Yao, "The Y-Architecture for On-Chip Interconnect: Evaluations and Methodologies," accepted for publication by *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems.*

H. Chen, M. Mori, B. Yao, and C. K. Cheng, "A Multi-level Network Approach for Clock Skew Minimization Under Process Variations," submitted to *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems.*

H. Chen, C. Y. Yeh, S. M. Reddy, G. Wilke, H. V. Nguyen, and W. W. Walker, "A Sliding Window Scheme for Accurate Clock Mesh Analysis," to appear in *International Conference on Computer Aided Design* 2005.

H. Chen, R. Shi, C. K. Cheng, and D. M. Harris, "Distortionless Electrical Signaling for Speed of Light On-Chip Communications," to in *International Conference on Computer Design*, 2005.

Y. Hu, H. Chen, Y. Zhu, A. Chien, and C. K. Cheng, "Physical Synthesis of Energy Efficient Network-On-Chip Through Topology Exploration and Wire Style Optimization," to appear in *International Conference on Computer Design* 2005.

B. Yao, H. Chen, C. K. Cheng, N. C. Chou, L. T. Liu, and P. Suaris, "Unified Quadratic Programming Approach for Mixed Mode Placement," in *International Symposium on Physical Design* 2005.

H. Chen and C. K. Cheng, "A Multi-Level Transmission Line Network Approach for Multi-Giga Hertz Clock Distribution," in *Asia and South Pacific Design Automation Conference*, January 2005.

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

M. Mori, H. Chen, B. Yao, and C. K. Cheng, "A Multi-Level Network Approach for Clock Skew Minimization Under Process Variations," in *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 263-268, January 2004.

H. Chen, C. K. Cheng, A. B. Kahng, and Q. Wang, "Optimal Planning for Mesh-Based Power Distribution," in *Proceedings of Asia and South Pacific Design Automation Conference,* pp. 444-449, January 2004.

H. Chen, C. K. Cheng, A.B. Kahng, Ion Măndoiu, Q. Wang, and B. Yao, "The Y-Architecture for On-Chip Interconnect: Analysis and Methodologies," in *Proceedings of International Conference on Computer Aided Design,* pp. 13-19, November 2003.

H. Chen, C. K. Cheng, N. C. Chou, A. B. Kahng, J. MacDonald, B. Yao, and Z. Zhu, "An Algebraic Multigrid Solver for Analytical Placement with Layout Based Clustering," in *Proceedings of Design Automation Conference*, pp. 794-799, June, 2003.

H. Chen, C. K. Cheng, A. B. Kahng, I. Măndoiu, and Q. Wang, "Estimation of Wirelength Reduction for λ-Geometry vs. Manhattan Placement and Routing," in *Proceedings of ACM/IEEE International Workshop on System Level Interconnect Prediction*, pp. 71-76, April, 2003.

H. Chen, B. Yao, F. Zhou, and C. K. Cheng, "Y-architecture: Yet Another On-Chip Interconnect Solution," in *Proceedings of Asia and South Pacific Design Automation Conference,* pp.840-846, January, 2003.

H. Chen, B. Yao, F. Zhou, and C. K. Cheng, "Physical Planning of On-Chip Interconnect Architectures," in *Proceedings of IEEE International Conference on Computer Designs,* pp.30-35, September 2002.

H. Chen, C. Qiao, F. Zhou, and C. K. Cheng, "Refined Single Trunk Tree: A Rectilinear Steiner Tree Generator for Interconnect Prediction," in *Proceedings of ACM/IEEE International Workshop on System Level Interconnect Prediction*, pp.85-89, April, 2002.

B. Yao, H. Chen, C. K. Cheng, and R. Graham, "Revisiting Floorplan Representations," in *Proceedings of ACM International Symposium on Physical Design,* pp.138-143, April, 2001.

FIELDS OF STUDY

Major Field: Computer Science and Engineering

Studies in Computer Aided Design of Very Large Integrated Circuits.
Professor Chung-Kuan Cheng

ABSTRACT OF THE DISSERTATION

On-Chip Interconnect Architectures: Perspectives of
Layout, Circuits, and Systems

by

Hongyu Chen
Doctor of Philosophy in Computer Science
University of California, San Diego, 2006
Professor Chung-Kuan Cheng, Chair

With exponentially increasing integration densities and shrinking characteristic geometries on a chip, the wires, rather than devices, become the dominant factor in deciding the performance, power consumption, and reliabilities of VLSI systems. Previous researches on interconnect centric design methodologies mainly concentrate on optimizing individual nets. Instead of searching for the best algorithm to optimize each individual net, we take a view of the on-chip interconnection architectures, and improve the system performance by considering both geometrical arrangements of wires, electrical behaviors of global distribution networks, as well as adopting innovative interconnect circuit styles.

Traditional Manhattan routing restricts the wires on horizontal and vertical tracks. This artificial restriction causes excessive wirelength overhead over the Euclidean optimum and thus decreases the efficiency of the interconnect system. We investigated the optimal way to utilize the on-chip routing layers through non-Manhattan routing. We adopted multi-commodity flow models to measure the throughput of different on-chip interconnect architectures. Through careful analysis of the bottlenecks of the on-chip communication traffic, we found that the Y-architecture (3-directional routing) enjoys a lot of nice properties over other

routing architectures. We developed a design methodology for Y-architecture, including power and clock distribution and a novel way to hide the via blockage effect.

Clock distribution network is one of the most important interconnect on a chip. We studied the high speed clock distribution in the presence of parameter variations. We proposed a spectrum of solutions for circuits working at different frequencies: A variations aware clock tree synthesis algorithm for high-end ASICs, a multi-level mesh approach for microprocessors, and a transmission line network approach for future multi-giga hertz chips. Simulation results suggest that these approaches significantly improve clock distribution networks' resilience against process, voltage and temperature variations.

We proposed a novel scheme to implement distortionless transmission lines for on-chip electrical signaling. By introducing intentional leakage conductance between the wires of a differential pair, the distortionless transmission line eliminates dispersion caused by the resistive nature of on-chip wires and achieves speed of light transmission. We show that it is feasible to construct distortionless transmission line with conventional silicon process. Simulation results show significant improvements in both speed and power consumption over conventional RC wires with repeated buffers.

I

Introduction

I.A On-Chip Interconnect: Challenges and Opportunities

With the advance of the processing technology, interconnect is becoming the dominant factor in deciding the power consumption, performance, reliability, and cost of VLSI systems. This fact has been widely recognized in both academia and industry. As evidenced in the SRC Roadmap, the scalability of interconnect is remarked as a "principal challenge" to the continua of Moore's Law.

Interconnect power consumption grows with each technology node generation. The dynamic power is proportional to fCV^2, where f is the frequency of data activity, C is the total capacitance, and V is the supply voltage. As design rules shrink, frequency f increases. The total capacitance C per unit area also increases since we have more, finer pitch wires in the same area while the wire capacitance per unit length remains fairly constant [73, 74]. However, the supply voltage V does not scale much (Table 47a of [90]) due to the limit of threshold voltage. Consequently, the dynamic interconnect power increases. In [76], Magen, et al. at Intel found that interconnect power alone accounted for half the total dynamic power in a $0.13\mu m$ microprocessor that was designed for power efficiency.

At the technology nodes beyond 65nm, the leakage power constitutes a significant portion of total power consumption. In order to achieve the delay

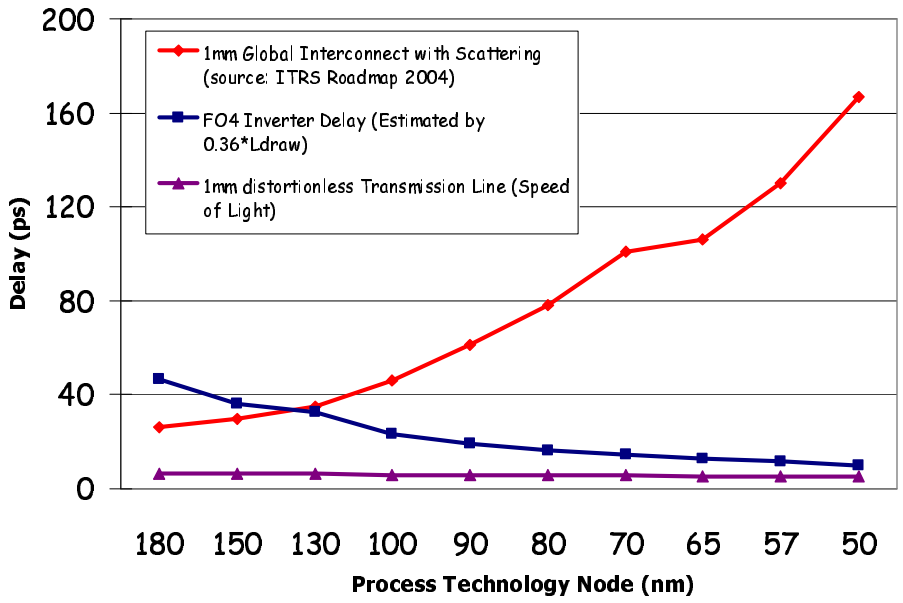


Figure I.1: Example of 45 degree mesh of $n = 4$

and slew rate specification, traditional RC wires require a substantial number of repeaters inserted. According to the estimation of [87], at 45nm node, the repeaters can account to over 30% of overall chip area in a high performance microprocessors. These repeaters consume a great amount of static power due to leakage current. As a result, interconnect power consumption can potentially be a limiting factor of Moore's law.

Interconnect delay dominans the system performance. In Figure I.1, we depict the gate and interconnect delay scaling trends according to the SRC roadmap. The delay of a 1mm Cu global interconnect at minimum pitch increases from 46ps at the 100nm node to 167ps at the 50nm node, assuming added resistance due to width dependent scattering and a conformal barrier of thickness (Table 81a of [71]). The fanout-of-4 (FO4) inverter delay is estimated using the equation $0.36ns/\mu m \times L_{drawn}$ proposed by Horowitz [75], where L_{drawn} is the printed channel length for the given technology. The curve at the bottom describes a transmission line carrying the differential signal as an electromagnetic wave. For an ideal transmission line, the wave propagates at the speed of light, which reduces from

5ps/mm at the 100nm node to 4ps/mm at 50nm node due to the reduction of effective dielectric constant. Since the speed of light is the physical limit of the given technologies, one of our ultimate goals is to achieve this speed.

In Ultra-Deep-Sub-Micron processes, process variations presents great challenges to the yield and reliability of a design. One of the most vulnerable components subject to the process and other parameters variations is the clock distribution network. As the clock frequency continuously getting higher, the clock uncertainty caused by the variations consumes increasing portion of clock period time. At the same time, clock distribution is one major consumer of the dynamic power. For high speed circuit, the clock

Facing all the challenges discussed above, it is imperative to identify solutions which address global wiring scaling issues. In this thesis, we optimized the on-chip interconnect architectures from different perspectives, including, optimal geometrical arrangement through non-Manhattan routing, variations aware clock distribution architectures, and an innovative on-chip signaling scheme.

I.B Previous Works

This section gives an overview of the previous works on the three problems we are studied.

I.B.1 Non-Manhattan Routing Architectures

With rapid technology scaling, the interconnect becomes one most precious resource on a chip. Traditional Manhattan interconnect architecture organizes wires on two orthogonal routing directions, 0-degree and 90-degree directions, for the simplicity of routing embedding and design rule checking. However, its artificial restriction on routing directions adds significant wire length over the Euclidean optimum and thus decreases the communication capability of the on-chip interconnects.

In the past decade, many researchers have explored the possibility of using non-rectilinear wires to improve the efficiency of on-chip interconnects.[4] Most of these work discussed about how to introduce 45-degree short jogs to improve the routability of the chip in the detailed routing stage. Majority of the wires on the chip are still routed on either 0-degree or 90-degree direction.

Recently, Teig et. al. [7] proposed a new on-chip interconnect architecture named the X-architecture, which is targeting at the designs with 5 or more routing layers. In the X-architecture, the wires are organized in 0-degree, 45-degree, 90-degree and 135-degree directions. The experimental results from the first industrial design [91] show that it achieves a chip performance improvement of 10% and power reduction of 20% than Manhattan architecture for a high performance design.

I.B.2 High-Speed Clock Distribution

Clock distribution has been widely studied in the past twenty years. A wealth of literature can be found on both CAD algorithms and clock distribution architectures.

The median and mean method (MMM) proposed by Jackson et *al.* [11] recursively partition the clock sinks on a plane according to the locations. This method produces a well balanced topology. However, it does not consider routing obstacles.

The geometric matching [12] and greedy-DME [6], uses bottom -up matching to construct the tree topology. These methods depend on the dynamic nearest neighbor queries, which is computationally expensive in the presence of routing obstacles. In a Manhattan plane without obstacles, the shortest path query only takes O(1) time, while with obstacles, the fastest algorithms have a time complexity of at least $O(m \log m)$, where m is the number of corner nodes of all obstacles.

Ellis et *al.*[7] and Chou et *al.* [5] both used simulated annealing to search for the optimal tree topology. Multiple objectives are optimized simultaneously in the simulated annealing framework.

Recently, several works target at the variations aware clock tree synthesis. Velenis et *al.* [21] first noticed that not all datapath have the same sensitivity to the clock uncertainty, and use a sequential merging scheme to construct the clock tree topology. Their method does not use any physical proximity information. It may results in excessively large total wire length, and the tree topology may be very unbalanced.

In [10], Hu et *al.* extended the DME [2] algorithm to accommodate the permissible clock uncertainty constraints for a given clock tree topology. This method can significantly reduce the timing violations caused by the process variations on the interconnect. However, since the tree topology is generated by the non-variations-aware DME algorithm, it could not reduce the clock uncertainties caused by the voltage variations on the clock buffers, which is believed to be one main cause of clock uncertainties. In [3], the clock sinks are partitioned into groups, the algorithm reduces the wirelength by only minimizing intra-group skew values.

Clock mesh has been applied to high performance microprocessors to achieve low skew distribution, one exemplary design is DEC Alpha Series [91,92,93]. A recent trend is to use the hybrid structure of a mesh and a symmetric buffer tree for the global clock network. For example, the global clock distribution of Intel Pentium4^{TR} microprocessor consists three spines each driven by a balanced binary buffer tree [1]. The bottom-level spines can be deemed as a "one dimensional" mesh structure. Restel et *al.* proposed a two-level hybrid clock network. The top level is an H-tree, and the bottom level is a uniform mesh that connects all of the leaves of the top level H-tree. This clock network structure has been successfully applied to six designs[1], including the latest Power4 microprocessors [9][14]. The measurements from the real produced chips proved that this hybrid tree and mesh structure accomplishes low clock skew under process variations.

One drawback of the dense mesh for clock distribution is its high power consumption. A transmission line network with a properly tailored length can achieve low power low skew distribution. Galton et *al.* [7] showed that when

the wire length is shorter than one quarter wavelength, the transmission line can synchronize oscillators both in phase and magnitude. Several other researches utilized the synchronization capability of the transmission line in clock distribution by connecting the distributed PLLs together with transmission lines [8,9,20]. In order to compensate the lossy nature of the on-chip transmission lines, [14] used distributed transconductors along the transmission line to generate the standing wave.

I.B.3 High Performance On-Chip Interconnects

Buffer insertion has been used to improve the speed of on-chip interconnect in deep sub-micron technologies. [24] gives a complete survey of the buffer insertion techniques.

In [25], pre-emphasizing and de-emphasizing along with data aliasing are used to modulate the input wave form. In [20], Afshari and Hajimiri adopted a non-linear transmission line approach to generate solitary wave propagation and thus compensate for the dispersion. In [23], a high frequency carrier modulates the input waveform and shifts the spectrum of transmitted signal to a less frequency sensitive region. In [27], a clocked discharging scheme is adopted to erase the data dependant delay variations. In [30], an adaptive equalization scheme is used to compensate the propagation loss.

I.C Our Approaches and Contributions

Our approaches to solve the three problems in the physical planning are summarized in this section. Our contributions are highlighted.

I.C.1 Non-Manhattan Routing Architectures

We developed mathematical models to measure the throughput of different on-chip interconnect architectures. Through careful analysis of the bot-

tlenecks of the on-chip communication traffic, we found that the Y-architecture (3-directional routing) enjoys a lot of nice properties over other routing architectures. We develop a design methodology for Y-architecture, including power and clock distribution and a novel way to hide the via blockage effect.

Our contributions include:

1. A multi-commodity flow (MCF) model to evaluate the throughput of different routing architectures and a combinatorial approximation scheme to find the optimal routing resource allocation under the MCF model
2. Y-Architecture and its associated design methodologies
3. Via tunnels and via tunnel banks to minimize the via blockage effects

I.C.2 High-Speed Clock Distribution

We studied the high-speed clock distribution in the presence of process, voltage, and temperature (PVT) variations. Our research addresses a spectrum of needs for variation tolerant clock distribution networks, including, a PVT variations aware clock tree synthesis algorithm, a multi-level mesh architecture, and a transmission line network architecture.

Process, voltage, and temperature variations aware clock tree synthesis in the presence of routing obstacles

Clock tree synthesis has been a classic problem in the CAD community because of its clean mathematical formulation. However, in industry practices, there are various engineering challenges. Two of them are the existence of routing obstacles and the requirement to minimizing clock uncertainties introduced by parameter variations. In this study, we minimize the clock tree wire length in the presence of routing obstacles and clock uncertainty constraints.

We extend the Delaunay triangular mesh to represent the physical proximity of clock sinks in the presence of rectilinear routing obstacles. Combining

the Delaunay triangular mesh and clock uncertainty constraint graph, we adopt graph partitioning technique to generate a clock tree topology, which balances the wirelength and timing. We further use a multilevel algorithm to optimize the tree topology and embedding.

Experimental results show significant improvement on PVT variations tolerances with very minor wire length overhead.

Multi-level mesh for clock skew minimization in the presence of process variations

In high performance systems, mesh structures has been used for low skew clock distribution. However, there is no design guideline on choosing the optimal mesh granularity and wire sizes. We investigate the skew reduction effect of shunt connections and seek for the best mesh structures for given total routing area.

We derive an analytical expression of the skew reduction effect of resistive shunts. Based on this theoretical study, we propose a multi-level mesh structure for process variations tolerant clock distribution. We adopt a convex programming approach to optimize the proposed clock network. Simulation results demonstrate more than 70% reduction on the clock skew comparing with widely adopted single level mesh structure.

Transmission line network for multi-giga hertz clock distributions

With the clock frequency thrusting into multi-giga hertz range, the shunt effect of RC wire diminishes because of the inductive effect. On the other hand, the transmission line has its unique property to lock the oscillators together. We study how to construct an extreme low skew clock distribution network which tailors the natural frequency of transmission lines.

We design a multiple level transmission line network which can work in the multiple giga-hertz range. We derive the analytical expression of phase locking capabilities of on-chip transmission lines. Based on the this expression, we use

mathematical programming technique to optimize the network. Simulation results suggest great improvement of variations tolerance and power consumption.

I.C.3 High Performance On-Chip Interconnects

We devised a novel scheme to implement distortionless transmission lines for on-chip electrical signaling. By introducing intentional leakage conductance between the wires of a differential pair, the distortionless transmission line eliminates dispersion caused by the resistive nature of on-chip wires and achieves speed of light transmission. We show that it is feasible to construct distortionless transmission line with conventional silicon process. Simulation results show that using 65nm technology, the proposed scheme can achieve 15Gbits/s bandwidth over a 20mm on-chip serial link without any equalization. This approach offers a six times improvement in delay and 85% reduction in power consumption over a conventional RC wire with repeated buffers.

I.D Organization of the Dissertation

The following chapters of the dissertation are organized as follows:

Chapter II discusses Non-Manhattan Routing Architectures. The chapter has two parts. Section II.A presented our research on the physical planning of on-chip interconnect architectures. After that, we discussed the Y-architecture and associated design methodologies in Section II.B.

Chapter III is on the high-speed clock distributions. In includes three parts. In Section III.A, a PVT variations aware clock tree algorithm is presented. In Section III.B, a multi-level mesh architecture is proposed. In Section III.C, a transmission line network approach is shown.

Chapter IV describes a distortionless transmission line approach for high-speed, low-power on-chip communications.

Chapter V draws the conclusions. Several possible directions on extending

the works in this dissertation are also given.

II

Non-Manhattan Routing Architectures

In this chapter, we discuss the use of Non-Manhattan routing architectures to improve the efficiency of on-chip interconnects. We studied both the physical planning of the interconnect architectures and the design methodologies associated with the Y-architecture (3-directional routing).

II.A Physical Planning of Interconnect Architectures

Interconnect architecture plays an important role in determining the throughput of meshed communication structures. We assume a mesh structure with uniform communication demand for communication. A multi-commodity flow (MCF) model is proposed to find the throughput for several different routing architectures. The experimental results reveal several trends: 1. The throughput is limited by the capacity of the middle row and column in the mesh, simply enlarging the congested channel cannot produce better throughput. A flexible chip shape provides around 30% throughput improvement over a square chip of equal area. 2. A 45-degree mesh allows 17% throughput improvement over 90-degree mesh and a 90-degree and 45-degree mixed mesh provides 30% throughput improvement. 3. To achieve maximum throughput on a mixed Manhattan and diagonal intercon-

nect architecture, the best ratio of the capacity for diagonal routing layers and the capacity for Manhattan routing layers is 5.6. 4.Incorporating a simplified via model, interleaving diagonal routing layers and Manhattan routing layer is the best way to organize the wiring directions on different layers.

II.A.1 Introduction

Mesh is a common routing architecture for many reconfigurable computing systems. Both conventional FPGAs[12] and recently proposed on-chip multiprocessors systems[7] [6] use mesh networks as communication backbones. With rapid technology scaling, wires become one most precious resource on a chip. Unreasonable distribution of wire resources will result in bottlenecks that stall the data flows, meanwhile leave other routing resources wasted. Simply enlarging the channel capacity of the whole array is by no means an effective solution.

Our goal is to allocate channel capacities in the mesh routing architecture to maximize its communication capability. The communication capability is measured by the throughput, the amount of information that every pair of nodes can exchange simultaneously. Throughput is a function of channel capacity and the dimension of the processor array.

Khalid and Betz investigated the channel allocation problems for FPGAs in [5] [2]. They applied placement and routing to benchmark circuits on FPGAs with different routing track distributions. They conclude that uneven track distribution do not improve the routability of FPGA interconnects.

Multi-commodity flow (MCF) is a natural way to model the communication network traffic. Many previous works used MCF in studying the wide area communication network traffic [10]. Due to the high computing complexity of MCF, most of these works [10] adopted heuristic methods to approximate the MCF solution.

Recent advance in MCF algorithm [4] allows us to compute MCF more efficiently. In this paper, we choose MCF to model communication traffic. We

extend the MCF algorithm in [4] to solve various MCF problems. Solution of MCF finds the optimal throughput for a given routing architecture. In our MCF model, the routing demand is equal for every pair of nodes. Thus, the result is independent of test cases. Moreover, the result of MCF is independent of placement and routing.

In [9], Mutsunori et al. demonstrated that on-chip diagonal routing is feasible based on state-of-the-art technologies. The progress of diagonal routing technology provides another opportunity to explore different arrangements of interconnect structure. We compare the throughput of three different mesh structures, the 90-degree mesh, the 45-degree mesh, and the 90-degree and 45-degree mixed mesh. Experimental results show that a 45-degree mesh can achieve better throughput than a 90-degree mesh. Moreover, 90-degree and 45-degree mixed mesh can further improve the throughput. Mixed 90-degree and 45-degree mesh allows more freedom on routing directions. We explore the allocation of routing resources and the arrangement of routing resources of routing directions between 90-degree and 45-degree wires. We propose a simplified via model to derive the optimal solution.

Our contributions include:

- We use MCF model to analyze the detailed communication traffic on mesh interconnect networks. We find the exact traffic bottlenecks of the network and the throughput of communications in the mesh structure. This provides a feasible upper bound of communication.

- We extend the flow approach [4] to compute the optimal routing resource allocation for mesh interconnect structures with reasonable sizes. The results reveal some basic trends of throughput related to the scale and structure of the communication mesh:

  - For uniform capacity mesh, the congested edges lie in the center rows and columns. The total throughput of each node is inversely propor-

tional to the dimension of the mesh

- The re-arrangement of capacities between different columns or rows will not improve the throughput if we keep the total capacity of the columns or rows a constant.
- A flexible chip shape provides a throughput improvement of around 30% over a square chip of equal area.
- A 45-degree mesh structure produces a 17% more throughput than a 90-degree mesh for a processor array of 144 nodes.
- A mixture of 90-degree and 45-degree mesh structure can achieve even a 30% more throughput. To achieve maximum throughput, the ratio of resources allocated to the 45-degree routing layers verses those to the 90-degree routing layers approaches 5.6 as number of nodes increases.
- In the 90-degree and 45-degree mixed routing, interleaving the diagonal routing layer and Manhattan routing layers can reduce the number of vias and hence increase the communication throughput.

The rest of this paper is organized as follows: Section 2 presents the problem formulation in MCF model. Section 3 introduces six different interconnect structures we consider. Section 4 gives the experimental results and our observations. We draw conclusion in section 5.

II.A.2 Problem Formulation

We decompose the communication resources into an array of n n slots. Each slot contains a communication terminal, say, a processor. The slots are aligned in rows and columns. The slot array forms a 90-degree mesh structure. Figure 1(a) illustrates an example of a 90-degree mesh structure with 25 slots. Each square tile represents a slot. The mesh structure can be mapped to a graph $G = \{V, E\}$ according the following rules:

1. Each slot corresponds to a node in the graph.

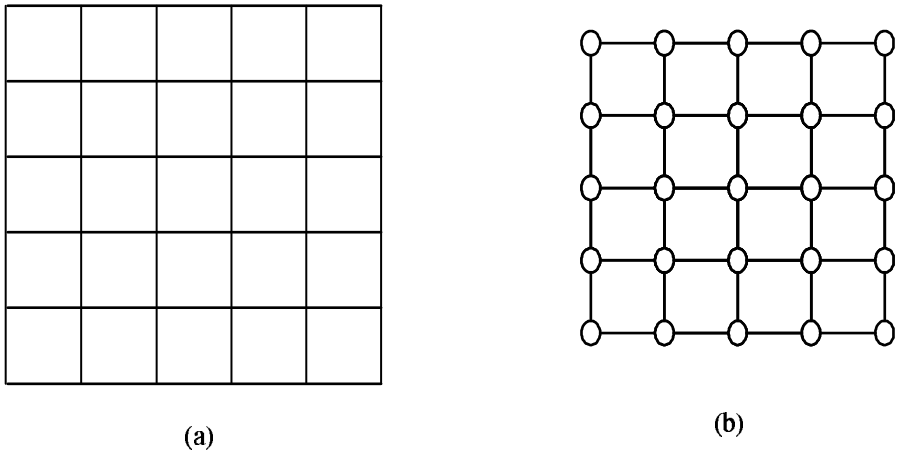


Figure II.1: A 5 by 5 communication mesh and its graph representation

2. The adjacency between two slots is represented by an edge connecting two corresponding nodes.

3. The edge capacity is proportional to the length of the line segment separating the adjacent slots, and the number of routing layers.

Figure II.1(b) describes the graph corresponding to the mesh in Fig.II.1(a).

We assume a uniform communication requirement, *i.e.* every pair of nodes communicate with an equal demand. All communications happen at the same time. Note that the model can be extended to various communication demands, *e.g.*, Poisson distribution, Rents rule, *etc.*, depending on specific applications. In this paper, the uniform pairwise communication model is adopted because of its simplicity and genericalness. Moreover, the communication demand represents an unbiased symmetry, which makes the solution independent of the test cases, placement, and routing.

We define the throughput, z, to be the maximum amount of communication flow between every pair of nodes. We try to find the throughput using a multi-commodity flow model. The flow that starts from node i is deemed as commodity i. Commodity i starts from node i with the amount of $z \cdot (N-1)$, where $N = n^2$ is the number of nodes in the graph, to each of the rest nodes

with the amount of z. We solve the multi-commodity flow problem to find the maximum value of z. We can use following linear program to express the above MCF problem:

$$Max: \quad z \tag{II.1}$$

$$S.t. \quad \text{For each commodity } v \text{ , on each node } i$$

$$\sum_{j \in \text{neighbor of } i} (f_{ji}^v - f_{ij}^v) = a \tag{II.2}$$

$$\text{for each edge } (i,j) \text{ in the graph,}$$

$$\sum_{v=1}^{n^2} (f_{ji}^v - f_{ij}^v) \leq c_i j \tag{II.3}$$

In this linear program, flow variable f_{ij}^v represents the flow amount of commodity v on edge (i,j). The edge capacity c_{ij} represents the flow capacity of edge (i,j). We set that the flow injecting to a node is positive and the flow ejecting from a node is negative.

The linear program includes two sets of constraints. Constraint (1) describes the flow conservation of each commodity v at each node i. Constraint (2) denotes that the total amount of flow on each edge is no more than the capacity of that edge.

In subsections 3.3, 3.4 and 3.5, we allow edge capacity to be changed. Thus, the edge capacities become variables in the linear constraints. Thus, allows us to optimize the capacities under the area constraints.

In [4], a fast combinatorial $(1+\epsilon)$-approximation algorithm was introduced to solve the MCF problem. We extend the approach to incorporate edge capacities as variables.

According to the algorithm in [4], we adopt the primal-dual structure of the linear program. The algorithm assigns a nonnegative shadow cost [12] to each edge according to the congestion level on that edge. Initially, all the shadow costs are set to be equal. Then, the algorithm proceeds in iterations. In each iteration,

we reroute a fixed amount of flow along the shortest path for every commodity. At the end of each iteration, we adjust the capacity of every edge and its shadow cost according to the dual linear program.

In our model, we all fractional flows. Note that the throughput, Z_f , of the fractional flow model, is an upper bound of the throughput,Z_i , of the integer flow model [3][1]. In [8], Motwani and Raghavan showed that by randomized rounding, with the probability of $1-\epsilon$, we can find Z_i approaches Z_f with inequality $Z_f \geq Z_i/1 + \Delta^+(1/Z_f, \epsilon/2N)$, where N is the number of nodes in the mesh, ϵ is any real number between 0 and 1, and $\Delta^+(1/Z_f, \epsilon/2N)$ is the value of *delta* such that $[e^\delta/(1+\delta)^{(1+\delta)}]^{\frac{1}{2}} = \epsilon/2N$.

II.A.3 Routing Architectures

We construct six routing architectures with different capacities and routing orientations. The first three structures are 90-degree meshes with different edge capacities. In the first architecture, every edge has a unit capacity. In the second architecture, edges on the same row or column have equal capacity. In the third architecture, edge capacities are flexible but the sum of the capacities of all the edges is fixed. The fourth architecture is a 45-degree mesh where interconnection goes in 45 degree. The fifth is a mixture of 90-degree and 45-degree mesh. And the last one is the mixed 90-degree and 45-degree mesh with different routing direction assignments.

Uniform Edge Capacity

For the model of uniform edge capacity, All the edge capacity is set to a unit, i.e. $c_{i,j} = 1$ for all edges (i, j) in the graph. This case is the basis of our experiments. We assume that the $n \times n$ array of slots is evenly distributed in a square area.

[1]For packet switching network in RAW and Smart Memories, we do not require the flow to be integer. For wire switching network in FPGAs, the flow amounts can be interpreted as the number of wires, which needs to be integers.

Uniform row and column capacity

In the second structure for interconnection, edge capacities ce are set as variables. However, the capacities of edges in the same row are set to be equal. Likewise, the vertical capacities of edges in the same column are set to be equal. The sum of the vertical edge capacities in a row is set to be n, and the sum of the horizontal edge capacities in a column is also set to be n. In other words, we assume that the height and the width of the array remain to be n.

Let c_{Hi} be the capacity of horizontal edges in the i-th row, and c_{Vi} be the capacity of vertical edges in the i-th column. We add the $2n$ variables, $c_{H1}, c_{H2}, \ldots, c_{Hn}, c_{V1}, c_{V2}, c_{Vn}$, to the linear program. The height and width constraints of the array can be expressed as:

$\sum_{i=1}^{n} C_{Hi} = n$ and $\sum_{i=1}^{n} C_{Vi} = n$.

For this structure, we assume that we can adjust the row height and the column width of the array of processors.

Fixed total edge capacity

For the third structure we give the program more freedom to choose the best edge capacities. We require only that the total capacity of all edges to be a constant. This structure represents the best edge capacity we can allocate for a 90-degree mesh. The resultant throughput is an upper bound of a 90-degree mesh architecture.

We set the edge capacities, c_{ij}, as variables. The total capacity constraint is expressed as: $\sum_{\text{for all edges } (i,j)} c_{ij} = 2(n^2 - n)$.

Note that $2(n^2 - n)$ is the number of edges in an n n mesh.

For this structure, we assume that the area of each slot is flexible. We adjust the height and width of each individual slot so that the total area remains the same.

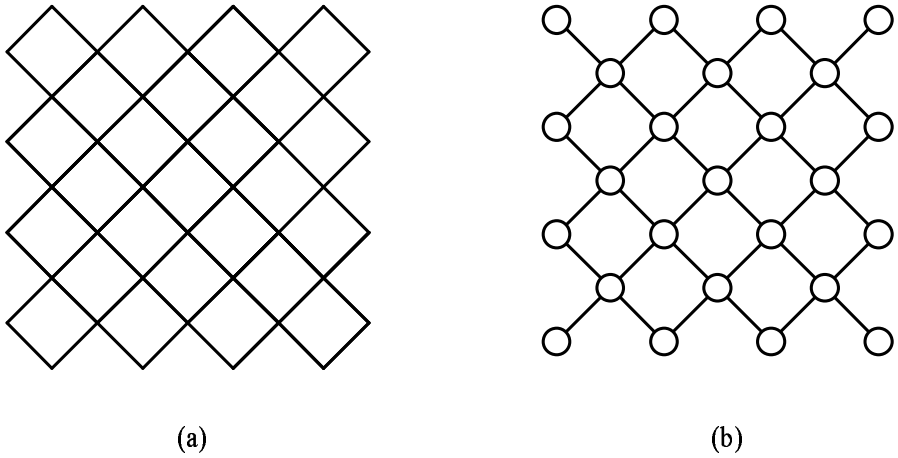


Figure II.2: Example of 45 degree mesh of $n = 4$

45-degree mesh

The fourth structure adopts a 45-degree mesh. All wires are oriented in 45 degree or 135 degree. The size of the mesh increases with n. For a 45-degree mesh of n, the number of nodes is $n^2+(n-1)^2$ and the number of edges is $4(n-1)^2$.

Figure II.2(a) shows an example of 45-degree mesh of $n = 5$. Figure II.2(b) illustrates the corresponding graph to the mesh. In this structure, we assume that the slots are shaped in diamonds (a square rotated by 45-degree) and are aligned in 45-degree and 135-degree directions. Thus, the edge capacity remains to be a unit, i.e. $c_{ij} = 1$.

90-degree and 45-degree mixed mesh

In the fifth structure, we add 45-degree channels to the 90-degree mesh. Figure II.3 illustrates an example of the mixed mesh for $n = 5$. Figure II.3(a) shows the slots arrangement. For an n by n mixed mesh, the number of nodes is n^2 and the number of edges is $2(n-1)^2 + 2(n^2-n)$.

In Figure II.3(b), the edges are oriented in 0, 90, 45 or 135-degree angle. All nodes are aligned in rows and columns. Thus the channels for 45- and 135-degree wires are scaled by . In other words, for a pair of routing layers, if we can

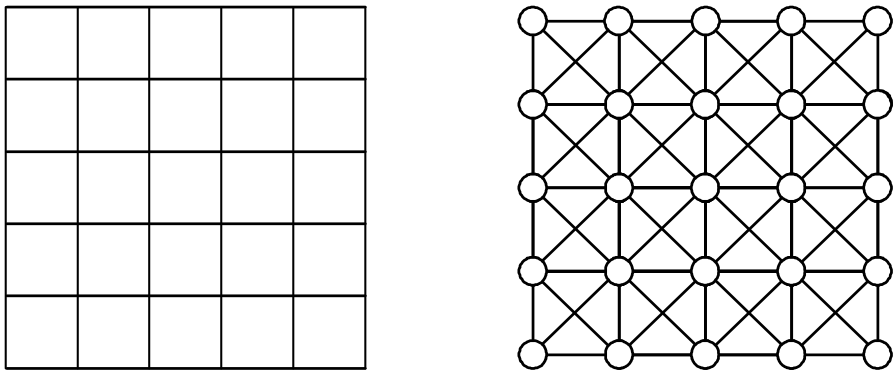


Figure II.3: A 90-degree and 45-degree mixed mesh of $n = 5$

allocate a capacity of x to 0- and 90-degree edges, we can only allocate a capacity of $x/\sqrt{2}$ to 45- and 135- edges. Let c_1 be the capacity of horizontal and vertical edges, c_2 be the capacity of diagonal edges. The area constraints can be expressed as $c_1 + \sqrt{2}c_2 = 1$. Thus, the total area is equal to the area of uniform structure.

Routing direction Assignment

Vias become an important concern when number of routing layers increases. In [1] a global routing graph with via edges is used to model the multi-player routing. In that model vias are not considered as routing blockages.

We propose a network flow model shown in Fig. II.4 to take vias into. Our basic assumption is that each via will block one routing track. For each slot, we set an upper bound on the total number of vias and wires across the node.

Suppose there are k routing layers. Each slot is now represented by k routing cells (Fig. II.4(a)). Each routing cell consists of two nodes (Fig. II.4(b)): n_a and n_b. Node n_a takes all the incoming edges from the neighboring routing cells, and node n_b ejects edges to neighboring routing cells. An edge with capacity c direct from node n_a to node n_b. This edge is used to restrict the total number of vias and wires crossing the routing cell. Using this flow model, we compare the

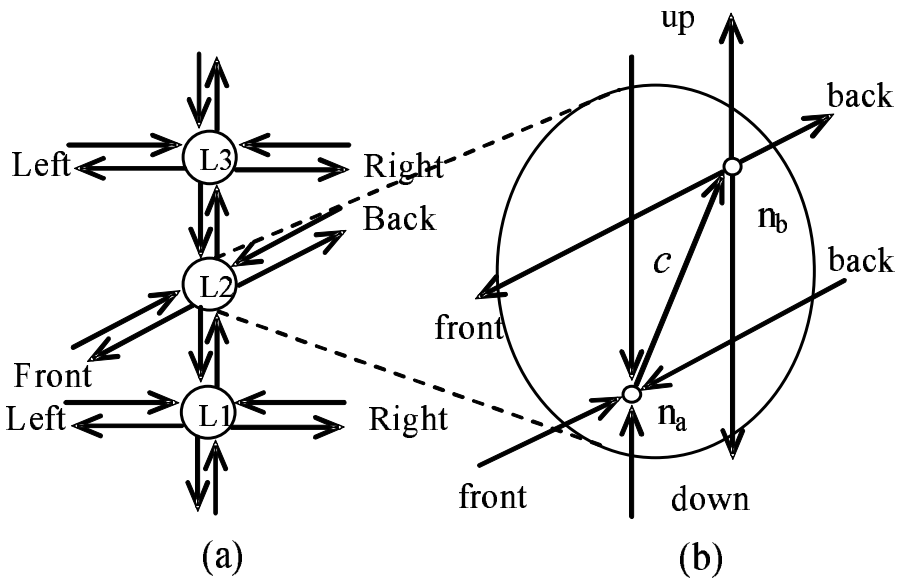


Figure II.4: A network flow model for multiplayer routing

communication throughput with different routing layer assignments.

II.A.4 Experimental Results

We first use Matlab's linear program package on a Sun Ultra10 workstation to compute MCF solutions. For the case with 100 nodes, the run time exceeds 24 hours. We then implement the MCF algorithm [4] using C programming language. The MCF algorithm derives the MCF solutions for cases with up to 319 nodes within 12 hours.

Results for uniform edge capacity mesh

Table II.1 describes the results of uniform edge capacity meshes with $n = 2$ to 10. We list the number of nodes and the throughput z.

From the experimental result, we have the following observations:

1. The throughput is $1/n$ when n is odd and $(n^2 - 1)/n^3$ when n is even.

2. The throughput is limited by edges on the middle column and row. When n

Table II.1: Results of uniform edge capacity mesh

n	Number of nodes	z
2	4	0.3750
3	9	0.3333
4	16	0.2343
5	25	0.2000
6	36	0.1620
7	49	0.1429
8	64	0.1229
9	81	0.1111
10	100	0.0990

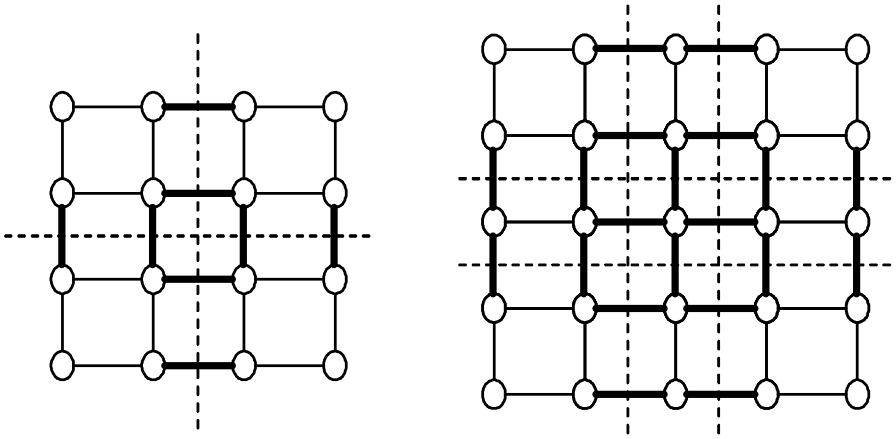


Figure II.5: Flow congestion for uniform edge capacities

is an even number, edges in the central row and column form the bottleneck of the flow. When n is an odd number, the two columns and two rows form the bottleneck.

Figure II.5 shows the bottleneck of communication flow for $n = 4$ and 5. The congested edges are marked with bold lines. Note that the bottleneck form the horizontal and vertical cut sets. The cut lines are shown with dashed lines.

Results of uniform row and column capacity mesh

For equal n, the throughput of a 90-degree mesh with uniform row and column capacities is exactly the same as that of the 90-degree mesh with fixed edge

Table II.2: Results of fixed total edge capacities

n	Number of nodes	z	improvement on z
2	4	0.375	0.00
3	9	0.333	0.00
4	16	0.281	20.01
5	25	0.240	20.00
6	36	0.208	28.57
7	49	0.185	28.56
8	64	0.169	33.32
9	81	0.148	33.35
10	100	0.134	36.36

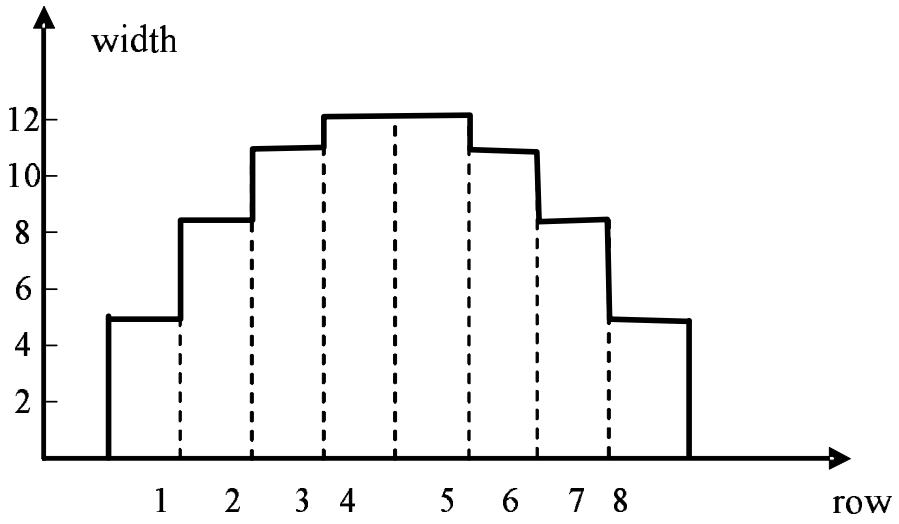


Figure II.6: Optimal row width of 9 by 9 mesh to maximize total throughput

capacities. We obtain no throughput improvement because the total capacity of the edges in each column and row is fixed.

Results for fixed total edge capacities mesh

For $n = 2$ to 10, Table II.2 shows the results of 90-degree mesh with fixed total edge capacities. The fourth column provides the throughput improvement compared to that of 90-degree mesh with uniform edge capacity. As we no longer limit the total capacity of each row or column, the average throughput improves 29.7% for $n = 4$ to 10.

Table II.3: Optimal capacities for vertical edges in 6 by 6 mesh

(Row—Col)	1	2	3	4	5	6	Sum
1	0.60	0.74	0.79	0.79	0.74	0.61	4.28
2	0.95	1.19	1.27	1.28	1.19	0.96	6.85
3	1.07	1.34	1.44	1.44	1.34	1.07	7.71
4	0.95	1.19	1.27	1.27	1.19	0.96	6.85
5	0.60	0.74	0.79	0.79	0.74	0.60	4.28

The results also show that all edges are congested. The optimal edge capacity is no longer uniform. The capacity is larger for the edges in the middle row and column. Table II.3 shows the optimal edge capacities for all the vertical edges in a 6 by 6 mesh. We list the sum of all the capacities of each row. Figure II.6 illustrates the optimal sums of the rows in a 9 by 9 mesh. Note that there are eight rows of vertical edges in a 9 by 9 mesh. The chip area is no longer a square, but a convex area as is shown in Figure II.6.

Results for 45-degree mesh

Table II.4 shows the results of 45-degree mesh for n = 2 to 12. To compare the results in table 4 and table 1, we use the cases with almost the same number of nodes. For instance, both the case of n = 4 in table 4 and the case of n = 5 in table 1 contain 25 nodes. The case with 45-degree mesh achieves the throughput of 0.209, which gains a 4.18 percent improvement. Also we compare the case of n = 7 in table 5 with the case of n = 9 in table 1. The case in table 5 contains 85 nodes, which has 4 more nodes than the case in table 1. The throughput of the 45-degree mesh case is 0.1260, which is 13.16% more than that of the 90-degree mesh case.

The congested edges also present a different pattern: they form 4 cut sets at four corners. Figure II.7 shows the flow congestion in 45-degree mesh for n = 5 and n = 6. The congested edges are in bold lines and the cut lines are in dashed lines.

Fig.II.8 explains why 45-degree routing is better than 90-degree routing.

Table II.4: Results of 45-degree mesh

n	Number of nodes	z
2	5	0.250
3	13	0.250
4	25	0.209
5	41	0.174
6	61	0.147
7	85	0.126
8	113	0.106
9	145	0.101
10	181	0.0828
11	221	0.0759
12	265	0.0673

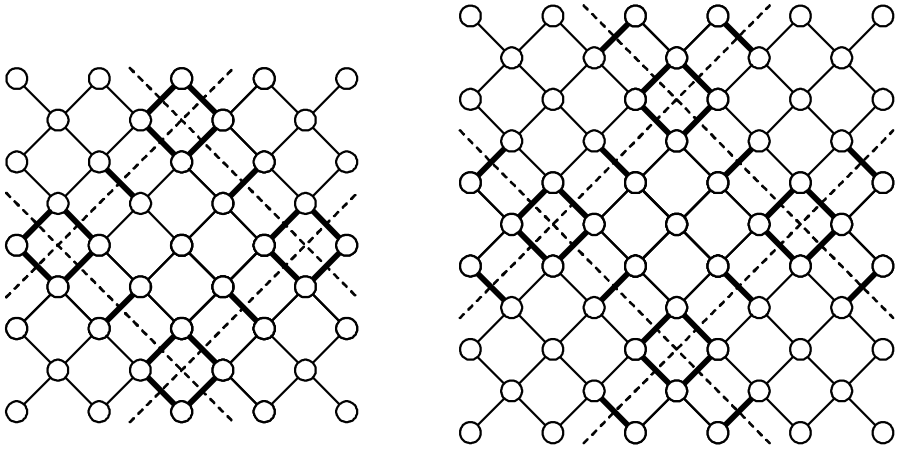


Figure II.7: Flow congestion in 45-degree mesh

Assume that we have a square shaped chip with two routing layers. Figure II.8(a) illustrates the case of 90-degree routing and Fig. 8(b) depicts the case of 45-degree routing. We draw a cut line for the horizontal congested edges in dashed lines in Figure II.8(a). Only the wires on the horizontal routing layer could cross the cut line and the number of wires across the cut line is d/D, where d is the wire pitch and D is the dimension of the chip. We then draw a similar cut line on Fig.II.8 (b). The number of edges across the cut line in each layer is $d/\sqrt{2}D$. The total number of wires crossing the cut line for the two layers in Fig. II.8(b) is $\sqrt{2}d/D$. Thus the upper bound of throughput increase to $\sqrt{2} = 1.414$. However, the

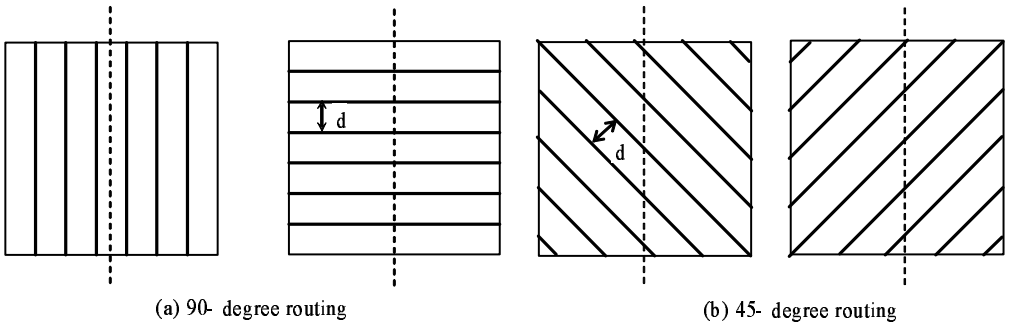


Figure II.8: Explanation of throughput increase for a 45-degree mesh

throughput is now limited by the cut edges at four corners.

Results for 90-degree and 45-degree mixed mesh

Table II.5 depicts the results for the 90-degree and 45-degree mixed mesh structure. Column 2 lists the throughput z. Column 3 lists the throughput improvements over the 90-degree meshes with uniform edge capacity. Columns 5 and 6 list the best capacity for horizontal and vertical edges, c1, and the best capacity for 45-degree edges, c2, respectively. Column 7 lists the normalized capacity ratio of the diagonal edges to the Manhattan edges. We have the following observations:

1. The throughput of the mixed mesh is better than the 90-degree mesh, given the equal communication resource. The improvement in the throughput is up to 20.04% for large number of nodes. It is also better than 45-degree mesh in terms of throughput.

2. With n increasing, the optimal ratio for the capacity of the 45-degree edge to the 90-degree edge approaches 5.6.

Results for routing layer assignment for 45-degree and 90-degree mixed mesh

We use our MCF model introduced in section 3.6 to compute the optimal routing direction assignment for mixed 45-degree and 90-degree routing. Assume

Table II.5: Results of 90-degree and 45-degree mixed mesh

n	z	Improvement on z (%)	c_1	c_2	$\sqrt{2}c_2/c_1$
2	0.375	0.00	1.0000	0.0000	0.00
3	0.333	0.00	1.0000	0.0000	0.00
4	0.245	4.85	0.2290	0.5452	3.36
5	0.219	9.53	0.2577	0.5249	2.88
6	0.185	14.04	0.1853	0.5761	4.39
7	0.166	16.01	0.2022	0.5641	3.94
8	0.148	20.11	0.1614	0.5930	5.19
9	0.134	20.40	0.1696	0.5872	4.89
10	0.120	21.31	0.1553	0.5988	5.44
11	0.110	21.48	0.1608	0.5935	5.22
12	0.101	22.05	0.1527	0.5992	5.55
13	0.094	22.14	0.1562	0.5967	5.40
14	0.087	22.68	0.1510	0.6004	5.62
15	0.082	22.71	0.1536	0.5986	5.51
16	0.076	22.95	0.1504	0.6008	5.65
17	0.0723	23.02	0.1524	0.5994	5.56

Table II.6: Throughput with different routing layer assignments

n	z(I)	z(II)	z(III)	z(IV)
5	0.0173	0.0147	0.0147	0.0171
6	0.0102	0.0083	0.0.0083	0.0101
7	0.0065	0.0053	0.0051	0.0064
8	0.0041	0.0034	0.0034	0.0041

that there are four routing layers and each of them is assigned to a different routing direction. Fig. II.9 shows four different routing layer assignments. The throughputs under four different assignments are listed in Table II.6. The throughputs with assignments IV and I are about 16 percent larger than the throughputs with assignments II and III.

Fig. II.10 explains why interleaving the Manhattan routing layers and diagonal routing layers can produce better throughput. In Fig. 10, we can see, given two points on the plane the shortest way to connect them is always a Manhattan line plus a diagonal line. Thus if we interleave the Manhattan routing layer and diagonal routing layer, the wires can go along shortest paths without paying more vias. This will produce better throughput.

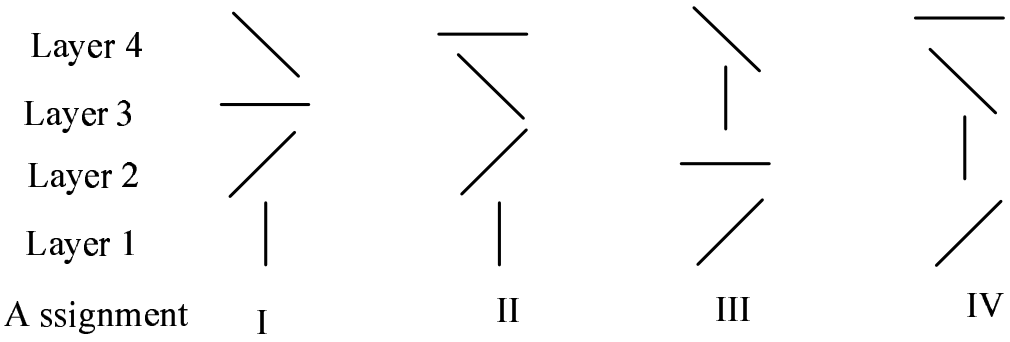


Figure II.9: Different routing directions assignments

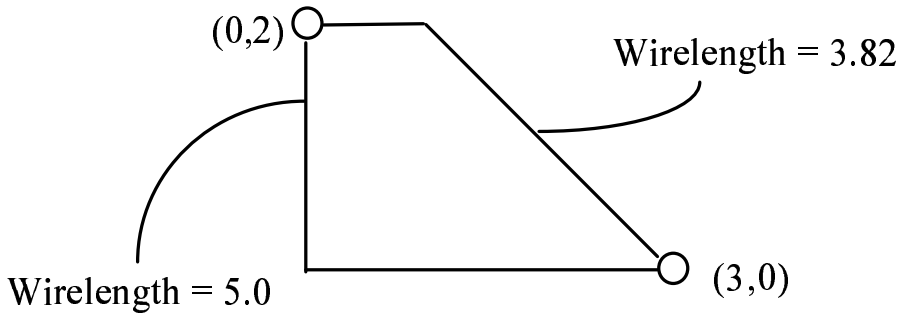


Figure II.10: Shortest path between two points on the plane

II.A.5 Conclusions

Interconnect sets important limitations for throughput of communications. For meshed interconnect structures, we propose a multi-commodity flow model to find the throughput and the best communication flow pattern. This provides a feasible upper bound of communication throughput. We study several different interconnect structures, including 90-degree meshes with uniform and non-uniform edge capacities, the 45-degree mesh, and the 90-degree and 45-degree mixed mesh. The results reveal the following basic trends:

- For the 90-degree mesh with uniform edge capacity, the congested edges lie in the center rows and columns. Moreover, the throughput can be improved by more than 30 percent if we choose optimal edge capacities instead of uniform edge capacities.

- For a 45-degree mesh structure, the throughput is better than that of 90-degree mesh with the same number of nodes. The bottleneck edges lie off the diagonal lines and form 4 cut sets.
- For a 90-degree and 45-degree mixed mesh, the throughput is better than either a 90-degree-only or a 45-degree-only mesh structure. Moreover, more resource should be allocated to 45-degree edges in a mixed structure as the number of nodes increases. With the consideration of vias in mind, we should interleave the Manhattan routing layers and diagonal routing layers.

II.B The Y-Architecture for On-Chip Interconnect: Analysis and Methodology

The *Y-architecture* for on-chip interconnect is based on pervasive use of 0-, 120-, and 240-degree oriented semi-global and global wiring. Its use of three uniform directions exploits on-chip routing resources more efficiently than traditional Manhattan wiring architecture. This paper gives in-depth analysis of deployment issues associated with the Y-architecture. Our contributions are as follows: (1) We analyze communication capability (throughput of meshes) for different interconnect architectures using a multi-commodity flow approach and a Rentian communication model. Throughput of the Y-architecture is largely improved compared to the Manhattan architecture, and is close to the throughput of the X-architecture. (2) We improve existing estimates for the wirelength reduction of various interconnect architectures by taking into account the effect of routing-geometry-aware placement. (3) We propose a symmetrical Y clock tree structure with better total wire length compared to both H and X clock tree structures, and better path length compared to the H tree. (4) We discuss power distribution under the Y-architecture, and give analytical and SPICE simulation results showing that the power network in Y-architecture can achieve 8.5% less IR drop than an equally-resourced power network in Manhattan architecture. (5) We propose the

use of *via tunnels* and *banks of via tunnels* as a technique for improving routability for Manhattan and Y-architectures.

II.B.1 Introduction

The *Y-architecture* refers to the use of 0-, 120-, and 240-degree oriented wires for on-chip interconnect, along with supporting methodologies including hexagonal die shapes, hexagonal power and clock distribution, etc. This name is first used in [63] in the same spirit as the "X architecture" for pervasive use of 45- and 135-degree angles [86].

Compared to the traditional Manhattan (M-) architecture, the Y-architecture offers many potential advantages, such as substantially reduced wirelength and power consumption, and increased communication bandwidth for a wide range of demand topologies. Combined with the M-architecture, the Y-architecture can be applied to the upper two layers to improve global interconnects, such as clock and power distribution networks. Moreover, unlike the X-architecture, the Y-architecture supports a regular routing grid and novel means of avoiding via blockage effects.

Two previous series of works examine the potential use of Y-architecture for integrated circuits: a series of LSI Logic patents by Rostoker et al. [78] [79] [80], and a series of works by Cheng and coauthors [62] [63]. Together, these works set out a number of ideas for device architecture, floorplanning, and place-and-route. However, a number of technical gaps still exist, ranging from clock and power distribution methodology to wireability and throughput analysis. In this work, we provide a more complete, technically in-depth analysis of key deployment and methodology issues associated with the Y-architecture. Our main contributions are as follows:

- We give a more realistic throughput analysis using a communication model based on Rent's rule. Our results show that the Y-architecture provides a throughput improvement of about 20% over the M-architecture for a square

chip, very close to the throughput of the X-architecture.

- We improve existing estimates for the wirelength reduction of various interconnect architectures by taking into account the effect of routing-geometry-aware placement. Our estimate is based on a simulated annealing placer, driven by wirelength in different routing geometries. We also discuss and analyze a "virtuous cycle" effect: reduction of overall wirelength results in decreased routing area, which in turn leads to further wirelength reduction.

- We discuss clock and power distribution under the Y-architecture. For clock distribution we propose a symmetrical Y clock tree structure with better total wire length compared to both H and X clock tree structures, and better path length compared to the H tree. For power distribution we give analytical and SPICE simulation results showing that a mesh power network in Y-architecture can achieve 8.6% less IR drop than an equally-resourced mesh power network in M-architecture.

- To fully utilize the uniform routing grid available in M- and Y-architectures, and to deal with future increases in via demand due to repeaters [81], we propose the use of *via tunnels* and *banks of via tunnels* to improve routability in these architectures. Such techniques are not obvious with the X-architecture.

- We discuss lithography and manufacturing infrastructure needs, particularly in mask write, related to possible adoption of the Y-architecture.

The remainder of the paper is organized as follows. Section II.B.2 presents throughput analysis for square-shaped chips. Section II.B.3 discusses wirelength reduction with hexagonal routing. Sections II.B.4 and II.B.5 examine clock and power distribution, and Section II.B.6 discusses routability issues. The paper concludes in Section IV.E. Discussion about the "virtuous cycle" wirelength reduction effect, manufacturing issues and a supporting approximation of IR-drop are given in the Appendices.

II.B.2 Communication Throughput in Meshes

A multi-commodity flow (MCF) approach was developed by Chen and coauthors [63] [64] to evaluate communication efficiency of different interconnect architectures. Communication resources are decomposed into a 2D array of slots. A uniform communication requirement is assumed, i.e., every pair of nodes communicates with equal demand and all communications occur at the same time. The throughput, defined as the maximum amount of communication flow simultaneously achievable between every pair of nodes, is computed by a provably good multicommodity flow (MCF) algorithm [68] and is used to measure communication capabilities of different interconnect architectures.

Rentian Communication Demand

The uniform pairwise communication used in [63] is simple and general. However, it is not very realistic, since in a well-designed layout the probability of communication decreases with increasing distance between nodes. Stroobandt and Campenhout [82] derive from Rent's rule an expression for *occupation probability*, i.e., the probability that a given pair of points will be connected by a wire in an optimal physical placement of the circuit. For a hierarchical placement of a circuit with Rent exponent p in a two-dimensional Manhattan grid, the occupation probability of a pair of points with Manhattan distance D between them can be approximated by CD^{2p-4} where C a normalization constant.[2] When only 2-pin nets are considered, the occupation probability indicates the probability of communication between pairs of nodes. In the following, to ensure a fair comparison of the communication throughput capabilities of different interconnect architectures, we assume a Rentian communication demand, i.e., we set the communication demand between any two unit-area slots to be proportional to D^{2p-4}, where D is the Euclidean distance D between them.

[2] C depends on the routing architectures and the underlying distance metric.

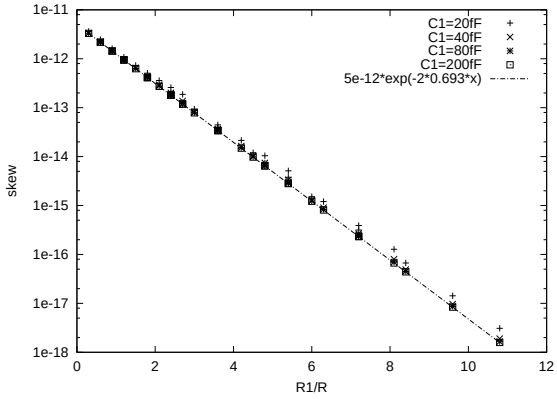


Figure II.11: 7×7 meshes with different interconnect architectures.

Communication Throughput

A widely quoted survey of Bakoglu [58] indicates that the Rent exponent at the chip and module level of high-speed computers is approximately 0.63. We compute the *throughput* – defined to be the maximum fraction of communication demand simultaneously satisfied between every pair of nodes in $n \times n$ square meshes – using the MCF algorithm. The throughput is tightly correlated to routability, and describes communication capabilities of different interconnect architectures. Figure II.11 illustrates three 7×7 meshes using different interconnect architectures. For Y-architecture, the shape of each slot is hexagonal, and the enclosing box of the slots is close to square. Although Y-architecture meshes are different from M- and X-architecture meshes, this does not significantly affect the communication demand. For the 17×17 Y-mesh, total communication demand is only 1.8% different from that for other architectures.

In the experiments, total routing area is set to be the same for all meshes. We normalize the computed throughput so that it is independent of the dimension of meshes and total communication demand.[3] Table II.7 lists the results for $n \times n$ meshes with n ranging between 9 and 17. Compared to the M-architecture,

[3]For example, the computed throughput on a $n \times n$ mesh using Y-architecture is normalized by $\frac{TD_M}{TD_Y} \cdot D_c/n$, where TD_M and TD_Y are total demand for M- and Y-architectures, respectively, and D_c is the communication demand crossing the horizontal middle cut line on the Manhattan mesh.

Table II.7: Normalized throughput (and improvement vs. M-architecture) in square meshes with Rentian demand.

n	#Mesh Nodes	M-architecture Thrpt	Y-architecture		X-architecture	
			Thrpt	Impr. (%)	Thrpt	Impr. (%)
9	81	1.989	2.354	18.30	2.412	21.25
10	100	1.989	2.366	18.92	2.419	21.59
11	121	1.987	2.374	19.47	2.420	21.78
12	144	1.986	2.382	19.94	2.423	22.00
13	169	1.991	2.386	19.84	2.425	21.76
14	196	1.990	2.392	20.19	2.429	22.02
15	225	1.988	2.395	20.47	2.429	22.14
16	256	1.992	2.400	20.44	2.430	21.98
17	289	1.992	2.402	20.58	2.433	22.11

the Y-architecture provides an average throughput improvement of 19.8% for these meshes, which is comparable to the 21.9% improvement achieved by the X-architecture. For a 17×17 mesh, Y-architecture provides a throughput improvement of 20.6% while X-architecture achieves an improvement of 22.1%.

A rectangular chip has communication bottlenecks on two (horizontal and vertical) middle cut lines. The physical dimension of the middle part of the chip restricts the communication flow and thus prevents us from achieving larger throughput. For M- and Y-architectures, convex-shaped chips (diamond chip for M-architecture and hexagonal chip for Y-architecture) produce better throughput by allowing more wires to cross the original middle cut lines [63].[4] Note that the use of octagonal chips for the X-architecture is undesirable, since the wafer cannot be tiled by octagons without waste.

II.B.3 Wirelength Reduction

Because of its restrictions on routing directions, the M-architecture entails significant added wirelength beyond the Euclidean optimum. In the Y-architecture, routing is allowed along three uniform orientations, and total wirelength is ex-

[4]Note that it is not necessarily to use a regular hexagon for the Y-architecture: either horizontal or vertical symmetry suffices.

pected to be reduced. An accurate cost-benefit analysis of Non-Manhattan routing is impossible without good estimation of the expected wirelength reduction when switching from Manhattan to Non-Manhattan routing. Some estimates appeared in the literature, including (i) experiments with exact [76] and heuristic Steiner algorithms [71] [72] for nets generated randomly or extracted from real VLSI designs, and (ii) analysis of wirelength reduction for 2-pin nets with pins randomly located in the plane [78] [83] [63].

The previous estimates do not adequately address the effect of routing-geometry-aware placement on the overall wirelength improvement. Previous studies of the routing demand using different traditional placers [74] show that Manhattan placers tend to align circuit elements either vertically or horizontally, leaving few opportunities to exploit additional routing directions. A Y-aware or X-aware placer factors in hexagonal or octilinear wiring during placement, and results in better placements of nets when such wiring is used to route the nets. Therefore, total wirelength can be greatly reduced.

To estimate the wirelength improvement achieved by Y-aware or X-aware placement and routing versus Manhattan placement and routing, we have built a simplified placer which uses simulated annealing driven by hexagonal or octilinear wirelength estimation. The input of the placer is a simplified netlist extracted from MCNC instances, in which a list of cells is specified for each net. After a random initial placement of cells, two cells are randomly selected, and we decide whether to swap these two cells based on the current annealing temperature and the new SMT cost with hexagonal or octilinear routing, which is computed using an exact SMT algorithm, GeoSteiner [76]. The initial temperature of the simulated annealing algorithm is specified so that it is far larger than the standard deviation of total wirelength distribution [69]. For each temperature, the number of swaps is on the order of 100 times the number of cells [75]. The new temperature is generated by multiplying the current temperature by $\alpha = 0.95$, which is a relatively large α for simulated annealing [75].

Table II.8: Average wirelength improvements for Non-Manhattan placement and routing vs. Manhattan placement and routing (%).

Instance	#nets	Y-Arch	X-Arch	Euclidean
C2	601	4.81	8.92	11.04
BALU	658	7.13	9.29	11.07
PRIMARY1	695	7.32	10.31	13.03
C5	1438	8.34	11.48	12.73

Table II.9: Total wirelength of instance "C2" with different combinations of placement and routing.

routing-geometry driven placement	routing-geometry			
	Rect	**Hex**	**Oct**	**Euclidean**
Rectilinear	1805	1841.8	1719.2	1683.9
Hexagonal	2002	1690.3	1718.3	1640.8
Octilinear	1908	1799.6	1644.0	1617.4
Euclidean	1865	1772.1	1646.9	1605.7

For each instance and each routing geometry (rectilinear, hexagonal, octilinear and Euclidean), we run the placer 5 times, and get the best wirelength with routing-geometry-aware placement and routing. The wirelength improvements achieved by Non-Manhattan placement and routing are summarized in Table II.8. In Tables II.9–II.12 we give total wirelengths obtained with different combinations of placement and routing for each of the four testcases.

According to the results, the Y-architecture achieves a wirelength improvement up to about 8.3%. The X-architecture further reduces total wirelength to be up to about 11.4% over M-architecture and it produces about 3.3% wirelength reduction over Y-architecture with the cost of one more routing direction.

We note that in the above experiments the placer uses a fixed area die. However, reduction of overall wirelength results in decreased routing area, which in turn leads to further wirelength reduction, creating a "virtuous cycle" effect. An analysis of this effect is given in Appendix Appendix A.

Table II.10: Total wirelength of instance "Balu" with different combinations of placement and routing.

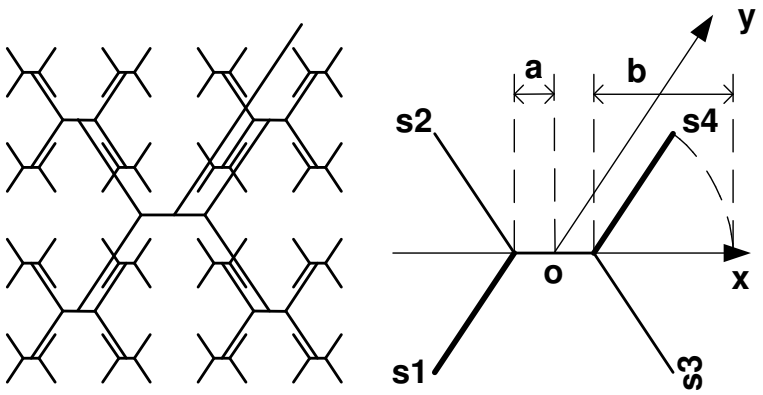
routing-geometry driven placement	routing-geometry			
	Rect	**Hex**	**Oct**	**Euclidean**
Rectilinear	1820	1856.0	1728.4	1694.7
Hexagonal	2010	1718.2	1744.0	1669.3
Octilinear	1886	1785.6	1650.9	1621.3
Euclidean	1898	1769.8	1654.6	1616.5

Table II.11: Total wirelength of instance "Primary1" with different combinations of placement and routing.

routing-geometry driven placement	routing-geometry			
	Rect	**Hex**	**Oct**	**Euclidean**
Rectilinear	2058	2080.8	1942.5	1903.7
Hexagonal	2250	1907.4	1931.9	1844.5
Octilinear	2136	2004.8	1862.0	1828.0
Euclidean	2124	1976.8	1854.1	1805.5

Table II.12: Total wirelength of instance "C5" with different combinations of placement and routing.

routing-geometry driven placement	routing-geometry			
	Rect	**Hex**	**Oct**	**Euclidean**
Rectilinear	3557	3625	3340	3272
Hexagonal	3848	3260	3306	3157
Octilinear	3569	3390	3149	3097
Euclidean	3628	3397	3183	3104



(a) Y Clock Tree.

(b) A One-Level Y Clock Tree.

Figure II.12: Y Clock Tree.

Table II.13: Path length and total wirelength of H-tree, X-tree and Y-tree.

	Path Length	**Total Wirelength**
H-tree	$(2^n - 1)$	$\frac{3}{2} \cdot 2^n(2^n - 1)$
X-tree	$\frac{\sqrt{2}}{2} \cdot (2^n - 1)$	$\sqrt{2} \cdot 2^n(2^n - 1)$
Y-tree	$\frac{1}{2}(1 + \frac{\sqrt{3}}{3}) \cdot (2^n - 1)$	$\frac{1+\sqrt{3}}{2} \cdot 2^n(2^n - 1)$

II.B.4 Y Clock Tree

Clock distribution networks synchronize the flow of data signals among synchronous data paths. The design of these networks can dramatically affect system-wide performance and reliability. The "H" clock tree [59] is widely used in the IC industry. In the H-tree, clock terminals are arranged in a symmetric fashion, and are connected by a planar hierarchy of symmetric "H" structures. When octilinear routing is allowed, the "H" structure can be replaced with an "X" structure, so that source-sink path (i.e., insertion) delay and total wirelength are decreased. However, significant undesirable overlapping (superposition) will occur between parallel interconnect wires in the X-tree.

With three uniform routing directions, a *Y clock tree* can be built as depicted in Figure II.12(a), essentially giving a "distorted X-tree" with reduced wirelength and no superposed parallel wires. Let the distance between two adjacent

clock terminals be 1. Path length from the clock source to clock terminal, as well as total wirelength, are compared with H-tree and X-tree in Table II.13. The Y clock tree has a path length of $.7887 \cdot (2^n - 1)$, 21.1% less than the H-tree. Its total wirelength is $1.366 \cdot 2^n(2^n - 1)$, 8.9% less than H-tree, and 3.4% less than X-tree. Actually, the one-level Y-tree shown in Figure II.12(b) is the optimal Euclidean Steiner Minimum Tree to connect four adjacent clock terminals $s1, s2, s3, s4$ and the clock source o. Thus the Y clock tree provides minimal total wirelength among all clock trees with similar symmetric structure. The further advantage of Y clock tree is that there is no overlapping of parallel interconnect wires. It can be shown:

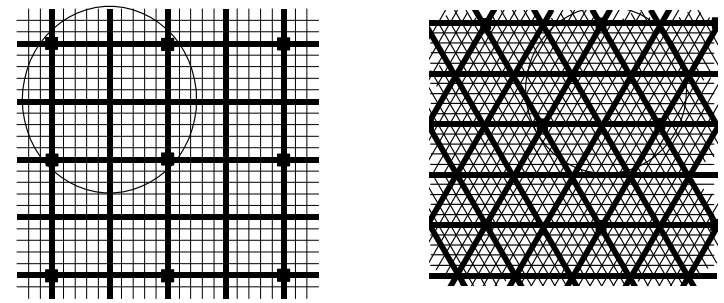
Theorem 1 *Let the distance between two adjacent clock terminals be D. The minimum distance between two parallel interconnect wires is $\frac{\sqrt{3}-1}{4}D$.*

Proof. Suppose there is a coordinate system with a 0° x-axis, a 60° y-axis and the origin $(0, 0)$ at the center of the main Y-tree structure (see Figure II.12(b)). Then in a one-level Y-tree, the two bold interconnect wires that are parallel to the y-axis in the figure have x-coordinates of $\pm a$. In a two-level Y-tree, the lowest-level y-axis-parallel interconnect wires have x-coordinates of $\pm a \pm 2a$ and $\pm a \pm 2(a+b)$. Generally, in an n-level Y-tree, x-coordinates of the lowest-level y-axis-parallel interconnect wires are $\pm a \pm (2a \text{ or } 2(a+b)) \pm ... \pm (2^{n-1}a \text{ or } 2^{n-1}(a+b))$.

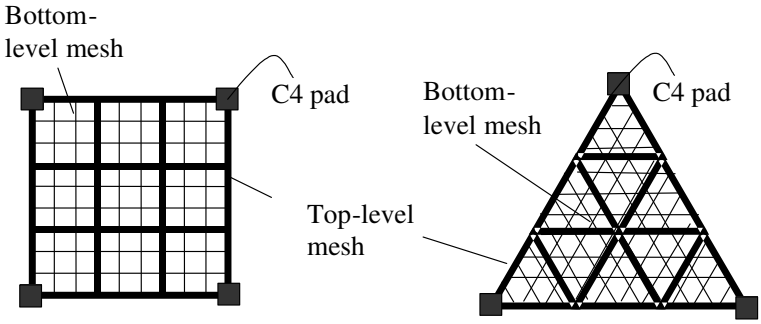
Since $a = \frac{D}{2}(1 - \frac{\sqrt{3}}{3})$, and $(a+b) = \frac{D}{2}(1 + \frac{\sqrt{3}}{3})$, the y-coordinates can be written as $(\pm 2^0 \pm 2^1 \pm ... \pm 2^{n-1}) \cdot \frac{1}{2}D + (\pm 2^0 \pm 2^1 \pm ... \pm 2^{n-1}) \cdot \frac{\sqrt{3}}{6}D$. These values cannot be zero because the values of $\pm 2^0 \pm 2^1 \pm ... \pm 2^{n-1}$ must not be zero, and the minimum absolute value among them is $a = \frac{D}{2}(1 - \frac{\sqrt{3}}{3})$. Thus the minimal distance between two parallel interconnect wires in the Y clock tree is $\frac{\sqrt{3}}{2}a = \frac{\sqrt{3}-1}{4}D$.

II.B.5 Y Power Distribution

Excessive voltage drop in the power grid can slow device switching speed and reduce noise margin. Robust power distribution within available area resource is critical to chip performance and reliability. Hierarchical mesh structures are



(a) Two-level power mesh.



(b) Representative areas.

Figure II.13: Power distribution networks and representative areas for M- and Y-architectures.

widely used for power distribution in high performance chips because of their robustness [60]. In this section, we show that power distribution in the Y-architecture is not only natural, but achieves less IR drop than equally-resourced mesh distribution in the M-architecture.

Our comparison is based on the following model of the power distribution network.

- The power distribution network is constructed by a hierarchy of mesh structures connected by vias at crossing points of wires. Each mesh has equal wire spacing and wire width. Ignoring the resistance of vias,[5] we assume perfect

[5] In practice, high current density on vias often causes reliability problems. In the Y-architecture, assuming same wire width, the area of intersection (overlap) between two adjacent-layer wires is larger than in the M-architecture. Hence, we can place a bigger via between adjacent layers or place more vias in the via array between adjacent layers to reduce resistance and current density for vias. Let A_Y, A_X, and A_M represent this area for Y-, X- and M-architectures, respectively. We have $A_Y = 1.1547 A_M$ and $A_X = 1.414 A_M$.

contact at each crossing point.

- On top of metal layers, there are arrays of C4 power pads evenly distributed on the surface of the power mesh.
- Under the bottom-level mesh, there are devices connected to the wires of the bottom-level mesh. The devices are modeled as uniform current sinks and placed at crossing points of the bottom-level mesh.

In state-of-art designs, there is a fairly large number (> 100) of power pads evenly distributed on the surface of the top-level power mesh [88]. It is reasonable to assume that the whole power mesh is an infinite resistive grid constructed by replicating the area surrounded by adjacent power pads. Figure II.13 illustrates two-level power meshes and the *representative areas* in the M- and Y-architectures. Our analysis and circuit simulations consider only the worst-case IR-drop on the representative area. This method is also used in [67].

IR-Drop on Single-Level Power Mesh

Static IR-drop on a hierarchical power mesh depends largely on the top-level mesh since usually the top-level mesh is wider and coarser and most current flows along the top-level mesh. Here we analyze and compare the worst-case static IR-drop on a single-level power mesh in the M- and Y-architectures.

IR-Drop on Single-Level Power Mesh in the Y-Architecture

A single-level power mesh in the Y-architecture is abstracted as an infinite triangular resistive lattice with edge resistance R_Y.[6] We examine IR-drop in the triangular area with N_Y rows surrounded by three adjacent power pads[7]. In this case, the worst-case IR-drop appears at the center of this representative area. Each power pad supplies a current $I_Y = N_Y^2 i$ to the power mesh, where i is the current drain at each intersection on the mesh.

[6] Note that for a uniform mesh with fixed total routing area, the edge resistance is independent of the number of metal lines on the mesh. When the number lines increases, wire pitch and wire width decreases with the same ratio, and the edge resistance remains the same.

[7] E.g., for the top-level Y-architecture mesh shown in Figure II.13(b), N_Y is equal to 3.

Assume there is a coordinate system with the origin at the center of the power mesh, and 0-degree and 120-degree lines used as m-axis and n-axis, respectively. We analyze the voltage drop between the node $(0,0)$ and the power pad at $(\frac{N_Y}{3}, -\frac{N_Y}{3})$ by considering currents from power pads and evenly distributed current sinks separately.

IR-drop caused by currents from power pads. Suppose that a current I_Y enters the lattice at the node (m_s, n_s) and leaves at infinity. The voltage drop for any node on the lattice is analyzed in [57]. The voltage drop between (m_s, n_s) and (m, n), denoted as $V_{(m_s,n_s)}(m, n)$, is given by the integral

$$\frac{I_Y R_Y}{2\pi} \int_0^{\pi/2} \left(1 - e^{-|(m-m_s)-(n-n_s)|x}\right) \cos(((m-m_s)+(n-n_s))y)/(\sinh x \cos y)\ \mathrm{d}y, \tag{II.4}$$

where $2\cosh x \cos y + \cos 2y = 3$. When $|(m-m_s)-(n-n_s)|$ is large, the voltage drop $V_{(m_s,n_s)}(m, n)$ can be approximated as

$$\frac{I_Y R_Y}{4\sqrt{3}\pi}[\ln((m-m_s)^2 + (n-n_s)^2 - (m-m_s)(n-n_s)) + c_1], \tag{II.5}$$

where $c_1 = 3.6393$ is a constant.[8]

Let $V_{(m_s,n_s)}$ denote the voltage drop between $(0,0)$ and the power pad at $(\frac{N_Y}{3}, -\frac{N_Y}{3})$ caused by the current source at (m_s, n_s). According to the above approximation, we have

- when $(m_s, n_s) = (\frac{N_Y}{3}, -\frac{N_Y}{3})$, $V_{(m_s,n_s)} \approx (I_Y R_Y / 4\sqrt{3}\pi)(2\ln N_Y - \ln 3 + c_1)$;
- when $(m_s, n_s) \neq (\frac{N_Y}{3}, -\frac{N_Y}{3})$, $V_{(m_s,n_s)} = V_{(m_s,n_s)}(0,0) - V_{(m_s,n_s)}(\frac{N_Y}{3}, -\frac{N_Y}{3}) \approx (I_Y R_Y / 2\sqrt{3}\pi) \ln \frac{D_0}{D_s}$, where D_s is the Euclidean distance between (m_s, n_s) and $(\frac{N_Y}{3}, -\frac{N_Y}{3})$, and D_0 is the Euclidean distance between (m_s, n_s) and $(0,0)$. The constant $c_2 = \sum_{(m_s,n_s)\neq(\frac{N_Y}{3},-\frac{N_Y}{3})} \ln \frac{D_0}{D_s}$ can be computed by a simple algorithm, which calculates the summation for all the current sources within a circle around the origin. As the radius of the circle increases, the summation converges to a value of $c_2 = -1.173679$.

[8] See the Appendix Appendix B for details of this approximation.

Therefore, if only currents from power pads are considered, the voltage drop between $(0,0)$ and the power pad at $(\frac{N_Y}{3}, -\frac{N_Y}{3})$ is

$$V_{source} = \sum_{(m_s,n_s)} V_{(m_s,n_s)} = \frac{I_Y R_Y}{2\sqrt{3}\pi}(\ln N_Y + C_Y), \tag{II.6}$$

where $C_Y = c_1/2 - \ln 3/2 + c_2 = 0.09666$.

IR-drop caused by evenly distributed current sinks. Next, we consider the voltage drop caused by current sinks at the intersections of the power mesh. If the voltage between $(0,0)$ and (m,n) is denoted by $V_{sink}(m,n)$, by a combination of Ohm's and Kirchhoff's Laws we have

$$\begin{aligned} &V_{sink}(m-1,n) + V_{sink}(m+1,n) + V_{sink}(m,n+1) + V_{sink}(m,n-1) + \\ &V_{sink}(m-1,n-1) + V_{sink}(m+1,n+1) - 6V_{sink}(m,n) = iR_Y. \end{aligned} \tag{II.7}$$

If the resistive lattice is regarded as a discrete approximation to a continuous resistive medium, we will obtain a potential function proportional to D^2, where D is the Euclidean distance from the origin. Therefore, we assume the following representation for the voltage between $(0,0)$ and (m,n):

$$V_{sink}(m,n) = k\ (m^2 + n^2 - mn), \tag{II.8}$$

where k is a constant. Equation (II.7) then yields

$$V_{sink}(m,n) = \frac{iR_Y}{6}\ (m^2 + n^2 - mn). \tag{II.9}$$

When only current sinks are considered, the voltage drop between $(0,0)$ and the power pad at $(\frac{N_Y}{3}, -\frac{N_Y}{3})$ is

$$V_{sink} = V_{sink}(\frac{N_Y}{3}, -\frac{N_Y}{3}) = \frac{I_Y R_Y}{18}. \tag{II.10}$$

Verification of Worst-Case IR-Drop. From the above analysis, we obtain the voltage drop at the center:

$$\mathbf{V_Y = V_{source} + V_{sink}} \approx \frac{\mathbf{I_Y R_Y}}{\mathbf{18}} + \frac{\mathbf{I_Y R_Y}}{\mathbf{2\sqrt{3}}\pi}(\ln \mathbf{N_Y} + \mathbf{C_Y}), \tag{II.11}$$

Table II.14: Simulation results for worst-case IR-drop on the single-level power mesh in the Y-architecture, compared to estimated values (mV).

N_Y	**IR-Drop**	**Estimated IR-Drop**	**Error**
3	166.67	165.39	1.28
6	229.17	229.08	0.09
9	266.36	266.34	0.02
12	292.78	292.77	0.01
15	313.28	313.27	0.01
18	330.03	330.03	0.00
21	344.20	344.19	0.00

where $C_Y = 0.09666$.

To verify the above formula for worst-case IR-drop on the single-level power mesh, we use HSpice to simulate various power meshes with different values of N_Y's. Since the problem is linear in nature, in our experiments the resistance of each wire segment R_Y is simply set to be $1K\Omega$, and the total current drain in the area I_Y is set to be $1mA$. We list simulation results for N_Y from 3 to 21 in Table II.14, and compare them with the estimated values from the formula. The results show that the formula is accurate, with error less than 1%.

Comparing IR-Drop on Single-Level Power Mesh

For a single-level power mesh in the M-architecture, worst-case IR-drop is analyzed and verified in [66]. Suppose the power mesh has edge resistance R_M, number of rows within the representative area N_M and current supplied by each power pad I_M, the worst-case IR-drop on the single-level Manhattan (M-) mesh is:

$$\mathbf{V_M} \approx \frac{\mathbf{I_M R_M}}{\mathbf{8}} + \frac{\mathbf{I_M R_M}}{\mathbf{2}\pi}(\ln \mathbf{N_M} + \mathbf{C_M}), \tag{II.12}$$

where $C_M = -0.1324$.

To fairly compare the Y-mesh and M-mesh, we constrain the two meshes to have the same wire material and thickness, cover the same area (same total current drain) with the same wiring resource, and have the same number of crossing points and power pads. Therefore, we have $R_Y = \sqrt{3}R_M$, $I_Y = I_M$, and $N_Y = N_M$.

Table II.15: IR-drop improvements in single-level Y-mesh vs. M-mesh.

N_M	**Estimated IR-Drop (mV) in M-mesh**	**IR-Drop Impr. (%) with Y-mesh**
2	214.25	10.78
3	278.78	8.28
4	324.56	7.11
5	360.08	6.41
6	389.09	5.93
7	413.63	5.58
8	434.88	5.31
9	453.63	5.09

According to Equations (II.11) and (II.12), worst-case IR-drop on the single-level Y-mesh is less than that on the M-mesh by

$$\Delta \mathbf{V} = \mathbf{V_M} - \mathbf{V_Y} = \mathbf{c I_M R_M}, \tag{II.13}$$

where $c = 0.02309$. We list IR-drop improvements with Y-mesh for different values of N_M. The number of wire lines between two adjacent power pads on the top-level power mesh is usually small [66]. When $N_M = 4$, static IR-drop improvement of the Y-mesh over M-mesh is 7.1%.

IR-Drop on Hierarchical Power Mesh

In practice, power is distributed through a hierarchy of six or more metal layers. In this section, we simulate hierarchical power networks for the Y- and M-architectures using HSpice, explore different configurations of power networks, and compare the best solutions. We assume an equal sum of routing resources (i.e., total routing area) for Y- and M-architecture power distribution across layers M6, M5 and M4. In our experiment below, we set the total wiring area of M6, M5 and M4 to be 52% of the total representative area. The representative area for the Manhattan mesh is set to be a $1.2mm$ by $1.2mm$ square. To achieve the same power pad density, the representative area for the Y power grid is an equilateral triangle with edge length $1.289mm$. Further details of our comparison are as follows.

- Layer thickness and resistivity parameters of a 6-layer process are taken from TSMC $0.13\mu m$ copper process information [84]. Layer thicknesses are $0.33\mu m$ for M1, $0.36\mu m$ for M2-5, and $1.02\mu m$ for M6.

- M1-M3 power distribution is native to library cells and blocks, requiring a common interface (0-degree) at M4. Power routing in M1-3 has the same pitch in both the Y and Manhattan solutions: M1 has pitch of $8\mu m$ and wire width of $2\mu m$, M2 has pitch of $60\mu m$ and wire width of $4\mu m$, and M3 has pitch of $60\mu m$ and wire width of $4\mu m$. M4 pitch is fixed at $75\mu m$ to enable matchup with M1-3 macros and an apples-to-apples comparison.

- Allowed values of wiring separations (= pitches) on M5 and M6, denoted by S5 and S6, are $\{600\mu m, 300\mu m, 150\mu m, 75\mu m\}$. Allowed percentages of total wiring area used on M4 and M5, denoted as P4 and P5, are {10%, 20%, 30%, 40%, ..., 80%}.

- $1V$ voltage sources are placed at the corners of representative areas. Each current sink on M1 (between two adjacent vias) is 5.21×10^{-7}A.

All combinations of wire pitch and wire width of M4, M5, and M6 are exhaustively searched. In the best M-architecture configuration, M6 has wire pitch of $300\mu m$ and uses 70% of the power routing resource; M5 has wire pitch of $75\mu m$ and uses 20% of the resource. The IR-drop produced by this configuration is $38.5mV$. In the best Y-architecture configuration, M6 has pitch $600\mu m$ and uses 70% of the power routing resource, while M5 has pitch $150\mu m$ and again uses 20% of wiring area. The IR-drop is $35.2mV$, which is 8.6% smaller than that of the best M-architecture solution. Ongoing research seeks a more general and formal comparison.

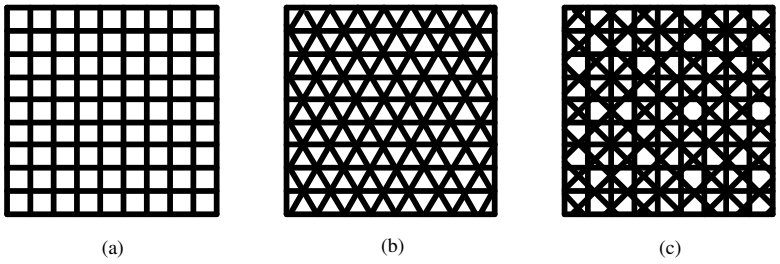


Figure II.14: Routing grids in M-, Y- and X-architectures.

II.B.6 Routability in the Y-Architecture

Uniform Routing Grid

A nice property of the Y-architecture is that there is a natural, uniform routing grid. Figure II.14(a)(b) illustrates the routing grid in the M- and Y-architectures, wherein each routing layer has exactly the same wiring pitch. Figure II.14(c) shows the X-architecture grid, where identical layer pitches imply that wire intersection points are not coincident. It is therefore difficult to find a natural, resource-efficient, uniform wiring grid in the X-architecture.

A uniform routing grid is expected to benefit large VLSI designs for three main reasons. (1) It enables continued use of today's dominating gridded routing algorithms. (2) Most advanced manufacturing processes require uniform width and spacing for M2 through M5, e.g., to simplify determination of legal via locations. Uniform pitch and dimension is also increasingly required for printability in subwavelength lithography. (3) The uniform routing grid can permit integral coordinates (even if absolute positions have irrational coordinates!), significantly simplifying detailed routing and design rule checking algorithms.

Via Tunnels and Via Tunnel Banks

Another advantage of the uniform global routing grid is that we can utilize *via tunnels* and *via tunnel banks* to avoid the fragmentation of routing resources caused by vias; this improves overall chip routability. In multi-layer routing, wire

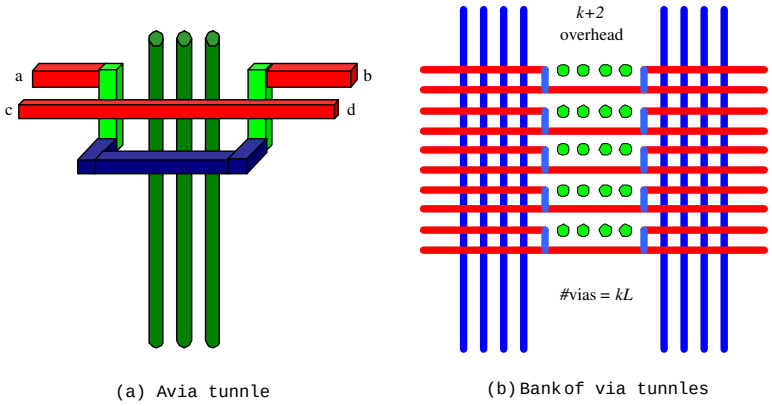
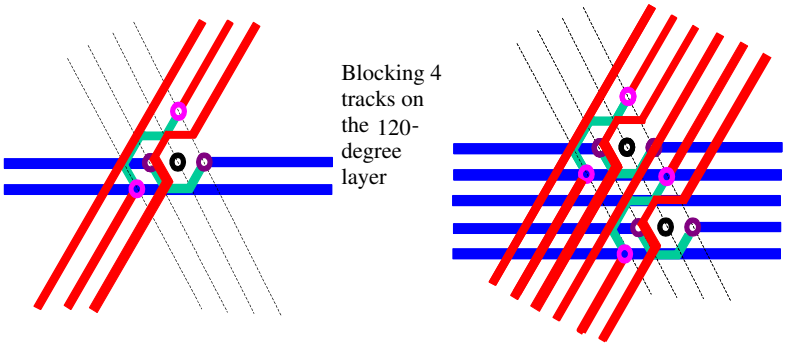


Figure II.15: Via tunnel and bank of via tunnels in the M-architecture.

tracks are blocked on the layers that a via passes through. Traditional routing schemes scatter vias all over the chip, and this fragmentation of routing resources may cause serious wireability problems; this is called "via blockage effect". As we approach the 65nm technology node, this effect becomes more serious, since buffering of global wires introduces many via chains that go through all the way from the top-level metal down to the gate layer. We believe that the proposed use of via tunnels and via tunnel banks will reduce the via blockage effect and thus improve routability and wiring density.

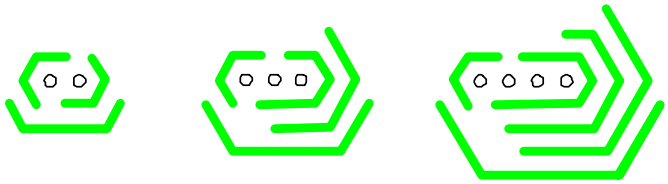
Figure II.15(a) shows an example of a *via tunnel* in the Manhattan architecture. There are two routing layers shown in the figure: the upper layer is for horizontal routing and the lower layer is for vertical routing. Terminals a and b are connected by detouring the horizontal wires around the via using the space on the vertical layer. Because the detour happens on the lower layer, it will not affect the wire between terminals c and d on the upper layer.

By aligning a number of *via tunnels* in vertical direction, we obtain a *bank of via tunnels*, which is shown in Figure II.15(b). Suppose each *via tunnel* have k vias arranged in a horizontal line (in Figure II.15(b), $k = 3$), and we align L *via tunnels* into a bank. In the resulting bank, all the horizontal tracks are free to route, and only $k + 2$ vertical tracks are blocked. Note that there are a total of kL vias in the bank; without the bank of via tunnels, up to kL tracks could be



(a) A via tunnel in Y-architecture. (b) Two tunnels aligned together.

Figure II.16: Via tunnels and bank of via tunnels in Y-architecture.



(a) $k = 2$
overhead = 5

(b) $k = 3$
overhead = 7

(c) $k = 4$
overhead = 9

Figure II.17: Via tunnels with vias aligned in a line in Y-architecture.

blocked on each layer that the vias pass through. The use of *via tunnel banks* can thus significantly reduce the "via blocking effect".

We have designed similar *via tunnel* and *bank of via tunnels* for the Y-architecture.

- Figure II.16(a) shows the birds-eye view of a *via tunnel* design in the Y-architecture. In this example, we have three layers. From top to bottom, the routing direction is 60-degree, 120-degree and 0-degree in each layer, respectively. The circle in the center represents a through via. The space in the middle layer is used to detour wires around the via. We can achieve blockage-free routing on the top and bottom layers, and have four tracks blocked on the middle layer.

- Similar to the construction of *banks of via tunnels* in M-architecture, we align

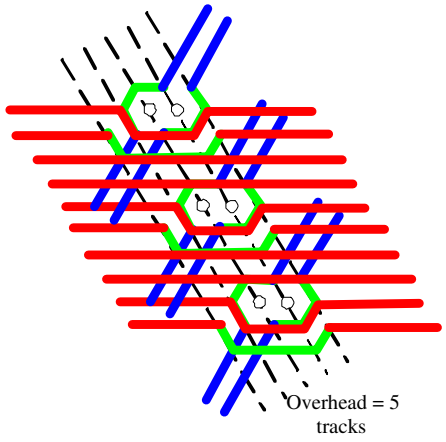


Figure II.18: Three tunnels with $k = 2$ aligned together

the *via tunnels* together to obtain a *bank of via tunnels* in the Y-architecture. Figure II.16(b) illustrates how two *via tunnels* shown in Figure II.16(a) are aligned along the 120-degree direction.

- In order to reduce the average track overhead, each via tunnel can have more than one vias in a line. Figure II.17 illustrates the construction of via tunnels with k ($k = 2, 3, 4$) vias aligned in a line. The figures show detour routing patterns on the middle layer for $k = 2, 3, 4$, respectively. Figure II.18 is an example of three via tunnels with $k = 2$ aligned along the 120-degree direction. From these examples, we can see that for via tunnel with k vias aligned in a line, the track overhead on the middle layer is $2k + 1$.

- Figure II.19 depicts a *bank of via tunnels* in the Y-architecture. Suppose the bottom m layers are used to perform intra-cell routing, and the top $n - m$ layers are used for distributing signals to the banks. Assume each *via tunnel* has k vias in a line, and there are L *via tunnels* in the bank. All the kL vias introduce only $2k + 1$ tracks of routing blockage on the 120-degree routing layers.

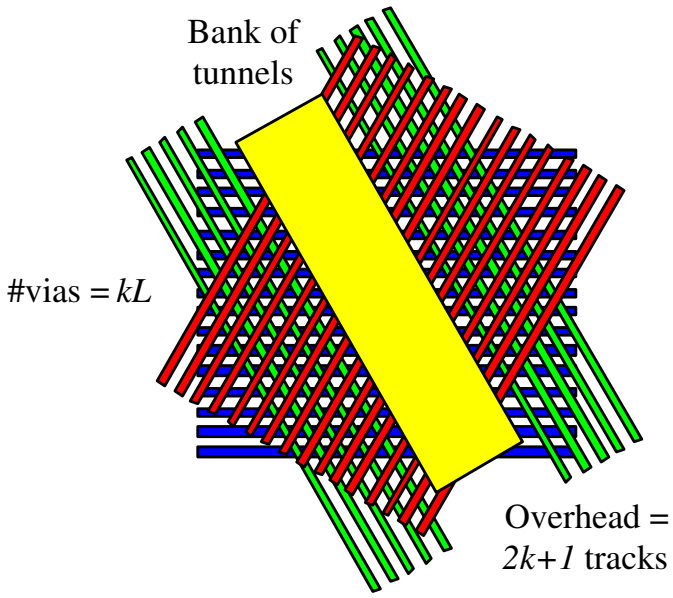


Figure II.19: Bank of via tunnels in Y-architecture.

II.B.7 Conclusions

In this paper, we have examined key issues concerning the potential use of Y-architecture for semiconductor ICs, including throughput analysis, estimates of wirelength savings, clock and power distribution methodology, wireability, and manufacturing. We have not discussed such issues as graphics engine changes, computational-geometric data structures, number and coordinate systems, calibration of parasitic extraction (especially capacitance extraction) models, etc. Such "mundane" issues are part of the necessary groundwork for the eventual deployment of the Y-architecture, and the subject of ongoing work in our group, but are beyond the scope of the present paper.

Further research directions include: (1) theoretical analysis and high-impact designs or codes to demonstrate Y-architecture advantages; (2) more accurate estimations of expected wirelength improvement which formalizes interactions between nets; and (3) interfaces to current library cells and new Y-specific library cells. Many parts of a commercially successful Y-architecture methodology remain open. The Y-architecture also has applications beyond the die, e.g., it may be

valuable on laminates used for multi-die integration, and on the buildup layers (e.g., BBUL [87]) that will replace traditional packages.

III

High Speed Clock Distribution in the Presence of Parameter Variations

III.A A Multiple-Level Network Approach for Clock Skew Minimization with Process Variations

III.A.1 Introduction

The clock distribution network design has been a great challenge in the state-of-the-art high performance chip designs. With tens or even hundreds of millions of transistors integrated, distributing the clock signal to the local areas all over the chip with near-zero skew becomes a very difficult task. Moreover, as the clock frequency climbs to giga hertz range and the interconnect delay dominates in deep sub-micron technology, the portion of the clock skew introduced by the process variations on the wire width and the clock buffers length can no longer be ignored. A robust clock distribution network that is less sensitive to the process variations is desired.

In microprocessor designs, the clock distribution networks can often be partitioned into two parts: the global and the local clock networks (Fig. III.1).

The global clock network distributes the clock signal from the clock source in the center of the chip to local regions. It usually has a symmetric structure. The local distribution networks deliver clock signals to numerous clocking elements in each local region. Their structures are often non-symmetric because the locations of registers are not necessarily regular. In this paper, we focus on the global clock networks.

A lot of works have been done in the past two decades to find the best structure for global clock distributions. Tree-based structures are widely used to achieve low clock skew and power consumption [1], because tree structures have the advantages of being easy to tune and simulate.

However, a mesh structure is more robust than a tree structure under process variations, since the mesh has more local connections that can smooth out the local delay variations and yield a smaller clock skew.

A recent trend is to use the hybrid structure of a mesh and a symmetric buffer tree for the global clock network. For example, the global clock distribution of Intel Pentium4^{TR} microprocessor consists three spines each driven by a balanced binary buffer tree [1]. The bottom-level spines can be deemed as a "one dimensional" mesh structure. Restel et *al.* proposed a two-level hybrid clock network. The top level is an H-tree, and the bottom level is a uniform mesh that connects all of the leaves of the top level H-tree. This clock network structure has been successfully applied to six designs[1], including the latest Power4 microprocessors [9][14]. The measurements from the real produced chips proved that this hybrid tree and mesh structure accomplishes low clock skew under process variations.

Su et *al.* [18] propose a two level clock network which makes a departure from the popular "mesh at the lowest level" structure. In their design, the top level is a zero-skew mesh that delivers the clock to the centers of four quadrants of the chip and the bottom level contains four zero-skew trees. Their method inspires us to ask at which level of the tree does the shortcut mesh works most efficiently in terms of skew reduction. The mixture of the tree structure and the mesh structure

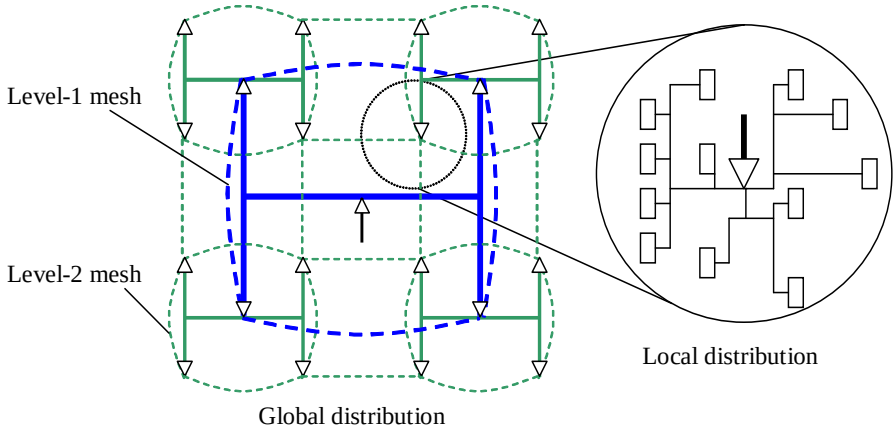


Figure III.1: global and local clock distribution networks

complicates the simulation problem. The simple Elmore delay model that fits the tree structure no longer applies to the mixed tree and mesh structures. A new model is needed to calculate the skew and so as to provide guidelines for the design of the hybrid networks.

In this paper, we address this problem. In order to keep the presentation easy to understand we focus on the hybrid clock network similar to the one proposed by [50], which consists a symmetric H-tree and a mesh connecting all of the bottom level leaves. Our method can also be applied to other hybrid tree and mesh structure, like the binary tree plus spine structure [1]. Based on our study on the skew reduction effect of mesh, we propose a multilevel network for global clock distribution. Figure III.1 shows a schematic example of our multilevel network. The dotted line represents the meshes which connect together all of the nodes at the same level in the original H-tree. In order to reduce the inductance of the shunt segments, we may use grounded shielding or differential pairs for mesh connections.

Our contributions in this paper include the following:

- We use a simplified *RC* circuit model to study the skew reduction effect of adding shunt connections between two leaf nodes of the clock tree. Based

on this model, we derive an analytical skew approximation, which fits the SPICE simulation very well. And further analysis and simulation suggests that the RC model remains valid for differential clock nets running at the frequency of less than 4GHz.

- We extend our skew approximation formula to the mesh network. We get the equivalent resistance factor of "one-dimensional" meshes and two-dimensional meshes both from the least square linear regression on SPICE simulation results and the analytical approximation on a distributed parameter circuit model. This analysis can be used to guide the design of the hybrid clock network.

- We propose a mixed multi-level mesh and tree structure for global clock distribution networks. We adopt mathematical programming technique to optimize the routing resource distribution of a mixed multi-level mesh and tree clock network. The optimized multi-level mesh/tree network produces a 30% skew reduction over the single-level mesh and tree network and is more robust in the presence of voltage fluctuation.

The rest of the paper is organized as follows: In Section II, we formulate the hybrid multi-level mesh/tree optimization problem. In Section III, we propose a simplified circuit model for hybrid mesh/tree networks and derive the analytical skew expression. SPICE simulation results show that our skew formula is very accurate for single branch. In section IV, we extend the skew formula to uniform meshes. Following that, we introduce our multi-level mesh optimization scheme in Section V, and in Section VI, we present the experimental results. In Section VII, we discuss the inductive effect and then, we conclude this paper in Section VIII.

III.A.2 Problem Formulation

In this section, we formulate the multi-level clok network design problem as a optimal sizing problem. We first introduce a simple process variations

model in subsection A, and then present the optimal sizing problem formulation in subsection B.

Process Variation Model

Semiconductor manufacturing variations occur when process parameters deviate from their ideal, as-designed values. Process variations have always been a key concern for manufacturability, process control, and circuit design. With rapid technology scaling, the importance of the impact of variations on the circuit design is further increasing. Variation can be categorized into temporal and spatial sources [3]. Temporal sources are time-varying and change depending on circuit activities. Spatial effects are depend on physical factors and impact the geometry of a structure and can lead to undesirable effects such as yield loss. In this paper, we mainly consider the variations on the geometrical parameters of interconnect and devices in a clock distribution network. Conventional circuit techniques typically represent the interconnect and device parameter variations as random variables. However, recent studies [11] have shown that strong spatial pattern dependencies exist, especially when considering interconnect variations in strong chemical mechanical polishing (CMP) processes. Therefore, the total variation can be separated into systematic and random components. [11] shows that considering the systematic variations is the key to reducing design uncertainty and maximizing circuit performance.

In this paper, we adopt a simple linear variation model to represent the systematic spatial variations on wire widths and transistor lengths. For any circuit element at location (x, y), the actual geometrical parameter $d = d_0 + k_x \cdot x + k_y \cdot y$, where d_0 is the nominal parameter and k_x and k_y are the horizontal, vertical variation coefficient, respectively. Without loss of generality, we assume that the origin of the coordinate $(0, 0)$ is located in the center of the chip, and k_x and k_y are both positive numbers. We set the maximum variations across the chip to be 10% of the ideal values [17].

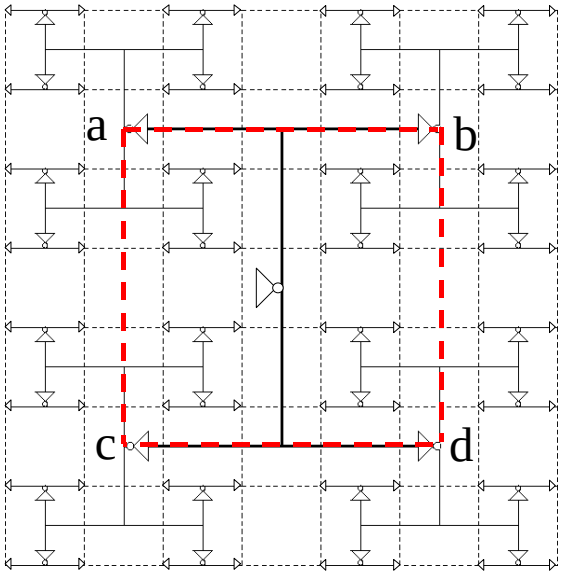


Figure III.2: An H-tree with A Two Level Mesh

Optimal Sizing of Multilevel Clock Network

Fig. (III.2) illustrates a hybrid clock distribution network. In the figure, there is a three level H-tree distributing the clock signal from the center of the chip to an 8 by 8 array of leaf buffers. In addition to the H-tree, there are two level of meshes both are drawn in dotted lines. On the top level, four wire segments connect four points, a,b,c,d, the outputs of four leaf buffers of the first level H-tree. On the bottom level, a 8 by 8 uniform mesh connect the output of all the leaf nodes of the third level H-tree.

With process variations [13], we can observe the clock skew between the nodes in the same level. Suppose that we are given the layout information and process variations model of a symmetric zero-skew clock tree (either an H-tree or a binary tree), we can obtain the clock skew between the nodes at the same level using Monte Carlo simulation [18] or variational circuit analysis[10]. We denote T_i to be the worst clock skew between any two nodes at the level i in the H-tree. Adding shunt segments between nodes at the same level in a tree is a common way to reduce skew and is widely accepted by industrial practice.

For example, in the symmetrical H-tree shown in Fig. III.2, all of the leaf nodes on level 3 are connected by an 8 by 8 mesh, which is drawn by dotted line in Figure 2. We can also connect the four level one nodes a, b, c, d, by a 2 by 2 mesh. When the wire width of the mesh is wide enough, the nodes at the same level are almost short-circuited and the skew between them can approach zero. On the other side, using too wide wire may waste too much routing resource. Hence degrade the routability. In addition, wide wires in the mesh can increase the clock slew because it increases the load capacitance of clock buffers. Consequently, the design of clock distribution networks must follow the total routing area budget. For the same amount of routing resources, adding them to the meshes at different levels may have different impacts on the clock skew. In this paper, we are interested in the optimal way to distribute the routing resources to the meshes on different levels such that the minimum skew is achieved at the leaf nodes with a given routing area budget. We formulate this problem as the following optimum balanced clock tree augmentation problem.

Optimal Balanced Clock Tree Augmentation Problem:

Given: An n level symmetric clock tree (wire width of segments in each level, buffer locations and buffer sizes);

The clock skew between nodes at the same level introduced by process variations

Input: The total routing area budget for all the meshes

Output: The optimal wire width w_i of shunt connections at level i for $i = 1, ..., n$, such that the clock skew is minimized.

Topology Constraints: Uniform mesh or spine for the shunt connections in each level

III.A.3 Skew Shunt Resistance Relations in A Simplified Circuit Model

We use a simplified circuit model shown in Fig.III.3 to study the skew-shunt resistance relations in a hybrid tree and mesh structure. In the model,

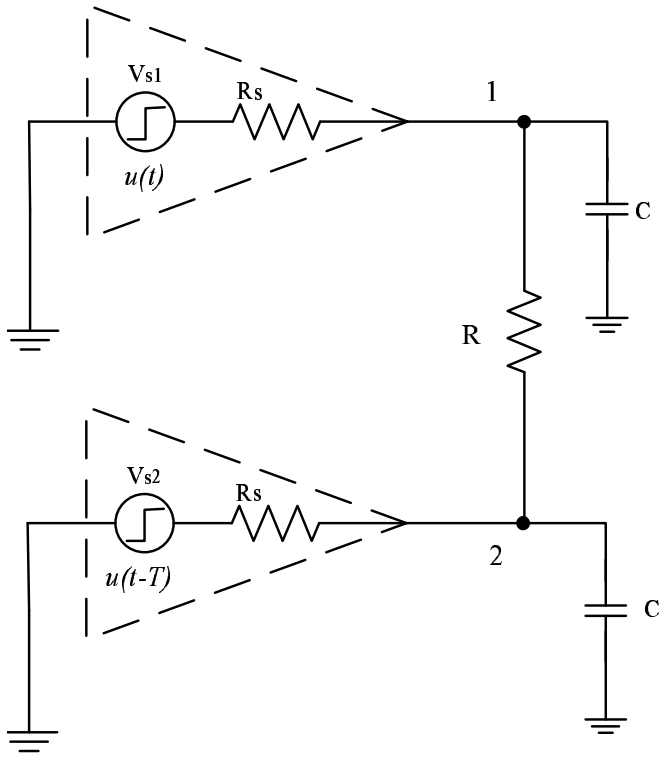


Figure III.3: simplified Circuit Model of RC Shunt

there are two leaf driver of a clock tree, s_1 and s_2. Each of them drives a load capacitance with value C, which is the summation of the sink capacitance and the wire capacitance. R_s is the driving resistance of a clock buffer. A mesh segment with resistance R connects two tree branches to reduce the skew between node n_1 and node n_2. We assume that V_{s_1} and V_{s_2} are step functions and V_{s_2} arrives behind a time difference T after V_{s_1}. Note that many factors can contribute to this timing difference T between node s_1 and s_2. For example, the skew effect due to the distribution of the upstream network, variations of R_s, and C, and supply voltage variations at the clock buffers. For the simplicity of modeling, we summarize all these effects into the timing difference T between the input step functions. We assume that T is given, and seek for the expression of arrival time difference, ΔT, between the output signals of two clock buffers.

The following differential equation describes the time domain response of

the circuit shown in Fig.III.3.

$$\begin{bmatrix} \frac{dV_1}{dt} \\ \frac{dV_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC} - \frac{1}{R_sC} & \frac{1}{RC} \\ \frac{1}{RC} & -\frac{1}{RC} - \frac{1}{R_sC} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{R_sC} & 0 \\ 0 & \frac{1}{R_sC} \end{bmatrix} \begin{bmatrix} V_{s1} \\ V_{s2} \end{bmatrix} \tag{III.1}$$

Assuming the input skew is small, i.e. $T \ll R_sC$, we derive a simple skew expression from equation (III.1)

$$\Delta T = T \cdot e^{2ln2\frac{R_s}{R}} \tag{III.2}$$

For detailed derivation, the readers are referred to the Appendix A.

Surprisingly, in the obtained skew model (III.2), skew ΔT is independent with the value of load capacitance, and only determined by R_s/R, the ratio between driving resistance and the shunt resistance. This simple relation enables us to easily estimate the skew on a hybrid tree and mesh clock network. We verify this relation by SPICE simulations.

We change the value of R_s from 1200Ω to 300Ω, R from 100Ω to 1000Ω. [1] We set the range of C from 10fF to 200fF and simulate the circuit with different parameters using SPICE. We show the relations between skew and R_s, C, R when T is 5 ps. Figure III.4 depicts the effect of R_s and C. This result shows that the skew decreases proportionally to the exponential of R_s/R and that C barely affects the skew in the range we are interested in.

III.A.4 Clock Skew on Meshes

In last subsection, we derive the skew and shunt resistance relation (III.2). The skew reduction is inverse proportional to the exponential of the driving resistance/shunt resistance ratio. We conjecture that when multiple clock drivers with input skew are connected by a uniform mesh, the skew reduction effect has the

[1] 1200Ω and 300Ω are typical driving resistance of minimum size buffer and 4 times wide buffer, respectively. 400Ω is the typical value of resistance of 1mm wire with minimum wire width in 70nm technology.

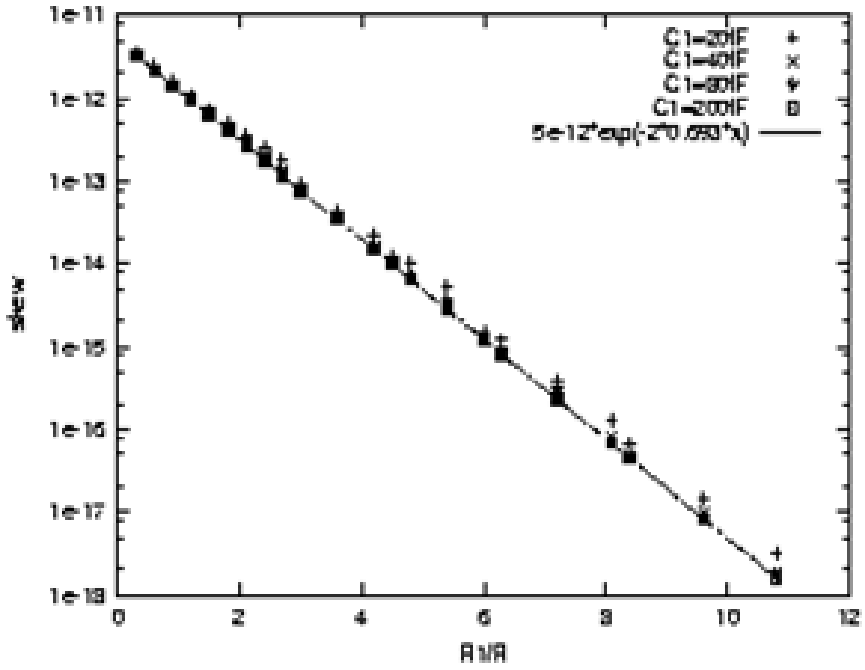


Figure III.4: Skew R_s/R relation in the simplified RC circuit model

similar form $\Delta T = Te^{-kR_s/R}$, where, R is the resistance of each shunt segment, T is the maximum input skew among all the clock drivers, and ΔT is the maximum output skew. Comparing with equation (III.2), the only difference is that there is an equivalent resistance factor k to capture the different locking capabilities of different meshes.

The value of the equivalent resistance ratio, k, depends both on the dimension of the mesh and the pattern of input skews of the clock drivers. When the input skews are independent random numbers, the mesh can smooth out the output skew most effectively. In this scenario, we have a large k value. However, if the input skew gradually changes according to the geometrical location of the drivers, the mesh has less capability to reduce the global skew, because the fastest driver and the slowest drivers are far away from each other. In this section, we assume the worse scenario where the input delay surface is linearly tilted from one end of the mesh to the other end of the mesh. The input arrival time of each clock

driver is decided by a linear function on its location.

There are two ways to derive the equivalent resistance ratio, k. The first one is to conduct a set of simulations, sweeping different values of driving resistance R_s and shunt resistance R. Then, use linear regression on the skew value obtained through simulations to get the k value for different meshes. An alternative way is to derive the k value through analytical expressions. In this section, we derive the asymptotic bound of the k values for both on-dimensional meshes and two-dimensional meshes.

Clock Skew On a Series of Shunts

In this subsection, we derive the skew reduction effect of a "one-dimensional" mesh. In our model, a series of clock drivers with input skews are connected by a wire. We assume the worst case scenario, the input delay is linearly increasing from one end to the other end. The input signals are square waves with 50% duty cycle and clock period T_p. Without loss of generality, we assume the clock driver at the the left end has phase shift $T_0 = 0$, the clock driver at the right end has phase shift $T_n = T$. The phase shift of the i^{th} driver, $T_i = \frac{i}{n}T$.

Fig. III.5 depicts the circuit model we use to calculate the equivalent resistance ration, k, for a series of shunts. Fig. III.5(a) is the discrete RC circuit model when only the k^{th} driver takes effect. Under that condition, the driving resistance of other drivers can be viewed as the leakage conductance to the ground. When the number of drivers is large enough, we can use a continuous R,G,C transmission line model (Fig. III.5)(b) to model the circuit, where the ground conductance G is the inverse of the driving resistance, R_s:

$$G = 1/R_s \tag{III.3}$$

For any sinusoidal wave with unit magnitude incident to this uniform R, G, C transmission line, the forwarding wave can be described as:

$$v(z,t) = e^{-\alpha z} Sin(\omega t + \beta z) \tag{III.4}$$

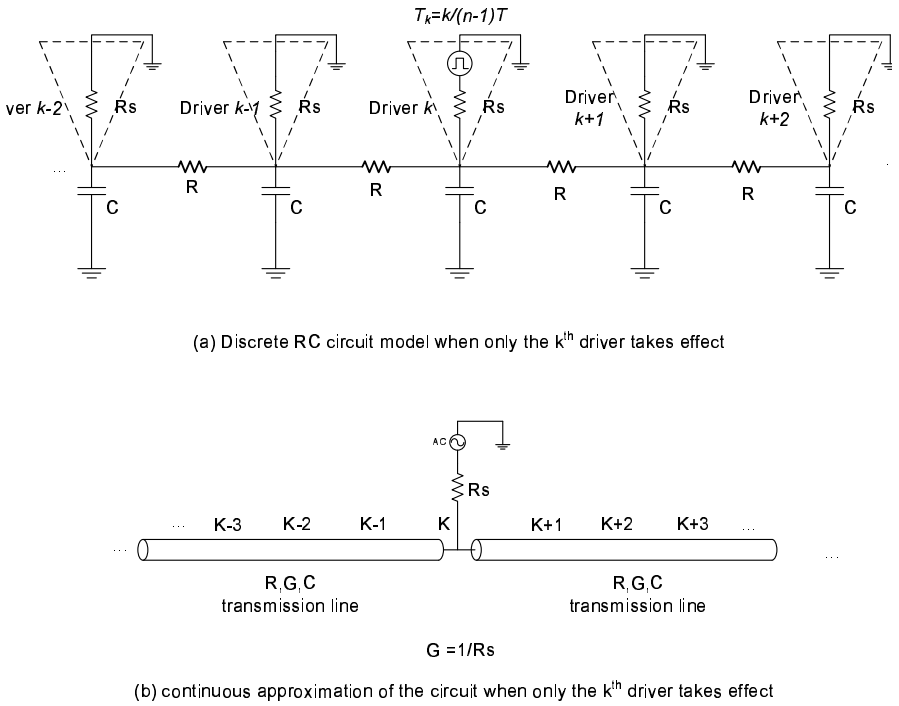


Figure III.5: RGC transmission line approximation when only one driver takes effect

where, $\alpha + j\beta = \gamma$ is the propagation function of the transmission line. The propagation function for an R, G, C transmission line can be written as

$$\gamma = \alpha + j\beta = \sqrt{R(G + j\omega C)} \tag{III.5}$$

Through algebraic operations, we can decompose the propagation function into the real part, α, which corresponds to the attenuation of the wave magnitude, and the imaginary part, β, which corresponds to the phase velocity of the wave propagation.

$$\alpha = \sqrt{\frac{1}{2}}\sqrt{\frac{R}{R_s}}\sqrt{1 + \sqrt{1 + \omega^2 R_s^2 C^2}} \tag{III.6}$$

$$\beta = \sqrt{\frac{1}{2}}\sqrt{\frac{R}{R_s}}\sqrt{-1 + \sqrt{1 + \omega^2 R_s^2 C^2}} \tag{III.7}$$

Through Fourier expansion, the input square wave can be expressed as

the sum of a series of harmonics.

$$u_i(t) = \frac{4}{\pi} \sum_{k=1,3,5...}^{\infty} \frac{1}{k} Sin(k\frac{2\pi}{T_p}t) \tag{III.8}$$

We take the first seven harmonics to approximate the input square waves. This approximation gives us very high accuracy, because the first seven harmonics constitute 95% energy of the square wave.

For any input sinusoidal wave with radial frequency ω, the characteristic impedance of the R,G,C transmission line is

$$Z_0 = \sqrt{\frac{R}{G+j\omega C}} = \sqrt{\frac{RR_s}{2(1+\omega^2 R_s^2 C^2)}}[\sqrt{1+\sqrt{1+\omega^2 R_s^2 C^2}} + j\sqrt{-1+\sqrt{1+\omega^2 R_s^2 C^2}}] \tag{III.9}$$

Denote $v_{i,j,k}$ as the voltage response at node i, assuming only the k^{th} harmonic of the j^{th} source takes effect. We have following expression for $v_{i,i,k}$, the incident wave at the node i, caused by the k^{th} harmonic of the input at the k^{th} driver.

If the driver is at any of the two ends of the transmission line, $i = 0$ or $i = n$

$$v_{i,i,k} = \frac{Z_0}{Z_0 + R_s} sin(k\omega t) \tag{III.10}$$

If the driver is in the middle of the transmission line, $i \neq 1, n$

$$v_{i,i,k} = \frac{Z_0}{Z_0 + 2R_s} sin(k\omega t) \tag{III.11}$$

From equations (III.6), (III.7), (III.10), and (III.11), we can get the voltage response at node i, caused by the k^{th} harmonic of the input signal of the j^{th} driver

$$v_{i,j,k} = v_{j,j,k} e^{\alpha + j\beta} \tag{III.12}$$

The output voltage at the node i is

$$v_i = \sum_{j=1}^{n} \sum_{k}^{8} v_{i,j,k} \tag{III.13}$$

With following two assumptions :

Table III.1: The calculated and simulated k-values for one-dimensional meshes

n	2	3	4	5	6	7	8
k_{calc}	1.14	0.462	0.474	0.263	0.193	0.153	0.113
k_{sim}	1.00	0.401	0.364	0.227	0.186	0.149	0.103
n	9	10	11	12	13	14	15
k_{calc}	0.0917	0.0855	0.0742	0.0646	0.0577	0.0419	0.0381
k_{sim}	0.0903	0.0847	0.0735	0.0606	0.0563	0.0412	0.0361

1. The input skew is smaller than the clock slew: $T \ll R_s C$

2. The clock period is much larger than the clock skew: $\frac{2\pi}{\omega} \ll R_s C$

The clock arrival time at node i can be approximated by

$$t_i = \frac{\sum_{j=0}^{n} \sum_{k=1}^{7} \dot{v}_{i,j,k}(t = ln2R_sC)(\frac{i}{n}T)}{\frac{ln2}{2}} \tag{III.14}$$

The global skew is

$$\Delta T = t_n - t_0 \tag{III.15}$$

We calculate the k value as:

$$k = \frac{R}{R_s} ln \frac{T}{\Delta T} \tag{III.16}$$

We list the calculated and simulated k-values in TABLE III.1. The first row of the table lists the number of drivers driving the wire. The second row of the table are the calculated values and the third row shows the k-values obtained through SPICE simulation and least square linear regressions. When the number of drivers is less than 5, the calculated values have relatively large errors comparing with the simulation results. When the number of drivers is large, the k-value obtained through simulation approaches the calculated asymptotic bounds.

Clock Skew on a Two-Dimensional Mesh

The clock skew on a two-dimensional mesh has similar behavior to that on a one-dimensional mesh. We conduct following SPICE simulation to justify our conjecture.

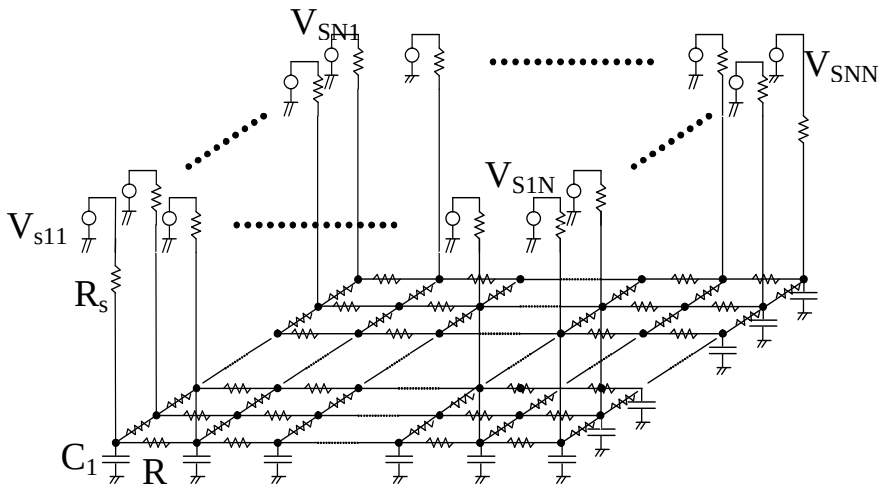


Figure III.6: Circuit model used to calculate the skew on a mesh

Assuming an n by n resistive mesh, the resistance of each edge is R. Under each crossing point, there is a load capacitance C. The mesh is driven at each crossing point by an inverter with driving resistance R_s (Fig. III.6). The input signal of inverter at the i^{th} column and the j^{th} row is $T_{i,j}$. Assume that the maximum input skew among all the clock drivers is T. We consider following two different cases.

- Case A: $T_{i,j} = \frac{i+j}{2(n-1)}T$
- Case B: $T_{i,j} = \frac{i}{n-1}T$

In Case A, the input delay of each clock driver is proportional to its Manhattan distance to the lower left corner of the mesh. In Case B, the clock drivers on the same column have same input delay. The input delay of each clock driver is proportional to its distance to the left edge of the mesh. In this case, the k value is equivalent to the "one-dimensional" mesh case with n shunt resistors in a series.

We use SPICE simulation and least square linear regression to get the different k values of above two different input skew patterns. TABLE III.2 lists the results for those two cases. The error between two different input skew patterns

Table III.2: k-value of clock meshes for different input skew patterns

n	2	4	6	8	16
k (Case A)	1.167	0.373	0.184	0.107	0.030
k (Case B)	1.00	0.364	0.177	0.103	0.031

is at most 17%, which occurs when the mesh is just a two by two mesh. When the number of mesh columns and rows increases to 16, the error is only 3%. This phenomena enables us to apply the analytical bound of the equivalent resistance ratio of one-dimensional meshes to the two-dimensional meshes.

III.A.5 Optimization of Multi-level Clock Networks

For a uniform mesh, the resistance of a wire segment, R, is inverse proportional to its width, w. We rewrite the skew expression $\Delta T = T \cdot e^{-kR_s/R}$ as $\Delta T = T \cdot e^{-k'R_s w}$. Where, k is a constant determined by the number of columns and rows of the mesh, and k' is another constant determined by the wire geometry and material, (i.e. wire height, length, and metal conductivity). We formulate the Optimum Balanced Clock Tree Augmentation Problem as the following nonlinear programming problem.

MLMOP(Multi-Level Mesh Optimization Problem):

$$\text{Min:} \quad \Delta T = (\cdots((T_1 \cdot e^{-k_1 w_1} + T_2) \cdot e^{-k_2 w_2} + T_3) \cdots + T_n) \cdot e^{-k_n w_n} \tag{III.17}$$

$$\text{S.t.:} \quad \sum_{i=1}^{n} l_i w_i = A \tag{III.18}$$

Where, the constant A is the total routing area budget for all the meshes. l_i is the total wire length of the i^{th} level mesh. T_i is the initial skew between level $i-1$ nodes and level i nodes on the H-tree. A, l_i, and T_i are all constants decided by the dimension of the chip and the process variation model. The wire width of the i^{th} level mesh, w_i, are variables.

In the above nonlinear program, the cost function (III.17) is the skew at the bottom level leaves, and the constraint (III.18) is the budget of total routing

area, which can be translated to the power consumption overhead. By solving this non-linear program for any given total routing area of multi-level meshes, we can find the minimum skew can be achieved by the multi-level mesh as well as the best way to assign routing resources to meshes at different levels. From the property of exponential functions, the equation (III.17) is a convex function. And because the constraint (III.18) is also a convex set, we have following theorem about the nonlinear program MLMOP.

Theorem 2 *The local optimal solution of the nonlinear program MLMOP is also the global optimum.*

Because of the convex property of the skew function (III.17), many optimization techniques (e.g. many gradient methods and line search methods [2]) can be used to find the best w_i assignment such that the skew is minimized. In our experiments, we use the line search algorithm provided in the *Optimization Toolkit* of Matlab to solve this nonlinear program.

III.A.6 Experimental Results

We apply our method to a clock distribution network design in 70 nm technology. In our experiments setting, the chip size is 24 mm x 24 mm. We synthesize a 4-level symmetric H-tree using the *P-tree* algorithm described in [5]. The first level mesh connects 4 leaf nodes of the first level H-tree. The length of each segment is 12 mm, thus total wire length of the first level mesh is 48 mm. In the second level mesh, 16 leaf nodes of the second-level H-tree are connected by a 4 by 4 mesh. The length of each wire segment is 6 mm, and total wire length is 144 mm. The third level mesh is an 8 by 8 mesh. The wire segment length and the total wire length is 3 mm and 336 mm, respectively. The forth (bottom) level mesh has a dimension of 16 by 16. The length of each wire segment and the total wire length are 1.5 mm and 720 mm, respectively.

TABLE III.3 presents the optimized wire widths for each level mesh.

Table III.3: Optimal wire sizing of a 4-level clock network

total area	opt. wire width of each level mesh (μm)			
($10^3 \mu m^2$)	1st	2nd	3rd	4th
28.04	0.29	0.16	0.00	0.00
46.08	0.29	0.30	0.00	0.00
115.2	0.29	0.30	0.20	0.00
345.6	0.29	0.30	0.41	0.22
576.0	0.29	0.31	0.41	0.54

Table III.4: Skew comparison between single level and mutli-level meshes

total area	skew		
(μm^2)	S-Mesh (ps)	M-Mesh (ps)	imprv (%)
0.00	29.2	29.2	0
28.04	27.9	26.0	6.8
46.08	27.1	24.5	9.6
115.2	24.2	19.8	18.2
345.6	17.0	12.4	26.8
576.0	12.4	8.72	30.5

The first column shows the total routing area of meshes. The second to the fifth column shows the optimized wire width of each level mesh. The result suggests that when the routing resource budget is tight, we should put wire resources into a higher (top) level mesh until that level saturates. When routing resource gradually increases, we should put more wiring resources into lower (bottom) level meshes.

TABLE III.4 is the skew comparison between optimized multi-level meshes and single level meshes. We obtain skews through SPICE simulation. In a single level mesh, we put all of the resources into the bottom level mesh. From this simulation, the more wire resource we used for the mesh, the more the optimized mesh

Table III.5: Skew comparison with supply voltage variations

total area	multi-level mesh (ps)		single-level mesh(ps)	
($10^3 \mu m^2$)	Ave	Worst	Ave	Worst
115.2	8.38	11.4	8.26	14.3
230.2	2.71	4.42	6.18	11.1
345.6	1.89	3.33	4.83	8.73
460.8	1.45	2.48	3.88	6.96
576.0	1.16	2.02	3.18	5.64

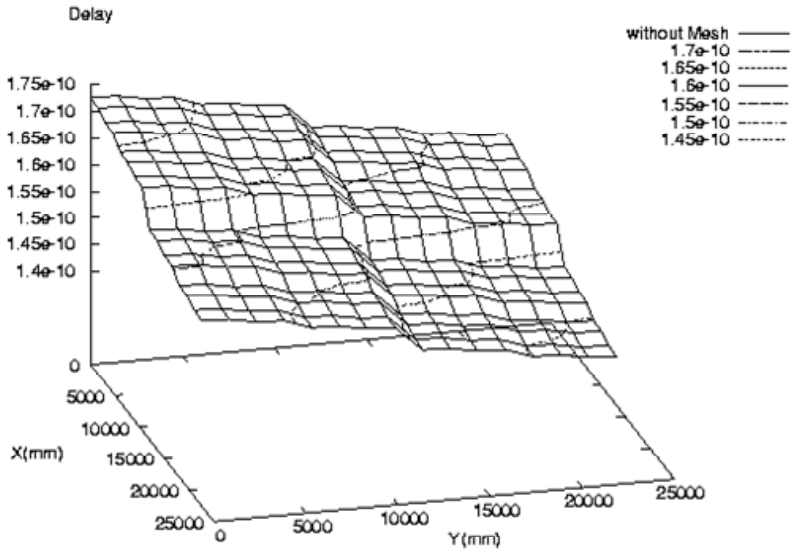


Figure III.7: Delay of leaf nodes on an H-tree without mesh

reduces skew comparing with single level mesh. When we used only $2.804E4\mu m^2$ total wiring area, improvement from optimization is only 2.2%, however when we use $5.76E5\mu m^2$ resources, the optimized mesh can reduce skew by 30%.

Fig. III.7 and Fig. III.8 demonstrate the effect of the mesh on clock skew. In these two figures, the crossing points mean that the sink node at bottom level H-tree, x- and y-axis indicate the position in a chip and z-axis is the delay of sink nodes. Fig. III.7 shows the delay map for a H-tree without a mesh, and Fig. III.8 demonstrate the case of a multi-level network. The worst local skew and global skew in Fig. III.7 are 5.9ps and 29.2ps respectively. By adopting a multilevel network, these values decrease to 3.1ps and 19.8ps respectively.

We test the robustness of the multi-level mesh against voltage fluctuation in a set of experiments. In our experiments, we perturb the supply voltage of each clock buffer randomly by 10%. For each pair of multi-level mesh and single level mesh with same total routing area, we perform 10 simulations with different random seeds. TABLE III.5 shows the average and the worst skew of these 10

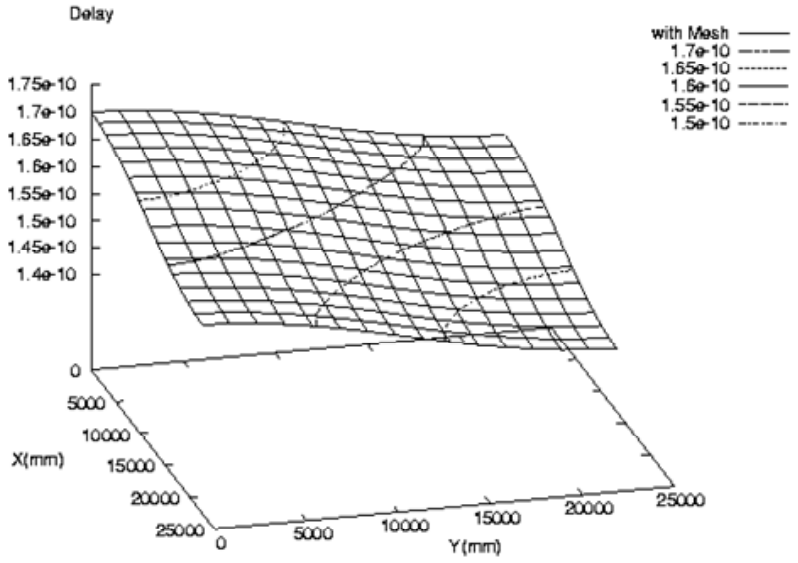


Figure III.8: Delay of leaf nodes on an H-tree with multilevel meshes

cases. Note that in the experiments, in order to focus on the voltage fluctuation effect, we ignore the process variations. For the multi-level mesh with total area 5, the average and worst clock skew are 1.16ps and 2.02ps, respectively, which are 60% less than those produced by a single level mesh.

III.A.7 Discussions on Inductive Effect

In our previous analysis and experiments, we ignore the inductive effect of interconnect. When the clock frequency keeps climbing, the inductance's effect becomes more and more important. However, a lot of techniques can be used to control the parasitic inductance of clock interconnect, such as grounded shielding and using differential signals. In [24], a set of rules have been developed to help us deciding under which conditions the inductive effect can be ignored. According to [24], the error between RC and RLC representations will not exceed 15% for a single wire, if i) $C_L \gg C$, ii) $R/Z_0 > 2$, and iii) $R_1 > nZ_0$, where n is a constant with value between 0.5 and 1, C_L is the loading at the far end of the line, C is the

wire capacitance, and Z_0 is the characteristic impendence of the wire.

On the top level of our proposed multiple level mesh, the load capacitance C_L has value of 149.4fF, which is much larger than 14.3fF, the wire capacitance C. For a pair of 1.2cm copper differential wires with minimal wire spacing on metal layer 10, the inductance is 2.7nH [8]. At the frequency of 2GHz, with the clock slew of 50 ps, the characteristic impendence of the differential pair is 139Ω, which is much smaller than 5130Ω, the wire resistance, and also smaller than 367Ω, the driving resistance.

We conduct a SPICE simulation for a multiple level clock network using both RC and RLC circuit models. In our experiments, we use FastCap and FastHenry to extract the values of parasitic capacitance and inductance. When extracting the frequency dependant resistance and inductance values, we assume the return path is parallel to the wire with a 1um separation. At 2GHz, the error on maximum skew between RC and RLC circuit is less than 1%.

III.A.8 Conclusion and Future Directions

We demonstrated the effect of the mesh network to clock skew. From the result of the simplified circuit, the skew decreases proportionally to the exponential of $-R_s/R$. This analytical relation can be used to guide the design of hybrid mesh/tree clock networks. We propose to use a hybrid multi-level mesh/tree structure to reduce the clock skew. By solving a very simple non-linear programming, we can get the optimum resource distribution among the meshes in different levels. Our experiments show that by adding an 16 by 16 mesh at the bottom level leaves of an H-tree, the clock skew can be reduced from 29.2ps to 12.4ps and the optimized hybrid multi-level mesh and tree structure produces a clock skew of 8.72ps, which is 30% less than the single level mesh. The experiments also demonstrate that the optimized hybrid multi-level mesh and tree structure is much more robust than a single-level mesh and tree structure in the presence of voltage variations. Some interesting future research directions include:

- Theoretical analysis of clock signal propagation on a uniform mesh
- The behavior of RLC mesh in the multigiga hertz range
- The use of a non-uniform mesh to further reduce the clock skew

III.B PVT Variations Aware Clock Tree Synthesis in the Presence of Routing Obstacles

This paper describes a clock tree synthesis methodology for high performance ASICs. The main goal is to produce process, voltage, and temperature (PVT) variations tolerant clock distribution network in the presence of complex rectilinear routing obstacles. We introduce three key ideas. First, we note that not all data paths between registers have same sensitivity to the clock uncertainty. [53] The proposed methodology respects this difference, and let those more sensitive to the clock uncertainties share longer common path in the clock tree. Second, we use extended Delaunay triangular mesh to represent the physical proximity of clock sinks in the presence of routing obstacles. Based on the physical proximity information and timing constraint information, we use recursive graph partitioning to generate the initial clock tree topology. Third, we adopt multi-level optimization techniques to refine the clock tree topology and physical embedding. [36] The topology and embedding is optimized with accurate timing and wire length estimation. Experimental results show significant improvements on PVT variation tolerance with little wirelength overhead.

III.B.1 Introduction

Motivations

With rapidly increasing clock frequency, the clock uncertainties introduced by process, voltage, and temperature (PVT) variations consume significant portion of a clock cycle time and consequently decrease the circuit performance [35].

Designing a PVT variation tolerant clock distribution network becomes a vital part of high performance digital circuits. Recently, non-tree clock distribution topologies [47] [49] have been proposed to reduce the clock uncertainty by adding shunt connections. Some of them have been applied to the high performance processor designs [15][19][1]. However, for the automated ASIC design flow, the tree structure is still favorable for following two reasons. First, the tree structure is easier to analyze and can be integrated into current static timing analysis flow. Second, tree consumes less routing area and hence has lower power consumption and causes less routability problems.

For clock tree synthesis and routing algorithms, one practical challenge is the existence of routing obstacles. The state-of-the-art SoCs usually consist of a number of memory blocks and IP macros. Some of these blocks form the routing obstacles for clock wires. If the clock topology generation algorithm does not take the routing obstacles into account, resulted clock tree may have a lot of undesired wiring detours, which may cause severe skew and routability problemes. In this paper we present an automated clock tree synthesis tool for high performance digital circuits. The tool handles rectilinear routing obstacles and produces PVT variation tolerant clock tree with short total wire length.

Previous Works

As a classic CAD problem, clock tree synthesis has received intensive research efforts in the past twenty years. The proposed clock tree topology generation algorithms can be roughly classified into three categories: top-down partitioning, bottom up merging, and iterative searching. The main objectives are balancing the load, minimizing the total wire length and delay.

The median and mean method (MMM) proposed by Jackson et *al.* [11] recursively partition the clock sinks on a plane according to the locations. This method produces a well balanced topology. However, it does not consider routing obstacles.

The geometric matching [12] and greedy-DME [6], uses bottom -up matching to construct the tree topology. These methods depend on the dynamic nearest neighbor queries, which is computationally expensive in the presence of routing obstacles. In a Manhattan plane without obstacles, the shortest path query only takes O(1) time, while with obstacles, the fastest algorithms have a time complexity of at least $O(m \log m)$, where m is the number of corner nodes of all obstacles.

Ellis et *al.*[7] and Chou et *al.* [5] both used simulated annealing to search for the optimal tree topology. Multiple objectives are optimized simultaneously in the simulated annealing framework.

Recently, several works target at the variations aware clock tree synthesis. Velenis et *al.* [21] first noticed that not all datapath have the same sensitivity to the clock uncertainty, and use a sequential merging scheme to construct the clock tree topology. Their method does not use any physical proximity information. It may results in excessively large total wire length, and the tree topology may be very unbalanced.

In [10], Hu et *al.* extended the DME [2] algorithm to accommodate the permissible clock uncertainty constraints for a given clock tree topology. This method can significantly reduce the timing violations caused by the process variations on the interconnect. However, since the tree topology is generated by the non-variations-aware DME algorithm, it could not reduce the clock uncertainties caused by the voltage variations on the clock buffers, which is believed to be one main cause of clock uncertainties. In [3], the clock sinks are partitioned into groups, the algorithm reduces the wirelength by only minimizing intra-group skew values.

Our Contributions

As stated in [52], only small portions of the datapaths in synchronous circuits are most sensitive to the clock uncertainties. For those registers, we ought to put them topologically close to each other in the clock tree. At the same time, in order to minimize the total wire length, we also need to consider the physical

proximity between clock sinks when we construct the clock tree. We extend the basic Delaunay triangulization by adding virtual nodes to the boundary of the obstacles. The extended Delaunay triangular mesh provides a graph representation of the spatial relation of the clock sinks. We combine the Delaunay triangular mesh with the clock uncertainty constraint graph and use graph partitioning to balance the PVT variation tolerance and total wire length cost. We use multi-level optimization to further refine the clock tree topology with actual physical embedding information. We utilize the accurate wirelength and delay estimations based on actual physical embedding to guide the search of optimal topology. The main contributions of this work are:

- We explicitly address the requirement of both clock uncertainties and wire-length minimization in the clock tree topology generation. By considering both the spatial and temporal relations between clock sinks, we reduce the number of timing violations by 88% with only 1.5% of wire length increasing.
- We use extended Delaunay triangular mesh to represent the physical proximity information in the presence of routing obstacles.
- We adopt a multi-level optimization technique to simultaneously refine the clock tree topology and embedding. With multi-level optimization technique, we are able to synthesis clock tree with 200K flip-flops, 1.6 million timing constraints, and 200 routing obstacles within 6 hours.

The rest of this paper is organized as follows. In Section II, we formulate the PVT variations aware clock tree synthesis problem. We then describe the flow of the methodology in Section III. In Section IV, we present the initial clock tree topology generation by recursive graph partitioning. After that, we introduce the multilevel optimization framework for clock tree topology refinement and physical embedding. We show experimental results in Section VI. Finally, we conclude the paper in Section VII.

III.B.2 Problem Statement

We formulate the PVT variation aware clock tree synthesis problem in this section. In subsection A, we clarify the definitions of skew and clock uncertainty, and discuss their effect on circuit behavior. In subsection B, we present the problem formulation.

Clock Uncertainties and CRPR

We make distinction between clock skew and clock uncertainties. In this paper, we refer clock delay to the signal propagation delay from the clock source to the clock input pins. For a pair of registers, A and B, the difference of clock delay t_a and t_b can be decomposed into two parts, the deterministic part and the probabilistic part. We call the deterministic part $t_a - t_b$ the skew, which is due to the designed mismatch of delay and can be calculated using nominal design values. We call the probabilistic part, $\Delta_{a,b}$, the clock uncertainty, which is caused by the PVT variations. Figure 1 shows a schematic of a datapath in sequential circuits. Assuming the combinational logic has maximal delay Dmax and minimal delay Dmin, we have following two timing constraints.

$$t_a + D_{max} + t_{setup} - t_b + \Delta_{a,b} \leq t_p \tag{III.19}$$

$$t_a + D_{min} - t_b - \Delta_{a,b} \geq t_{hold} \tag{III.20}$$

According to above two constraints, the clock skew $t_a - t_b$, may be useful and can be introduced intentionally [8]. On the other hand, the clock uncertainties $\Delta_{a,b}$ are always harmful to the performance and reliability of the circuits. Traditionally, the designers use a certain portion of the clock delay, e.g. 15% of the insertion delay, as a safe margin of the clock uncertainty estimation. However, this approach overestimates the clock uncertainty. From Figure III.B.2, we see that the delay variations on the common path in the clock tree do not contribute to the clock uncertainty between two registers. Only the variations on the distinct paths make the contribution. The state-of-art static timing analyzer[22] has already

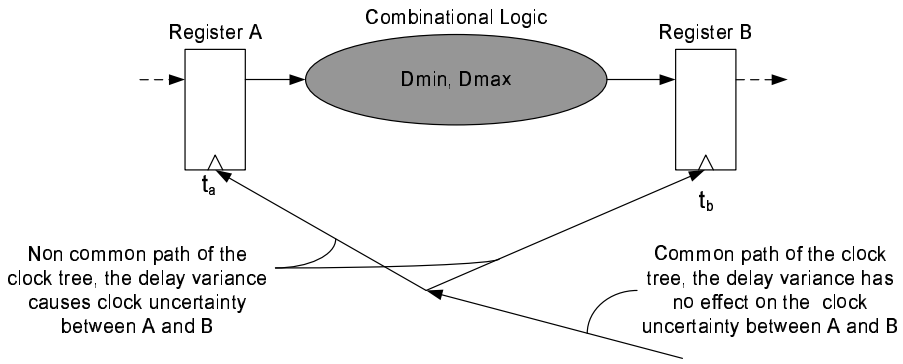


Figure III.9: Clock Uncertainties and Timing Constraints in Sequential Circuits

implemented the clock reconvergence common path removal (CRPR) algorithm, which can exclude the pessimism introduced by considering the variations on the common path. In this paper, we use a constant portion of the delay from the nearest common ancestor to the clock sink as the estimation of the clock uncertainty between two registers. We call this constant as the clock uncertainty coefficient, k.

PVT Variation Aware Clock Tree Synthesis Problem

Let $S = \{s_i = (x_i, y_i)\}$ denote a set of clock sinks on a Manhattan plane and s_0 the clock source. Each clock sink s_i has a capacitive load C_i. The routing blockages B are a set of rectangles on the plane. And $u_{i,j}$ is the maximum permissible clock uncertainty between register i and register j. Clock tree $T(S)$ is a tree rooted at s_0 and spanning on the union set of Steiner points S' and clock sinks S. We formulate the PVT variations aware clock tree synthesis problem as follows.

PVT Variations Aware Clock Tree Synthesis Problem: *Given a set of clock sinks S=(xi,yi,) on a Manhattan plane, a set of rectangular routing blockage B, the permissible clock uncertainty between clock sinks* $u_{i,j}$*, the clock uncertainty coefficient k, construct a buffered clock tree T(S), such that the total wirelength is minimized while all the permissible clock uncertainty constraints are*

met.

An alternative way to formulate this problem is to maximize minimum slackness for given total wire length or power budget. This formulation is a dual of our formulation, and the proposed method can solve both problems.

III.B.3 Overall flow of the methodology

Figure 3 shows the pseudo code of our PVT variations aware clock tree synthesis algorithm. The algorithm inputs the clock sink distributions, routing obstacles, and maximal pairwise clock uncertainty constraints. it proceeds mainly in three steps. First, it extracts the spatial relation and temporal relation between clock sinks using extended Delaunay triangulization, and generate the initial tree topology by recursive partitioning. The tradeoff between wire length and PVT variation tolerance is controlled by the net weighting. Then, a multi-level optimization scheme refines the tree topology and routing with accurate physical embedding information. Finally, we tune the buffer size and wire size to further improve the solution quality.

III.B.4 Initial tree topology generation

In this section, we describe our method for initial clock tree topology generation. In subsection A, we first introduce the extended Delaunay triangulization and prove several useful properties of it. Then, we describe our initial topology generation algorithm in subsection B.

Extended Delaunay Triangulization

Delaunay triangulization and its dual format, Voronoi diagram, have been used in several clock tree synthesis algorithms for simple representation of near neighbor information of objects in a 2D space [7]. However, in large ASIC designs, the existence of various routing blockages distorted the original distance matrix and makes the distance computation much more expensive.

Algorithm: Variation Aware Clock Tree Synthesis

Input: Clock sinks set S
Routing obstacles set B
Clock uncertainty constraint graph Gc.

Output: clock tree T

Procedure:

1. Generate extended Delaunay triangulization Gs = (S, Es), and assign edge weights
2. Estimate the minimal size of the bottom level cluster, s_min
3. Construct the uncertainty constraint graph Gt = (S, Et), and assign edge weights
4. while (cluster size > s_min) do recursive partitioning on G = (S, Et + Eu)
5. While not converged
   {
   5.a For (i = 1 to n) do
      Low temperature simulated annealing to optimize the permutation of level *i* subtrees;
   5.b For (i = n to 1) do
      Low temperature simulated annealing to optimize the permutation of level *i* subtrees
   }
6. Clock tuning

Figure III.10: PVT Variations Aware Clock Tree Synthesis Algorithm

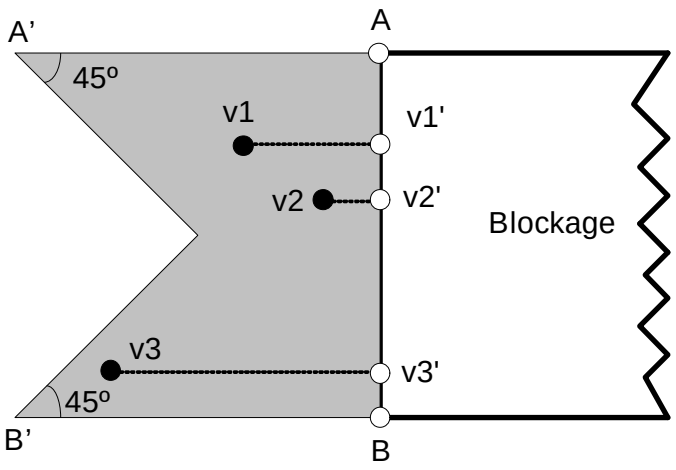


Figure III.11: *Shadow region* and direct virtual node insertion

The Voronoi diagram problem in the presence of obstacles is referred as geodesic Voronoi diagram problem. Many algorithms have been proposed for the construction. However, most of them require the support of complicated data structures [9]. In our application, we only need a basic representation of the geometrical proximity structure. We extended the basic Delaunay triangulation by adding virtual nodes to the boundaries of the obstacles.

In order to describe the virtual nodes adding scheme, we first introduce the concept of a *shadow region*. Figure III.B.4 shows an example of a *shadow region*. Without loss of generality, we assume that AB is the right boundary of a rectangular obstacle. We draw a square $AA'B'B$. Let O be the center of the square. We take away the triangle $OA'B'$, the rest of the square is called the *shadow region* of obstacle boundary AB.

Given a set of points, V, and a set of rectangular obstacles, B, we construct the extended Delaunay triangular mesh in following steps. First, we add every corner point of the obstacles to the virtual nodes set V'. Then, we construct the shadow region for each obstacle boundary segment. For every node v_i lies inside the shadow region of obstacle boundary AB, we obtain its projection, v_i', on AB and add virtual node v_i' to the virtual node set V'. We compute the Delaunay triangulization on point set $V + V'$. By deleting all of the edges crossing obsta-

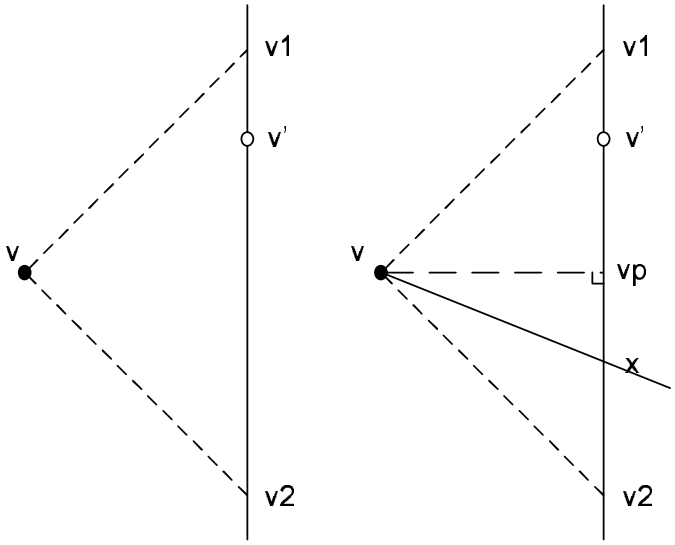


Figure III.12: Proof of Theorem 1

cles, we obtain the extended Delaunay triangular mesh $G_s = (V + V', E_s)$. For the correctness of our virtual node insertion and edge deletion scheme, we prove following lemmas and theorems.

Lemma 1: *Assume an obstacle boundary edge, l, and a node v. From node v, draw two lines each with the slope of $+1$ and -1. Let these two lines intersect l at $v1$ and $v2$, respectively. If there is a virtual node v' between $v1$ and $v2$, then there is not edge in the Delaunay triangular mesh incident to v and intersects l.*

Sketch of the proof: From Figure 4, we see that if there is an edge of Delaunay triangular mesh starting from v intersect with l at point x. The Manhattan distance between v' and x is always smaller than the Manhattan distance between v and x, which contradicts to the definition of Delaunay triangulization. Hence, such edge does not exist.

From Lemma 1, we prove following theorems. The first theorem shows the correctness of virtual node insertion, and the second theorem shows that the graph will not become disconnected after edge deletion.

Theorem 1: *Assume an obstacle boundary edge, l, and a node v. From*

node v, draw two lines each with the slope of $+1$ and -1. Let these two lines intersect l at $v1$ and $v2$, respectively. If there is a virtual node v' between $v1$ and $v2$, then there is not edge in the Delaunay triangular mesh incident to v and intersects l.

Theorem 2: *The extended Delaunay triangular mesh is connected after deleting all the obstacle crossing edges.*

Assume there are n points and k rectangular obstacles on the plane. The extended Delaunay triangulization has following properties.

Lemma 2: *The total number of virtual nodes is $O(n) + O(k)$*

Lemma 3: *The extended Delaunay triangulization can be conducted in $O((n+k)\log(n+k))$ time.*

Lemma 4: *On an extended Delaunay triangular mesh $G = \{V + V', E\}$, if shortest paths between node A and node B are all* non-x-monotone *or* non-y-monotone*, all these paths must contain at least one virtual node.*

Topology Generation through Graph Partitioning

We take the union of the extended Delaunay triangular mesh and clock uncertainty constraint graph, and use recursive graph partitioning to generate the tree topology. The key of this process is the edge weight assignment. For a Delaunay triangular mesh $G = (V + V', E)$ we assign the edge weight using following equation:

$$c(e) = \frac{MAXdist(e')}{dist(e)^{\alpha}} \tag{III.21}$$

The intuition behind this equation is that the cost of separating two nodes should be inversely proportional to the physical distance between them. In our implementation, we set the value of α to 1.0. For clock uncertainty constraint graph $G = \{V, E\}$, the edge weight is set using equation $c(u,v) = ke^{\frac{1}{\Delta u,v}}$, where, $\Delta u, v$ is the maximal permissible clock uncertainty between node u and node v. The constant k is decided by the ratio of the total edge weight of G_t and G_s. In our implementation, the value of k is set to let the total edge weight of G_t and G_s

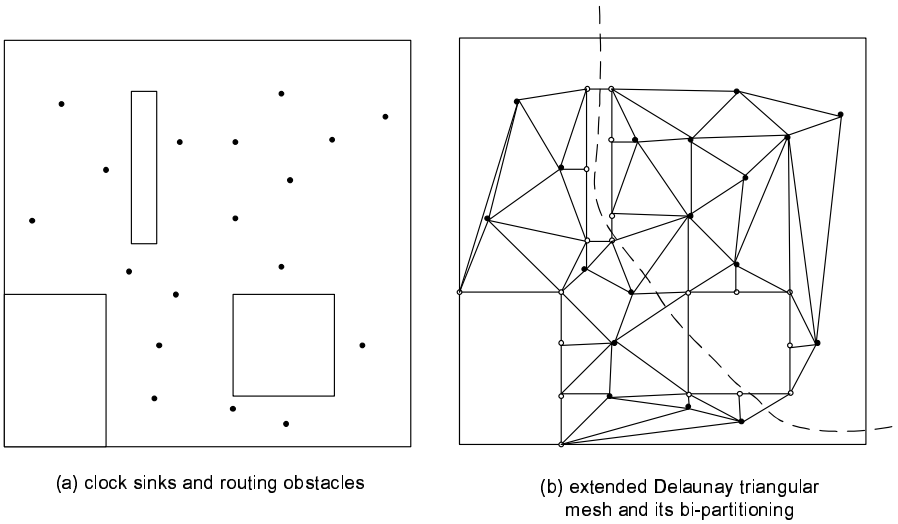


Figure III.13: Extended Delaunay Triangulization and a Bi-Partitioning of Triangular Graph

to be equal. The k value decides the tradeoff between PVT variations tolerance and wire length. Choosing the larger k implies higher preference for variations tolerance while the smaller k gives the preference to shorter wire length.

Note that when constructing the Delaunay triangular mesh, we do not distinguish the flip flops and the virtual nodes. The weight of each edge is assigned purely based on the physical distance. The flip flops and the virtual nodes are distinguished by different node weight. When we do partitioning, the node weight of a flip flop is proportional to its capacitive load, while all the virtual nodes have zero weight.

Figure 5 shows an example of a bipartitioning on an extended Delaunay triangular mesh. Figure 5(a) illustrates the clock sinks distribution and routing obstacles. Figure 5(b) is the extended Delaunay triangular mesh and its 2-way partitioning. The dashed line illustrates the cut line of the graph. With this partition, the clock sinks are clearly partitioned into two parts by its physical location and no obstacle is immersed inside the points set of the same cluster.

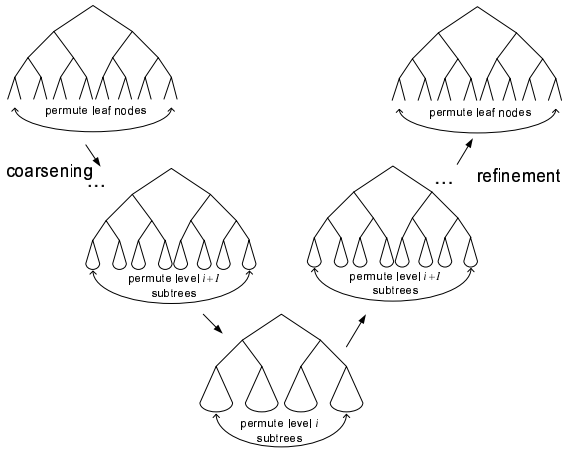


Figure III.14: A v-cycle of multilevel clock tree refinement

III.B.5 Multi-level refinement and tree embedding

The recursive graph partitioning presented in the previous section produces a good balance between physical proximity and temporary constraints. However, at the abstract level of graph partitioning, there is a lack of detail physical embedding information; the optimization can not be driven by accurate wirelength and delay cost. Hence, the topology generated is suboptimal. We adopt a multi-level optimization scheme to further improve the clock tree topology and embedding. While exploring the search space of different topologies, we simultaneously construct the physical embedding and insert buffers. The accurate wirelength and delay estimations guide the topology optimization. According to our experiments, the multi-level refinement can improve the total wirelength by 10% to 15% as well as reduce the number of timing violations by 10% to 30%.

Figure 6 illustrates the process of a "v-cycle" in the multilevel clock tree refinement. The procedure starts from the finest level optimization, where the permutation of the leaf level nodes is optimized. Through a coarsening process, we optimize the tree topology at a higher and higher level. At level i, the permutation of the i^{th} level subtrees are optimized. When we finish the bottom level optimization, we go back to a refinement process, where the optimizations are performed at finer and finer level. We repeat this V-cycle several times, until we find a satisfied

solution or the quality of solution can not be improved. Typically, it takes 4 to 6 V-cycles to optimize a clock tree with about 200K clock sinks.

At level i, a low temperature simulated annealing procedure optimizes the permutation of the i^{th} level subtrees. The cost function used for simulated annealing consists of three parts, the total wirelength, the insertion delay, and the clock uncertainty constraints violations. The fundamental movement of the simulated annealing is an exchange of subtrees rooted at the same level [5]. A fast clock tree embedding subroutine incrementally constructs the embedding of the tree and calculates the cost for simulated annealing.

In order to accelerate the wire length and delay estimation, we adopt a simplified tree embedding and buffer insertion scheme. The buffer insertion scheme is a simple fanout rule based insertion. We restricted the clock buffers to be placed only at the Steiner points, and enforce the load to input capacitance ratio for every clock buffer as a constant, for example, 4. This scheme requires $O(\log n)$ time for an incremental adjustment.

The embedding algorithm is based on bottom-up merging. At each non-leaf node in the tree structure, we record the location of the center of mass for the subtree rooted at that node. When we merge two nodes, we use Dijkstra's shortest path algorithm on the Delaunay graph to find a path between two nodes and let all L shaped connections bend toward the center of mass of the subtree rooted at its parent node. We use Tsay's zero skew emerging scheme [20] to decide the location of tapping points and apply wire snaking to balance the delay when needed. This scheme requires only $O(\log n)$ time for an incremental tree construction.

After obtaining the optimized topology, we can use DME algorithm to find the best embedding and use more accurate delay calculation methods to adjust the tapping point position and buffer sizes.

Table III.6: Comparisons between our method and DME

ckt	PVAT				DME			
	#vio	MV(ps)	w.l.(mm)	cpu(s)	#vio	MV(ps)	w.l.(mm)	cpu(s)
r1	7	61	188.1	9.2	45	165	187.2	0.1
r2	5	107	362.3	16.0	63	331	355.0	0.3
r3	9	375	447.6	47.6	82	979	443.9	0.4
r4	14	214	906.1	103.4	295	642	894.2	2.6
r5	28	213	1340.9	189.7	457	1204	1316.9	7.4

III.B.6 Experimental Results

We implement the clock tree synthesis flow in C programming language. We use Metis library [14] for graph partitioning and Triangle 1.5 package [23] for Delaunay triangulization. The platform is a 2.4GHz Pentium 4 desktop running Linux.

We perform two sets of experiments. First, we compare the performance of our methodology with DME algorithm on a set of publicly available benchmarks [24]. This set of benchmarks does not contain routing obstacles. Then, we demonstrate the experimental results on a large synthetic benchmark. The design has 200K flip flops, 1.6million timing constraints and 200 rectangular routing blockages. The chip size is 14 mm by 14 mm. The capacitive load of each flip flop is 3.4fF.

We randomly generate the permissible clock uncertainty constraints for all pairs of flip flops. For any pair of flip flops, we assume there is a data path between them with probability 0.1. If there is a data path between two flip-flops, we assume the maximal permissible clock uncertainty between them is a random number uniformly distributed in the range of 1 to 1000 ps.

We generate clock trees using both our program and the DME algorithm. The original DME algorithm generates unbuffered clock trees. We use the same buffer insertion/sizing routines we used in our PVAT algorithm to insert the buffer on the clock trees generated by DME algorithm. We extract the SPICE netlist from the routed clock tree. We set all the R, C values, supply voltages, and

transistors lengths to be random variables with Gaussian distribution. The 3σ values are 10% of their nominal values. To model the temperature variations, we sweep the environment temperature from 20C to 80C. Due to the simulation tool limitation, we did not model the effect of the on-chip temperature gradient. Part of its effect is reflected by the interconnect resistance variations. We perform Monte Carlo analysis using HSpice. We get the clock uncertainty by taking the maximal difference between clock delays of two flip-flops in 40 runs of HSpice simulations.

Table 1 are the comparisons between our method and DME algorithm. Column 2 to column 5 are the results of our algorithm. Column 6 to column 9 list the results for DME. The second and the sixth columns are the numbers of clock uncertainty violations. Our method on average has 12.6 violations for every circuit while the DME has 108. The third and the seventh column show the maximal violations. Our method reduces the average value of maximal violations by 70%. The fourth and eighth columns are total wirelength. Comparing with DME, our method only increase the wirelength by 1.5%.

Figure 7 shows the top level routing with routing blockages for the large scale testcase. We can see from the figure that the existence of obstacles significantly distorted the physical proximity structure. Our program completes in 6 hours. Among the 1.6 million clock uncertainty constraints, only 65 of them are violated.

III.B.7 Conclusions and Future Directions

We present a novel clock tree synthesis methodology for high performance ASICs. The proposed scheme recognizes different requirements on clock uncertainty by different data path and let the registers on the two ends of a critical path share more common path on a clock tree, thus decrease the effect of clock uncertainty caused by PVT variations. We use an extended Delaunay triangular mesh to represent the clock sink proximity in the presence of routing obstacles. Taking both spatial and temporal relations into consideration, we use graph par-

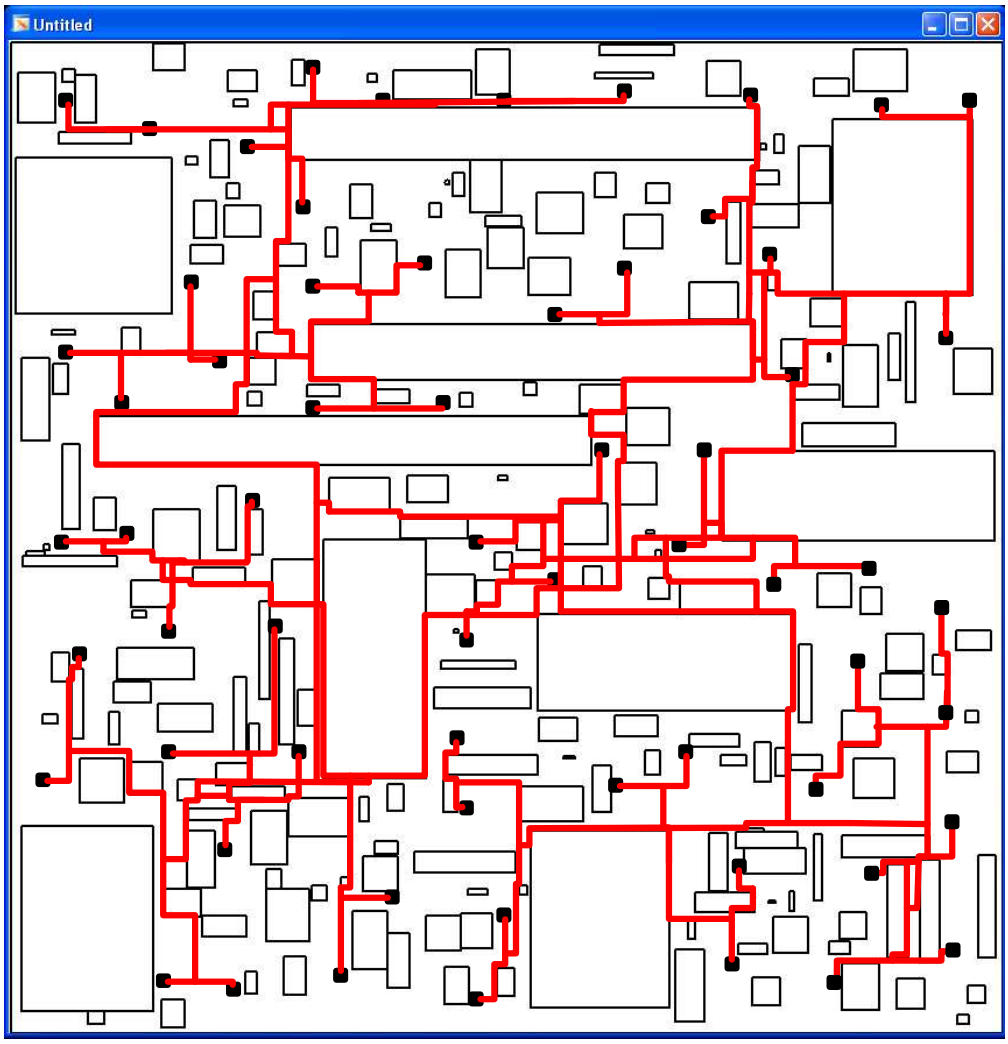


Figure III.15: Clock tree for a 200K flip-flops, 200 obstacles chip top level route are shown, small black squares indicate the root of local routing trees

titioning to get an initial clock tree topology. We then refine the clock topology and embedding through a multi-level optimization process.

Experimental results show that our method produces only one ninth of the clock uncertainty violations comparing with DME algorithm. It reduces the maximal violations by 70% with only 1.5% increase in wirelength. We also demonstrate that our algorithm can complete a complex design with 200K flip-flops, 1.6 million timing constraints, and 200 rectangular routing obstacles in 6 hours of CPU time.

Our ongoing efforts include following:

- Integrate the proposed clock tree topology generation method with variation aware clock embedding/routing [7].
- Better post processing method including wire sizing, variations aware buffer insertion.
- Clock tree topology generation considering more grouping constraints, e.g. clock gating, multiple clock domains, etc.

III.C A Multi-Level Transmission Line Network Approach for Multi-Giga Hertz Clock Distribution

In high performance systems, process variations and fluctuations of operating environments have significant impact on the clock skew. Recently, hybrid structures of H-tree and mesh [2,15,18,19] were proposed to distribute the clock signal with a balanced H-tree and lock the skew using the shunt effect of the mesh. However, in multi-giga hertz regime, the RC model [15] of the mesh is no longer valid. The inductance effect of the mesh can even make the skew worse. In this section, we investigate the use of a novel architecture which incorporates multiple level transmission line shunts to distribute global clock signal. We derive the analytical expression of the skew reduction contributed by the shunt of a transmission

line with the length of an integral multiple of clock wavelength. Based on the analytical skew expression, we adopt convex programming techniques to optimize the wire widths of the multi-level transmission line network. Simulation results show that the multilevel network achieves below 4ps skew for 10GHz clock rate.

III.C.1 Introduction

With increasing clock frequency, the clock skew caused by many non-deterministic factors such as process variations, supply voltage fluctuation and temperature gradient consumes a significant portion of clock period. For high performance synchronous circuitry, the design of a robust global clock distribution system which can sustain various parameter variations becomes an increasingly difficult and time-consuming task.

RC shunted networks have been successfully used to reduce the clock skew under process variations. In [11], three wide spine shunts are adopted to reduce the skew between the leaf nodes of a very deep buffer tree. In [15,18,19], a clock mesh driven by balanced H-tree is used for global clock distribution. In [2,15], multiple level shunts are applied to further reduce the clock skew.

However, when the clock frequency increases to multi-giga hertz range, the inductance effect of the shunt wires becomes significant. At 10GHz clock rate, the time of flight between two corners of a chip is comparable to the clock cycle. The RC model [15] of the shunt effect is no longer valid. The inductance of the shunt can even cause worse skew. We have to view the shunt wires as transmission lines.

On the other hand, a transmission line with a properly tailored length can lock the oscillators together. Galton et al [7] showed that when the wire length is shorter than one quarter wavelength, the transmission line can synchronize oscillators both in phase and magnitude. Several other researches utilized the synchronization capability of the transmission line in clock distribution by connecting the distributed PLLs together with transmission lines [8,9,20]. In order to

compensate the lossy nature of the on-chip transmission lines, [14] used distributed transconductors along the transmission line to generate the standing wave.

In this paper, we propose a hybrid structure of H-tree and the transmission line shunts. We simplify the clock to differential signals of sinusoidal waves. The transmission lines are driven at discrete points and bent into spiral pattern in order to shunt the drivers of the H-tree. The drivers of the H-tree are shunted level by level. The shunt lengths between the drivers are an integral multiple of wavelength. For an ideal case that the line is lossless, a standing wave can lock the clock drivers to zero skew. For lossy shunts, we derive the skew reduction as a function of the wire width. We optimize the wire widths of the multi-level network based on the analytical skew function.

Comparing with other methods, the proposed global clock distribution architecture enjoys several advantages. First, there is no direct feedback path from the transmission line network to the clock source. The transmission lines are linear network and thus the design and optimization involve no active components. Second, the energy storage capability of the locked standing wave in the transmission line can mitigate the clock jitter. And finally, the power consumption of the whole network is much less because the resonance effect of the transmission line.

Our contributions in this paper include:

- We derive the analytical expression of the skew reduction effect of a multiple wavelength-long transmission line shunt.
- We propose to use multi-level spiral networks for multi-giga hertz global clock distributions.
- We adopt a convex programming technique to optimize the wire widths of spirals of each level. Simulation results demonstrate that the optimized clock network can achieve below 4ps skew at 10GHz.

The rest of this paper is organized as follows: In Section 2. we formulate the optimal hierarchical transmission line spiral network sizing problem. In

Section 3, we derive the analytical skew expression for multiple wavelength long transmission line shunted network. In Section 4, we describe the implementation of multiple level transmission line spiral network structure and use convex programming approach to optimize it. We then present the experimental results in Section 5. Finally, we conclude the paper in Section 6.

III.C.2 Problem formulation

The goal of this work is to construct a global clock distribution network that has a working frequency higher than 5GHz and yet robust against various parameter variations. In this section, we first address the inductive effect on shunt. Then we propose to use differential sinusoidal wave to distribute global clock signal. We describe the model of variations in 2.3, and introduce the multi-level network in 2.4. After that, we formulate the optimal spiral sizing problem in 2.5.

Inductance Diminishes Shunt Effect

With rapid increasing of clock rate, the inductance makes skew reduction effect of shunt diminish and even makes the skew worse. In the circuit shown in Figure III.16 two identical clock drivers with input skew T are shunted by a segment of RLC wire. The shunt connection is a 0.5um wide, 1.2cm long copper wire on the 6th metal layer. The R, L, C, the wire inductance, resistance and capacitance values are 2.6nH, 544Ω and 4.6fF, respectively. We set the driving resistance R_s to be 600ohm and the input skew between two drivers to be 20ps.

Table III.7 shows the simulated value of skew at different frequencies. At 0.5GHz, because of the shunt effect, the clock skew between 1 and 2 is reduced from 20ps to 3.9ps. However, at the frequency of 5GHz, the skew between nodes 1 and 2 becomes 26ps, which is even larger than the input skew. The reason why the shunt effect get diminished is that when frequency is above 5GHz, the natural frequency of the shunt wire segment, = 10GHz is comparable to the clock frequency, which makes the wire no longer behave as RC circuit, and the transmission line effect

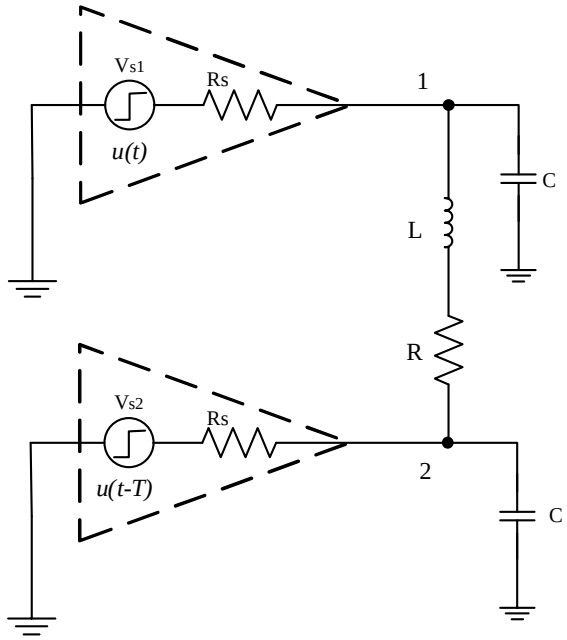


Figure III.16: Two clock drivers shunted by a lumped RLC line

Table III.7: Skew and frequency relation of a shunt segment

f(GHz)	0.5	1	1.5	2	3	3.5	4	5
Skew(ps)	3.9	4.2	5.8	7.5	9.9	13	17	26

must be considered.

Differential Sinusoidal Wave

We adopt differential sinusoidal waves in the global clock distribution. The sinusoidal waveform simplifies the analysis of resonance phenomena of the transmission line. And the differential signals provide well controlled current return loop, thus improve the predictability of inductance value.

A practical concern is how to distribute a square-wave clock signal instead of sinusoidal ones to the clocking elements all over the chip. One approach is that we can use a sinusoidal standing wave to distribute the global clock signal to regions on a chip, and convert it into digital signal at each local region. This approach has been successfully applied to several high-speed clock distribution systems [14][5][7].

In [14], a two stage clock buffer that can convert low-swing differential sinusoids to digital levels. The buffer has two stages. The first stage is a differential amplifier and the second stage uses a cross-coupled inverters and a shunt resistor to convert the sine wave to square one. This buffer can achieve below 1ps amplitude-dependant skew.

Model of parameters variations

represent the systematic spatial variations on wire widths and transistor lengths [12]. For any location (x, y) on the chip, the actual geometrical parameter $d = d_0 + k_x \cdot x + k_y \cdot y$, where d_0 is the nominal parameter and k_x, k_y are the horizontal, vertical variation coefficient, respectively. Without loss of generality, we assume that the origin of the coordinate $(0, 0)$ locates in the center of the chip, and k_x, k_y are positive numbers. We set the maximum variations across the chip to be $\pm 10\%$ of the ideal value [3]. We choose this "pseudo-deterministic" linear variation model because it can be regarded as a "worst case" scenario of the probabilistic variations. However, it is easy to replace that with more sophisticated variation model in our design framework.

We also take the supply voltage fluctuation into account when we analyze the clock skew. In this paper, we assume that the supply voltages of all the clock drivers are a set of independent random variables within $\pm 10\%$ of the nominal Vdd value.

Hybrid H-tree and shunt network

We tailor the natural frequency of the shunt wire to reduce the skew between clock terminals. Figure III.17 shows a hierarchical transmission line spirals network for global clock distribution. Each spiral consists of a pair of multiple wavelength long coplanar differential transmission line. Clock drivers are evenly distributed on every spiral and the separation between two neighboring drivers is one wavelength. An H-tree distributes the sinusoidal clock signals from the center

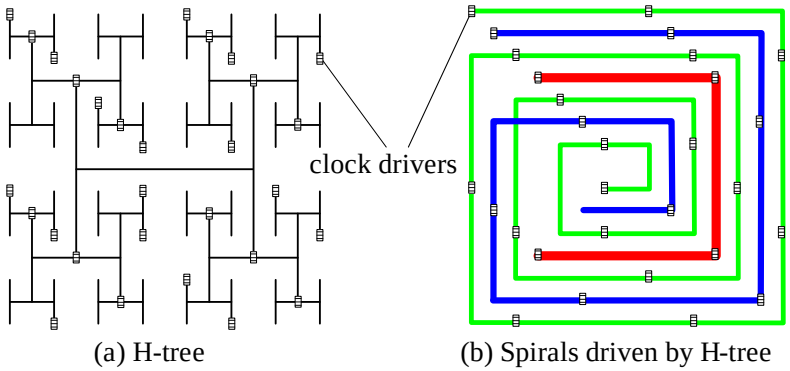


Figure III.17: Multi-level Tansmission Line Spirals Network

of the chip to all the clock drivers. The signal arriving time of all the clock buffers on the same spiral is designed to be equal.

The proposed symmetrical global distribution network distributes low skew clock signals to the lowest level spiral. For each local region, a local distribution tree or mesh is needed to send clock signals from the clock buffers on the lowest level spiral to clocking elements. The clock buffers decoupled the global distribution network and lock distribution network. The skew on the global network is mainly caused by parameters variations.

Problem statement

For same amount of routing resources, assigning them to the spirals at different level may have different impact on the clock skew. In this paper, we are interested in the optimal way to distribute the routing resources to the spirals at different levels such that the minimum skew is achieved on the bottom level spiral with given routing area budget.

We formulate this problem as the following optimum hierarchical transmission line spirals sizing problem.

Optimum Hierarchical Transmission Line Spirals Sizing Problem:

Given: model of parameters variations.

Input: H-tree and n-level spiral network

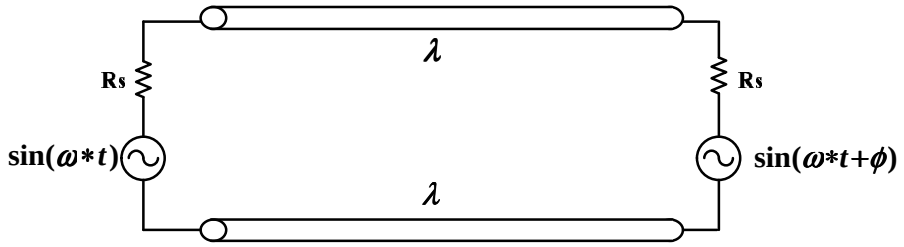


Figure III.18: Two sinusoidal sources with phase shift shunted by a wavelength long transmission line

Constraint: total routing area

Object Function: minimize the skew on the bottom level spiral

Output: The optimum wire width w_i of spirals at level i, for i =1 to n, such that the clock skew is minimized

An alternative formulation to the above one is to minimize the total routing area under given skew tolerance. These two formulations are dual. The solution of one problem can be used to solve another one.

III.C.3 Skew reduction effect of transmission line shunts

Two sources case

We use a simplified circuit model shown in Figure III.18 to study the skew reduction mechanism of a one wavelength long transmission line shunt. In Figure III.18, two clock drivers with driving resistance R_s and input phase shift (skew) Φ are connected by an $RLGC$ transmission line of exactly one wave length long. The output at two separated terminals, V_1 and V_2 are synchronized together by the transmission line. Figure III.19 shows the simulated wave forms. The input skew between input voltages V_s1 and V_s2 is 30 degree, the resulted skew between output voltages V_1 and V_2 is only 0.7 degree.

Assume that input skew is small and $R < \omega L$, by superposition of all possible traveling and standing waves in the transmission line we obtain following

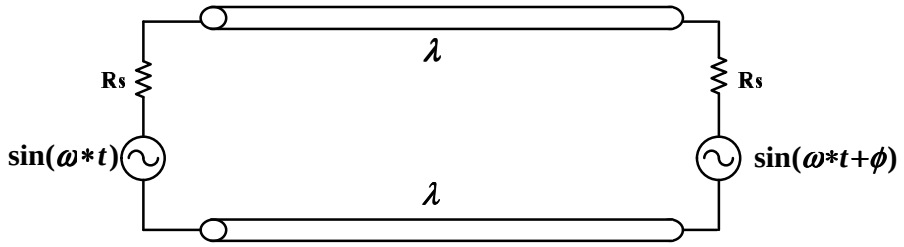


Figure III.19: Simulated waveform of two sources shunted by one wavelength long transmission line

skew expression. The detail derivation is shown in the Appendix.

$$\Delta\Phi = \frac{1 - e^{-\frac{\pi R}{\omega L}}}{1 + e^{-\frac{\pi R}{\omega L}}}\Phi \tag{III.22}$$

From the skew equation (1), we observe that when resistance R approaches zero, the transmission line becomes lossless. As a result, $\Delta\Phi$, the phase shift between voltages V_1 and V_2, also approaches zero. Two clock drivers get fully synchronized. When R approaches infinity, nodes 1 and 2 are open. There is no shunt effect. The phase shift between nodes 1 and 2 remains the input phase shift, Φ.

We use SPICE simulation to validate our skew expression (III.22). We sweep the value of unit length resistance R from $0.5k\Omega/m$ to $200k\Omega/m$, and driving resistance R_s from 5 to 10 times of the characteristic impedance of the transmission line. We set the clock frequency to be 10.336GHz and the input skew Φ to be 45 degree. We use the W-element model in HSPICE to simulate the transmission line behavior. Fig. 5 illustrates both simulated data points and the curve of equation (1). In the range we tested, the analytical skew expression (1) produces less than 2 percent error. In the figure, when R is small, the simulated results match the skew equation very well. When R is larger than 100 $k\Omega$, small errors appear. One possible explanation of the errors is that when R is big enough, the wave propagation speed will be changed, which makes the transmission line no longer exactly one wavelength long.

Table III.8: Skew on a spiral from simulation and calculation

Skew	IIa	IIb	IIc	IIIa	IIIb	IIIc
Sim. (ps)	2.04	2.03	2.01	3.77	3.76	3.75
Cal. (ps)	2.21	2.21	2.21	3.95	3.95	3.95
Err. (%)	7.6	8.3	9.0	4.5	4.8	5.1

Multiple sources case

We use a random model for multiple sources case. Fig. 6 shows the model of multiple sources shunted by a transmission line. We assume that the transmission line is infinitely long and the clock buffers are placed evenly on the line with separation of one wavelength. We assume the input phase of of each voltage source to be a random number uniformly distributed in [0,]. Because it is an infinitely long line, we can assume there are two nodes a, b having exact phase 0 and Φ, respectively. We compute the expected phase of these two points, and take the difference of the expectations as the skew.

We assume the driving resistance is much larger than the characteristic impedance of the transmission line and the input skew is small. Using similar technique in the derivation of equation (III.22), we obtain following skew equation.

$$\Delta\Phi = \frac{1 - e^{-\frac{3\pi R}{\omega L}}}{1 + e^{-\frac{3\pi R}{\omega L}}}\Phi \tag{III.23}$$

The skew expression is also very similar to (III.22) in the format–only with a constant factor of 3.

The skew expression derived from random input phase shift model also matches the SPICE simulation results very well. In our simulation, we use a 4um wide transmission line for the second and third level spirals in Fig. 2. The input phase shifts of the clock drivers are set using simulated arriving time of each buffer in the H- tree under our process variation model. We use three different ways to connect each level spiral. We compare the skew calculated from our equation and that of simulation results in table 2. In The skews calculated from equation (III.23) are all within 10% of the SPICE simulation results.

Table III.9: Frequency dependant R and L at 10GHz

w(um)	0.5	1	2	3	4	5	6
R(oh)	2156.32	1080.64	547.545	374.306	290.479	241.794	210.189
L(nH)	8.40645	7.57603	6.83747	6.45693	6.19725	5.99344	5.82214
R/L	2.5651	1.4264	0.8008	0.5797	0.4687	0.4034	0.3610
w(um)	7	8	9	10	20	30	40
R(oh)	188.033	171.59	158.849	148.633	100.406	92.5334	71.7877
L(nH)	5.67327	5.5416	5.42392	5.318	4.64553	4.4737	4.07527
R/L	0.3314	0.3096	0.2929	0.2795	0.2161	0.1940	0.1762

III.C.4 Optimal sizing of multiple level spirals network

We use a pair of coplanar copper transmission lines to construct the spiral shunt. We follow the technology of [14]. Fig. 7 demonstrates the actual geometrical configuration of the transmission line wires. The two parallel differential wires have height 240nm, and the same width w. The separation between them is 2um, and the wires are 3.5um above a ground plane. Typical value of w ranges from 0.5 to 40 um.

At a clock rate of 10GHz, the skin effect and proximity effect can make the wire resistance and loop inductance deviate significantly from its static value. Simply plugging the static R, L value into equation (3) can lead inaccurate estimation of skew.

According to equation (3), for a given frequency, the skew only depends on the ratio of R/L, which is a function of wire widths w. We use 3D filed solver FastHenry to extract the frequency dependant resistance and inductance. Based on the extraction results, we use curve fitting technique to determine the R/L w relation. Table 3 lists the extracted R and L value. Because of skin and proximity effects, the resistance and inductance of 40um wide wire deviate $+200\%$ and -75% from its static value, respectively.

Linear regression gets the following $R/L \sim w$ function at 10GHz: $R/L = a/w + b$, where, a and b are constants.

Figure III.20 plots the curve of $a/w+b$, and the data points of extraction

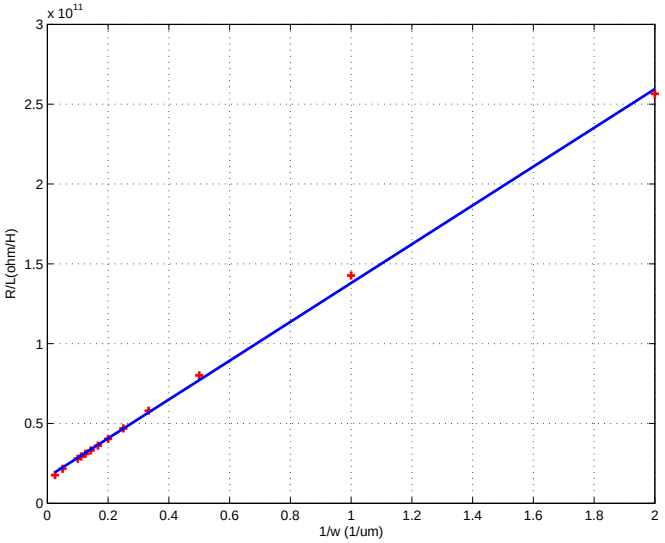


Figure III.20: R/L and $1/w$ relation at 10GHz

results. The $R/L \sim 1/w$ relation displays excellent linearity, and matches the curve of linear regression very well. We rewrite the skew function of each level spiral as

$$\Delta\Phi = \frac{1 - c_i e^{-\frac{k_i}{w_i}}}{1 - c_i e^{-\frac{k_i}{w_i}}}\Phi \tag{III.24}$$

Where, wi is the width of the ith level spiral and ci, ki are constants for level i spiral. We write the optimal spiral sizing problem as following mathematical programming:

$$\begin{aligned} Min: \quad & \Delta\Phi = ((\Phi_1 e^{-\frac{k_1}{w_2 1}} + \Phi_2) e^{-\frac{k_2}{w_2}} \ldots + \Phi_n) \\ S.t.: \quad & \sum_{i=1}^{n} l_i w_i = A \end{aligned} \tag{III.25}$$

In the programming III.25, Φ_i is the skew of signal propagation from level $i-1$ to level i spiral. L_i and w_i are length and width of the spiral of level i. The object is to minimize skew under the maximum routing area constraint A. One practical concern about the programming III.25 is that the cost function is not a convex function all over its domain. Fortunately, it is convex in the region we are interested in. We prove following lemma.

Lemma 1 *$f(w) = \frac{1-ce^{-k/w}}{1-ce^{-k/w}}$ is a convex function on $w \in [\frac{k}{2}, \infty)$, where, k is a positive constant.*

The above lemma suggests that, when the wire of the transmission line is wide enough, the skew and wire-width relation is convex. In other words, the skew reduction equation III.22 is convex when each level spiral reduces the skew by at least $1 - \frac{1-e^{-1}}{1+e^{-1}} = 23.9\%$. In order to make the programming convex, we impose a set of minimal wire width constraints to each level spiral. In our experiments, the minimal wire widths of each level mesh are 0.6um, 1.3um, 1.3um, respectively. With the minimal wire width constraints for each level spiral, we get following convex program.

$$\begin{aligned} Min: \quad & \Delta\Phi = ((\Phi_1 e^{-\frac{k_1}{w_2 1}} + \Phi_2) e^{-\frac{k_2}{w_2}} \ldots + \Phi_n) \\ S.t.: \quad & \sum_{i=1}^{n} l_i w_i = A \\ & w_i < c_i \end{aligned} \tag{III.26}$$

Due to the convex property of the program III.26, we have the following theorem.

Theorem 3 *The local optimum of the programming III.26 is the global optimum*

According to the above theorem, many numerical methods such as gradient descendant and line search methods can be adopted to solve this class of programming. In our experiments, we solve them using the optimization package of MATLAB.

III.C.5 Experimental results

In our experiments, we set the chip size to be 2cm by 2cm, and use a three level spiral to distribute clock signal. The clock frequency is 10.336GHz. And the wave length is exactly 1cm. Each of the spirals has 4, 9, and 17 clock drivers respectively. We synthesize a balanced H-tree[4] to distribute clock signal from the center of the chip to the clock drivers. The designed arriving time of all drivers on the same level spiral is equal. With given process variations model, we

Table III.10: Optimized wire width of each level spiral

Area	W1(um)	W2(um)	W3(um)	Skew M (ps)	Skew S (ps)	Impr.(%)
0	0	0	0	23.15	23.15	0
0.5	1.7	0	0	17.796	20.50	13
1	1.9308	1.0501	0	12.838	14.764	13
3	2.5751	1.3104	1.3294	8.6087	8.7309	15
5	2.9043	3.7559	2.3295	6.2015	6.3169	16
10	3.1919	4.5029	6.8651	4.2755	5.2131	18
15	3.6722	6.1303	10.891	2.4917	3.5182	29
20	4.0704	7.5001	15.072	1.7070	2.6501	37
25	4.4040	8.6979	19.359	1.2804	2.1243	40

obtain the worst skew of the signal propagation form one level to the next level based on SPICE simulation. We use these skews as the values of Φ_i in the convex programming. We normalize the routing area to the area of bottom level spiral with 1um wire width.

Table III.10 lists the optimized wire width of each level spiral for different total routing area. W1, W2, and W3 are optimal wire widths of level 1, level 2 and level 3 spirals, respectively. For the comparison reason, we also simulate the skew on a single-level spiral network, which only uses bottom level spiral to shunt all the leaf nodes of the H-tree. We let the single level spiral network has same total routing area of the multi-level spirals network. Column 5 and 6 are the skews of multi-level spirals and single level spiral. Column 7 shows the skew improvement of multi-level spirals over single level spiral. When total routing area is small, the optimal configurations prefer to allocate routing resources to the higher level mesh. With gradually increasing of the routing area, more resources are allocated to the bottom level mesh. Comparing with the single-level spiral, optimized multi-level spiral can reduce the skew by 40%.

Figures III.21 and III.22 demonstrate the simulated transient and steady state waveforms of voltages on the bottom level spiral. The lower plot is the steady state wave form. The sin waves of larger magnitude are input waveforms of clock buffers and the waves of smaller magnitude are output waveforms on the spiral.

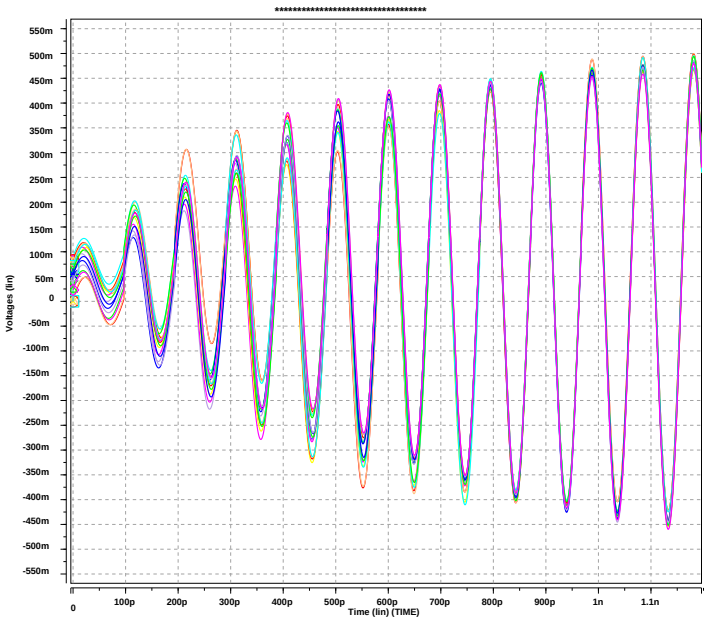


Figure III.21: Transient Waveforms on Bottom Level Spiral

Table III.11: Power Consumption Comparisons

Area	3	4	5	7	10	15	20	25
PM(mw)	0.4	0.5	0.7	0.9	1.0	1.4	1.5	1.6
PS(mw)	0.83	1.5	2.1	2.64	3.04	4.7	7.2	8.3
reduction(%)	48	67	67	66	67	70	79	81

Because of the transmission line shunt effect, the skew reduced from 8.4ps to 1.2ps. The upper plot is the transient response of the output voltages. The output signals get locked together within 10 clock cycles.

We also compare the power consumption of optimized multilevel spiral network and that of single level spiral. In Table III.11, the first row are the total routing areas of the multi-level spirals; the second row and the third row list the power consumption of the multilevel spiral and single level spiral with given amount of total routing area. The simulated results show that multilevel spiral can reduce the power consumption by 81%.

We test the robustness of our optimized spirals network against supply voltage fluctuations. We perturb the supply voltage of every clock drivers inde-

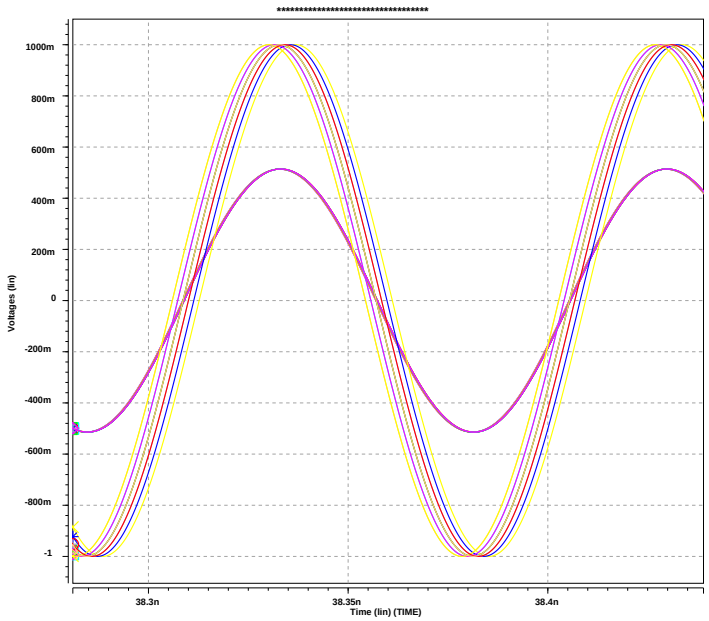


Figure III.22: Steady States Waveforms on Bottom Level Spiral

Table III.12: Skew comparison in the presence of voltage variations

Area	Skew-S		Skew-M		
	Ave (ps)	Worst (ps)	Ave (ps)	Worst (ps)	imprv (%)
0	28.4	36.5	28.4	36.5	0
3	9.75	12.33	8.75	9.07	11
5	7.32	9.06	6.55	6.91	12
10	6.31	805	4.41	5.41	30
15	5.03	7.33	2.81	4.93	44
25	3.83	4.61	1.72	3.06	55

pendently by a random number within 10% of its nominal value. We perform 5 experiments on each network, and record the worst case skew and average case skew in the Table III.12. We compare the skew of optimized multilevel spiral and single level spiral networks. The last column of Table 6 lists the improvement of the average case skew. Multilevel spiral network improves the skew by up to 55%.

When the clock frequency deviates from its as-designed value or the electrical length of transmission lines varies from integral multiple of the wavelength, the resonance phenomena of the transmission line shunts diminishes. As a result, the synchronization capabilities of transmission line shunts degrade accordingly.

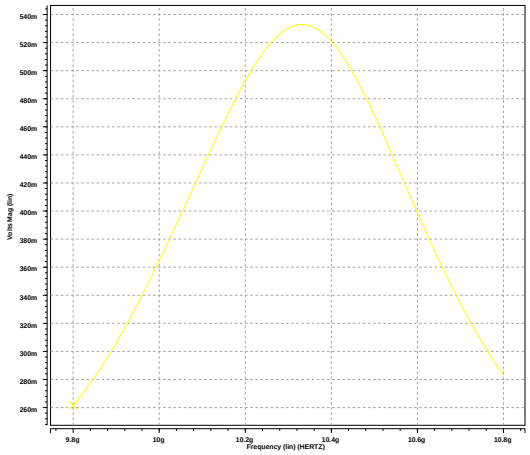


Figure III.23: Frequency response of the transmission line network: voltage $\sim$ frequency relation

We explore the frequency response properties of the proposed multilevel clock network. We set the wire width of the lowest level transmission line to be 5um wide and the clock rate to be 10.33 giga hertz. Figure III.23 shows the output voltage and frequency relation, the -3db bandwidth of the output voltages is 0.42 giga hertz. And Figure III.7 illustrates the skew and frequency relation. At 10.33 giga hertz, the minimal skew of 1.38 degree is achieved. In the frequency range of 10.2 to 10.5 giga hertz, the skew lies between 2.5 degree and 1.38 degree.

III.C.6 Conclusions

When the clock frequency thrust into multi-giga hertz regime, transmission line shunts demonstrate its unique potential of achieving low skew low jitter global clock distribution under parameter variations. We propose a new architecture of global clock distribution that incorporates multiple level transmission line spirals.

We derive the analytical expression of the skew reduction contributed by multiple wavelength long transmission line shunts. The skew formulas display very good fidelity to the spice simulation. Based on the theoretical analysis of

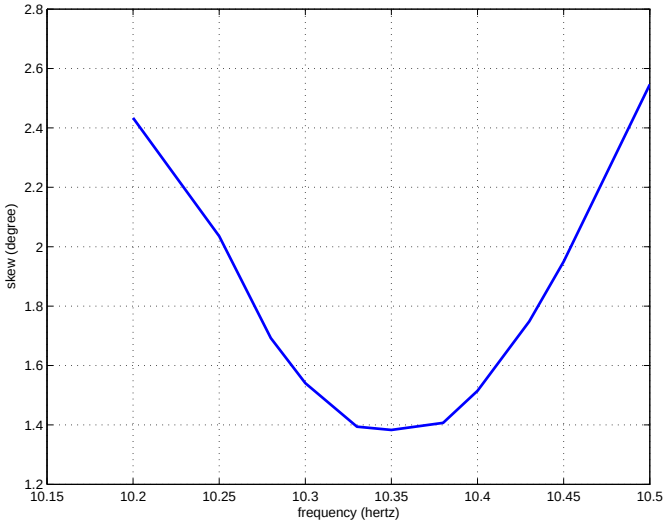


Figure III.24: Frequency response of the transmission line network: skew $\sim$ frequency relation

the transmission line shunt behaviors, we use a convex programming technique to optimize the wire width of each level spiral in the multiple level spirals clock network. The experimental results show that: 1. Comparing with the traditional RC buffer tree, the optimized multilevel transmission line spiral network can reduce the clock skew by 84%; 2. With 10% random supply voltage fluctuation, our proposed multilevel transmission line spiral network provides a less than 2ps global clock distribution at 10GHz; 3. the power dissipation overhead of the transmission line shunts is only 1.6mW.

Our future research directions on multiple level transmission line shunted clock networks include: 1.exploring other innovative topologies of the transmission line shunts; 2. design of suitable differential clock buffers for the network, and 3.actual layout and fabrication of the test circuit to justify our analysis.

IV

Distortionless Transmission Line for On-Chip Electrical Signaling

We present a novel scheme to implement distortionless transmission lines for on-chip electrical signaling. By introducing intentional leakage conductance between the wires of a differential pair, the distortionless transmission line eliminates dispersion caused by the resistive nature of on-chip wires and achieves speed of light transmission. We show that it is feasible to construct distortionless transmission line with conventional silicon process. Simulation results show that using 65nm technology, the proposed scheme can achieve 15Gbits/s bandwidth over a 20mm on-chip serial link without any equalization. This approach offers a six times improvement in delay and 85% reduction in power consumption over a conventional RC wire with repeated buffers.

IV.A Introduction

Interconnects, especially the global interconnects, have been widely recognized as the dominating factor in deciding the system performance and power consumption. With ever increasing clock frequency, the inverter repeated wires can no longer keep pace with advances in transistor speed at a satisfactory cost of power consumption [26]. In order to break this "interconnect wall", many innov-

ative interconnect technologies, ranging from optical interconnect [28] to on-chip RF communication [22], have been proposed by various research groups.

Due to the cost and design complexity considerations, the electrical signaling over on-chip transmission lines is one of the most attractive solutions for high performance on-chip communications [21]. Comparing with traditional inverter repeated RC wires, the transmission line has two main advantages. First, the signal propagates at the speed of light on a transmission line. It can achieve higher throughput at lower latency. Second, the transmission line signaling has much smaller power consumption because it eliminates the forced swing of wire capacitance in the RC wires with repeated repeaters.

One challenge to the implementation of transmission line for on-chip communication is resistive nature of on-chip metal wires. The high wire resistance causes significant frequency dependency on both wave propagation speed and attenuation. For a random digital sequence, the spectrum of the signal waveform spans a wide range. The signal phase velocity and attenuation change substantially in the operational frequencies [23]. This phenomenon renders excessive dispersion at the receiver. Inter-symbol interference (ISI) causes significant data-dependent jitter and limits communication throughput.

In order to control the waveform dispersion, several innovative approaches have been proposed. In [25], pre-emphasizing and de-emphasizing along with data aliasing are used to modulate the input wave form. In [20], Afshari and Hajimiri adopted a non-linear transmission line approach to generate solitary wave propagation and thus compensate for the dispersion. In [23], a high frequency carrier modulates the input waveform and shifts the spectrum of transmitted signal to a less frequency sensitive region. In [27], a clocked discharging scheme is adopted to erase the data dependant delay variations. In [30], an adaptive equalization scheme is used to compensate the propagation loss.

In this paper, we present a new on-chip electrical signaling scheme using distortionless transmission line. With intentionally inserted leakage conductance,

the wave can preserve its original form and propagates at the speed of light, independent of its frequency. We name this scheme *Surfliner* because the way we discretely insert shunt conductors between two parallel wiring tracks resembles the look of rail road[1]. The concept of distortionless transmission line was first proposed by O. Heaviside in 1887. The original idea is to introduce some intentional leakage on the long-distance telegraph cable, such that the wave form can be easily distinguished at the receiver's end with the fidelity to its original shape but smaller amplitude. Through careful mathematical derivation, we shall see that with exactly matched $RLGC$ values, the waveform can be transmitted without any distortion in shape along the transmission line.

We show the feasibility of the implementation of Surfliner on silicon for speed of light global communications. By periodically inserting leakage resistors between two wires of a differential pair, we can achieve near distortionless wave propagation. Experimental results suggest that: 1) At 15Gbit/s data rate, the jitter caused by the communication over a 2cm long transmission line is lower than 10ps; 2) The average power consumption of a data transportation through a 2cm long distance can be as low as 3.1pJ/bit; and, 3) The wiring channel requires less than $1000um^2$ area on poly for a 2cm long serial link.

Comparing with other schemes, the Surfliner has the following advantages:

- The signal propagation on the Surfliner is exactly the speed of light in the dielectric. This property is attractive for the connections with extreme requirement on the signal latency, for example, the global control signal in a large processor or the global data communication in a large network-on-chip.

- The waveform remains undistorted at the receiver end, and there is no ISI. As a result, the transmission produces extremely low jitter. This property enables very high bandwidth communications.

[1] *Surfliner* is the name of railroad runs between San Diego and San Luis Obispo

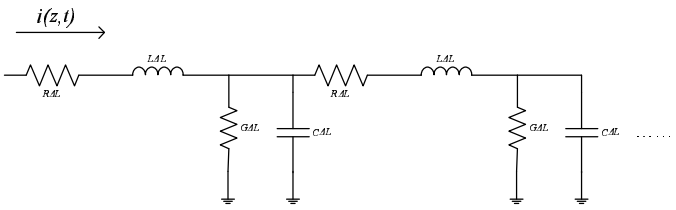


Figure IV.1: *RLGC* model of a transmission line

- Because the signal does not take full swing on the entire wire and no buffer is inserted, the power consumption is much less than the RC wires with repeaters.
- The scheme requires very simple sender and receiver circuits
- There are no active components between the sender and the receiver. The system is robust against process, voltage and temperature variations.

The rest of the paper is organized as follows. In Section IV.B, we review the theory of distortionless transmission line. In Section IV.C, we describe how to exploit the distortionless transmission line for on-chip communications. In Section IV.D, we show the simulation results of our proposed design. At last, we conclude our paper in Section IV.E.

IV.B Theory of Distortionless Transmission Lines

In this section, we review the theory behind the distortionless transmission line. Fig. IV.1 illustrates a discrete *RLGC* transmission line circuit model. Where, R, L, G, C are the unit-length resistance, inductance, capacitance, and conductance, respectively[2]. The wave is described as a function of distance, z, and time, t, by the Telegrapher's equations:

$$\frac{dV(z,t)}{dz} = -RI(z,t) - L\frac{dI(z,t)}{dt} \tag{IV.1}$$

[2]Here, we assume that R, L, G, C are frequency independent constants. We shall see later in our simulation results that the jitter caused by the frequency dependency of R, L, G, C value is not significant for on-chip interconnect.

$$\frac{dI(z,t)}{dz} = -C\frac{dV(z,t)}{dt} - GV(z,t) \tag{IV.2}$$

For a sinusoidal signal of angular frequency ω, the propagation of the incident wave along the transmission line can be expressed as:

$$V(z) = V_0 e^{-\alpha z - j\beta z} \tag{IV.3}$$

where, α and $j\beta$ are the real and imaginary part of the propagation function γ, respectively, i.e.

$$\gamma = \sqrt{(R + j\omega L)(G + j\omega C)} = \alpha + j\beta, \tag{IV.4}$$

From equation (IV.3) we see that the exponential, $e^{-\alpha}$, of the real part of the propagation function represents the unit distance attenuation of the transmission line. The imaginary part, β, of the propagation function corresponds to the phase shift of the waves along the distance. The phase velocity of the incident wave is ω/β.

IV.B.1 Distortion and ISI

Because silicon dioxide is a very good insulator, there is no leakage conductance, i.e. $G = 0$, for on-chip transmission lines. The real part, α, and imaginary part, β, of the propagation function can be written as equations (IV.5) and (IV.6). The attenuation, α, and phase velocity, ω/β, both depend on frequency ω, especially when the wire resistance, R, is comparable to or larger than the impedance ωL contributed by wire inductance.

$$\alpha = \sqrt{\frac{1}{2}}\sqrt{-\omega^2 LC + \omega C\sqrt{\omega^2 L^2 + R^2}} \tag{IV.5}$$

$$\beta = \sqrt{\frac{1}{2}}\sqrt{\omega^2 LC + \omega C\sqrt{\omega^2 L^2 + R^2}} \tag{IV.6}$$

.

IV.B.2 Distortionless Transmission Line

We design a transmission line by inserting leakage conductance. The shunt conductance provides an additional current path to compensate for the voltage drop due to serial resistance. The attenuation still exists, but becomes frequency independent. In other words, the effects of the series resistor and shunt conductance cancel out so that the waves propagate without distortion. The net effect is that the signal can now move at the speed of light in the media.

We set the leakage conductance G per unit-length as the following equation.

$$G = RC/L \tag{IV.7}$$

Substituting equation (IV.7) in equation (IV.4), we get the *frequency independent* attenuation and phase velocity, i.e. distortionless transmission.

$$\alpha = R/\sqrt{L/C} \tag{IV.8}$$

$$\beta = \omega\sqrt{LC} \tag{IV.9}$$

For this distortionless transmission line, we obtain the following.

Characteristic impedance:

$$Z_0 = \sqrt{\frac{L}{C}} \tag{IV.10}$$

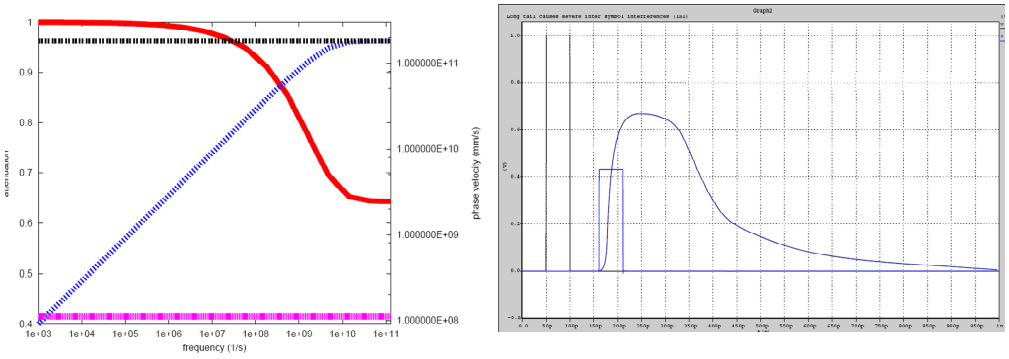
Phase velocity:

$$v = \frac{1}{\sqrt{LC}} = c \tag{IV.11}$$

Attenuation:

$$A(z) = e^{-\frac{R}{Z_0}z} \tag{IV.12}$$

The distortionless transmission line has pure resistive characteristic impedance (8). The attenuation is an exponential function of the ratio between wire resistance and characteristic impedance, i.e. $e^{-\frac{R}{Z_o}}$. The phase velocity is exactly



(a) frequency dependency of attenuation and phase velocity

(b) Time domain pulse response of typical on-chip wire and distortionless transmission line

Figure IV.2: The attenuation and phase velocity v.s. frequency for a on-chip wire

the speed of light in the dielectrics, $\frac{1}{\sqrt{LC}}$. Both the attenuation and the velocity are independent of frequency. There is no distortion on the signal waves from direct current (DC) mode to very high frequency as long as the $RLCG$ values remain the same.

For typical on-chip transmission line implemented on upper low impedance metal layers, the resistance of a several micron wide wire can be less than 10ohm/mm, and the differential characteristic impedance of transmission lines is usually around 100ohm. Thus, an input signal with magnitude of 1.0V will have a magnitude of 135mV after traveling a distance of 2cm.The state-of-art sense amplifier can easily detect the output signal at this magnitude. [29]

Fig. IV.2 illustrates the characteristics of a differential pair of $2cm$-length and $4\mu m$-width wires. We observe significant changes in attenuation and phase velocity (Fig. IV.2(a)). When shunt conductance $G = 0$, the attenuation ranges from 0.9997 at 1KHz to 0.644 at 1GHz. Note that there is no voltage magnitude drop when the attenuation value is 1. The phase velocity ranges from $9.0 \times 10^7 mm/s$ at $1KHz$ to $8.5 \times 10^{10} mm/s$ at 1GHz. The curve saturates at the speed of light in dielectric $1.8 \times 10^{11} mm/s$ when the frequency is above 1THz. For our proposed distortionless pairs, the attenuation is 0.4147 and the phase velocity is at the speed of light. The curves of the distortionless wires are flat.

Fig. IV.2 (b) shows the dispersion in time domain. The input is the

square wave at left, which rises at 50ps. When shunt conductance $G = 0$, the output disperses on the rising and falling edges. There is a very long tail at the falling edge, which can interfere with the following input bits unless we wait until the wave drops below some threshold. This intersymbol interference is one limiting factor of the performance of the transmission lines.

For the distortionless pairs, the magnitude of the signal drops. The rising edge starts at 161ps, the same rising time as the wires with no shunts. However, the output maintains a square waveform. The delay is $161 - 50 = 111ps$, which is at the speed of light, $1.8 \times 10^{11} mm/s$ for $2cm$ length.

IV.B.3 Sensitivity of Distortionless Transmission Lines to the Parameters Variations

One nice property of the distortionless transmission line is that it is much less sensitive to the process, voltage, and temperature variations than traditional interconnects. First, the speed of light is determined by the dielectric constant. The feature size variations do not significantly affect the speed. Second, in the following, we show that distortionless wires are designed to minimize the sensitivity of the wire resistance and shunt conductance variations.

We observe the sensitivity due to shunt conductance variations. The derivation of the sensitivity due to wire resistance changes is similar. Assume that the leakage conductance G varies from its perfectly matched value RC/L by a constant factor Δ, i.e. $G = (1 + \Delta)RC/L$. Substituting this expression into equations (IV.5) and (IV.6), using Taylor's expansion, we derive the attenuation constant and phase velocity in the second order.

$$\alpha = \frac{R}{\sqrt{L/C}}(1 + \frac{1}{2}\Delta - \frac{1}{8}\frac{R^2}{R^2 + \omega^2 L^2}\Delta^2) \tag{IV.13}$$

$$v = \frac{1}{\sqrt{LC}}(1 - \frac{1}{8}\frac{R^2}{R^2 + \omega^2 L^2}\Delta^2) \tag{IV.14}$$

In equations (IV.13) and (IV.14), the frequency dependent terms occur not at the first but the second order. Note that for the attenuation constant, we have a first order term independent of the frequency. In other words, this first order term does not contribute to the distortion. For the phase velocity, the first order term is zero. Therefore, we can derive that the shunt conductance $G = RC/L$ is the solution to minimize the skew sensitivity. Applying a similar procedure, we can also derive that given the shunt conductance $G = RC/L$, the serial resistance R is the solution to minimize the skew sensitivity.

The coefficients of the second order terms in equation (IV.13) and equation (IV.14) are limited by an upper bound $\frac{1}{8}$. Suppose that the shunt conductance changes by ten percent, i.e. $\Delta = 0.1$. We can derive that the third order terms in equations (IV.13) and (IV.14) deviate by no more than $\Delta^2/8 \approx 0.0012$.

IV.C Exploiting the Surfliner for On-Chip Communications

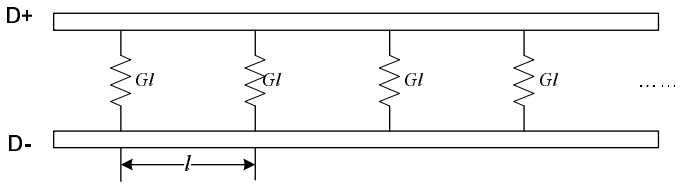


Figure IV.3: On-Chip Implementation of Surfliner

For on-chip wires, we insert the resistors between two wires of a differential pair to realize the leakage conductance G (Fig. IV.3). We periodically insert a leakage conductor with conductance Gl at every interval l in z direction. When the interval l is small enough comparing with the wavelength of the data signal, the discontinuity caused by this discrete resistor insertion scheme is negligible. According to our simulation, when interval $l < \frac{c}{20}t_p$, the jitter caused by ISI is smaller than 5% of clock period, t_p, where, c is the speed-of-light in the dielectrics. For a 15GHz signal, we need to keep the interval l to be smaller than $600\mu m$.

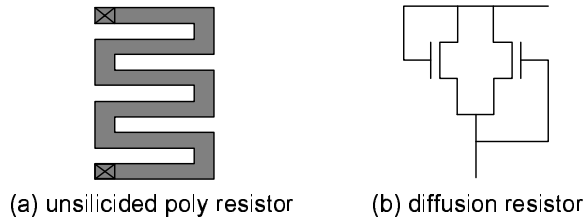


Figure IV.4: Design of shunt conductors

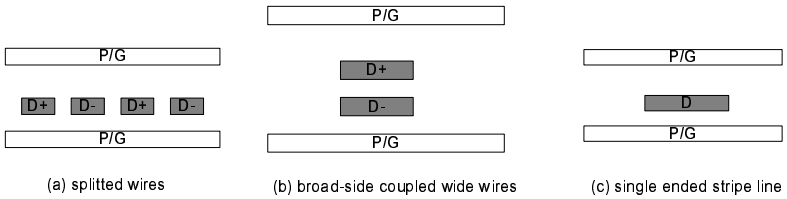


Figure IV.5: Wire configurations to increase the coupling between differential wires

The leakage resistor can be implemented using either unsilicided poly resistor or diffusion resistor (Fig. IV.4). In our implementation, we choose unsilicided poly resistor because it occupies less area. The sheet resistance of unsilicided poly can be as high as 1000ohm/square. We assume 90nm technology, a 3um-width copper wire at metal 6, and 500ohm/square sheet resistance of unsilicided poly. If we insert a leakage resistor every 200um and implement each resistor using poly wires with minimal width (100nm), for a 20mm long link, the leakage conductors only use $126um^2$ of poly area.

For Surfliner, because the attenuation of the signal is proportional to the exponential of the ratio between wire resistance and characteristic impedance, reducing the wire resistance as well as increasing the transmission line characteristic impedance are important to reduce the attenuation. Reducing attenuation can benefit the simplicity of the receiver, the power consumption of the system, and the robustness against the crosstalk and other variations. Different wire configurations, such as the split wires (Fig. IV.5(a)), broad-side coupled wide wires (Fig. IV.5(b)), and single ended stripe lines (Fig. IV.5(c)), can be adopted to optimize under different design specs.

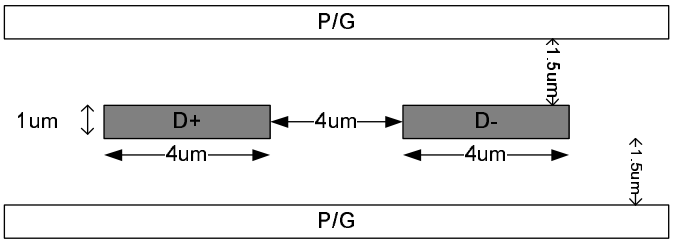


Figure IV.6: Implementation of Wires

IV.D Simulation Results

We implement the surfliner using a pair of edge-coupled stripelines (Fig. IV.6). The copper wires reside on the low resistive upper metal layers, where the wire thickness is $1um$. Each wire has a width of $4um$ and the separation between the wires is also $4um$. The wires are sandwiched by a pair of power/ground shields with separation of $1.5um$. The shields above and below the wires runs in parallel with the wires.

We extract the per unit length resistance, inductance, and capacitance values, R, L, C, of wires using FastHenry and FastCap. In our design, at $15GHz$, $R = 4.4\Omega/mm$, $L = 0.44nH/mm$, and $C = 196.418fF/mm$, the characteristic impedance of the differential pair is $84.5 \times 2 = 169\Omega$. We define the segment of each inserted shunt conductor a stages. We change the number of stages from 4 to 200. For a pair of 2cm long wires, for the distortionless pair, the total leakage conductance is $6.15 \times 10^{-4}S$. The resistance of each resistor ranges from $6.5k\Omega$ to $325k\Omega$. Assuming 1.0V swing level at the output of the sender, the signal amplitude at the input of the receivers is 365mV.

Table IV.1: Jitter and silicon area usage

# Stages	4	10	20	40	80	120	160
Jitter(ps)	27	9.5	5.4	4.2	3.9	2.1	2.08
Area(um^2)	0.52	3.25	13.0	52	208	468	832

For each stage, we use Agilent ADS Momentum to extract the 4-port S-parameter description. Then, we perform the transient analysis of the circuit using

Table IV.2: power consumption w/ different wire width and sepeartion

(width, spacing (um))	(3, 3)	(4, 4)	(5, 4)	(10, 5)
Power (mW)	4.98	3.62	3.02	2.13
Attenuation	0.307	0.415	0.496	0.60

HSpice. We generate $2^{10} - 1 = 1023$ bits pseudo random bit sequences (PRBS) as the input [31]. The initial bit vector of PRBS is 1010101, and the generation polynomial is $x^7 + x + 1$. We set the clock frequency to 15GHz, and the input signal has a transition slope of 10% clock cycle for each rising and falling edge.

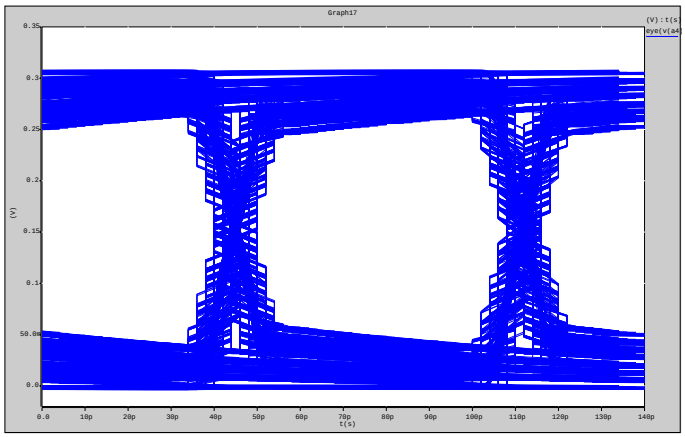


Figure IV.7: Eye diagram of the output voltage for a 4 stage 2cm-length surfliner

We simulate surfliners with different numbers of leakage resistors. Table IV.1 shows the jitter of output voltages and the usage of poly area. When the number of stages increases from 4 to 160, the jitter reduces from 27ps to 2.08ps. The poly area usage increases from $0.52um^2$ to $832um^2$.

Figs. IV.7 and IV.8 show the eye diagrams of the output signal for surfliners with 4 stages and 120 stages, respectively. Both of the cases show clear eye opening. For 4-stage case, we see jitters caused by reflections. (Fig. IV.7). For 120-stage case (Fig. IV.8, the transmission line achieves almost distortionless transportation. The data dependent jitter is only 2.1ps.

We also explore the effect of different configurations of wire geometries. Table IV.2 shows the power consumption and signal attenuation through a 2cm

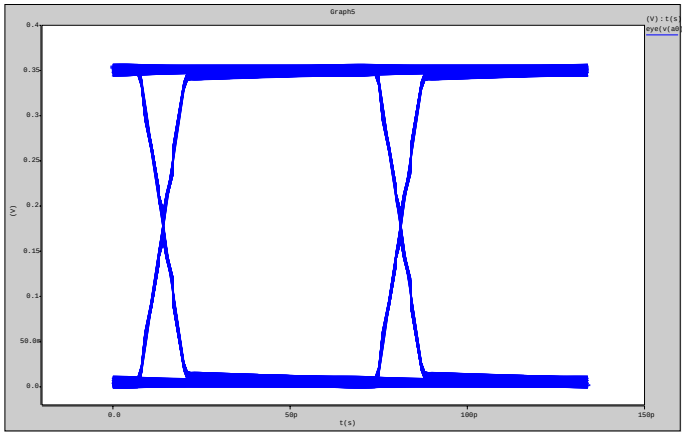


Figure IV.8: Eye diagram of the output voltage for a 120 stage 2cm-length surfliner

long transmission line with different wire widths and separations. The driver is designed to ensure that the signal magnitude at the receiver end is no less than 150mV. The wires are terminated at the receiver ends. When we use wider wires, we can get lower attenuation. As a result, the wider wire requires less power consumption.

IV.E Conclusion

We propose a novel scheme to implement a distortionless transmission line for on-chip global communications. By inserting shunt conductors between two wires of a differential pair, we can achieve near distortionless wave propagations. Our on-going efforts include:

- Design and fabrication of test chip for this new architecture.
- Implement and evaluate wiring splitting schemes shown in Fig. IV.5.
- Investigate novel on-chip interconnect architectures which exploit unique advantages of distortionless transmission lines to realize low-latency low-power on-chip communications.

V

Conclusion and Future Directions

This chapter summarize the research works presented in this dissertation and gives several future directions.

We studied on-chip interconnect architectures in this dissertation.

For Non-Manhattan Routing, we adopted a multi-commodity flow model to describe the on-chip communication traffic. Based on this model, we studied the physical planning of different on-chip routing architectures. Through careful analysis of communication bottlenecks, we found that the Y-architecture (3-directional routing) enjoys a lot of advantages over other routing architectures. We evaluated the Y-architecture through MCF model and explored the design methodologies associated with it, including the power and clock distribution, layer assignment, and an innovative way to handle the via blockage effects.

For high speed clock distribution, we proposed three different solutions to three different classes of chips. The main objectives are reducing the clock uncertainty caused by PVT variations and improving the power efficiency.

For ASICs, which usually has an automatic generated clock tree, we proposed a PVT variations aware clock tree synthesis algorithm. The algorithm construct the clock tree topology considering both the physical proximity of the clock sinks and their timing criticality. Experimental results showed great reduction of timing violations caused by PVT variations with the cost of marginal wirelength

overhead.

For micro-processors, we presented a multi-level mesh architecture for low skew clock distribution. We derived the analytical expression of the skew reduction effect of a clock shunt. Guided by this theoretical results, we used a mathematical programming technique to optimize the topology and size of a multi-level mesh.

For multi-giga hertz circuit, we devised a multi-level transmission line network approach for low uncertainty low power clock distributions. We found that if two clock drivers are connected by a wire with integral multiple of the signal wavelength, the phase and magnitude of two drivers can be synchronized together. Utilizing this phenomena, we design a multi-level network with spiral shaped shunts to achieve high speed low power distribution.

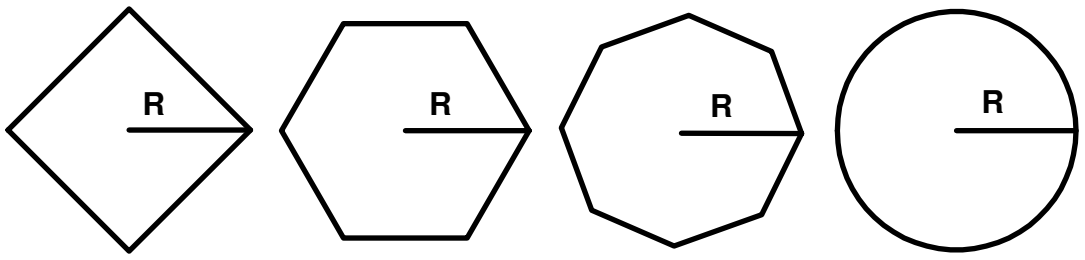
For high performance on-chip interconnect, we proposed a distortionless transmission line scheme for speed of light on-chip communications. With intentionally inserted leakage conductors, the signal can propagate along the wire at the speed of light with no distortion in waveform. This scheme offers much better throughput and lower per bit energy consumption comparing with traditional RC wires with repeaters.

Appendices

Appendix A Analysis of the "Virtuous Cycle" Wirelength Reduction Effect

The simulated annealing placer in Section II.B.3 places cells within a chip that has fixed area. However, reduction of overall wirelength results in decreased routing area, which in turn leads to further wirelength reduction, creating a "virtuous cycle" effect.

Consider a cluster of two-pin nets which are connected to one pin A. All other pins are uniformly located in a circle by a routing-geometry-aware placer. Circles for different routing geometries are shown in Figure A.1. Based on the "virtuous cycle" effect, the circle will have an area proportional to the total routing area. For Manhattan placement and routing, suppose the pins are placed in a



(a) Rectilinear Circle (a) Hexagonal Circle (a) Octilinear Circle (a) Euclidean Circle

Figure A.1: Circles with radius R for each routing geometry (rectilinear, hexagonal, octilinear and Euclidean).

rectilinear circle with radius R. We have the area of the rectilinear circle, $A = 2R^2$, and the total routing area, $A_{\text{routing}} = 4\int_0^R (x \cdot \frac{xdx}{A}N) = \frac{4}{3}\frac{R^3}{A}N = \frac{2}{3}RN$, where N is the number of two-pin nets and $\frac{xdx}{A}N$ is the number of pins located between unit circles with radii x and $x + dx$. Let $A \sim A_{\text{routing}}$. We have $R \sim N/3$ and $A_{\text{routing}} \sim \frac{2}{9}N^2$. Similar analysis can be done for other routing geometries, with the results summarized as follows:

Rectilinear: $A_{\text{routing}} \sim \frac{2}{9}N^2$

Hexagonal: $A_{\text{routing}} \sim \frac{8\sqrt{3}}{81}N^2$, 23.0% less compared to Manhattan placement and routing.

Octilinear: $A_{\text{routing}} \sim \frac{\sqrt{2}}{9}N^2$, 29.3% less compared to Manhattan placement and routing.

Euclidean: $A_{\text{routing}} \sim \frac{4}{9\pi}N^2$, 36.3% less compared to Manhattan placement and routing.

This simple analysis shows that the wirelength reduction caused by the "virtuous cycle" effect is significant, and can partly explain the large wirelength reductions reported in [70] and [85].

Appendix B Approximation of Equation (II.4)

Suppose a current I enters a uniform infinite triangular resistive lattice with edge resistance R at the origin and leaves at infinity. The voltage drop for any node on the lattice is analyzed in [57]. The final result for the voltage drop is expressed as an integral representation. The voltage between $(0, 0)$ and (m, n), $V(m, n)$, is:

$$V(m,n) = \frac{IR}{2\pi}\int_0^{\pi/2} \frac{(1 - e^{-|(m-n)|x}\cos(m+n)y)}{\sinh x \cos y}\, dy, \qquad \text{(App-1)}$$

where $2\cosh x \cos y + \cos 2y = 3$. When $|m-n|$ is large, the exponential term in the above expression become negligible except when x is very small. When x is very small, we have:

- $\cosh x \approx 1 + x^2/2$,
- $\sinh x \approx x$,
- $\cos y = [(8 + (\cosh x)^2)^{1/2} - \cosh x]/2 \approx 1 - x^2/6$, and
- $y \approx x/\sqrt{3}$.

The above expression can be rewritten as the sum of three integrals: $V_{m,n}/IR = I_1 + I_2 + I_3$, where

$$\begin{aligned} I_1 = &\ (1/2\pi)\ \textstyle\int_0^{\pi/2}\ (1 - e^{-|m-n|\sqrt{3}y}\cos(m+n)y)/\sqrt{3}y\ dy \\ I_2 = &\ (1/2\pi)\ \textstyle\int_0^{\pi/2}(1/\sinh x\cos y - 1/\sqrt{3}y)dy \\ I_3 = &\ (1/2\pi)\ [\textstyle\int_0^{\pi/2}\ e^{-|m-n|\sqrt{3}y}\cos(m+n)y/\sqrt{3}y\ dy \\ &\ - \textstyle\int_0^{\pi/2}\ e^{-|m-n|x}\cos(m+n)y/\sinh x\cos y\ dy] \end{aligned} \qquad \text{(App-2)}$$

The first integral can be expressed in terms of the exponential integral $Ein(z)$,

$$Ein(z) = \int_0^z\ [(1-e^{-t})/t]\ dt = \int_0^{\pi/2}\ [(1-e^{-2yz/\pi})/y]\ dy, \qquad \text{(App-3)}$$

so that

$$I_1 = \frac{1}{2\sqrt{3}\pi}\mathrm{Re}\{\ Ein(\ \frac{\pi}{2}[|m-n|\sqrt{3} - i(m+n)]\ )\ \}. \qquad \text{(App-4)}$$

For large values of its argument, $Ein(z) \approx \ln z + c_1$, where $c_1 = 0.57721$. So we have

$$I_1 \approx \frac{1}{4\sqrt{3}\pi}[\ln(m^2+n^2-mn) + 2(\ln\pi + c_1)] \qquad \text{(App-5)}$$

The second integral can be integrated numerically. Let $I_2 = \frac{1}{2\sqrt{3}\pi}\ c_2$, where $c_2 = \int_0^{\pi/2}(\sqrt{3}/\sinh x\cos y - 1/y)dy = 0.09\,772\,.$

The exponentials in **the third integral** are negligible, except for small values of x and y, and for those values, $\sinh x\cos y \approx x \approx \sqrt{3}y$, so the third integral can be neglected.

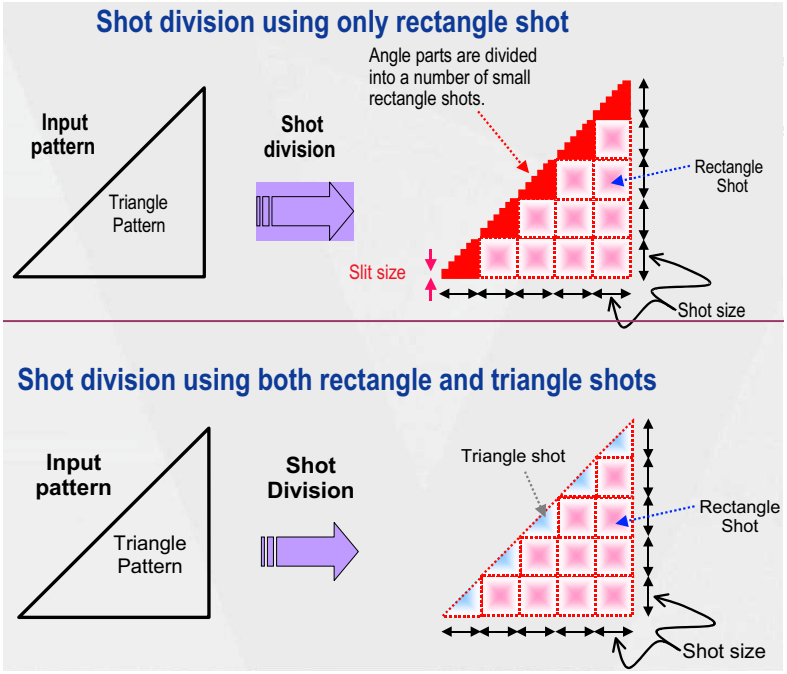


Figure A.2: Toshiba machine triangle shots.

Finally, we have

$$V_{m,n} \approx \frac{IR}{4\sqrt{3}\pi}[\ln(m^2 + n^2 - mn) + c], \qquad \text{(App-6)}$$

where $c = 2(\ln \pi + c_1 + c_2) = 3.6393$.

Appendix C Manufacturing and Other Issues

As is well-known from the example of the X Initiative [86], any new back end of the line (BEOL) architecture requires engagement throughout the mask and process infrastructure. According to our discussions with domain experts [61] [77], the Y-architecture presents a number of generic challenges to manufacturing; there are no show-stoppers, but engineering efforts will be required across several domains. Space limits preclude detailed discussion here, but we sketch several main points.

With respect to mask making, Vector Shaped Beam (VSB) ebeam litho-

graphy tools [56] create "shots" of varying shape and size by imaging the overlap of two apertures, typically both square. This allows a range of rectangular shots to be created and exposed on the mask. Existing Toshiba ebeam lithography systems can produce 45-degree pattern at high speed through the combination of one rectangular aperture and one with 45- and 135-degree edges [86]. The new JEOL JBX3030 tool [73] also has apertures to produce 45- and 135-degree edges. These new tools mitigate the write time implications of angled data since they provide an alternative to approximating an angled line with a series of small rectangles; Figure A.2 illustrates mask fracturing using both rectangle and triangle shots versus mask fracturing using only rectangle shots [86]. With successful experiences with 45-degree edges in mind, 60- and 120-degree edges can be printed with the availability of 30- and 60-degree angles in apertures.

Current support for angular edges is really focused on small edge segments rather than long lines. To produce long lines efficiently, it is necessary to have a pair of rectangular apertures rotated to still produce rectangular shots, but rotated to the desired angle. On the other hand, if the Y architecture is applied only to the upper, lower resolution metal layers - as we have proposed - the write time issue could be solved if the masks could be made with optical (laser) lithography (e.g., ETEC Alta writers), where throughput is independent of angular edges.

The potential of non-rectangular die also presents challenges to package I/O design and dicing. Current side-to-side die sawing cannot cut hexagonal dies due to the silicon lattice structure. New technologies, such as waterjet-guided laser [89], are emerging to confront the challenges.

There are other challenges related to inspection, exposure, repair, metrology and pattern compensation. Ultimately, the deployment of the Y-architecture will depend on careful engineering, and provable cost reductions vis-a-vis achievable design quality with pervasive 60- and 120-degree wiring.

Appendix D Derivation of the skew expression

First, we get the close form expression of V_1 and V_2 by solving differential equation (III.1), without loss of generality, we set $V_{s1} = V_{s2} = 1$:

for $t \leq T$:

$$\begin{cases} V_1 = \frac{1}{2}((1 - e^{-\frac{1}{R_s C}t}) + \frac{1}{1+2\frac{R_s}{R}}(1 - e^{-\frac{1+2\frac{R_s}{R}}{R_s C}t})) & \text{(App-7)} \\ V_2 = \frac{1}{2}((1 - e^{-\frac{t}{R_s C}}) - \frac{1}{1+2\frac{R_s}{R}}(1 - e^{-\frac{1+2\frac{R_s}{R}}{R_s C}t})) & \text{(App-8)} \end{cases}$$

for $t > T$:

$$\begin{cases} V_1 = 1 + K_1 e^{-\frac{1}{R_s C}t} + K_2 e^{-\frac{1+2\frac{R_s}{R}}{R_s C}t} & \text{(App-9)} \\ V_2 = 1 + K_1 e^{-\frac{1}{R_s C}t} - K_2 e^{-\frac{1+2\frac{R_s}{R}}{R_s C}t} & \text{(App-10)} \end{cases}$$

where,

$$\begin{cases} K_1 = -\frac{1}{2}(e^{\frac{1}{R_s C}t} + 1) & \text{(App-11)} \\ K_2 = \frac{1}{2(1+2\frac{R_s}{R})}(e^{\frac{1+2\frac{R_s}{R}}{R_s C}t} - 1) & \text{(App-12)} \end{cases}$$

From equations (App-9) and (App-10), for $t > T$, both V_1 and V_2 have the common term $1 + K_1 e^{-\frac{1}{R_s C}t}$, while the term $K_2 e^{-\frac{1+2\frac{R_s}{R}}{R_s C}t}$ causes the clock skew.

We define t_1 and t_2 to be the clock signal arriving time of node n_1 and node n_2, respectively. In other words, $V_1(t_1) = V_2(t_2) = 0.5$. Hence, the clock skew $\Delta T = t_2 - t_1$.

We assume that the initial clock skew T is much smaller than the clock delay $ln2R_sC$. This assumption is reasonable for most symmetric clock trees with typical design parameters. Based on this assumption, we have $t_1 \approx t_2 \approx ln2R_sC$.

We compute the voltage slew rate of V_2 and voltage difference between V_1 and V_2 at time t_1. By dividing these two numbers, we can get the time V_2 needed to achieve 0.5V. We compute the skew ΔT using following approximation:

$$\Delta T = \frac{V_1(t=t_1) - V_2(t=t_1)}{\dot{V}_2(t=t_1)} \quad \text{(App-13)}$$

$$= \frac{V_1(t=2ln2R_sC) - V_2(t=2ln2R_sC)}{\dot{V}_2(t=2ln2R_sC)} \quad \text{(App-14)}$$

$$= \frac{2K_2 e^{-ln2(1+2\frac{R_s}{R})}}{0.5(-\frac{K_1}{R_sC} + K_2)} \quad \text{(App-15)}$$

$$= \frac{K_2 e^{-2ln2\frac{R_s}{R}}}{\frac{1}{2}(-\frac{K_1}{R_sC} + K_2 \frac{R_sC}{1+2\frac{R_s}{R}} e^{-2ln2\frac{R_s}{R}})} \quad \text{(App-16)}$$

Because $T \ll R_sC$, we have $T/(R_sC) \ll 1$. When $x \ll 1$, we can use first order Taylor's expansion $e^x = 1 + x$ to approximate the value of exponential function e^x. We utilize this approximation to simplify the expression of K_1 and K_2

$$\begin{cases} K_1 = -\frac{1}{2}(2 + \frac{T}{R_sC}) & \text{(App-17)} \\ K_2 = \frac{T}{2R_sC} & \text{(App-18)} \end{cases}$$

Plug (App-17) and (App-18) into (App-16), and omit all of the small terms containing $\frac{T}{R_sC}$, and we get following skew expression:

$$\Delta T \approx \frac{\frac{T}{2R_sC} e^{-2ln2\frac{R_s}{R}}}{\frac{1}{2R_sC}} = T \cdot e^{-2ln2\frac{R_s}{R}} \quad \text{(App-19)}$$

Bibliography

[1] F. E. Anderson, et *al.*, *The Core Clock System on the Next Generation Itanium^{TM} Microprocessor,* in the Digest of Technical Papers, ISSCC 2002 Session 8.5.

[2] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms,* 2nd ed. New York: Wiley, 1997

[3] D. Boning and S. Nassif, *Models of Process Variations in Device and Interconnect,* in Design of High Performance Microprocessor Circuits, Editors: A. Chandrakasan, W. Bowhill, F. Fox, IEEE Press, 2000

[4] P.J. Camporese, et *al.*, *X-Y Grid Tree Tuning Method,* U.S. Patent No. 6205571 B1, March, 2001

[5] C.K. Cheng, J. Lillis, S. Lin, and N. Chang, *Interconnect Analysis and Synthesis,* Wiley Interscience, 2000

[6] M.P. Desai, R. Cvijetic, and J. Jensen, *Sizing of clock distribution networks for high performance CPU chips,* in Prod. of Design Automation Conference, pp.389-394, June 1996

[7] D. Harris, and S. Naffziger, *Statistical Clock Skew Modeling With Data Delay Variations,* IEEE trans. on VLSI, Vol.9, No. 6, pp. 888-898, Dec 2001

[8] M. Kamon, M. J. Tsuk, and J. K. White, *FastHenry: A multipole-accelerated 3-d inductance extraction program,* IEEE Trans. on Microwave Theory and Techniques, 42(9):1750–8, September 1994

[9] N.A. Kurd, et *al.*, *A Multigigahertz Clocking Scheme for the Pentium4 Microprocessor,* IEEE Journal of Solid-State Circuits, Vol. 36, No. 11, pp. 1647-53, Nov. 2001

[10] Y. Liu, S.R. Nassif, L.T. Pilleggi, and A.J. Strojwas, *Impact of Interconnect Variations on the Clock Skew of a Gigahertz Microprocessor,* in Porc. Of Design Automation Conference, pp. 168-171, 2000

[11] V. Mehrotra, *Modeling the Effects of Systematic Process Variation on Circuit Performance,* Ph.D. Thesis, MIT, May, 2001

[12] M. Mori, H. Chen, B. Yao, and C.K. Cheng, *A Multilevel Network Approach for Clock Skew Minimization with Process Variations,* in Proceeding of Asian and South Pacific Design Automation Conference, to appear, Feburary, 2004

[13] M. Orshansky, et *al.*, *Impact of Spatial Intrachip Gate Length Variability on the Performance of High-Speed Digital Circuit,* IEEE trans. on CAD, vol. 21, No. 5, pp.544-553, May 2002

[14] P.J. Restle, et *al.*, *A Clock Distribution Network for Microprocessors,* IEEE Journal of Solid-State Circuits, Vol. 36, No. 5, pp. 792-99, May 2001

[15] P.J. Restle, et *al.*, *The Clock Distribution of the Power4 Microprocessor,* ISSCC 2002, Session 8.4

[16] B.E. Stine, et *al.*, *Simulating the Impact of Pattern-Dependent Poly-CD Variation on Circuit Performance,* IEEE Trans on semiconductor manufacturing, vol. 11, No. 4, pp. 552-556, Nov. 1998

[17] S. Sauter, et *al.*, *Clock skew determination from parameter variations at chip and wafer level,* International Workshop Statistical Methdology, pp. 7-9, 1999

[18] H. Su, and S.S. Sapatnekar, *Hybrid Structured Clock Network Construction,* in Proc. of ICCAD, pp. 333-336, 2001

[19] P. Zarkesh-Ha, T. Mule, and J. Meindl, *Characterization and Modeling of Clock skew with Process Variations*, in Proc. CICC, pp. 441-444, 1999

[20] E. Afshari and A. Hajimiri, "Non-Linear Transmission Lines for Pulse Shaping in Silicon," *IEEE Custom Integrated Circuits Conf.*, pp. 94-94, Sept. 2003.

[21] B. M. Beckmann, and D. A. Wood, "TLC: Transmission Line Caches,", *IEEE Int. Symp. on Microarchitecture*, pp. 43-54, Dec. 2003

[22] M. F. Chang, V. P. Roychowdhury, L. Zhang, H. Shin, and Y. Qian, "RF/Wireless Interconnect for Inter- and Intra-Chip Communications," *Proc. of IEEE*, vol.89, no.4, pp.456-66, Apr. 2001.

[23] R. T. Chang, N. Talwalkar, C. P. Yue, and S. S. Wong, "Near Speed-of-Light Signaling Over On-Chip Electrical Interconnects", *IEEE J. of Solid-State Circuits*, vol. 38, no. 5, pp. 834-838, May 2003.

[24] C. K. Cheng, J. Lillis, S. Lin, and N. Chang, "Interconnect Analysis and Synthesis," Wiley Interscience, 2000.

[25] W. J. Dally, M.-J. E. Lee, F.-T. An, J. Poulton, and S. Tell, "High-Performance Electrical Signaling," *IEEE Int. Conf. on Massively Parallel Processing Using Optical Interconnections*, pp. 11-16, June, 1998.

[26] R. Ho, K. W. Mai, and M. A. Horowitz, "The Future of Wires," *Proc. of IEEE*, vol. 89, no. 4, pp. 490-504, April. 2001.

[27] R. Ho, K. Mai, and M. Horowitz, "Efficient On-Chip Global Interconnects", *IEEE Symp. on VLSI Circuits*, pp. 271– 274, June 2003.

[28] N. M. Jokerst et *al.*, "The Heterogeneous Integration of Optical Interconnections Into Integrated Microsystems," *IEEE J. of Selected Topics in Quantum Electronics,* vol.9, no.2, pp. 350-360, Mar/Apr. 2003.

[29] A. Maheshwari and W. Burleson, "Differential Current-Sensing for On-Chip Interconnects," *IEEE Trans. on VLSI,* vol.12, no.12, pp. 1321-1329, Dec. 2004.

[30] V. Stojanovic, et *al.*, " Adaptive Equalization and Data Recovery in a Dual-Mode (PAM2/4) Serial Link Transceiver,", *IEEE Symp. On VLSI*, pp. 348-351, 2004.

[31] "Agilent N4901 Serial BERT User's Guid", Agilent Technologies, 2004

[32] D. W. Bailey and B. J. Benscheneider, *Clocking Design and Analysis for a 600-MHz Alpha Microprocessor,* IEEE Journalof Solid-State Circuits, Vol. 33, No. 11, pp.1627-1633, November, 1998

[33] K. D. Boese and A. B. Kahng, *Zero-skew Clock Routing with Wirelength Minimization,* in Proc. IEEE Int'l Conf. on ASICs, pp. 17-21, September, 1992

[34] Y. Chen, A. B. Kahng, G. Qu, and A. Zelikovsky, *The Associative-Skew Clock Routing Problem,* in Proc. ICCAD, pp. 168-172, November, 1999

[35] D. Chinnery and K. Keutzer, *Closing the Gap Between ASIC & Custom: Tools and Techniques for High-Performance ASIC Design,* Kluwer Academic Publishers, 2002

[36] N.C. Chou, and C. K. Cheng, "Wire length and delay minimization in general clock net routings," IEEE trans. On CAD, 1996

[37] M. Edahiro, *A Clustering-Based Optimization Algorithm in Zero-Skew Routings,* in Proc. DAC, pp. 612-616, June, 1993

[38] G. Ellis, L. T. Pilleggi, and R. A. Rutenbar, *A Hierarchical Decomposition Methodology for Multistage Clock Circuits,* in Proc. ICCAD, pp. 266-273, November, 1997

[39] J. P. Fishburn, *Clock Skew Optimization,* IEEE Trans. Computers, pp. 945-951, July 1990

[40] S. Guha and I. Suzuki, *Proximity Problems For Points On A Rectilinear Plane With Rectilinear Obstacles,* Algorithmica,vol. 17, pp. 281 - 307, 1997

[41] B. Lu, J. Hu, G. Ellis, and H. Su, *Process Variation Aware Clock Tree Routing,* in Proc. ISPD, pp. April. 2003

[42] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, *Clock Routing for High-Performance ICs,* in Proc. DAC, pp. 573-579, 1990

[43] A. B. Kahng, J. Cong, and G. Robins, *High Performance Clock Routing Based on Recursive Geometric Matching,* in Proc. DAC, pp. 322-327, 1991

[44] A. B. Kahng and C. W. Tsao, *More Practical Bounded-Skew Clock Routing,* in Proc. DAC, pp. 594-599, June 1997

[45] G. Karypis and V. Kumar, *Multilevel Algorithms for Multi-Constraint Graph Partitioning,* Technical Report TR 98-019, Department of Computer Science, University of Minesota, 1998

[46] N. A. Kurd, et *al.*, *A Multigigahertz Clocking Scheme for the Pentium 4 Microprocessor,* IEEE Journal of Solid-State Circuits, vol. 36, No. 11, pp. 1647 - 53, November 2001

[47] M. Mori, H. Chen, B. Yao, and C. K. Cheng, *A Multiple Level Network Approach for Clock Skew Minimization with Process Variations,* in Proc. ASP-DAC, 2005

[48] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams,* Wiley & Sons, 1992

[49] A. Rajaram, J. Hu, and R. Mahapatra, *Reducing Clock Skew Variability Via Cross Links,* in Proc. DAC, pp. 18 -23, 2004

[50] P.J. Restle, et *al.* , *The Clock Distribution of the Power4 Microprocessor,* ISSCC 2002, Session 8.4

[51] R. S. Tsay, *An Exact Zero-Skew Clock Routing Algorithm,* IEEE Trans. On CAD, Vol. 12, No. 2, pp. 242-249, February, 1993

[52] D. Velenis, E.G. Friedman, and M. C. Papaefthymiou, *A Clock Tree Topology Extraction Algorithm For Improving the Tolerance of Clock Distribution Networks to Delay Uncertainty,* in Proc. ISCAS, pp. 4.422-4.425, May, 2001

[53] J. Zejda and P. Frain, *General Framework for Removal of Clock Network Pessimism,* in Proc. ICCAD, 2002

[54] http://www-2.cs.cmu.edu/ quake/triangle.html

[55] http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST/

[56] F. Abboud, S. Babin, V. Charkarian, et al., "Design Considerations for an Electron-Beam Pattern Generator for the 130-nm Generation of Masks", *SPIE Symp. Photomask and X-Ray Mask Technology VI*, SPIE Vol. 3748, 1999, pp. 385-399.

[57] D. Atkinson and F. J. van Steenwijk, "Infinite Resistive Lattices", *Am. J. Phys.* 67 (1999), pp. 486-492.

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

[59] H. B. Bakoglu, J. T. Walker and J. D. Meindl, "A Symmetric Clock Distribution Tree and Optimized High-Speed Interconnections for Reduced Clock Skew in ULSI and WSI Circuits", *Proc. IEEE Int. Conf. Computer Design*, Oct. 1986, pp. 118-122.

[60] S. Boyd, L. Vandenberghe and A. El Gamal, "Design of Robust Global Power and Ground Networks", *Proc. ACM/SIGDA Int. Symp. Physical Design*, 2001, pp. 283-288.

[61] P. Buck, Dupont Photomasks, *personal communication*, Nov. 2002.

[62] H. Chen, B. Yao, F. Zhou and C. K. Cheng, "Physical Planning of On-Chip Interconnect Architectures", *Proc. IEEE Int. Conf. Computer Design*, Sep. 2002, pp. 30-35.

[63] H. Chen, B. Yao, F. Zhou and C. K. Cheng, "The Y-Architecture: Yet Another On-Chip Interconnect Solution", *Proc. Asia and South Pacific Design Automation Conf.*, 2003, pp. 840-846.

[64] H. Chen, C.-K. Cheng, A. B. Kahng, I. Măndoiu and Q. Wang, "Estimation of Wirelength Reduction for λ-Geometry vs. Manhattan Placement and Routing", *Proc. ACM/IEEE Workshop on System Level Interconnect Prediction*, 2003, pp. 71-76.

[65] H. Chen, C.-K. Cheng, A. B. Kahng, I. Măndoiu, Q. Wang and B. Yao, "The Y-Architecture for On-Chip Interconnect: Analysis and Methodology", *Proc. Int. Conf. Computer Aided Design*, 2003, to appear.

[66] H. Chen, C.-K. Cheng, A. B. Kahng, Q. Wang and B. Yao, "Optimal Sizing Analyses for Mesh-Based Power Plans", *unpublished manuscript*, 2003.

[67] A. Dharchoudhury and R. Panda, "Design and Analysis of Power Distribution Networks in POwerPC Microprocessors", *Proc. Design Automation Conf.*, 1998, pp. 738-743.

[68] N. Garg, and J. Konemann, "Faster and Simpler Algorithms for Multicommodity Flow and other Fractional Packing Problems", *Proc. 39th Annual Symp. Foundations of Computer Science*, 1998, pp. 300-309.

[69] T. Hildebrandt, "An Annotated Placement Bibliography", *ACM SIGDA Newsletter*, Dec. 1985, pp. 12-21.

[70] M. Igarashi, T. Mitsuhashi, A. Lee, et al., "A Diagonal-Interconnect Architecture and Its Application to RISC Core Design", *Proc. Int. Solid-State Circuits Conf.*, 2002, pp. 166-167.

[71] A. B. Kahng, I. I. Măndoiu and A. Z. Zelikovsky, "Highly Scalable Algorithms for Rectilinear and Octilinear Steiner Trees", *Proc. Asia and South Pacific Design Automation Conf.*, 2003, pp. 827-833.

[72] C. K. Koh and P. H. Madden, "Non-Manhattan Routing", *IEEE Trans. Computer-Aided Design*, to appear.

[73] M. Lemke, J. Gramss, H. J. Doering, et. al., "Advanced Writing Strategies for High-End Mask Making", *Proc. SPIE*, Vol. 3996, 2000, pp. 166-172.

[74] P. H. Madden, "Congestion Reduction in Traditional and New Routing Architectures", to appear.

[75] Carl Sechen, "Placement and Global Routing of Integrated Circuits Using Simulated Annealing", Ph.D. Dissertation, U. California, Berkeley, 1987, Chapter 2.

[76] B. K. Nielsen, P. Winter and M. Zachariasen, "An Exact Algorithm for the Uniformly-Oriented Steiner Tree Problem", *Proc. 10th European Symp. Algorithms*, Springer LNCS Vol. 2461, 2002, pp. 760-772.

[77] C. Progler, Photronics Inc., *personal communication*, Nov. 2002.

[78] M. D. Rostoker et al., "Hexagonal Architecture", *U.S. Patent*, No. US6407434B1, June 2002.

[79] M. D. Rostoker et al., "CAD for Hexagonal Architecture", *U.S. Patent*, No. US5822214, Oct. 1998.

[80] R. Scepanovic et al., "Microelectronic Integrated Circuit Structure and Method Using Three Directional Interconnect Routing Based on Hexagonal Geometry", *U.S. Patent*, No. US5578840, Nov. 1996.

[81] P. Saxena, N. Menezes, P. Cocchini and D. A. Kirkpatrick, "The Scaling Chanllenge: Can Correct-by-Construction Design Help?", *Proc. Intl. Symp. Physical Design*, 2003, pp. 51-58.

[82] D. Stroobandt and J. V. Campenhout, "Accurate Interconnection Length Estimations for Predictions Early in the Design Cycle", *VLSI Design, Special Issue on Physical Design in Deep Submicron* 10(1) (1999), pp. 1-20.

[83] S. Teig and J. L. Ganley, "Method and Apparatus for Considering Diagonal Wiring in Placement", *Int. Patent Application*, No. WO 02/47165 A2, June 2002.

[84] TSMC 0.13μm Design Rules. http://www.tsmc.com.

[85] S. Teig, "The X Architecture", *Proc. ACM/IEEE Workshop on System Level Interconnect Prediction*, 2002, pp. 33-37.

[86] http://www.xinitiative.org.

[87] Intel Research Webpage on Packaging.
http://www.intel.com/research/silicon/packaging.htm.

[88] The ITRS Assembly and Packaging roadmap.
http://public.itrs.net.

[89] WaterJet-Guided Laser In Wafer Cutting – Synova SA.
http://www.gemcity.com/downloads/synova01.pdf.

[90] http://www.itrs.net/Common/2004Update/2004Update.htm