

國立交通大學

電信工程學系

碩士論文

LPGC: 一個基於最佳閘控制時脈拓樸結構之新穎的低功
耗電路擺置演算法

LPGC: A Novel Low Power Driven Placement Algorithm Based on
Optimal Gated Clock Topology

研究生：陳逸宏

指導教授：李育民 教授

中華民國九十四年七月

LPGC: 一個基於最佳閘控制時脈拓樸結構之新穎的低功耗電路擺置演
算法

LPGC: A Novel Low Power Driven Placement Algorithm Based on
Optimal Gated Clock Topology

研究生：陳逸宏

Student : Simon Yi-Hung Chen

指導教授：李育民

Advisor : Yu-Min Lee



Submitted to Department of Communication Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Communication Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

LPGC: 一個基於最佳閘控制時脈拓樸結構之新穎的低功耗電路擺置演算法

研究生：陳逸宏

指導教授：李育民 博士

國立交通大學電信工程學系碩士班

摘 要

本論文研製之電路擺置器，乃基於將用於低功耗設計下之閘控制時脈網路之功率消耗最佳化。首先，基於考慮時脈網路之實體設計連線關係及將其切換機率最小化，先建製一個最佳化閘控制時脈拓樸結構。再透過新穎的評估函數，本論文研製之擺置器可提供讓最佳化閘控制時脈拓樸結構成為可繞結構之適當電路擺置。本擺置器可提供專為低功耗時脈網路繞線演算法之專用電路擺置，這是因為本擺置器提供該演算法將積體電路連線功耗最小化之全域最佳化解。積體電路連線之功耗是一項在奈米製程技術下的嚴重問題。實驗結果證明本擺置器可將閘控制時脈網路功耗減少至少 25%。

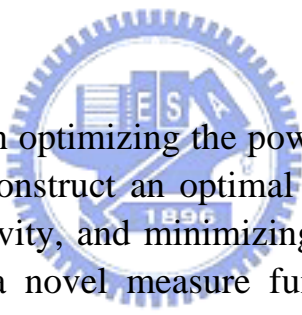
LPGC: A Novel Low Power Driven Placement Algorithm Based on Optimal Gated Clock Topology

Student : Simon Yi-Hung Chen

Advisor : Dr. Yu-Min Lee

Department of Communication Engineering
National Chiao Tung University

ABSTRACT



In this thesis, a placer based on optimizing the power consumption of clock gating network is presented. First, we construct an optimal gated clock topology based on considering the physical connectivity, and minimizing the total switching activity of the clock network. Then, with a novel measure function, our placer can build a suitable placement solution which makes the optimal gated clock topology being routable. Our placer can give a dedicated placement seed to any low power gated clock routing algorithm since we provide them a global optimized solution for reducing the interconnect power which is a severe factor in the coming nano-scale era. Experimental results show that our placer indeed works and can help gated clock routing algorithms to additionally save power consumption by 25%.

誌 謝

本篇論文得以順利完成，首先要感謝指導教授李育民博士，當研究出現困難之時，李老師總是給予相當多的建議，在關鍵的時刻，也總是由李老師扮演推手，將研究中不足的地方加強。

其次，我由衷的感謝實驗室與我一起奮鬥的朋友們，飛鴻、震軒、阿琅與小宇，有各位的陪伴，才能讓我在交大的學習生活更有效率，更多彩多姿。

最後，我將最誠摯的謝意獻給一直支持我的家人，這包含了我的父母、我即將就讀交大研究所的弟弟、我深愛的女友芝婷，因為有你們無悔的付出與關懷，我才能從低潮中振作起來。因為你們的陪伴，我的喜悅與快樂才得以分享。



Contents

1	Introduction	1
1.1	Introduction	1
1.2	Our Contributions	3
1.3	Organization of this Thesis	4
2	Preliminaries and Literatures Survey	5
2.1	Basic Concepts of General Placement	5
2.1.1	Problem Formulation	5
2.1.2	Wire-length Estimation	7
2.2	Previous Work Related to Low Power Placement	8
2.3	Basic Concept of General Clock Routing	11
2.3.1	Elmore Delay Model	12
2.3.2	Clock Timing Considerations	12
2.3.3	Problem Formulation	16
2.4	Previous Work Related to General and Low Power Clock Routing	17
2.4.1	Zero-skew Clock Routing	17
2.4.2	Low Power Zero-skew Clock Routing	22
3	Activity analysis and optimal gated clock topology	26
3.1	Activity Analysis	26
3.1.1	Extraction from High Level Description of the System	27
3.1.2	Instruction Level Simulation	27
3.2	Optimal Gated Clock Topology	29
4	LPGC Placer	33
4.1	Problem Formulation	33
4.2	Data Structrue	34
4.3	LPGC Novel Gated Clock Network Construction Flow	34
4.4	The Novel Low Power Placement Algorithm	36
4.4.1	The Optimal Gated Clock Topology Construction	37
4.4.2	Approaching to Optimal Gated Clock Topology and Pseudo Clock Router Heuristic	38
5	Experimental Result	45
6	Conclusions and Future Work	50

List of Figures

2.1	Comparison between two placements. (a) The placement with shorter total wire-length. (b) The placement with longer total wire-length.	6
2.2	Models of wire-length estimation. (a) Steiner tree. (b) Spanning tree. (c) Semi-perimeter. (d) Complete graph.	8
2.3	(a) Clock sinks' locations for general placement. (b) Gated clock routing under general placement. (c) After performing the existed low power driven placement algorithm, the skew problem becomes much more severe (under optimal gated clock topology). (d) Dedicated low power placement and gated clock routing.	9
2.4	(a) Elmore delay model for a segment of wire. (b) Elmore delay model for a driver.	13
2.5	A typical synchronous system.	14
2.6	A simple diagram which demonstrates clock skew effect.	16
2.7	Method of mean and median clock routing algorithm.	19
2.8	Geometric matching based clock routing algorithm.	21
2.9	A simple example of tiled rectangle region (TRR).	22
2.10	A tree of merging segments. Solid lines are merging segments and dotted lines indicate edges between merging segments. The squares represent the clock sinks.	23
3.1	Methodology for transformation of an ASIC function from hardware description language to a control-data flow graph. (a) Transformation from a simple ASIC function to hardware description language. (b) A simple example of CDFG after performing HDL scheduling and allocation.	28
3.2	A typical topology of gated clocking circuitry.	30
3.3	An example for switching activities comparison under two different clock topology. (a) Topology with smaller average switching activity. (b) Topology with larger average switching activity.	31
4.1	B*-tree data structure. (a) An eleven blocks B*-tree representation. (b) Modules' actual positions and placement of (a).	35
4.2	(a) Traditional design flow of the gated clock network construction. (b) The LPGC novel flow for gated clock network construction. The dotted rectangle circles the operation flow of <i>LPGC Placer</i>	36
4.3	Optimal gated clock topology construction procedure.	38
4.4	Gated clock placement cost computation procedure for pseudo clock router heuristic.	40

4.5	(a) A typical placement result under general constraints, the optimal topology seems to be unroutable. (b) A dedicated placement which makes the optimal topology routable.	40
4.6	Evaluate a placement through pseudo clock router heuristic. (a) Longer pseudo clock wire-length. (b) Shorter pseudo cLock wire-length.	42
4.7	(a) A dedicated placement with considering the optimal gated clock topology. (b) A typical placement without considering the optimal gated clock topology.	44
5.1	(a) Gated clock network comparison between a general placer and <i>LPGC Placer</i> with additionally consideration on gated clock routing. (b) Additional gated clock network Wire-length Saving. (c) Additional gated clock control network wire-length saving. (d) Reduction rate comparison.	47
5.2	Snapshot of the placement result on circuit plbench576.	48
5.3	Snapshot of the placement result on circuit plbench3136.	49



List of Tables

3.1	A simple RTL description of instructions for a microprocessor.	29
3.2	Instruction level simulation for 15 Clock Cycles.	29
5.1	Comparison between typical placer with <i>LPGC Placer</i> on benchmark circuits	48



Chapter 1

Introduction

1.1 Introduction

The need for low power driven VLSI designs has emerged rapidly in past ten years because of the growing demands for the high-performance portable computing devices. Specifically, the saving of power dissipation for the portable electronics is extremely important to extend the battery life and improve device usability.

Modern digital systems are designed with a target clock period (or clock frequency) which determines the rate of data processing. The clock network distributes the clock signal from the clock generator/source to the clock inputs/sinks of the synchronous modules. As we know, the clock distribution network consumes a large portion (more than 40%) [19] of the total circuit power. Therefore, the design of a low power synchronous system tends to minimize the total power consumed by the clock tree subject to performance constraints, such as the operating frequency and the maximum clock skew. Clock gating [19]–[26] is a well-known technique to reduce the dynamic clock network power consumption of a digital circuit. It disables portion of the clock distribution network to save the dynamic power. The existing gated clock routing algorithms first formulate a specifically objective function based on the switching activities, capacitances of clock sinks, gated clock control circuits, the wire length of clock network and gated clock control network, then optimize the power consumption through different gated clock routing algorithms.

According to [21], most of the publications [20, 22, 23] have suffered from the same shortcoming that there is no deeply physical level consideration while performing the gated clock routing algorithm. As the result, the clock tree constraints might not be met or the clock power dissipation might go up after the gated clock routing. Furthermore, the trend of continuously increasing interconnect power is another more severe problem which has been pointed out by [19]. Therefore, we need to pay more effort in reducing the interconnect power to save more power. These facts motivate us to develop a dedicated placement algorithm for reducing the power dissipation of clock distribution network. Placement/Floorplanning is the first stage of the VLSI physical layout design. Several techniques [5]–[10] have been developed to reduce the power consumption at this level, but none of them cooperated with the technique of clock gating. They only optimized dynamic switching power through different approaches with consideration on the combinational networks. This thesis revisits the low power driven placement problem with a focus on gated clock and provides a novel placer to reduce the interconnect power of the gated clock network and gated clock control network. In other word, we can provide a good seed for those gated clock routing algorithms [20]–[23].

The power consumed by complementary metal-oxide-semiconductor (CMOS) circuits consists of two components: dynamic and static power. The static power is mainly determined by the technology. In this thesis, we only consider the dynamic power. The dynamic power consumed by a synchronous module is $\alpha f V_{dd}^2 C_L$, where α is the circuit activity, f is the clock frequency, V_{dd} is the supply voltage, and C_L is the total load capacitance of the circuit. Therefore, a given synchronous CMOS circuits with fixed clock frequency, supply voltage, and total load capacitance, the power consumption can be reduced by decreasing its total activity.

The effects of interconnect such as signal integrity, transmission line effect, and I-R drop ··· etc, are revealed in the dep-submicron circuit design, and those effects become more serious because of the continuously shrinking of feature size. Furthermore, the power dissipation of interconnects gets closer to the loading power as the line is long

enough. Hence, the clock network power consumption is no longer dominated by the loading registers. Therefore, the design method need to be modified to deal with those effects. This thesis develops a novel way to optimize the power consumption of the gated clock network.

1.2 Our Contributions

In this thesis, a *low power driven placer* based on the optimal *gated clock* topology is developed to minimize the power consumption of clock network with taking the total wire-length and area occupation of each module into account. We call this placer as “*LPGC Placer*”. Firstly, the activity pattern is acquired through the analysis of functional blocks, and the optimal gated clock topology is constructed by minimizing the total activity. After that, by using a simulated annealing based optimization algorithm with concurrently evaluating the routability of optimal gated clock topology, the blocks are placed to their suitable positions. Finally, we modify the low power clock routing algorithm [20, 23], and compare our results with the general placement without considering the optimal gated clock topology.

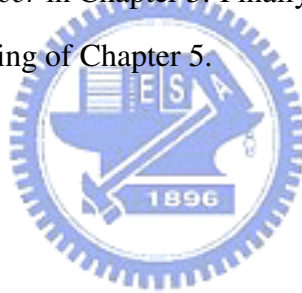
The major contribution of this thesis is to provide a good seed for constructing the low power gated clock routing network. Therefore, the low power gated clock routing algorithms are more likely to find the global optimal solutions instead of the local optimal solutions. We also develop an efficient technique to estimate the total wire length of clock network at the placement level, and utilize this technique to test the routability of the optimal gated clock topology.

The experimental results are excellent. With feeding the placement of the proposed placer into a gated clock routing algorithm [20]–[23], this gated clock algorithm can save more than 25% of the clock power consumption compared with its routing result by using the placement of a general placement algorithm [48], at the cost of the total wire-length increase of 1.3% on average.

1.3 Organization of this Thesis

The remainder of this thesis is organized as follows. First, in Chapter 2, the basic ideas of circuit placement and clock network distribution will be introduced. The related literatures of low power design will also be surveyed thoroughly in Chapter 2. The activity analysis method which we utilize to obtain the precisely distribution of modules' activity and the optimal gated clock topology will be described in Chapter 3.

In Chapter 4, the techniques constructing the *LPGC Placer* will be presented. In order to demonstrate our *LPGC Placer*, both the placement results from *LPCG Placer* and a general placement method without considering the gated clock topology will be fed into a real low power gated clock router [20, 23] to find out the extra power consumption saving percentage of *LPGC Placer* in Chapter 5. Finally, the conclusions and future work will also be given in the remaining of Chapter 5.



Chapter 2

Preliminaries and Literatures Survey

In this chapter, the basic concepts of placement and clock routing problems will be presented, and the previous related researches will be reviewed.

2.1 Basic Concepts of General Placement

Placement is a crucial step in physical design cycle. A poor placement results in larger area occupation, performance degradation, higher power consumption, and even the unevenly thermal distribution. A poor placement also leads to a difficult or impossible routing task. The input of the placement phase is a set of blocks, the information of terminals of each block, and the netlist which describes the connectivity of all modules.

The placement has a deeply impact on the performance of overall layout because an ill-placed layout cannot be improved by any high quality routing algorithm. In other word, the overall quality of the layout such as area, power, total wire-length or thermal distribution is mainly determined in the placement phase. For example, Fig. 2.1 shows two different placements of a circuit, you can see that Fig. 2.1(a) is better than Fig. 2.1(b) since the total wire-length of Fig. 2.1(a) is seem to be shorter than Fig. 2.1(b)'s.

2.1.1 Problem Formulation

Given a set of circuit modules, a placement problem is to arrange modules on the layout surface to satisfy some prescribed constraints such as non-overlapping criterion or circuit boundary constraint, and to optimize certain objective functions such as total area,

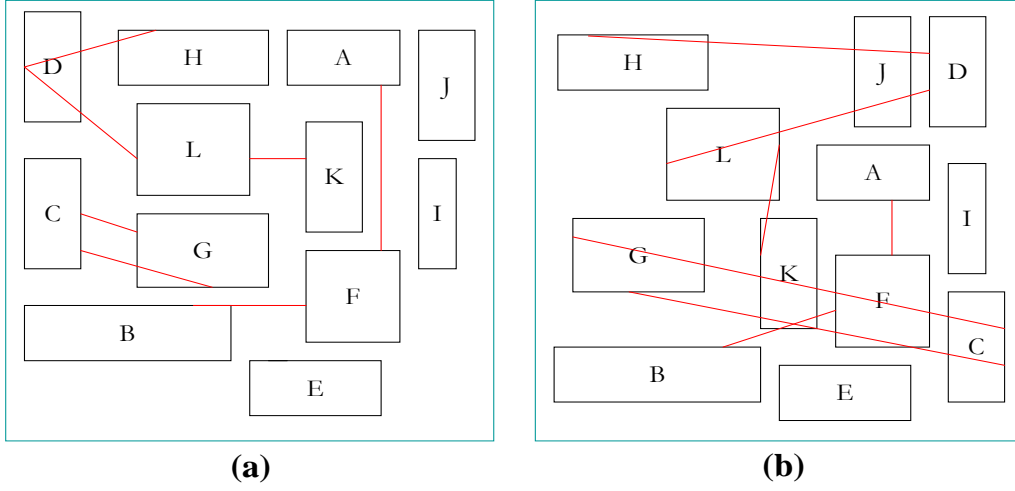


Fig. 2.1: Comparison between two placements. (a) The placement with shorter total wire-length. (b) The placement with longer total wire-length.

total estimated wire-length, total power consumption or/and thermal balancing. At the placement level, a circuit is considered as a group of rectangular modules with fixed geometric ratios. The interconnect relationships among each modules are described in the netlist of interconnect. The different operations such as rotate, flip, move, and swap can be performed on the modules to achieve an optimal placement according to different cost functions.

The formal definition of placement problem is as follows.

Given a set of modules $M = \{m_1, m_2, \dots, m_n\}$, a set of signals $S = \{s_1, s_2, \dots, s_k\}$, and a set of slots/locations $L = \{L_1, L_2, \dots, L_p\}$ where $p \geq n$, S_{m_i} is a set of signals with $S_{m_i} \subseteq S$ for each module m_i , and M_{s_i} is a set of modules with $M_{s_i} = \{m_j \mid s_i \in S_{m_j}; \forall m_j \in M\}$ for each signal s_i . M_{s_i} is called a signal net. The placement problem is to assign each $m_j \in M$ to an unique location L_j such that some objective function, $f(M, S, L)$, is

optimized. Sometimes a subset of modules in M is pre-assigned to several specific locations.

The placement problem is extended to be the floorplanning problem if the geometries or aspect ratios of given modules are more flexible. For example, the modules can be rotated, their shapes can be modified, or the locations of pins can be changed. It is obvious that floorplanning is a generalized version of placement problem, and has higher flexibility to place the modules than the placement's. Of course, it is more complicated.

2.1.2 Wire-length Estimation

The first objective of the high performance integrated circuits is to maximize the operating frequency or to minimize the total delay of the whole system. This objective can be complete by minimizing the length of the critical nets. Usually, it could be approximated by minimization of the length of the longest signal wire-length. After this approximated approach, the objective for maximizing the operating frequency becomes a simpler problem.

Although the real signal wire-lengths are not available at placement level, a placer needs to model the actual topology of the interconnection effectively and accurately. By using the *interconnection graph structure*, the requirements could be satisfied. The interconnection graph structure for modeling the connection between two terminals is simply an edge between the two vertices corresponding to the terminals. For the purpose of modeling a net with more terminals, *rectilinear steiner trees* are used as shown in Fig. 2.2(a) to estimate the optimal signal paths for a net. This model is seldom used by placers, because of the *NP-completeness* of steiner tree problem.

As a result, minimum spanning tree (shown in Fig. 2.2(b)) and semi-perimeter representations (shown in Fig. 2.2(c)) are the most two commonly used models to represent the connections of a net in the placement level. In the *Minimum spanning tree* model, connections are allowed to branch only at the pins. The other one is *semi-perimeter representation*, it can estimate the wire-length effectively by bounding pins with a rectangle

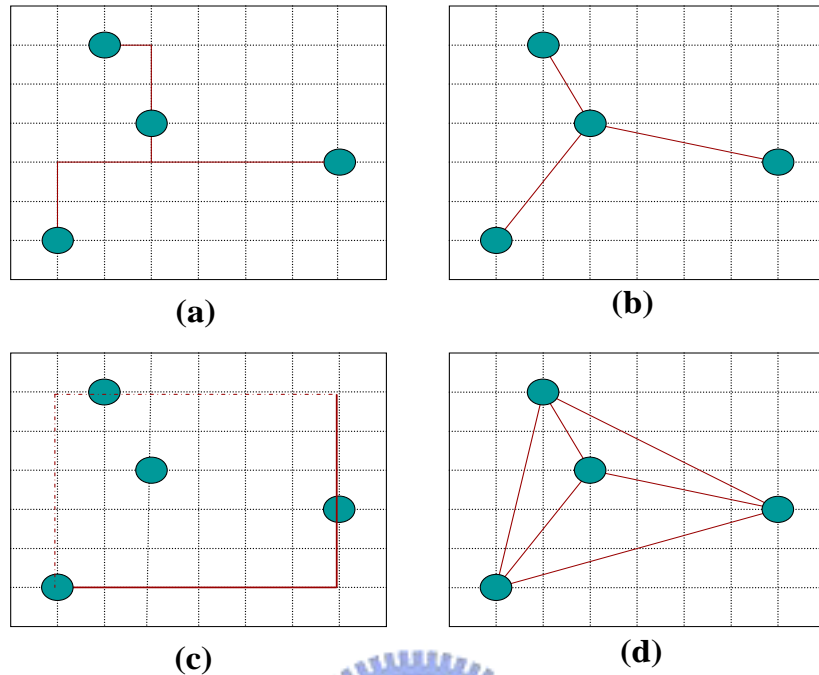


Fig. 2.2: Models of wire-length estimation. (a) Steiner tree. (b) Spanning tree. (c) Semi-perimeter. (d) Complete graph.

then calculates its semi-perimeter.

Complete graph interconnection is shown in Fig. 2.2(d). This method causes many redundant wires, and result in longer wire length. The purpose of such model is to give one the upper bound of a net's wire-length.

2.2 Previous Work Related to Low Power Placement

As we mentioned before, placement/floorplanning is the first stage of the VLSI physical layout design methodology. Several techniques [5]–[10] have been developed to reduce the power consumption at this level, but none of them cooperated with the technique of clock gating. They only optimized dynamic switching power through different approaches with consideration on the combinational networks.

Generally speaking, the total power consumption, P_m , in a typical CMOS circuit can

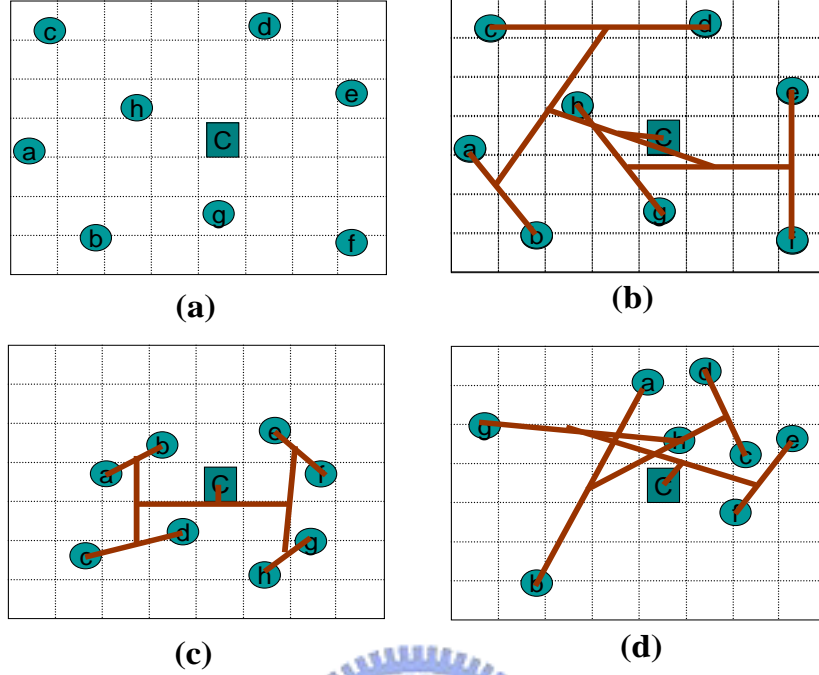


Fig. 2.3: (a) Clock sinks' locations for general placement. (b) Gated clock routing under general placement. (c) After performing the existed low power driven placement algorithm, the skew problem becomes much more severe (under optimal gated clock topology). (d) Dedicated low power placement and gated clock routing.

be formulated as follow [5]

$$\begin{aligned}
 P_{total} &= \sum_{\forall \text{module } m} P_m & (2.1) \\
 P_{wire} &= \alpha_{wire} C_{load,wire} f V_{dd}^2 \\
 P_m &= \sum_{gate \in m} \left\{ \frac{V_{dd}^2}{R_{d,gate}} + \alpha_{gate} C_{load,gate} f V_{dd}^2 \right\} + \sum_{wire \in m} P_{wire},
 \end{aligned}$$

where P_{total} is the total power consumption, P_m is the power consumption of *module* m , V_{dd} is the supply voltage, $R_{d,gate}$ is the equivalent depletion resistance of a gate, $C_{load,gate}$ is the total loading capacitance that a gate drives, P_{wire} is the power consumption of a wire, $C_{load,wire}$ is the wire capacitance, f is the clock frequency, α_{wire} is the activity of a wire, and α_{gate} is the activity of a gate.

The existing low power driven placement methods [5]–[10] optimized the following cost function through different approaches since they believed that this cost function was

the only one related to the power consumption at the placement level.

$$Cost = \sum_{\forall net\ i} \alpha_i C_i^{wire}, \quad (2.2)$$

where α_i is the activity of *net i*, and C_i^{wire} is the wire capacitance of *net i*. Optimizing the above cost function would tend to shrink the interconnect wire length of high switching activity modules. However, the form of Equation (2.2) can not be directly applied to the low power driven placement with considering the clock routing since the clock routing problem and the general routing problem are completely different [27]. The reason is that the clock distribution inherently consists skew problem while the general interconnect network does not.

The skew problem becomes worsen for gated clock routing. Suppose we have a gated clock topology which minimizes the average activity. As we will see in the Section 3.2, we define this topology as *optimal gated clock topology*. Suppose we could make this topology routable then we will save the additional power. For an example, given a distribution of clock sinks under general placement as shown in Fig. 2.3(a), pin C is the clock source, pins a, c, d, e, f and h are very high activity clock sinks, and pins b and g are very low activity clock sinks. Fig. 2.3(b) is a gated clock routing after performing the general placement. We say this solution is feasible since the skew problem is less. But it depends on luck since it places modules in a uniformly distribution. However, a dedicated low power placement for gated clock routing could save more as we could see from Fig. 2.3(c) under minimizing skew problem.

If we adopt the existing low power placement methods would probably be placed as Fig. 2.3(d), which make skew problem much worse. If we want to solve the problem in Fig. 2.3(b) and Fig. 2.3(d), one might re-synthesis the clock topology hence in order to minimize the skew. Even we can handle the problem of skew, the overhead may increase the total clock wire-length. Therefore, the total clock network power consumption will actually be increased. Specifically, this solution is not good since a dedicated placement could replace it as it says in Fig. 2.3(c).

To the best knowledge of the authors, all existing gated clock routing algorithms are restricted by the placement. In other word, they couldn't achieve minimum power consumption unless they have a “dedicated seed at placement level for the the gated clock routing.

2.3 Basic Concept of General Clock Routing

In a typical synchronous system, its performance is directly proportional to the clock operational frequency since it handles more instructions or commands within a same duration. Clock signal is global in nature and the length of a clock line is very long. Hence, the clock network is accompanied by great capacitances so that it limits the performance of the whole system. Therefore, clock nets should be routed with great precision, since the actual length of the path of a net from its entry point to its terminals determines the maximum operation frequency for a chip. By the way, a clock router needs to take several factors into account, including the resistance and capacitance of the routed wire, the noise and cross-talk between wires, the type of load to be driven, the arrival time from clock source to sinks and the total clock network's power consumption. Among all the clock routing requirements, the most important factor is “*clock skew*”. The clock signals must arrive simultaneously at all functional units with slightly or no waveform distortion.

Because of the importance of clock network, optimization of the clock signal can have a significant impact on the chip's cycle time, especially in high-performance VLSI designs. Non-optimal clock behavior is caused by either of two phenomena: the routing to the chip's synchronizing elements (flip-flops or registers), or in the non-symmetric behavior of the clock distribution. Therefore, specialized routers are required for clock nets due to strict specifications for routing such network. Before the introduction of these algorithms, we discuss some fundamentals of clock network such as delay model, timing and clock skew.

2.3.1 Elmore Delay Model

The well-known and powerful model for wire delay estimation is *Elmore delay model* [47]. Let's see Fig. 2.4(a), a Π (pi) model is used to model a wire segment. A via is also modeled as a Π model, with R and C are twice of a wire segment.

The source and all sinks of a net are treated as drivers (shown in Fig. 2.4(a)) which can be modeled by a capacitance connected to the upstream and a resistance connected to the downstream. Calculation of Elmore delays is the sum of *resistance* \times *downstream capacitances*.

Let T be an RC tree, with vertices $V = \{v_1, v_2, \dots, v_n\}$, c_k is the capacitance of a vertex v_k , r_k is the resistance of the edge between v_k and its immediate predecessor. The subtree capacitance at node k is given as

$$C_k = c_k + \sum_{i \in S_k} C_i \quad (2.3)$$

where S_k is the set of all the immediate successor of v_k . Let $\delta_{m,n}$ be the path between v_m and v_n , excluding v_m and including v_n . Then the delay between two nodes m and n is

$$t_{mn} = \sum_{n \in \delta_{m,n}} r_n C_n \quad (2.4)$$

2.3.2 Clock Timing Considerations

The majority of digital chips are synchronous in nature. Synchronous designs are often modeled as a *Moore finite state machine* for the purpose of ease for analysis. The topological requirement imposed on such a finite state machine is that all closed signal paths must contain at least one synchronous element. Satisfaction of this constraint has several benefits, two of which are: the assurance of deterministic behavior if the physical aspects of the design are correct, and the elimination of the requirement that the combinational logic be free of transient as long as next state sampling is performed after the longest path has settled to its final value.

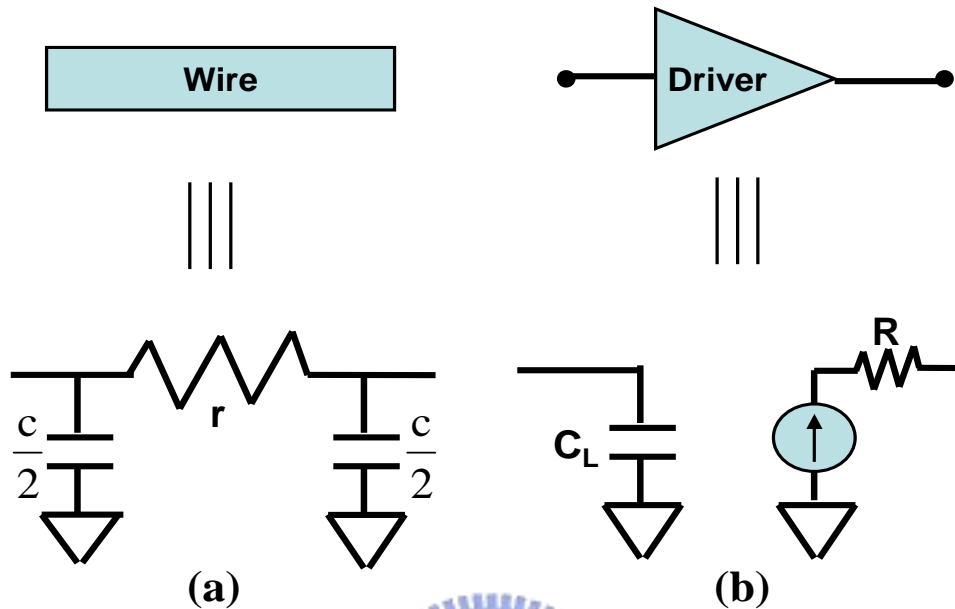


Fig. 2.4: (a) Elmore delay model for a segment of wire. (b) Elmore delay model for a driver.

Clock Period Constraint

For simplicity, and without loss of generality, let the synchronizing elements be edge-triggered. Furthermore, let CP denotes the clock period, d_L the longest path delay through the combinational logic (longest combinational logic path delay), t_{SKEW} the clock skew, t_{SU} the set-up time of the edge-triggered synchronizing elements, and t_{CQ} the delay from the synchronizing element's clock pins to its out pins. In order to guarantee that no long-path timing violations occur in the design, the following equation must be satisfied

$$CP \geq d_L + t_{SKEW} + t_{SU} + t_{CQ} \quad (2.5)$$

This expression demonstrates the important relationship between the clock period, the longest path delay, and the clock skew.

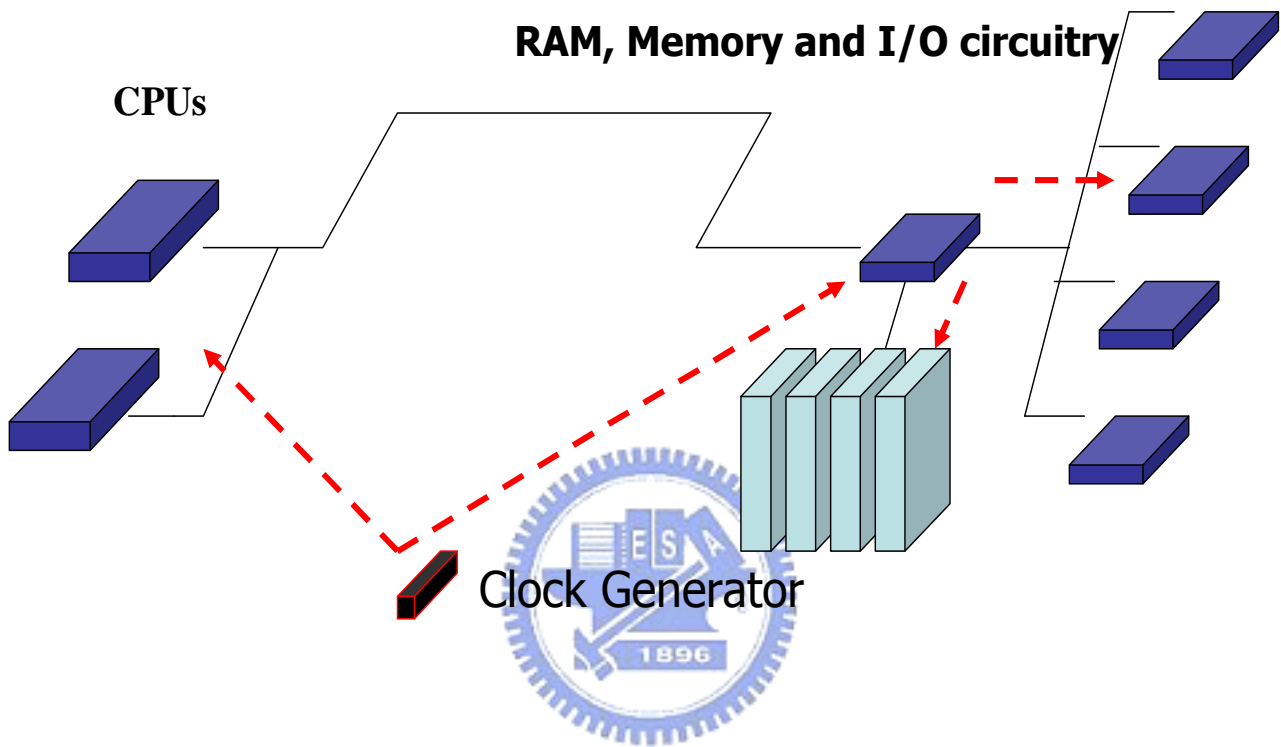


Fig. 2.5: A typical synchronous system.

Clock Skew

Fig 2.5 demonstrate a typical synchronous system, dotted lines represent clock path. The clock signal is often generated external to the chip and supplied to the chips through the clock source pins. All chips would have their own clock network distribution by some dedicated clock router in order to transmit clock signals from clock source pin to every functional unit which needs clock signal. Hence, clock controls the flow of data transmission within the synchronous system. In this figure we omitted the internal clock networks for a brief introduction.

Designers hope that the arrive times from clock source pin to all the functional units are completely and precisely the same so that all data transmissions can start at the same time without data stall and the all functional units can be designed faster to achieve maximum performance.

Actually, the clock signals do not arrive at all functional units simultaneously. We give the definition for the maximum difference from the clock source to each of the clock pins *clock skew*, just as Fig.2.6 demonstrate. With the clock skew limitation, we can not design our system ideally. In other word, one should first calculate the budget or the upper bound for the maximum clock skew can affect. As skew increases with the clock period held fixed, the efficiency of the digital system is reduced because valuable computation time is “stolen” from the total cycle time. Frequently in high performance design environments, skew is constrained to be less than five percent of the clock period. Thus, in a 1 Ghz design, skew would be less than 5000ps. However, there is still some clock skew that we cannot solve which comes from manufacturing process. It forces us to be conservative and use little bigger clock period in order to minimize the skew effect so that we sacrifice the maximum frequency from malfunction comes from clock skew.

We adopts the period which allows for the maximum completion time of all the functional units' in addition to the extra time for handling the maximum clock skew as the system clock period. As we know, we should minimize the clock period as hard we can so that we should evaluate for the second term in Equation (2.5) which represent the bound of the clock skew. However, this term is hard to be evaluated since it deviated from logic style, clock routing network, register distribution and technology. More precisely, clock skew is caused by several phenomena: asymmetric routes to the clocked elements, differing interconnect line parameters, different delays through the clock distribution elements, and different device threshold voltages for the clock distribution logic.

Therefore, one should keep any possibilities in mind so as to handle the worst skew problem. By the way, once we minimize the skew effect in every level of the design, we will have higher probability for improving the performance.

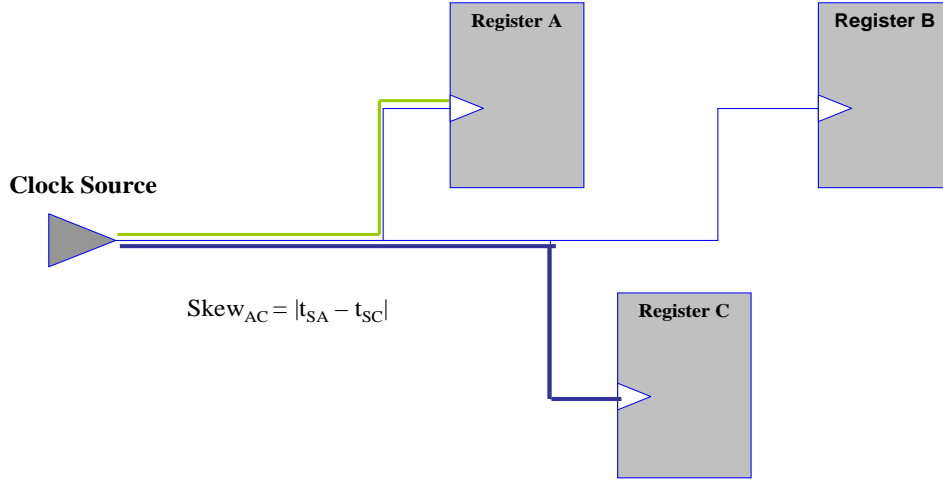


Fig. 2.6: A simple diagram which demonstrates clock skew effect.

In the following subsections, we will introduce some classic algorithm which minimize the skew effect even minimize the clock power simultaneously.

2.3.3 Problem Formulation

The clock routing problem defines as follows:

Given the routing plane and a set of sinks $C = \{c_1, c_2, \dots, c_n\}$ lying within the plane and c_s represent the clock source pin. The target is to construct a network to connect all of the clock pins c_i with clock source c_s hence to minimize the source to sink delay, clock skew and total clock wire-length.

The cost function can be formulated as follows:

$$Clock\ Routing\ Cost = Minimize \left(\max_{\tau_{(c_s, c_j)}} , \max_{(m, n) \in C} | \tau_{(c_s, c_m)} - \tau_{(c_s, c_n)} |, wirelength_{clock} \right)$$

2.4 Previous Work Related to General and Low Power Clock Routing

Prior to delving into the clock routing algorithm, it is necessary to define its role in the context of the overall design flow. In the contemporary physical design flow, it consists three classical steps: floorplanning/placement, global routing and detailed routing. Clock routing is a specific problem interposed between placement and global routing. Specifically, during the clock routing step, the global and detail routes of the clock net are determined and passed to the global router as blockages. The determined routes are constructed so that the clock signal behavior is optimized.

2.4.1 Zero-skew Clock Routing

The skew can be minimized by distributing the clock sinks in a way such that for each path from the clock source to sink has same delay time under some wire delay model. The higher precision of a delay model is, the fewer impact of the clock skew has. Many approaches are devoted to minimize the skew problem in this field such as [15, 14, 17, 16, 11, 13, 12, 18]. They design their routing algorithms to solve the zero-skew clock routing problem distinctly, and their various skew minimization techniques also motivate many works on low power zero-skew clock routing. Additionally, as we will introduce in the coming chapters, the *LPGC Placer* built an “pseudo clock router” for evaluation on gated clock topology. This pseudo clock router is also motivated by some of the zero-skew clock router. We introduce the classic zero-skew clock algorithms in the coming sections.

The Method of Means and Medians

The first clock routing algorithm is the *H-tree clock router* which is a well-known algorithm. However, it could not solve the problem with unevenly distribution of clock sinks. This motivated authors in [12] to invent a top-down clock routing algorithm called method of means and medians(MMM).

The MMM algorithm follows a technique very similar to the H-tree algorithm by

recursively partitioning a circuit into two equal parts, and then connects the center of the mass of the whole circuit to the centers of the mass of the two sub-circuits. The algorithm is simple and yields good results.

The definition of some terminologies is that: Let s_i be the clock sink i , and $P_s = \{s_1, s_2, \dots, s_n\}$ is the set of the clock sinks. L_x is the list of clock sinks which sorted according to their x-coordinate. P_{x_m} is the median of the L_x . Given P_{x_m} , assign points in list L_x to the left of P_{x_m} to the set P_{Left} . Assign the remaining points to the set P_{Right} . Similarly, P_{Bottom} and P_{Top} represent the division of P into two sets about P_{y_m} , the median of y axis.

Because of the geometric nature of the problem, one may consider the partition of the set of clock sinks as the partition of a region. Thus P_{Bottom} and P_{Top} , P_{Left} and P_{Right} partition the original region by y-median or x-median into two sub-regions with an approximately equal number of sinks in each sub-regions.

The algorithm operates as follows: First, splits P_s into two sets (in the x or y directions). Suppose that a split P_s into P_{Bottom} and P_{Top} is chose. Then, the algorithm connects the center of the mass of P_s to each of the center of mass of P_{Top} and P_{Bottom} respectively. Then split the regions P_{Top} and P_{Bottom} in the x direction respectively. Thus splits between x and y directions are introduced on the set of points recursively until there is only one point in each sub-region. A simple example of this algorithm is shown in Fig. 2.7.

The Geometric Matching based Algorithm

Another binary tree based routing algorithm [14, 15] is presented by Kahng, Cong and Robins. In their approach, clock routing is completed by constructing a binary tree through recursively *geometric matching*. This algorithm is so called *Geometric Matching Algorithm (GMA)*. It works in a bottom-up fashion while MMM algorithm is a top down algorithm.

Given a set P_s of n sinks, a geometric matching on P_s is a set of $n/2$ line segments

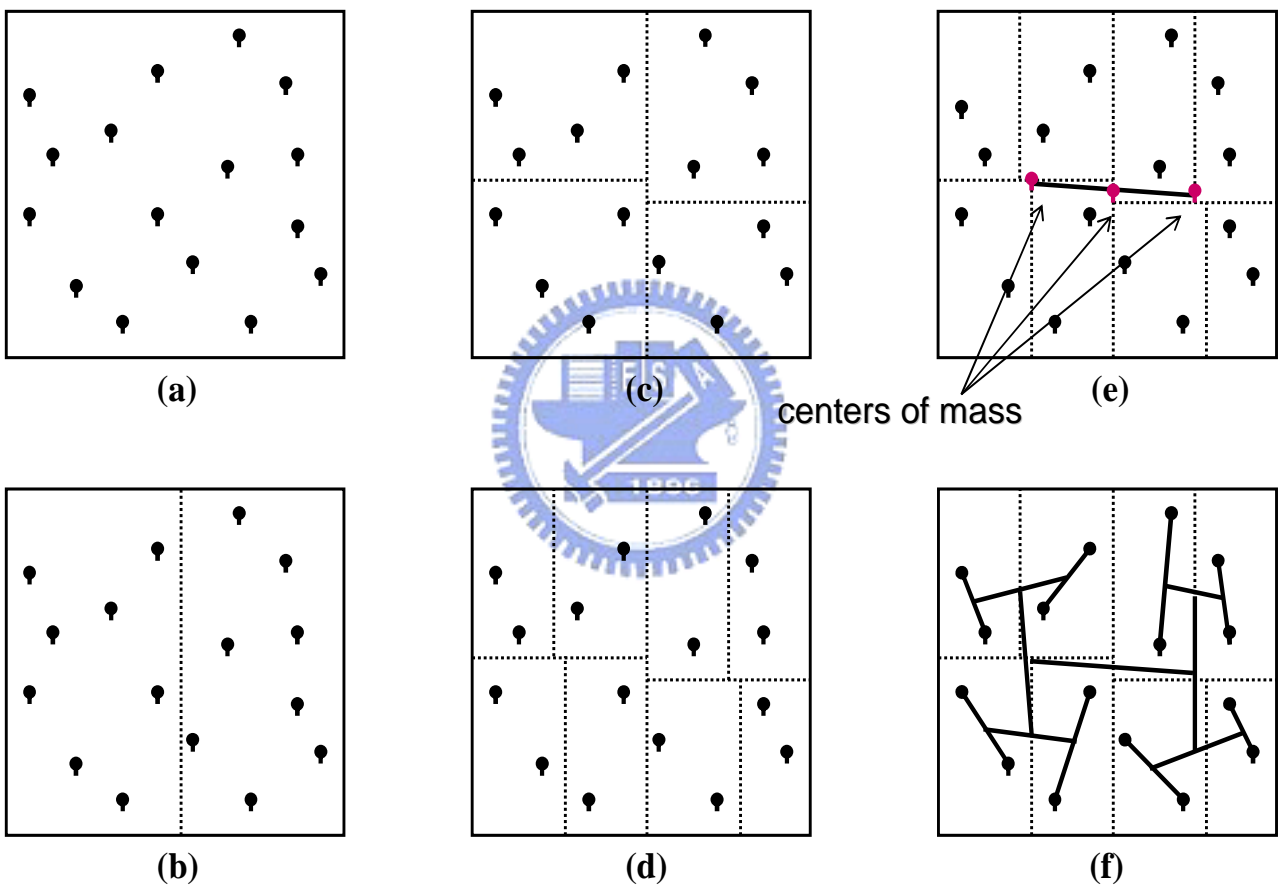


Fig. 2.7: Method of mean and median clock routing algorithm.

whose endpoints are in P_s , with no two line segments sharing the same endpoint. Every line segment in the matching is defined as an *edge*. The geometric matching cost is the sum of the lengths of its edges.

The target is to construct a clock tree by recursive matching. It starts by considering a forest of n isolated sinks, The minimum-cost matching on these sinks yields $n/2$ line segments, each of which defines a subtree with two leaf nodes. The center point of each line segment is called the tapping point and if the clock signal is transmitted through the tapping point, then the clock signal will arrive at the two endpoints of the line segment with zero skew. The set of tapping points serves as the set of points for the next iteration of the matching algorithm. This algorithm recursively reduce the number of points by $\frac{1}{2}$ once a iteration. By the way, the algorithm optimizes the wire-length in each iteration of matching. In each iteration, the algorithm selects the root of the newly merged tree to be the tapping point which minimizes the path length skew to the leaves of the two subtrees. Because this algorithm is based on geometric matching, its time complexity is $O(n^3)$. Fig 2.8 shows GMA algorithm running on eight-point set.

Fig. 2.8(a) and Fig. 2.8(b) represent the first iteration of matching, the points is reduced to be 4. Fig. 2.8(c) demonstrates the next matching iteration. However, the figure shows that the connection path between the tapping points of T_{12} and T_{34} is impossible. Therefore, this algorithm has a refinement step called “H-flipping”(shown in Fig. 2.8(d)) in order to fix the problem of unbalance path length when merging two points.

The Deferred Merge Embedding Algorithm

The purpose of deferred merge embedding algorithm [16, 17, 13] was to improve the two techniques which we have discussed. This DME algorithm could minimize the total clock wirelength with zero-skew under any given clock topology. In other word, this algorithm should be performed after the clock topology is determined.

The MMM and GMA are the algorithms could solve clock topology construction and clock routing at the same time while DME just solve the second one. However, this prop-

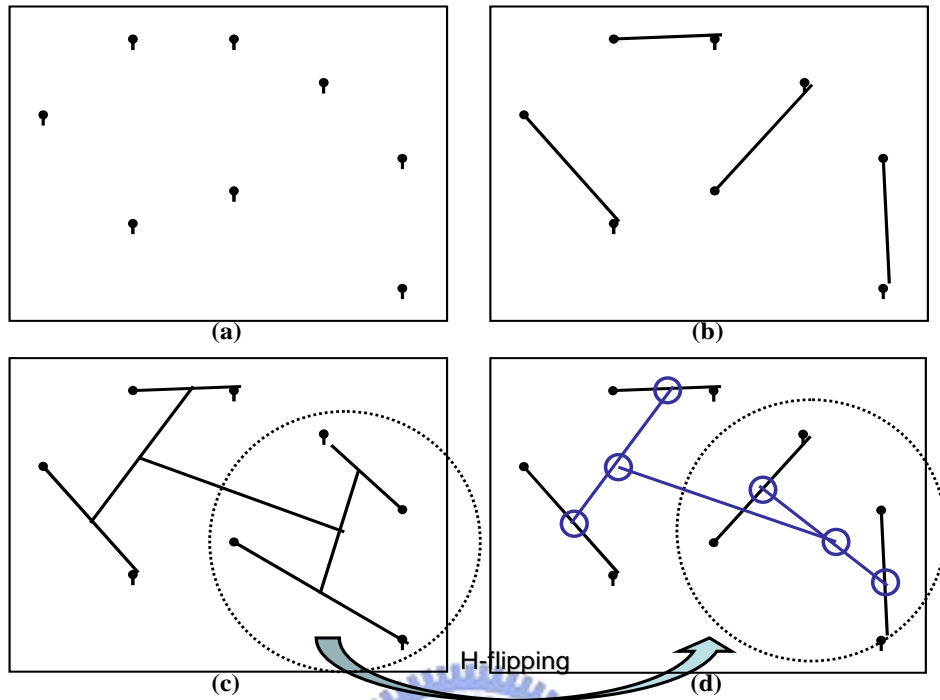


Fig. 2.8: Geometric matching based clock routing algorithm.

erty make DME a multi-purpose clock router since different clock topology could brings different advantages and disadvantages. We will introduce the optimal clock topology which optimized the total clock power in Section 3.2.

A generic DME algorithm is a two phase: *bottom up merge* and *top down placement*. The name placement here just referred to determine the precise locations for every steiner point (merge point) in order to eliminate the skew. First, we define *merging segment* and *Tilted Rectangle Region (TRR)* for the clarity in this section. A *manhattan arc* is defined to be a line segment, possibly of zero length, with slope -1 or $+1$ (45° tilted). The collection of the different manhattan arcs with same slope and fixed distance is called *tilted rectangular region*. We can examine Fig. 2.9 for an example. The central manhattan arc in a TRR is called its *core* and the *radius* is the distance between the core to boundary. The definition of *merging segment* is simpler, it represents the possible internal nodes' locations with the minimum wire cost when we merge two nodes.

The bottom-up phase is recursively merging each node's TRR in order to find out all

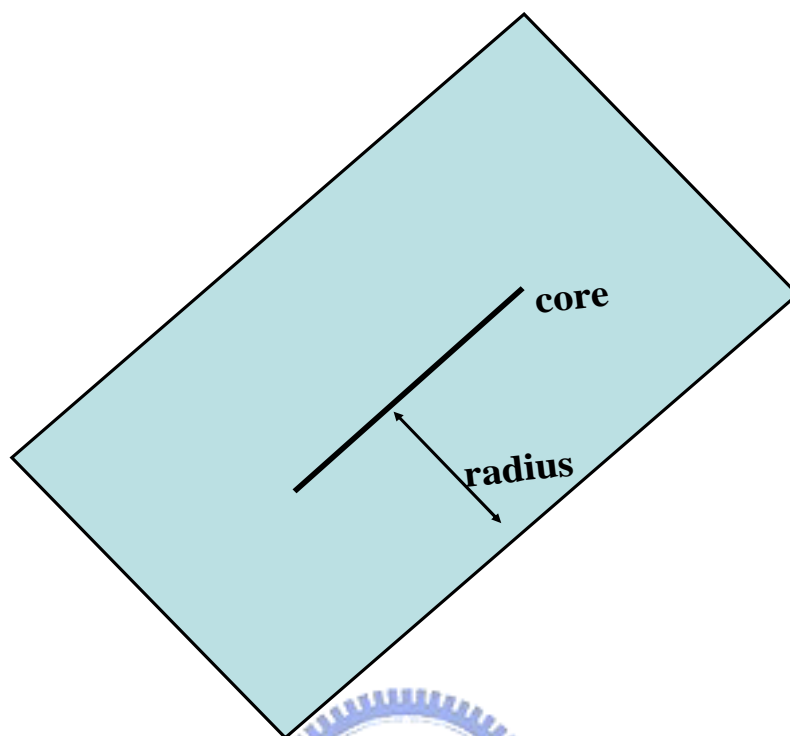


Fig. 2.9: A simple example of tiled rectangle region(TRR).

the merging segments which we call it *tree of merging segments*, Fig. 2.10 is an example. Given the tree of merging segments corresponding to a clock topology, the top-down phase resolves the exact positions of the internal nodes with minimum wire cost while preserving zero skew. DME algorithm requires the pre-determined clock topology so that this algorithm could be applied to many clock routing problem under different objective function as long as providing the related clock topology.

2.4.2 Low Power Zero-skew Clock Routing

For a long time, the general theory behind clock design was that the signal should be kept as clean as possible, and that a circuit designer should not interrupt or disable a clock signal. Traditionally, all the clock routing algorithms aim to solve the clock skew and minimum clock wirelength. However, as the technology keep improving and keep shrinking the feature size, the interconnect power has the same magnitude as compared to gate power [2, 3, 4, 19]. Clock network is a network with long wires and highly switching

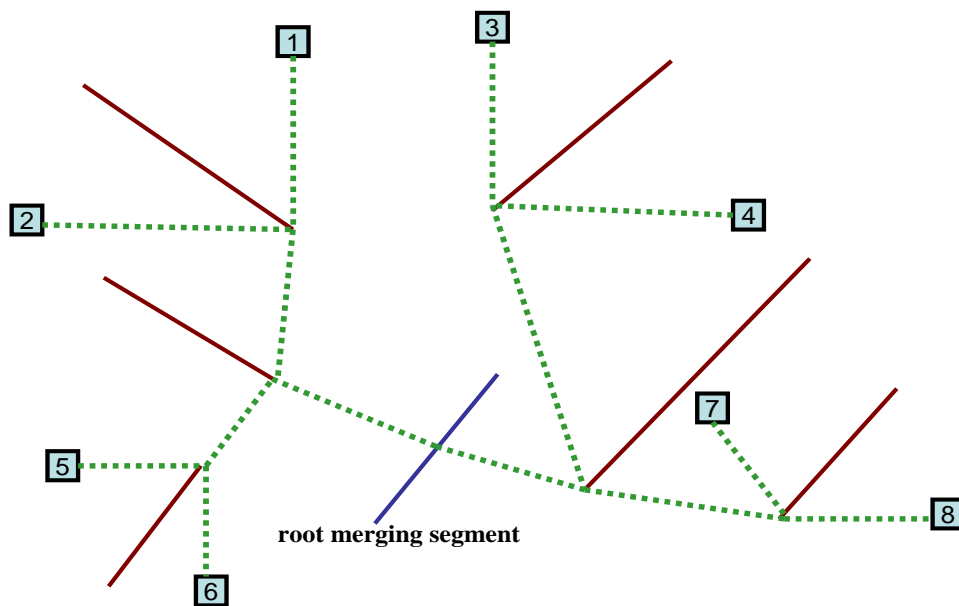


Fig. 2.10: A tree of merging segments. Solid lines are merging segments and dotted lines indicate edges between merging segments. The squares represent the clock sinks.

activity, hence, the higher power. It has been researched that the clock power is at least 40% of the total power for a typical microprocessor [19].

Therefore, today's research focus on distributing a clock network with minimizing clock skew, wirelength, timing and finally clock power. It is a new field with many chances and improvements in it.

Activity Driven Clock Design

Clock power can be saved by disabling clock signals from inactive registers(flip-flops) in idle circuit portions. We can carefully insert the controlling logic gates to control the clock signals in front of registers. By shutting down idle registers, the downstream combinational circuitry and data transmission path would not be active so that saving a substantial amount of power. This technique is so-called *gated clock routing* or *clock gating*.

By performing sophisticated algorithms, one could achieved the low power clock net-

work with inserting the control gates. Since registers that should be gated or merged together may be placed far apart, it is possible that gating control wirelength will finally be increased and the total so that the clock power finally goes up.

The authors in [22, 25] suggested that by putting the pseudo net between commonly-gated registers together in the netlist and then the placer would tend to put them closely. However, here are the drawbacks in this idea:

1. The amount of pseudo nets would be enormously so that the weight of net which made by manually dictated would probably be failed when handling larger circuits.
2. There is no way to handle the global optimization.
3. The pseudo net would destroy the timing of critical nets.

Gated clock routing for microprocessor design.

In this publication, the authors have proposed an approach for clock tree routing, that ensures zero-skew and reduces the clock power dissipation by minimizing the effective switched capacitance at the internal nodes of the tree. This algorithm refined the DME algorithm which we have known as a multi-purpose clock router. The authors [23, 20] replace the traditional clock routing objective from minimizing total wirelength to minimizing total switched capacitance.

The algorithm first formulates the clock network power and clock control network power as follows: Consider a clock tree without gates, the power dissipation on a particular clock edge e_i is defined as

$$Power_{e_i} = |e_i|c_0fV_{dd}^2 \quad (2.6)$$

where $|e_i|c_0$ represent the switched capacitance of this clock edge (c_0 is the unit wire capacitance and e_i is the wirelength of the clock edge e_i). The power dissipation for gated clock network, the Equation (2.6) should be modified as:

$$Power_{e_i,gated} = |e_i|c_0fV_{dd}^2P(e_i) \quad (2.7)$$

with additional consideration on the switching activity of the clock edge e_i . Since the operation frequency and supply voltage are fixed we define *effective switched capacitance* at any edge as:

$$SC_{e_i} = (|e_i|c_0 + C_i)P(e_i) \quad (2.8)$$

where the C_i represents the load capacitance of the clock edge e_i . Thus, the switched capacitance of the entire gated clock network can be defined as:

$$SC_{GCN} = \sum_{\forall e_i} SC_{e_i} \quad (2.9)$$

Finally, we take the gated clock control network into account. We can follow the definition as same as Equation (2.9):

$$SC_{GCCN} = \frac{1}{2} \sum_{\forall g_{e_i}} (c_0|g_{e_i}| + C_g)P_{tr}(g_{e_i}) \quad (2.10)$$

where the $|g_{e_i}|$ and $P_{tr}(g_{e_i})$ represent the wirelength and transition probability of the gated clock control wire g_{e_i} . We assume that the source of the gated clock control network is be laid in the center of the chip, C_g means the control gate's capacitance. The objective of this routing algorithm is to minimize the term $SC = SC_{GCN} + SC_{GCCN}$ through DME algorithm.

First, given all the clock sinks' locations, they could built a binary tree which represent the gated clock topology which could minimize SC . Then, by passing this topology to DME router hence force it to route this clock network with zero-skew. As we can realize from the idea of this algorithm, there are many aspects which we could improve. Because they have no process for global optimization, they may achieve either a non-optimal solution or a poor solution. Our work improves these drawbacks then provides a process or a good seed for global optimization.

Chapter 3

Activity analysis and optimal gated clock topology

In this chapter, the activity analysis and optimal gated clock topology will be introduced. As we mentioned in the previously chapters, the dynamic power of a CMOS circuitry is proportional to its switching activity. The VLSI designers can utilize different techniques on their designs to reduce the power consumption as long as the correct distribution of activity among every portion of their circuits is available. However, obtaining the precise activity distribution of each module is not an easy task. After the correct activity distribution is obtained, a procedure of constructing the optimal low power gated clock topology can be executed. *LPGC Placer* develops a procedure to build the optimal gated clock topology. Before introducing the em LPGC placer, the essential ideas of activity analysis and the optimal gated clock topology are presented in this chapter.

3.1 Activity Analysis

Since the problem of placement which we deal with is at the physical design level, the information of activity from the higher level of the design must be acquired. Many different methods have been investigated in this field [20, 22, 50, 51, 52, 53, 54, 55, 56, 57], two of them which are related to our work will be introduced.

3.1.1 Extraction from High Level Description of the System

In [22], the authors started from a high-level description of the system, and transformed a hardware description language (HDL) to the representation of control-data flow. Then, all of the modules were scheduled and allocated into the control steps. In other word, they analyzed the data flow in every clock cycle, then calculated the distribution of activity.

This method might be failed if the circuit if complicated. The reason is that one may not have the time to simulate the whole system in this manner because it's too slow. One simple example from [22] is shown in Fig. 3.1. This simple case can be perfectly transformed since its function is very simple. However, as we face the more complicated designs such as SOC or CPU design, this method is inefficient.

As we may see from the example that the authors provided which we put in Fig. 3.1, this case can be perfectly transformed since the function is simple. However, when we face the more complicated designs such as SOC or CPU design, this method is inefficient.

3.1.2 Instruction Level Simulation

The other group provided another solution of activity extraction [20, 23]. Their work was based on detailed statistics about the instruction frequency because different instructions had different operation probabilities. Hence, the activities of the modules related to the instructions can be obtained. They simulated in the instruction level with numbers of CPU benchmarks then analyzed how often each instruction was executed. They developed an algorithm to calculated each module's activity effectively.

We also demonstrate this method through an example. For simplicity, we assume that the processor which we shall analyze has only three instruction and five modules. The *RTL description* of each instruction lists the modules used when executing as what Table 3.1 shows. We can simulate the processor with benchmarks so as to obtain the instruction stream that is occurred most frequently. For example, we get the instruction stream through simulations for 15 clock cycles as Table 3.2 shows. Then, we can scan the table in order to get the executing probability for each module. For instance, $P(M_2) =$

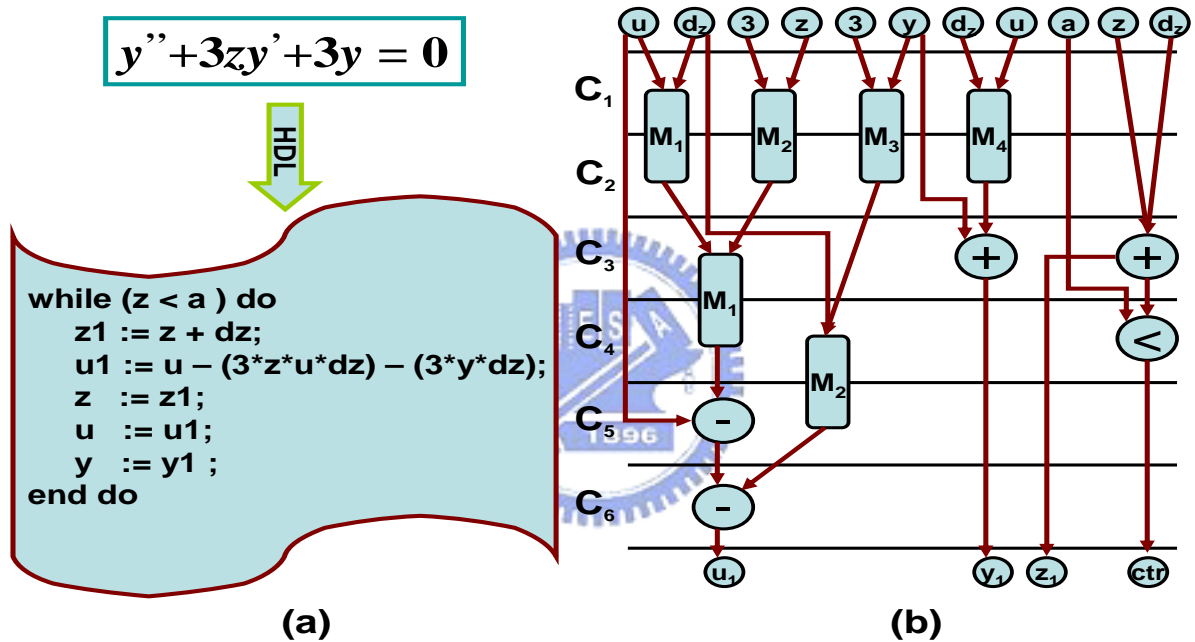


Fig. 3.1: Methodology for transformation of an ASIC function from hardware description language to a control-data flow graph. (a) Transformation from a simple ASIC function to hardware description language. (b) A simple example of CDFG after performing HDL scheduling and allocation.

Instruction	M_1	M_2	M_3	M_4	M_5
I_A	✓		✓		✓
I_B	✓			✓	
I_C		✓	✓	✓	

Table 3.1: A simple RTL description of instructions for a microprocessor.

Clock Period	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
Instruction	I_A	I_B	I_C	I_A	I_A	I_B	I_C	I_A	I_C	I_A	I_B	I_A	I_C	I_A	I_C

Table 3.2: Instruction level simulation for 15 Clock Cycles.

$$P(I_C) = \frac{C_3+C_7+C_9+C_{13}+C_{15}}{\sum_{i=1}^{15} C_i} = 0.33.$$

According to these tables, if we build a clock tree which merges module 1 and 5, we can calculate the probability of their control gate. We should take consideration on all instructions which contain module 1 and 5. Therefore, $P(gate_{M_1, M_5}) = P(M_1 \cup M_5) = P(I_A) + P(I_B) = 0.67$.

For the simplicity, we adopt this method to obtain the activity distribution. Researching on how to do it with more efficiently is one of our future work.

3.2 Optimal Gated Clock Topology

Fig. 3.2 demonstrates the circuitry with gated clock network. The *power management unit* is essentially a finite state machine which determines the active/inactive state of the gates on the clock network. The power management unit send the command to turn on or turn off a gate through *gated clock control network*. In Fig. 3.2, the rectangle “CLK” denotes the clock source. For the simplicity, we use “GCN” and “GCCN” to denote the gated clock network and gated clock control network.

The total power consumption of this system under such topology is

$$P_{sys} = P_{GCN} + P_{GCCN} + P_{COM} \quad (3.1)$$

where P_{GCN} and P_{GCCN} refer to the power consumption of the *gated clock network* and *gated clock control network* respectively, P_{COM} is the power consumption of the combinational circuit that can be calculated by using Equation (2.1).

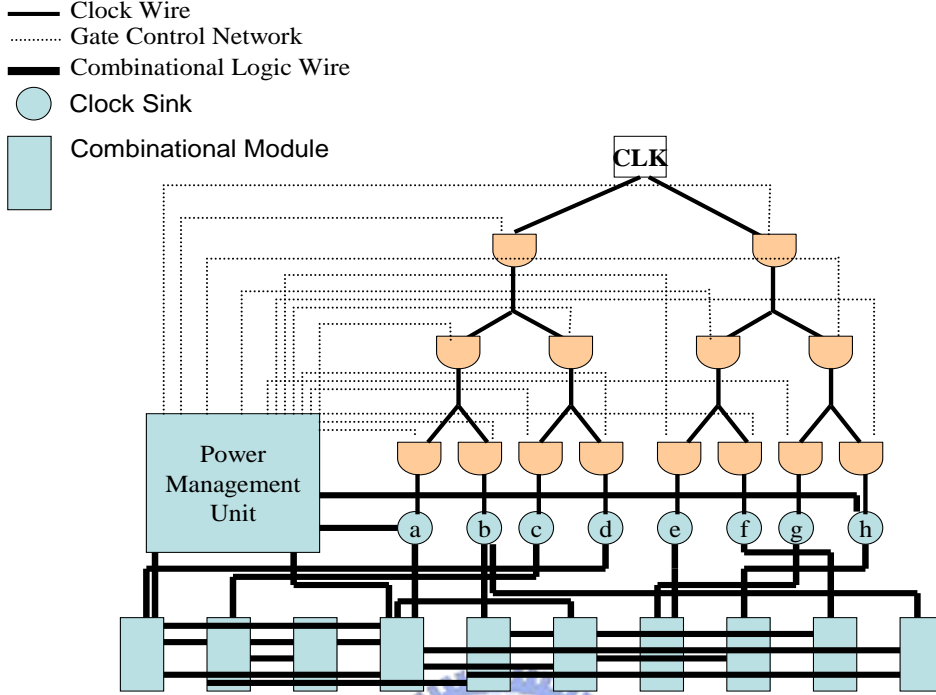


Fig. 3.2: A typical topology of gated clocking circuitry.

$$P_{GCN} = fV_{dd}^2 \left(\sum_{\forall \text{register } k} \alpha_k C_k + \sum_{\forall e_i} \alpha_{\text{topology}, e_i} c_0 |e_i| \right) \quad (3.2)$$

$$P_{GCCN} = fV_{dd}^2 \sum_{\forall w_j} (\alpha_{w_j} c_{w_j} + C_{\text{Gate}, j}) \quad (3.3)$$

In Equation (3.3), w_j represents a gated clock control wire with capacitance c_{w_j} , and α_{w_j} denotes the switching activity of w_j . Each w_j is connected to a control gate with input capacitance $C_{\text{Gate}, j}$.

In Equation (3.2), the power consumption of gated clock network can be divided into two parts, say registers' power and clock wire segments' power. As we can see in Equation (3.2), C_k denotes the input capacitance of a register, c_0 is the unit clock wire capacitance, and $\alpha_{\text{topology}, e_i}$ is the switching activity at each edge of the gated clock topology. Each value of $\alpha_{\text{topology}, e_i}$ is highly dependent on the gated clock topology.

Let's consider Fig. 3.3 for an example to explain the relationship between switching activity and gated clock topology. There are two different gated clock topologies in

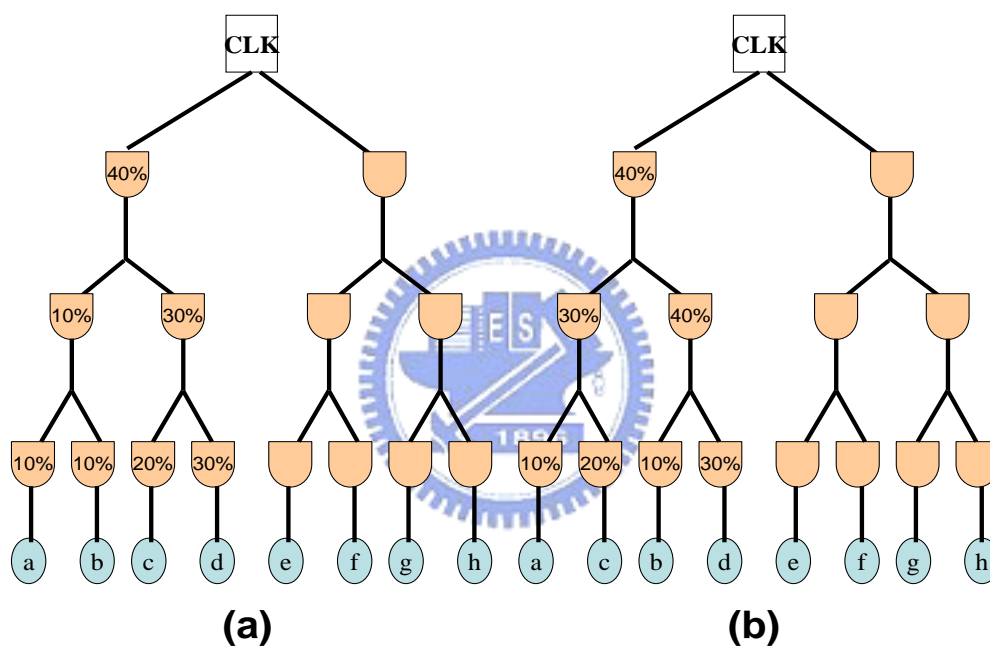


Fig. 3.3: An example for switching activities comparison under two different clock topology. (a) Topology with smaller average switching activity. (b) Topology with larger average switching activity.

Fig. 3.3.

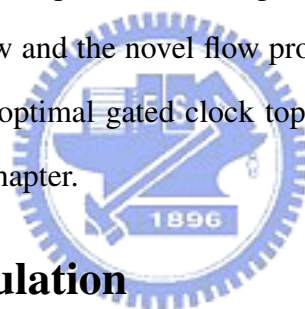
For simplicity, we assume that the right hand side of each topology is the same. The activity of each clock module activity is: $a=10\%$, $b=10\%$, $c=20\%$, and $d=30\%$. We also assume that the union activities for (a,b) pair and (c,d) pair are 10% and 30% respectively. By the way, the union activities for (a,c) pair and (b,d) pair are 30% and 40% . Therefore, we can easily compare the average activity for this two topology. The average switching activity for the gates in the left hand side of Fig. 3.3(a) is $\frac{10\%+10\%+20\%+30\%+10\%+30\%+40\%}{7} = 21.43\%$ while $\frac{10\%+20\%+10\%+30\%+30\%+40\%+40\%}{7} = 25.71\%$ for Fig. 3.3(b). If each module in each topology in the figure has been placed in the same location, the topology in Fig. 3.3(a) could save additional power from wasted. Therefore, we define the topology which has the smallest activity at each tree node as the “optimal gated clock topology”.

As we mentioned in the last section, we could develop an optimal gated clock topology which minimizing the P_{sys} . This would be the first goal of our placer. The second goal of our placer is to place each module at suitable position in order to make the optimal gated clock topology become routable, and then minimize P_{sys} , while the gated clock routing algorithms would probably view that topology as an unroutable structure since the placements provided by the typical placement algorithms do not consider the factor of gated clock network. The major techniques of our placer will be introduced in the next chapter.

Chapter 4

LPGC Placer

In this chapter, the details of *LPGC Placer* will be described. First, we formulate the problem for *LPGC Placer*. Second, the data structure adopted by *LPGC Placer* will be introduced. Then, this chapter will present the comparison between the traditional gated clock network construction flow and the novel flow proposed by *LPGC Placer*. We will also introduce the notions of “optimal gated clock topology construction” and “pseudo clock router heuristic” in this chapter.



4.1 Problem Formulation

The low power driven placement problem based on optimal gated clock topology can be formulated as follows.

Given a set of modules $M = \{m_1, m_2, \dots, m_n\}$, the interconnect netlist I , and a clock activity relationship matrix, $L^{r \times r}$, where r is the number of clock pins, and each entry l_{ij} in $L^{r \times r}$ represents the joint activity of clock pins i and j , the goal of the low power driven placement based on optimal gated clock topology is to force the placement to approach the optimal gated clock topology through novel pseudo clock router heuristics. At the same time, the traditional placement constraints such as area and the total wire-length are also taken into account.

4.2 Data Structure

Many different representations and data structures, such as B*-tree [32, 48], O-tree [33], corner block lists [34], sequence-pair [35], TCG [37], and the other outstanding works [42, 41, 40, 39, 38, 36], have been proposed to handle the floorplanning/placement problem. In order to make a comparison between a dedicated placement and typical placement for the gated clock power optimization, we adopt the data structure of B*-tree [32, 48].

B*-tree is an efficient, flexible, and effective representation of geometric relationship for floorplanning/placement designs. It is essentially a binary tree as shown in Fig. 4.1(a), but with several efficient operations, rotate, flip, move, and swap, for the placement problem. After a series of the above operations onto the B*-tree data structure, a “packing” function can be utilized to map the B*-tree representation to a real placement as shown in Fig. 4.1(b). The above mechanisms of B*-tree could provide us a simple way to “disturb” the placement. By using the optimization algorithms such as *simulated annealing*, *force directed* and *simulated evolution* we can take the advantage of this merit since these algorithms need disturb the target thoroughly.

By the way, B*-tree is a good data structure for the area constraint since it places modules compactly from the bottom left corner, and this can minimize the dead space. We adopt this data structure for the *LPGC Placer*, and compare the experimental results of *LPGC Placer* with an outstanding B*-tree placer [48].

4.3 LPGC Novel Gated Clock Network Construction Flow

The novel flow for gated clock network construction will be introduced in this section. Fig. 4.2 is the comparison between the traditional gated clock network construction flow and the LPGC novel flow. Traditionally, a placer doesn't consider the distribution of the clock sinks which has a dramatically impact on the clock power. As shown in Fig. 4.2, the traditional gated clock network construction flow is a simply general placement followed by the gated clock routing. Therefore, the gated clock routing algorithms usually obtain

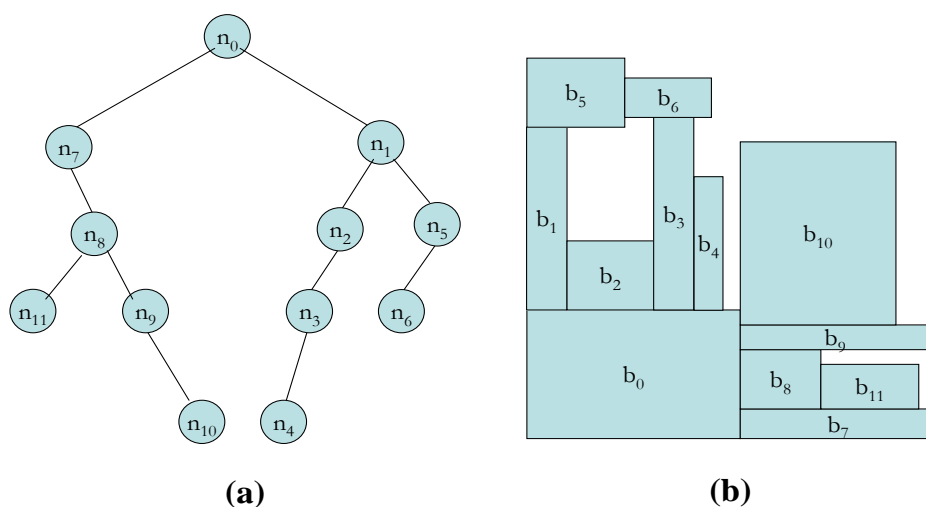


Fig. 4.1: B*-tree data structure. (a) An eleven blocks B*-tree representation. (b) Modules' actual positions and placement of (a).

a “uniform distribution” of clock sinks so that the chance to get the optimal solution is limited.

LPGC Placer provides a novel flow. First, activity analysis is performed to obtain the activity distribution among all the clock sinks. Second, according to the activity distribution, an optimal gated clock topology is constructed. Then a “dedicated seed” for the gated clock routing is constructed by using the placement core and pseudo clock router heuristic as shown in Fig. 4.2(b). After clock gating, an arbitrary gated clock router [21, 22, 23, 20] can be applied to the placement to construct the low power gated clock network.

The key role of the *LPGC Novel flow* is the “*pseudo clock router heuristic*” for evaluating whether a placement is prone to make the optimal topology routable or not. The pseudo clock router uses a quick way to estimate the clock wire-length instead of performing a real routing algorithm. With the criterion of minimizing the total estimated clock wire-length, *LPGC Placer* can place the modules close to their optimal locations

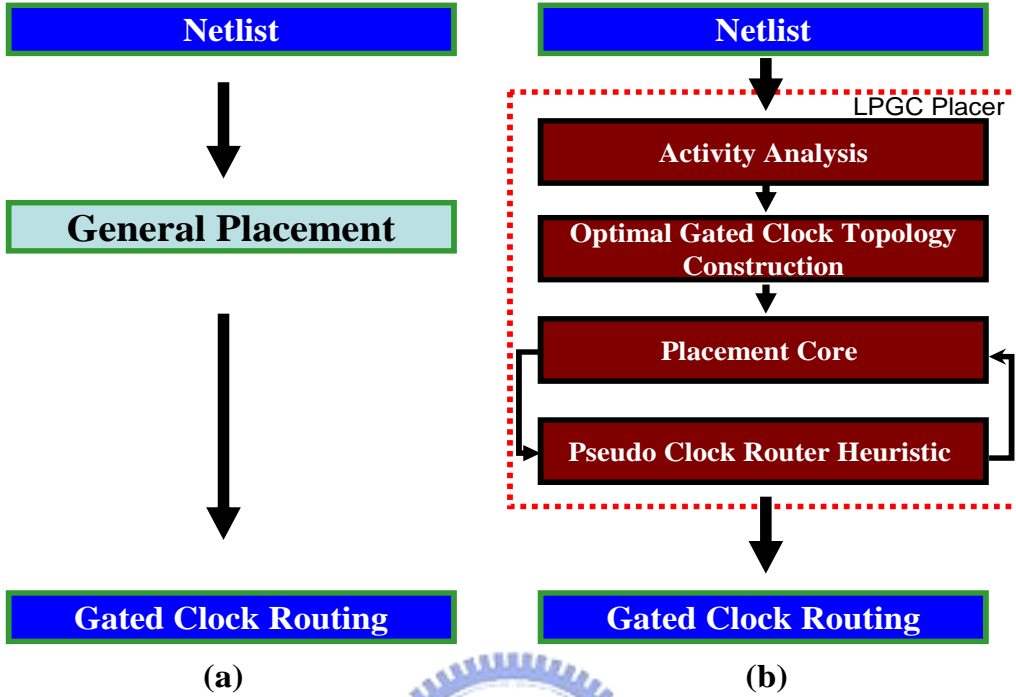


Fig. 4.2: (a) Traditional design flow of the gated clock network construction. (b) The LPGC novel flow for gated clock network construction. The dotted rectangle circles the operation flow of *LPGC Placer*.

for the routing result to approach the optimal gated clock topology.

The placement core of *LPGC Placer* can be any kind of placement. Here, the simulated annealing method is used to set up the placement core. All the details of *LPGC Placer* will be described in the next section.

4.4 The Novel Low Power Placement Algorithm

In this section, two main techniques will be described to set up the *LPGC Placer*, a low power driven placer based on the optimal gated clock topology. *LPGC Placer* is a simulated annealing based placer with consideration on minimizing area, total wire-length, gated clock network power, and gated clock control network power. The cost function of *LPGC Placer* is defined as

$$Cost_{LPGC} = \alpha \cdot Area + \beta \cdot TotalWireLength + \gamma \cdot GatedClockCost, \quad (4.1)$$

where α , β , and γ are the weighting factors corresponding to *Area*, *TotalWireLength*, and *GatedClockCost*, and $\alpha + \beta + \gamma = 1$. Here, the *GatedClockCost* is a function proportional to the power consumption of clock and gated clock control networks. First, a circuit's activity pattern will be acquired by simulation with the same technique shown in [20, 23] which has been introduced in Section 3.1.2. After that, the *LPGC Placer* will construct the optimal gated clock topology. Then, the *LPGC Placer* will also minimize the $Cost_{LPGC}$ through the optimizing algorithm, say *simulated annealing*. By the way, the *LPGC Placer* minimizes the *GatedClockCost* by utilizing the pseudo clock router heuristic based on the optimal gated clock topology.

4.4.1 The Optimal Gated Clock Topology Construction

The first step of *LPGC Placer* is to determine the topology structure of optimal gated clock network distribution. In other word, we seek for the routing tree which could optimize the total switching activity at every node in the gated clock network under the constraint of routability.

We modify the recursive construction method used in [20]–[23] with an additional control parameter, the physical connectivity of different blocks, to build the optimal gated clock routing tree. The physical connectivity is indeed an important parameter since it directly determines whether the activity-optimized clock topology can be achieved or not. The other difference is that our procedure determines the optimal gated clock topology since each block's location on the plane hasn't been determined. In other word, we try to find a topology which has the minimum average activity.

This construction procedure is described in Fig. 4.3. The Huffman coding algorithm [49] is recursively used to construct the clock tree since this algorithm is a effective procedure for constructing a binary tree with the lowest average activity. Each new node is created at each time, and its activity is the joint activity of its two children. With the bottom-up manner of the Huffman coding algorithm, there is a sequence of merging order. We utilize this sequence of merging order to speed up the calculation of *GatedClockCost*.

Optimal Gated Clock Topology Construction Procedure	
Input:	<i>Activity Patterns of the Modules</i>
Output:	<i>Optimal Gated Clock Topology T</i>
1	<i>set S ← total modules</i>
2	<i>T ← null</i>
3	<i>While(S.size ≠ 1)</i>
4	<i>find the two smallest joint activity modules in S → (x, y)</i>
5	<i>if tie, choose stronger connectivity with T</i>
6	<i>z ← newnode(x, y) //parent of x,y</i>
7	<i>S.insert(z)</i>
8	<i>T.insert(x)</i>
9	<i>S.delete(x)</i>
10	<i>T.insert(y)</i>
11	<i>S.delete(y)</i>
12	<i>T.insert(the one left in S) //Construct the root</i>
13	<i>return T</i>

Fig. 4.3: Optimal gated clock topology construction procedure.

The binary tree is very suitable for clock distribution [27]. As we introduced in Section 2.3, minimizing the skew effect is necessary in the clock routing. Therefore, using a binary tree for building the optimal gated clock topology is very reasonable.

The connectivity is a metric of connection between two modules and can be estimated from the netlist of non-placed modules. If block a and b are highly connected, they should be placed closer than other blocks. With this criterion, the total wire-length between each block is potentially reduced.

4.4.2 Approaching to Optimal Gated Clock Topology and Pseudo Clock Router Heuristic

The optimal gated clock topology T can be constructed after performing the above procedure. Equation (3.2) can be rewritten as

$$P_{GCN} = fV_{dd}^2 \left(\sum_{\forall \text{register } k} \alpha_k C_k + \sum_{\forall e_i} \alpha_{Optimal\ topology, e_i} c_0 |e_i| \right) \quad (4.2)$$

where $\alpha_{Optimal\ topology, e_i}$ is the switching activity at each node of the optimal gated clock topology.

Since the control gates are not inserted at the placement level, the second term of the Equation (3.1) can not be directly calculated. However, all clock gating algorithms tend to insert gates at the positions which can minimize both the control network's wire-length and the gated clock network power. In other word, we try to disturb the placement to make the optimal gated clock topology becoming a routable structure. We observe that if we could place those modules which are in the same families of the optimal gated clock topology closer, the gated clock router would tend to actually route the topology. Hence, the power consumption of the gated clock control network can be reduced. For example, given an optimal clock topology T as shown in Fig. 3.2, its typical placement without considering the clock gating is illustrated in Fig. 4.5(a). *LPGC Placer* can move the cells to approach a suitable solution of placement which makes the optimal gated clock topology being routable, and the wire-length of gated clock control network is minimized by the clock gating router. As a result, Fig. 4.5(b) does have smaller clock and control network wire-length. The reason is that the clock gating routers [20]–[23] always merge the modules with minimum switched capacitances to achieve the optimal gated clock topology, and *LPGC Placer* places the modules based on their optimal topology to provide a better placement for the gated clock router. In other word, since the routers cannot move the modules, they can only construct a local optimal clock gating network with an arbitrary placement while we can provide a global optimal placement for the gated clock routing methods.

Given an optimal gated clock topology T and a placement P , *LPGC Placer* utilizes the following mechanism to drive the given placement to a better solution. *LPGC Placer* first maps each leaf node of T to the corresponding module of P . Then, it estimates the total clock wire length as if the gated clock network has been really routed as topology T , and chooses this estimate as the cost function. To estimate the total wire length, *LPGC Placer* utilizes the idea behind the zero-skew clock routing algorithms proposed by [12, 14], and uses a special data structure to store T in the sequence of bottom-up merging in order to speed up the procedure of updating the total estimated clock wire length. We call

Gated Clock Placement Cost Computation Procedure

Input: *Optimal Gated Clock Topology T; Given Placement P***Output:** *Clock Cost of Some Placement*

```
1 left ← Head(T)
2 ClockCost ← 0
3 While(Next(left) ≠ null)
4   right ← Next(left)
5   UpdateLocation(left, right, P, T)
   /*compute their parent's approximately zero skew location by LPGC pseudo clock router*/
6   ClockCost ← ClockCost + WireLength(left, right)
7   left ← Next(right)
8 ClockCost ← ClockCost + WireLength(left, clock source)
9 return ClockCost
```

Fig. 4.4: Gated clock placement cost computation procedure for pseudo clock router heuristic.

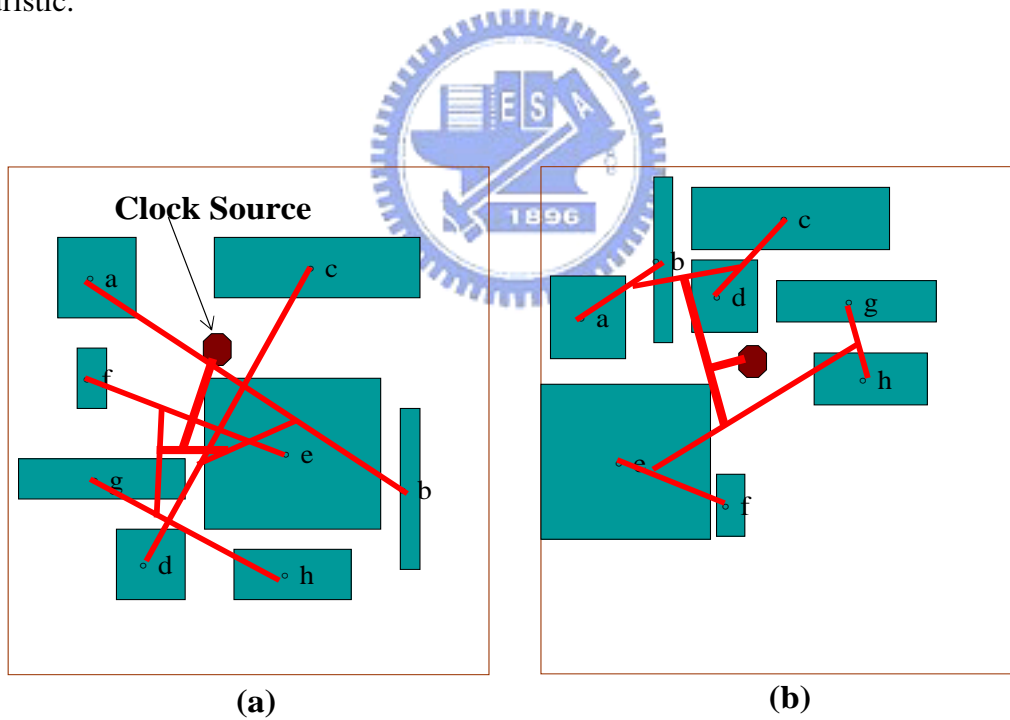


Fig. 4.5: (a) A typical placement result under general constraints, the optimal topology seems to be unroutable. (b) A dedicated placement which makes the optimal topology routable.

this idea as “*pseudo clock router heuristic*”. The idea behind the real zero-skew clock router [12, 14] is that minimizing the skew problem by routing through the central of the mass. The procedure of constructing the optimal gated clock topology combines a pair of modules once a time. Therefore, memorizing the sequence of the merging order of the Huffman coding algorithm can speed up the calculation process of the estimated central of the mass.

♠ *Pseudo Clock Router Heuristic* ♠

As we introduced in Section 2.4.1, the MMM algorithm is a top-down algorithm and the GMA algorithm is a bottom-up algorithm. They route the clock net to achieve zero-skew and minimize the clock wire-length.

Our optimal gated clock topology can not be a routable topology if the placement is not “suitable”. We evaluate whether a placement is suitable or not by using a pseudo clock router which performs pseudo clock routing algorithm according to the optimal gated clock topology.

This pseudo clock router executes the following steps.

Step 1 Dictate the siblings and father for each clock module top-down according to the optimal gated clock topology.

Step 2 Create steiner points for storing the merging points of each level, and those merging points are called “fathers”.

Step 3 Compute the location of each merging point bottom-up by using the idea of means of medians.

Step 4 Sum the total pseudo clock wire-length to estimate the total clock wire-length.

Fig. 4.6 shows two different placements according to the optimal gated clock topology of Fig. 3.2. You can see that the placement shown in Fig. 4.6(a) has longer pseudo clock wire-length than the placement’s in Fig. 4.6(b). Therefore, the placement of Fig. 4.6(b) is more suitable than Fig. 4.6(a) since it seems to be more possible for a real router to route

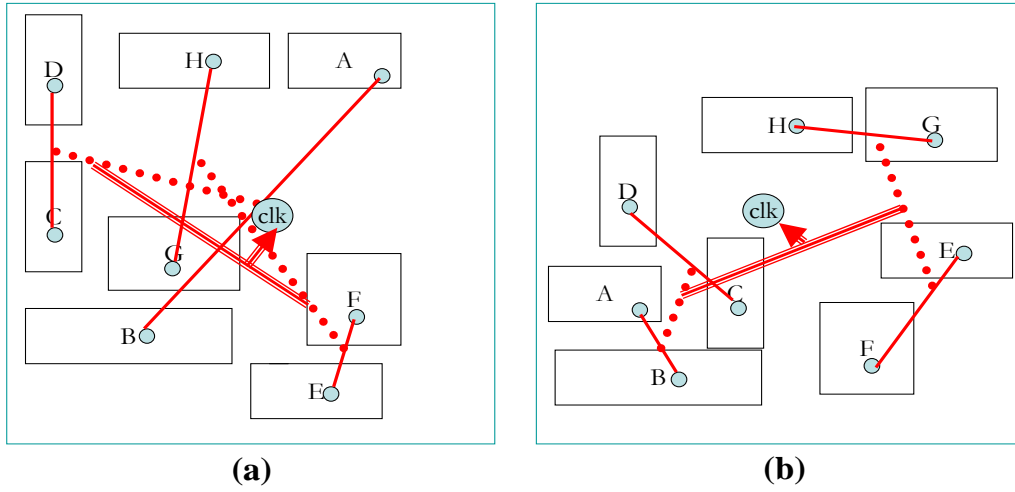


Fig. 4.6: Evaluate a placement through pseudo clock router heuristic. (a) Longer pseudo clock wire-length. (b) Shorter pseudo cLock wire-length.

it. Therefore, by using this pseudo clock router heuristic, we could evaluate whether a placement is more suitable to let optimal gated clock topology become routable or not. One goal of Equation (4.1) is to minimize the *GatedClockCost* which can be approximated by the pseudo clock wire-length. Therefore, the cost function of gated clock routing can be formulated as

$$GatedClockCost = PseudoClockWirelength \quad (4.3)$$

The procedure of calculating the *GatedClockCost* by using the pseudo clock router heuristic is shown in Fig. 4.4. *LPGC Placer* utilizes this procedure to update the value of *GatedClockCost* for each different placement, and tries to minimize it to make the optimal gated clock topology routable. Actually, this procedure is a revised version cost function. We revisited the essential core of the pseudo clock router and then redesign our algorithm since we should calculate it as fast as we can. We just “memorize” the sequence of the merging order which was occurred in the construction of optimal gated clock topology. By

design a new data structure for storing the pseudo clock router, our procedure could be performed faster.

The computational complexity of the pseudo clock router estimation heuristic is $O(n)$. The simple proof of the complexity is described as follows: Given an ordered sequence $P = p_1, p_2, \dots, p_{2n-1}$ which is referred to the positions of the clock sinks or internal nodes. The number n denotes the total number of the clock sinks. We store the sequence P as the merging order in the procedure of the optimal gated clock topology construction. The pseudo clock wirelength can be calculated as:

$$\left(\sum_{i=1}^{n-1} |P_{2i-1} - P_{2i}| \right) + |P_{2n-1} - P_s| \quad (4.4)$$

where P_s denotes the position of the clock source. In Equation (4.4), $|P_{2i-1} - P_{2i}|$ represent the pseudo clock wire-length calculation with complexity $O(1)$. Finally we compute the pseudo clock wire-length with n times so that the complexity of pseudo clock router heuristic is $O(n)$.

Minimizing Equation (4.3) can reduce the power consumption gated clock network under the optimal topology. Let's compare two different placements with the same optimal gated clock topology as shown in Fig. 4.7. Fig. 4.7(a) represents a dedicated placement with considering the gated clock topology. In Fig. 4.7(a), the wire length of each clock net can be reduced since the clock sinks are close to the clock source. Therefore, each $C_{clock_wire,i}$ is reduced, and so is P_{sys} . However, a typical placement without considering the gated clock topology might prohibit the gated clock routing algorithm constructing the optimal gated clock network, as illustrated in Fig. 4.7(b). The clock sinks (a and b , or c and d) which should be merged are too far to be merged. This is the reason why a dedicated placer is important for clock gating routers.

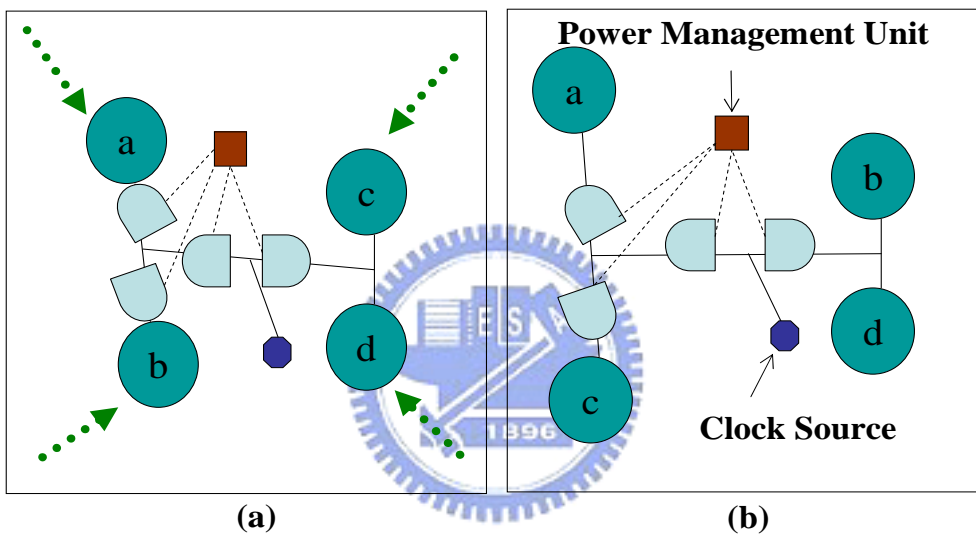


Fig. 4.7: (a) A dedicated placement with considering the optimal gated clock topology.
 (b) A typical placement without considering the optimal gated clock topology.

Chapter 5

Experimental Result

We have implemented our LPGC placer in C++ programming language on a 1GHz SUN Blade 1500 workstation with 2.5GB memory, and acquired the source code of B*-tree placer [48] and a gated clock router [20] from the authors. We apply both of our *LPGC Placer* and the outstanding B*-tree placer on the placement benchmark circuits. The benchmarks are designed to make them reasonable and suitable for this experiment.

The placement benchmark “plbench100” denotes a benchmark with 100 blocks. The design process of a benchmark with n blocks is explained as follows. First, we construct n blocks with same width and same height. Each of them is a 1000x1000 unit rectangle. Second, we randomly shrink the width and height of each block. The bounds of the shrinking ratio are $0.2 < \frac{height}{width} < 5$, and $0.6 < \frac{height \times width}{100 \times 100} < 1$. Finally, we randomly assign the interconnect relationship between each block.

In order to demonstrate the importance of a dedicated placement for the clock gating, we compare two different gated clock network construction flows as shown in Fig. 4.2. In this experiment, we adopt B*-tree placer [48] and the gated clock router [20] to represent the traditional flow. Then, we replace the B*-tree placer with *LPGC Placer* for the novel flow in Fig. 4.2(b). We first compare the general metrics of a placement, which are the total wire-length and total area. Then, we compare their gated clock networks’ wire-length, gated clock control networks’ wire-length and the total switched capacitance. We use total switched capacitance to denote the total clock power of them.

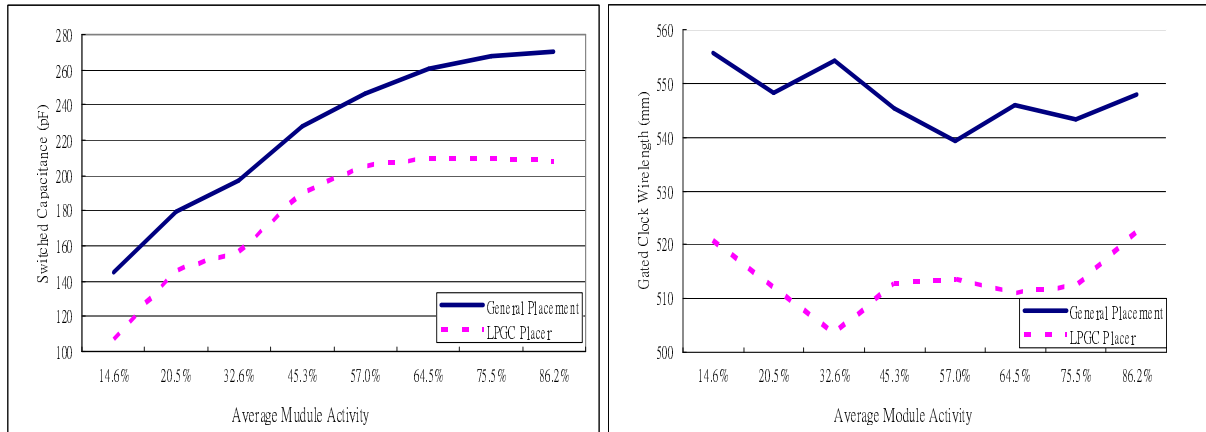
The results are reported in Fig. 5.1, and Table 5.1. Fig. 5.1 shows the trends under dif-

ferent average activity of modules for benchmark plbench576. The “General Placement” denotes the result of traditional gated clock network construction flow while the “*LPGC Placer*” denotes the result of novel flow. Fig. 5.1(a), (b) and (c) show that *LPGC Placer* can reduce about 20% in the total switched capacitance, 6% in the gated clock network’s wire-length and 28% in the gated clock control network’s wire-length among different average activity of modules in plbench576.

Fig. 5.1(b) shows that *LPGC Placer* does really save much of gated clock network power by shrinking the clock wire length. The authors of [20] pointed out that the power consumption of the gated clock control network could be ignored because it was very small. However, Fig. 5.1(c) shows that it is not always true, and *LPGC Placer* can save the gated clock control network power significantly. Hence, in Fig. 5.1(a), *LPGC Placer* can still reduce the power consumption as the average activity of modules is high. Therefore, these factors result in the power reduction gap whether adopting a dedicated placement or not.

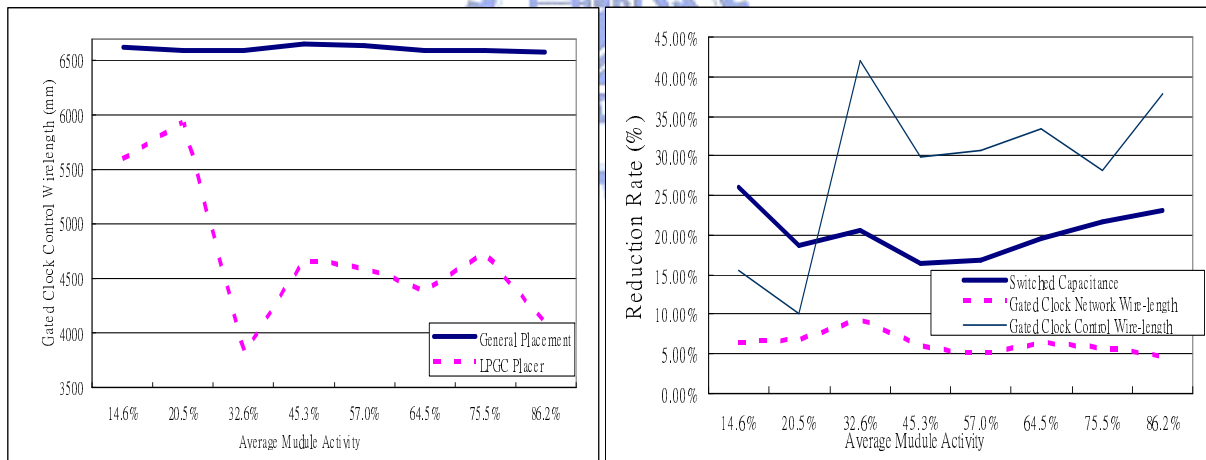
Table 5.1 shows the comparison between the traditional flow and the novel flow among different benchmarks. The “Est. Wire” is the estimation of total wire length between modules without including the gated clock and gated clock control networks. In Table 5.1, we use CSC, the total switched capacitance of gated clock and gated clock control networks, to represent the power consumption of the clock routing. The GCW is the actual wire length of gated clock network, and GCCW is the actual wire length of gated clock control network.

The results are extremely excellent since on the average the *LPGC Placer* can provide a dedicated placement for clock gating methods to additionally 19.2% and 33.2% of the gated clock network’s and gated clock control network’s wire-length. The power consumption of the total gated clock network is reduced by 27.4%. The total area is also reduced by 0.5%. However, the total wire length is increased by 0.4%. We should note that the clock power what the *LPGC Placer* saves is the additionally amount which any typical placer could not provide. Fig. 5.2 and Fig. 5.3 present the snapshots of the place-



(a)

(b)



(c)

(d)

Fig. 5.1: (a) Gated clock network comparison between a general placer and *LPGC Placer* with additionally consideration on gated clock routing. (b) Additional gated clock network Wire-length Saving. (c) Additional gated clock control network wire-length saving. (d) Reduction rate comparison.

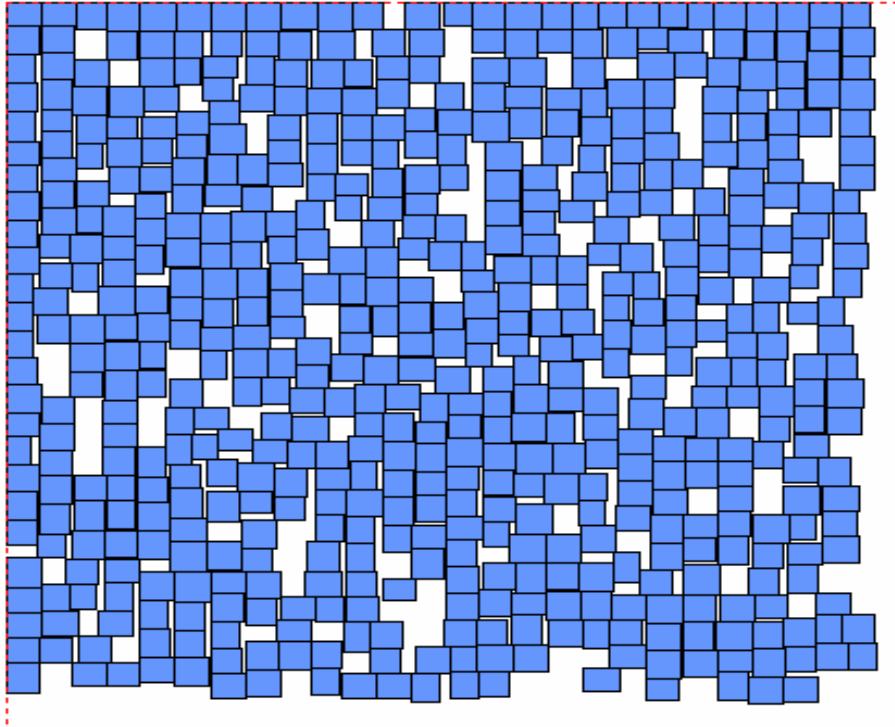


Fig. 5.2: Snapshot of the placement result on circuit plbench576.

Benchmark	B* -tree Placer [48]				
	Est. Wire (mm)	Area (mm ²)	SCC (pF)	GCW (mm)	GCCW (mm)
plbench250	66.5	16.1	33.7	60.2	500.2
plbench576	1940.5	568.5	242.2	547.5	6608.9
plbench1000	614.1	1012.0	569.2	983.5	15903.3
plbench3136	5135.3	3646.1	3007.1	4333.8	94829.4
plbench10000	20104.0	10264.1	16222.9	10451.0	593076.9
Benchmark	LPGC Placer				
	Est. Wire	Area	SCC	GCW	GCCW
plbench250	70.6(+6.1%)	16.1(+0.0%)	27.1(-20.0%)	48.1(-20.2%)	344.5(-31.1%)
plbench576	1905.9(-1.8%)	544.5 (-4.2%)	178.9 (-20.4%)	513.5 (-6.20%)	4727.6 (-28.5%)
plbench1000	616.0(+0.3%)	1019.1(+0.7%)	403.0(-29.2%)	721.5(-26.6%)	10128.8(-36.3%)
plbench3136	4972.0(-3.2%)	3698.1(+1.4%)	2151.1(-28.5%)	3354.2(-22.6%)	67297.0(-29.0%)
plbench10000	20204.0(+0.5%)	10212.1(-0.5%)	9819.0(-39.4%)	8312.4(-19.3%)	349180.4(-36.6%)
Avg. Reduction	+0.387%	-0.523%	-27.4%	-19.2%	-33.2%

Table 5.1: Comparison between typical placer with *LPGC Placer* on benchmark circuits

ment results of benchmark *plbench576* and *plbench3136* respectively by *LPGC Placer*.

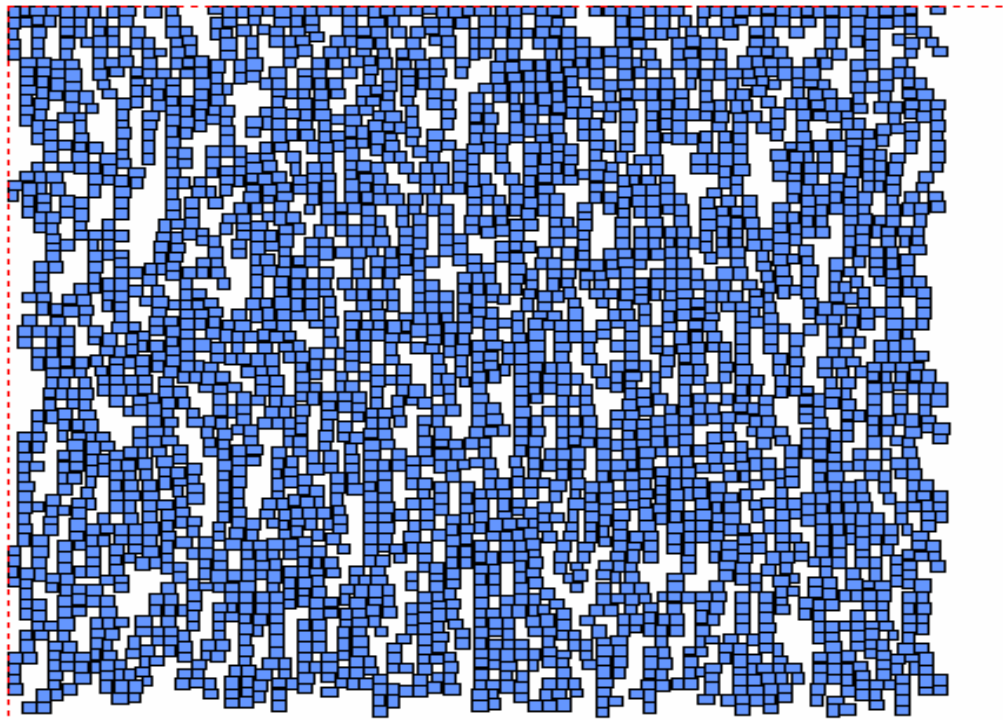


Fig. 5.3: Snapshot of the placement result on circuit plbench3136.

Chapter 6

Conclusions and Future Work

In the coming nano-scale era, the power consumption of interconnects would become a serious factor which is hard to be solved. More and more researchers are devoted themselves to this field. We know that amount of electronic products increases the rate which obeys Moore's law. The total power consumption of the products becomes a huge number.

Therefore, an integrated solution from every level is no doubt highly desired. We have developed a novel placer, *LPGC Placer*, to provide a low power solution at the placement level and prove it works through experimental results.

We truly hope our work could motivate other designer's inspiration for the low power design. or minimize their product's power consumption.

We will revisit the following subjects as our future work.

Thermal Balancing

We believe in the coming SOC era, the thermal problem would become worsen than before since we put too many devices on the small area. In recent decades, the chip area has no obviously increasing trend but the number of devices on it has been increased in a very sharp rate. By the continuously advancing technology and design method, the thermal balancing problem becomes more important. Especially, by the lack of EDA tool which could handle this problem, we would devote ourselves to be the pioneer.

Gate Insertion

We find that there is an over-optimistic assumption for every low power clock gating router. They assumed that after the sophisticated calculation of locations for additional gates, each gate always can be placed on the best location it deserved.

When the design goes complicated, the available space for gate insertion would be rare. We will try to investigate this problem in the higher level of the physical design.

Multilevel Placement

Multilevel methodology is a new trend that force every traditional algorithm to try to revisit the problem as to handle the extremely large design. We will try to find every possibility for adopting the multilevel technique into our *LPGC Placer*.

Reconstruction of Optimal Gated Clock Topology

When the feature size becomes more tiny, today's optimal topology might become tomorrow's non-optimum while considering many nano-scale problems such as signal integrity, crosstalk, or the manufacture yield. We will revisit the problem of construction the optimal gated clock topology in different directions.

Bibliography

- [1] A. P. Chandrakasan, S. Sheng, R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 4, pp. 473–484, April 1992.
- [2] Ralph H. J. M. Otten, Raul Camposano and Patrick R. Groeneveld, "Design Automation for Deepsubmicron: present and future," *Proceedings of the 2002 Design, and Test in Europe Conference and Exhibition(Date'02)*, page(s):650-659
- [3] D. Duarte, V. Narayanan, and M. J. Irwin, "Impact of Technology Scaling in the Clock System Power," *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, p.59, April 25-26, 2002
- [4] David Garrett, Mircea Stan and Alvar Dean, "Challenges in clockgating for a low power ASIC methodology," *Proceedings of the 1999 international symposium on Low power electronics and design*, p.176-181, August 16-17, 1999, San Diego, California, United States
- [5] Chao, K.-Y., and Wong, D.F., "Floorplanning for low power designs," *IEEE International Symposium on Circuits and Systems*, Volume 1, 28 April-3 May 1995
Page(s):45 - 48 vol.1
- [6] Hirendu Vaishnav, and Massoud Pedram, "PCUBE: A Performance Driven Placement Algorithm for Low Power Designs," *Proceedings of European Design Automation Conference*, pp. 72-77, September 1993.

- [7] Sadiq M. Sait, Mahmood R. Minhas, and Junaid A. Khan, "Performance and low power driven VLSI standard cell placement using tabu search," *IEEE Congress on Evolutionary Computation*, May 2002, Honolulu, Hawaii, USA, pp 372-377.
- [8] Sadiq M. Sait, Habib Youssef, Junaid A. Khan, and Aiman El-Maleh "Fuzzified Iterative Algorithms for Performance Driven Low Power VLSI Placement," *19th International Conference on Computer Design (ICCD 2001)*, 23-26 Sept. 2001 Pages:484 - 487
- [9] M. Jiménez , and M. Shanblatt, "Integrating a Low-Power Objective into the Placement of Macro Block-based Layouts," *Proceedings of the IEEE. Midwest Symposium on Circuits and Systems*, pp. 62-65, Ohio, Aug. 2001.
- [10] Sadiq M. Sait, H. Youssef, and Junaid Khan, "Fuzzy simulated evolution for power and performance optimization of VLSI placement" *International Joint INNS-IEEE Conference on Neural Networks*, Volume: 1, pages: 738-743, Washington DC, July 14-19, 2001.
- [11] T.-H. Chao , J.-M. Ho , Y.-C. Hsu, "Zero skew clock net routing," *Proceedings of the 29th ACM/IEEE conference on Design automation*, p.518-523, June 08-12, 1992, Anaheim, California, United States
- [12] Michael A. B. Jackson, Arvind Srinivasan, E. S. Kuh, "Clock routing for high-performance ICs," *Proceedings of the 27th ACM/IEEE conference on Design automation*, p.573-579, June 24-27, 1990, Orlando, Florida, United States
- [13] Masato Edahiro, "A clustering-based optimization algorithm in zero-skew routings," *Proceedings of the 30th international conference on Design automation*, p.612-616, June 14-18, 1993, Dallas, Texas, United States

- [14] J. Cong, A. B. Kahng, and G. Robins, "Matching-Based Methods for High-Performance Clock Routing," *IEEE Transactions on Computer-Aided Design*, 12(8), August 1993, pp. 1157-1169.
- [15] A. B. Kahng, J. Cong and G. Robins, "High-Performance Clock Routing Based on Recursive Geometric Matching", *Proc. ACM/IEEE Design Automation Conference*," June 1991, pp. 322-327.
- [16] K. D. Boese, A. B. Kahng, "Zero-Skew Clock Routing Trees with Minimum Wire Length," *IEEE International Conference on ASIC*, pp. 1.1.1–1.1.5, Rochester, NY, September 1992.
- [17] Ting-Hai Chao, Yu-Chin Hsu, Jan-Ming Ho, Kneeth D. Boese, and Andrew B. Kahng, "Zero skew clock routing with minimum wirelength," *IEEE Transactions on Circuits and Systems*, Volume 39, Issue 11, Nov. 1992 Page(s):799 - 814
- [18] Ran-Song Tsay, "Exact Zero Skew," *IEEE Int. Conference on Computer-Aided Design (ICCAD-91)*, pp. 336- 339, 1991. Nov. 1991
- [19] M. Donno, A. Ivaldi, L. Benini, and E. Macii, "Clock-tree power optimization based on RTL clock-gating," *ACM/IEEE 40th Design Automation Conference (DAC-03)*, pages 622-627, Anaheim, CA, June 2-6 2003.
- [20] J. Oh, and M. Pedram, "Gated Clock Routing for Low-Power Microprocessor Design," *IEEE Transactions on CAD/ICAS*, Vol. 20, No. 6, pp. 715-722, June 2001.
- [21] Qi-Wang, and Roy.-S, "Power minimization by clock root gating," *Proceedings of the ASP-DAC 2003* , Pages:249 - 254, 21-24,Jan.2003
- [22] Amir H. Farrahi, Chunhong Chen, Ankur Srivastava, Gustavo Tellez, and Majid Sarrafzadeh, "Activity-Driven Clock Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, No. 6, June 2001, pp. 705-714.

- [23] J. Oh, and M. Pedram, "Gated clock routing minimizing the switched capacitance," *Proc. of Design Automation and Test in Europe*, Feb. 1998, pp. 692-697.
- [24] L. Benini, G. De Micheli, "Transformation and Synthesis of FSMs for Low-Power Gated-Clock Implementation," *IEEE Transactions on CAD/ICAS*, Vol. 15, No. 6, pp. 630-643, June 1996
- [25] Chunhong Chen, Changjun Kang and Majid Sarrafzadeh, "Activity-sensitive clock tree construction for low power," *Proceedings of the 2002 international symposium on Low power electronics and design*, August 12-14, 2002, Monterey, California, USA
- [26] Q. Wu, M. Pedram, and X. Wu. "Clock-gating and its application to low power design of sequential circuits," *IEEE Custom Integrated Circuits Conference*, 1997, pp. 479-482
- [27] Naveed Sherwani, *Algorithms for VLSI Physical Design Automation*, Kluwer Academic Publishers, 3rd edition, 1999
- [28] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd Ed, MIT Press and McGraw-Hill Book Company, 2001
- [29] Sadiq M. Sait and H. Youssef, *VLSI Physical Design Automation: Theory and Practice*, World Scientific, 1999.
- [30] Ralph H.J.M. Otten, "Automatic floorplan design," *Proceedings of the 19th conference on Design automation*, p.261-267, January 1982
- [31] D. F. Wong , C. L. Liu, "A new algorithm for floorplan design," *Proceedings of the 23rd ACM/IEEE conference on Design automation*, p.101-107, July 1986, Las Vegas, Nevada, United States

- [32] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, and S.-W. Wu, "B*-trees: A New Representation for Non-slicing Floorplans," *Proc. of ACM/IEEE Design Automation Conference (DAC-2000)*, pp. 458-463, LA, CA, June 2000. p.268-273, June 21-25, 1999, New Orleans, Louisiana, United States
- [33] Pei-Ning Guo , Chung-Kuan Cheng , Takeshi Yoshimura, "An O-tree representation of non-slicing floorplan and its applications," *Proceedings of the 36th ACM/IEEE conference on Design automation conference*,
- [34] Xianlong Hong , Gang Huang , Yici Cai , Jiangchun Gu , Sheqin Dong , Chung Kuan Cheng , Jun Gu, "Corner block list: an effective and efficient topological representation of non-slicing floorplan," *Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design*, November 05-09, 2000, San Jose, California
- [35] Jianbang Lai , Ming-Shiun Lin , Ting-Chi Wang , Li-C. Wang, "Module placement with boundary constraints using the sequence-pair representation," *Proceedings of the 2001 conference on Asia South Pacific design automation*, p.515-520, January 2001, Yokohama, Japan
- [36] Jason Cong , Gabriele Nataneli , Michail Romesis , Joseph R. Shinnerl, "An area-optimality study of floorplanning," *Proceedings of the 2004 international symposium on Physical design*, April 18-21, 2004, Phoenix, Arizona, USA
- [37] Jai-Ming Lin , Yao-Wen Chang, "TCG: a transitive closure graph-based representation for non-slicing floorplans," *Proceedings of the 38th conference on Design automation*, p.764-769, June 2001, Las Vegas, Nevada, United States
- [38] Yuchun Ma , Sheqin Dong , Xianiong Hong , Yici Cai , Chung-Kuan Cheng , Jun Gu, "VLSI floorplanning with boundary constraints based on corner block list," *Proceedings of the 2001 conference on Asia South Pacific design automation*, p.509-514, January 2001, Yokohama, Japan

- [39] Hiroshi Murata , Kunihiro Fujiyoshi , Shigetoshi Nakatake , Yoji Kajitani, “Rectangle-packing-based module placement,” *Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design*, p.472-479, November 05-09, 1995, San Jose, California, United States
- [40] H. Murata , K. Fujiyoshi , M. Kaneko, “VLSI/PCB placement with obstacles based on sequence-pair,” *Proceedings of the 1997 international symposium on Physical design*, p.26-31, April 14-16, 1997, Napa Valley, California, United States
- [41] Shigetoshi Nakatake , Kunihiro Fujiyoshi , Hiroshi Murata , Yoji Kajitani, “Module placement on BSG-structure and IC layout applications,” *Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, p.484-491, November 10-14, 1996, San Jose, California, United States
- [42] Xiaoping Tang , D. F. Wong, “FAST-SP: a fast algorithm for block placement based on sequence pair,” *Proceedings of the 2001 conference on Asia South Pacific design automation*, p.521-526, January 2001, Yokohama, Japan
- [43] D. Duarte, V. Narayanan, M. J. Irwin, “A Clock Power Model to Evaluate Impact of Architectural and Technology Optimizations,” *IEEE Transactions on VLSI Systems*, Vol. 10, No. 6, pp. 844–855, December 2002.
- [44] Joe G. Xi , Wayne W. M. Dai, “Buffer insertion and sizing under process variations for low power clock distribution,” *Proceedings of the 32nd ACM/IEEE conference on Design automation*, p.491-496, June 12-16, 1995, San Francisco, California, United States
- [45] A. Vittal, M. Marek-Sadowska, “Low-Power Buffered Clock Tree Design,” *IEEE Transactions on CAD/ICAS*, Vol. 16, No. 9, pp. 965–975, September 1997.

- [46] Jatuchai Pangjun, Sachin S. Sapatnekar, "Clock distribution using multiple voltages," *Proceedings of the 1999 international symposium on Low power electronics and design*, p.145-150, August 16-17, 1999, San Diego, California, United States
- [47] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide band amplifiers," *Journal of Applied Physics*, vol. 19, pp 55-63, January 1948.
- [48] Guang-Ming Wu, Yun-Chih Chang, and Yao-Wen Chang "Rectilinear block placement using B*-trees," *Journal of Applied Physics*, vol. 19, pp 55-63, January 1948.
- [49] David A. Huffman, "A Method for The Construction of Minimum Redundancy Codes," *Proceedings of IRE*, 40(9):1098-1101, September 1952.
- [50] T. Ishihara , H. Yasuura, "Basic experimentation on accuracy of power estimation for CMOS VLSI circuits," *Proceedings of the 1996 international symposium on Low power electronics and design*, p.117-120, August 12-14, 1996, Monterey, California, United States
- [51] Taku Uchino , Jason Cong, "An interconnect energy model considering coupling effects," *Proceedings of the 38th conference on Design automation*, p.555-558, June 2001, Las Vegas, Nevada, United States
- [52] T. Uchino , F. Minami , M. Murakata , T. Mitsuhashi, "Switching activity analysis for sequential circuits using Boolean approximation method," *Proceedings of the 1996 international symposium on Low power electronics and design*, p.79-84, August 12-14, 1996, Monterey, California, United States
- [53] Masako Murofushi , Takashi Ishioka , Masami Murakata , Takashi Mitsuhashi, "Layout driven re-synthesis for low power consumption LSIs," *Proceedings of the 34th annual conference on Design automation conference*, p.666-669, June 09-13, 1997, Anaheim, California, United States

- [54] Dennis J. Ciplickas , Ronald A. Rohrer, “Expected current distributions for CMOS circuits,” *Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, p.589-592, November 10-14, 1996, San Jose, California, United States
- [55] Jose C. Costa , Jose C. Monteiro , Srinivas Devadas, “Switching activity estimation using limited depth reconvergent path analysis,” *Proceedings of the 1997 international symposium on Low power electronics and design*, p.184-189, August 18-20, 1997, Monterey, California, United States
- [56] Michael S. Hsiao , Elizabeth M. Rudnick , Janak H. Patel, “Effects of delay models on peak power estimation of VLSI sequential circuits,” *Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design*, p.45-51, November 09-13, 1997, San Jose, California, United States
- [57] Massoud Pedram, “Power minimization in IC design: principles and applications,” *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, v.1 n.1, p.3-56, Jan. 1996