UNIVERSITY OF CALIFORNIA
SANTA CRUZ

ANALYSIS AND APPLICATION OF INDUCTANCE IN CLOCK DISTRIBUTION NETWORKS

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Xuchu Hu

March 2012

The Dissertation of Xuchu Hu
is approved:

Professor Matthew R. Guthaus, Chair

Professor Jose Renau

Dr. Chin Ngai Sze

Dean Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Analysis and Application of Inductance

in Clock Distribution Networks

by

Xuchu Hu

With better manufacturing technologies, each generation of processors grows smaller, faster, and consumes more power. As microprocessors are operating at multi-GHz speed, power consumption has become a major concern in modern processor design. Especially in portable devices which are battery operated, low power design becomes extreme important.

The on-chip clock distribution network (CDN) consumes in excess of 35% of total chip power and occasionally as much as 70% [61]. Most of this power is due to the dynamic switching of the large number of sequential element clock pins that span the entire chip. Clock distribution using inductance (Resonant clock) has a potential to reduce the maximum power consumption without degrading the clock network performance. Some previous research works demonstrated power savings by connecting extra inductors to clock network. Compared with clock trees, clock grids are often used in high performance processors which operate at higher frequency and consume more power. Previous resonant works either assume simplified clock network or only consider a small sector of the clock network. Inductance is often used in RF designs. In digital circuits, designers usually try to minimize the inductance effect of long interconnections. So the resonant clock synthesis which is related to both digital and analogy design has not been well studied.

In this thesis, a methodology to design low power resonant clock grid is described. The key synthesis procedures involved in resonant clock grid design are discussed. The CDN which has a top-level tree driving a resonant grid shows at least 40% power savings and 53% buffer area reduction while using only 30% of a single metal layer for inductors on average. With more advanced on-chip inductor integration techniques, resonant clock grids hold the potential to save up to 90% of the clock grid power. The automated methods can make these multi-disciplinary clocking techniques practical for use in high-performance ASIC designs. At the end of the thesis, the practical issues of resonant clock will be discussed.

Acknowledgments

First of all, I would like to express my deep and sincere gratitude to my advisor, Professor Matthew R. Guthaus, for all his contributions of time, ideas, and funding to make my Ph.D experience productive and stimulating. It is my honor to be his first Ph.D graduate. I appreciate his hands-on help in the initial stage and his encouragement for innovation as a senior Ph.D student. His consistent encouragement and support helped me to complete the Ph.D smoothly. I am also thankful for the excellent example he has provided as an enthusiastic and diligent researcher.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor Jose Renau and Dr. Cliff (Chin Ngai) Sze for their time, dedication and insightful comments.

I thank my fellow labmates in UCSC VLSI-DA group: Seokjoong Kim, Sheldon Logan, Rajsaktish Sankaranarayanan, Marcelo Siero, Walter Condley, Derek Chan and Keven Woo, for both technical discussions and all kinds of fun discussions.

I also wish to thank the graduate advisor and the international student advisor, Carol Mullane and Ally Modzeleski. They are always willing to help and provide helpful suggestions. They helped me with a lot of paperwork and special problems.

I would say without the help of my sincere friends, I could not be able to successfully finish my Ph.D study. I am grateful to my friends, Bin Wu, Xinhua Hu and Jingpei Yang, who took care of my daily life after surgery and helped me go though those tough times.

Last but not the least, I would thank my parents for always being there when I

needed them most, and always considering everything for me. They never complain how far I am away from home and how infrequently I go home to visit them. They deserve far more credit than I can ever give them.

Chapter 1

Introduction

The integrated circuit (IC) industry has undergone more evolutionary change in the past 50 years than any other industry and has changed the world tremendously. When the first IC was created in 1958 by Jack Kilby, it contained only a few transistors. The latest Intel processor Poulson [62] has been introduced in ISSCC 2011 and contains 3.1 billion transistors. Early integrated circuit is called SSL (small-scale integration), and later developed to MSI (medium-scale integration), LSI (large-scale integration) and VLSI (very large-scale integration). Nowadays, ULSI which stands for "ultra-large-scale integration" is used to describe an IC with more than one million components per chip.

To describe the technical details of a processor, CPU frequency is always the number one spec. CPU frequency is the internal operating frequency of the processor. Generally, the higher frequency is, the faster and the better the processor is. So it is not surprising that the clock design, or to be more precise, the clock distribution network (CDN) in modern chip design is an important aspect, which, to a certain extent, decides how fast the chip can work.

With the sustained and rapid development of IC industry, however, we rarely see processors working over 5GHz as IC designers are facing more challenging power, cooling, and stability problems when targeting for higher clock frequency. This speed threshold is because of the extreme power dissipation of the circuit. According to the CMOS circuit dynamic power consumption equation,

$$P = \alpha C V^2 f; \tag{1.1}$$

the dynamic power is proportional to the circuit capacitance (C), supply voltage (V), clock frequency (f) and switching activity (α). Doubling the clock frequency is at the expense of twice power dissipation. The power density of the highly-integrated chip with millions of transistors requires extremely efficient cooling techniques to make the chip reliable.

Clock gating, power gating, dynamic frequency scaling, and dynamic voltage scaling are used at different design abstractions to reduce dynamic power by carefully looking into Equation 1.1 and minimizing each factor. However, the power is usually saved by exploiting inactivity or dynamically adjusting performance. Multi-core processors are considered as a viable solution for keeping clock rates and heat production manageable. Multi-core processors operate at a lower frequency but still achieve the same or even better performance than a single core processor by distributing the tasks to multiple CPU cores. However multi-core still can not break the physical constraints preventing frequency scaling.

The clock frequency f determines the single-thread performance. Because of its high activity and great fanout, the on-chip clock distribution network (CDN) consumes in excess of 35% of total chip power and occasionally as much as 70% [61]. Obviously,

reducing the power of clock network is a great help to control chip power. However, considering process, voltage, and temperature variation issues arised from technical scaling, it is tremendously difficult to design a low power but still robust and stable clock network.

As frequency and power are correlated, only a breakthrough technology could be able to solve the power and frequency problem by breaking this relationship. One possible solution to this problem is resonant clocks which will be investigated in this thesis.

1.1 Thesis Contributions and Outline

The on-chip interconnect inductance leads to signal ringing, signal reflection and additional inductive crosstalk under fast slew rates. Inductance in clock wires with high switching activity of clock signals can also cause undesirable effects such as Electromagnetic Interference (EMI) which is a primary concern among many RF designers because unchecked eletromagetic field may interfere with nearby devices. Clock shielding is used to minimize the inductance effect [39]. The EMI reduction in clock network is briefly investigated in Appendix A [1]. On the other hand, research works [18, 32] show that on-chip inductance can be used to improve the performance of high-speed integrated circuits. Inductance improves the signal slew rate, eliminates short-circuit power and reduces the area of the active devices and repeaters inserted to optimize the performance of long interconnects. However, in most recently published clock design works [13, 41, 88], inductance of the CDN is ignored.

Besides the improvement of clock signal slew rate, the application of inducatance has great potential to save power in clock network. This thesis proposes a design automa-

[1] This work is not directly related to the inductance application in resonant clock.

tion methodology to synthesis low-power and robustness clock network by utilizing the resonance phenomenon. By adding inductors to clock networks, the power of CDN can be reduced by up to 90% without scracific frequency, or robustness.

This dissertation begins with an overview of clock distribution networks in Chapter 2. The general structures and metrics in CDN design are introduced. In Chapter 3, resonant theory is introduced. With the background of the resonant theory, different types of resonant clocks which utilize resonant phenomenon in clock network to save power are introduced with their advantages and disadvantages. At the end of this chapter, on-chip inductors and on-chip capacitors, are introduced. Chapter 4 is an overview of resonant clock synthesis. The application of small signal analysis (AC analysis) is introduced at the begining of this chapter, which is different from previous transient analysis in CDN design. Special challenging issues in resonant clock design are summaried with a complete synthesis flow to address these problems. Chaper 5, 6 and 7 are the three major procedures in the resonant clock synthesis: LC tank placement/sizing, buffer placement/sizing and phase tuning. Each chapter introduces the methodologies used in each step to synthesis resonant clocks. Chapter 8 summarizes the experimental setup, and experimental design to validate the proposed methods. Chapter 9 includes all the experimental results with different experimental setups followed with conclusions. Chapter 10 raises some practical issues related to application of resonant clock, and provides solutions to these special issues. Finally, Chapter 11 summarizes the work presented in this dissertation and describes potential areas for further studies.

Chapter 2

Clock Distribution Network Background

Normally, an integrated Phase-Lock Loop (PLL) generates the clock signal and the CDN will distribute the clock signal from the PLL to all the registers/flip-flops across the chip. Figure 2.1 is an example to show how the combinational circuits are synchronized by the clock signal which is generated by the PLL. The red solid line in the figure represents the clock from the source to each flip-flop. These flip-flops are often called *clock sinks*. Buffers are inserted in the clock network for delay, slew, and noise optimization.

2.1 Clock Network Design Metrics

Skew: Skew is defined as the maximum difference among the source to sink delays. In the example in Figure 2.1, the clock skew is the difference between the earliest and latest arrival time in Clk-a, Clk-b, Clk-c and Clk-d. Buffering, symmetric structure, and redundant connections are commonly used methodologies [22, 58, 77] to minimize clock skew.

Power: While the clock signal has to traverse longer distance with the increasing chip size, the CDN consumes more power and low-power clock design became an urgent require-

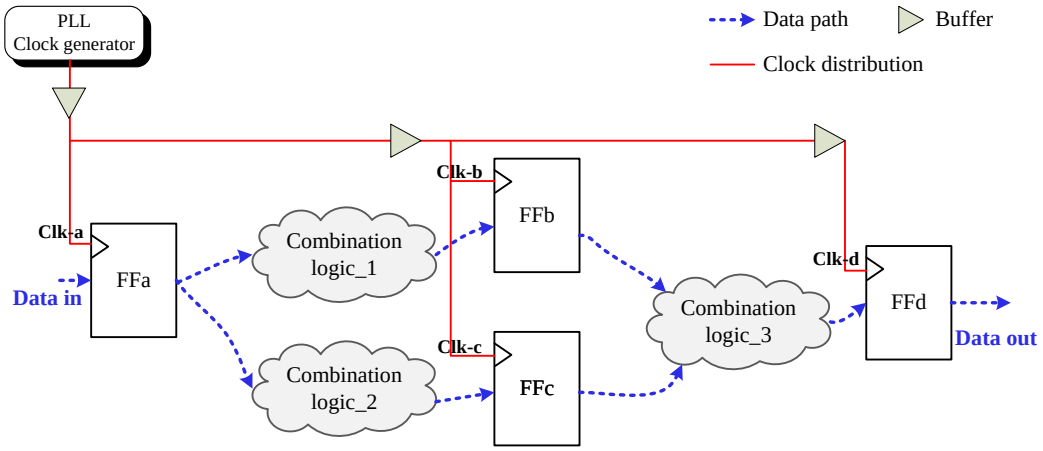


Figure 2.1: Combination circuits synchronized by clock signal

ment. Figure 2.2 shows the processor power and frequency trend over the time. From ITRS roadmap, the maximum power in the future will be limited to 150W. Clock gating is the most commonly used low-power method [10, 11, 49]. Dynamic voltage and frequency scaling (DVFS) [48] and low-voltage swing clocks [97] are also used in low-power clock design.

Variations: Another consideration in CDN design is the uncertainty in the CMOS design technologies. Technology scaling has lead to an increase in on-die PVT (process, voltage, temperature) variations which may degrade performance. Many previous works [24–29, 40, 74, 80, 94] discussed the variation problem in clock design.

2.2 Clock Structure

There are various clock distribution structures. In a rough classification, CDN structures can be classified into four types: trees, spines, grids and hybrid structures. Fig-

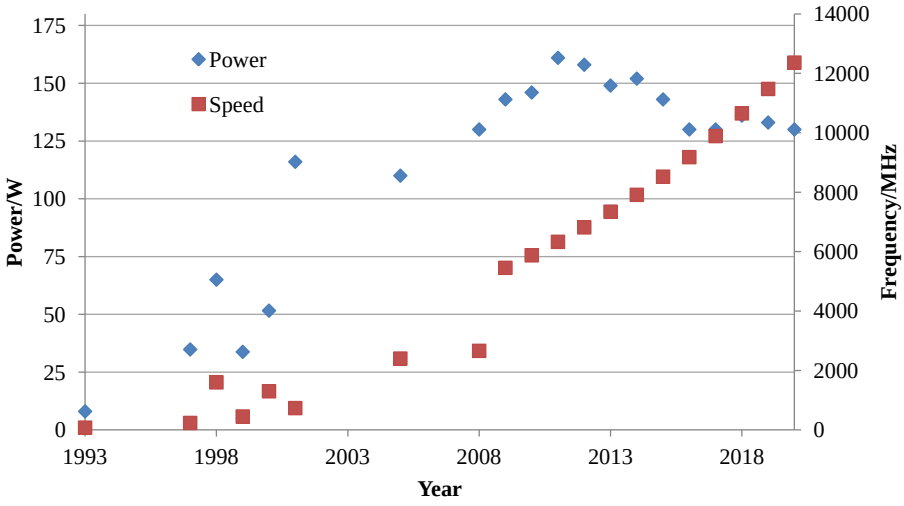


Figure 2.2: Power and frequency trend over the years. Data before 2009 is from [57], data after 2010 is from ITRS2011 [79]

ure 2.3 and Figure 2.4 are tree structures. The symmetric structure of H-tree and binary tree is good for skew and often consume less power. Though H-trees are robust to die-to-die variations, tree structures are very sensitive to variations. The spine structure is a specific implementation of a binary tree. Figure 2.5 is a clock spine structure. The spine structure is simple to implement but there will be residual skew due to asymmetry. A clock grid is formed by a set of vertical and horizontal wires as shown in Figure 2.6. Figure 2.7 is a hybrid structure with a top-level H-tree driving a clock grid.

Clock grid is able to provide low skew and very good robustness to variations because of the redundant connections, so clock grid is often used high-performance designs. However, power consumption is the major concern in clock grid application. Recently, some clock grid optimization methods [30, 59, 88] are proposed to reduce the clock grid power. By removing redundant wires, the total capacitance of the clock grid can be reduced and hence reduce the power. The buffer placement/sizing optimization methods find a more

efficient way to utilize buffers to drive the clock grid to reduce the buffer area and short circuit power.

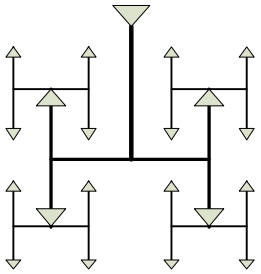


Figure 2.3: H-tree

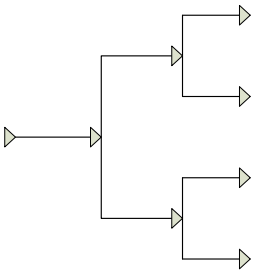


Figure 2.4: Binary tree

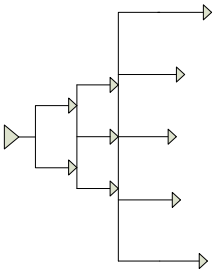


Figure 2.5: Clock spine

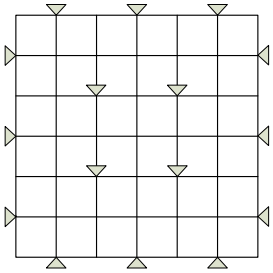


Figure 2.6: Clock grid

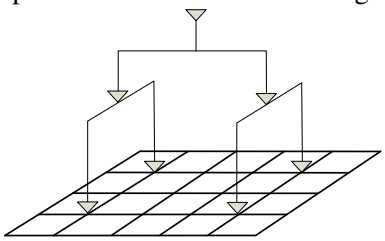


Figure 2.7: Hybrid clock: H-tree driving grid

2.3 Analytical Models in CDN

Clock signal generator, clock sink, buffer and wire constitute the clock network. Usually, the clock generator is not included in the clock network synthesis. Therefore, this section will focus on the sink, buffer and interconnect analytical models.

Sink models

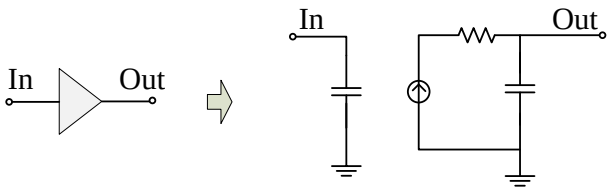


Figure 2.8: Buffer RC model

A clock sink is either a flip-flop or a latch. It can also be a buffer in the hierarchical clock network. The clock sink is often modeled as a capacitor with a specific capacitance the same as the flip-flop (or latch, buffer) capacitance seen by the clock network.

Buffer models

A buffer/inverter is often modeled as a switch-level RC circuit as Figure 2.8 in previous clock network designs [82, 87]. The delay of buffer can be calculated according to the buffer size, buffer load and the intrinsic delay. This model is able to estimate the buffer delay very quickly but the accuracy is getting worse while as the interconnect resistance gets more significant. In recent clock synthesis, the BSIM model is more often used in accurate timing analysis.

Wire models

Three wire segment models are often discussed in the CDN analysis: RC model, RLC model and transmission line model as shown in Figure 2.9.

The RC model is a first-order extraction of wire. Usually, a wire segment is replaced by a resistor and two capacitors which is also called π-model. The RC model is simple. When frequency is not very high and wire length is relatively short, RC model is often used.

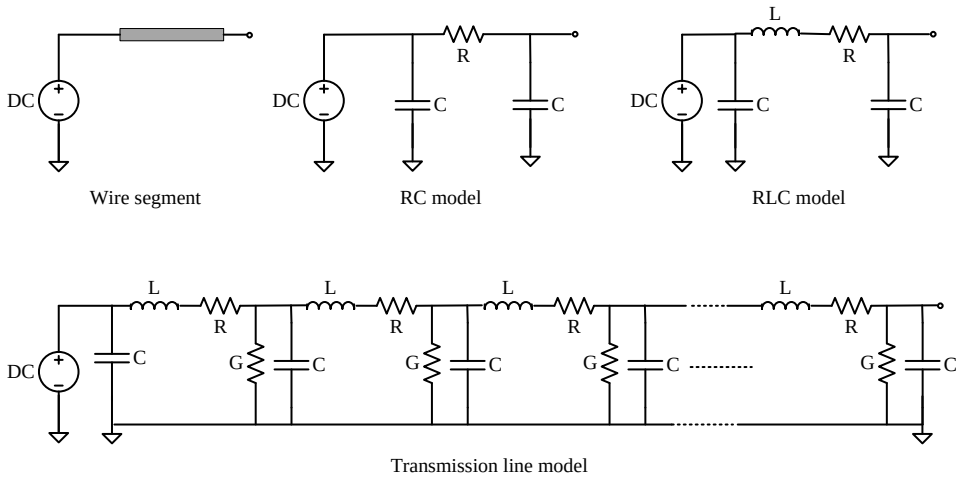


Figure 2.9: Wire models

However, due to increasing clock speeds, increasing interconnect lengths and decreasing signal rise times, the inductive effects of on-chip interconnects become more significant. The on-chip inductance can cause reflections and overshoots which may cause reliability problems. The on-chip interconnect inductance also effects delay and rise/fall time. There are many research works [34, 36, 37, 43, 44] about the inductance calculation, extraction and modeling.

Transmission line models are the most accurate and closet model. As shown in Figure 2.9, the lossy transmission line model represents a wire with many distributed "lumps" of R, L, and C elements. Because of the cascaded structure, the T-line model often takes much longer time than RC and RLC models. In real designs, if the signal frequency is low compared to the size of the circuit, a reasonable approximation can be used to simplify the circuit for calculation when doing transient analysis [16].

Though unchecked inductance will introduce many uncertainties, the application

of inductance in clock network is a fundamental solution to break the frequency and power relationship as in Equation 1.1 which is an insolvable problem in present low power design. Moreover, this method is more applicable in high frequency designs which often consume more power.

Chapter 3

Clock Distribution Using Inductance: Resonant Clock

As discussed before, power consumption is a predominant challenge in modern high-performance systems. One way to decrease this dynamic power consumption is with resonant clocking which can circumvent the dynamic power Equation 1.1.

Many previous works related to inductance try to minimize the effect of inductance [35, 42, 43, 60], however, a large amount of power can potentially be saved by using this same inductance in resonant clocks.

In this chapter, the background of resonant theory is introduced. Different types of resonant clocks and their advantages and disadvantages are discussed. Finally, the extra cost of resonant clock will be discussed.

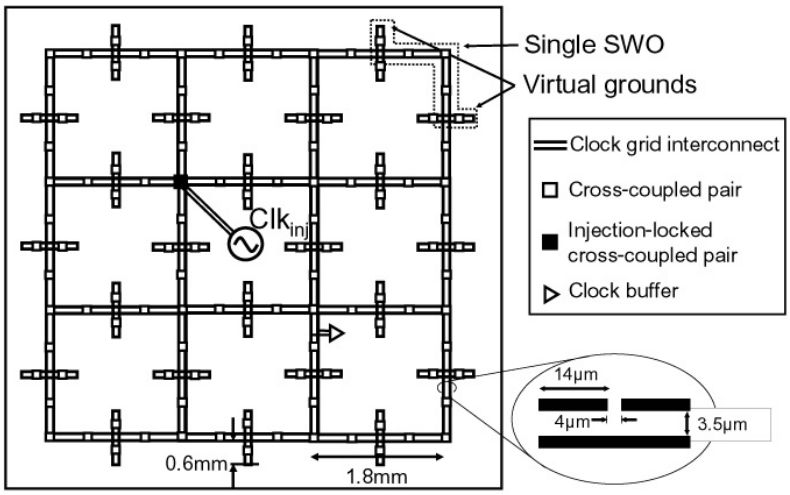


Figure 3.1: Standing-wave clock distribution network [50]

3.1 Types of Resonant Clocks

There are various approaches to resonant clocks including standing wave [50], rotary/salphasic [14, 71, 86], and inductor-capacitor (LC) tank [7, 8, 21, 92, 98].

3.1.1 Standing Wave Resonant Clock

A standing wave is a non-traveling vibration formed by the interference of two harmonic waves of the same frequency and amplitude. In standing wave clocks, the phases of all points are the same which is ideal for skew but the maximum amplitude varies with position.

A simple method to generate a standing wave is to send a wave along a transmission line and reflect it back at the end of the transmission line [50]. Figure 3.1 is a standing wave clock distribution network.

The loss of the transmission lines is one of the practical issues of standing wave

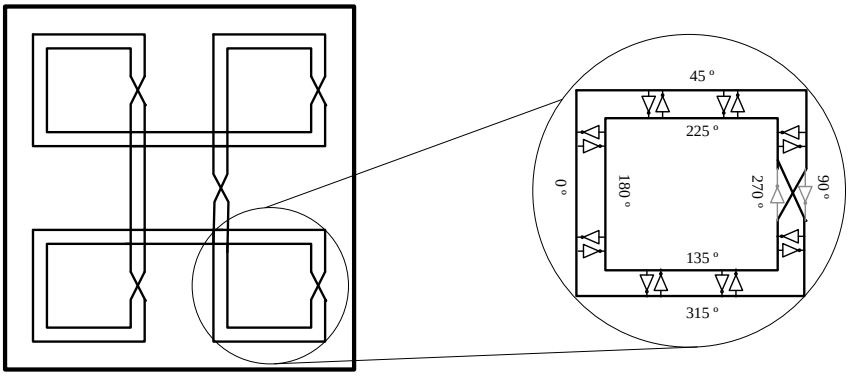


Figure 3.2: Rotary resonant clock [86]

clock. The transmission line loss attenuates the amplitude of the waves and hence, introduces a residual traveling wave that leads to clock skew. Distributed trans conductors [5, 19, 86] are introduced to compensate for signal attenuation because of wire loss. Another practical issues of standing wave clock is the varying amplitude. To connect standing wave clock to conventional clock network, specific two-stage clock buffers [45] are required to convert the low-swing standing wave signal to digital level signal.

3.1.2 Rotary Resonant Clock

A rotary clock (shown as in Figure 3.2) is a closed parallel loop formed by an inner and outer transmission line [86]. When a signal is applied to the cross-connected inner and outer loop, the signal could travel on this loop indefinitely if there is no loss. Same as standing wave clock, negative impedance components are needed to overcome the signal attenuation.

Unlike standing wave clocks, the maximum amplitude of all positions in a rotary clock loop are the same while the phase varies with position. The differences between sink

phases bring about extra work in clock timing and synchronization since clock sinks may be attached to different positions on the loop. Though there are two-phase latched logic [86], connecting conventional signal D-latch to the rotary clock loop needs extra work on skew scheduling which makes it inapplicable in practical designs. Up to now, there is no logic synthesis tools supporting the design with different phase clock network.

3.1.3 Distributed Inductor-Capacitor (LC) Resonant Clock

Both standing wave clocks and rotary clocks have severe restrictions on the physical implementation so that the correct phase or amplitude is used. This makes them difficult to apply in practice, because most clock distributions are asymmetric and unbalanced. On the other hand, LC (inductor-capacitor) tank resonant clocks ideally have constant phase and constant magnitude which allows these CDNs to have structure similar to previous non-resonant clocks. Industry has demonstrated several variants of distributed LC resonant clocks on uniform H-trees [7,8] and academia has demonstrated several monolithic LC tank clocks on small, low-speed designs [21, 92, 98].

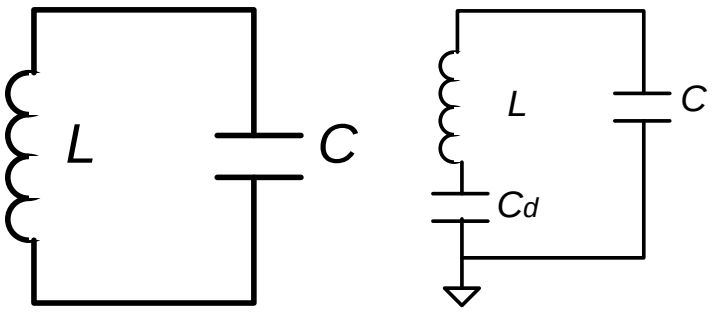
LC-based resonant clocks are extensions of conventional clock with the addition of spiral inductors. Therefore, this thesis is focused on the application of inductor in LC-based resonant clocks. In the remainder of this thesis, the term "resonant clock" refers to a distributed LC resonant clock network.

3.2 Theory of LC Resonant Oscillators

The admittance of inductance, capacitance, and resistance for Alternating Current (AC) signals are most often represented using complex numbers: $G_R = 1/R$, $G_L = 1/(j\omega L)$, and $G_C = j\omega C$ that depend on the frequency of operation (ω). A parallel or series LC tank has capacitive reactance and inductive reactance exactly equal at one specific frequency which is known as the *resonant frequency*. By equating the inductive and capacitive reactances ($G_L = G_C$), we can obtain the unique resonant frequency as

$$f = \frac{\omega}{2\pi} = \frac{1}{2\pi\sqrt{LC}} \tag{3.1}$$

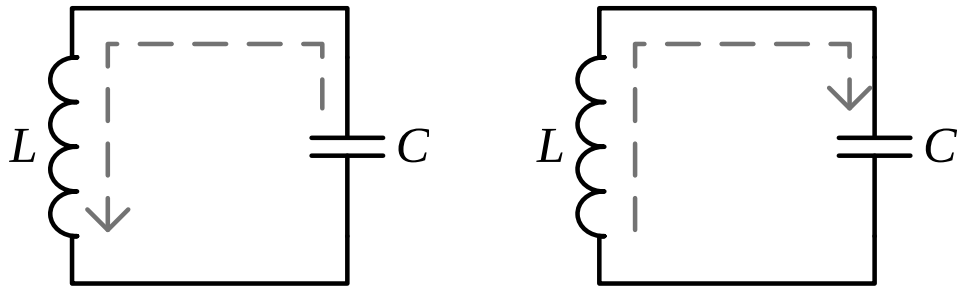
where L is the inductor size and C is the capacitor size. In a parallel LC tank, this ideally presents infinite impedance, while in a series LC tank this ideally presents zero impedance. This has been used extensively in building high-quality passive filter networks, but it also has enormous potential to lower the power required to drive clock distributions.



(a) LC tank oscillator

(b) LC tank oscillator with decoupling capacitance to positively bias the voltage of capacitor C

Figure 3.3: LC Tank Oscillators



(a) Energy transfer from electric field in capacitor C to magnetic field in inductor L

(b) Energy transfer from magnetic field in inductor L to electric field in capacitor C

Figure 3.4: Energy exchange in LC tank oscillators

When an AC signal such as a clock is applied to these LC tanks, they form a *resonant oscillator*. Resonant oscillators can circumvent the active power in Equation 1.1 by using inductive reactance. In a clock distribution, the wire and sink capacitances form the clock distribution capacitance (C) as shown in Figure 3.3(a), which is parallel to L. The energy in the system is exchanged between an electric field in C and a magnetic field in the inductor (L) as in Figure 3.4(a). The charge stored on the capacitor C is discharged through the inductor L which then induces a current and subsequently the opposite charge on the capacitor in Figure 3.4(b). The voltage across C is maximized when this voltage and the input current are in phase at the resonant frequency. With no parasitic resistances, this will oscillate indefinitely.

A simple LC tank will naturally oscillate between a positive and negative voltage so we must also add an additional decoupling capacitor (C_d) to the grounded end of each inductor as shown in Figure 3.3(b) to positively bias the voltage to a compatible CMOS logic range (0 to V_{dd}) [7, 8]. This additional capacitor, however, creates an additional undesirable parasitic series LC tank. Care must be taken in sizing C_d to ensure that the series LC tank

resonant frequency is distinctly separated from the parallel LC tank resonant frequency or, more formally,

$$\frac{1}{2\pi\sqrt{LC_d}} \ll \frac{1}{2\pi\sqrt{LC}}. \tag{3.2}$$

3.3 Resonant Clock Grid

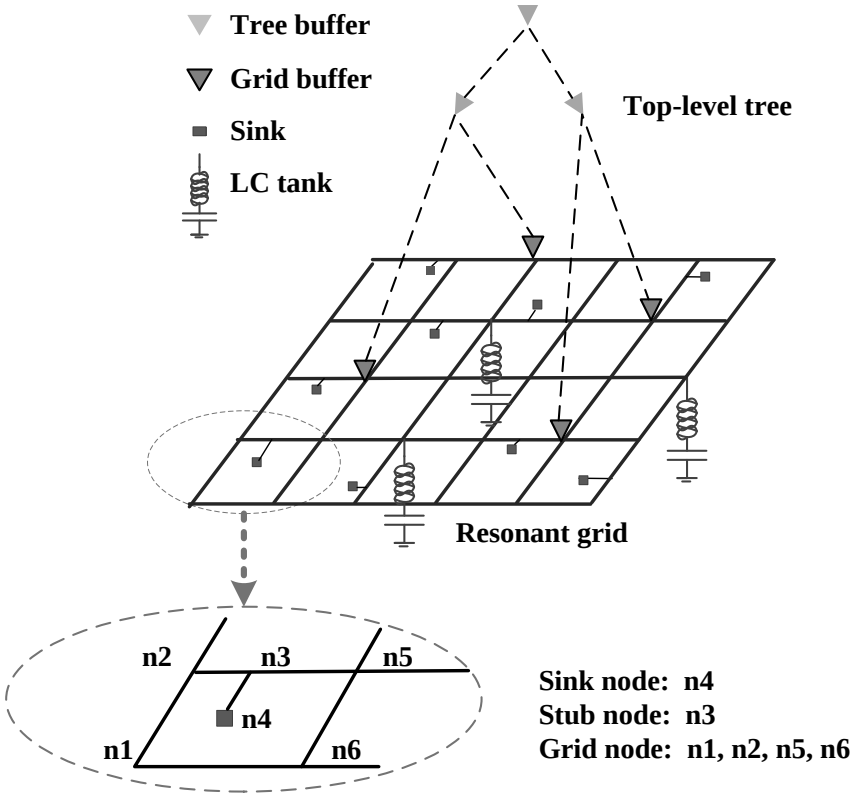


Figure 3.5: Structure of a distributed LC resonant clock grid

Clock grids are often used for regional clock distribution in high-performance designs due to their robustness to process and environmental variations. A clock grid is formed by a set of vertical and horizontal wires with stubs connecting clock sinks. A top-level tree is often used to drive the grid. A *resonant clock grid* is a clock grid with

distributed LC tanks which resonate at the operating frequency attached directly to the grid wires. Figure 3.5 shows a resonant grid with a top-level tree and three such LC tanks. The intersection of grid wires are called *grid nodes*. Clock sinks are connected to grid wires through *stubs*. The intersection nodes of grid wires and stubs are called *stub nodes*. A *node capacitance* is the sum of the capacitance of all wires, sinks and buffers which are adjacent to a node. The grid wire, grid buffer and sink capacitances form the capacitor (C) in Figure 3.3(a) and, as such, the inductors (L) and decoupling capacitors (C_d) should be placed/sized according to the capacitive load of the grid. The capacitors and inductors have parasitic resistance and the capacitance C is distributed throughout a chip which introduces parasitic resistance from the CDN itself. The grid buffers are inserted to compensate for the energy loss of the tanks due to parasitics.

3.4 On-Chip Inductors

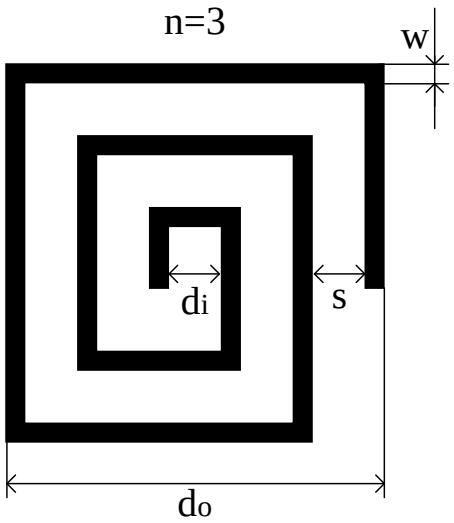


Figure 3.6: On-chip square spiral inductor made with normal metal layers

The on-chip inductors used in LC tank resonant clock grids can be created using normal metal layers, special layers in RF processes, or using free-standing MEMs devices [68]. Multi-layer spiral inductors can also use mutual inductance between adjacent spirals increases the total inductance [12,46]. The total inductance of a n-layer inductor inductor is n^2 of one inductor [99]. These inductors can be planar or 3D. With 3D inductor techniques, inductors can be designed with less area and comparable performance.

Our analysis conservatively assumes that we will be using a common commercial process and that on-chip inductors will be created using single-layer square spiral topologies with ground shields [93] as shown in Figure 3.6. The inductance of inductor has two parts, self and mutual inductance. The total inductance is the sum of self inductance, positive mutual inductance which contributes to increase self magnetic flux, and the negative mutual inductance which reduces the magnetic flux. Mutual inductance between two segments depends on their distance, shape, intersection angle [2] and there are many algorithm and models have been provided to calculate the self and mutual inductance [23, 67].

In the model we use [2], the inductance is

$$L = 0.0002l\left[ln\frac{2l}{(w+t)} + 0.5 + \frac{w+t}{3l}\right]nH \tag{3.3}$$

where the n is the number of turns, w is the width of trace, t is the thickness of the metal, l is the length of trace and s is the spacing between turns. Given n, s , w, and d_i (the inner diameter of the square spiral inductor), the metal area occupied by an inductor is

$$Area = d_o^2 = (d_i + 2n(s+w))^2 \tag{3.4}$$

where d_o is the outer diameter.

Significant parasitic resistances and capacitances are also associated with an inductor and can alter their efficiency. The quality factor Q of an inductor is the ratio of its reactance to its resistance at a certain frequency

$$Q = \frac{\omega L}{R}. \tag{3.5}$$

The thick top metal layer with higher conductivity is often used to implemented the inductor to reduce the ohmic losses.

Given modern process dimensions and oxides, it is feasible to create high-Q spiral inductors with densities of $32 - 40nH/mm^2$ at chip speeds of $1 - 5$ GHz [15]. This is the first work to consider actual inductor area based on Equation 3.4 during the distributed LC tank clock synthesis while all previous LC tank resonant clock works have not quantify the inductor overhead in this way. All inductors are created on the single, top-level metal layer and do not obstruct active devices or other routing layers. In later results, the percentage of this metal layer area is used to quantify the overhead consumed by all on-chip inductors. Specialized EM inductor simulation tools such as ASITIC [47] can be used to model the inductors at various accuracies.

3.5 On-Chip Capacitors

The on-chip decoupling capacitors which used in resonant clock are also widely utilized to manage transient power supply noise and are commonly implemented as polysilicon-insulator-polysilicon (PIP), MOS-based or metal-insulator-metal (MIM) capacitors [33].

Chapter 4

Resonant Clock Grid Synthesis Overview

The basic elements in CDN and their analytical models have been introduced in Chapter 2.3. In this chapter, the resonant clock analysis method is introduced which is different from the method used in non-resonant clock design. The inductors in the resonant clock cause special problems which do not exist in non-resonant clock network. All these challenges are summarized in this chapter. At the end of this chapter, a complete resonant clock synthesis flow is provided with each step addressing different challenges.

4.1 Resonant Clock Analysis

In clock tree synthesis, Elmore delay is widely used to quickly estimate the timing [17, 38, 76]. In the clock grid synthesis, as there are more components and possibly loops in the circuits, some other methods has been proposed to simplify the massively network to speed the analysis of clock grid. By merging the drivers and reducing model order [81, 89, 90], the massive network size is reduced and fast clock mesh simulation is possible.

However, these kinds of fast simulation methods are not applicable to resonant clock grids. If two buffers are merged together as a super driver as in [90], the new super buffer shows different parasitic resistance and capacitance to the LC tanks which changes the resonant behavior. How this change effects the resonant grid is discussed in detail later. Moreover, the buffer driving ability can not be easily estimated by logical effort due to the interaction with the LC tanks. These fast simulation methods used in non-resonant grid analysis are not applicable to resonant grid analysis.

4.1.1 Transient Analysis

A widely used clock network analysis method is transient analysis. At every time point in transient analysis, the time-dependent components, such as capacitors and inductors, are replaced by their equivalent circuit according to their instantaneous I-V characters [56]. At different time points, these I-V relationships change, so the transient analysis has to run many iterations and it is usually very time consuming for large circuits. With detailed calculations at each time point, transient analysis is able to plot any time-domain waveform, like voltage, current and power. Because the clock signal varies with time in a clock cycle, transient analysis is usually used to measure power and to check that the timing related clock metrics, such as skew and slew, are satisfied.

4.1.2 AC Analysis

In the frequency-domain, small signal analysis, or Alternating Current analysis (AC analysis) is used to simulate the circuit at different frequencies. Similar to transient analysis, AC frequency sweep is a small-signal linear analysis. While transient analysis

solves circuit equation at different timing analysis points, AC analysis build the circuit equations at different frequencies and solves the linear equations to simulate the circuit in the frequency-domain. The AC analysis is a linear modified nodal analysis, thus no iteration is necessary and it is much faster than transient analysis.

To consider the frequency effect on the circuit, complex quantities are applied in the AC analysis and non-linear circuit components are replaced by their linear small-signal models. As introduced in Chapter 3.2, the voltage, current, and RLC in a circuit are represented as complex numbers which include both phase and magnitude information. The clock grid circuit is written as a set of nodal equations which describe all the elements in the circuit. Using the complex admittance values of the CDN wires, sinks and LC tanks, we formulate the resonant grid as a complex linear system

$$GV = I \tag{4.1}$$

where G is the complex admittance matrix of the mesh, I are the mesh buffer driving currents, and V are the (complex) voltages of each sink/node in the grid. The complex voltages V include information about the voltage amplitude and phase of each node in the CDN.

The complex voltage is $v = x + jy$, where x and y are both real numbers. The amplitude of v is $\sqrt{x^2 + y^2}$ and the phase of v is $tan^{-1}(\frac{y}{x})$. Figure 4.1 shows the amplitude and phase relationship between two AC signals. There is a 90 degree phase between signal A and signal B.

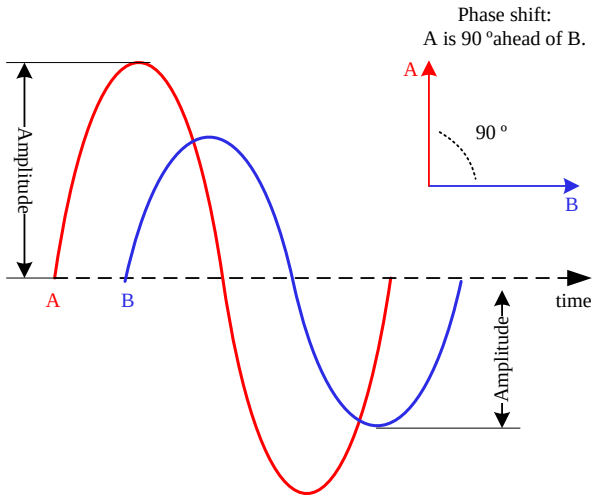


Figure 4.1: Amplitude and phase of AC signals

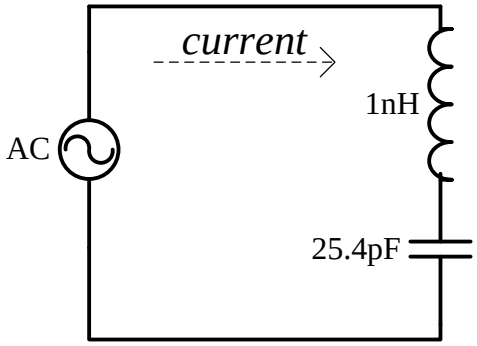
4.2 Challenges in Resonant Clock Synthesis

4.2.1 Resonant Frequency Shift

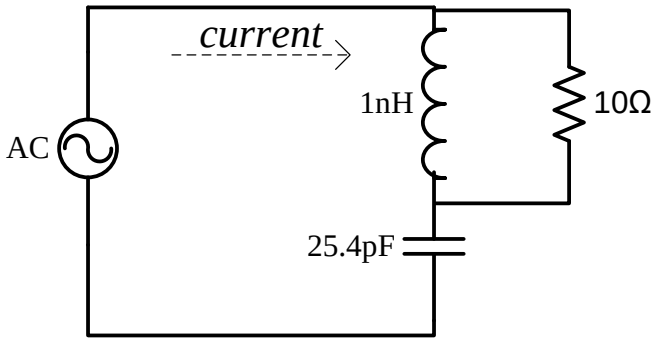
The parasitic resistances and capacitances in the CDN alter the resonant frequency. To demonstrate the frequency shift caused by parasitics, we design a set of circuits and verify their resonant characteristics with AC analysis.

Figure 4.2(a) is an ideal serial LC circuit without parasitic resistance. According to the resonant frequency calculation Equation 3.1, the resonant frequency is about 1GHz. Figure 4.3(a) plots the current of this LC circuit by running AC analysis. The current at the resonant frequency is maximized and because it is an ideal LC tank, the waveform is very sharp showing a very good qualify factor Q which is introduced in Chapter 3.4.

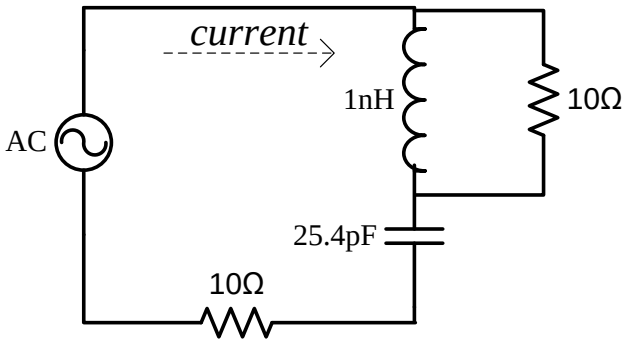
Figure 4.2(b) is the same LC circuit but considering the parasitic resistance of inductor. The 10Ω resistor is selected to illustrate the parasitic resistance effect. Figure 4.3(b)



(a) Ideal LC resonant circuit

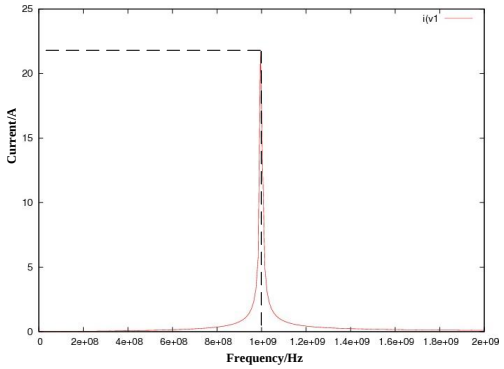


(b) LC resonant circuit with resistance in parallel with L, the parallel resistance represents the parasitic of inductor L

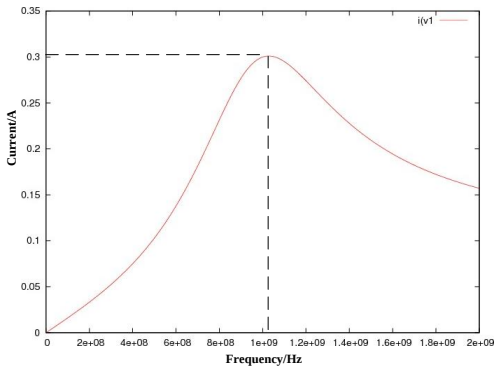


(c) LC resonant circuit with resistance in parallel with L and resistance in series with R, the series resistance represents the parasitic of wire

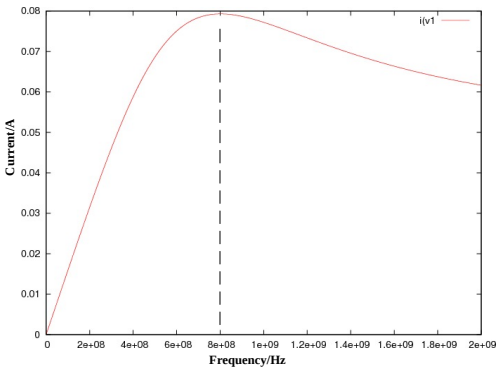
Figure 4.2: Parasitic resistances and capacitances in LC circuit alter the resonant frequency.



(a) AC analysis of ideal serial LC circuit



(b) AC analysis of LC circuit with resistance in parallel with L



(c) AC analysis of LC circuit with resistance in parallel with L and resistance in serial with R

Figure 4.3: Current plots in AC analysis of circuit 4.2

is the corresponding current plot from AC analysis. The current reaches the maximum value around 1GHz, which means the parallel parasitic resistance of the inductor doesn't change the resonant frequency much. However, the current waveform is very different from Figure 4.3(a). The wave is flatter which means the quality factor Q of this circuit is smaller than the ideal LC circuit.

In Figure 4.2(c), another series resistor is added which represents the interconnect wire resistance or the capacitor parasitic resistance. The AC analysis 4.3(c) clearly shows that the current reaches its maximum at 0.8GHz which means the resonant frequency of this circuit has been shifted to 0.8GHz because of the new serial 10Ω resistor. The much flatter waveform shows the Q factor of this circuit is even worse.

Though the parallel resistor of inductor in Figure 4.2(b) doesn't significantly change resonant frequency, the serial resistor in Figure 4.2(c) shifts the resonant frequency by 20% even the circuit is as simple as a single LC tank like Figure 4.2. The clock grid structure is much more complex with a huge number of components, loops, parallel and series resistors. The LC tanks placement and sizing should carefully consider this parasitic resistance, otherwise, the final resonant frequency may shift away from the designated clock frequency as in previous hand-made designs [9, 21].

4.2.2 Signal Attenuation

On closer inspection, besides alternating the resonant frequency, the CDN wire resistance also changes the maximum current at resonant frequency. The maximum current of ideal resonant circuit is $23A$ which is attenuated to $0.3A$ with one parallel resistor in Figure 4.3(b) and further attenuated to $0.08A$ with two resistors in Figure 4.3(c). This

indicates another negative effect of parasitic to the resonant clock: it attenuates voltage swings at high frequencies. The signal attenuation can also be caused by a driver with high resistance.

To make sure the circuits work properly, the amplitude of each clock sink signal should be greater than $V_{dd} - V_{th}$ and less than V_{dd} as shown in Figure 4.1. Partial swing voltages at clock sinks can lead to logical failures. Therefore, buffers are needed to compensate. On the other hand, over-sized buffers will guarantee the swing with more power consumption which may offset the power savings gained by LC tanks. Therefore proper buffer sizing is another key procedure in resonant clock design.

4.2.3 Interaction Between Buffers and LC Tanks

In non-resonant clocks, buffers are the only components which drives the clock grid. In resonant clocks, buffers and LC tanks simultaneously drive the grid as a shared output. The energy charge and discharge of LC tanks follow the resonant theory while the buffers driving activity are determined by the arriving input clock signal. To avoid these two energy sources counteracting with each other, the resonant clock design must ensure that LC tanks and buffers are in phase. This is also similar to a non-resonant clock grid. Skewed grid buffers can result in either large skew, large short-circuit power, and ineffective buffer usage [30]. Hence, phase tuning is also an important step in resonant clock synthesis flow.

4.2.4 On-chip Inductor Overhead

To build a resonant clock with distributed LC tanks, on-chip inductors are a necessity. Most on-chip inductors use normal metal layers as in many radio frequency (RF)

designs. Figure 4.4 shows three spiral inductors on metal M_{n+1} connected to the CDN in metal M_n. Though the interconnect and device area below the shielded inductor can still be used, the on-chip inductors cost metal area should be considered as the major cost in the resonant clock synthesis. Usually, global routing layers with higher conductivity are often in high demand for global signals and they are better candidates to make inductors.

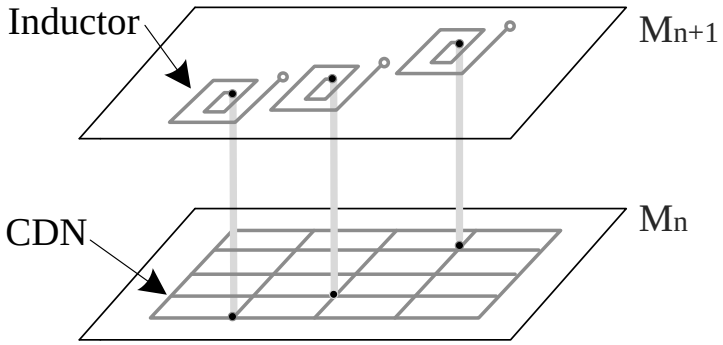


Figure 4.4: LC tanks and CDN are implemented in adjacent global metal layers.

Beside the inductor area overhead, another challenge in resonant clock design is the inductor design. Specialized EM inductor simulation should be used to model the inductors. As on-chip inductor design is another research area and there are many research works related to inductor models and design [47, 93, 99], it will not be discussed in this thesis. However, resonant clock synthesis should consider the inductor design challenges.

4.3 Resonant Clock Grid Synthesis Flow

4.3.1 Problem Formulation

The resonant clock grid synthesis problem is to minimize the clock skew and power with minimum inductor area overhead by LC tank placement, LC tank sizing and grid buffer sizing and optimization.

4.3.2 Synthesis Flow

Figure 4.5 is the flow of the proposed resonant clock network synthesis methodology. Contrary to rotary and standing wave resonant clocks, distributed LC resonant clocks have similar structure to previous non-resonant clock grids. Therefore, previous non-resonant clock grid methods [59, 81] are used to generate the initial basic non-resonant structure.

The LC tanks are placed and sized to convert the non-resonant clock grid into a resonant one. Previous LC tank resonant clock methods [7, 8] place uniform LC tanks in a symmetric H-tree structure. The final resonant clocks often cannot achieve the original design target [9, 21]. Clock grids are more complex structure than H-trees and without proper placing and sizing the LC tanks, the resonant power saving is not guaranteed. The detailed algorithms of LC placement and sizing is in Chapter 5.

Though LC tanks will save power, buffers are still needed in resonant clocks. This is because that the parasitic resistance resistance of wires and inductors will dissipate power as heat which introduces energy loss. Buffers are placed and sized only to compensate for this kind of power dissipation. The buffer insertion and sizing, however, is different from

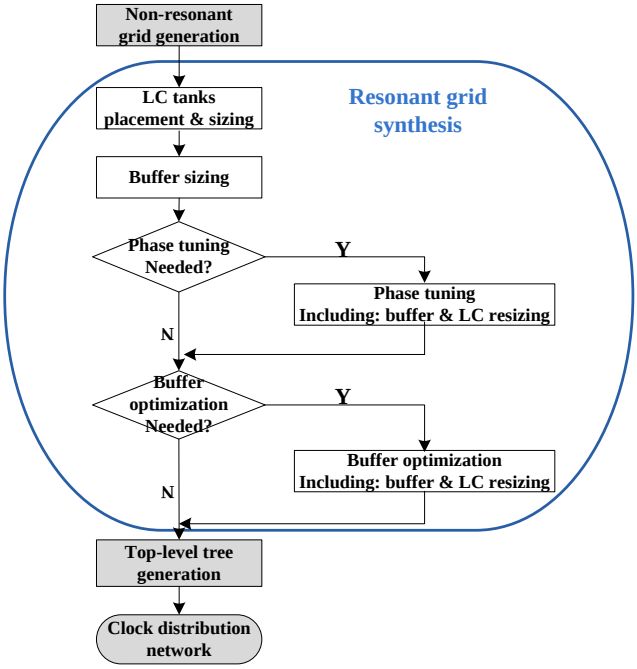


Figure 4.5: Overview of distributed LC resonant grid synthesis methodology.

non-resonant clock design because smaller buffers are needed in a resonant grid and the buffer driving ability cannot be easily estimated by logical effort due to interaction with the LC tanks. The buffer insertion and sizing in resonant clock grid design is a new topic which will be explained in Chapter 6. An incremental buffer optimization after LC tanks and buffers are placed and sized is also discussed at the end of Chapter 6. With properly sized LC tanks and buffers, the clock almost completely resonates with the LC tanks while buffers only compensate for the energy loss. The contribution of buffers to the clock network is much less than in a non-resonant clock. Considering this, there is a possibility to reduce the number of buffers to save area and power.

The two AC signals, A and B in Figure 4.1, have a phase shift of 90^o. In time domain, this phase shift appears as signal B will reach its maximum voltage $\frac{1}{4}$ clock cycle

after signal A reaches. If consider A and B as two clock sinks, this phase shift appears as *phase-induced skew* in the clock network. Therefore, in the resonant clock synthesis, the phase difference between all sinks should be considered because it is extra skew in the time domain. A method to analyze and tune the phase will is introduced in Chapter 7.

After the resonant clock is generated, a top level buffered clock tree is built to drive the resonant clock grid. The clock tree and resonant clock grid constitute a complete hybrid resonant clock distribution network.

Chapter 5

LC Tank Placement and Sizing

As discussed in 4.2, inductors, capacitors, and CDNs have parasitic resistances in practice. These resistances shield the capacitance seen by the LC tanks and change the expected resonant frequency. Resistance also causes signal attenuation and result in decreased voltage swings at clock sinks if unchecked. More importantly, the resistances can affect the energy efficiency. The resistances of the clock network are much larger and have more significant impact on performance than the parasitic resistance of the inductors. Unlike previous works [9, 21] which only consider the inductor parasitic resistance, this work considers the CDN resistances and buffer output resistances which are the major resistive components in the resonant grid.

The currents passing through the inductors into the capacitors can be very big at the resonant frequency. Distributing the LC tanks reduces the peak current passing through each individual inductor. Hence, it is impossible and impractical to resonate a clock grid with one LC tank with satisfactory voltage swing for a large chip. The LC tanks must be placed and sized on the grid to address these issues while limiting the area occupied by the

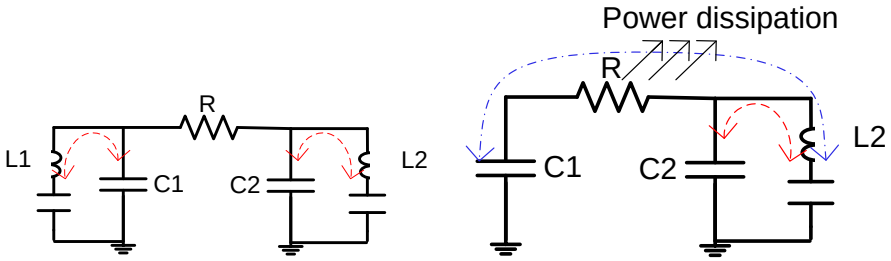
inductors to a reasonable value.

This chapter will first introduce an "optimal" resonant clock grid in terms of skew and power. Starting with the optimal grid, Chapter 5.2 discusses the placement LC tanks to maximize efficiency with reasonable inductor overhead. Chapter 5.3.2 discusses a method to place LC tanks with a pre-designed inductor library to reduce the design efforts of on-chip inductors.

5.1 Optimal Resonant Clock Grid

The starting point of our method is a uniform non-resonant clock grid buffered to obtain slew and skew constraints [59, 81]. The target resonant frequency f_0 is the clock frequency (1GHz in our case).

It is actually trivial to find the optimal solution of a resonant grid in terms of skew and energy loss. This can be done by inserting an LC tank at every node and sizing the inductor to make each LC tank resonate with the capacitor at its node. The energy is completely transferred from the voltage potential on the clock capacitance to the magnetic field of the inductor since the inductors are "fully distributed" the parasitics are small and efficiency is maximized. In Figure 5.1(a), a wire is modeled as two capacitors, $C1$ and $C2$, and one resistor R according to a simple π-interconnect model. An LC tank is attached to each node. As energy is only transferred between $C1$ and $L1$, $C2$ and $L2$, there is no current passing through resistor R. Hence, there is no energy loss in this circuit. The waveforms at $C1$ and $C2$ are also in phase which appears as zero skew. At this point, we have an optimal resonant grid and the grid exhibits ideal performance. However, the size of



(a) Power-optimal distributed LC tank placement (i.e. at every node)

(b) Energy loss through resistor by non-optimal placement

Figure 5.1: LC tank removal starts with optimal distributed placement and reduces inductor area with minimal power loss.

the inductors are extremely large since the small capacitances require very large inductors at a fixed operating frequency according to Equation 3.1. Though the ideal resonant clock grid shows zero skew and the best power efficiency, it is impractical to design such a grid by inserting an inductor to every node in the grid.

5.2 LC Tank Placement and Sizing with Continuous Inductors

The optimal resonant solution serves as a starting point and is reduced to a more practical solution while considering the total inductor area, skew, and voltage swing. Algorithm 1 is the pseudo-code for LC tank insertion and sizing to convert the optimal resonant grid (generated by Line 1 - 4) into a practical solution.

5.2.1 LC Tanks Removal

Instead of picking a better LC tank placement, we instead remove inefficient LC tanks. The following cost function is used to evaluate every inductor on Lines 6-8:

Algorithm 1 Inductor placement and sizing algorithm

Inputs: Grid nodes N; resonant frequency f_0; maximum inductor area A_{max}; skew constraint SK

Output: Inductors size and location L: a set of nodes with properly sized inductor inserted at each node.

1: **for** All $n \in N$ **do**

2: $L \Leftarrow (n)$

3: L_sizing(), solve $f = \frac{1}{2\pi\sqrt{LC_n}}$, C_n: the cap. seen at node n

4: **end for**

5: **while** $\sum Area(L) > A_{max}$ && $skew \leq SK$ **do**

6: **for** $l_i \in L$ **do**

7: $cost_i = \beta \times \frac{l_i}{L_{avg}} + (1-\beta) \times \frac{S\_LC_{avg}}{S\_LC_{min}}$

8: **end for**

9: sort($cost$)

10: remove_lc(10% of L with largest cost)

11: L_sizing()

12: **if** $\sum Area(L) < 1.2 \times A_{max}$ **then**

13: AC_buf_sizing()

14: **end if**

15: **end while**

$$cost_i = \beta \times \frac{l_i}{L_{avg}} + (1-\beta) \times \frac{S\_LC_{avg}}{S\_LC_{min}}, (0 \leq \beta \leq 1) \tag{5.1}$$

where l_i is the inductance of the i^{th} LC tank, L_{avg} is the average inductance, S_LC_{avg} is the average resistance from sinks to their nearest LC tank, S_LC_{min} is the minimum resistance from sinks to this LC tank, and β is the relative weight of each term. The first term of Equation 5.1 penalizes larger than average inductors. According to Equation 3.1, a small capacitance requires a large inductance to resonate at a fixed frequency, f_0. Large inductances, however, occupy more metal area and resonate very little capacitance. Therefore, big inductors are inefficient in terms of metal area usage per power savings. The second term of Equation 5.1 ensures that each sink has nearly the same resistance to an inductor. If a sink has a low resistance path to an LC tank, this sink will have phase offset (i.e. skew) relative to sinks with longer resistance path to LC tank. This is the phase-conflict induced skew introduced in Chapter 4.3.

Lines 9-10 remove the costliest 10% of the inductors. After removing inductors from the "optimal" resonant grid, the grid dissipates additional power. This is caused primarily by the current passing through parasitic resistances as shown in Figure 5.1(b). After removing an inductor $L1$, charging and discharging $C1$ will lose energy through R each clock cycle.

The two previous cost terms are weighted (β) depending on the benchmark profile. If a benchmark has relatively high sink capacitance density compared to the grid, the first term is more important. Vice versa, the second term is more important. In general, more weight should be placed on the resistance term since only a small number of inductors fit on the chip.

After removing the LC tanks, the capacitance that resonates with each removed inductor is re-allocated to nearby inductors according to Algorithm 2 which re-sizes the remaining inductors (Line 11). In Chapter 5.2 of Algorithm 1, Line 13 performs our AC-based buffer sizing which is detailed in Chapter 6. By resizing the buffers, we obtain a more accurate buffer output capacitance estimation and hence more accurate inductor sizes. To save run time, we only size the buffers during the final iterations (i.e. when total inductor area is less than $1.2 \times A_{max}$). We repeat this procedure (Lines 5-15) until the total inductor area is less than a user-specified maximum inductor area A_{max} or, alternatively, until a maximum skew or power limit is reached.

5.2.2 LC Tank Sizing

Algorithm 2 describes the methodology L_sizing() to calculate the total capacitance assigned to each inductor and the size of each inductor. The capacitance at each grid node is the sum of half the wire capacitance, the sink capacitance and the buffer capacitance which are connected to the node (Line 1). For each node n in the grid, we find the lowest resistance path from n to all LC tanks (Line 4). The capacitance of a node n, C_n, will primarily resonate with the inductor l_i which is resistively closest to it. The total capacitance resonating with inductor i is updated by adding C_n (Line 5). We then calculate the inductance of l_i on Line 7 using the capacitance in the resistively close region, CR_i.

Algorithm 2 Inductor sizing, L_sizing()

Input: Inductors L; grid nodes N; resonant frequency f_0

Output: Sizes for each inductor in L

1: $C_n \Leftarrow \sum (\sum wire\_cap, buf\_cap, sink\_cap)$ connected to n, $n \in N$

2: $CR = 0$ *//Cap. resonates with each inductor*

3: **for** Each $n \in N$ **do**

4: Find inductor l_i which is resistively closest to n, $l_i \in L$

5: $CR_i += C_n$ *//node n resonates with inductor l_i*

6: **end for**

7: $l_i = \frac{1}{{\omega_0}^2 \times CR_i}$, $l_i \in L$, $\omega_0 = 2\pi f_0$

5.3 LC Tanks Placement with Discrete Inductors

The previous inductor sizing method requires continuous inductor sizes at the end of Algorithm 1. Continuously sized on-chip inductors are not practical with current inductor design techniques. The design of high-Q inductors requires significant manual design and electromagnetic characterization. The rounding to discrete inductor sizes, however, requires either significant overhead for accurate matches or results in high skew due to coarse matches. This section discusses the synthesis of distributed resonant clock grids using a pre-made library of inductors.

The LC tank placement problem can be re-formulated as follows: Given an initial non-resonant clock grid, an inductor library $\mathcal{L}$, and a maximum total inductor area A, insert LC tanks and insert/size buffers to minimize the power and skew of the resonant clock grid. We formulate this LC tank placement problem as a set covering problem of all grid nodes N. Candidate LC tanks are placed at various grid nodes which cover subsets of the nodes.

The objective is to find the optimal subset T to cover all nodes N with minimum inductor area cost A. Each candidate w represents a set of nearby grid nodes which resonate with an inductor $l_w \in \mathcal{L}$ placed at grid node $p_w \in N$. The set of all candidates is W and $T \subset W$ is the optimal subset. The cost of the subset T is the total inductor area A required to complete the covering. Some common terms are defined to facilitate the algorithm discussion in Table 5.1.

5.3.1 Resistance (R) and Capacitance Mismatch (ΔC) Constraints

The parasitic resistance in the clock grid determines the efficiency of the resonant clock, because power is dissipated as heat when current passes through a resistance. A highly resistive path between a node n and an LC tank (i.e. a large r) means the charging and discharging currents between node n and the LC tank will pass through large resistances and consume more power. In addition, the resistance affects the phase offset and can create skew. We address these issues by constraining each node capacitance to be less than a maximum resistance R away from a placed inductor in order to properly resonate as an LC tank.

The mismatch between the total clock grid capacitance C_{clk} and the sum of all capacitances by resonated inductors C_T is ΔC. Ideally, ΔC equals zero which means that the clock capacitance is exactly covered by the LC tanks. In other words, each grid node resonates with only one LC tank within resistance R. However, since inductors can only be chosen from the library and they are discrete values, it is difficult to guarantee that ΔC equals zero. ΔC is always non-negative because we require a full coverage of all grid nodes N by LC tanks. A positive ΔC, however, shifts the resonant frequency because

some nodes resonate with more than one inductor which can increase power and skew. A large ΔC also requires a significant amount of dummy capacitance to balance phase in Chapter 7, so our algorithm minimizes ΔC by setting an upper bound C_{MAX} during the LC placement algorithm and gradually loosening the constraint until a solution is found.

Table 5.1: Common terms in algorithm

N	the set of all grid nodes (capacitances).
$\mathcal{L}$	inductor library.
r	resistance from an LC tank to a node.
R	resistance limit from an LC tank to a node. When $r > R$, the LC tank is assumed to not resonate with this node capacitance.
R_{MAX}	maximum permissible value of R.
W	set of candidate subsets of grid nodes N. Each element w in W is a subset of grid nodes N. All nodes in w are within resistance R of a sized LC tank at a particular grid node.
l_w	the inductor size of candidate subset w, $l_w \in \mathcal{L}$.
p_w	the inductor position of candidate subset w, $p_w \in N$.
T	subset of W, ($T \subset W$), that will cover all grid nodes N. Each grid node n will at least resonate with one inductor in T or equivalently $\bigcup_{T_i \in T} T_i = N$.
A_l	area of inductor l.
A	total inductor area used to cover T.
C_{clk}	total grid capacitance, including sink capacitance and wire capacitance.
C_l	ideal capacitance for an inductor l to resonate at the target frequency, $C_l = \frac{1}{(2\pi f)^2 l}$
C_T	total capacitance of subset cover T. $C_T = \sum_{t \in T} C_{l_t}$
ΔC	difference between clock capacitance C_{clk} and total cover capacitance C_T. $\Delta C = C_T - C_{clk}$
C_{MAX}	maximum permissible value of ΔC.

5.3.2 Discrete LC Tank Placement

Algorithm 3 is the pseudo-code of the LC tank placement method using the prior R and ΔC constraints. Initially, we set very restrictive bounds on both R and ΔC. After generating all candidate subsets W (Line 2) according to the R constraint, the LC tank placement is solved as an Integer Linear Programming (ILP) covering problem (Line 3) with a ΔC constraint. If a feasible solution is found with these tight constraints, the resonant clock grid will have a good performance in both power and skew. Normally, however, the ILP cannot find a feasible solution with the initial constraints so we iteratively loosen the R and ΔC constraints and re-run (Line 2 - 3) until the ILP successfully returns a feasible solution.

A large ΔC mismatch will induce extra skew, but we can compensate for this by adding dummy capacitance which is expensive and not preferable as described later in Chapter 7. Therefore, we instead first increase the R limit which only decreases efficiency but does not add much skew. However, we do not allow the R limit to increase beyond R_{MAX} (Line 5), because a very large resistance means that an LC tank can not resonate efficiently with a node and may not save significant power as discussed in Section 5.3.1. We try to keep the resistance limit small, but increase it first to find a feasible solution. If a feasible solution is not found with $R = R_{MAX}$, we must start sacrificing skew by increasing C_{MAX} to find a feasible solution.

5.3.3 Candidate Subset Generation

We place each inductor in the library $\mathcal{L}$ at every potential grid node in N to generate the candidate subsets W. When an inductor $l_w \in \mathcal{L}$ is inserted at a grid node

Algorithm 3 ILP set covering LC tank placement

Inputs: Inductor library $\mathcal{L}$, initial non-resonant clock grid

Output: Subset T cover all grid nodes N. Each node in T is connected to an inductor in library $\mathcal{L}$.

1: Initialize R and C_{MAX} with small values
2: Generate candidate subsets for each $\mathcal{L}$ with $r < R$ constraint
3: Solve ILP set covering problem with ΔC constraint
4: **if** ILP is infeasible with R and ΔC constraints **then**
5: **if** R is less than a threshold value R_{MAX} **then**
6: Loosen R constraint (increase R)
7: **else**
8: Loosen ΔC constraint (increase C_{MAX})
9: Reset R constraint (assign a small value to R)
10: **end if**
11: Go to step 2
12: **end if**

$p_w \in N$, it is assumed to resonate with the set of nodes within resistance R. We add the closest (i.e. least resistance away) nodes until the total capacitance is greater than the ideal capacitance C_l for the inductor at the target resonant frequency. The resulting set of grid nodes is the candidate subset $w \subset N$ that is covered by inductor l_w at node p_w. Since the number of library inductors $|\mathcal{L}|$ is typically small (3 in our case), the number of candidate subsets $|W|$ is not very large and does not add significant complexity to the ILP.

5.3.4 Candidate Subset Selection

We solve the set covering problem as $ILP(W, C_{clk}, C_{MAX}, A)$ which is

$$Min : \sum_{t \in T} A_t \times x_t, (t \in T, T \subset W) \tag{5.2}$$

subject to:

$$x_t \in \{0, 1\}, t \in W \tag{5.3}$$

$$\sum_{t \in T_n} x_t \geq 1, T_n = \{u | n \in u, u \in T\}, \forall n \in N \tag{5.4}$$

$$\sum_{t \in T} A_{l_t} \times x_t \leq A \tag{5.5}$$

$$0 \leq \Delta C \leq C_{MAX} \tag{5.6}$$

$$\Delta C = \sum_{t \in T} (C_t \times x_t) - C_{clk} \tag{5.7}$$

The inputs to the ILP are the candidate subsets of nodes (W) covered by sized inductors at grid nodes, the total grid capacitance (C_{clk}), the maximum allowed capacitance difference (C_{MAX}) between C_{clk} and total capacitance resonating with all inductors C_T, and the maximum total inductor area (A). The objective is to find a set T which is the subset of W that satisfies all the constraints (Equation 5.3 -5.7) and minimizes the total inductor area A.

In Equation 5.3, x_t is a decision variable (0 or 1) that decides whether inductor t

is selected to cover the subset or not. Equation 5.4 requires that every grid node n resonates with at least one inductor. T_n is a set of all subsets of T which contain node n. Equation 5.5 is the total inductor area constraint where l_t is the library inductor of subset t and A_{l_t} is its physical area. Equation 5.6- 5.7 are the capacitance mismatch constraints that ensure the capacitance C_t resonated by each x_t is sufficient to resonate the total clock grid capacitance C_{clk} by limiting ΔC mismatch. If a subset of T can be found to satisfy all these constraints, the LC tanks of specified size are placed in the grid.

Chapter 6

Resonant Clock Buffer Optimization

Resonant clock grids are "driven" by both LC tanks and buffers. A well designed resonant grid with good power savings and small skew should primarily be driven by the LC tanks which will reuse the energy. In other words, buffers are only needed to compensate for the inevitable power consumed by the paracistic resistances in suboptimal resonant clocks. Therefore, resonant clock grid buffers are much smaller than non-resonant buffers.

In this chapter, previous works on non-resonant buffering are introduced. Then the resonant clock grid buffering problem is discussed and two sizing methods are presented. The first is a novel AC-based buffer sizing method and the second is an AC- and resistance-based buffer sizing extension.

6.1 Non-resonant Clock Grid Buffering

In previous non-resonant clock grid design, most buffering methods [59, 81] use a greedy set covering heuristic. The driving ability of a buffer is estimated by its size and is assumed to drive a maximum capacitance. A buffer at a grid node covers the clock grid

capacitance in a region. The buffer placement and sizing problem is to choose a set of buffers with minimum cost (buffer area) to cover the whole grid. In another approach, an iterative removal buffer placement algorithm [30] considers skew instead of capacitance in by measuring SPICE transient analysis or faster Elmore delay.

The previous buffering methods based on the buffer driving load estimation are not applicable to resonant clock design because buffers in resonant clocks only compensate for the power loss instead of fully driving the clock network. Therefore, a novel small signal AC-based buffer sizing method is proposed. The AC analysis includes all the components in the resonant grid (sinks, buffers, wires and LC tanks) including their parasitic resistances. After running AC analysis, the amplitude and phase of the sinks voltages guide the buffer sizing.

6.2 AC-based Buffer Sizing

The original non-resonant clock grid is generated using previous clock synthesis methods. Usually, each buffer has a specific buffer size. The AC-based buffer sizing methodology finds the minimum buffer sizes to drive the resonant grid with the same buffer positions as the initial non-resonant grid. For simplicity, we set all the buffers to the same size. Since the buffers are very small compared to a non-resonant grid, the relative sizing is actually not very important. Algorithm 4 outlines the AC-based buffer sizing method.

The core function AC_analysis() runs an AC small signal analysis of the clock grid as introduced in Chapter 4.1.2. A matrix using nodal analysis at the target frequency as shown by Equation 4.1 is built and the complex voltage of each node is obtained by solving

Algorithm 4 AC-based buffer sizing, AC_buf_sizing()

Input: clock grid with LC tanks; buffers B;

minimum/maximum buffer size b_{min}/b_{max}

Output: Buffer sizes, ensure that $\forall v_s \geq V_{swing}$, $s \in S$

1: $b_{cur} = (b_{min} + b_{max})/2$

2: set_buf_size($b = b_{cur}, b \in B$)

3: L_sizing()

4: **while** $(b_{max} - b_{min}) > \delta$ (δ is a small number) **do**

5: AC_analysis()

6: **if** $v_{min} = MIN(v_s, s \in S) < V_{swing}$ **then**

7: $b_{min} = b_{cur}$

8: **else**

9: $b_{max} = b_{cur}$

10: **end if**

11: $b_{cur} = (b_{min} + b_{max})/2$

12: set_buf_size($b = b_{cur}, b \in B$)

13: L_sizing()

14: **end while**

the complex linear system. Algorithm 4 then uses a bisection method to find the minimum buffer sizes to guarantee the full voltage swing at the sink nodes.

The initial buffer sizes are set to the average value of b_{max} and b_{min} which are the allowed maximum and minimum buffer sizes (Lines 1-2). After running AC_analysis(), the buffer sizes are chosen (Lines 6-12) by comparing the minimum voltage swing of all sink nodes v_{min} with the required V_{swing} (Line 6). After altering the buffer sizes, the buffer output capacitance and output resistance change so that inductor sizes are updated accordingly (Line 3 and 13). Re-sizing the inductors enables the grid to resonate at the designated frequency which will minimize the phase-conflict induced skew.

6.3 AC- and Resistance-based Buffer Sizing

The AC-based method in previous section quickly finds the minimum buffer size needed to drive the resonant grid. The final buffer sizes are all the same which is convenient to implement but may not minimize power. The uniform buffer sizing assumes all buffers contribute the same effort to the grid. However, considering the non-uniform distribution of clock sinks, the parasitic resistance seen by each buffer is different which means each buffer's compensation for heat loss is also different.

Algorithm 5 is an improved buffer sizing algorithm that considers the resistance from each buffer to the sinks. We initially set all buffers to the minimum size on Line 1 and perform an AC analysis to update the voltage amplitude/phase at all sinks. S_u on Line 3 is the subset of sinks that do not achieve full voltage swing. For each sink s in S_u, we find the buffers which are within resistance (R) of s that may influence the voltage swing and

Algorithm 5 AC- and Resistance-based buffer sizing

Inputs: Resonant grid with LC tanks and buffers

Output: Buffer sizes

1: Set all buffers to minimum buffer size

2: AC_analysis()

3: Find partial-swing sinks set S_u

4: **while** $|S_u| > 0$ **do**

5: Find all buffers RB that drive S_u

6: Increase each buffer size in RB

7: AC_analysis()

8: Find partial-swing sinks set S_u

9: **end while**

define these as the regional buffer set (RB) on Line 5. Note that R is the same parameter used in the tank placement of Algorithm 3 in Chapter 5.3.2. Increasing the size of each buffer in RB improves the voltage swing of the violating sinks, S_u. Before starting the next iteration, we update the sink voltage amplitude/phase using AC analysis and repopulate the set S_u (Lines 7 - 8). This buffer sizing procedure is repeated until all sinks have full swing. Even if a sink is far away from LC tanks, it will be mainly driven by the RB buffers while the clock grid is mainly driven by LC tanks. So after several iterations, Algorithm 5 is able to find a solution.

6.4 Resonant Grid Buffer Placement Optimization

A top-level tree usually drives the buffers of a clock grid. Since the buffer sizes in a resonant grid are much smaller than a non-resonant grid, the top-level tree power is reduced when driving a resonant grid. The initial grid buffer positions in Algorithm 4 are the same as a normal non-resonant grid [59, 81] except that the buffers are re-sized. The resonant grid buffers can be re-distributed to provide a reduction in the number of grid buffers which further reduces power because of shortened wire routes and fewer grid buffers (sinks) in the top-level tree.

An improved grid buffer methodology uses an iterative removal approach [30]. Instead of deciding where to insert grid buffers, we perform buffer sizing and remove the smallest, least useful buffers. Several significant differences between this approach and the prior one [30] are that we start with the initial buffer locations instead of buffering all nodes and we perform an AC- and resistance-based sizing rather than Linear Programming (LP) sizing.

Algorithm 6 is the overall grid buffer reduction algorithm in resonant clock synthesis. The input of Algorithm 6 is original buffer set of sizes and locations B and the goal is to reduce the number of buffers to a certain number, it is half of the original buffer number in our implementation, while ensuring that all sinks have full voltage swing. In practice, a reduction of half the buffers obtained competitive results in most situations, but this may depend on the specific clock distribution.

In order to find which buffers are more important, AC- and resistance-based buffer sizing, AC_res_buf_sizing() (Algorithm 5), sizes buffers such that all sinks fully swing.

Algorithm 6 Resonant grid buffer reduction algorithm

Input: Original buffer sizes and locations B

Output: Optimized buffer sizes and locations B'

1: $B' = B$

2: AC_res_buf_sizing(B')

3: **while** Buffer number in B' ¿ maximum allowed buffer number **do**

4: $B_{rm} \Leftarrow 10\%$ of minimum size buffers in B'

5: $B' = B' - B_{rm}$

6: AC_res_buf_sizing(B')

7: L_sizing()

8: **end while**

After buffer sizing (Line 2), smallest 10% of the buffers which have the least influence on the sink voltage swings are removed (Lines 4-5). With fewer buffers driving the grid, the remaining buffers are re-sized (Line 6) and the LC tank inductors are re-sized (Line 7). The sizing and removal procedure is repeated (Lines 3-8) or it can also be terminate at a given skew bound if skew, rather than power, is the primary objective.

Contrary to Algorithm 4, we size each buffer separately in AC_res_buf_sizing() to decide which buffers are more important. By removing smaller buffers and re-distributing the remaining buffers to better drive the resonant grid, more power can be save without significant performance degradation.

Chapter 7

Resonant Clock Phase Tuning

After inserting LC tanks and buffers, the resonant grid achieves full voltage swing, but there may be phase conflicts among LC tanks and the sized buffers. The phase conflict will appear as skew, but, more importantly, it means that extraneous power is consumed when buffers and LC tanks conflict while driving the grid. The phase conflicts are caused by both the resistance of the CDN and the remaining ΔC after the LC tank placement with discrete inductor library in Chapter 5.3.

Though these two effects are minimized, the sink phases do not match perfectly at the resonant frequency and can be fine tuned. If a capacitance resonates with an inductor at resonant frequency, the phase of this sink will ideally be zero. Some sinks will have a phase less than zero which means that these sinks see a larger inductor than required for their capacitance. Other sinks will see a greater than zero phase due to long resistive paths to the nearest LC tank which appears as a smaller inductor than required due to resistive shielding. While we cannot fix the phase less than zero, we can reduce the positive phase differences by adding extra dummy capacitance.

7.1 Dummy Capacitance for Phase Tuning

Algorithm 7 Resonant clock phase tuning

Input: Resonant grid with buffers and LC tanks

Output: Phase tuning capacitances

1: Assign each node in N to the nearest LC tank, l

2: Calculate the capacitance seen by each LC tank, C_g

3: Add extra capacitance to the LC tank if $C_g < C_l$

4: AC- and Resistance-based buffer sizing (Algorithm 5)

5: **repeat**

6: Add extra cap. to stub nodes of sinks with maximum phase

7: AC- and Resistance-based buffer sizing (Algorithm 5)

8: AC_analysis()

9: **until** No improvement in phase conflict and buffer area

Algorithm 7 minimizes the negative phase in resonant grids by adding dummy capacitances. Lines 1-4 reduce the phase conflict induced by ΔC resulting from the discrete LC tank placement. For simplicity, we assume a node only resonates with the closest LC tank (Line 1). After calculating the grid capacitance C_g of each LC tank (Line 2), the difference between C_g and C_l and the ideal capacitance value C_l is added. The extra $C_l - C_g$ of dummy capacitance reduces the phase conflict and improves resonant performance. After this, buffer resizing (Line 4) reduces the total buffer size required since less compensation by the buffers is required due to the improved efficiency.

While Lines 1-4 reduce the ΔC of each LC tanks, there is still potentially phase

difference to individual sinks. In Lines 5-9, the phase of individual sink nodes are adjusted by adding small extra dummy capacitance. After each capacitor is added, the buffers are resized and AC analysis is run (Lines 7-8). This procedure is repeated until no improvement is seen in total buffer area and phase conflict.

Usually, in non-resonant CDNs, the switching power can be quickly estimated by the total capacitance of the CDN. For more accurate power measurement, SPICE transient simulation is needed. In resonant clock, the power can not be estimated by the capaciatance any more since LC tanks are the major source of energy supply. SPICE transient simulation often is very timing consuming especially when the circuit is large. So the buffer area is used as a quick estimate of the power of the grid during because buffers are used to compensate for the power loss from phase conflicts and parasitics. The phase tuning procedure based on the AC analysis is very fast.

By inserting dummy capacitance to tune the phase conflict, the skew is improved and we also see a reduction in the buffer area because of the less conflicts among buffers and LC tanks. For the library-based resonant clock design, phase tuning is a necessary procedure.

Chapter 8

Experimental Methodology

8.1 Technology Setup

The resonant clock grid synthesis methods proposed in this thesis were implemented in C++ and use the 45nm technology data from 2009/2010 ISPD Clock Synthesis contest for all experiments [70]. Lower frequencies are the most challenging for resonant clocks since they require larger inductors than high frequencies as shown in Equation 3.1. So the target resonant frequency is set to 1GHz and V_{dd} is $1V$ as these are typical values for high-performance ASIC designs.

SuperLU [20] solves the matrix equations in our AC-based buffer sizing. HSPICE accurately measures the power and skew of the CDN. The power measurements include all buffers, wires, inductors, and sinks.

On-chip inductor Q-factors can be as high as 20 in the 1-5GHz frequency range [15]. A very poor quality $3nH$ inductor with Q=3 will have an intrinsic resistance of 6.28Ω at 1GHz which is orders of magnitude smaller than typical CDN buffer and wire resistances.

Previous work [9,21] considered inductor parasitic resistance, but did not consider this more prominent CDN parasitic resistance. The final resonant frequency by Chan et al [9] is 1GHz which is much lower than the target of 3.7GHz. Similarly, the final resonant frequency by Drake et al [21] is 140MHz which is much lower than the design target of 2GHz. These previous comparisons demonstrate that the resistance in the CDN circuit itself has a large impact on the resonant frequency. Only CDN parasitic resistance is included in our models, but we later analyze the resistance of inductors in Chapter 9.3.6.

The inductor parameters we used are: $w = 15um$, $t = 10um$, $s = 15um$ and $d_i = 100um$ as described in Section 3.4 and [15]. We use these parameters and Equation 3.3-3.4 to calculate the inductor area overhead. The inductor library $\mathcal{L}$ contains three inductors: $3nH$, $5nH$ and $8nH$ with areas of $0.16mm^2$, $0.21mm^2$, and $0.30mm^2$, respectively. The on-chip inductors have parasitic resistance that effects the performance of the resonant clock. The quality factor $Q = \frac{\omega L}{R}$ of the inductors is 10 which is reasonable for modern on-chip inductors [15] in this frequency range.

For AC analysis and optimization, each buffer is modeled as a current source with a resistance and input/output capacitance. BSIM level 14 models and transient simulation are used for all final results. Wires are modeled by simple π-interconnect model in both AC analysis and HSPICE simulation. The wire length in the CDN is relatively short compared with the wavelength and frequency, so transmission line models are not necessary [16]. In the AC analysis procedure, we can add the inductance of clock grid wires with a small run time increase.

8.2 Experiment Design

8.2.1 Non-resonant Clock Grid Baseline

To validate the power saving achieved by resonant clock, we compare with baseline non-resonant clock grids. We implemented a methodology similar to Meshworks [59] to generate the initial buffered grid. The top-level tree driving the clock grid is routed with minimum wire length using DME [76], equal levels of buffers are inserted with skew and slew constraints [72], and buffers are greedily inserted in a bottom-up manner to minimize skew. The buffered tree along with the clock grid compose the entire CDN.

A common design practice is to allocate $5 - 10\%$ of the clock period for clock skew. The initial CDN is generated with a $50ps$ skew bound which is reasonable for a 1GHz ASIC. Our method can target any clock frequency and, in fact, has greater power savings at higher frequencies.

8.2.2 Resonant Clock Grids

Three groups of experiments validate the resonant clock synthesis methods proposed in this thesis.

The first group experiment validates the continuous inductor sizing and placement methodology. Though it is difficult to design continuous inductors, this methodology is the most efficient way to use inductors with least extra cost – no dummy capacitance is needed in phase tuning. In a partial custom methodology, the effort to design the inductors may be acceptable to obtain the best performance. Usually, this method results in a small number of inductors, with advanced on-chip inductor design tools, this continuous inductor method

could be practical.

The second experiment validates the resonant clock synthesis with pre-made inductor library considering the practical difficulties in current inductor design. We assume there are three pre-designed inductors in the library and only the inductors in the library can be used. Since dummy capacitance is needed because of the capacitance mismatch, at the end of this synthesis flow, phase tuning is needed. Compared with previous method, this method requires less inductor design effort but need extra capacitance cost.

The third experiment is to validate the incremental buffer optimization to reduce resonant grid buffers. While LC tanks are the major drivers of the resonant clock grid, the contribution of buffers to the clock network is reduced. With the incremental buffer optimization, buffers are better distributed and more properly sized.

8.2.3 Comparison with State-of-the-Art Resonant Clocks

All the previous distributed resonant clock research is for small, often hand-tuned, clock sectors driven by H-trees [7–9,63,65] so a direct comparison on the large benchmarks is not feasible. However, we do compare with a uniform placement of LC tanks at a fixed pitch that is very similar to the approach used by IBM [6].

8.2.4 Comparison with ISPD Clock Synthesis Contest

Beside comparing the resonant clock with non-resonant clock grid to show the power saving with inductors. The proposed methods are also compared to the state-of-the-art non-resonant clock synthesis methodologies which include some CDN with tree structures. In recently published work [88], four different non-resonant clock synthesis

methods which were the best results in ISPD clock synthesis contest are compared. The CDN in [88] is a tree driving grid structure targeting to minimize the local skew. Contango [41] and NTUclock [69] are based on tree structure. Another compared method is CNSRouter [69] which also generates tree-grid CDN.

We summarize the synthesis results of all these four methods and compare them with the resonant clock results. Resonant clock is promising in power saving with better robustness.

Chapter 9

Experimental Results

9.1 Inductor Area

We first want to investigate the practicality of on-chip distributed inductors for high-performance resonant clock grids. The resonant clock grid performance and power changes according to the metal area limit inductors in Chapter 5.1. Figure 9.1 shows the resonant grid power, skew and buffer area with different inductor area limits. The power, skew and buffer area are normalized to the original non-resonant clock grid and inductor area is the percentage of the top-level metal layer area. As shown in Figure 9.1, the power and skew improve with more inductor area because distributed inductors results in a better resonance among LC tanks and clock capacitances. When the inductor area is equal to one metal layer area (one entire metal layer for on-chip inductors), the power saving is about 90% and skew is also reduced. With only 20% metal area, the power savings is about 60%. It is important to note that this is not wasted area, however, since the interconnect and device area below the shielded inductor can still be used. In this experiment, the spacing

as well as mutual inductance among inductors are not considered. To accurately simulate the interactive behavior among inductors, special simulation tools are needed which is not included in this thesis.

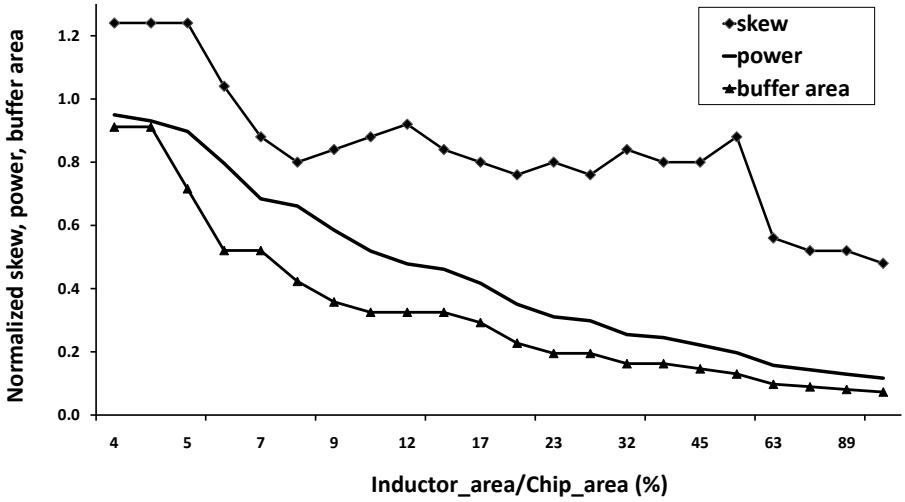


Figure 9.1: The relationship between inductor area and clock grid performance, power saving and skew improve with more on-chip inductors.

9.2 Resonant Clock Grid with Continuous Inductors

9.2.1 Synthesis Flow

Figure 9.2 is the synthesis flow of this experiment. This set of experiments validate the continuous LC tank placement and sizing method (Algorithm 1). The AC-based buffer sizing (Algorithm 4) finds the minimum buffer size needed in resonant grid. Because inductors in these experiments are sized exactly according to the capacitance, phase tuning is not necessary.

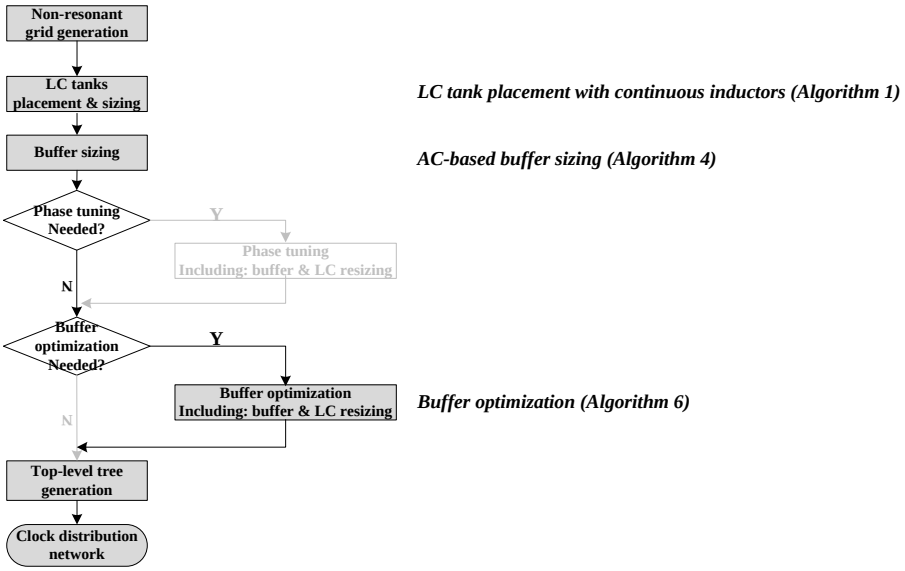


Figure 9.2: Resonant clock grid synthesis flow with continuous inductors. Uses Algorithm 1 for LC tank placement and Algorithm 4 for buffer sizing, phase tuning is not required in this flow.

Table 9.1: ISPD2009 benchmark simulation results. With 28% one metal layer for on-chip inductors, resonant grid without buffer reduction reduce the average total power by 39.5% and improve the skew by 3.5ps. With buffer reduction (Algorithm 6), power saving is 60.9% while skew slightly increase by 4.8ps.

Benchmark				Tree+Non-resonant			Tree+Resonant								
							AC-sizing				AC-sizing & buffer reduction				
Name	Sink #	Cap pF	CA mm^2	Pwr mW	Skew ps	BA um^2	Pwr mW	Skew ps	BA um^2	LA/CA %	Pwr mW	Skew ps	BA um^2	LA/CA %	LC# #
s1r1	81	2.8	69.4	189.4	7	397.8	87.2	7	152.5	26.7	71.1	20	93.3	26.7	54
s2r1	88	3.0	54.6	156.5	12	409.2	69.6	12	118.0	27.7	58.0	25	93.3	27.6	43
s3r1	131	4.6	165.6	360.0	33	628.0	159.0	14	312.3	29.3	112.4	27	182.6	29.2	114
s4r3	623	10.9	120.7	421.1	32	1966.0	183.5	28	368.9	30.3	155.2	40	261.4	25.8	92
f11	121	4.2	109.2	295.2	23	601.8	138.0	16	270.5	25.7	116.7	24	208.0	25.9	83
f12	117	4.1	91.2	250.0	20	566.7	116.7	17	215.7	30.5	97.1	25	161.6	25.6	69
f21	117	3.6	133.3	245.6	28	576.7	146.9	16	236.8	29.7	125.0	22	198.3	29.9	110
f22	91	3.4	50.4	151.1	13	425.7	69.7	18	127.3	28.3	56.7	21	93.1	28.5	41
f31	273	9.6	275.6	459.0	27	1933.7	403.6	25	1087.9	22.0	186.3	32	360.4	31.0	174
f32	190	6.7	269.0	417.7	29	1186.7	422.8	31	1253.6	29.3	182.7	30	302.2	28.0	180
Avg.	183	5.3	133.9	297.1	21.9	888.0	179.7	18.4	414.4	28.0	116.1	26.7	196.8	27.8	96
Average improvement							39.5%	3.5	53.3%	-	60.9%	-4.8	77.8%	-	-

Sink: sink number Cap: sinks capacitance CA: chip area Pwr: HSPICE power
BA: buffer area LA/CA: on-chip inductors metal area normalized to the area of one mental layer

Table 9.2: ISPD2010 benchmark simulation results. With 28% one metal layer for on-chip inductors, resonant grid without buffer reduction reduce the average total power by 48% and increase the skew by 7.6ps. With buffer reduction (Algorithm 6), power saving is 61% while skew increase by 27.3ps.

Benchmark				Tree+Non-resonant			Tree+Resonant								
							AC-sizing				AC-sizing & buffer reduction				
Name	Sink #	Cap pF	CA mm^2	Pwr mW	Skew ps	BA um^2	Pwr mW	Skew ps	BA um^2	LA/CA %	Pwr mW	Skew ps	BA um^2	LA/CA %	LC# #
01	1107	18.9	64.0	327.1	8	2329.0	231.4	19	645.1	27.5	179.3	77	308.3	27.5	15
02	2249	39.2	91.0	503.2	15	4397.5	160.4	23	361.2	34.0	98.1	24	270.2	33.8	89
03	1200	18.1	1.4	66.4	9	559.7	49.1	23	124.6	29.3	42.1	37	76.0	29.3	3
04	1845	12.3	5.7	94.2	17	858.3	62.6	11	162.5	13.8	55.6	41	86.3	13.8	3
05	1016	5.2	5.8	62.0	11	549.4	35.3	10	66.5	28.9	30.8	34	61.1	28.9	5
06	981	12.6	1.5	49.7	16	401.1	28.0	37	47.6	32.8	24.4	43	36.2	32.8	3
07	1915	18.1	3.5	77.8	27	602.5	50.3	28	123.9	28.5	34.9	46	55.9	24.5	5
08	1134	13.0	2.6	58.7	21	494.4	27.4	34	56.7	32.6	18.4	40	32.1	29.3	4
Avg.	1431	17.2	21.9	154.9	16	1274.0	80.5	23.1	198.5	28.4	60.5	42.8	115.7	27.5	16
Average improvement							48.0%	−7.6	84.4%	-	61.0%	−27.3	90.9%	-	-

Sink: sink number Cap: sinks capacitance CA: chip area Pwr: HSPICE power

BA: buffer area LA/CA: on-chip inductors metal area normalized to the area of one mental layer

The methodology is run on both the 2009 and 2010 ISPD clock benchmarks. The results are separated because the ISPD 2009 benchmarks only include the global clock distribution whereas the ISPD 2010 benchmarks include all routing down to local clock buffers. This means there is significant difference in the number of sinks and the total capacitance.

In this group of experiments, we compare three different CDN methodologies. The baseline methodology is a buffered clock tree driving a non-resonant clock grid similar to Meshworks [59]. This is labeled as "Tree+Non-resonant" in Table 9.1 and 9.2. The second is a buffered tree driving a resonant clock grid which has the same grid buffer locations as the first non-resonant grid but sized according to Algorithm 4 (AC-based buffer sizing). This is labeled as "AC-sizing". Because the top-level tree effects the resonant clock grid, we have a direct comparison with the non-resonant clock grid by keeping the same buffer locations as the non-resonant clock grid. The final approach, labeled as "AC-sizing & buffer reduction", includes the buffer reduction in Algorithm 6 to save additional power. We limit the maximum grid skew (SK) to $50ps$ and the metal area occupied by inductors (A_{max}) to be $< 30\%$ of the top metal layer for each benchmark.

9.2.2 Power

In Table 9.1, the resonant grid without buffer reduction reduces the average total power by 39.5%. The average total power is reduced further to 60.9% after the grid buffer reduction step. The average total inductor area is 27.8% of the top metal layer. In Table 9.2, the average power is reduced by 48% without buffer reduction and 61% with buffer reduction.

The total dynamic power (grid power and tree power) is shown in Figure 9.3 to analyze the power savings achieved by reducing the grid buffers. Figure 9.3 summarizes several benchmarks using each of the three approaches: "tree+Non-resonant", "AC-sizing" and "AC-sizing & buffer reduction." All power values are normalized to the "tree+Non-resonant" methodology. The dark bar represents the clock grid power which includes the grid buffer power used to drive the sinks and grid wires. The light bar is the power of top-level tree, which includes the tree wires, tree buffers, and grid buffer input pin capacitance. As previously seen, the total dynamic power is reduced by 39.5-61.0%. In addition, the grid buffer reduction procedure reduces the top-level tree power by at least 10%. The grid buffer reduction also slightly reduces the grid power due to better buffer placement.

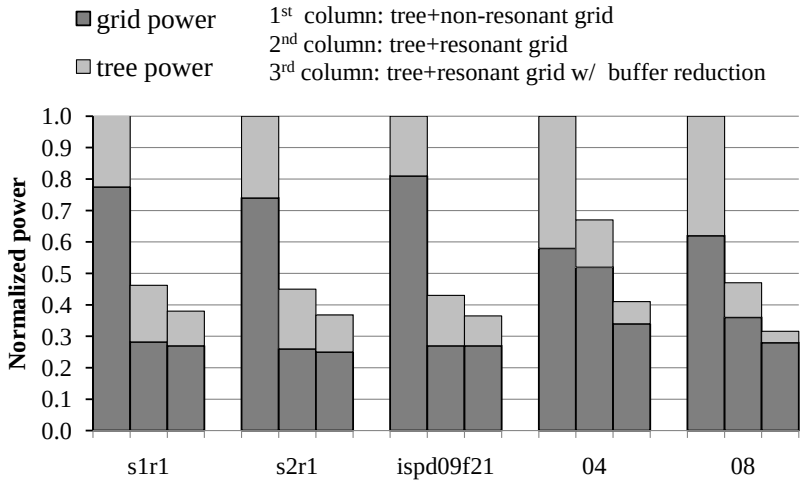


Figure 9.3: Representative grid and tree power reductions achieved by resonant clock grids

9.2.3 Skew

The skew of the ISPD2009 resonant grids without buffer reduction is slightly ($3.5ps$) better than non-resonant grids. The skews of other resonant CDNs are slightly

increased compared to the non-resonant CDNs, but all the skews are within the skew bound of $50ps$ (except for benchmark 01 after buffer reduction). The skew with the additional grid buffer reduction is slightly worse than the previous resonant CDN on average, because fewer buffers are driving the resonant grid and the phase-conflicts are increased. However, we achieve at least 13% more power savings at the expense of clock skew that is less than 3% of the clock period. Depending on the design requirements, a skew constraint can easily be added to the buffer reduction step in Algorithm 6 to ensure better skew for benchmarks such as 01.

Usually, skew is measured when the clock voltage reaches $V_{dd}/2$. When clock sinks are local clock buffers, these clock buffers starts to turn on/off when voltage is greater than V_{th}. A more accurate skew measurement can be taken at V_{th}. Or if the clock sinks are local clock buffers, the skew can be measured at the output of local clock buffers which represents the skew seen at the local clock trees. The transient analysis in Section 9.4, the skew at V_{th} is similar to the skew measured at $V_{dd}/2$.

9.2.4 Buffer Size and Area

Fewer, smaller buffers are needed to drive resonant clock grids since the majority of the grid instead resonates with the LC tanks. Less dependence on buffers to drive the grid results in less sensitivity to power supply and threshold voltage variation as shown in H-trees [65]. In our experiments, the final buffer sizes of resonant grids are often $3-7\times$ the minimum clock buffer size and about 50-90% smaller than the non-resonant clock grid buffers. We list the total buffer area including the top-level tree buffers and resonant grid buffers in Table 9.1 and 9.2.

For the ISPD2009 benchmarks, the average buffer area is reduced by 53.3% without the additional buffer reduction and by 77.8% with the additional buffer reduction. For the ISPD2010 benchmarks, we see buffer area reduced by 84.4% on average without the additional buffer reduction and by 90.9% with the additional buffer reduction step.

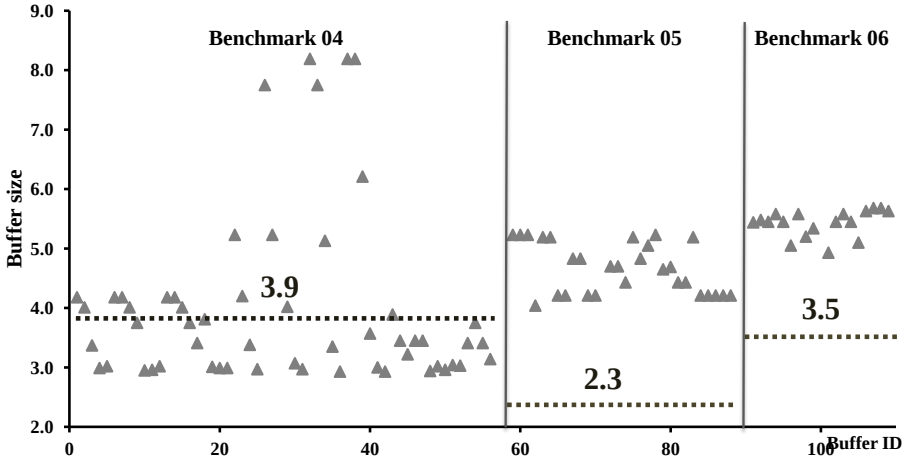


Figure 9.4: Buffer reduction can increase or decrease individual buffer sizes, but overall buffer area is reduced.

The minimum buffer (with size 1) used in experiments is the small buffer provided by IBM [70]. Figure 9.4 plots the resonant grid buffer sizes of several benchmarks. The dashed line in the figure is the resonant grid buffer size after AC-based sizing without buffer reduction. The triangle data represent buffer sizes after buffer reduction with AC- and resistance-based buffer sizing. The buffer reduction step removes half of the buffers from the grid so some of the remaining buffer sizes are increased to compensate. For benchmark 04 in Figure 9.4, some of remaining buffer sizes are increased, but many buffers are smaller than the initial size. In the other benchmarks, all sizes are increased but the total buffer size is still less than without buffer reduction due to fewer buffers. The AC- and resistance-based

buffer sizing method is able to size buffers more effectively by considering the capacitance and resistance distribution.

9.2.5 Inductor Overhead

The cost of implementing a resonant clock is the time to create and characterize the inductors along with the total metal area occupied by the instances of these on-chip inductors. The average inductor area is 28% of the top metal layer. The inductors are distributed sparsely in these designs which means that mutual inductance between inductors is not significant. The number of inductors in the ISPD2010 benchmarks is 3-5 except for the two biggest chips. In the ISPD2009 benchmarks, the chips are much bigger and more inductors are needed, but they are still sparse. On these benchmarks, the average number of inductors is 96 which is much less than the 830 inductors on the resonant clock of the IBM cell broadband engine processor [6].

The total inductor area before and after the grid buffer reduction step is almost the same. This verifies our discussion in Chapter 6.4 that the buffer capacitances in resonant grids are small and do not change the inductance much. However, the inductance re-sizing procedure is necessary to reduce phase conflict induced skew as much as possible.

9.2.6 Capacitor Overhead

Decoupling capacitance is added to make positive voltage swing for resonate clock as introduced in Chapter 3.2. Both the decoupling capacitance and the clock grid capacitance constitute one LC tank. While the inductors are carefully sized according to the clock capacitance, the decoupling capacitance size is also sized to ensure the tank com-

posed of the decoupling capacitance and inductor does not interfere the primary tank. All the decoupling are sized according to Equation 3.2.

9.2.7 Run Time

The largest benchmark 01 took 10.5 hours including extra HSPICE simulations after each LC/buffer sizing iteration, the top-level tree generation and skew tuning. For smaller ISPD2010 benchmarks (03-08), the total run time is 0.5-5 hours. As one single HSPICE simulation takes about 10 minutes for the big benchmarks, the actual run-time of the LC placement and buffer sizing is much less than this total run-time.

9.3 Resonant Clock Grid with Discrete Inductors

9.3.1 Synthesis Flow

Figure 9.5 is the synthesis flow of this experiment. This set of experiments validates the discrete LC tank placement and sizing method (Algorithm 3). As have shown in Chapter 9.2.4, though AC-based buffer sizing is able to find the minimum buffer size for the resonant grid, AC- and resistance-based buffer sizing is more efficient to size the buffers by considering the capacitance and resistance of the CDN. In this flow, the AC- and resistance-based method (Algorithm 5) sizes buffers. Because inductors can only be chosen from the library, phase tuning is a necessary step otherwise inductors and grid capacitance do not resonate at the designated clock frequency. Buffer incremental optimization is not performed as the results of buffer optimization have been shown in previous experiments.

The methodology is run on the ISPD 2010 benchmarks [69] using the synthesis

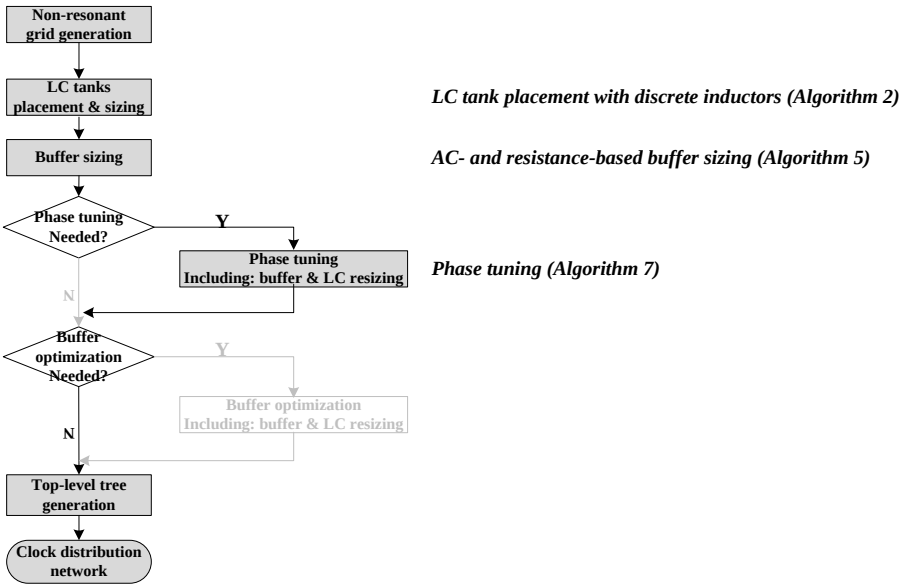


Figure 9.5: Resonant clock grid synthesis flow with discrete inductors. Uses Algorithm 3 for LC tank placement, Algorithm 5 for buffer sizing and Algorithm 7 for phase tuning.

flow in Figure 9.5. In this group experiments, $4\times$ minimum wire width is used in order to reduce the overall grid resistance. Table 9.3 lists the simulation results of the resonant grid with top-level tree. Figure 9.6 is a plot of a typical resonant grid with top-level tree. Sink and buffer mark sizes are proportional to sink and buffer sizes, but are not to scale.

9.3.2 Power

The dynamic power of a non-resonant clock is $P_{dyn} = CV_{dd}^2 f$ and is shown as "Pwr.^{1}" in Table 9.3. The "Pwr.^{2}" is the resonant clock power measured by HSPICE including all buffers, wires, inductors, sinks and short-circuit power. The "Pwr.^{2}/Pwr.^{1}" ratio in Table 9.3 shows the relative power savings of the resonant clock compared to the total switched capacitance (dynamic) power. On average, the power of resonant grid is 41% of the switched capacitance dynamic power.

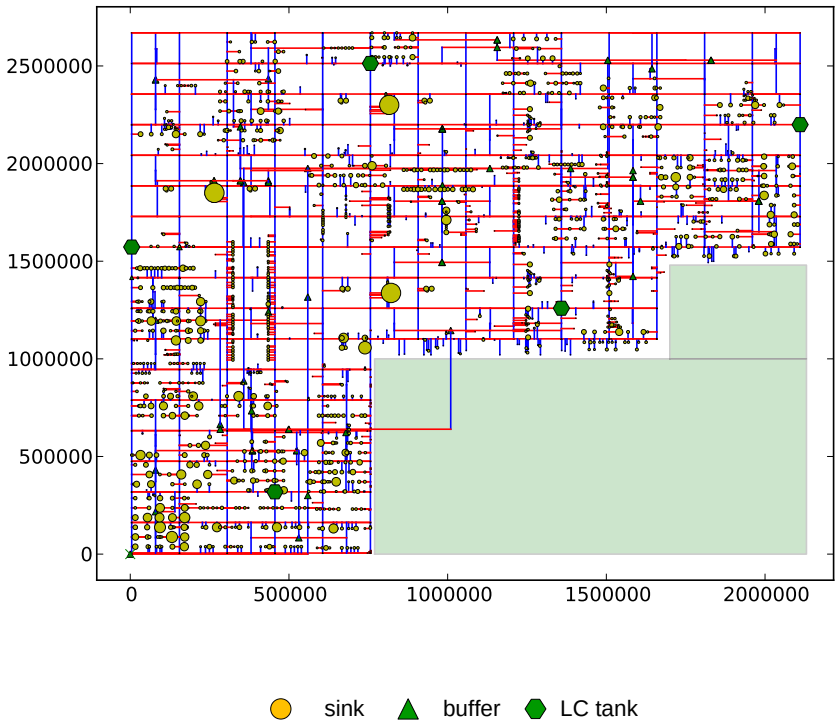


Figure 9.6: Clock network with LC tanks (benchmark 04) produced by synthesis flow (Figure 9.5. Dimensions in nm. Buffers, sinks and LC tanks are relatively sized but not to scale.)

9.3.3 Skew

The average global skew of the resonant clocks is $17ps$. Except for the two largest benchmarks, 01 and 02, all the other skews are significantly less than $20ps$. At 1GHz, $20ps$ global skew is about 2% of clock cycle which is very aggressive.

The skews of benchmark 01 and 02 are just above $30ps$. Compared with other benchmarks, these two benchmarks are much larger and have relatively low sink density as shown in the column "$\frac{S.Cap}{MA}$" in Table 9.3. Because of this, the resistive shielding effect becomes more significant and makes it difficult to tune for phase conflicts. To improve the

Table 9.3: Experimental results on ISPD 2010 benchmarks show that power is 41% of non-resonant power while skew is limited to $17ps$ on average.

	Non-resonant CDN				Resonant CDN				
	Sink	MA	$\frac{S.Cap}{MA}$	Pwr.[1]	Pwr.[2]	$\frac{Pwr.^2}{Pwr.^1}$	$\frac{LA}{MA}$	Skew	Time
	#	mm^2	$\frac{pF}{mm^2}$	mW	mW	%	%	ps	m
01	1107	64.0	0.3	472	368	78	29	32	130
02	2249	91.0	0.4	708	364	51	30	33	212
03	1200	1.4	12.9	66	26	39	92	12	2
04	1845	5.7	2.2	177	46	26	69	14	58
05	1016	5.8	0.9	85	48	57	29	16	23
06	981	1.5	8.4	78	23	29	100	10	2
07	1915	3.5	5.2	156	39	25	96	14	197
08	1134	2.6	5.0	111	27	24	81	8	3
Avg.	1431	21.9	4.4	232	118	41	66	17	78

Pwr.[1]: Switched capacitance CDN power. Pwr.[2]: Resonant grid power in HSPICE.
$\frac{S.Cap}{MA}$: sink density per area. MA: single metal layer (chip) area.
LA/MA: total inductor area normalized to metal layer area.

skew of a big chip, we can use wider wires to reduce the wire resistance and put a tighter constraint on the maximum resistance (R_{max}) when generating the subsets for LC tank placement.

9.3.4 Inductor Overhead

The average inductor area is 66% of one metal layer. Compared with the continuous inductor synthesis flow, more metal area is required to implement on-chip inductors. With multi-layer and 3D inductor technologies, the area of inductors can be reduced. However, only 3 sizes inductors are required whereas the continuous method required 16 sizes on average.

Table 9.4: Capacitor usage in resonant clock.

Name	ΔC (pf)	$\sum C_s$ (pf)	Decap. (pf)	$\sum C$ (pf)	$\frac{\Delta C+\sum C_s}{\sum C}$ (%)
01	195.0	0.0	5216.6	5411.6	3.6
02	402.8	123.8	9703.2	10229.8	5.1
03	17.8	0.0	760.7	778.5	2.3
04	31.4	0.4	1808.7	1840.5	1.7
05	20.1	4.3	812.7	837.1	2.9
06	22.1	0.0	929.8	951.9	2.3
07	37.2	1.3	1654.7	1693.2	2.3
08	16.7	0.0	1098.8	1115.5	1.5

ΔC	Capacitance used to compensate for ΔC mismatch in Algorithm 3.
$\sum C_s$	Capacitance inserted to sink stub nodes to tune the sink node phase.
Decap.	Decoupling capacitance inserted to make positive voltage swing.
$\sum C$	Total capacitance to resonant clock grid.
$\frac{\Delta C+\sum C_s}{\sum C}$	The percentage of phase tuning capacitance in the total capacitance.

9.3.5 Capacitor Overhead

The decoupling capacitance of the LC tanks are sized to 10 times the resonant capacitance [7,64] and additional capacitors are used to reduce the phase conflict. Table 9.4 lists total capacitor used in the resonant grid. From our experiments, ΔC is observed to be between $0.1 - 0.3\times$ the grid capacitance and the capacitance inserted to compensate for ΔC is typically $15 - 30pF$ except for benchmarks 01-02 which have odd sink densities as described earlier. Including both phase tuning and decoupling capacitance, the maximum total capacitance required of all benchmarks is $10.2nF$ for benchmark 02 while the average is just $2.5nF$. To put this in perspective, this maximum capacitance is only 5.7% of the power grid decoupling capacitance used on a Pentium II [53] which is $113mm^2$, a little bit larger than benchmark 02.

9.3.6 Inductor Q factor

The quality factor of the inductors affects the performance of the resonant clock. We simulated benchmark 03 with inductor Q ranging from 1 to 100 and show the power reduction in Figure 9.7. The inductor parasitic resistance is significant for low quality inductors ($Q < 10$) and becomes insignificant with $Q > 20$. Even with a very poor $Q = 1$, we see more than 25% power savings. The effect of inductor Q on skew is is only $2 - 5ps$ over the entire range of Q.

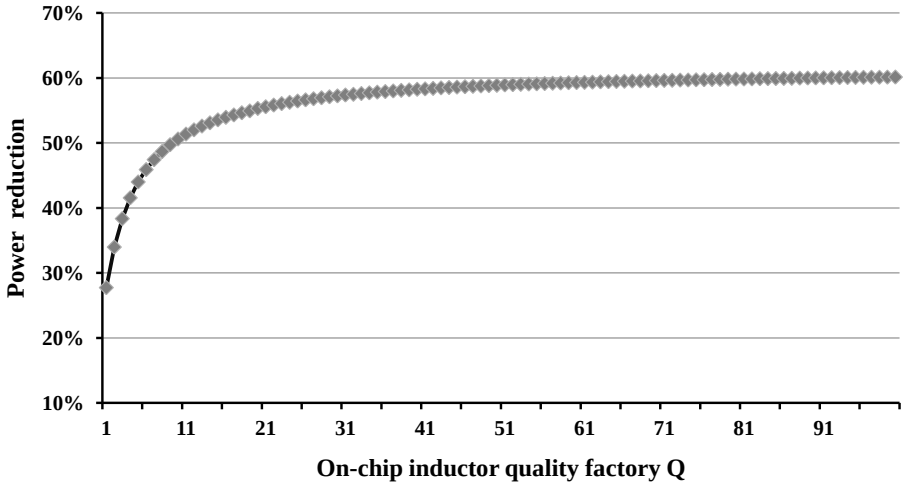


Figure 9.7: Significant power gains are obtained even with low-Q on-chip inductors

9.3.7 Run Time

The average time to synthesize the resonant clocks is 78.5 minutes. The CPU time in Table 9.3 is the total run time including grid generation, LC tank placement and sizing, phase tuning and top-level tree generation with skew reduction. In the experiments, 5%-30% of the total run time is spent on the actual resonant clock grid generation with the other portion on analysis and simulation.

9.4 Resonant Grid Transient and AC Analysis

Figure 9.8 is the frequency domain amplitude and phase plot of several clock sinks in the benchmark s1r1 resonant grid. As shown in Figure 9.8, there are two resonant frequencies: one is at 0.2GHz and the other is at 1GHz. The first resonant frequency is the decoupling capacitance resonant frequency, which is much less than the clock resonant frequency at 1GHz so that the two resonant frequencies do not interfere. The V_{swing} in Algorithm 4 is set to $0.75V$ and, at 1GHz, the magnitude of the sink voltages are greater than this. We also observe in Figure 9.8 that there is very low phase difference (skew) at the clock frequency of 1GHz.

Figure 9.9 is the transient analysis of the same resonant grid. Since AC analysis uses a sinusoidal signal instead of a digital clock input, we do not need to set the V_{swing} to a full $1V$. This is shown by Figure 9.9 where the clock input is a trapezoidal wave and the final transient wave has a full swing from $0V$ to $1V$ despite the target V_{swing} being only $0.75V$. The skew can again be verified to be quite small in Figure 9.9.

9.5 Comparison with State-of-the-Art Resonant Clocks

As there is no other resonant clock grid synthesis methodology, we implemented resonant versions of the ISPD2010 benchmarks that have uniform distributed LC tanks similar to a previous IBM resonant clock design [6] in which each sector is approximately $600\mu m \times 400\mu m$. In our experiments, one LC tank is inserted for a sector with area $500\mu m \times 500\mu m$. Inductor sizing, buffer sizing and top-level tree construction use the same methods as Figure 9.2. This is actually an improvement over the method in [6], be-

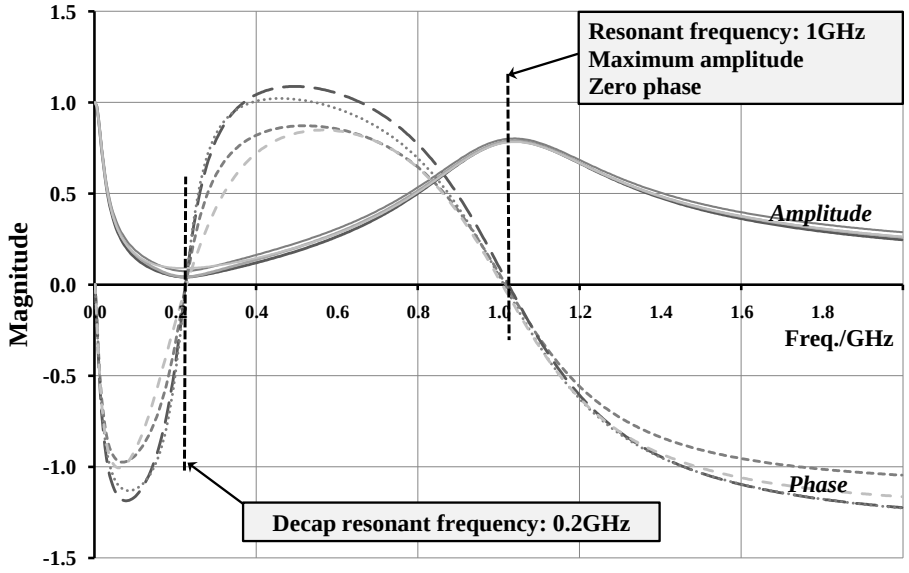


Figure 9.8: Resonant clock grid AC analysis

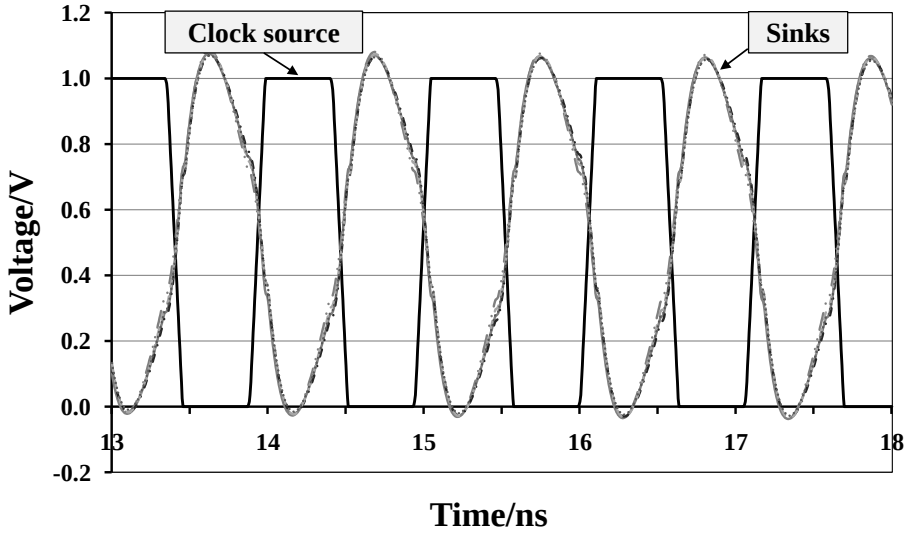


Figure 9.9: Resonant clock grid transient analysis

cause we custom size each inductor in the uniform placement while they assumed uniform placement and sizes.

Table 9.5 lists the results of the uniform LC tanks and the proposed continu-

ous flow. Benchmarks 01 and 02 failed to achieve full voltage swing with uniform LC tanks even with maximum buffer sizes, but power savings were still observed. For benchmarks 03-05 and 08, the uniform method achieves better power reduction, but the uniform LC placement typically uses more inductor area and has significantly worse skew than the method proposed in this thesis. Most often, our methodology is better when power, skew, and inductor overhead are all considered.

Table 9.5: Continuous inductor placement/sizing and uniform LC tank placement

	Uniform LC tanks			Resonant clock (Figure 9.2)		
	Skew (ps)	Power (mW)	LA/CA (%)	Skew (ps)	Power (mW)	LA/CA (%)
01	-	1229.5	3.5	19	231.4	27.5
02	-	1388.4	3.6	23	160.4	34.0
03	33	23.5	60.5	23	49.1	29.3
04	39	26.2	131.9	11	62.6	13.8
05	18	32.4	140.1	10	35.3	28.9
06	37	44.2	25.8	37	28.0	32.8
07	43	72.6	53.2	28	50.3	28.5
08	30	16.4	103.3	34	27.4	32.6
Avg.	28.4	354.2	65.2	23.1	80.6	28.4

9.6 Comparison with ISPD Clock Synthesis Contest

Table 9.6 lists all the previous published results of ISPD 2010 benchmarks in which we normalize the power and CPU time to the resonant clock in Table 9.3 for easier comparison. In Table 9.6, we also include the absolute difference in skew. It is important to note that the previous published results do not include short-circuit and leakage power in their results, whereas our results do. Therefore, the power comparison is actually conservative.

Most clock networks, however, require more buffering since they do not have the benefit of the LC tanks. As shown in Table 9.6, the average power of previous clock networks is more than $3.4-5.4\times$ our resonant clocks with the exception of Contango which is about $1.5\times$. LCS_mesh, also a clock grid architecture, has over $3.4\times$ the power.

We also measured the LCS in the resonant grid despite optimizing for global skew. In general, we have a few picoseconds of additional LCS than most ISPD 2010 results, but, in general, our results are very competitive. With such a drastic power savings, a small degradation in skew is likely to be acceptable. The power consumption of the ISPD results are typically more than $3\times$ of resonant grid. LCS_mesh has, on average, $8ps$ better LCS, but with $3.4\times$ more power. Contango has only $2.8ps$ better LCS and still consumes about 50% more power. Excluding benchmarks 01-02, CNSRouter and NTUclock have more than $5\times$ power than our resonant grid with roughly the same LCS.

Table 9.6: Comparison of resonant clock generated with Figure 9.5 flow with non-resonant ISPD2010 CDNs [88]

	LCS_mesh [88]			Contango [41]			CNSRouter			NTUclock		
Name	Pwr† ($\times$)	ΔLCS‡ (ps)	Time* ($\times$)	Pwr† ($\times$)	ΔLCS‡ (ps)	Time* ($\times$)	Pwr† ($\times$)	ΔLCS‡ (ps)	Time* ($\times$)	Pwr† ($\times$)	ΔLCS‡ (ps)	Time* ($\times$)
01	3.17	-16.7	0.09	0.54	-16.6	1.54	2.28	-16.5	0.00	0.80	-15.1	0.00
02	5.77	-20.0	0.17	1.03	-19.8	1.97	2.75	-19.1	0.01	2.29	-17.1	0.01
03	3.67	-5.5	0.16	2.18	-5.4	30.00	6.34	-7.0	0.06	6.52	-1.6	0.05
04	2.72	-3.9	0.01	1.56	-5.9	1.73	6.01	-6.4	0.01	7.05	-1.7	0.02
05	1.53	-11.8	0.01	0.73	9.1	1.00	3.62	-13.2	0.01	2.69	-10.8	0.01
06	3.82	-2.5	0.34	2.00	15.6	4.07	5.83	4.1	0.08	8.75	398.6	0.07
07	3.25	-3.5	0.00	1.84	-4.3	0.29	7.84	4.7	0.00	7.49	-0.5	0.01
08	3.61	-3.1	0.10	1.91	4.8	3.97	8.25	-0.5	0.11	6.14	-0.8	0.04
Avg.	3.44	-8.4	0.11	1.47	-2.8	5.57	5.37	-6.7	0.04	5.22	43.9	0.03

†Pwr: total capacitance normalized to the resonant CDN power

$^{\ddagger}\Delta$LCS: the difference between non-resonant clock LCS and resonant LCS

*Time: CPU time normalized to our CPU time

9.7 Robustness

9.7.1 Process Variation

We perform Monte Carlo HSPICE simulations with $\pm 15\%$ 3-sigma variation of buffer gate lengths, wire widths and sink load variations to assess robustness. We assume ideal inductors as Q was shown to have little effect on skew in Section 9.3.6. Each benchmark is simulated 500 times and the mean (μ) and standard deviation (σ) of global skew are measured. We compare the resonant CDNs with the baseline CDNs which are uniform grids driven by top-level buffered trees. The grid buffers in the baseline grid are placed and sized using a method similar to [81]. The results are listed in Table 9.7.

Table 9.7: Resonant clock distributions exhibit decreased worst case skew variation ($29ps$ compared to $32ps$) when considering process variations.

	Baseline CDN				Resonant clock (Figure 9.5)			
Ben.	Det (ps)	μ (ps)	σ (ps)	μ+3σ (ps)	Det (ps)	μ (ps)	σ (ps)	μ+3σ (ps)
01	16	27	5.6	43.8	32	38	5.7	55.1
02	26	39	8.1	63.3	33	41	5.3	57.0
03	15	17	3.1	26.3	12	15	1.6	19.8
04	12	14	3.2	23.6	14	20	1.6	24.5
05	13	15	3.1	24.3	16	18	2.1	23.8
06	13	15	2.5	22.5	10	13	1.1	16.9
07	13	15	3.6	25.8	14	19	1.3	23.1
08	17	19	3.1	28.3	8	10	1.0	13.1
Avg.	15.6	20.1	4.0	32.2	17.2	21.8	2.5	29.1

The deterministic skew (without variations) is shown as Det in Table 9.7. While the average deterministic and mean expected skew (μ) skew of the resonant CDN are slightly worse than baseline CDN, the standard deviation (σ) of the resonant CDN is $2.5ps$ compared to $4.0ps$ for the baseline. The resonant clocks have an average $\mu + 3\sigma$ skew of

$29ps$ while the baseline clocks are $32ps$. While clock grids are known for their robustness to variation, the resonant grid is able to provide more robustness with less power due to reduced dependence on clock buffers.

9.7.2 Inductor Variation

To address the concern of metal process and inductor variation, we ran Monte Carlo simulations of some benchmarks in HSPICE with $\pm 15\%$ 3-sigma variation of inductor sizes for 500 iterations. In Table 9.8, the mean skew (μ) is almost same as the deterministic skew without variations Det.. Each standard deviation (σ) is less than $1ps$ and the 3σ skew is within $3ps$ of the deterministic skew. Resonant clock grids exhibit very good robustness to process variation. Because of the extreme long run time, benchmark 01 and 02 are not simulated in this experiment.

Table 9.8: Monte Carlo simulations of inductor variation

	Det. (ps)	μ (ps)	σ (ps)	$\mu + 3\sigma$ (ps)
03	37	37	0.4	38
04	41	41	0.0	41
05	34	35	0.4	36
06	43	42	0.5	44
07	46	46	0.8	49
08	40	39	0.9	42

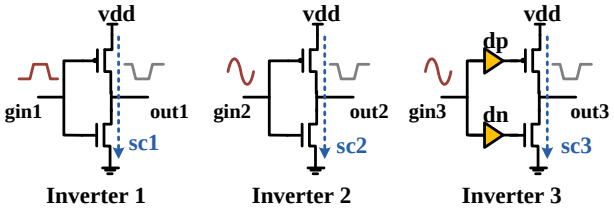
Chapter 10

Practical Issues with Resonant Clocks

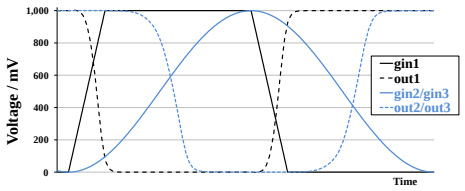
The application of resonant clocks has numerous practical issues which we discuss and present potential solutions in this Chapter.

10.1 Short Circuit Power of Sinusoidal Resonant Clocks

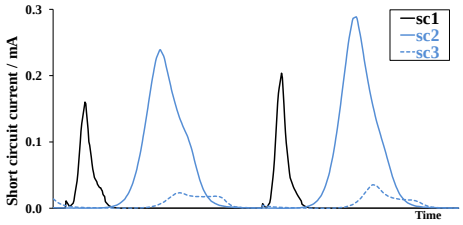
In the previous benchmarks, the clock sinks are represented as capacitors. However, for a regional CDN, the clock sinks are typically Local Clock Buffers (LCBs) which drive local clock trees. One frequent concern about resonant clocking is the sinusoidal shape of the resonant signal which has not been addressed in any previous resonant clocking methods [7, 8, 14, 21, 50, 71, 86, 92, 98]. A sinusoid has a slew rate that is 25% of the clock period compared to a more commonly used 10% in clock distributions, but the LCBs restore the slew to 10% by the time it is used by the sequential elements. The additional transition time, however, increases the total short circuit power in the LCBs. As a good feature, however, there is much less variation in the slew rate which can cause setup and hold time variation.



(a) Normal inverter and low short circuit power inverter



(b) Normal/sinusoidal clock inputs/outputs



(c) Short circuit currents

Figure 10.1: Short circuit current of an inverter

The short circuit current of an inverter is due to the simultaneous "on" state of both PMOS and NMOS transistors which forms a direct path from V_{dd} to V_{ss}. The short circuit current is a strong function of the input slew rate and load. A sinusoidal input wave can significantly increase the short circuit power by turning on the direct current path from V_{dd} to V_{ss} for a longer time. To quantify this effect in resonant clock networks, we measured the short circuit power of Inverter 1 in Figure 10.1(a) with an input slew of 10% of the clock period and Inverter 2 with a sinusoidal input of same clock period. Figure 10.1(b) shows the normal clock input $gin1$, sinusoidal clock input $gin2/gin3$ and the corresponding inverter outputs $out1$ and $out2$, and $out3$. Figure 10.1(c) shows that the short circuit current $sc2$ is much larger than $sc1$.

If we replace all the benchmarks sinks with LCBs that have equivalent input capacitances and place a $4\times$ load on the output, we see at least $10\times$ increase in buffer short

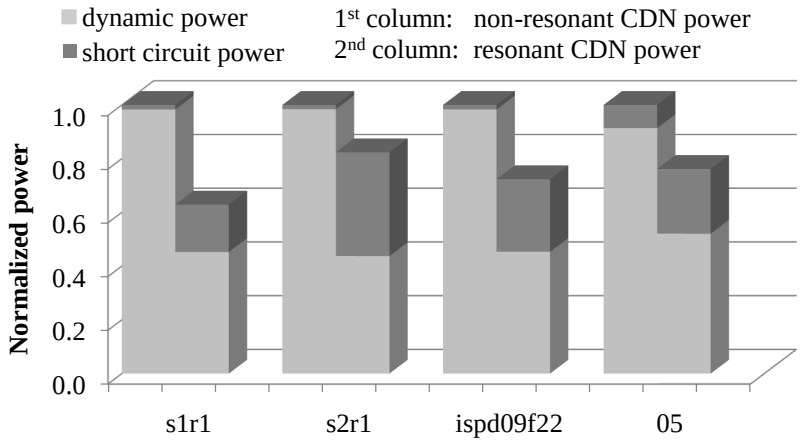


Figure 10.2: Dynamic and short circuit power

current power in resonant grids compared to non-resonant grids as shown in (Figure 10.2). This short circuit power constitutes more than 20% of the total power of resonant grid. Despite this, the reduction in the dynamic power is far larger than the increase in short circuit power and we still see 20-40% total power savings when including short circuit power. As speeds increase beyond 1GHz, the relative slew rate of the sinusoidal signal improves and this problem becomes less significant.

At 1GHz frequency, however, there are ways to reduce the short-circuit power. Several previous works have proposed special buffer designs to decrease short-circuit power [84, 91]. The basic idea is to avoid turning on the NMOS and PMOS transistors at the same time. This is demonstrated by Inverter 3 in Figure 10.1(a) which adds delay components dp and dn that adjust the clock arrival times at each transistor. By turning on transistors at specific times for rising and falling edges, the short circuit current can be reduced as shown by current $sc3$ in Figure 10.1(c). This is even smaller than the original short circuit current of Inverter 1 with a non-sinusoidal clock input. Therefore, the short-circuit power of the

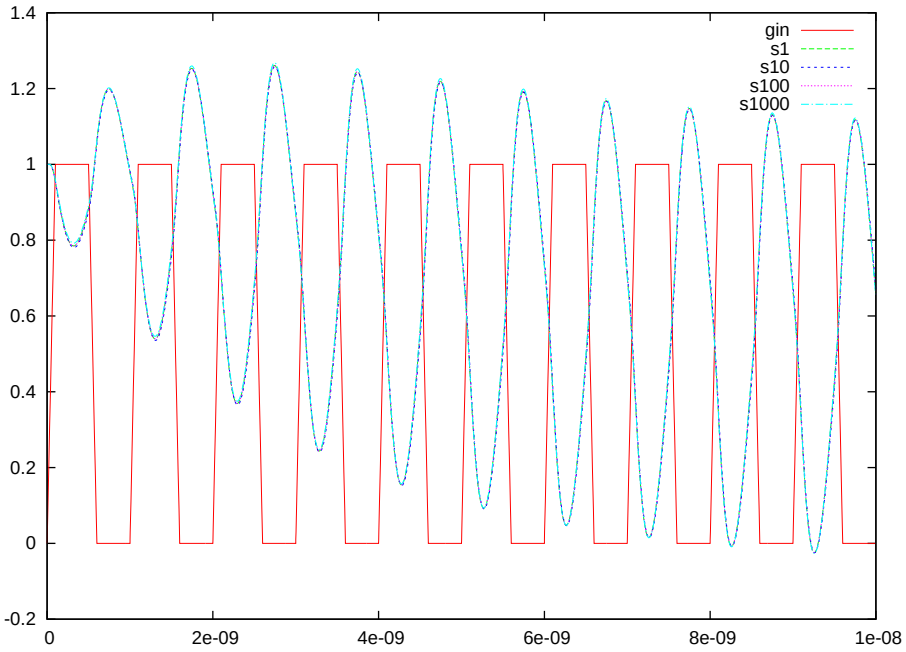


Figure 10.3: Starting waveform of resonant clocks: several clock cycles are needed to pre-charge the initial LC tanks

sinusoidal clocks is not of large concern.

10.2 LC Tank Precharge

Figure 10.3 is the transient behavior of some sinks during the first several clock cycles. The signal gin is the clock source while s1, s10, s100 and s1000 are clock sinks. The clock frequency is 1GHz. As shown in the figure, the clock swings are small, from 0.8V to 1.2V, in the first clock cycle. Six clock cycles later, the clock sinks are swing fully from 0V to 1V. The partial swing is because it takes time to pre-change the on-chip inductors at the first several clock cycles when the circuit is initially turned on. Several clock cycles after startup, the decoupling capacitors have reached steady state $\frac{V_{dd}}{2}$ and the clock signal becomes stable.

This is similar to a Phase-Locked Loop (PLL) or a Delay-Locked Loop (DLL). Usually, a PLL or DLL takes many cycles to reach steady state. Lock clock gating is usually at the LCBs which needs single cycle on/off time. In our resonant CDNs, the on/off of LCBs is not related to the LC tank precharge. Global clock gating (at chip level) only needs coarse on/off time, so this is acceptable.

10.3 Frequency Scaling

From Equation 3.1, the resonant frequency f is inversely proportional to the root of the product of inductance and capacitance, that is $f \propto \frac{1}{\sqrt{LC}}$. Considering the cost of resonant clock grid – extra on-chip inductor overhead – resonant clock design at lower frequency is more challenging as it requires much bigger inductors to resonate a fixed size capacitor. The previous experiments intentionally target at a relative slow speed 1GHz to ensure the feasibility of the proposed methods.

Some additional experiments test the resonant clock synthesis methods at higher frequencies with the synthesis flow in Figure 9.5. Figure 10.4 plots some random clock sink waveforms at 1GHz, 2GHz and 3GHz. The transient waveforms prove that the resonant clock skew is small. Figure 10.5 plots the power and skew of several benchmarks synthesized at different frequency. The power saving at higher frequency are consistent. Skew is improved at higher frequencies. At higher frequencies, smaller and less inductors are needed according to Equation 3.1.

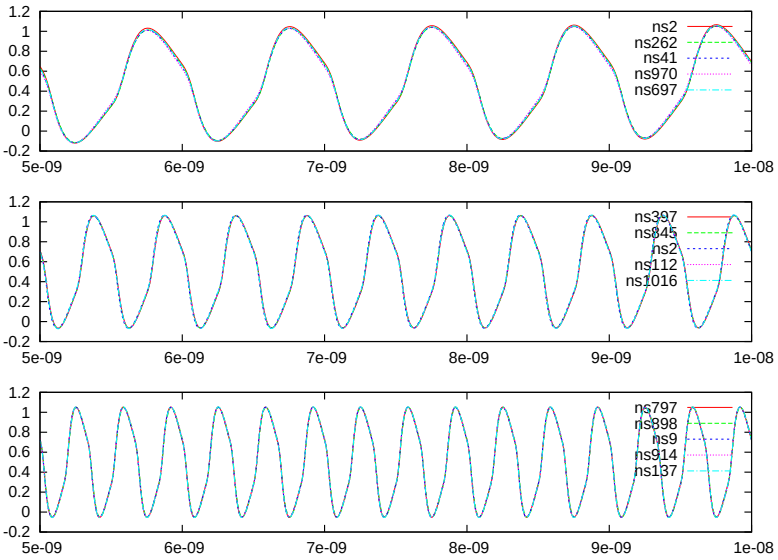


Figure 10.4: Transient resonant clock waves at different frequencies. Resonant clocks can be successfully synthesized at different frequencies, shown are 1GHz, 2GHz and 3GHz.

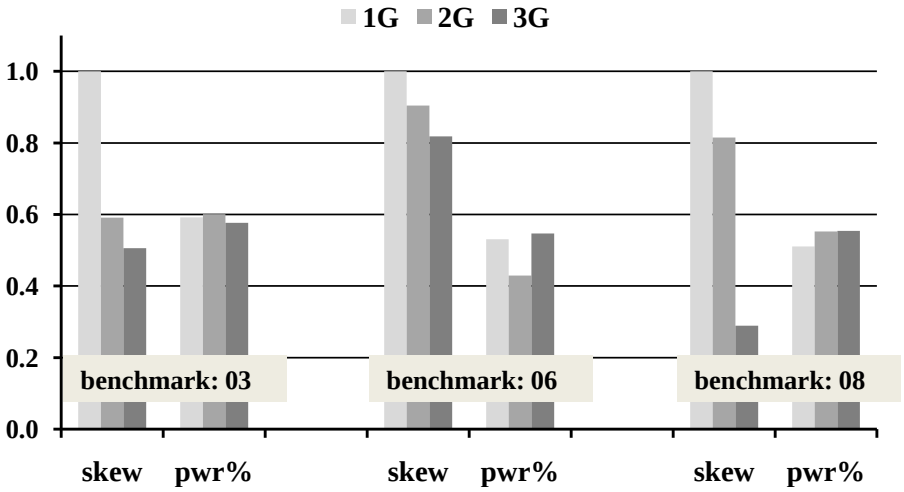
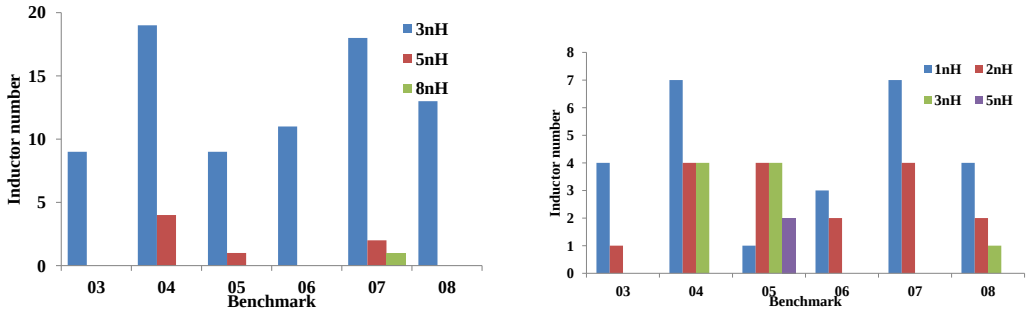


Figure 10.5: Resonant clock power and skew at different frequencies. Skew at 2GHz and 3GHz is normalized to skew at 1GHz.

10.4 Inductor Library Design

For the library LC tank placement introduced in Chapter 5.3, with more inductors in the library, a larger library will have more candidate placement which leads to better results with less phase tuning capacitor overhead. However, a larger library generates more candidate subsets for the ILP which means the run time will increase. Very big inductors do not help to improve the final result since they occupy large areas and it is hard to meet the area constraint, whereas very small inductors are also less useful because of the maximum distance though they are able to resonate capacitance. Therefore, the inductors library design should consider the benchmark characteristics in practice.

In the synthesis flow of Figure 9.5, there are three inductors in the library: 3nH, 5nH and 8nH. Figure 10.6(a) shows the inductors usage in the final results. The 5nH and 8nH inductors are rarely used. In another group of experiments which use the same flow, the inductor library includes smaller inductors: 1nH, 2nH, 3nH and 5nH.



(a) Inductor usages with 3nH, 5nH and 8nH inductor library

(b) Inductor usages with 1nH, 2nH, 3nH and 5nH inductor library

Figure 10.6: Inductor usage statistic of resonant clock grid with different inductor library. The synthesis flow uses fewer inductors with smaller inductor area.

Figure 10.6(b) shows the inductor usage of each benchmark and Table 10.1 lists

Table 10.1: Inductors and phase tuning capacitance usage with different inductor libraries. Using smaller inductors reduces on-chip inductor area but needs more capacitance to compensate for the ΔC mismatch.

L lib	03		04		05		06		07		08	
	$\frac{LA}{CA}$	Cap	$\frac{LA}{CA}$	Cap	$\frac{LA}{CA}$	Cap	$\frac{LA}{CA}$	Cap	$\frac{LA}{CA}$	Cap	$\frac{LA}{CA}$	Cap
nH	(%)	(pF)	(%)	(pF)	(%)	(pF)	(%)	(pF)	(%)	(pF)	(%)	(pF)
3,5,8	92	18	69	32	29	24	100	22	96	39	81	17
1,2,3,5	30	76	29	138	29	64	24	39	30	121	28	71

the total inductor area and phase tuning capacitance used of all benchmarks in the inductor library experiments. After adding smaller inductors to the library, the synthesis flow uses smaller and fewer inductors. The on-chip inductor area is also much smaller. While smaller inductors will cost less inductor area, the disadvantage is that more phase tuning capacitance is required to compensate for the larger ΔC as shown in Table 10.1. The smaller inductors cover more total capacitance so resistive effects are more prominent. This also leads to larger ΔC mismatch in the inductor covering. The designer has to decide the tradeoff between inductor cost and capacitance cost.

Chapter 11

Conclusions

While power consumption is a major problem in modern portable device design, resonant clocks are a promising technique to reduce the total power consumption by recycling energy. However, the design automation of resonant clocks is not well studied. Most previous resonant clocks are small hand tuned designs or restricted to simplified regular CDNs. This thesis is the first systemic research on resonant clock design automation including the discussion of the practical problems in application of resonant clocks.

11.1 Thesis Contributions

This thesis has explored the application of on-chip inductors in clock distribution networks to save dynamic power. The key contributions are:

Distributed LC tank placement and sizing

Most published LC tank resonant clock synthesis methods uniformly distribute LC tanks and size the on-chip inductors with inaccurate capacitance estimations. Therefore, previous published works fail to meet the designated frequency target or use a large amount

of on-chip inductors. In Chapter 5.2, an LC tank placement methodology is proposed to distribute LC tanks to better resonate with CDN capacitance. Moreover, the inductor sizes are exactly sized according the CDN capacitance distribution which was not considered in previous works. In this way, we are able to resonate the CDN at the target frequency with a small number of on-chip inductors.

The on-chip inductor design is often very complicated and requires significant design effort. A practical way to create a resonate CDN is to use a pre-designed inductor library. In Chapter 5.3.2, another LC tank placement method addresses this problem. This library-based LC tank placement methodology considers the wire, sink capacitance, parasitics and their distribution. This guarantees that the final resonant frequency will meet the design requirement.

Novel AC-based buffer sizing

Because of the parasitic resistance of CDNs, buffers are needed to compensate for the energy loss though most of the CDN will be driven by the LC tanks. Fewer smaller buffers are needed than non-resonant clock grids. A novel AC-based buffer sizing method is proposed in Chapter 6 that relies on small signal analysis which is widely used in analog design but has never been used in CDN design. AC analysis is much faster than transient analysis. Though AC analysis ignores the non-linear behaviors of buffers, capacitors and inductors, it is able to validate that the CDN resonant performance at the designated frequency quickly. By exploring the results of AC analysis, we are able to estimate the skew of the resonant clock without expensive transient analysis or simplified, inaccurate Elmore models. Therefore, the AC analysis is able to guide to optimize the transient performance of the clock network. Chapter 6 introduce the application of AC analysis in resonant clock

design.

Phase tuning

Phase conflict is a major cause of skew in resonant clock design. Chapter 7 introduces a phase tuning step in resonant clock design to optimize the performance of the CDN, which is based on the result of the previous AC analysis.

11.2 Future Work

This thesis proposes a complete resonant clock synthesis methodology, but there are several directions for future exploration. First of all, this work doesn't include the inductor design and the inductor model used is relatively simple. As there are several or tens of inductors on the same chip, the mutual inductance is not considered in this thesis, which could increase or decrease the inductance and change the resonant frequency. The on-chip inductor design is a complicated topic and needs special electromagnetic simulation tools. Another future topic should consider an effective way to do dynamic frequency scaling. Though resonant CDNs can change frequency in a small range, controlling the operation frequency in wide range needs additional research on how to modify capacitors or inductors. It would be interesting to investigate which inductors should be connected or disconnected when changing frequency while still providing small skew and power savings. This is also the same if we want to connect or disconnect extra capacitors to change the frequency.

Bibliography

[1] K. Agarwal, D. Sylvester, and D. Blaauw. A simple metric for slew rate of RC circuits based on two circuit moments. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 23(9):1346–1354, 2004.

[2] J. Aguilera and R. Berenguer. *Design and Test of Integrated Inductors for RF Application*. Kluwer Academic Publishers, 2003.

[3] C. Albrecht, A. Kahng, B. Liu, and I. Mandoiu. On the skew-bounded minimum-buffer routing tree problem. *IEEE Transactions On Computer Aided Design of Integrated Circuits*, 22:937 – 945, 2003.

[4] C. Alpert, A. B. Kahng, B. Liu, I. Mandoiu, and A. Zelikovsky. Minimum-buffered routing of non-critical nets for slew rate and reliability control. In *International Conference on Computer-Aided Design (ICCAD)*, pages 408–415, 2001.

[5] M. Bussmann and U. Langmann. Active compensation of interconnect losses for multi-GHz clock distribution networks. *IEEE Transactions on Circuits and Systems II (TCAS-II)*, 39:790–798, 1992.

[6] S. Chan, P. Restle, T. Bucelot, J. Liberty, S. Weitzel, J. Keaty, B. Flachs, R. Volant,

P. Kapusta, and J. Zimmerman. A resonant global clock distribution for the cell broadband engine processor. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 512–632, 2008.

[7] S. Chan, P. Restle, K. Shepard, N. James, and R. Franch. A 4.6GHz resonant global clock distribution network. *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 342 – 343, 2004.

[8] S. Chan, K. Shepard, and P. Restle. Design of resonant global clock distributions. *International Conference on Computer Design (ICCD)*, pages 248–253, 2003.

[9] S. Chan, K. Shepard, and P. Restle. Uniform-phase uniform-amplitude resonant-load global clock distributions. *IEEE Journal of Solid-State Circuits (JSSC)*, 40:102–109, 2005.

[10] C. Chang, S. Huang, Y. Ho, J. Lin, H. Wang, and Y. Lu. Type-matching clock tree for zero skew clock gating. In *Design Automation Conference (DAC)*, pages 714–719, 2008.

[11] W. Chao and W. Mak. Low-power gated and buffered clock network construction. *Transactions on Design Automation of Electronic Systems (TODAES)*, 13(1):1–20, 2008.

[12] W. Chen, W. Chen, and K. Hsu. Three-dimensional fully symmetric inductors, transformer, and balun in CMOS technology. *IEEE Transactions on Circuits and Systems I (TCAS-I)*, 54:1413–1423, 2007.

[13] Y. Chen, C. Dong, and D. Chen. Clock tree synthesis under aggressive buffer insertion. In *Design Automation Conference (DAC)*, pages 86–89, 2010.

[14] V. Chi. Salphasic distribution of clock signals for synchronous systems. *IEEE Transactions on Computers*, 43(5):597 – 602, 1994.

[15] Y. Choi and J. Yoon. Experimental analysis of the effect of metal thickness on the quality factor in integrated spiral inductors for RF ICs. *Electron Device Letters, IEEE*, 25:76–79, 2004.

[16] W. Condley, X. Hu, and M. R. Guthaus. Analysis of high-performance clock networks with RLC and transmission line effects. In *International Workshop on System level interconnect prediction (SLIP)*, pages 51–58, 2010.

[17] J. Cong and C.-K. Koh. Minimum-cost bounded-skew clock routing. In *International Symposium on Circuits and Systems (ISCAS)*, pages 215–218, 1995.

[18] S. de Rivaz., T. Lacrevaz, C. Bermond, B. Flechet, and A. Farcy. Rise time reduction of high speed digital signals on interconnects of the CMOS 45nm node by optimizing interconnect inductance. In *Microwave and Optoelectronics Conference (IMOC)*, pages 407 – 411, 2009.

[19] S. Deibele and J.B. Beyer. Attenuation compensation in distributed amplifier design. *IEEE Transactions on Microwave Theory and Techniques*, 37:1425–1433, 1989.

[20] J. Demmel, S. Eisenstat, J. Gilbert, X. Li, and J. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications*, 20(3):720–755, 1999.

[21] A. Drake, K. Nowka, T. Nguyen, J. Burns, and R. Brown. Resonant clocking using distributed parasitic capacitance. *IEEE Journal of Solid-State Circuits (JSSC)*, 39(9):1520 – 1528, Sep 2004.

[22] M. Edahiro and R. Lipton. Clock buffer placement algorithm for wire-delay-dominated timing model. In *Great Lakes Symposium on VLSI (GLVLSI)*, pages 143–147, 1996.

[23] J.C. Guo and T.Y. Tan. A broadband and scalable on-chip inductor model appropriate for operation modes of varying substrate resistivities. In *Radio Frequency Integrated Circuits (RFIC) Symposium*, pages 486–489, 2006.

[24] M. Guthaus, D. Kung, J. Narasimhan, R. Puri, L. Trevillyan, and C. Visweswariah. Variability-aware design optimizations using statistical timing. In *IBM Academy Conference on Design for Manufacturability*, 2004.

[25] M. Guthaus, D. Sylvester, and R. Brown. Clock tree synthesis with data-path sensitivity matching. In *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 498–503, 2008.

[26] M. Guthaus, N. Venkateswaran, and V. Zolotov. Optimization objectives and models of variation for statistical gate sizing. *Great Lakes Symposium on VLSI GLSVLSI*, 2005.

[27] M. Guthaus, N. Venkateswarant, C. Visweswariah, and V. Zolotov. Gate sizing using incremental parameterized statistical timing analysis. *International Conference on Computer-Aided Design (ICCAD)*, 2005.

[28] M. R. Guthaus. *Clock Tree Analysis and Synthesis Considering Process Parameters and Variation*. PhD thesis, University of Michigan, 2006.

[29] M. R. Guthaus, D. Sylvester, and R. B. Brown. Process-induced skew reduction in deterministic zero-skew clock trees. In *Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 84–89, 2006.

[30] M. R. Guthaus, G. Wilke, and R. Reis. Non-uniform clock mesh optimization with linear programming buffer insertion. In *Design Automation Conference (DAC)*, pages 74–79, 2010.

[31] K. Hardin, J. Fessler, D. Bush, and L. Inc. Spread spectrum clock generation for the reduction of radiated emissions. *Symposium on Electromagnetic Compatibility*, pages 227–231, 1994.

[32] Y.I. Ismail, E.G. Friedman, and J.L. Neves. Exploiting the on-chip inductance in high-speed clock distribution networks. In *IEEE Transactions on VLSI Systems (TVLSI)*, volume 9, pages 963 – 973, 2001.

[33] R. Jakushokas, M. Popovich, A. Mezhiba, S. Kse, and E. G. Friedman. *Power Distribution Networks with On-Chip Decoupling Capacitors*. Springer, 2011.

[34] M. Kamon, M. Tsuk, and J. White. Fasthenry: a multipole-accelerated 3-d inductance extraction program. In *Design Automation Conference (DAC)*, pages 678–683, 1993.

[35] H. Kaul, D. Sylvester, and D. Blaauw. Active shielding of rlc global interconnects. In *International workshop on Timing issues in the specification and synthesis of digital systems*, pages 98–104, 2002.

[36] B. Krauter and S. Mehrotra. Layout based frequency dependent inductance and resistance extraction for on-chip interconnect timing analysis. In *Design Automation Conference (DAC)*, pages 303–308, 1998.

[37] S. Kundu and U. Ghoshal. Inductance analysis of on-chip interconnects. In *European conference on Design and Test*, pages 252–255, 1997.

[38] C. Lee, C. Su, S. Huang, C. Lin, and T. Hsieh. Floorplanning with clock tree estimation. *IEEE International Symposium on Circuits and Systems (ISCAS)*, 6:6244–6247, 2005.

[39] M. Lee, A. Hill, and M.H. Darley. Interconnect inductance effects on delay and crosstalk for long on-chip nets with fast input slew rates. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 248–251, 1998.

[40] B. Lu, J. Hu, G. Ellis, and H. Su. Process variation aware clock tree routing. In *International Symposium on Physical Design (ISPD)*, pages 174–181, 2003.

[41] D. Leeand I. Markov. Contango: Integrated optimization of SoC clock networks. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1468–1473, 2010.

[42] Y. Massoud, J. Kawa, D. MacMillen, and J. White. Modeling and analysis of differential signaling for minimizing inductive cross-talk. In *Design Automation Conference (DAC)*, pages 804–809, 2001.

[43] Y. Massoud, S. Majors, T. Bustami, and J. White. Layout techniques for minimizing

on-chip interconnect self inductance. In *Design Automation Conference (DAC)*, pages 566–571, 1998.

[44] G.L. Matthaei, G.C. Chinn, C.H. Plott, and N. Dagli. A simplified means for computation for interconnect distributed capacitances and inductances. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 11:513 – 524, 2002.

[45] A. Maxim, B. Scott, E. Schneider, M. Hagge, S. Chacko, and D. Stiurca. A low-jitter 125-1250-MHz process-independent and ripple-poleless 0.18-um CMOS PLL based on a sample-reset loop filter. *IEEE International Solid-State Circuits Conference (ISSCC)*, 36:1673–1683, 2001.

[46] S. Musunuri and P.L. Chapman. Multi-layer spiral inductor design for monolithic DC-DC converters. *Industry Applications Conference*, pages 1270–1275, 2003.

[47] A. M. Niknejad and R. G. Meyer. Analysis, design, and optimization of spiral inductors and transformers for Si RF IC's. *IEEE Journal of Solid-State Circuits (JSSC)*, 33:1470–1481, 1998.

[48] K. J. Nowka, G. Carpenter, E. MacDonald, H. Ngo, B. Brock, K. Ishii, T. Nguyen, and J. Burns. A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling. *IEEE Journal of Solid-State Circuits (JSSC)*, 37(11):1441–1447, 2002.

[49] J. Oh and M. Pedram. Gated clock routing minimizing the switched capacitance. In *Design, Automation and Test in Europe (DATE)*, pages 692–697, 1998.

[50] F. O'Mahony, C. Yue, M. Horowitz, and S. Wong. Design of a 10GHz clock distribution network using coupled standing-wave oscillators. In *Design Automation Conference (DAC)*, pages 682–687, 2003.

[51] D. Pandini and G. Repetto. Spectral analysis of the on-chip waveforms to generate guidelines for EMC-aware design. *Springer Lecture Notes in Computer Science*, pages 339–366, 2006.

[52] D. Pandini, G. Repetto, and V. Sinisi. Clock distribution techniques for low-EMI design. *Springer Lecture Notes in Computer Science*, 2007.

[53] M. Pant, P. Pant, and D. Wills. On-chip decoupling capacitor optimization using architectural level prediction. *IEEE Transactions on VLSI Systems (TVLSI)*, 10:319–326, June 2002.

[54] C.R. Paul. *Introduction to Electromagnetic Compatibility*. John Wiley Interscience, 1992.

[55] C.R. Paul. *Electromagnetics for Engineers: With Applications to Digital Systems and Electromagnetic Interference*. John Wiley & Sons, 2004.

[56] Lawrence Pillage. *Electronic circuit and system simulation methods*. McGraw-Hill Professional, 1998.

[57] CPU power dissipation. http://en.wikipedia.org/wiki/List_of_CPU_power_dissipation.

[58] A. Rajaram, J. Hu, and R. Mahapatra. Reducing clock skew variability via cross links. In *Design Automation Conference (DAC)*, pages 18–23, 2004.

[59] A. Rajaram and D. Z. Pan. Meshworks: an efficient framework for planning, synthesis and optimization of clock mesh networks. In *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 250–257, 2008.

[60] P. Restle, A. Ruehli, and S. Walker. Dealing with inductance in high-speed chip design. In *Design Automation Conference (DAC)*, pages 904–909, 1999.

[61] P.J. Restle, C.A. Carter, J.P. Eckhardt, B.L. Krauter, B.D. McCredie, K.A. Jenkins, A.J. Weger, and A.V. Mule. The clock distribution of the POWER4 microprocessor. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 144–145, 2002.

[62] R.J. Riedlinger, R. Bhatia, L. Biro, B. Bowhill, E. Fetzer, P. Gronowski, and T. Grutkowski. A 32nm 3.1 billion transistor 12-wide-issue Itanium processor for mission-critical servers. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 84–86, 2011.

[63] J. Rosenfeld and E. Friedman. Design methodology for global resonant H-tree clock distribution networks. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2073–2076, 2006.

[64] J. Rosenfeld and E. Friedman. Design methodology for global resonant H-tree clock distribution networks. *IEEE Transactions on VLSI Systems (TVLSI)*, 15(2):135–148, 2007.

[65] J. Rosenfeld and E. G. Friedman. Sensitivity evaluation of global resonant H-tree clock distribution networks. *Great Lakes Symposium on VLSI (GLSVLSI)*, pages 192–197, 2006.

[66] R. Samanta, G. Venkataraman, and J. Hu. Clock buffer polarity assignment for power noise reduction. *International Conference on Computer-Aided Design (ICCAD)*, pages 558–562, 2006.

[67] A. Scuderi, T. Biondi, E. Ragonese, and G. Palmisano. A lumped scalable model for silicon integrated spiral inductors. *IEEE Transactions on Circuits and Systems I (TCAS-I)*, 51:1203–1209, 2004.

[68] R. M. Senger, E. D. Marsman, M. S. McCorquodale, F. H. Gebara, K. L. Kraver, M. R. Guthaus, and R. B. Brown. A 16-bit mixed-signal microsystem with integrated CMOS-MEMS clock reference. In *Design Automation Conference (DAC)*, pages 520–525, 2003.

[69] C. Sze. ISPD 2010 high performance clock network synthesis contest. In *International Symposium on Physical Design (ISPD)*, pages 143–143, 2010.

[70] C. Sze, P. Restle, G. Nam, and C. Alpert. Clocking and the ISPD'09 clock synthesis contest. In *International Symposium on Physical Design (ISPD)*, pages 149–150, 2009.

[71] B. Taskin, J. Demaio, O. Farell, M. Hazeltine, and R. Ketner. Custom topology rotary clock router with tree subnetworks. *Transactions on Design Automation of Electronic Systems (TODAES)*, 14(3):44:1–44:14, 2009.

[72] G. E. Tellez and M. Sarrafzadeh. Minimal buffer insertion in clock trees with skew and slew rate constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 16(4):333–342, 1997.

[73] M. Thomson, P. Restle, and N. James. A 5GHz duty-cycle correcting clock distribution network for the POWER6 microprocessor. *IEEE Journal of Solid-State Circuits (JSSC)*, pages 1522–1529, 2006.

[74] J.-L. Tsai and C. C.-P. Chen. Process variation robust and low-power zero-skew buffered clock-tree synthesis using projected scan-line sampling. In *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 1168–1171, 2005.

[75] J.-L. Tsai, T.-H. Chen, and C. C.P. Chen. Zero skew clock-tree optimization with buffer insertion/sizing and wire sizing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(4):565–573, 2004.

[76] R.-S. Tsay. Exact zero skew. In *International Conference on Computer-Aided Design (ICCAD)*, pages 336–339, 1991.

[77] R.-S. Tsay. An exact zero-skew clock routing algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(2):242–249, 1993.

[78] L. P. P. P. van Ginneken. Buffer placement in distributed RC-tree networks for minimal Elmore delay. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 865–868, 1990.

[79] International technology roadmap for semiconductors (ITRS), 2010. `http://www.itrs.net/Links/2010ITRS/Home2010.htm`.

[80] D. Velenis, M. C. Papaefthymiou, and E. G. Friedman. Reduced delay uncertainty in high performance clock distribution networks. In *Design, Automation and Test in Europe*, pages 68–73, 2003.

[81] G. Venkataraman, Z. Feng, J. Hu, and P. Li. Combinatorial algorithms for fast clock mesh optimization. In *International Conference on Computer-Aided Design (ICCAD)*, pages 563–567, 2006.

[82] A. Vittal and M. Marek-Sadowska. Power optimal buffered clock tree design. In *Design Automation Conference (DAC)*, pages 497–502, 1995.

[83] P. Vuillod, L. Benini, A. Bogliolo, and G. Micheli. Clock skew optimization for peak current reduction. *International Symposium on Low Power Electronics and Design*, pages 265–270, 1996.

[84] G. Wilke, R. Fonseca, C. Mezzomo, and R. Reis. A novel scheme to reduce short-circuit power in mesh-based clock architectures. In *Symposium on Integrated Circuits and System Design (SBCCI)*, pages 117–122, 2008.

[85] E. Wong, J. Minz, and S. Lim. Decoupling-capacitor planning and sizing for noise and leakage reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, pages 395–400, 2007.

[86] J. Wood, T. Edwards, and S. Lipa. Rotary traveling-wave oscillator arrays: A new clock technology. *IEEE Journal of Solid-State Circuits (JSSC)*, 36(11):1654–1664, 2001.

[87] J. G. Xi and W. W.-M. Dai. Buffer insertion and sizing under process variation for low power clock distribution. In *Design Automation Conference (DAC)*, pages 491–496, 1995.

[88] L. Xiao, Z. Xiao, Z. Qian, Y. Jiang, T. Huang, H. Tian, and E.F.Y. Yong. Local

clock skew minimization using blockage-aware mixed tree-mesh clock network. In *International Conference on Computer-Aided Design (ICCAD)*, pages 458–462, 2010.

[89] X. Ye, P. Li, M. Zhao, R. Panda, and J. Hu. Analysis of large clock meshes via harmonic-weighted model order reduction and port sliding. In *International Conference on Computer-Aided Design (ICCAD)*, pages 627–631, 2007.

[90] X. Ye, S. Narasimhan, and P. Li. Leveraging efficient parallel pattern search for clock mesh optimization. In *International Conference on Computer-Aided Design (ICCAD)*, pages 529–534, 2009.

[91] C. Yoo. A CMOS buffer without short-circuit power consumption. *IEEE Transactions on Circuits and Systems II (TCAS-II)*, 47:935–937, 2000.

[92] Z. Yu and X. Liu. Implementing multiphase resonant clocking on a finite-impulse response filter. *IEEE Transactions on VLSI Systems (TVLSI)*, 17(11):1593 – 1601, 2009.

[93] C.P. Yue and S.S. Wong. On-chip spiral inductors with patterned ground shields for Si-based RF ICs. *IEEE Journal of Solid-State Circuits (JSSC)*, 33(5):743–752, 1998.

[94] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani. Analysis of the impact of process variations on clock skew. *IEEE Transactions on Semiconductor Manufacturing*, 13(4):401–407, 2000.

[95] X. Zeng, D. Zhou, and W. Li. Buffer insertion for clock delay and skew minimization. In *International Symposium on Physical Design (ISPD)*, pages 36–41, 1999.

[96] P. Zhou, K. Sridharan, and S. Sapatnekar. Congestion-aware power grid optimization for 3D circuits using MIM and CMOS decoupling capacitors. pages 179–184, 2009.

[97] Q. Zhu and M. Zhang. Low-voltage swing clock distribution schemes. In *International Symposium on Circuits and Systems*, pages 418 – 421, 2001.

[98] C. Ziesler, S. Kim, and M. Papaefthymiou. A resonant clock generator for single-phase adiabatic systems. pages 159–164, 2001.

[99] A. Zolfaghari, A. Chan, and B. Razavi. Stacked inductors and transformers in CMOS technology. *IEEE Journal of Solid-State Circuits (JSSC)*, 36:620–628, 2001.

Appendix A

Clock Tree Optimization for Electromagnetic Compatibility (EMC)

The on-chip inductance and high activity switching signals can cause Electromagnetic Interference (EMI) problems. The high frequency clock signal is one of the major sources of on-chip EMI. This appendix introduce the background of clock EMI problem and provides a solution to reduce the clock EMI effect. As this work is not directly related to the resonant clock design, the clock tree EMI optimization work is attached as an appendix. This work is orthogonal to resonant clocks as it tries to remove problems caused by inductance in on-chip interconnects.

A.1 On-chip Electromagnetic Interference

EMI generated by active components is a primary concern among many mobile embedded systems designers. Besides being a requirement for Federal Communications Commission (FCC) approval, it is also necessary to minimize noise in mixed-signal sys-

tems. EMI is a complex topic because it is related to the IC design, the package and enclosure, and the board wiring among other things.

On-chip EMI noise has two major causes: on-chip signal switching for high-frequency noise and off-chip supply current demand for low-frequency noise. The high-frequency component is primarily due to Signal Integrity (SI) and the low-frequency component is primarily due to Simultaneous Switching Noise (SSN) and supply network inductance.

Previously, several methods have been proposed to address EMI through decoupling capacitors (decaps) at the die and package-level [85, 96] and random jitter insertion for spread spectrum PLLs [31]. SSN can also be reduced by intentionally skewing the switching of clock sinks [83] to reduce SSN by spreading the activity of different modules to reduce peak demand. The SSN of the clock network itself can be reduced by selecting the polarity of clock buffers [66]. These methods, however, only address supply-related EMI.

The high-frequency resonance in clock networks can be quite significant due to long global interconnects in the clock [73]. Previous works [51, 52] have cited anecdotal evidence that suggests high-frequency EMI can be reduced by manually sizing clock buffers in a small digital logic block. They criticize commercial tools for being inadequate for the job, but the problem is that no such algorithms exist.

There is much previous work on buffer insertion and sizing in clock tree synthesis. Buffers are inserted or relocated on long wires to reduce the interconnect delay [75, 95] or reduce power [82]. Buffer insertion for maximum slew rates has been used in signal networks to improve noise immunity [3, 4].

No previous work on clock tree design has investigated the minimum slew rates of signal. As we will show, minimum slew rate is very important in EMC-aware design. Minimum slew rates can also result in significant over-shoot and increase additional skew as on-chip inductances continue to increase [16]. Two prior works [51, 52] considered EMI of clock nets using hand optimization on a small digital block, but a clock synthesis algorithm was not investigated.

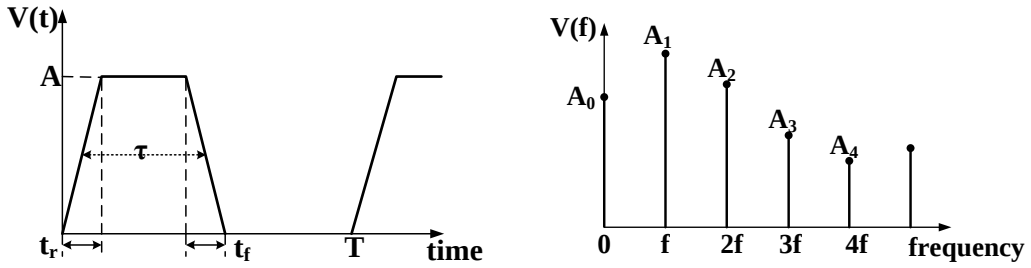
The appendix will first introduce background of clock signal spectral analysis and formally defines the problem of clock tree optimization considering EMC in Appendix A.2. Appendix A.3 proposes the dynamic programming algorithm to solve the problem. Then in Appendix A.4 the experimental methods and results are presented.

A.2 Problem Statement and Models

A.2.1 Spectral Analysis of Clock Signal

A clock signal can be represented as a periodic pulse with several main parameters as shown in Figure A.1(a): period T, frequency f, amplitude A, duty cycle D, rising time t_r and falling time t_f (usually 10%-90% or 90%-10% slew). The duty cycle D is defined as τ/P, where τ is the pulse width between $A/2$ points.

Given any signal, Fourier analysis can be used to transfer the time domain function into a frequency domain function as shown in Figure A.1(b). With the Fourier analysis, a complicated periodic function can be written as the sum of a series of sine and cosine functions. Each of these sine and cosine functions represents a distinct frequency and the entire



(a) In time domain

(b) In frequency domain

Figure A.1: Clock signal in time and frequency domain

periodic clock signal can be represented as

$$V(t) = A_0 + A_1\cos(\omega_0 t + \phi_1) + A_2\cos(2\omega_0 t + \phi_2) \\ +A_3\cos(3\omega_0 t + \phi_3) + \cdots, \tag{A.1}$$

where $\omega_0 = 2\pi f$ and $f = 1/T$ is the fundamental frequency, ϕ_i $(i \geq 1)$ is the phase angle of each spectral component. The sinusoidal components can be calculated as

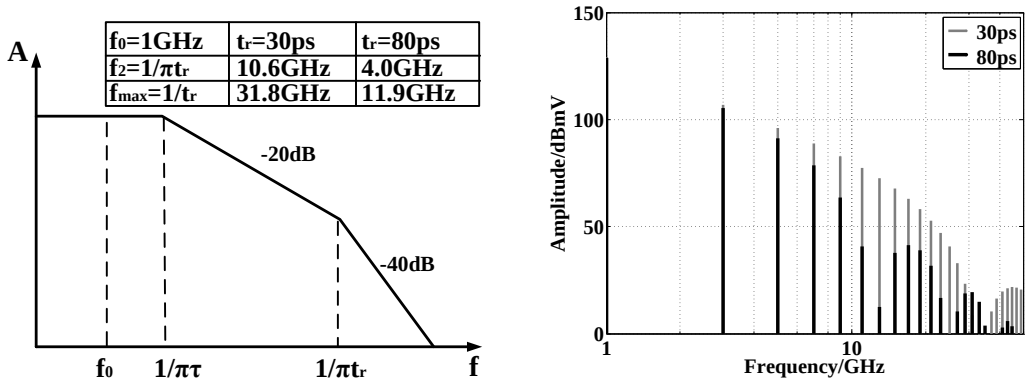
$$A_0 = \frac{1}{T}\int_0^T V(t)dt = A\frac{\tau}{T} \tag{A.2}$$

$$|A_n| = 2A\frac{\tau}{T}\left|\frac{\sin(n\pi\tau/T)}{n\pi\tau/T}\right|\left|\frac{\sin(n\pi t_r/T)}{n\pi t_r/T}\right| \tag{A.3}$$

where we assume rising time equals the falling time ($t_r = t_f$) as in [54].

Figure A.2(a) shows the bounds on these spectral components along with two distinct break points. The first one is $f_1 = 1/\pi\tau$. The spectral component amplitude above this point will decrease at -20dB/decade. The spectral component amplitude above the second break point, $f_2 = 1/\pi t_r$, will decrease at -40dB/decade. The majority of spectrum power lies below three times the second break point [55], that is $f_{max} = 3 \times f_2 \approx 1/t_r$. So the high-frequency spectral component is determined by the rise/fall times (t_r/t_f).

Figure A.2(b) is the spectral component of a clock signal when $t_r = 30ps$ and



(a) Spectral amplitude bounds

(b) Spectral amplitude of clock ($f = 1GHz$)

Figure A.2: Clock Signal Spectral analysis

$t_r = 80ps$. It clearly shows that most spectral power is under $f_{max} = 12GHz$ when $t_r = 80ps$. When $t_r = 30ps$, f_{max} is $32GHz$ and there is more spectral power distributed beyond $12GHz$. With higher operating frequency and faster signal switching, this high-frequency spectral power will lead to potential EMI problems. In order to reduce the amplitude of the high-frequency spectral content, we need to increase the rise/fall times. In other words, we need to control the fastest slew rates of the clock signal without introducing additional skew.

A.2.2 Radiation Emission

Each segment of wire is surrounded by an electric field. The radiation power from the wire is proportional to the frequency, wire length and current as

$$Radiation\ power \propto frequency \times length \times current. \tag{A.4}$$

Since the clock is the longest and most active signal on a chip, the radiation emission of the clock signal is more significant than other nets. Typically, buffers are inserted into long wires to reduce the interconnect delay and improve the signal slew. Since radiation power is proportional to frequency, wire length and current, the buffer locations and sizes become the key problem to reduce total radiation power. By changing the buffer position, the current through the wire and wire length therefore change the radiation power. The radiation power is highly computational and SPICE simulation is needed for accurate current measurement.

A.2.3 Problem Statement

The clock tree EMC optimization problem can be formulated as follows. Given an initial buffered clock tree $T(V, E, B)$, a buffer library L, a maximum slew rate constraint S_{max}, and a minimum slew rate constraint S_{min}, determine the location p_i and size s_i of each buffer such that clock tree high-frequency spectrum contents are minimized while skew and power of the clock tree are also minimized. Table A.1 summarizes all the notations used in our algorithm.

A.2.4 Maximum/Minimum Slew Constraints

There are several parameters that determine the output slew of a buffer: buffer size, input slew to the buffer and the effective load of the buffer. By carefully setting maximum/minimum load limits of each buffer, we can restrict the output slew rate of the buffer to a feasible range. C^H and C^L represent the maximum and minimum load limits. C^H is decided to preserve good noise immunity while C^L determines how aggressively we wish to address EMI noise as discussed in Section A.2.1.

Table A.1: Notations

T	The initial buffered clock tree, $T = V \cup E \cup B$
V	Nodes in clock tree, $V = \{v_1, v_2, \cdots, v_m\}$
E	Edges in clock tree, $E = \{e_1, e_2, \cdots, e_n\}$
B	Buffers in clock tree, $B = \{b_1, b_2, \cdots, b_k\}$
L	Buffer library, $L = \{l_1, l_2, \cdots, l_w\}$
S_{max},S_{min}	Maximum, minimum slew constraint
S	Sizes of buffers, s_i is the size of b_i, $s_i \in L$
$C^H_{l_i}$,$C^L_{l_i}$	The maximum, minimum buffer load of l_i
D_{b_i}	Intrinsic delay of buffer b_i
R_{b_i}	Output resistance of buffer b_i
C_{b_i}	Load of buffer b_i
d^v,D^v	Min, max delay to sinks at node v in original tree
d^v_i,D^v_i	Min, max delay to sinks of solution i at node v

Deriving C^H: The input slew is set to the slowest slew rate S_{max} and the buffer load is increased until the output slew is greater than S_{max}. This setup guarantees that the "slowest" case slew is bounded by S_{max}.

Deriving C^L: The input slew is set to the fastest slew rate S_{min} and the buffer load is decreased until the output slew is less than S_{min}. This setup guarantees that the "fastest" case slew will be greater than S_{min}.

A.2.5 Delay and Slew Models

We use the Elmore delay model to calculate interconnect delay. The Elmore delay of an edge $e = (v_i, v_j)$ is $R(e)(C(v_j) + C(e)/2)$, where $C(v_j)$ is the subtree capacitance of node v_j, $R(e)$ is the resistance of e, and $C(e)$ is the capacitance of e. We use a linear buffer delay model of $D_{b_i} + R_{b_i} \times C_{b_i}$ where D_{b_i} is the intrinsic buffer delay and R_{b_i} is the buffer output resistance of buffer size b_i. Based on C^H, C^L and S_{max}, S_{min}, buffer output slews are approximated according to the buffer load. According to the S2M model [1], the

interconnect slew rate is $\sqrt{Slew^2(step) + T_R^2}$, where T_R is the input slew, $Slew(step)$ is the slew for step excitation and can be denoted by the RLC of the circuit [1].

Though Elmore delay is not accurate compared with SPICE simulation, Elmore delay can estimate the delay very fast with reasonable accuracy. SPICE simulation can easily be used for more accurate delay and slew estimations. Our algorithm is compatible with any delay model.

A.3 Clock Tree EMI Optimization Algorithm

In this section, we propose a dynamic programming algorithm for EMC design. Starting with a buffered clock tree, this algorithm searches for the optimal buffer positions and buffer sizes.

A.3.1 Van Ginneken's Dynamic Programming

Van Ginneken proposed a dynamic programming to minimize the delay of buffered trees [78] and there has been much research on buffer insertion based on Van Ginneken's algorithm. However, these are typically not clock networks. The problem of using dynamic programming in clock tree buffer insertion is that the aim of clock tree synthesis is to minimize the skew while Van Ginneken's method tries to minimize the delay. Suboptimal solutions are pruned based on the delay in Van Ginneken's algorithm.

In clock tree buffer insertion, the pruning is more complicated than in normal wire buffer insertion. Figure A.3 is a small example to show that no solution can be pruned if the higher level delay information is unknown. In a bottom-up dynamic buffer insertion,

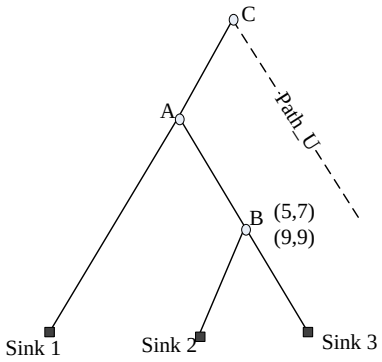


Figure A.3: Pruning in dynamic programming of clock tree

at node B, we assume that there are two solutions. The maximum and minimum delay of the first solution are 5 and 7, while the second are 9 and 9. If only considering skew, the second solution is better than the first one. However, without knowing the delay from sink1 to node A, we can't prune any solution at node B. If the delay from sink 1 to node A is 5, the first solution at node B is better than the other one. For further steps, no solution can be pruned at node A without delay information of path_U. Therefore, it is not possible to prune solutions while using dynamic programming to insert buffers for a whole tree.

In our dynamic programming algorithm, instead of optimizing the whole tree, we successfully reduce the complexity of dynamic programming by iteratively optimizing critical paths.

A.3.2 Top Level Algorithm

Algorithm 8 is an overview of our approach. Before starting the dynamic programming, we segment the edges of the clock tree in step 1. Each long edge is segmented into several smaller edges. Buffers can only be placed at the existing nodes V in the tree.

Segmenting the tree will generate more candidate nodes for possible buffer locations in the following steps.

In the next step, a critical path, either maximum or minimum delay path, is selected. The buffers which are not on the critical path are assumed fixed and do not change location or size. For more accurate delay models, slew influence of the shared buffer in critical and non-critical paths can be considered when evaluating the non-critical path delay. The non-critical path delay is used as a reference while pruning in later steps. We then relocate and size the buffers in the critical path as described in the next section.

Our goal is to optimize the clock tree EMC without scarifying the skew and power. So we choose skew as the optimizing objective and repeat our path optimization until skew can not be further improved. In our implementation, we optimize the maximum delay path first then optimize the minimum delay path. The algorithm can be extended to optimize the radiation power or maximize the slew rate of each buffer for more emphasis on EMC.

A.3.3 Relocating and Sizing a Critical Path

Algorithm 9 is the pseudo-code of relocating and sizing a path simultaneously by dynamic programming. Figure A.4 is an example of this procedure. The black buffer is fixed and white buffers are buffers on the critical path. All the nodes on the path are visited in a bottom-up order. If there is a buffer b_j inserted at node v_i, we are going to relocate and size b_j. In this example, b_j is originally inserted at node v_4. We define a $start\_node$ as a merging node or a buffer which b_j drives. In Figure A.4, node v_1 is the $start\_node$. We call the buffer which drives b_j as the driver buffer. The driver buffer of b_j is the buffer

Algorithm 8 Dynamic programming clock tree optimization algorithm for EMI

Require:

An initial T; initial buffer positions P, buffer sizes S, buffer library L; slew constraints S_{max}, S_{min}

Ensure:

Optimize skew and tree power while satisfying the slew constraints.

1: Segment tree, update V, E;
2: **repeat**
3: Find a critical path with maximum or minimum delay;
4: Relocate and size the critical path;
5: Analyze tree, update delay;
6: **until** Skew is not improved

inserted at node v_8 in our example. If there is no merging node between the driver buffer and v_i, we set the end_node to be the node where the driver buffer is inserted. Otherwise, if there is a merging node between them, we set the end_node to be the merging node. In this case, end_node is node v_7. All the nodes between these two nodes are candidates for the new buffer position. We check each node between $start\_node$ and end_node from step 5 to step 11 in Algorithm 9.

From step 6 to step 10 in Algorithm 9 we try each different buffer size l_k in the buffer library L and update the buffer load at each candidate node $v_{checking}$. out_cap represents the buffer capacitance load of b_j when it is inserted at $v_{checking}$. in_cap is defined as the capacitance from end_node to the current checking node $v_{checking}$. In order to satisfy

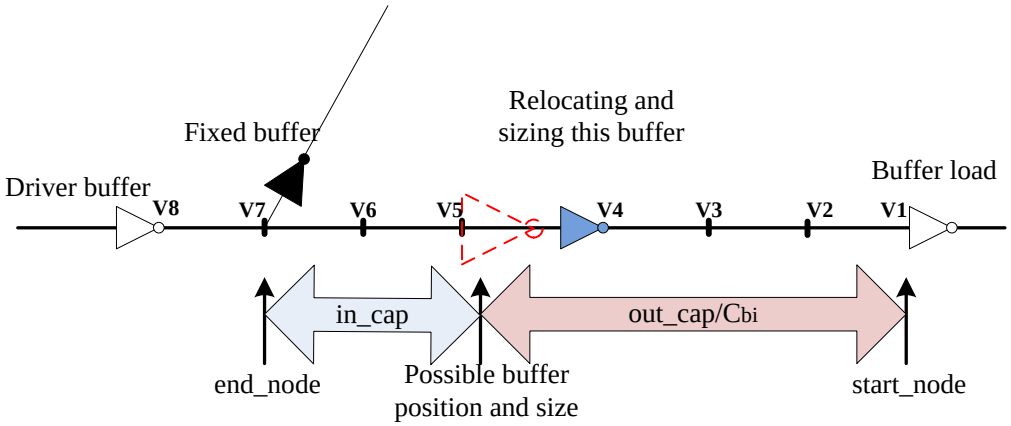


Figure A.4: $load\_cap$ and in_cap constraints

the slew constraints, we check the two capacitance constraints:

$$out\_cap : C_{l_k}^L \leq out\_cap \leq C_{l_k}^H \tag{A.5}$$

$$in\_cap : in\_cap \leq C^H \tag{A.6}$$

Equation A.5 controls the minimum and maximum slew rate of the buffer. Equation A.6 is checked only when the driver buffer drives a merging node. If the capacitance can not be driven by a buffer inserted right after the merging node, the solution is infeasible. In Figure A.4, the dotted buffer is a candidate solution for buffer at node v_4 if both capacitance constraints are satisfied.

When a merging node is visited, we will generate all solutions according to its two branches and perform a pruning which is detailed in the next section. This is from step 13 to step 16 in Algorithm 9.

A.3.4 Pruning

Pruning is a very important procedure in dynamic programming. Without pruning, the number of solutions will increase exponentially. As previously shown in Figure A.3, it is very difficult to decide whether the solution should be pruned or not without later delay information.

In our algorithm, however, only the buffers on the critical path are movable. When a merging node v_M is visited, the maximum/minimum delay of v_M is the maximum/minimum delay of both branches. D^M/d^M is used to represent the maximum/minimum delay of node v_M in the original tree. We will use D^M/d^M as reference in solution pruning.

A rough pruning is based on the global skew. If the skew of the solution i, $D_i^M - d_i^M$, is worse than the original whole clock tree skew, $D^M - d^M$, this solution is pruned.

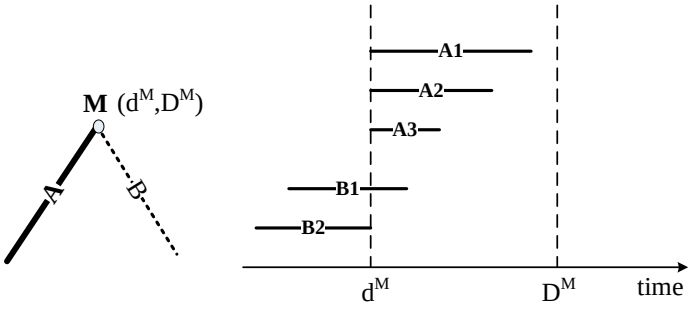


Figure A.5: Interval solution pruning based on D^M/d^M

After this, we do another more accurate pruning. Figure A.5 is used to illustrate the pruning procedure in **maximum** path optimization. In Figure A.5, path A and B merge at node M. We assume path A is the maximum delay path. The maximum delay D_M is determined by path A delay while minimum delay d^M is determined by path B which is

non-critical. Assume that there are N solutions at node M before pruning. In our example, there are five solutions: $A1$, $A2$, $A3$, $B1$ and $B2$. The left endpoint of each horizontal line in Figure A.5 represents the minimum delay d_i^M and the right endpoint represents the maximum delay D_i^M of solution i.

By examining all the solutions at node M, we can classify the solutions into two categories: $d_j^M = d^M$ and $d_j^M < d^M$. If $d_j^M = d^M$, as solutions $A1$, $A2$ and $A3$ in Figure A.5, the minimum delay at M is determined by path B which is same as the original tree. In this case, we only need to keep the solution which has the minimum delay of path A. $A1$ and $A2$ are pruned. If $d_j^M < d^M$, the minimum delay of the solution is determined by the delay of path A. In this case, the delay of path A is decreased which is what we want. We will keep all these solutions unless the skew is worse than global skew. In Figure A.5, the final solutions are $A3$, $B1$ and $B2$.

A.3.5 Infeasible Slew Limits

Clock tree synthesis tools often keep the buffer depth of each sink to root path same for skew and variation purposes. In Algorithm 9, we are trying to control the buffer load in the range $[C^L, C^H]$. It might be too strict to find a solution if additional buffers are inserted in the original path to keep same buffer depth. In this case, we have to compromise the solution quality by adjusting C_L to allow a feasible solution. We slowly decrease the C^L until we find a solution. This is acceptable because radiation power is cumulative effect and a small violation is not detrimental whereas maximum slew violations can result in noise susceptibility and logical errors.

A.4 Experimental Setup and Results

A.4.1 Experimental Setup

Our setup uses 45nm technology parameters. In our experiments, we consider four different buffer sizes. The initial buffered clock tree in our experiments are generated by [72]. The minimum and maximum slew, S_{min} and S_{max}, are set to 50ps and 100ps, respectively. We implemented our dynamic programming optimization algorithm in C++ and ran all experiments on a CentOS 5.3 system with 2.6GHz CPU and 8GB memory. We chose the ISCAS benchmarks instead of the ISPD benchmarks because they include both regional and local clock distribution. The ISPD benchmarks only consider regional distribution down to local clock buffers.

A.4.2 Skew and Power

Table A.2 lists the results of the initial buffered tree and the optimized buffered tree with our algorithm. The results are normalized to the initial tree for easier comparison. With buffer relocating and sizing, both skew and slew are improved while the total clock tree power increases by only 1%. The largest benchmark s35932 is able to complete in four minutes. The average number of iterations for each benchmark is about 10. The CPU time of our algorithm is very fast. This is because of the efficient pruning in the single path relocation and sizing. Since there is no other clock synthesis tools that consider the EMC problem, we cannot compare our results with other methods.

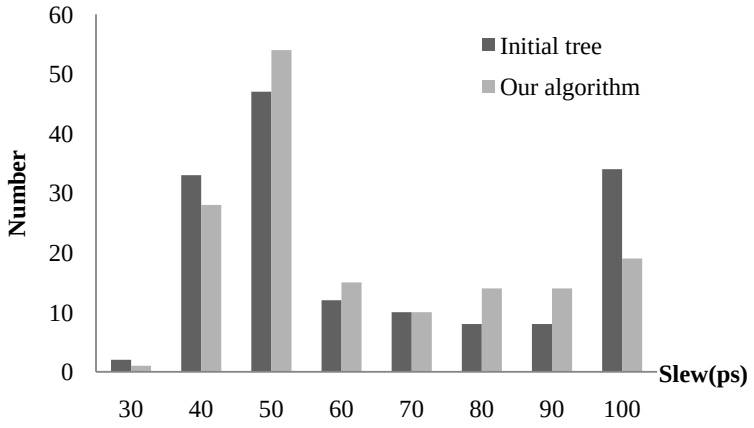


Figure A.6: Slew distribution of benchmark s15850

A.4.3 Slew Distribution

One contribution of our algorithm is the minimum slew control. Table A.2 lists the percentage of slews which are less than the minimum slew rate constraint S_{min}. Compared with the original clock tree, the average percentage is reduced by 12%. Figure A.6 is the slew distribution of benchmark s15850. We find there are several very small slew rates in the original tree. After optimization with our method, more slews are distributed in the $S_{min} = 50ps$ and $S_{max} = 100ps$ range. There are still some slews that are less than S_{min}, however. This is because of over-buffered paths which forced the C_L to be dynamically adjusted as discussed in Section A.3.5. The trend, however, is that more buffers are driving larger loads and that the total number of slew between 50ps and 100ps are increased.

Algorithm 9 Relocate and size one path

Require:

Critical $Path = V_p \cup E_p \cup B_p$; V_p, E_p and B_p are the nodes,edges and buffers on the $path$; slew constraints S_{max}, S_{min}.

Ensure:

Minimize skew and power, satisfy the slew constraints.

1: Sort V_p in a bottom up order;
2: **for** Each $v_i \in V_p$ bottom up **do**
3: **if** A buffer b_j is inserted at node v_i **then**
4: Find $start\_node$,end_node of b_j;
5: **for** Each node $v_{checking} \in [start\_node, end\_node]$ **do**
6: **for** Each buffer $l_k \in L$ **do**
7: Check out_cap constraint;
8: Check in_cap constraint if driver buffer drivers a merging node;
9: Add $v_{checking}$, l_k as a possible solution for b_j if constraints are satisfied;
10: **end for**
11: **end for**
12: **end if**
13: **if** v_i is a merging node **then**
14: Pack the two branches, generate all possible solutions at node v_i;
15: Do pruning ;
16: **end if**
17: **end for**

Table A.2: Benchmark simulation results

Benchmark		Initial tree				Our method					
Name	Sink	Power	Radiation	Skew	Slew$< 50ps$ (%)	Power	Radiation	Skew	Slew$< 50ps$ (%)	CPU (s)	Iteration (#)
s5378	179	1.0	1.00	1.00	44	1.02	0.99	0.57	33	0.7	10
s9234	211	1.0	1.00	1.00	43	1.01	0.95	0.49	31	1.5	10
s13207	638	1.0	1.00	1.00	50	1.00	0.88	0.74	30	9.9	9
s15850	534	1.0	1.00	1.00	38	1.00	0.88	0.78	28	5.9	4
s35932	1728	1.0	1.00	1.00	38	1.00	0.94	0.63	32	203.2	19
s38584	1426	1.0	1.00	1.00	64	1.00	0.98	0.81	48	90.7	8
Average	786	1.0	1.00	1.00	46	1.01	0.94	0.67	34	52.0	10

A.4.4 Radiation Power

Equation A.4 is used to estimate the radiation power in Table A.2. Due to the large run-time, however, radiation power is not used as a criterion in our pruning procedure. By increasing minimum slew rates without sacrificing the power or skew of the tree, our algorithm reduces the overall radiation of the clock tree in every case. Most of this radiation power is removed from the very high frequencies as we illustrated in Section **??**. If we want to emphasize the importance of radiation power in certain frequency bands (e.g. GSM or CDMA), we can call SPICE for accurate current measurement and add radiation power as a criterion in the pruning.

A.4.5 Segment Size

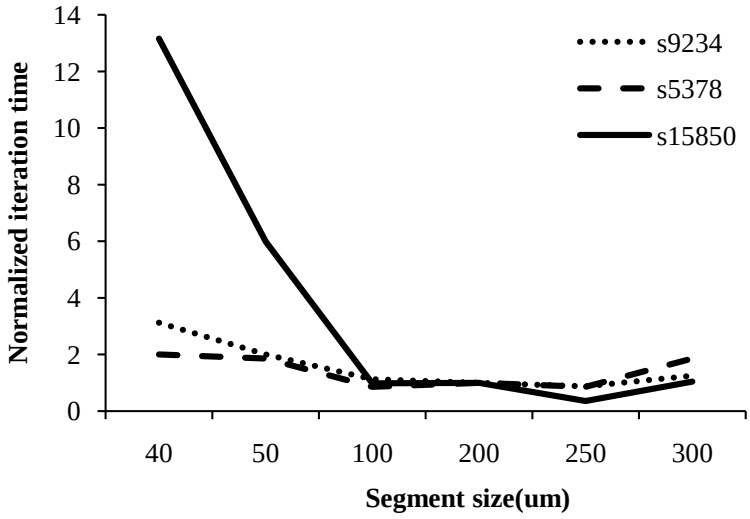


Figure A.7: Segment size vs iteration run time

In our algorithm, the first step is to segment the tree. The segment size is related to the CPU time. If the tree is segmented with a small size, there will be more candidate nodes

for buffer locations and better results for skew and slew. However, too many candidate nodes will significantly increase the CPU time. Figure A.7 is the average iteration time with different segment sizes. The iteration time is normalized to the time when segment size is $200\mu m$. The run time significantly increases when segment size is less than $50\mu m$. When segment size is greater than $300\mu m$, the optimization results are not satisfactory. In the previous experiments, the segment size was set to $200\mu m$.

A.5 Conclusion

In conclusion, this is the first clock buffer optimization algorithm for Electromagnetic Compatible clock synthesis. By controlling the minimum slew rate, we can reduce the high-frequency spectral noise generated by the clock tree. All traditional clock tree metrics, power and skew, are also considered in our algorithm. This is also the first work to use an incremental dynamic programming in clock tree synthesis and can efficiently optimize the clock tree in very short time. Our method can be used to optimize any buffered clock tree for EMC design with more accurate delay models or SPICE simulation.