Efficient Inductance Extraction for On-chip Interconnect

by

Clement Luk

A technical report submitted to

Department of Electrical and Computer Engineering

in partial fulfillment for requirements of

Master Of Science

at the

UNIVERSITY OF WISCONSIN-MADISON

May 2003

Abstract

This technical report describes the efficient inductance and resistance extraction methodology using the reluctance element (inverse inductance). Reluctance exhibits locality and hence has a greater potential for sparsification. A window search algorithm is developed to handle irregular VLSI geometries. In order to handle high frequency effect namely, skin and proximity effect, the extraction algorithm is revised to extract the frequency dependent reluctance and resistance values. Results from examples are given to demonstrate the efficiency and accuracy of this methodology can be realized in realistic integrated circuit interconnect extraction.

Academic Advisor: Chen, Charlie Chung-Ping

Title: Assistant Professor, Department of Electrical and Computer Engineering

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The phenomenal progression of process integration in semiconductor industry driven by Moor's Law has been providing designer with more and more silicon estate to work with. (See figure 1.2 and table 1.1 for technology trend in scaling and integration.) Designer can now integrate more sophisticated logic, functionality, and caches into a single silicon chip. However, as we move toward the deep-sub-micron era, signal integrity issue arises, and has been an impediment of design closure.

Modeling of interconnects used to be a negligible effect but now becomes a major issue in signal integrity. Traditionally, gate delay has been a dominating factor in the delay model and interconnect plays an insignificant part. But with the introduction of more metal layer and reduction of feature size, "interconnect effect" becomes an indispensable issue for circuit designer to tackle. Figure 1.1 shows a delay comparison for with and without repeater inserted. As we could see, for the global interconnect without the repeater inserted, there is more than ten times difference in delay compare to no repeater inserted. This demonstrates how "interconnect effect" impacts designer for design closure.

Signal Integrity engineering has become a necessary requirement for today's high-speed logic design. Various design issues arise from signal integrity engineering: crosstalk, ground bounce, ringing from inductance coupling, noise margins, impedance matching, and decoupling is now critical to a successful design. But with operating frequency approaches multi giga-hertz, correct and efficient modeling of on-chip inductance for VLSI circuit becomes an indispensable issue. Inductance effect is not only as a result of faster switching frequency but also of many other factors: reduction of resistance by copper and capacitance by low-κ dielectric, denser geometries, and more metal layers. (See table 1.1 for technology trend.) The challenge of inductance modeling and analysis were discussed in [2] and [3].

Inductance modeling has been posed as a challenging issue in VLSI circuit due to its long-range effect and difficulties in predetermined return path. With the introduction of partial inductance under the partial element equivalent circuit (PEEC) model [4], it solved the return path problem with the assumption of return path at infinity. However, because of PEEC model, it also leads to a dense L matrix since partial inductance needs to define coupling term with each element. Although far-away term maybe small, truncation of them may lead to an unstable system and lose its desirable positive definite property [5].

Various techniques have been introduced to alleviate the problem. Shift and truncate method was proposed by Krauter [6]. This method assumes that the return path is no longer at infinity but within a shell. Other method such as the Halo method [7]and the block diagonal method [8]also reduce the number of mutual inductances by limiting the return path to nearest power and ground returns. FastHenry [9] speeds up the extraction process by multi-pole expansion. A. Pacelli [10] proposed to model circuit as a vector-potential equivalent circuit (VPEC) by magnetic resistance and vector-potential controlled voltage source. A interconnect modeling technique called wire duplication [11]has also been proposed.

Among the different sparsification techniques, a method called reluctance, which also known as K-method or susceptance, has been proposed by Hao Ji [12], [13]. The reluctance matrix or K matrix is defined as the inverse of the L matrix, and it exhibits high degree of locality similar to capacitance, and hence has a greater potential for sparsification. [12] shows that K matrix is diagonally dominant and hence positive definite. The off diagonal terms are negative and can be safely deleted without sacrificing stability.

However, there are several issues of existing inductance and reluctance extraction. Firstly, we show via a counter example that diagonally dominant may not be achieved if there are irregular geometries. Based on this observation, we develop a window-based search algorithm that could handle irregular geometries and also develop a recursive-bisection-cutting algorithm to guarantee stability. [14] Thirdly, existing reluctance extraction techniques did not consider any high frequency effect, namely skin and proximity effect. We need to investigate how to integrate existing techniques using reluctance method to capture skin and proximity effect. And, we may need to exploit methodology to efficiently handle increased number of filaments created to capture the uneven distribution of current. Furthermore, we extend our tool to handle ground plane as input and also develop a parallel extraction algorithm for handling large-scale extraction.

Year	2001	2002	2003	2004	2005	2006	2007
bulk dielectric constant (κ)	<2.7	<2.7	<2.7	<2.4	<2.4	<2.4	<2.1
DRAM 1/2 Pitch (nm)	130	115	100	90	90	70	65
MPU/ASIC 1/2 Pitch (nm)	150	130	107	90	80	70	65
(Gate delay)NMOS driver γ ($C_{gate} \times V_{dd}/I_{dd}$(ps))	1.6	1.3	1.1	0.99	0.83	0.76	0.68
Local interconnect RC delay (ps/mm)	86	121	176	198	256	303	342
Global interconnect RC delay (ps/mm)	21	29	40	37	59	74	79
Total interconnect length (m/cm^2)	4086	4843	5788	6879	9068	10022	11169

Table 1.1: International Technology Roadmap for Semiconductor (ITRS) 2002 [1]: technology trend

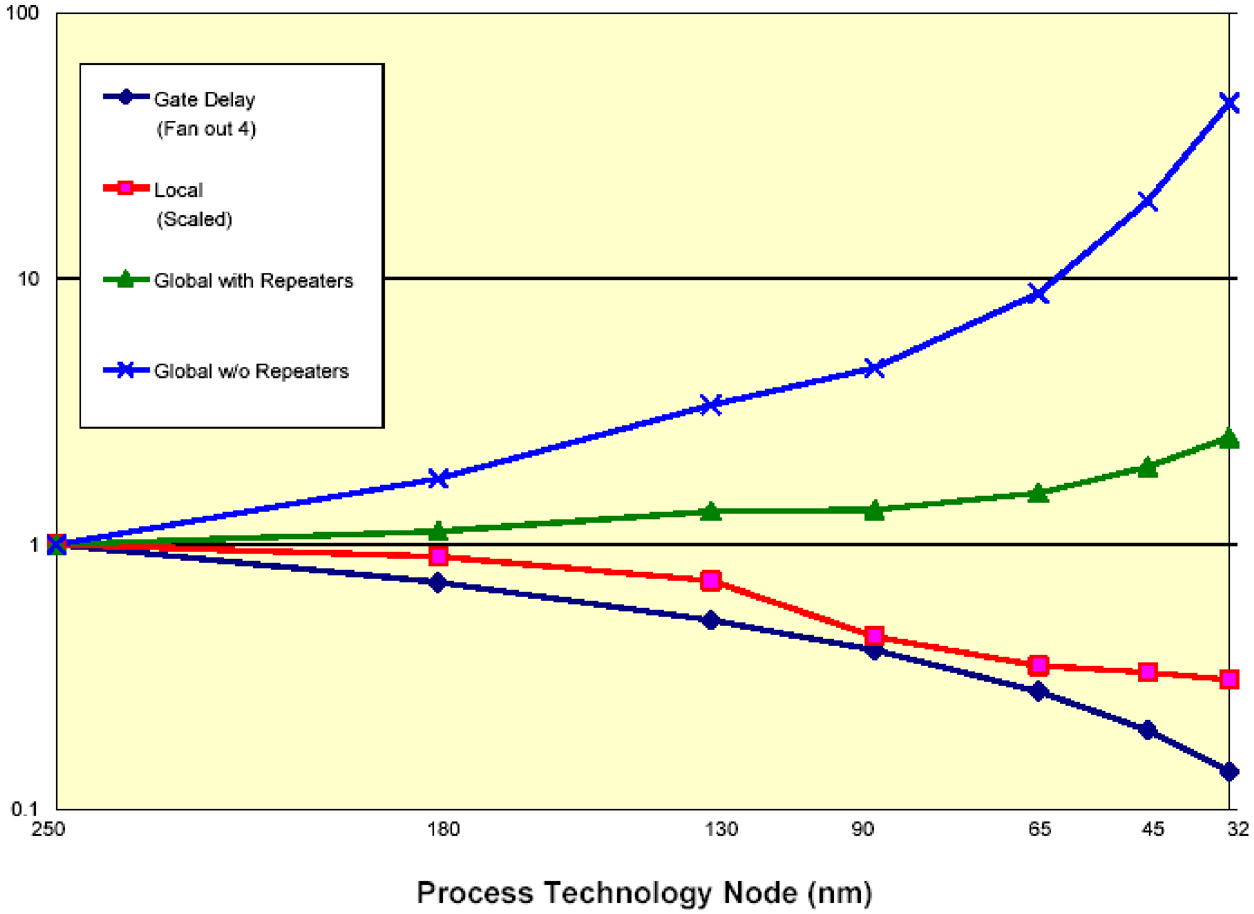


Figure 1.1: Delay comparison: with and without repeater inserted [1]

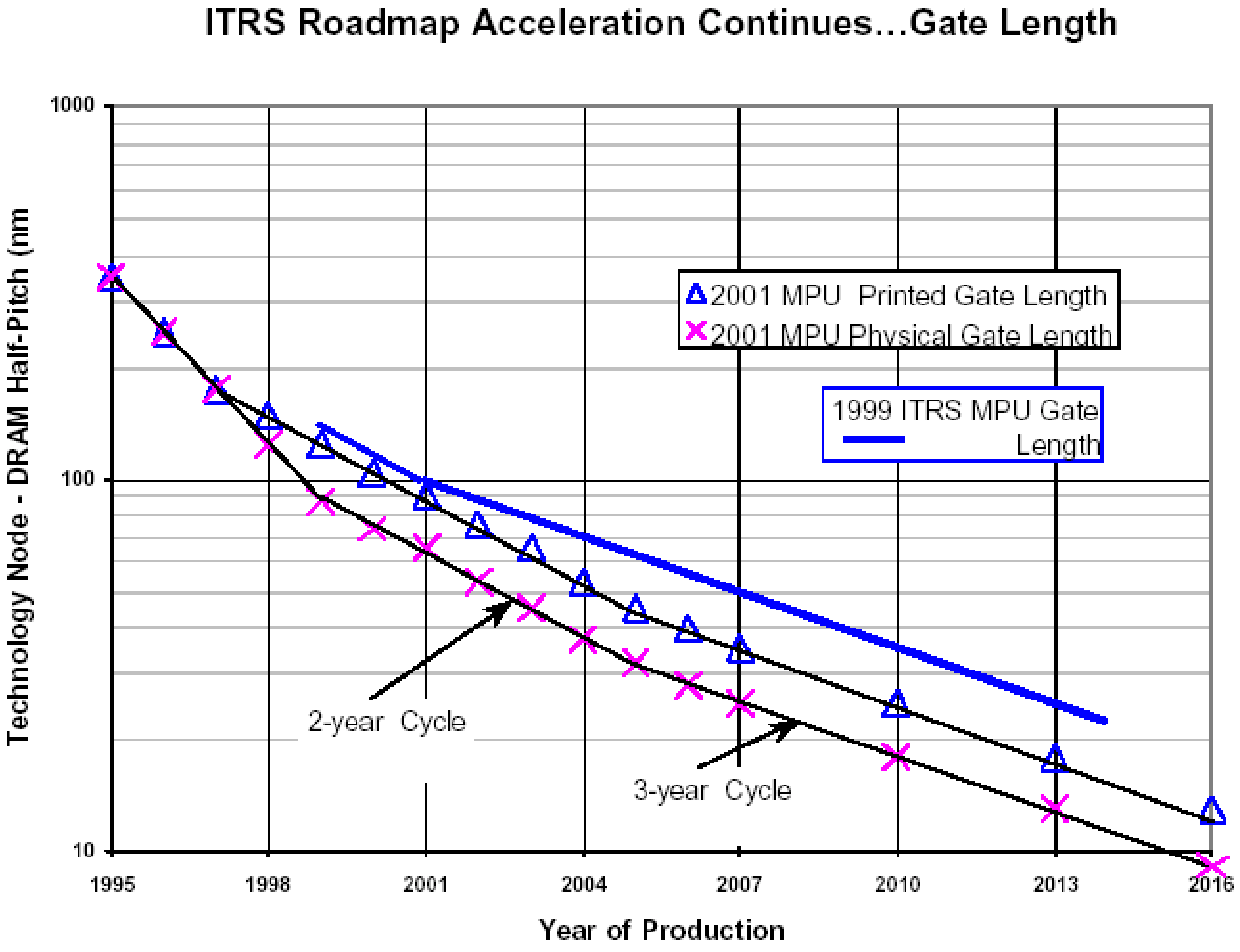


Figure 1.2: Gate length trend extracted from ITRS [1]

Chapter 2

Inductance Extraction

2.1 Inductance and Reluctance Matrix

Partial inductance is defined as flux per current.

$$\mathcal{L}_{ij} = \frac{\Phi_{ij}}{I_i} \tag{2.1}$$

$$\mathcal{L}_{ij} = \frac{1}{I_i}\int_{s_j} B_i \cdot ds_j \tag{2.2}$$

From Maxwell's equations, we know that $\nabla \cdot \vec{B} = 0$. And for a divergence free field, we could express it as the curl of another vector field. Therefore, we could define a vector field $\vec{A}$ (known as magnetic vector potential) such that $\vec{B} = \nabla \times \vec{A}$. Replacing $\vec{B}$ by the curl of $\vec{A}$ and utilizing Stoke's theorem, we continue the inductance formulation.

$$\mathcal{L}_{ij} = \frac{1}{I_i}\int_{s_j} (\nabla \times A_{ij}) \cdot ds_j \tag{2.3}$$

$$\mathcal{L}_{ij} = \frac{1}{I_i}\int_{l_j} A_{ij} \cdot dl_j \tag{2.4}$$

The magnetic vector potential along segment j induced by current I_i in segment i is defined as:

$$A_{ij} = \frac{\mu_0}{4\pi}\int_{li} \frac{Ii}{r_{ij}} dl_i \tag{2.5}$$

Combining (2.5) and (2.4) would give partial inductance formulation as:

$$\mathcal{L}_{ij} = \frac{\mu_0}{4\pi a_i a_j}\int_{a_i}\int_{a_j}\int_{l_i}\int_{l_j}\frac{dl_i \cdot dl_j}{r_{ij}}da_i da_j \tag{2.6}$$

Partial inductance can be calculated using integral form as published [15] or in a reduced form using Taylor series expansion in [16].

From this we could form $n \times n$ a partial inductance matrix by writing the corresponding linear system equations as:

$$\begin{bmatrix} \mathcal{L}_{11} & \mathcal{L}_{12} & \cdots \\ \mathcal{L}_{21} & \mathcal{L}_{22} & \cdots \\ \cdots & \cdots & \mathcal{L}_{nn} \end{bmatrix} \begin{bmatrix} i_1 \\ : \\ i_n \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ : \\ \Phi_n \end{bmatrix} \tag{2.7}$$

And the inverse of the system, which is also known as $\mathcal{K}$ matrix can be written as:

$$\begin{bmatrix} \mathcal{K}_{11} & \mathcal{K}_{12} & \cdots \\ \mathcal{K}_{21} & \mathcal{K}_{22} & \cdots \\ \cdots & \cdots & \mathcal{K}_{nn} \end{bmatrix} \begin{bmatrix} \Phi_1 \\ : \\ \Phi_n \end{bmatrix} = \begin{bmatrix} i_1 \\ : \\ i_n \end{bmatrix} \tag{2.8}$$

2.1.1 Locality Property

The following is a small example of a 5 conductors system to illustrate the idea of how off-diagonal value decreases much faster in L matrix than in $\mathcal{K}$ matrix. The parameter for this example is as follows: width for each conductor is 5e-6, spacing are 1e-6, length is 1000e-6, thickness is 3.6e-7, sigma is 4.996e7.

The extracted value of L matrix is as follows:

$$\mathcal{L} = \begin{bmatrix} 1.28 & 0.977 & 0.83 & 0.74 & 0.69 \\ 0.977 & 1.28 & 0.977 & 0.83 & 0.747 \\ 0.83 & 0.977 & 1.28 & 0.976 & 0.829 \\ 0.74 & 0.83 & 0.976 & 1.28 & 0.976 \\ 0.69 & 0.747 & 0.829 & 0.976 & 1.28 \end{bmatrix} \times 10^{-9}H\ . \tag{2.9}$$

The extracted value of K matrix is as follows:

$$\mathcal{K} = \begin{bmatrix} 1.933 & -1.18 & -0.148 & -0.129 & -0.157 \\ -1.18 & 2.64 & -1.10 & -0.081 & -0.129 \\ -0.148 & -1.10 & 2.65 & -1.10 & -0.148 \\ -0.129 & -0.081 & -1.10 & 2.64 & -1.18 \\ -0.157 & -0.129 & -0.148 & -1.18 & 1.93 \end{bmatrix} \times 10^9 H^{-1} \ . \tag{2.10}$$

As we could see, the off-diagonal term L_{51}in L matrix is about 53% of the self partial inductance or $\mid \frac{L_{51}}{L_{11}} \mid$. However for the off-diagonal term of K_{51} in $\mathcal{K}$ matrix is only 8.1% of the self reluctance value or $\mid \frac{K_{51}}{K_{11}} \mid$. The reason of this rapid decrease of off-diagonal term value can be understand from physical meaning of reluctance matrix. We will elaborate on this below.

To better understand the idea of locality in $\mathcal{K}$ matrix, we could begin from exploiting the physical meaning of the $\mathcal{K}$ matrix. From the system of equations of (2.8), the physical meaning of $\mathcal{K}_{ij}$ can be explained as: *The magnetic flux along all conductors is set to zero except the j^{th} conductor which is set to one. The induced current in i^{th} conductor is the entry of $\mathcal{K}_{ij}$.*

Therefore, to construct a $\mathcal{K}$ matrix, we could activate the j^{th} conductor by setting flux equals to 1. And the induced current flowing in each conductor corresponds to j^{th} column of $\mathcal{K}$ matrix.

Based on this notion, we could understand the locality property in terms of induced current in each conductor: For example, the j^{th} conductor is activated by setting flux to one. This accounts for carrying a positive current in the activated line. From Lenz's Law, we know that the induced emf will cause a current to flow in a direction such that it will oppose the change in the linking of magnetic flux. As a result, the immediate neighbor conductor, $j+1^{th}$, will carry a current that will create a counter flux. At the same time, the $j+1^{th}$ conductor's magnetic flux will also induce current in the $j+2^{th}$ conductor. Consequently, the induced current by $j+1^{th}$ conductor will mitigate the induced current by j^{th} conductor in $j+2^{th}$ conductor. As depicted in figure 2.1, the $j+2^{th}$ conductor results in a shorter arrow accounting for the overall effect.

Because of this rippling effect of magnetic field and induced current, the value of each element of $\mathcal{K}_{ij}$ is accounted for the overall effect rather than a single active line. Hence, the far away terms of $\mathcal{K}$ matrix would decrease more rapidly than $\mathcal{L}$ matrix, and thus it exhibits locality and shielding effect [[13],[12], [14], [17]].

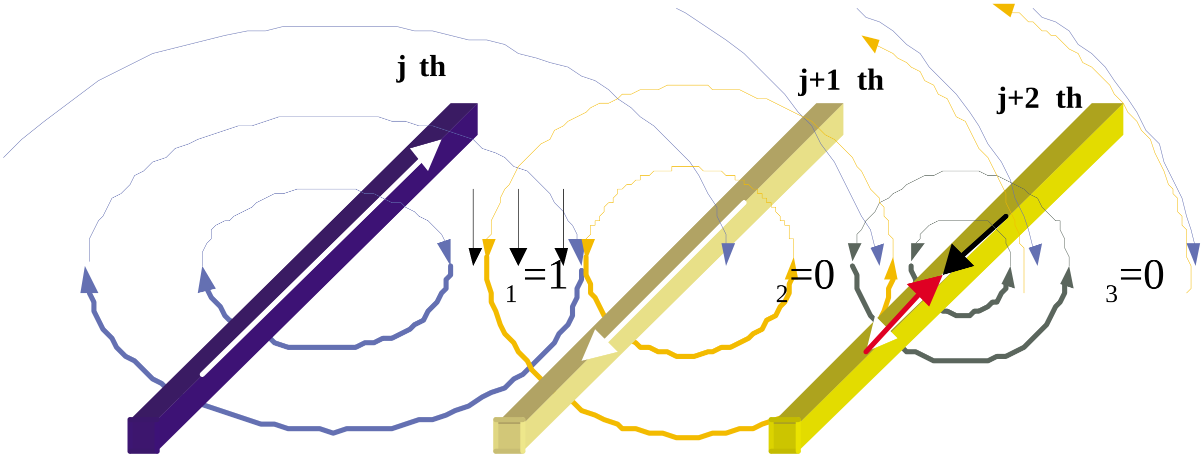


Figure 2.1: An example to explain the locality property

2.1.2 Stability Issue of Reluctance method

H. Ji, et. al [13] developed an advanced reluctance sparsification method called K-method. The proof of stability of their algorithm is based on the diagonal dominance property. This is derived from the assumption that all off-diagonal terms of $\mathcal{K}$ are negative. However, this condition may not be true under an irregular geometries. We would show a counter example below to illustrate this idea.

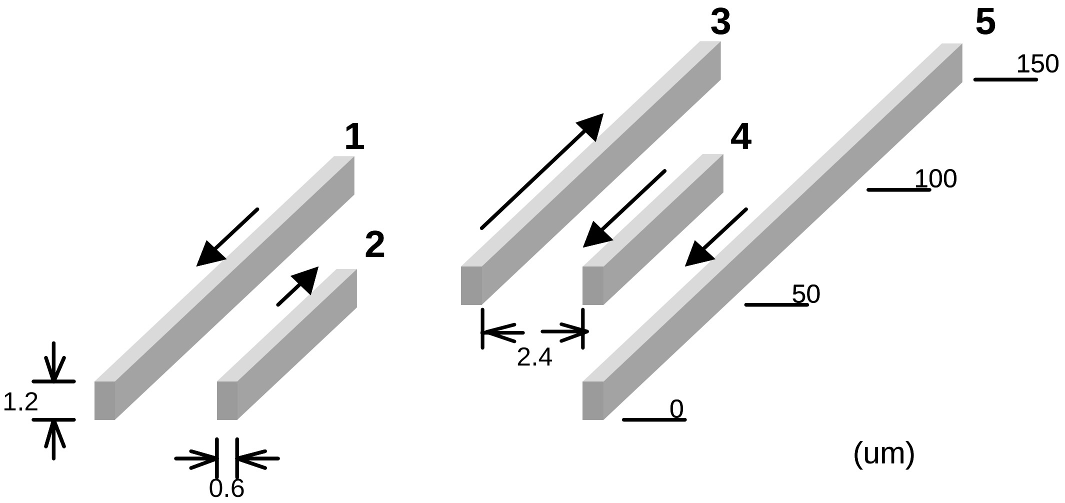


Figure 2.2: An example for parallel conductors with unequal lengths

In this example, we calculate the partial inductance matrix, and get

$$\mathcal{L} = \begin{bmatrix} 1.04 & 0.34 & 0.37 & 0.24 & 0.51 \\ 0.34 & 0.45 & 0.09 & 0.06 & 0.27 \\ 0.37 & 0.09 & 1.04 & 0.34 & 0.41 \\ 0.24 & 0.06 & 0.34 & 0.45 & 0.11 \\ 0.51 & 0.27 & 0.41 & 0.11 & 1.69 \end{bmatrix} \times 10^{-10} H \ . \tag{2.11}$$

Inverting $\mathcal{L}$, the $\mathcal{K}$ matrix can be obtained as follows,

$$\mathcal{K} = \left[\begin{array}{rrrrr} 1.57 & -0.94 & -0.22 & -0.47 & -0.25 \\ -0.94 & 3.02 & 0.15 & 0.01 & -0.23 \\ -0.22 & 0.15 & 1.42 & -0.93 & -0.24 \\ \hline -0.47 & 0.01 & -0.93 & 3.12 & 0.16 \\ -0.25 & -0.23 & -0.24 & 0.16 & 0.75 \end{array}\right] \times 10^{10} H^{-1} \ . \tag{2.12}$$

It is clear to see that some of the off-diagonal terms are positive in (2.12). For instance, when calculating the 3^{rd} row in (2.12), a unit flux is assigned on conductor 3, which demands a positive current along conductor 3 to accomplish. This current induces a positive magnetic flux along all other conductors (let's consider only conductors 1 and

2 in this explanation). To compensate this effect and make the net magnetic flux along conductor 1 and 2 equals to zero, they have to carry a negative current. However, the induced current along conductor 1 also induces another current along conductor 2. Since the coupling effect between 1 and 2 is much stronger than the effect between 3 and 2, the overall effect causes conductor 2 to carry a positive current direction.

Since the result is due to the overall effect (not a signal active line), negative off-diagonal elements are not guaranteed anymore. This invalids the proof of the diagonal dominance property in [13] and hence the stability for the K-method becomes unsure. We will propose our remedies in the following section.

2.1.3 Formal Analysis

In this section, we first present the K-method stability analysis through the duality of electrical and magnetic fields at Maxwell's equations. From the duality analysis, we shed the insights of the similarity of inductance and capacitance. Through the theorems provided in this section, we are able to propose a correction to the K-method to ensure the stability. From the Maxwell's equations, we have

$$\nabla \times \vec{E} = -\mu s \vec{H} \ , \tag{2.13}$$

$$\nabla \times \vec{H} = \epsilon s \vec{E} + \vec{J} \ , \tag{2.14}$$

$$\nabla \cdot \epsilon \vec{E} = \rho \ , \tag{2.15}$$

$$\nabla \cdot \mu \vec{H} = 0 \ . \tag{2.16}$$

The definition of the magnetic vector potential gives

$$\mu \vec{H} = \nabla \times \vec{A} \ . \tag{2.17}$$

Applying (2.17) to Equation (2.13), we get

$$\nabla \times (\vec{E} + s\vec{A}) = 0 \ . \tag{2.18}$$

This implies that there exists a scalar potential, V, such that

$$\vec{E} + s\vec{A} = -\nabla V \ . \tag{2.19}$$

To uniquely determine $\vec{A}$, we choose the Lorentz gauge,

$$\nabla \cdot \vec{A} = -\epsilon \mu s V \quad . \tag{2.20}$$

By Equations (2.14), (2.17), (2.20), and the identity $\nabla \times (\nabla \times \vec{A}) = \nabla(\nabla \cdot \vec{A}) - \nabla^2 \vec{A}$, we can get

$$\nabla^2 \vec{A} - \mu\epsilon s^2 \vec{A} \quad = \quad -\mu \vec{J}. \tag{2.21}$$

Similarly, by Equation (2.15), (2.18), and the Coulomb gauge, we get

$$\nabla^2 V - \mu\epsilon s^2 V \quad = \quad -\rho/\epsilon. \tag{2.22}$$

Equations (2.21) and (2.22) are often referred as the **nonhomogeneous Helmholtz's equations**. The solutions of Equations (2.21) and (2.22) are

$$\vec{A}(r) \quad = \quad \frac{\mu}{4\pi} \int_{V'} G(r, r') \vec{J}(r') e^{s/c|r-r'|} dv' \quad , \tag{2.23}$$

$$V(r) \quad = \quad \frac{1}{4\pi\epsilon} \int_{V'} G(r, r') \rho(r') e^{s/c|r-r'|} dv' \quad , \tag{2.24}$$

in which V' is the volume of all conductors, $c = 1/\sqrt{\mu\epsilon}$, and $G(r, r') = \frac{1}{4\pi R_{ij}}$, where $R_{ij} = |r - r'|$ is the Green's function. The dual property between a magnetic and an electric problems can be observed from Equations (2.21)(2.22)(2.23) and (2.24). The major difference between $\vec{A}$ and V is that $\vec{A}$ is a directional vector and V is a scalar. There exists a transformation from a magnetic problem to a electric problem, which is described in the following lemma.

Lemma 1 *Given an uni-directional magnetic non-homogeneous Helmholtz's equation problem, there exists a corresponding electric non-homogeneous Helmholtz's equation problem that has the same solution.*

Since all the magnetic sources and mediums are uni-directional, we can remove the vector natural by proper assigning the positive charge corresponding to the forward direction or negative otherwise. Hence, given current sources vector $\vec{J}$, we can create a corresponding charges $\rho = \mu\epsilon\vec{J}$ with proper signs assigned, and the solution of Equations (2.21) and (2.22) are identical.

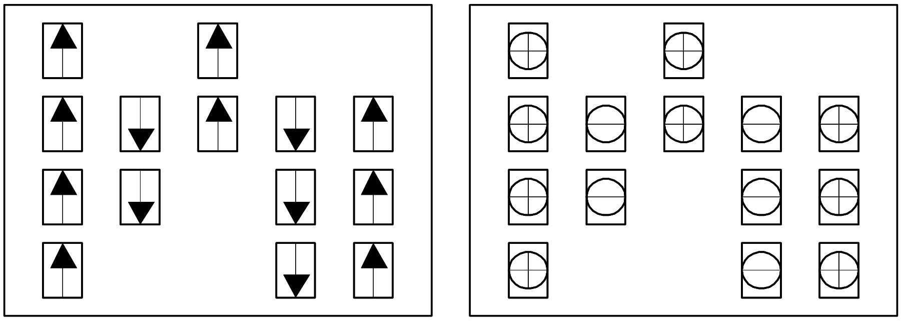
Figure 2.3 illustrates the transformation of Lemma 1. From this lemma, we can get the following theorem.

Theorem 1 *The reluctance matrix, $\mathcal{K}$, is diagonally dominant and symmetric positive definite when all the conductors are sufficiently discretized.*

PROOF:Lemma 1 shows that every uni-directional magnetic problem can be transformed into an electric one. Since it has been shown that the capacitance matrix is always diagonally dominant and symmetric positive definite for sufficiently discretized problem [4], Theorem 1 thus follows.

Theorem 1 reveals why the diagonal dominance property of the reluctance matrix does not always hold. The answer is **discretization**. When we perform capacitance extraction, conductors are usually well discretized. In the contrary, the conductors are always preserved as long wires while we perform inductance extraction. The length of inductance discretization is often hundred times larger than the capacitance discretization. Therefore, we come up with the remedy in the following subsection.

Figure Figure Figure F Figure Figure Figure F



(a) Inductance model

(b) Capacitance model

Figure 2.3: Dual property between inductance and capacitance problems

2.1.4 Recursive Bisection Cutting Algorithm

We have already shown that finer discretization guarantees the stability of the K-method. It can also increases the accuracy. However, if we uniformly discretize conductors into small pieces, the complexity of solving this problem will become enormous, which losses the original intention of the sparsification. Therefore, we propose a cutting algorithm to obtain a stable reluctance matrix without too much prejudice to the simulation run time. This algorithm is called Recursive Bisection Cutting Algorithm (RBCA), which is based on the idea that the reluctance matrix is diagonally dominant and symmetric and positive definite (s.p.d.) when the conductors are sufficiently discretized as in capacitance problem.

From the previous discussion, the diagonal dominance of $\mathcal{K}$ is strongly related to unequal-length and misalignment cases. Theorem 1 also tells that smaller discretization is better than larger one, which implies that longer conductor actually acts a critical role in this problem. Thus, the basic idea of the BCA is to recursively cut the longest conductor when the positive off-diagonal element occurs during the K-method procedure. In order to make sure the RBCA result in all negative off-diagonal elements in $\mathcal{K}$, we perform the K-method with a small window (will discuss how to choose this window in the later section) and check if every off-diagonal element in the small K-matrix is negative. If there exists any positive off-diagonal value, we cut the longest conductor in this window. After this cutting, back-trace those conductors that are reluctance-coupled with this cut one. If this cutting causes new positive off-diagonal value to any previously processed conductor, we recursively cut all the troublemaker conductors. Iteratively repeat this until the final K-matrix has all positive off-diagonal entries. The RBCA is summarized as in Table 2.1.

Theorem 2 *The RBCA can guarantee the all non-positive off-diagonal elements in the K-matrix, and hence the s.p.d. property validates the proof of stability in [13].*

PROOF: Since the RBCA recursively check the off-diagonal values during the ex-

```
Input: Given a parallel-conductor system
For each conductor $j$
  1. Choose a window $W$.
  2. Calculate $\mathcal{K}_{ij}$, where $i \in W$, as in the K-method.
  3. If $\exists \mathcal{K}_{kj} > 0$, where $k \in W$ and $k \neq j$
     Then
       a. cut the longest conductor $l$, $l \in W$, by half.
       b. Back-trace and reperform the K-method for
          the $i^{th}$ conductor, where $\mathcal{K}_{ki} \neq 0$, $i \in W$ and
          $i < j$. If this cutting causes any new positive
          off-diagonal terms, recursively perform the
          cutting to the troublemaker conductor.
     EndIf
EndFor
```

Table 2.1: Recursive Bisection Cutting Algorithm (RBCA)

traction process, and also by Theorem 1, Theorem 2 thus sustains.

As we know that positive off-diagonal values happen when conductors have seriously mismatched length or mis-aligned organization. Since all of the previous works [13][**?**][17] considered only equal-length parallel conductors, the case that we show do not exist. However, to build a full-chip inductance (reluctance) extractor, this possibility does indeed exists. Since lack of negative mutual reluctance value losses the stability guarantee for the K-method, we propose this cutting algorithm serving as the stability guard of the extractor to insure the s.p.d. of the sparse reluctance matrix. In the next section, we will show how to select the window in general for circuit configurations.

2.2 Window Search Algorithm

All previous works used equal-length parallel conductors as their examples to show the benefits of sparsification on reluctance matrices. It is still not clear if the K-method can work on general irregular geometries. The lack of generality limits the application of the K-method only to analysis of some special configurations such as buses. However, general routing cases are more irregular, which might contain uneven-length or misaligned conductors. For these cases, it is very difficult to determine what a "window" is when performing the window-based K-method. In this section, we propose a novel algorithm to determine what should be included in the window when we extract sparse reluctance matrices. Let us first define the terminology used in the following discussion by the example circuit in Figure 2.4.

Aggressors and Victims: When performing the K-method and calculating one of the columns in $\mathcal{K}$, we set the magnetic flux along the corresponding conductor to one, which we name as **aggressor**, and others to zero, which we call **victims**. In Figure 2.4, we assume that conductor 1 is now the aggressor, and other conductors are victims.

ESF, ESR and ESA: Supposed the aggressor has length L, we now define the **extended search factor (ESF)**, x, such that the **effective search range (ESR)** extends both ends of the aggressor out xL (i.e. segment $\overline{ah}$). The ESR is a strip with length $(1 + 2x)L$. Then the **effective search area (ESA)** is defined by sweeping from left infinity to right infinity with the ESR. The ESA is marked by slash lines in Figure 2.4.

Shields and Shielding Level: If a victim is partially or fully covered in the ESA, it is called a **shield** for the aggressor. For example, in Figure 2.4 conductors 2, 4, 5, and 6 are shields of 1, but 3 is not. The shielding level indicates how close the victim shields the aggressor. If there exists k shields between a shield and the aggressor, the shield is with the $(k+1)^{th}$ level. For example, conductor 2 is the 1^{st} level of shield for segment $\overline{ad}$, and conductor 4 is the 2^{nd} level of shield for $\overline{bd}$. Conductor 5 contains two part. The upper part is the 3^{nd} level of shield for $\overline{bd}$ and the lower part is the 1^{st} level of shield for $\overline{dg}$.

We now discuss how the K-method works. From the physical point of view, the experiment results in [13] demonstrates that the shielding effect for the mutual reluctance exist but not for partial mutual inductance. From the numerical point of view, the K-method selects the most significant values on a column (row) and inverts it. The inversion causes the off-diagonal values of the reluctance matrix decreasing in a rapider way than the inductance matrix. This makes the reluctance element has better locality than the partial inductance. Therefore, to properly include the relatively significant couplings within the small window actually dominates the accuracy of the sparse K approach.

However, there exists some difficulties for irregular geometries in K-method. First, the strength of coupling does not strictly decrease as the distant between conductors increase, so the closer one may not be the more significant one. This means that a conductor may have stronger coupling but is not included in the window. Second, an intuitive solution is to select the most significant inductive coupling conductors in the small window. To find the most significant value in the L-matrix, we have to extract all the partial mutual inductance values, which makes the extraction complexity $O(n^2)$ and losses the efficiency of the K-method. Moreover, one conductor may be the other's strongest coupling but the contrary does not necessary hold. In this case, the K-matrix becomes topologically asymmetric and leads the later-on symmetrization process to introduce more errors to the final sparse K-matrix.

In Equation (2.6), the inner product of $dl_i \cdot dl_j$ implies that the mutual inductance has a large value when two conductors are parallel and next to each other, and has a small value when they are mis-aligned. If two conductors are perpendicular, their partial mutual inductance is zero. Based on these observations and utilizing the shielding effect of reluctance elements, we select victim conductors by window selection algorithm (WSA) presented in the following section.

Window Selection Algorithm(WSA)

1. Divide all conductors into two sets, vertical and horizontal.
2. For the vertical set, sort all the conductors with their x coordinates.
3. For every conductor from left to right,
    (a) Search from the first victims right next to the aggressor, and select those shields (by definition) until all the points on the ESR (i.e. $\overline{ah}$ in Figure 2.4) are shielded no less than k levels. This forms the right part of the small window.
    (b) The left part of the window can be obtain from the previous iterations, since the result K-matrix is topologically symmetric.
    (c) Set the flux along the aggressor to one and others to zero, and solve the self and mutual reluctance elements as in the K-method.
4. repeat step 2 and 3 for the horizontal conductors.
5. Symmetrize by $\mathcal{K} = \frac{1}{2}(\mathcal{K}_{asym} + \mathcal{K}_{asym}^T)$

Notice that we select the shields until all the points on the ESR are shielded no less than k levels to ensure that we capture the significant effect. For example, if we set the level of shielding to 1 in Figure 2.4, the selected victims should be conductors 2, 5, and 6. Thus all the points on $\overline{ah}$ are shielded at least once.

In this algorithm, we only have to search the right-hand-side shields for each aggressor. The left-hand-side shields can be obtain from the previous results. It is obviously to see that A is B's n^{th} level of shield, then B must be also A's n^{th} level of shield. This observation allows us to use the previous evaluated mutual inductance values and the window information, which can save half of the extraction time. It also maintains the obtained K-matrix to be topologically symmetric. This would benefit the later-on symmetrization process.

2.3 Experimental Result

We develops our reluctance extractor in C/C++ programming language. The extractor implements our WSA-based K-method with RBCA stability guarantee. The simulations are run on an Intel Pentium IV 1.4 GHz system.

			Attacker		Faraway victim	
Shielding Level	ESF	Density	1^{st} peak error	2^{nd} peak error	1^{st} peak error	1^{st} droop error
1	0.0	4.0%	3.21%	-2.26%	-33.52%	-51.72%
1	0.5	4.6%	0.55%	0.62%	-26.46%	-37.70%
1	1.0	5.0%	0.52%	0.82%	-27.19%	-35.63%
2	0.5	8.1%	-0.33%	0.72%	-11.59%	-22.99%
3	0.5	11.4%	0.12%	1.23%	-11.30%	-22.07%
5	0.5	17.5%	0.20%	1.91%	-3.85%	-4.85%

Table 2.2: Accuracy comparison for different shielding levels and ESFs (154 conductor segments)

Table 2.2 shows the accuracy information for an example circuit with 154 conductor segments. Each driving end has a voltage source connected to the nearest ground wire, and each loading end has load capacitors connected to both the power and ground lines. We activate one of the driving sources, which is called attacker, with an 1-volt step function, and observe the responses of the loading ends of both the attacker and a faraway victim. The faraway victim is a conductor that is 10 conductors away from the attacker. From Table 2.2, we find that the shielding level and extended search factor (ESF) affect the accuracy in the following trend. Enlarging the ESF improves the accuracy on the attacker, but not the victim. On the contrary, enlarging the shielding level helps improve the accuracy on the faraway victim, but has less effectiveness to the attacker. Figure 2.5(a) and (b) shows the waveforms for different parameters and illustrates this trend. In this case, shielding level 1 with 0.5 ESF can already approach the exact solution very well for the active conductor, and a higher shielding level even improves the faraway accuracy more. We can see that the K-method with shielding level 5 and ESF 0.5 can almost match the exact solution. From this result, and by choosing a proper shielding level and ESF, we can capture the inductance effect precisely with very few mutual reluctance elements. Table 2.3 shows the runtime information of extractor and simulator (simulator is not part of this project, this part is developed by Tsung-Tao Chen[14]). The first column indicates the number of conductors in the circuit. We perform each simulation for 2000 time steps. The first case is the one we use to compare the accuracy previously. For this case, Tanner SPICE takes 1508.57 seconds to solve the exact solution while our tools only takes 10.33 seconds for a net speed-up of about 146x. Due to the superlinear dependence of solving time on matrix size, the speed up will be more dramatic for larger systems. For the sparse solution, we set the shielding level to 3 and the ESF to 0.5, and perform the WSA-based K-method. From Table 2.3, the extraction and simulation have 27.3x and 37.2x speed-up respectively for the circuit with 1367 conductors.

		Exact Sol.		K-based	
# of cond.	Density	Ext. time	Sim. time	Ext. time	Sim. time
154	11.4%	2.23	10.33	0.77 (2.9x)	7.20 (1.4x)
538	3.4%	26.32	119.65	2.76 (9.5x)	27.38 (4.4x)
989	1.8%	89.81	623.37	5.18 (17.3x)	54.20 (11.5x)
1367	1.1%	250.75	3565.71	9.17 (27.3x)	95.79 (37.2x)

Table 2.3: Run time (sec) of extractor and simulator

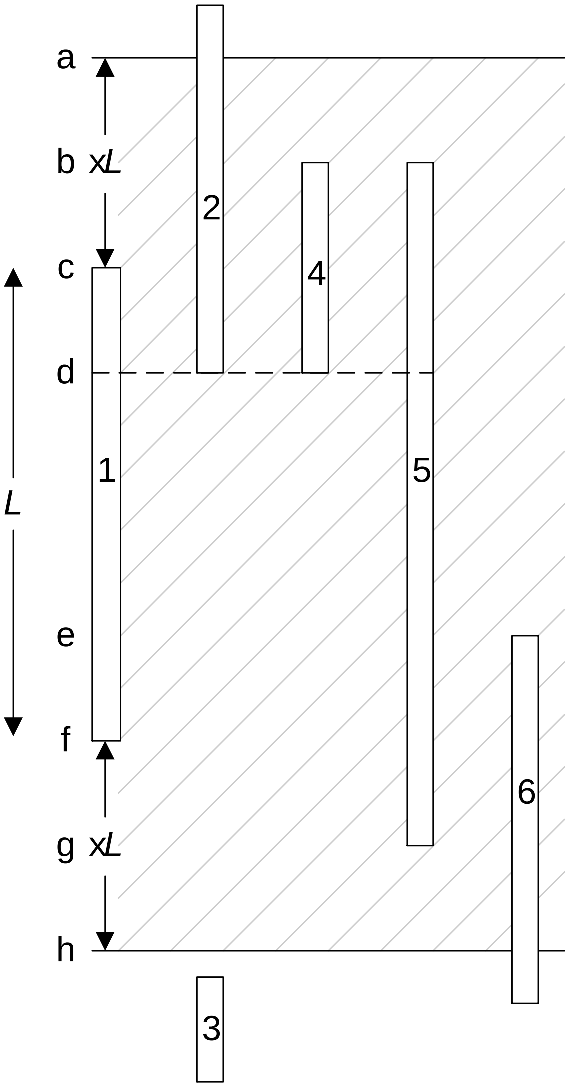


Figure 2.4: A circuit example for the definition of shielding

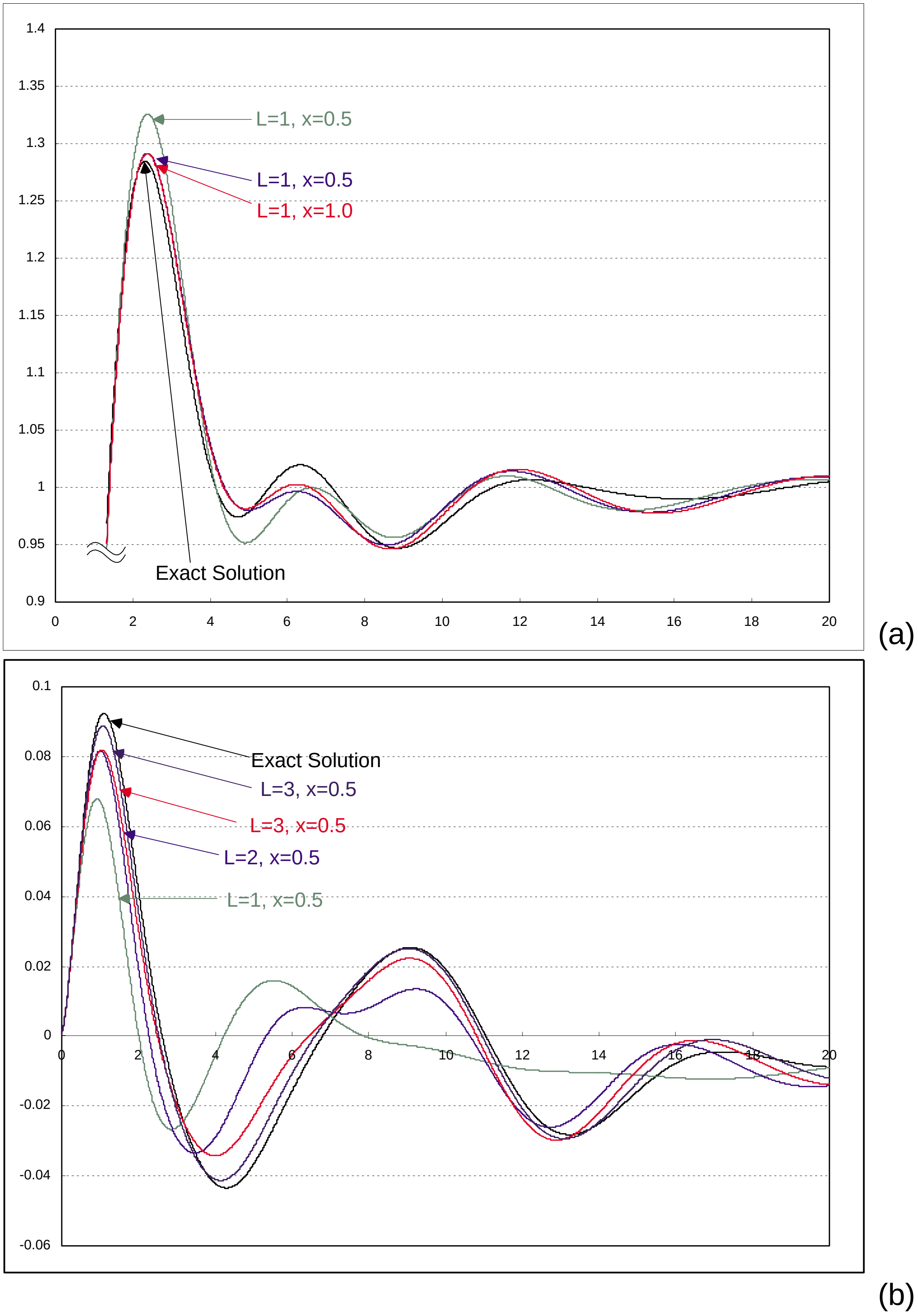


Figure 2.5: Waveforms for accuracy comparison (a) active line vs. different ESFs (b) faraway line vs. different shielding levels

Chapter 3

High Frequency Effect

3.1 Skin and Proximity Effect

At high frequency, current would become unevenly distributed, leading to changes in resistance and inductance value. This phenomenon can be characterized as skin effect and proximity effect. We briefly discuss each effect as follows.

3.1.1 Proximity Effect

At high frequencies current will tend to follow the path of least impedance. At low frequencies, the impedance is dominated by resistance of the interconnect. For a finite cross-section area of a interconnect, the current will spread out evenly over the cross-section to minimize the overall resistance. As switching frequency increases, the inductive part of the impedance, $R{+}jwL$, starts to dominate the overall term. In order to minimize the impedance, the loop size must be reduced to lower the loop inductance L. This results in current crowding on the return path and between signal lines. Refer to figure 3.1 for illustration of current distribution due to proximity effect.

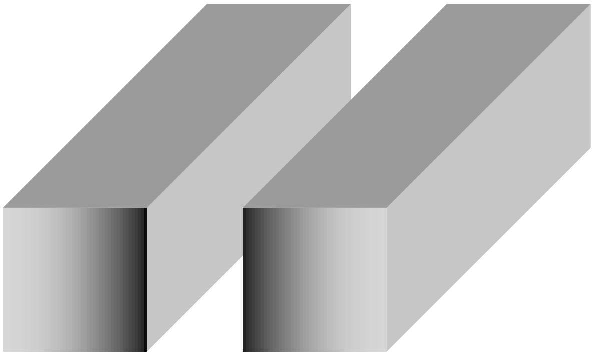


Figure 3.1: Illustration of proximity effect: current crowd to two side of the conductor

3.1.2 Skin Effect

Another effect of uneven current distribution in conductor due to high frequencies is called skin effect. At high frequencies, electro-magnetic wave is attenuated rapidly as it propagates in a good conductor. For practical purpose, at sufficient depth, all electro-magnetic wave are insignificant and no currents flow. As a result, all currents only flow close to the surface of the conductor. The overall effect is the increase in resistance as currents shift to the skin of the conductor. Intuitively speaking, since resistance is inversely proportion to cross-section area, and skin effect causes currents crowded to the surface, this in fact lowers the cross-section area and thus increases in resistance. Refer to figure 3.2 for illustration of current distribution due to skin effect.

3.1.3 Capturing skin and proximity effect

In order to capture both skin and proximity effect, conductors need to be discretized into filament (figure 3.3) so as to capture the non-uniform distribution of current within conductors. For a conductor system with k filaments, the impedance matrix at frequency ω can be denoted as follows:

$$Z_k(\omega)I_s(\omega) = V_s(\omega)$$

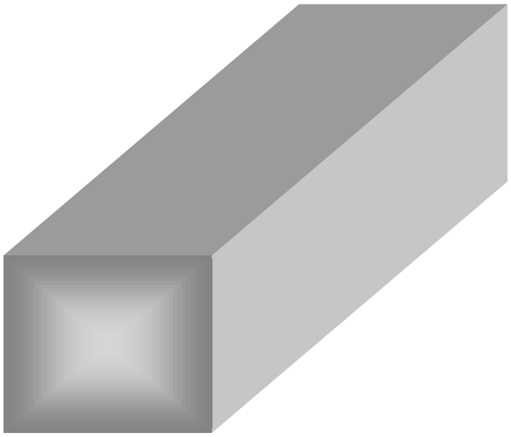


Figure 3.2: Illustration of skin effect: current crowd to conductor skin

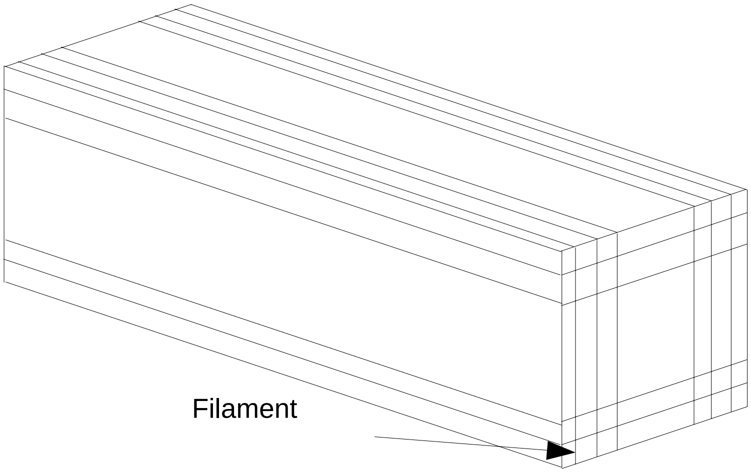


Figure 3.3: An example of discretization into filament

where I_s is branch current and V_s is voltage drop.

And the $Z_k(\omega)$ is as follows:

$$Z_k(\omega) = \begin{bmatrix} R_{11} + j\omega L_{11} & j\omega L_{12} & \cdots \\ \cdots & \cdots & j\omega L_{k-1k-1} \\ j\omega L_{k1} & \cdots & R_{kk} + j\omega L_{kk} \end{bmatrix}$$

where R_{ii} is the filament's resistance, L_{ij} is the inductance, and ω=$2\pi f$ is the angular frequency.

Given that $I_s(\omega) = Z(\omega)^{-1}V_s(\omega)$, V_s is being activated to voltage 1 each time, where $V_s \in \mathbf{R}^k$. By assembling each column of I_s, where $I_s \in \mathbf{R}^k$, the right hand side is in fact the identity matrix, and we essentially construct the effective admittance matrix, Y_k, where $Y_k \in \mathbf{R}^{k\times k}$. To get the lumped circuit element of the effective resistance and inductance for each conductor, we sum up current for each conductor to form a n×n admittance matrix, where n is the number of conductor. And we could obtain the lumped circuit element of R and L by inverting the n×n admittance matrix.

3.2 Frequency dependent reluctance extraction

In this session, we present our proposed frequency dependent reluctance extraction method.

Newmethod: As published in [14], we need to employ a window selection algorithm so as to capture the significant effect. However, if we only use the formula-based inductance value to form the small window, we cannot capture the skin and proximity effect. Based on this notion, we propose to use the original window selection algorithm. But when forming the small window we expand that to include filaments of each conductor. Hence, skin and proximity effect are captured, as they are short-range effect. Therefore, a n×n small window becomes a k×k window where n is the number of conductors and k is the number of filaments. After that, we solve the effective R and L as mention in previous section. Then use the effective L, and inverting it for the $\mathcal{K}$ matrix.

Differences:Previously proposed method capture the skin and proximity effect by cutting all the conductors in the system into filaments. This will lead to a very dense matrix and factorization will consume a lot of computation time. Instead, we capture the short-range effect of skin and proximity effect in the small window using window selection algorithm (WSA)[14]. The main advantage is that inversion of k×k filaments for each search window is a lot more computationally efficient than inversion of the entire conductor system with filaments. This results in significant speed up over the traditional method of cutting all the conductors in the system into filaments.

Justification:One may observe that this method involves an extra inversion from the effective L matrix into effective $\mathcal{K}$ matrix. However, this extra step is not as expensive as the inversion of entire system discretized into filaments. Because it only involves the inversion of a n×n matrix resulted from the search window. In a nutshell, the main idea is to capture the high frequency effect in a small window but expresses in the reluctance matrix domain, which exhibits locality.

A summarized comparison between the original and frequency dependent extraction is shown in table 3.1.

In the following example, we would show the difference between formula based method and frequency dependent method result using a 5×5 bus structure. The parameter for this example is as follows: width for each conductor is 5e-6, spacing are 1e-6, length is 1000e-6, thickness is 3.6e-7, sigma is 4.996e7, and evaluated by formula based method.

$$\mathcal{K} = \begin{bmatrix} 3.91 & -2.35 & -0.284 & -0.2474 & -0.289 \\ -2.35 & 5.31 & -2.20 & -0.157 & -0.247 \\ -0.284 & -2.20 & 5.31 & -2.20 & -0.284 \\ -0.2474 & -0.157 & -2.20 & 5.31 & -2.35 \\ -0.289 & -0.247 & -0.284 & -2.35 & 3.91 \end{bmatrix} \times 10^9 H_{-1} \ . \tag{3.1}$$

R_1=R_2=R_3=R_4=R_5=5.56 ohms

For the same conductor system, but evaluated at 30GHz, we can see difference as follows:

$$\mathcal{K}(\omega) = \begin{bmatrix} 2.13 & -1.32 & -0.2156 & -0.1295 & -0.155 \\ -1.32 & 2.96 & -1.20 & -0.1568 & -0.130 \\ -0.2156 & -1.20 & 2.976 & -2.156 & -0.216 \\ -0.1295 & -0.1568 & -2.156 & 2.96 & -1.32 \\ -0.155 & -0.130 & -0.216 & -1.32 & 2.13 \end{bmatrix} \times 10^9 H_{-1} \ . \qquad (3.2)$$

R_1=16.22, R_2=17.79, R_3=18.12, R_4=17.79, R_5=16.22 ohms

As stated above, we could notice the increase of resistance because of skin effect and the differences in reluctance resulting from the proximity and skin effect.

3.2.1 Comparison Study

In this section, we will present the frequency response comparison of formula-based L extraction with frequency-dependent L extraction. The formula-based L extraction data are produced by extracting a 50 conductors system using the formula-based L method. Then simulate the frequency response from 1GHz to 100 GHz. For the frequency dependent extraction data, we perform extraction for each frequency point, and then simulate the frequency response for each frequency. The setup of the experiment is as follows (see figure 3.4): Length=2000e-6, width=5e-6, height=3.6e-7, sigma=4.997e7;a ground in inserted every 9 conductor with 2ff as coupling capacitance, each end has a loading resistance of 1e-3 ohms, and each conductor has 1 driving resistance of 1 ohms. We activate the second conductor and observe the frequency respond of each conductor.

We selectively choose four different net to illustrate the differences. Aggressor net, the net next to aggressor, 4th neighbor, 14th neighbor.

As we observe from the following graphs, there are discrepancies between the formula based L extraction and frequency dependent L extraction. Before 30GHz, both methods

	Formula-based method	Frequency dependent method
Resistance	formula based	Use a small window to form the impedance matrix by filaments. Take the real part of effective impedance by inverting the n×n effective admittance matrix.
Reluctance	Use formula based L and invert it	Use a small window to form the impedance matrix by filaments. Take imaginary part of effective impedance from inverting effective admittance matrix, and divide it by ω. Then invert it to get the column of K_{asym}.
Advantage	Use window selection algorithm	Use window selection algorithm for sparsification Also capture skin and proximity effect.
Disadvantage	Cannot capture skin and proximity effect, which becomes a significant issue in high frequency.	Extra inversion involved: 1)Invert a k×k impedance matrix, where k is number of filaments. And sum up current for each conductor. to form the admittance matrix. 2)Invert a n×n admittance matrix to get effective R and L value. Then use the effective L to get $\mathcal{K}$ matrix.

Table 3.1: Comparison between original and frequency dependent reluctance extraction

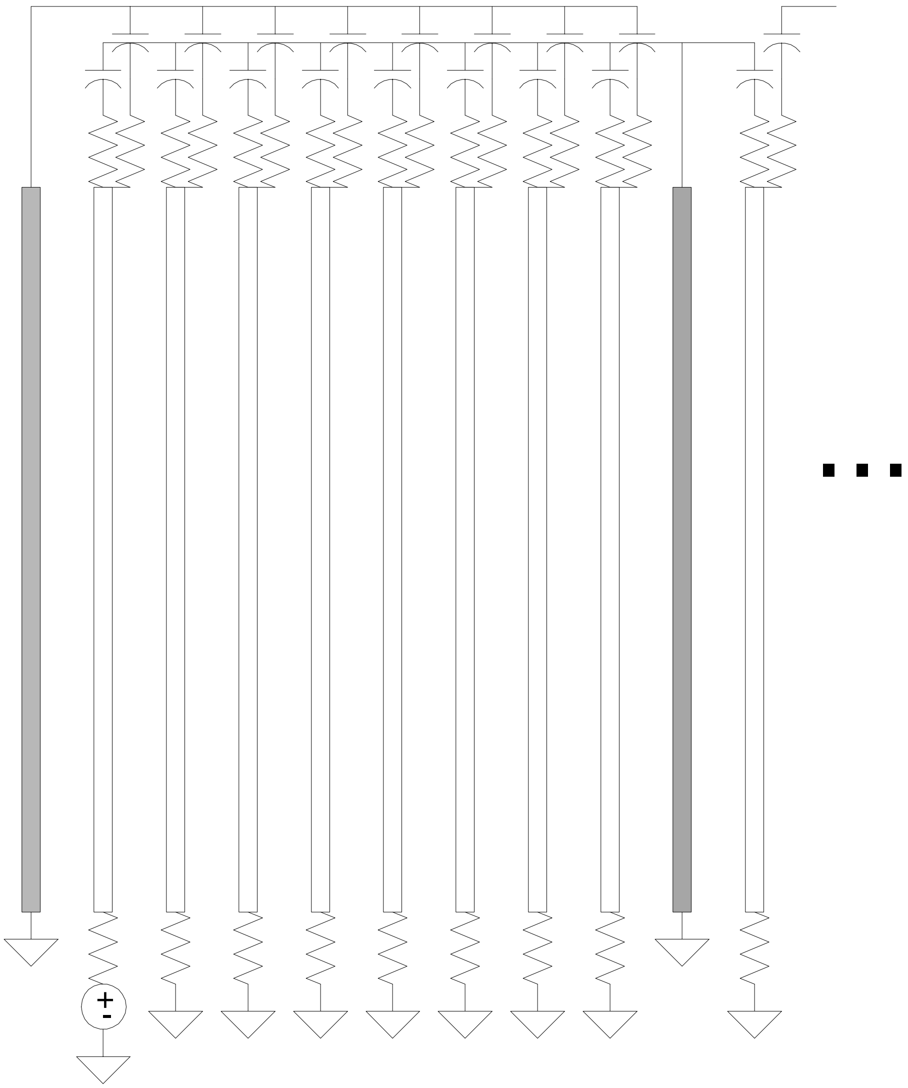


Figure 3.4: Experiment setup

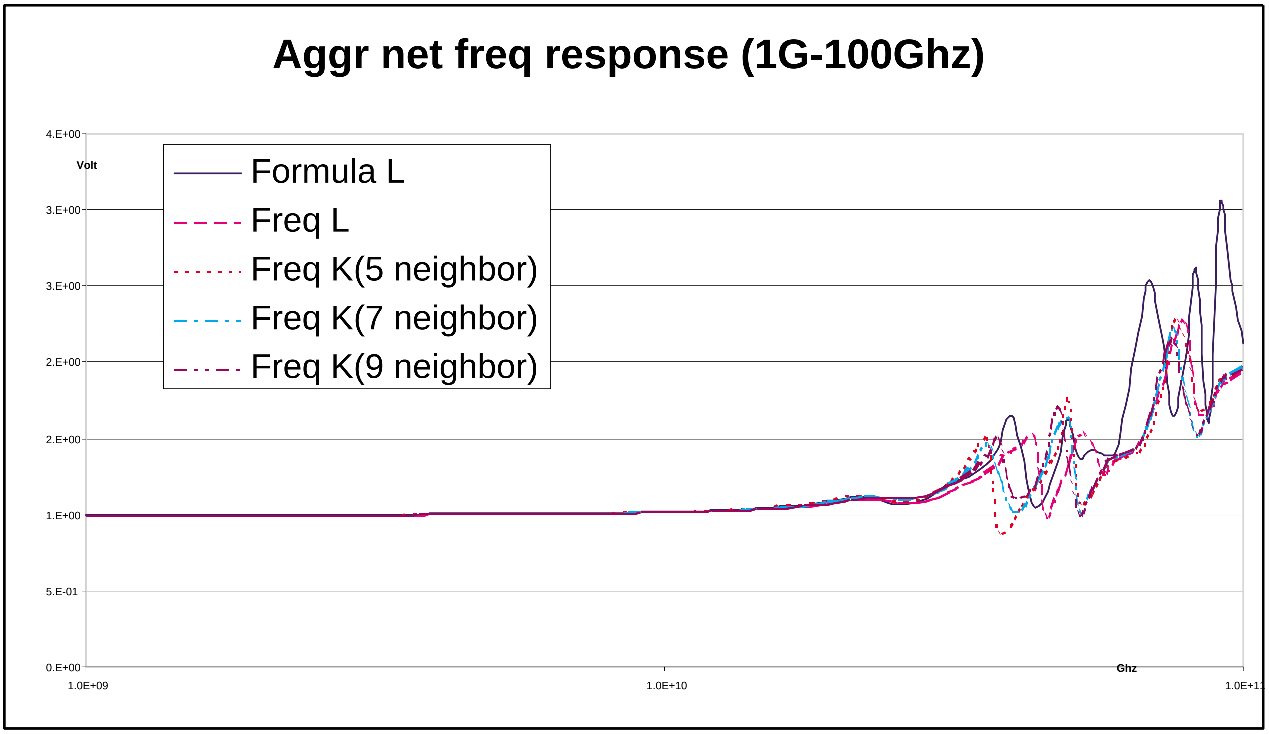


Figure 3.5: Frequency response of aggressor net 1-100Ghz

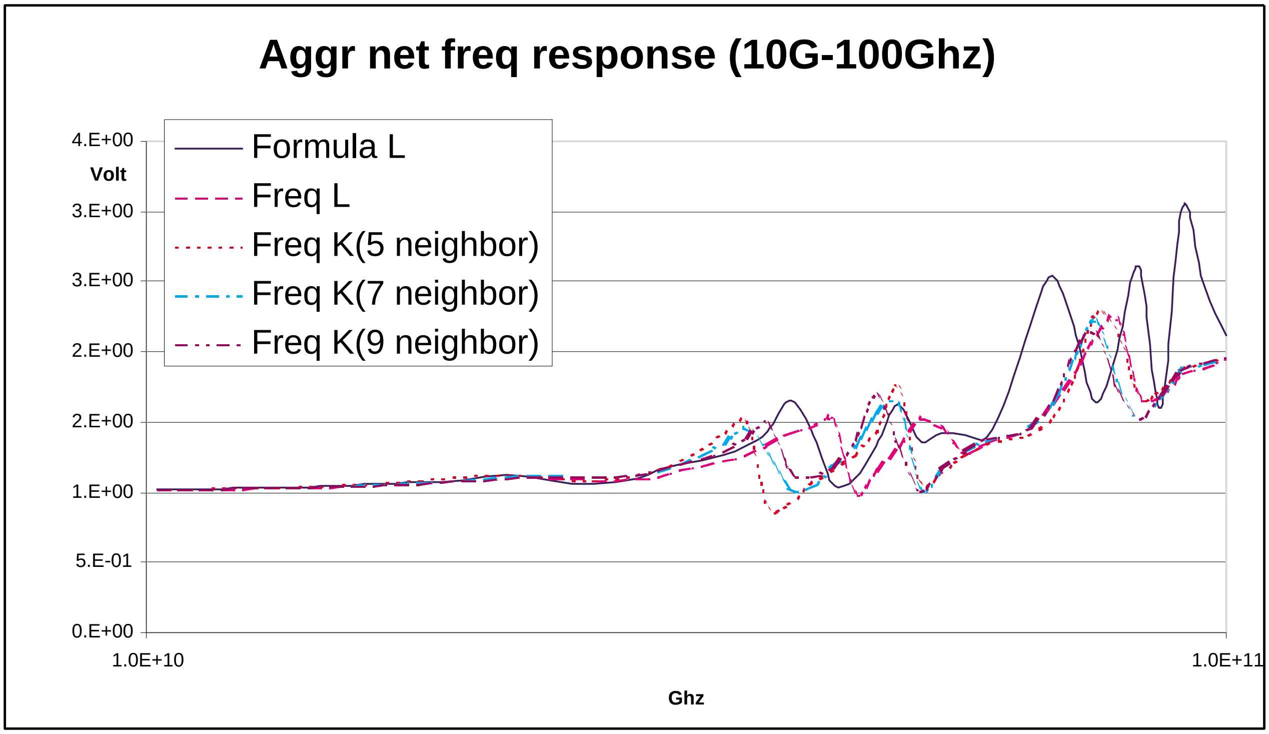


Figure 3.6: Frequency response of aggressor net 10-100Ghz

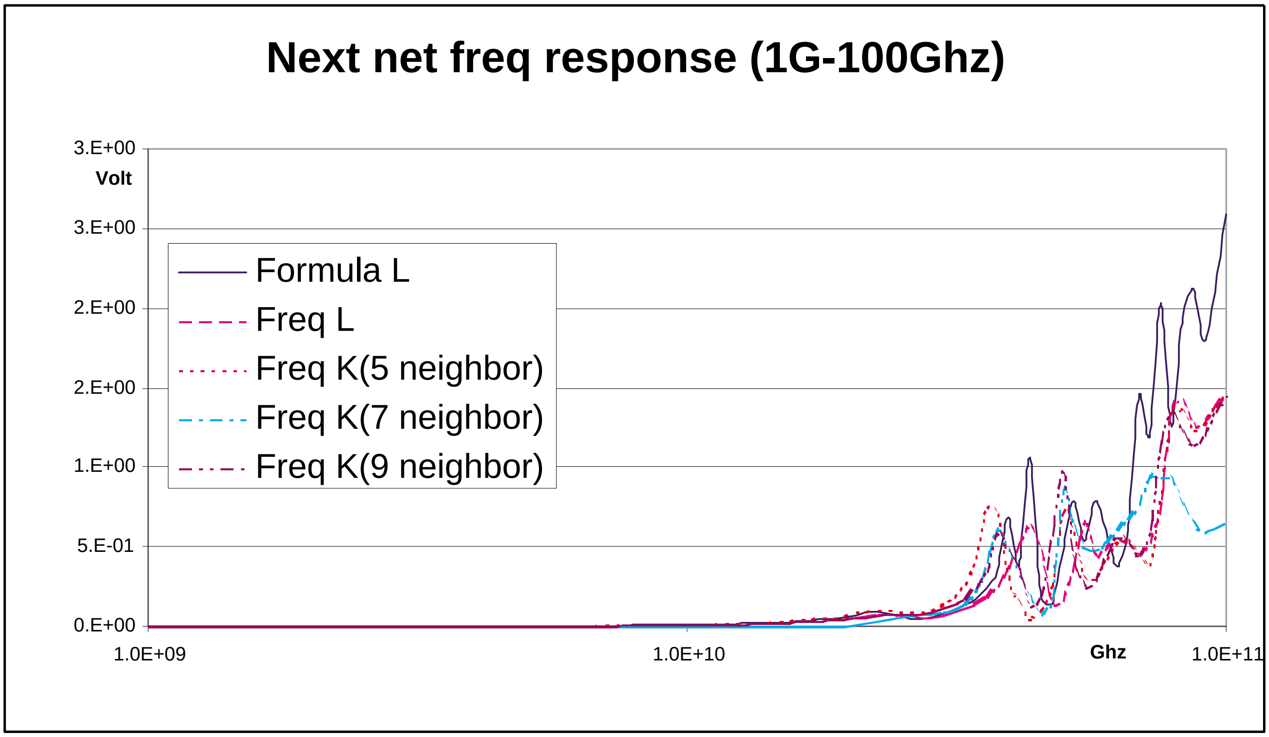


Figure 3.7: Frequency response of next net 1-100Ghz

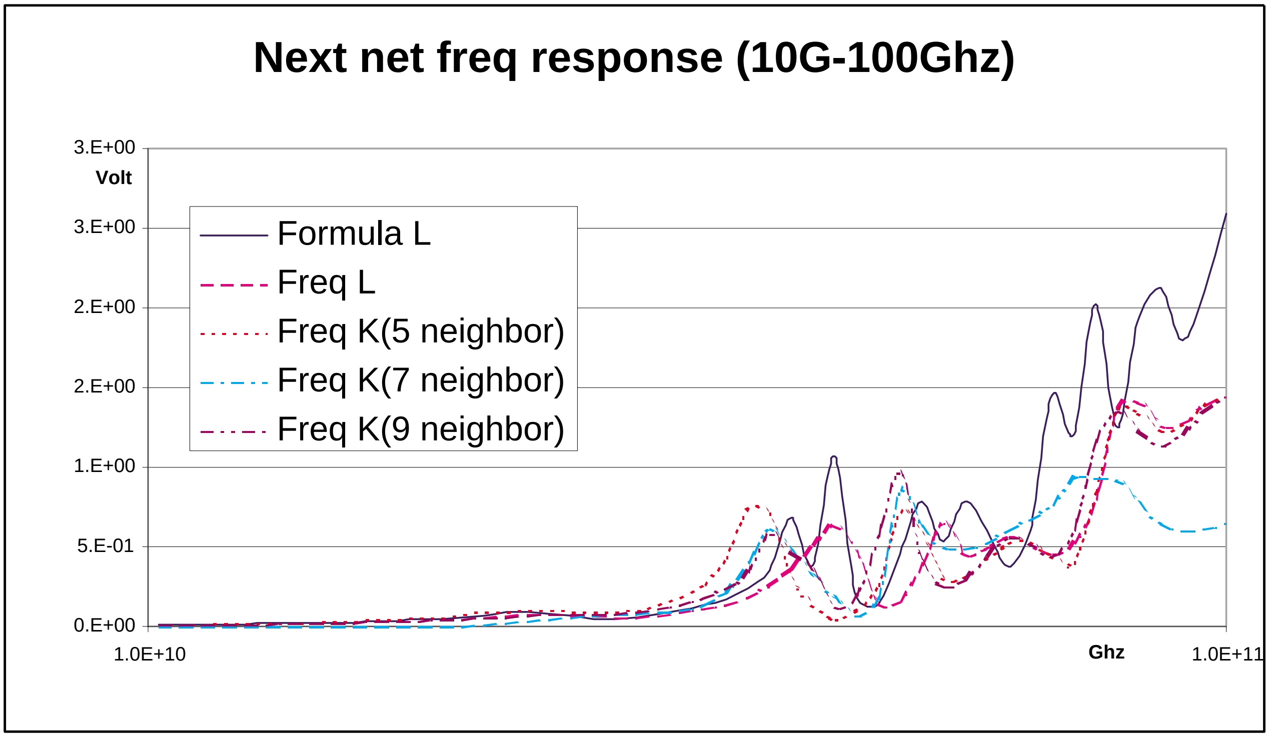


Figure 3.8: Frequency response of next net 10-100Ghz

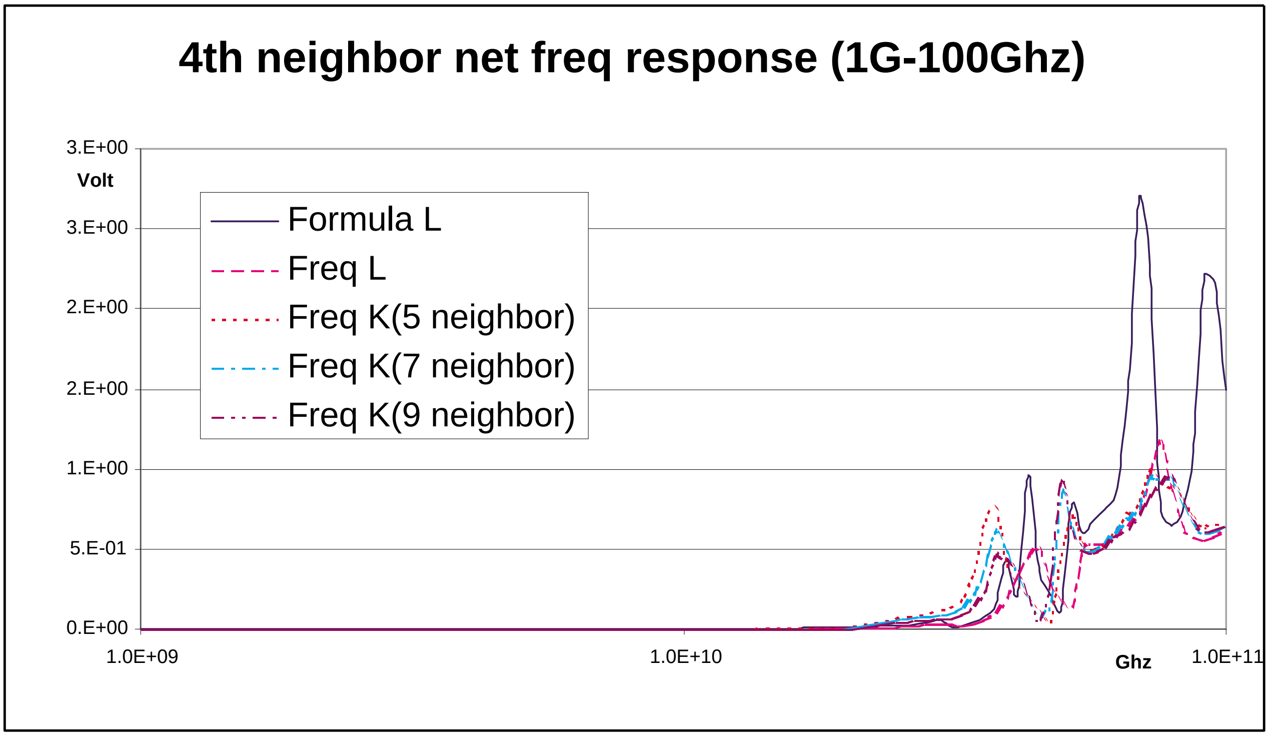


Figure 3.9: Frequency response of 4th neighbor net 1-100Ghz

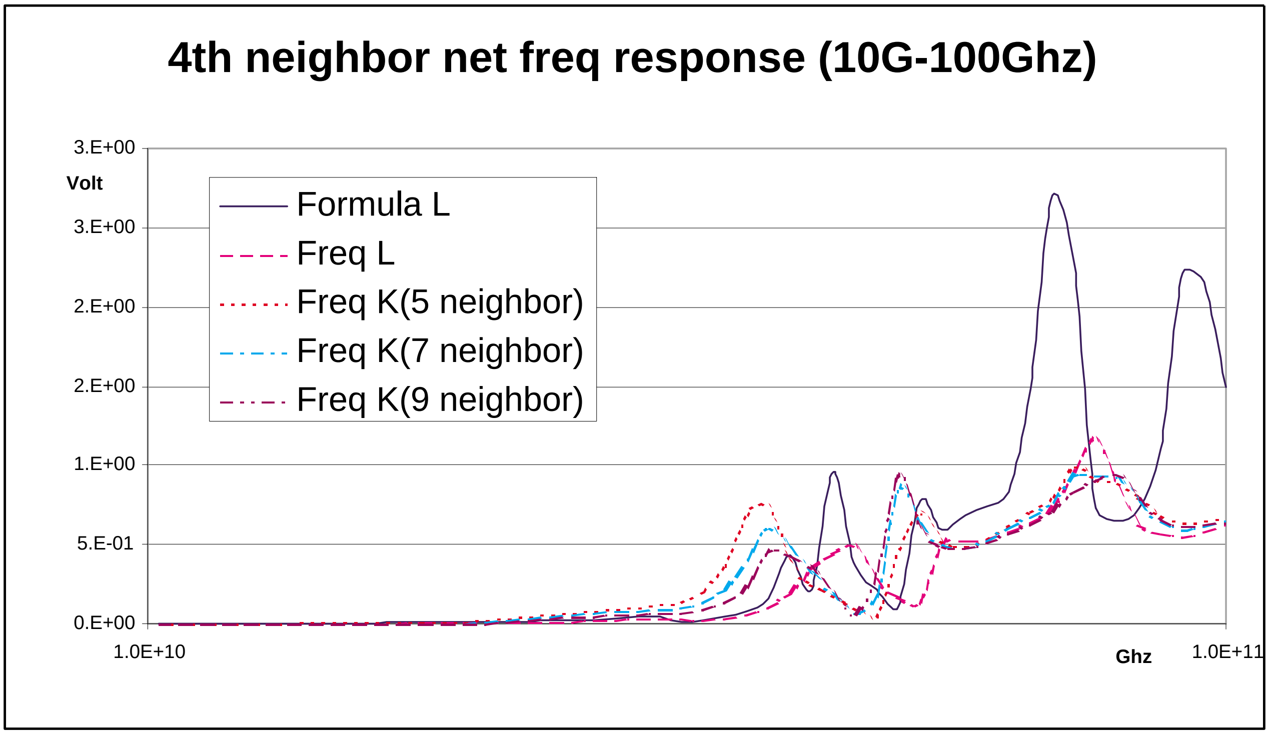


Figure 3.10: Frequency response of 4th neighbor net 10-100Ghz

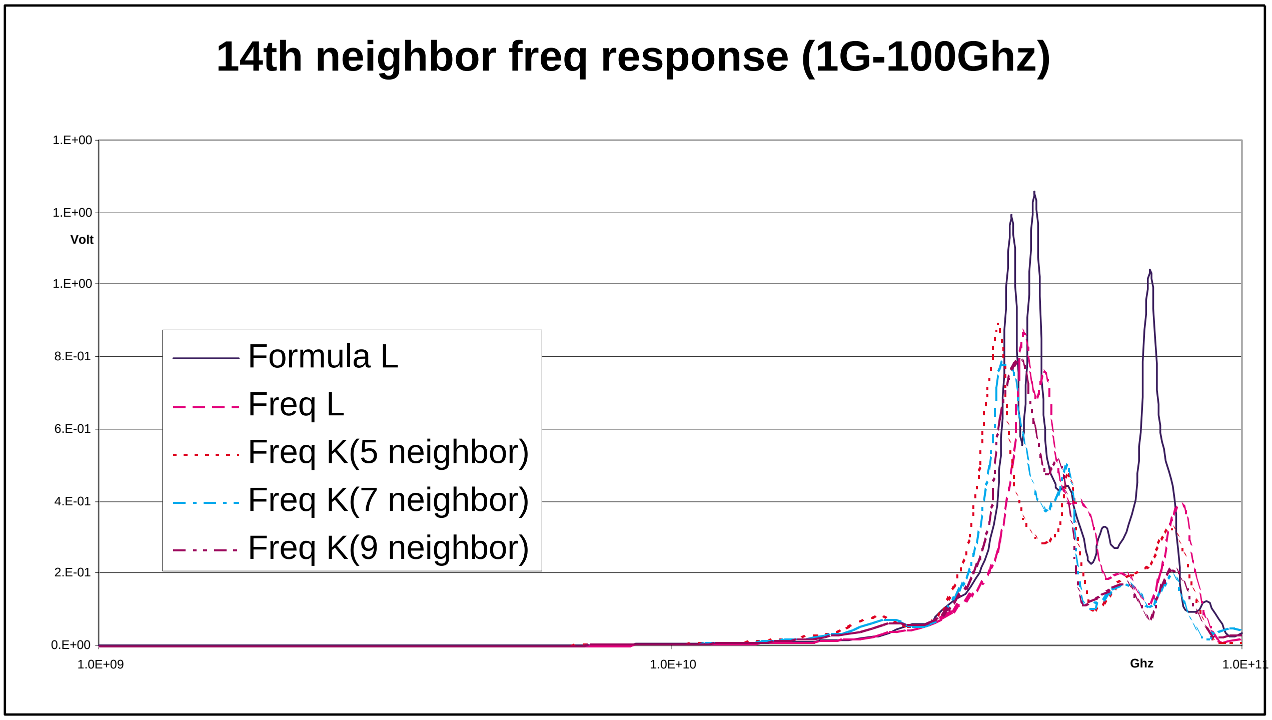


Figure 3.11: Frequency response of 14th neighbor net 1-100Ghz

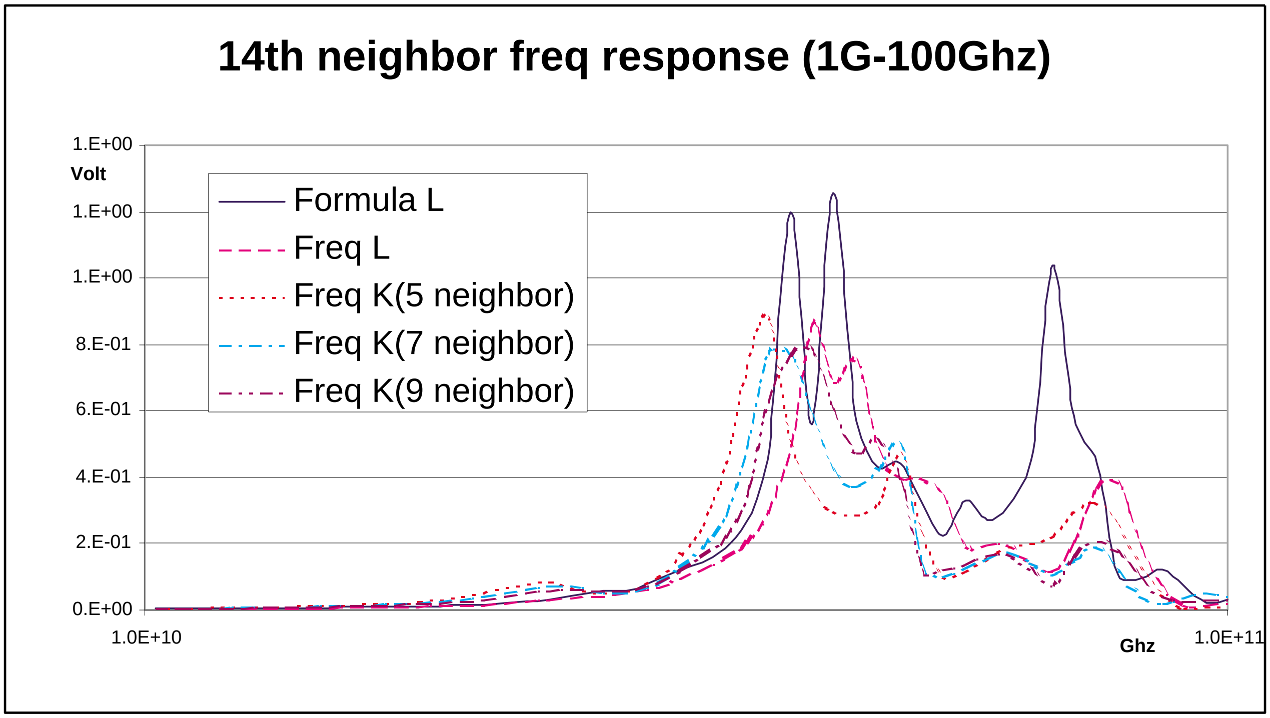


Figure 3.12: Frequency response of 14th neighbor net 10-100Ghz

are much in line with each other. Starting from 30GHz, for certain frequency there are dramatic differences. For some point it account for more than 50% difference. One may argue that such a high frequency is not going to be implemented in near future. However, what contributes to inductance effect is the switching frequency (clock edge), which is usually 10-15 times the clock frequency. In view of this, the need to correctly capture high frequency effect is crucial. Based on this notion, we have developed an efficient and accurate methodology to capture these high frequency effects using reluctance method and employing window selection algorithm. In the next section, we will present algorithm in detail.

3.3 Modified reluctance extraction algorithm

The overall algorithm of frequency dependent reluctance extraction can be summarized in the following table:

Frequency dependent reluctance extraction algorithm

1. Divide all conductors into two sets, vertical and horizontal.
2. For the vertical set, sort all the conductors with their x coordinates.
3. For every conductor from left to right (or bottom to top for horizontal net),
   (a) Use window selection algorithm to pick nearby conductors that posed as strong attacker
   (b) Formulate $Z_k(\omega) = R_k(\omega) + j\omega L_k(\omega)$

```
for i ← 1 to n
    do $Vs_i$ ← 1
        for j ← 1 to n
            do $Y_{ji}$ ← sum($I_k$) belongs to
conductor j
```

where k is number of filaments, n is number of conductors

(c) $Z_n(\omega)$=$Y_n^{-1}(\omega)$, where Y $\in \mathbf{R}^{n\times n}$, and n is number of conductors in the search window

(d) $R_n \leftarrow Re\{Z_n(\omega)\}$. Form L by $Im\{Z_n(\omega)\}/\omega$, then we set flux along the aggressor to one and others to zero. And solve the reluctance elements by inverting the small matrix corresponseing to the window. The resulting elements form the column of $\mathcal{K}_{asym}$ that corresponses to the aggressor.

4. repeat step 2 and 3 for the horizontal conductors.

5. Symmetrize by $\mathcal{K} = \frac{1}{2}(\mathcal{K}_{asym} + \mathcal{K}_{asym}^T)$

3.4 Experimental Result

We extend the formula-based extraction tool and implement the frequency-dependent reluctance and inductance in C++.

Table 3.2 shows the run time comparison between the exact solution (frequency L from previous graph) using frequency dependent L extraction technique compared with frequency dependent reluctance method with different number of neighbors. The simulations are run on an Intel Pentium IV 1.4 GHz system.

As illustrated in 3.5, 3.7, 3.9, 3.11, reluctance method with few neighbors has relatively good match compared with frequency-dependent L extraction. For closer look from 10G-100Ghz, please refer to 3.6, 3.8, 3.10, 3.12.

3.5 Implementation Detail

3.5.1 Basic Idea of Circuit Equations

Basically, we construct equation as nodal equation. The basic idea is to create a Z matrix by each individual filament. The real part is the resistance of each filament and the

	FastHenry	Frequency Dependent Reluctance		
# of	Extract	4 nei	5 nei	6 nei
cond.	time	ext. time	ext. time	extr. time
100	15 mins	2.91 s	4.46 s	6.39 s
300	2 hrs 17 mins	8.99 s	13.82 s	19.9 s
500	6 hrs 3 mins	15.04 s	23.21 s	33.43 s
1,000	-	30.17 s	46.6 s	67.25 s
10,000	-	302.24 s	467.61 s	674.59 s
100,000	-	50 mins	1 hr 18 mins	1 hr 52 mins

Table 3.2: Run time of Frequency dependent reluctance extraction

imaginary part is $\omega * L$.

That is $Z(\omega) = R + j\omega * L$.

Applying voltage 1 to activate them, then sum up all the current for each individual conductor. That means, if a conductor has 10 filament, we would need to sum up 10 filament's current. Repeat this procedure for activating each conductor. The result would be the current flowing in each conductor by activating each conductor. Since we are applying voltage 1 to activate them, this in fact is the admittance matrix.

Then, we invert the admittance matrix and it would become the impedance matrix. We use the real part as the frequency dependent resistance, where it takes into consideration of skin effect and proximity effect. And we use the imaginary part to construct the the effective inductance matrix and then invert it again to get the effective K matrix.

$$\begin{aligned} Z(\omega)I(\omega) &= V(\omega) \\ I(\omega) &= Z^{-1}(\omega) * 1 = Y(\omega) \end{aligned} \tag{3.3}$$

That is the basic idea of the frequency analysis.

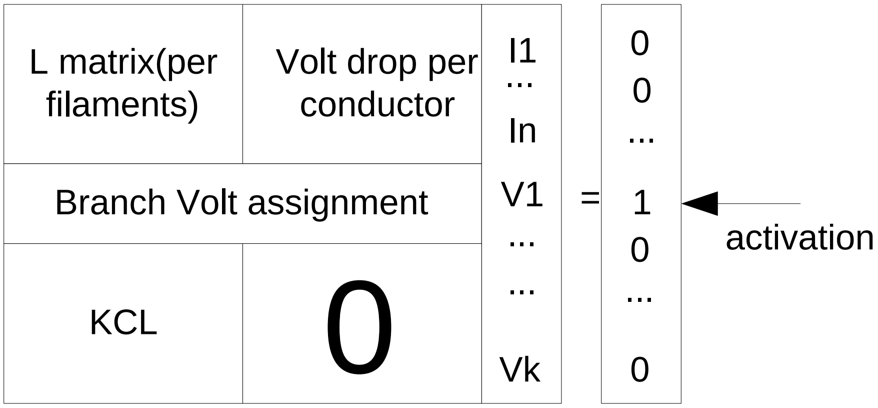


Figure 3.13: Matrix formulation for frequency analysis

3.5.2 Matrix Formulation

In general, the matrix is formulated as follows:

$$\begin{bmatrix} L_{11} & \cdots & L_{1n} & -1 & 0 & 0 & 0 \\ \cdots & \cdots & L_{2n} & -1 & 0 & 0 & 0 \\ L_{m1} & \cdots & L_{mn} & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & \cdots & \cdots & \cdots \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ \cdots & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_1 \\ \cdots \\ i_n \\ V1 \\ \cdots \\ Vk \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{3.4}$$

where n ∈ number of filaments, and k ∈ number of branches.

If we represent them as a block matrix, we could represent it as follows: So, we form the effective admittance matrix by activating each conductor and sum up the total current in each conductor. This means that the 1 on right hand side would be set to 1 if that conductor is activated, and the rest are 0.

One could form this matrix alternatively as MNA equation, which uses nodal voltage instead of branch voltage. However, it would make the matrix size much bigger. For frequency analysis without ground plane, we choose to formulate it in this way for the

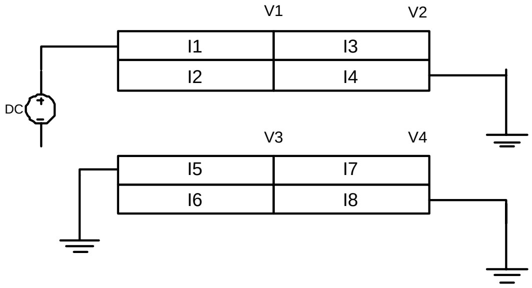


Figure 3.14: An example for forming the matrix

sake of smaller matrix size. Later, we would show a formulation using MNA nodal voltage when we consider ground plane.

3.5.3 Small Example

We will illustrate the idea of matrix formulation in the following small example. This is an example with 2 conductors and each with 2 filament and 2 segment. The voltage is defined as the branch voltage, which means V1 is is voltage dropped from voltage source to the point of V1. And current are defined as the branch current, which is just the current flowing in each filament. Figure 3.14 depicts the circuit. For such a configuration, the matrix would be formed as follow:

$$\begin{bmatrix} L_{11} & L_{12} & L_{13} & L_{14} & L_{15} & L_{16} & L_{17} & L_{18} & -1 & 0 & 0 & 0 \\ L_{21} & L_{22} & L_{23} & L_{24} & L_{25} & L_{26} & L_{27} & L_{28} & -1 & 0 & 0 & 0 \\ L_{31} & L_{32} & L_{33} & L_{34} & L_{35} & L_{36} & L_{37} & L_{38} & 0 & -1 & 0 & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} & L_{45} & L_{46} & L_{47} & L_{48} & 0 & -1 & 0 & 0 \\ L_{51} & L_{52} & L_{53} & L_{54} & L_{55} & L_{56} & L_{57} & L_{58} & 0 & 0 & -1 & 0 \\ L_{61} & L_{62} & L_{63} & L_{64} & L_{65} & L_{66} & L_{67} & L_{68} & 0 & 0 & -1 & 0 \\ L_{71} & L_{72} & L_{73} & L_{74} & L_{75} & L_{76} & L_{77} & L_{78} & 0 & 0 & 0 & -1 \\ L_{81} & L_{82} & L_{83} & L_{84} & L_{85} & L_{86} & L_{87} & L_{88} & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \\ i_8 \\ V1 \\ V2 \\ V3 \\ V4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.5}$$

Here we activate the first conductor by applying voltage 1. This is done by $V1+V2=1$, which means voltage drop across $V1+V2$ is 1. Then, we sum up current for conductor 1 from $I1+I2$ and conductor 2 from $I5+I6$. For the second time, we activate V3+V4=1 and solve again for current. The result would be:

$$\begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix} \tag{3.6}$$

And we invert this to get the effective Z.

3.5.4 Window Search Algorithm for Non-orthogonal wire

For non-orthogonal wire, they are treated as a regular wire but it is inserted into both vertical and horizontal structure for sorting. For vertical wires, they are sorted for its $x1$ coordinate (the leftmost x coordinate). And for horizontal wires, they are sorted for its $y2$ coordinate (the lower y coordinate).

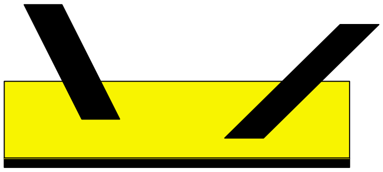


Figure 3.15: Horizontal window search for non-orthogonal wire

To elaborate on that, takes the example of horizontal window searching. If a non-orthogonal wire's lower y coordinate fit within the search range and as well as their x coordinate are within the range of the horizontal wire.

Then it would be chosen, and tested if that particular part has been shield. (using recursive_search(), just like regular wire).

An example is depicted in figure 3.15. Yellow space is the search range for the horizontal wire. And the same idea applies to vertical wire, where it uses leftmost x coordinate to check for the search range.

As far as data structure concern, I do not create any new data structure for storing the non-orthogonal wire. But to avoid printing duplicate self reluctance or inductance, (because they are in both vertical and horizontal sorted list), we will only print the self reluctance or inductance from the vertical sorted list.

3.5.5 Trick when calculating Non-orthogonal wire inductance

For the non-orthogonal wire, neither Grover's nor the Hoer's formula applies. Please refer to appendix B for the formula.

However, the formula need to be applied with caution. Because it does not work, if the projected intersecting point is in the middle. Refer to figure 3.16.

When this happens, the wire would be cut into two pieces like figure 3.17. Then the couple inductance can be calculated by the sum of the individual coupling inductance. That means, we applied the formula to each individual piece of wire. The sum of them is the total coupling inductance.

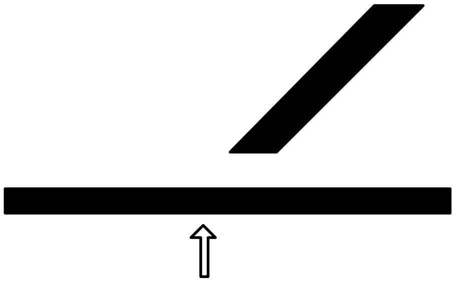


Figure 3.16: Project intersect point in the middle

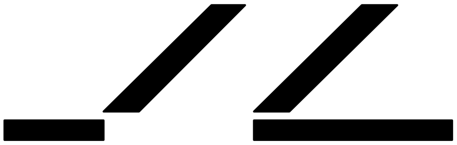


Figure 3.17: Wire is cut. And treated as two separate calculation

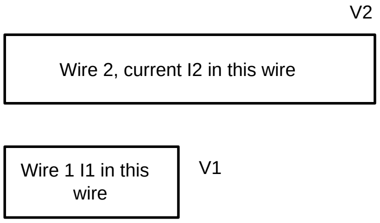


Figure 3.18: An example for no cutting in wire

For example there are two wires, and $I1$ is running in wire 1 and $I2$ is running in wire 2. We can describe them as follow:

$$\begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} V1 \\ V2 \end{bmatrix} \tag{3.7}$$

If we cut the wire, the structure would be like figure 3.19

We can describe figure 3.19 formulation as follows:

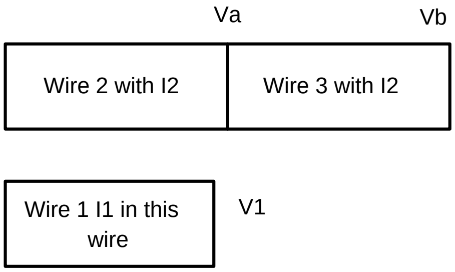


Figure 3.19: An example with cutting in wire

$$\begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} V1 \\ Va \\ Vb \end{bmatrix} \tag{3.8}$$

Since V2 = Va+Vb. We can equal them like this. $Va + Vb = (L_{21} + L_{31})I1 + (L_{22} + L_{23} + L_{32} + L_{33})I2$

And this also equal to: $V2 = L_{21} * I1 + L_{22} * I2$

In view of this, we know that the couple term can be sum be individually.

The class non_orth is the class that implement the wire cutting part. Basically, it find the project intersecting point and cut the wire into two pieces. The total coupling term are sum up by the individual term.

Chapter 4

Handling Ground plane

4.1 Ground plane consideration

For certain digital design, an explicit ground plane is added to provide return path for circuit as well as a shielding element to mitigate inductance-coupling effect. In our frequency reluctance extractor implementation, we also consider input as ground plane. In the following example, we show a simple 5 conductors system that inductance as well as resistance value differences extracted with and without ground plane. For the sake of simplicity, we show the data using the inductance elements instead of reluctance. The setup is as follows: width for each conductor is 2e-6, spacing are 1e-6, thickness is 3.6e-7, length is 1000e-6, sigma is 4.996e7, and with 5 filaments for width and 1 filament for height, and evaluated at 30GHz.

$$\mathcal{L}(\omega) =$$
$$\begin{bmatrix} 16.23 + j12.5 & 0 + j9.68 & 0 + j8.39 & 0 + j7.60 & 0 + j7.1 \\ 0 + j9.68e & 17.8 + j12.35 & 0 + j9.61 & 0 + j8.35 & 0 + j7.6 \\ 0 + j8.39e & 0 + j9.61 & 18.1 + j12.3 & 0 + j9.61 & 0 + j8.4 \\ 0 + j7.60e & 0 + j8.35 & 0 + j9.61 & 17.8 + j12.4 & 0 + j9.7 \\ 0 + j7.055 & 0 + j7.601 & 0 + j8.39 & 0 + j9.68 & 16.2 + j12.5 \end{bmatrix}$$
$$\times 10^{-10} H \ .$$

For the same setup, we evaluated again but with a ground plane in the center below the two conductors. The setup is as follows:200e-6 wide, 600e-6 long, thickness is 2.8e-7, sigma is 4.947e+7, and separate vertically by 3.8e-7. Now the L matrix becomes:

$$\mathcal{L}(\omega) =$$
$$\begin{bmatrix} 16.35 + j12.1 & 0 + j9.30 & 0 + j8.0 & 0 + j7.21e & 0 + j6.657 \\ 0 + j9.30 & 17.9 + j12.0 & 0 + j9.21 & 0 + j7.95 & 0 + j7.19 \\ 0 + j8.0 & 0 + j9.21 & 18.22 + j11.9 & 0 + j9.2 & 0 + j7.97 \\ 0 + j7.21 & 0 + j7.95 & 0 + j9.2 & 17.9 + j11.9 & 0 + j9.26 \\ 0 + j6.657 & 0 + j7.19 & 0 + j7.97 & 0 + j9.26e & 16.3 + j12.0 \end{bmatrix}$$
$$\times 10^{-10} H \ .$$

As we could see, with the insertion of ground plane it serves as a shielding element and alters both the inductance and resistance value. In our implementation, we discretize the ground plane with both horizontal and vertical discretization, and attach two ports in two ends (figure 4.1). Then we activate each conductor at a time similar to the algorithm mentioned above but skipped the ground plane activation. Instead, the ground plane ports are assigned to be zero. Then we collect the current to form the effective admittance matrix and proceed as before.

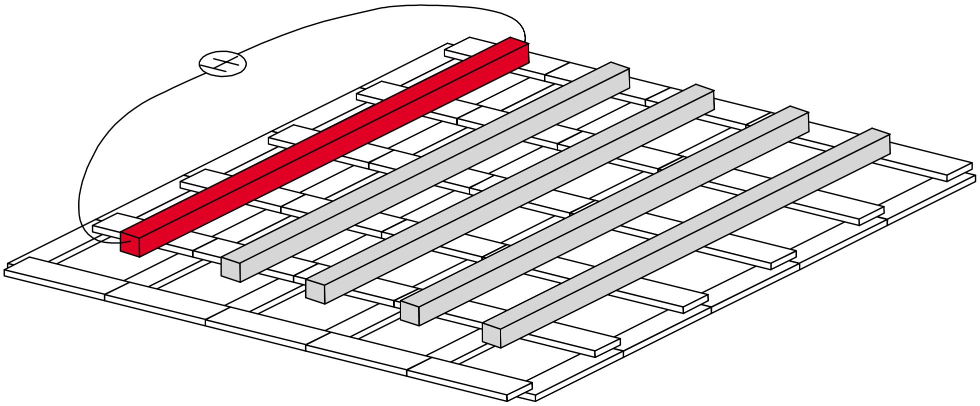


Figure 4.1: ground plane discretization example

4.2 Ground plane pre-computation

Because we discretize the ground plane in both x and y direction, the ground plane becomes a grid-like structure (See figure 4.1). Even with three to four discretized segments in either x or y direction, we would end up having a 15×15 or even 20×20 matrix. This grid-like structure mesh involves both self and mutual inductance between each discretized segment. Computing them on-the-fly would dramatically impact computational efficiency. Since for a large layout circuit, there may be numerous ground planes (or wide conductors) present in the layout.

Employing window-search algorithm would probably bring in several ground planes. (of course, that also depends on how it is layout.) As a result, if we compute all ground plane on-the-fly, it would seriously impact the performance. Furthermore, as one may observe, those grid-like structure self and mutual inductance values are constant relative to its segment within the grid. That is, if the window search algorithm picks the ground plane, it have to bring it a 15×15 or even 20×20 matrix. Based on this observation, we would pre-compute the ground plane into a grid-like structure and copy them if it is picked by the window search algorithm. However, the mutual inductance between ground planes and conductors would still be computed on-the-fly as it changes dynamically. We

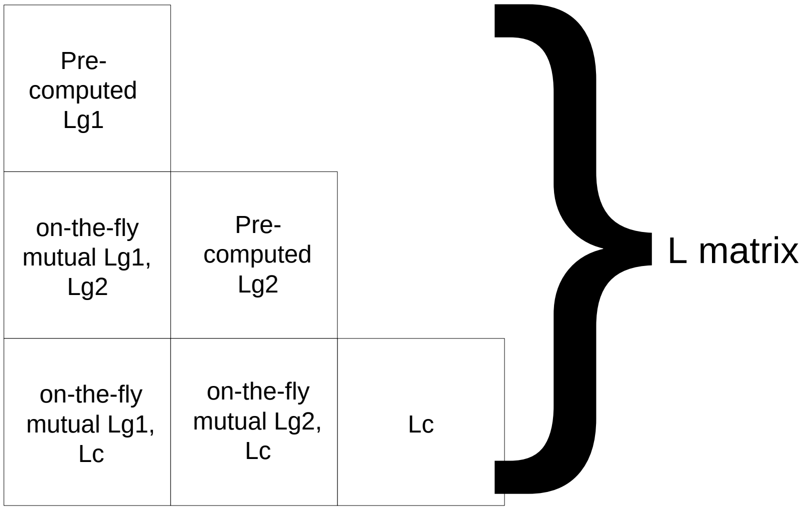

Pre-computed Lg1
on-the-fly mutual Lg1, Lg2
Pre-computed Lg2
on-the-fly mutual Lg1, Lc
on-the-fly mutual Lg2, Lc
Lc
L matrix

Figure 4.2: Ground plane pre-computation. Lg is L matrix for ground planes, Lc is L matrix for conductors.

illustrate this idea in figure 4.2.

4.3 Implementation Detail

4.3.1 Matrix formulation

For frequency dependent analysis involving ground plane, we formulate the matrix as MNA equations where we used nodal voltage instead of branch voltage.

A general form is as follows:

$$\begin{bmatrix} 1 & 2 & 4 & (0) & (0) & (0) & (0) \\ 2' & 1 & 4 & (0) & (0) & (0) & (0) \\ 4' & 4' & 3 & (0) & (0) & 6 & (0) \\ (0) & (0) & (0) & 7 & 7 & 7 & 7 \\ (0) & (0) & 8 & 8 & (0) & (0) & (0) \\ 9 & 9 & (0) & (0) & 9 & (0) & (0) \end{bmatrix} \begin{bmatrix} Lg \\ \cdots \\ Lc \\ Isc \\ Isg \\ Vc \\ Vg \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.1}$$

Each number is in fact a block-matrix. 0 really means zero.

Ig: current element of ground plane.

Ic: current element of conductor.

Is: port current of conductor, then ground plane i.e. Isc, Isg

Vc: node voltage of conductor.

Vg: node voltage of ground plane.

1: Lg, L matrix for ground plane.

2: Lg1 to Lg2 mutual L between ground plane and ground plane.

3: Lc, L matrix for conductor.

4: Lgc, mutual L between ground plane and conductor.

5: ground plane voltage drop of each segment.

6: conductor voltage drop of each segment.

7: voltage assignment of conductor and ground plane (which one to be activated)

8: KCL of conductor.

9: KCL of ground plane.

Chapter 5

Handling Large scale extraction

5.1 Introduction

For very large layout project, sometimes there will be far more conductors than the system memory can hold. When this happens, swap memory will be used and it will greatly impact the extraction performance since it uses hard disk space for temporary storage. One may observe that window-based search algorithm for reluctance has locality property. That is, we may not need to take in the entire search space to perform the window search algorithm. Based on this observation, it prompts us to develop a slightly modified version that is capable to extract large layout project.

5.2 Modified input and output

In order to handle such a large layout input, we divide up the entire layout input in number of smaller files that are sorted in the x-direction. (Files are sorted externally, for example use merge sort.) For practical purpose, we can limit the search window to be around 50-70 μm. We modify our extraction tool to take input from several input files. With this modification, extraction can be done in parallel with several instance of the program running in different machine. And for each instance of the program, we feed in number of input files up to the search window size.

Because of the parallel nature of the program, we lose the visibility of the entire search space. As a result, we cannot average the un-symmetrized pair to in the $\mathcal{K}$ matrix. That is, averaging (i,j) and (j,i) pair. This is the fifth step in Window Section Algorithm (WSA). (Symmetrize by $\mathcal{K} = \frac{1}{2}(\mathcal{K}_{asym} + \mathcal{K}_{asym}^T)$, See Window selection algorith(WSA)) The reason for un-symmetrized pair is that when forming the small window either (i,j) or (j,i) has the full visibility but not the other. Hence, one of the mutual term will have conductors missed in its small window.

To elaborate on this, we illustrate in the following small example. Refer to figure 5.1 and 5.2. In this example, we have 4 files and each file has 3 conductors. Assume that we use two neighbors for search window. Since the search window is up to two neighbors, each instance of the program will take in 3 files. Thus, machine A will take in file 0,1,2; machine B will take in file 1,2,3; machine C will take in file 2,3,4. For conductor 8 in machine A, the window search can only search its left-hand-side neighbor, 6 and 7 because conductor 9 and 10 belongs to file 3. Hence, for conductor 8, we do not has the full (i,j) and (j,i) pair in $\mathcal{K}$ matrix.

To solve this problem, we can simply output only one of the two pair value. During simulation, it externally averages out the (i,j) and (j,i) pair value from different file since the simulator needs to read in all extracted SPICE files anyway.

Based on this observation, we need to modify a bit of the input routine to handle multiple files as input and to flag those has full pair verse those who does not. Furthermore, for files at the left side and right side boundary we should also include them as having full pair since there are no more files at the boundary. In addition, we need to modify the output routine to determine whether or not to take the average from the pair or just use single (i,j) or (j,i) term. We outline this detail changes in 5.1 and 5.2.

```
input:file id (current file number), nei (number of neighbor to search)
for i $\leftarrow$ (file id-nei) to (file id+nei)
  a. open file i
    if (i $\leq$ file id)AND(boundary left flag ==1)
      set all conductors from file i as current block(i.e.full pair)
    else
    if (i $>$ file id)AND(boundary right flag ==1)
      set all conductors from file i as current block(i.e. we have full pair)
    else
      set all conductors from file i as different block(i.e. only (i,j) or (j,i) not full pair)
end
```

Table 5.1: Modified input routine for parallel extraction

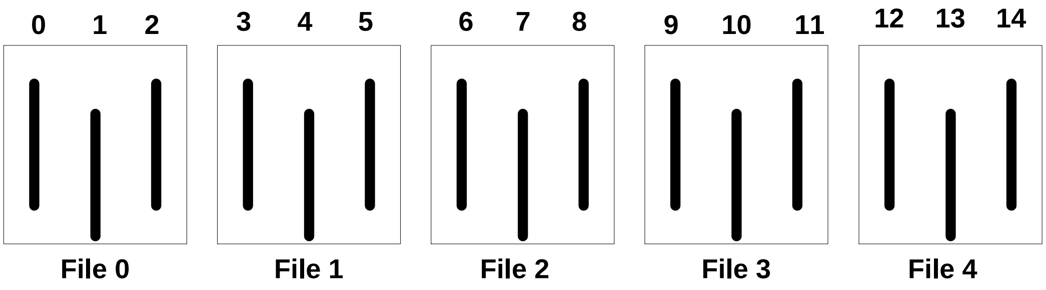


Figure 5.1: An example with 5 files

```
input:$\mathcal{K}_{asym}$ (a sparse matrix)
output:$\mathcal{K}_{reducedsize}$(a sparse matrix)
k ← total number of conductors
for i ← 1 to k
   we traverse from from diagonal term of $\mathcal{K}_{asym}$
   if (i==j)
        $\mathcal{K}_{reducedsize}$ ← $\mathcal{K}_{ii}$ value
   else
   if (i!=j)AND(i ∈ current block)AND((j ∈ current block))
         $\mathcal{K}_{ij}$ ← 1/2(i value+ j value)
   else
   if ((i!=j)AND(i ∈ current block)AND(j ∉ current block)
         $\mathcal{K}_{ij}$ ← (i value)
   else
   if ((i!=j)AND(j ∈ current block)AND(i ∉ current block)
         $\mathcal{K}_{ij}$ ← (j value)
   end
end
```

Table 5.2: Modified output routine for parallel extraction

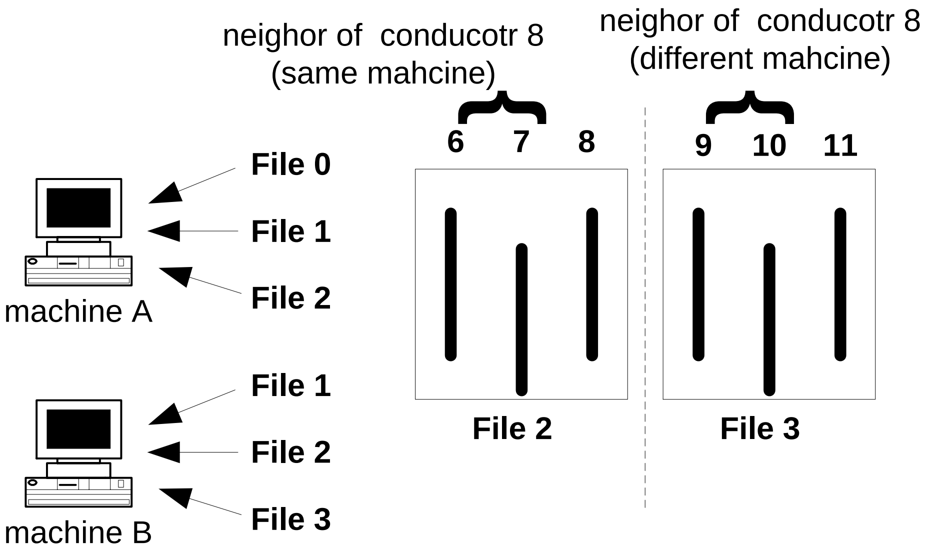


Figure 5.2: Illustration of not having full pair for $\mathcal{K}$ matrix

Chapter 6

Conclusion

Modeling inductance is a challenging task to both circuit designer and CAD designer. Due to its long-range effect, it leads to a dense inductance matrix. This posed as impediment to both circuit extraction and simulation. With the introduction of reluctance formulation, we develop an efficient, stable, and accurate reluctance extraction methodology. We address stability issue by using the recursive bisection algorithm. To address the issue for the irregular geometries, we develop a window search algorithm to handle VLSI-like geometries.

Next, we establish the need for modeling high frequency effect like skin and proximity effect via a frequency response analysis. Then we propose a frequency dependent reluctance methodology that cut conductor into filaments within a search window so as to capture skin and proximity effect. This significantly reduces the computational time since it avoids cutting entire system into filaments and factorizing them. The computational efficient is demonstrated in the run-time table and accuracy is shown in comparing frequency response with the exact solution.

Appendix A

A.1 Integration into Nodal Formulation

We would briefly review the formulation of nodal analysis for reluctance. For detail information, please refer to [14]. First, it start from forming MNA equations for $\mathcal{L}$ matrix. For a linear circuit, it could be generalized as follows:

$$\tilde{G}x + \tilde{C}\dot{x} = b, \tag{A.1}$$

$$\begin{aligned} & \tilde{G} = [\begin{matrix} G & A_l^T \\ -A_l & 0 \end{matrix}],\ x = [\begin{matrix} v_n \\ i_l \end{matrix}] \\ where \quad & \\ & \tilde{C} = [\begin{matrix} C & 0 \\ 0 & \mathcal{L} \end{matrix}],\ b = [\begin{matrix} -A_i^T I_s \\ 0 \end{matrix}] \end{aligned} \tag{A.2}$$

$\mathcal{G}$, $\mathcal{L}$, $\mathcal{C}$ are conductance, inductance, and capacitance matrices and $G = A_g^T \mathcal{G} A_g$, $C = A_c^T \mathcal{C} A_c$. I_s, i_l, v_n are vectors of current source, inductance current variables, and nodal voltage respectively. For transient analysis using trapezoidal integration approximation over time interval $[kh, (k+1)h]$ is given by

$$\tilde{G}(\frac{x^{k+1} + x^k}{2}) + \tilde{C}(\frac{x^{k+1} - x^k}{h}) = \frac{b^{k+1} + b^k}{2}$$

Rewritten as:

$$(\tilde{G} + \frac{2}{h}\tilde{C})x^{k+1} - b^{k+1} = (-\tilde{G} + \frac{2}{h}\tilde{C})x^k + b^k \tag{A.3}$$

Substituting (A.3) into (A.2), and performing block matrix operations, we could obtain two equations as follows:

$$(G + \frac{2}{h}C)v_n^{k+1} + A_l^T i_l^{k+1}$$

$$= (-G + \frac{2}{h}C)v_n^k - A_l^T i_l^k - A_i^T (I_s^{k+1} + I_s^k) \tag{A.4}$$

$$-A_l v_n^{k+1} + \frac{2}{h}\mathcal{L} i_l^{k+1} = A_l v_n^k + \frac{2}{h}\mathcal{L} i_l^k \tag{A.5}$$

For reluctance matrix, $\mathcal{K}$ equals to $\mathcal{L}^{-1}$, and we define K=$A_l^T \mathcal{K} A_l$. Rewriting (A.5) as follows and multiply $\mathcal{L}^{-1}$, and A_l^T to both side.

$$2\mathcal{L}(i_l^{k+1} - i_l^k) = hA_l(v_n^{k+1} + v_n^k) \tag{A.6}$$

$$2(i_l^{k+1} - i_l^k) = h\mathcal{L}^{-1}A_l(v_n^{k+1} + v_n^k) \tag{A.7}$$

$$2(i_l^{k+1} - i_l^k) = h\mathcal{K}A_l(v_n^{k+1} + v_n^k) \tag{A.8}$$

$$2A_l^T(i_l^{k+1} - i_l^k) = hA_l^T\mathcal{K}A_l(v_n^{k+1} + v_n^k) \tag{A.9}$$

$$2A_l^T i_l^{k+1} = hK(v_n^{k+1} + v_n^k) + 2A_l^T i_l^k \tag{A.10}$$

Substitute (A.10) into (A.4), we arrive at:

$$(G + \frac{2}{h}C + \frac{h}{2}K)v_n^{k+1} = (-G + \frac{2}{h}C - \frac{h}{2}K)v_n^k$$

$$-2A_l^T i_l^k - A_i^T(I_s^{k+1} + I_s^k) \tag{A.11}$$

Appendix B

B.1 Non-orthogonal wire mutual inductance

Refer to figure B.1 for illustration.

$$2cos\epsilon = \frac{\alpha^2}{lm}, where \quad \alpha^2 = {R_4}^2 - {R_3}^2 + {R_2}^2 - {R_1}^2 \tag{B.1}$$

$$\mu = \frac{[2m^2({R_2}^2 - {R_3}^2 - l^2) + \alpha^2({R_4}^2 - {R_3}^2 - m^2)]l}{4l^2m^2 - \alpha^4} \tag{B.2}$$

$$\nu = \frac{[2l^2({R_4}^2 - {R_3}^2 - m^2) + \alpha^2({R_2}^2 - {R_3}^2 - l^2)]m}{4l^2m^2 - \alpha^4} \tag{B.3}$$

$${R_1}^2 = (\mu + l)^2 + (\nu + l)^2 - 2(\mu + l)(\nu + m)cos(\epsilon) \tag{B.4}$$

$${R_2}^2 = (\mu + l)^2 + \nu^2 - 2\nu(\mu + l)cos(\epsilon) \tag{B.5}$$

$${R_3}^2 = \mu^2\nu^2 - 2\nu\mu cos(\epsilon) \tag{B.6}$$

$${R_4}^2 = \mu^2 + (\nu + m)^2 - 2\mu(\nu + m)cos(\epsilon) \tag{B.7}$$

The general formula for mutual inductance is

$$\frac{M}{2cos(\epsilon)} = \tag{B.8}$$

$$0.001[(\mu + l)tanh^{-}1\frac{m}{R_1 + R_2} + (\nu + m)tanh^{-}1\frac{l}{R_1 + R_4} -$$

$$\mu tanh^{-}1 fracmR_3 + R_4 - \nu tanh^{-}1\frac{l}{R_2 + R_3}]$$

where dimension is measured in cm.

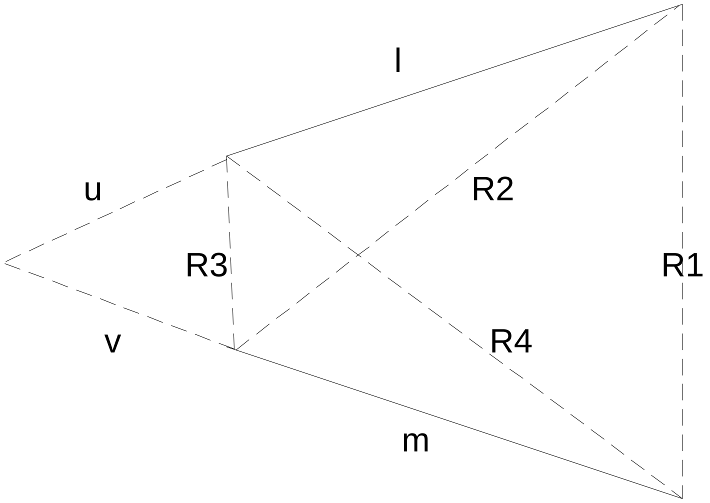


Figure B.1: Non-orthogonal inductance formula

Bibliography

[1] http://public.itrs.net/files/2002update/home.pdf.

[2] K. Gala, V. Zolotov, R. Panda, B. Young, J. Wang, and D. Blaauw. On-chip inductance modeling and analysis. In *DAC*, June 2000.

[3] M.W. Beattie and L.T. Pileggi. Inductance 101: Modeling and extraction. In *DAC*, June 2001.

[4] A.E. Ruehli. Inductance calculatioin in a complex integrated curcuit environment. *IBM Journal of Research and Development*, pages 470–481, September 1972.

[5] Z. He, M. Celik, and L.T. Pileggi. Spie: Sparse partial inductance extraction. In *DAC*, 1997.

[6] B. Krauter and L.T. Pileggi. Generating sparse partial inductance matrices with guaranteed stability. In *ICCAD*, November 1995.

[7] K. L. Shepard and Z. Tian. Return-limited inductances: A practial approach to on-chip inductance extraction. In *IEEE Trans. on CAD of integrated circuis and systems*, number 19(4), pages 425–436, April 2000.

[8] R. Panda B. Young J. Wang K. Gala, V. Zolotov and D. Blaauw. On-chip inductance modeling and analysis. In *DAC*, June 2000.

[9] M. Kamon, M.J. Tsuk, and J.K. White. Fasthenry: A multipole-accelerated 3-d inductance extraction program. In *DAC*, June 1993.

[10] A. Pacelli. A local circuit topology for inductive parasitics. In *ICCAD*, Nov 2002.

[11] C.-K. Koh G. Zhong and K.Roy. On-chip interconnect modeling by wire duplication. In *ICCAD*, Nov 2002.

[12] A. Devgan, H.Ji, and W. Dai. How to efficiently capture on-chip inductance effects: Introducing a new circuit element k. In *ICCAD*, November 2000.

[13] H. Ji, A. Devgan, and W. Dai. Ksim: A stable and efficient rkc simulator for capturing on-chip inductance effect. In *DAC*, June 2001.

[14] H. Kim T. H. Chen, C. Luk and C.C.P. Chen. Inductwise:inductance-wise interconnect simulator and extractor. In *ICCAD*, Nov 2002.

[15] C. Hoer and C. Love. Exact inductance equations for rectangular conductors with applications to more complicated geometires. *J. Res. Nat. Bureau of Standards*, 69C(2):127–137, April-June 1965.

[16] F. W. Grover. *Inductance calculations: Working Formulas and Tables*. Dover Publications, New York, N Y, 1946.

[17] H. Zheng, B. Krauter, M. Beattie, and L. Pileggi. Window-based susceptance models for large-scale rlc circuit analysis. In *DATE*, 2002.