Efficient PEEC-based Inductance Extraction using Circuit-Aware Techniques

Haitian Hu and Sachin S. Sapatnekar
Department of ECE, University of Minnesota, Minneapolis, MN 55455

Abstract

Practical approaches for on-chip inductance extraction to obtain a sparse, stable and accurate inverse inductance matrix K are proposed. The novelty of our work is in using circuit characteristics to define the concept of resistance-dominant and inductance-dominant lines. This notion is used to progressively refine a set of clusters that are inductively tightly-coupled. For reasonable designs, the more exact Algorithm 1 yields a sparsification of 97% for delay and oscillation magnitude errors of 10% and 15%, respectively, while the more approximate Algorithm 2 achieves up to 99% sparsification. An offshoot of this work is K-PRIMA, an extension of PRIMA to handle K matrices with guaranteed passivity.

1. Introduction

Inductance has grown in prominence with shrinking technologies and will become even more so in the future, thereby affecting the electrical behavior of on-chip wires. While treating inductance, it is convenient to classify on-chip wires into two types: *switching lines* (clock and signal nets), and *supply lines* (V_{dd} and ground lines).

Various efforts have been made in the past on modeling inductance. The PEEC model [1], where the loop current is assumed to return at infinity, can represent a complex multiconductor topology without predetermined current return paths. Formulae for partial self- and mutual inductance for typical structures are found in [2]. However, the PEEC model can result in a dense inductance matrix, *M*, that makes simulation computationally expensive. Although many entries in this matrix are small, zeroing them may make the inductance matrix non-positive semidefinite [3]. Several algorithms have been developed to sparsify *M* while maintaining its symmetry and positive semidefiniteness, such as the shift-and-truncate method [3,4] that uses spherical shells for current returns, return-limited inductances [5], and the block-diagonal approach [6]. The *K* matrix [7,8], defined as the inverse of *M*, has been proven to have better properties than *M* since it is diagonally dominant, in addition to being symmetric and positive semidefinite. Moreover, *K* can be easily sparsified like a capacitance matrix and for the same sparsification, can obtain a higher accuracy than an *M* matrix.

One major problem with most previous techniques is that they neglect circuit characteristics during extraction. Our *circuit-aware* approach explicitly incorporates the circuit environment during extraction. For example, when an inductive line is driven by a highly resistive driver, inductive effects would be suppressed by the driver. While a traditional approach would extract for all inductors, our approach examines the circuit context of an element and determines an appropriate level of accuracy of extraction. Comparable work in [4] extends [3] and heuristically determines the radius of the spherical shell using circuit constraints; however, since the shell radius must be the same for all inductors, the achievable sparsification is limited. Another recent approach [9] discards high resistance wires using a rules-based approach. However, it is not trivial to decide how these resistance values should be chosen, and even a relatively high resistance wire can be influenced by a nearby wire that is highly inductive. Our approach classifies the switching lines into two categories that are loosely defined as follows:

- ♦ inductance-dominant lines (ID lines): a self/mutual inductance of the line strongly affects a waveform in the circuit.
- ♦ resistance-dominant lines (RD lines): inductive effects are partially or completely damped out by the driver resistance, so that both the self- and mutual inductances associated with this line weakly impact other circuit waveforms

We will make this qualitative description quantitative in this paper.

The circuit-aware concept was first proposed in [10] to obtain a sparsified inductance matrix *M*, and this work improves upon [10]. Firstly, we develop two new circuit-aware algorithms that are specific to the element *K* (instead of the traditional PEEC inductance matrix *M*), thereby permitting greater sparsification. The algorithms use differing assumptions that trade off sparsity for accuracy. Secondly, as an offshoot of this work, we propose a simple way of using the *K* matrix within PRIMA [11] with guaranteed passivity.

Algorithm 1 works under the assumption that supply lines are imperfect conductors with their own *RKC*'s. The more approximate Algorithm 2 assumes that there is no $\Sigma_j L_{ij} (dI_j/dt)$ drop on the supply lines; however, it does *not* assume zero RC's on the supply lines, and it incorporates mutual inductances between supply and switching lines into the inductances of the switching lines. Algorithm 2 also assumes that the non-zero net magnetic field of aggressor lines and supply return currents goes up to a user-defined distance (which can be beyond the nearest supply lines, unlike [5]), which can be thought of as an order of the approximation. The two algorithms are suitable for inductance extraction for signal buses and clock nets in the presence of a power/ground grid structure.

The effectiveness of the circuit-aware algorithms is validated experimentally. Both algorithms provide a higher sparsification than the shift-and-truncate method [4]. The use of circuit-aware ideas is found to contribute to 80% of the improved sparsification, while the use of the *K* matrix contributes the remaining 20%. A comprehensive PEEC model, described in [10], is used in order to accurately estimate the current return paths and inductance effects. The circuit model includes grid supply lines that form the backbone of the power grid, dedicated supply lines, which are deliberately placed close to switching lines to provide good return paths for inductive currents, signal buses, clock nets, drivers and receivers, vias, pads and functional blocks connected to supply lines.

2. Proposed sparsification method

The motivation for the circuit-aware algorithm can be illustrated by considering the circuit equation for a signal line:

$$V_i = Z_i I_i + \Sigma_j L_{ij} (dI_j/dt) \qquad (1)$$

where V_i and I_i are the voltage across and the current flowing through line segment *i*, respectively; Z_i is the impedance of line segment *i*, not counting the inductance; L_{ij} is the self-inductance (if $i = j$) or mutual inductance (if $i \neq j$) between segment *i* and *j*; dI_j/dt is the time rate of change of the current in segment *j*. The significance of the inductance effect of an aggressor line segment *j* on a victim line segment *i* depends on the relative magnitudes of the terms in the above equation. Qualitatively speaking, strong inductance effects originate from the line segments that have a "large" dI_j/dt and take effect on line segments that are "not far away" and with a "large" value of L_{ij} and a "small" value of $Z_i I_i$. We start by using the *ID line criterion* to find ID lines with a large dI_j/dt, and then grouping nearby lines with large L_{ij} and small Z_iI_i values into a cluster, under a specified accuracy criterion.

2.1 ID line criterion

An important observation for developing circuit-aware algorithms is that inductance-dominant lines typically have a small transition time and a large oscillation magnitude and/or high frequency oscillation, so that they are the best candidates (due to their large value of dI_j/dt) for causing mutual inductance effects on other lines. To demarcate ID lines from RD lines, we use a relative criterion to define ID lines, called the *ID criterion*, described as follows. This criterion is applied individually to one line at a time to determine whether it is classified as ID or not; recall that the line is divided into segments.

A signal line is ID if the behavior of the output waveform in the presence of inductances (partial self- and mutual inductance between any two segments on that line) is significantly different, according to a specified metric, from the waveform when a pure RC model is used and inductances are ignored.

Our metric states that if the percentage variation in the oscillation magnitude is larger than a specified ε, or the delay of the output response is larger than a specified δ, then the line is ID. RD lines include all those lines that are not ID. Thus we classify all lines as either ID switching lines, RD switching lines or supply lines.

The ID criterion is pessimistic in labeling lines as ID since it works under the assumption that currents return at infinity. The partial inductance thus calculated overestimates the loop inductance, as the presence of nearby return paths can reduce the magnetic flux through the loop, thereby reducing the loop inductance to be lower than the partial inductance. By setting ε and δ appropriately, it can be ensured that ID lines are correctly identified, and no signal line that is marked as non-ID by the ID criterion will become ID later.

We use these ideas of RD and ID lines to identify clusters, each of which initially contains a single ID line, as identified above, and is subsequently grown further and possibly combined with other clusters and RD lines. Formally, we define a cluster as a group of line segments such that mutual inductances must be calculated between any pair in this group. A cluster can be seen as a small independent inductive system, and corresponds to a full inductance submatrix, and there is no mutual inductance between line segments within and outside a cluster. Any lines that are not contained in any cluster are eventually modeled as RC lines. Once these clusters have been formed, each cluster is approximated by a sparsified *K* submatrix that is guaranteed to be symmetric and positive semidefinite. *Therefore, by construction, the resulting sparse K matrix for the whole circuit is positive semidefinite and symmetric*.

In both algorithms, sparsification is carried out under a worst-case switching pattern and a set of worst-case switching currents drawn by the functional blocks. All of the inputs drivers switch simultaneously to excite currents in the same direction, resulting in the strongest inductive coupling, and the upper bounds of the driver sizes and input slopes are applied to each switching line. Under these conditions, a worst case K matrix is found that can safely be used under other switching patterns, driver sizes and input slopes.

2.2 The CMI operation

We define an operation CMI (choose mutual inductance) that can be applied between any two clusters, or between one cluster and supply and/or switching line(s) (which are modeled as RC-only) to test whether consideration of the mutual inductance is important or not.

Operation CMI: CMI is applied to two situations:

(a) Given two clusters, we compare the response in two cases:

Case 1: The two separated clusters are combined into one cluster and the mutual inductances between the clusters are considered.

Case 2: The mutual inductances between the clusters are ignored.

(b) Given one cluster and supply/switching line(s) modeled as RC-only, we compare the response in two cases:

Case 1: The line(s) is (are) grouped with the cluster and all mutual inductances between segments in this new group are considered.

Case 2: The mutual inductances between the cluster and the line(s) are ignored and the line is modeled as RC-only.

(We distinguish (a) from (b) as an RC-only line is not in a cluster.)

In each situation above, the operation proceeds by carrying out simulations for both cases and testing the delays and oscillation magnitudes of the outputs of switching lines in the two clusters or in the cluster and the switching line(s) added into the cluster. The simulations are carried out using K-PRIMA, described in Section 4. If the change in one of the oscillation magnitudes [delays] is larger than an ε [δ], we conclude that the mutual inductance between the clusters (or between the cluster and the supply and/or switching line(s)) is important, implying that the two clusters should be grouped into one cluster, or the supply and/or switching line(s) should be included into the cluster. CMI in situation (b) can be used to test the relation between the cluster and the supply and/or switching line(s), as shown in the next section.

2.3 Formation of clusters

Our approach is guided by a set of foundations that were inferred from exhaustive experiments described in [10].

Foundation 1: ID lines have strong mutual inductance effects on other ID lines. ID victims are easily influenced by aggressor lines. The more ID a switching line is, the more significant the effect.

Foundation 2: RD lines, especially highly RD lines, have very small mutual inductance effects on other lines. Moreover, highly RD lines are not easily influenced by aggressors, unless they are highly ID.

Foundation 3: Moderately ID lines may have mutual inductance effects on moderately RD lines.

Foundation 4: Supply lines have significant mutual inductance effects on nearby ID lines, which greatly reduces the inductance effect of ID lines.

Foundation 5: Supply lines do not have significant mutual inductance effects on RD lines.

Based on the above foundations, we separate the six possible combinations of mutual inductance interactions between ID lines, RD lines and supply lines into two classes:

- *Strong* mutual inductance interactions between ID lines and nearby ID lines or ID lines and nearby supply lines
- *Weak* mutual inductance interactions that include all the other interactions except between pairs of highly RD lines, or between highly RD and moderately RD lines.

Since strong mutual inductance interaction terms are the most important, circuit-aware algorithm first identifies them and forms clusters. These clusters are then iteratively grown by adding weak mutual inductance terms to achieve progressively higher accuracy.

A critical issue is to determine which supply lines, RD lines and other clusters on chip should be tested by the CMI operation for a given cluster. Next, we describe a method for choosing candidate lines and clusters to limit the number of tests to be performed. The lines and clusters found by this method have a large likelihood of having a significant mutual inductance interactions.

2.4 Choosing candidates for cluster enlargement

For any given cluster *C*, the detailed process for choosing candidate lines and clusters that it may be combined with is described below for the case of supply lines as an example. We divide the region outside cluster *C* into a set of concentric spheres S_i with inner radius R_{i_in}, outer radius R_{i_out} and thickness $\Delta R = R_{i\_in} - R_{i\_out}$, as shown in Figure 1. The inner radius of sphere S_{i+1} is the same as the outer radius of sphere S_i and all of the spheres share the same thickness except the innermost sphere. This sphere, with radius R_s, is centered at the cluster, and is the only sphere that is not hollow inside.

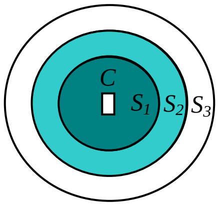

Fig. 1: Concentric spheres around a cluster *C*. Darker spheres are more likely to contain clusters with tight inductive couplings with *C*.

Checking for supply return paths starts from the nearest supply line to the cluster in the smallest sphere. Suppose there are N_i supply lines in sphere S_i, with the first being the nearest to, and the N_i^{th} being the farthest from *C*. We start the check with the first supply line by applying CMI between it and cluster *C*. If CMI indicates a strong effect, then the supply line is added into the cluster temporarily[1], followed by applying CMI on the enlarged cluster and the next nearest supply line in S_i; otherwise, CMI is applied between cluster *C* and the next nearest supply line in S_i.

If at least one supply line in sphere S_i has a strong influence on cluster *C*, we test the next sphere S_{i+1}. Otherwise, the check for supply return paths for cluster *C* is concluded. The effectiveness of this process depends on the value of R_s and ΔR, which are user-specified or empirical. The use of concentric spheres ensures that CMI checks are not unnecessarily considered between faraway lines; at the same time, by checking *all* clusters within distance ΔR, we allow for the possibility that interactions may take place between clusters that are not immediate neighbors.

3. Circuit-aware Algorithm 1

Oscillations on supply lines due to pad impedance, switching current drawn by the functional blocks connected to supply lines and the mutual inductance effects of nearby switching lines all serve to reduce the integrity of supply lines, which can potentially impact the output response noticeably. Therefore, for high-accuracy modeling, we should consider the *RKC*'s associated with the supply lines. Algorithm 1 uses such a model for supply lines, where the magnetic field of switching lines is considered to be capable of reaching infinity (or to the edge of the chip), and of influencing the responses of other switching lines and the integrity of faraway supply lines.

Since PEEC inductance estimates can be significantly reduced by the presence of nearby return paths, as we grow a cluster we *first attempt to include nearby supply lines*, and then determine which other clusters/RD lines it will affect, based on the updated inductive effects. If we do not do this, it is likely that we may overestimate the inductance effect of the aggressor cluster and include more interactions than necessary, thereby reducing the sparsification.

Algorithm 1 consists of the following stages:

Stage 1 - ID criterion: The ID criterion is applied to all switching lines, and each line identified as ID is placed in a separate cluster, called an ID cluster. Non-ID lines do not belong to any cluster.

Stage 2 – Initial return path identification: We will refer to the set of clusters at the beginning of this step as the "original clusters." The method for finding the supply return paths[2] is carried out by applying CMI on the cluster and the candidate supply line.

Once all of the original clusters have been processed, new ID clusters are formed by making the "temporary" additions of supply return paths to the cluster "permanent." Since the temporary additions could have caused a return path to be assigned to more than one cluster, clusters that share a line are now identified and combined into a larger cluster in order to ensure that no inductance terms are truncated. This method of adding and grouping is used throughout the algorithm. After this step, all the clusters include ID lines and their supply return paths, with only strong mutual inductance interactions among them.

Stage 3 – Cluster combination: The next step after finding supply return paths is to check if two ID clusters have a strong mutual inductance interaction between them by applying CMI on these two clusters, and grouping them together if so.

The candidate supply return paths for the combined clusters are then identified, and this process of cluster combination and return path identification repeated until no new interactions are found and no new clusters are formed. At the end of this process, all of the strong mutual inductance interactions have been identified.

Stage 4 – Identifying weak interactions: Next, we check for weak mutual inductance effects, corresponding to interactions between two nearby clusters or between one cluster and one nearby RD line, by applying CMI on the cluster and the RD line/clusters. After each such step, additional return paths are identified as before. This process of cluster combination and return path identification are repeated until no new mutual inductance interactions are found and no new clusters are formed.

In practice, the number of iterations to find strong and weak interactions, as well as the number of CMI operations in each iteration, is small. There are two reasons for this. Firstly, it is usually the case that there will be a small number of lines in a cluster, and it is highly unlikely that all lines will be grouped into a single final cluster. In a typical layout, for example, with a clock net or signal buses and a dense power grid distribution on the upper several metal layers, the influence of the magnetic field of ID lines is very localized, so that faraway lines do not have to be added to the clusters of the ID lines. Secondly, after each iteration, more than one line could be added into a cluster as clusters containing several lines are combined to create still larger clusters, so that cluster growth can be relatively rapid. Due to these effects, it was empirically observed that the total number of iterations and the number of CMI within each iteration could be bounded by a constant.

4. Implementation of K-PRIMA

As stated in Section 1, we use the *K* element [7] to represent the inductance system in our algorithm. The PRIMA algorithm [11] is adapted to generate a simulator, K-PRIMA, which can work with *K* elements and guarantee the passivity of the reduced system. This simulator is used extensively through the algorithm, in each CMI operation. Starting with the traditional inductance matrix *M*, simulation in each step of algorithm requires solving the following Modified Nodal Analysis (MNA) equations:

$$(G + sC)x = B \tag{2}$$

$$G = \begin{bmatrix} N & E \\ -E^T & 0 \end{bmatrix} \quad C = \begin{bmatrix} Q & 0 \\ 0 & M \end{bmatrix} \quad x = \begin{bmatrix} v \\ i \end{bmatrix} \tag{3}$$

where *(G+sC)* is the admittance matrix, *x* is a vector of unknown node voltages and unknown currents *i* of inductors and voltage

[1] The addition is "temporary" so that when a new cluster is considered, even supply lines that were previously incorporated into another cluster are considered as candidate return paths. As a result, the cluster formation does not depend on the sequence in which the clusters were processed, and lines may be temporarily assigned to more than one cluster.

[2] Note that this does not imply that these are the *only* return paths; other return paths are identified later.

sources *v*, *B* is a vector of independent time-varying voltage and current sources, and *M* is the traditional inductance matrix. The second set of equations implied by (2) is $-E^T v + s\,M\,i = 0$. Since *M* is symmetric and positive definite, it can be Cholesky-factored as $M = L\,L^T$. Substituting this and premultiplying by L^{-1}, we obtain

$$-L^{-1}\,E^T v + s\,L^T\,i = 0. \qquad (4)$$

Now defining $L^T i = i_b$, the MNA matrix can be written as:

$$G = \begin{bmatrix} N & E(L^T)^{-1} \\ -L^{-1}E^T & 0 \end{bmatrix} C = \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix} x = \begin{bmatrix} v \\ i_b \end{bmatrix} \qquad (5)$$

The off-diagonal submatrices have a negative transpose relationship, and the diagonal matrices are all positive semidefinite. This is a sufficient condition for the preservation of passivity of PRIMA [11].

However, finding *L* or L^{-1} can be computationally expensive unless done cleverly. Since $K = M^{-1} = (LL^T)^{-1} = L^{-T}\,L^{-1}$, L^{-1} is observed to also be a factor of the *K* matrix and both *K* and L^{-1} have the locality property. Our approach to find the sparsified L^{-1} is adapted from the method to find the K_{all} matrix in [7]. Since there are no mutual inductance terms between clusters, in constructing the *K* matrix, the window sizes/shapes may be different for different clusters as windowing operations are independently applied to them.

The following is our approach to construct the sparsified L^{-1} matrix (the notation used here is similar to [7]):

1. For each aggressor line segment *i*, find a traditional inductance matrix M_{small} including the line segments within the cluster that *i* belongs to that lie within in a small window size around *i*.
2. Cholesky-factorize M_{small} to find the Cholesky factor L_{small}, which is a lower triangular matrix.
3. Invert L_{small}.
4. Compose the large system L^{-1} by the column corresponding to the aggressor line segment in L^{-1}_{small}.

5. Circuit-Aware Algorithm 2

Algorithm 1 applies to the most realistic case with imperfect supply lines. However, for a very well designed supply grid or in cases where the requirement to the accuracy of modeling is not very high, the $\Sigma_j\,L_{ij}\,(dI_j/dt)$ drop on the supply grid can be assumed to be zero and the supply grid can be assumed to be perfect in this respect. However, we point out that we *do* consider the RC drops in supply lines, and that we *do not* consider the supply lines to be perfect ground planes. Algorithm 2 is a version of the circuit-aware algorithm under these assumptions and is developed as an extension of Algorithm 1. It is assumes that the currents return from the supply lines within a user-defined distance, within which we assume that the $\Sigma_j\,L_{ij}\,(dI_j/dt)$ drops on the supply lines are zero. Outside this distance, the net magnetic field of the aggressor lines and the return currents is zero. A similar assumption was also made in [5] and a primary difference (apart from the fact that [5] is not circuit-aware) is that our work allows currents to return from the supply lines beyond the nearest supply lines and up to a user-defined interaction distance, so that the switching lines be inductively coupled with other switching lines beyond the nearest supply lines.

The user-defined interaction distance is defined on the group of switching lines, called the aggressor group, between the nearest supply lines and is used as an approximation order. Under our assumption, Algorithm 2 generates a new and equivalent inductance system *Ms* by removing mutual inductance terms explicitly related to supply lines from the original system *M* and incorporating the effect of supply lines into the inductance values of *Ms*. By construction, we ensure that *Ms* is symmetric and positive semidefinite. Algorithm 1 is then applied to the new inductance system with a little adaptation. Figure 2 shows a schematic of a small example with an aggressor line, the aggressor group it belongs to, and the user-defined distance up to which the influence of the magnetic field of the aggressor group can reach. In the figure, supply lines are shown to be longer and thicker than the signal lines.

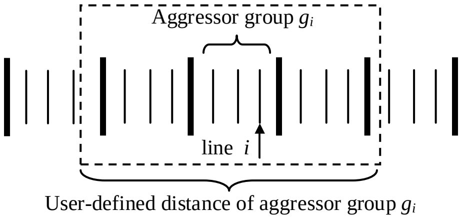

Fig. 2: A schematic showing an aggressor line and its corresponding aggressor group and the user-defined distance.

<u>*Definition and formation of the new matrix* M_s</u>

For a layout including both supply lines and switching lines, the device equation of inductors can be written as

$$\begin{bmatrix} V_{pg} \\ V_s \end{bmatrix} = s\begin{bmatrix} M_{11} & M_{12} \\ M_{12}{}^T & M_{22} \end{bmatrix}\begin{bmatrix} I_{pg} \\ I_s \end{bmatrix} \qquad (6)$$

where V_{pg} (V_s) represent the voltages difference across line segments on supply (switching) lines, respectively, I_{pg} (I_s) are the currents in these line segments on the supply (switching) lines, respectively, and M_{11}, M_{12} and M_{22} are inductance submatrices. For simplicity, we will work with *M* here instead of the *K* matrices, although the implementation uses the *K* matrix representation. Since the supply lines are assumed to have no $\Sigma_j\,L_{ij}\,(dI_j/dt)$ drop, V_{pg} is a zero vector. The first set of equations can then be written as $I_{pg} = -M_{11}^{-1}\,M_{12}\,I_s$. Substituting this into the second set of equations in (6) yields

$$V_s = s(M_{22} - M_{12}^T M_{11}^{-1} M_{12})I_s = sM_s I_s \qquad (7)$$

The calculation of M_s can be very efficient since *M* is symmetric and positive semidefinite and can be Cholesky factored as:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}\begin{bmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix} = LL^T$$

A few algebraic manipulations lead to the result

$$M_s = M_{22} - M_{12}{}^T M_{11}{}^{-1} M_{12} = L_{21}L_{21}{}^T + L_{22}L_{22}{}^T - L_{21}L_{11}{}^T(L_{11}L_{11}{}^T)^{-1}L_{11}L_{21}{}^T = L_{22}L_{22}{}^T$$

Since L_{22} and $L_{22}{}^T$ are triangular matrices, the computation for (7) is greatly reduced. It is easy to prove that the new inductance matrix M_s is symmetric and positive definite. We can think of M_s as an inductance matrix for a new inductance system that substitutes *M* and that has been demonstrated to have better locality. This locality provides further sparsification above and beyond that obtained by dropping inductance terms explicitly related to the supply lines.

1. Use the new values of self- and mutual inductance in M_s to find ID lines, and form ID clusters using the ID criterion.
2. Check all ID clusters to see if any two of them should be grouped into one larger cluster. At the end of this process, if any two of the newly formed clusters have common lines, group them into one cluster. Repeat step 2 until no new cluster is formed.
3. Test if any two clusters, or a cluster and an RD line, should be combined into a cluster. At the end of this process, if any two of the newly formed clusters have common lines, group them into one cluster. Repeat step 3 until no new cluster is formed.

Fig. 3: Outline of Algorithm 2

<u>*Description of Algorithm 2*</u>

Algorithm 2 is summarized in Figure 3. Unlike Algorithm 1, it does not work with the supply lines to add return paths to the cluster. All the other steps are similar to those in Algorithm 1, except that each inductance value comes not from *M*, but from M_s.

6. Experimental results

We have carried out a set of experiments on a 0.1μm technology. The topologies correspond to the top three metal layers of a five-layer metal structure, with wide and long switching lines being routed on M5. Supply lines in the vertical direction are routed in M5 and M3, while those in the orthogonal direction are on M4. A voltage swing of 1 V is used with a slope of 10 ps.

6.1 Accuracy comparisons

Two sets of experiments are performed in this section on two different configurations to compare the effectiveness of Algorithms 1 and 2. In Circuit 1, there are 10 vertical grid supply lines in M5, with 8 switching lines and 3 dedicated supply lines between the 5th and 6th grid supply lines. There are 10 vertical grid supply lines on M3 and 21 horizontal grid supply lines on M4. The *upper bounds on driver sizes* (which we will refer to henceforth, for convenience, as the driver sizes) of the 8 switching lines named, from left to right, S1 through S8, are 100×, 200×, 10×, 100×, 100×, 5×, 50×, and 200×, respectively. A line with a 200× driver size has a small driver resistance and is highly ID, while a line with a 5× driver is highly RD. The three dedicated supply lines are positioned, respectively, to the left of the first switching line, between the fourth and fifth switching lines, and to the right of the eighth switching line. Circuit 2 is identical to Circuit 1, except that all dedicated supply lines are removed, so that it is a "worse" design than Circuit 1.

In Circuit 1, the simulation results for the second switching line, which is one of the farthest switching lines from the dedicated supply lines and shows the largest inductance effect, are displayed in Figure 4. The waveforms shown correspond to the accurate response that considers all the inductance terms, to Algorithm 1, and to Algorithm 2. For the latter, two sets of user-defined distances are tested: in the first, this is set to be until the second nearest supply lines, i.e., all of the three dedicated supply lines are within the user-defined distance of each aggressor group, while in the second, the limit is set to the nearest supply lines. The errors in the 50% delay and oscillation magnitudes in all cases are summarized in Table 1. The ε and δ used in the CMI operation are 10% and 5% respectively. The larger user-defined distance can be seen to improve the accuracy of Algorithm 2, and illustrates that supply lines only weaken the magnetic coupling and do not fully block it. In general, if there are more aggressor groups nearby, this error may be larger and the necessary user-defined distance would be accordingly larger.

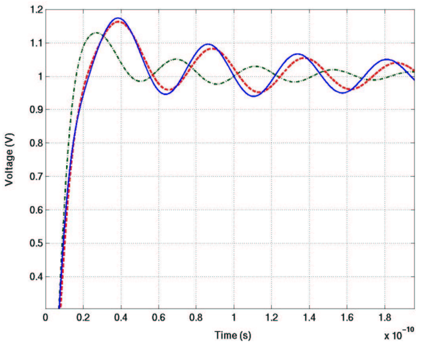

Fig. 4: Comparison of the output response with the accurate response for Circuit 1 (Solid line: accurate response, dashed line: Algorithm 1, dash-dot line: Algorithm 2).

The accurate waveform in Figure 4 is shown by the solid curve and yields a delay of 9.4ps and an oscillation magnitude of 170mV. Compared with the accurate response, the error in the 50% delay obtained by Algorithm 2 with the second nearest supply lines as the user-defined distance is only 6.3% but the error in the oscillation magnitude can reach 40mV, which is about 23% off. In comparison, Algorithm 1 is more accurate, with an error of under 10mV in the oscillation magnitude, but with a less sparse *K* matrix.

The error in the oscillation magnitude between the accurate waveform and that from Algorithm 2 implies that its underlying supply line assumption may not be good if a high accuracy in oscillation magnitude is desired in this circuit; however, the delays are acceptable in this experiment; and these assumptions result in a higher sparsification. Our results are consistent with the observations in [5], which sets the user-defined distance to be the nearest supply lines. The error in overshoot shown in [5] is about 45% (which is higher than our numbers), but the 50% delay is matched well in their work, as in ours. Algorithm 2 is sufficiently accurate for delay calculation, while Algorithm 1 ensures accuracy on the delay and oscillation magnitude over the entire waveform.

	Accurate Response	Algorithm1	Algorithm2	
			Nearest supply lines	Second nearest supply lines
50%delay	9.4ps	9.8ps	8.5ps	8.8ps
Osc.Mag.	170mV	160mV	110mV	130mV

Table 1: Delays and oscillation magnitudes for the accurate response, Algorithm1, and Algorithm2 (various user-defined distances).

R_s, the radius of the smallest sphere (defined in Section 2.4) when we choose the candidate lines and clusters, is 30mm and Δ*R*, the sphere thickness, is chosen to be 60mm. The window size for each line segment to construct sparsified *K* matrix in applying Algorithm 1 on Circuit 1 is such that in the direction of the lines, each segment only has a mutual inductance with the line segments nearest to it, while in the perpendicular direction, the window size is 25mm. Algorithm 2 finds that the mutual inductances of the nearest line segments and the second nearest line segments on the same line must be included in the window size for the above accuracy.

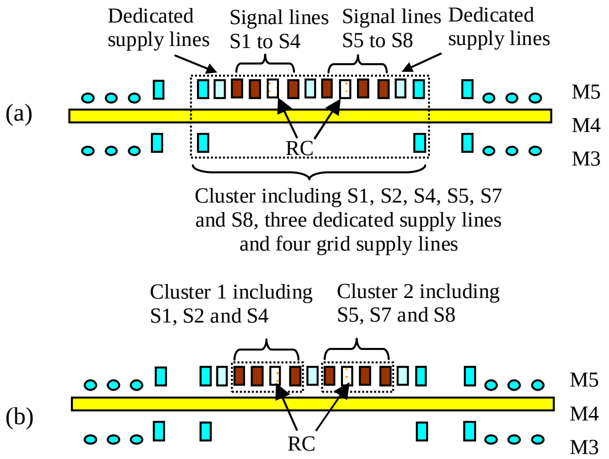
Dedicated supply lines
Signal lines S1 to S4
Signal lines S5 to S8
Dedicated supply lines
(a)
M5
M4
M3
RC
Cluster including S1, S2, S4, S5, S7 and S8, three dedicated supply lines and four grid supply lines
Cluster 1 including S1, S2 and S4
Cluster 2 including S5, S7 and S8
(b)
M5
M4
M3
RC

Fig. 5: Clusters for Circuit 1 in (a) Algorithm 1 and (b) Algorithm 2.

For Algorithm 1, as shown in Figure 5(a), all of the switching lines except lines S3 and S6, and the three dedicated supply lines and four nearest grid supply lines are included in one basic cluster, because the dedicated supply lines are shared by these switching lines. Two basic clusters for Circuit 1 in Algorithm 2 are formed as shown in Figure 5(b), with S1, S2 and S4 (with driver sizes of 100×, 200× and 100×, respectively) between the first and second dedicated supply lines being placed in one cluster, and S5, S7 and S8 (with driver sizes 100×, 50× and 200×, respectively) between the second and third dedicated lines in the second cluster. Lines S3 and S6, driven by 5 and 10 drivers, respectively, are RC only.

The imperfect integrity obtained from, for example, providing inadequate return paths, plays a significant role both in the inductance effects in a circuit and in the sparsity that can be obtained. To observe this, consider Circuit 2, which is a worse design of Circuit 1 due to the removal of all three dedicated supply lines, so that the nearby return paths are taken away. The oscillation magnitude of the second switching line jumps to 371mV and the 50% delay increases to 14.2ps. The window size and the clusters formed by Algorithm 1 and 2 for Circuit 2 are larger than that for Circuit 1. Even the mutual inductance with lines that would have been expected to be highly RD (such as lines S3 and S6) must be considered for the accurate modeling of the response of RD lines, and more grid supply lines must be included into clusters.

6.2 Sparsification comparisons

In this section, we compare the sparsification obtained from Algorithm 1 and the shift-and-truncate method under the same accuracy, as well as Algorithm 2 for four circuits, namely, Circuits 1 and 2 above, and two new layouts, Circuits 3 and 4. Circuit 3 is a good design with adequate return paths, while Circuit 4 is a poor design without the dedicated supply lines. Circuit 4 has 16 vertical grid supply lines in M5, with 4 switching lines between the 8th and 9th grid supply lines. There are 7 horizontal grid supply lines on M4 and 16 vertical grid supply lines on M3. The driver sizes for the switching lines named, from left to right, S1 through S4, are 220×, 150×, 200×, 5×, respectively. Circuit 3 also contains one dedicated supply line each on the left and right of the four switching lines.

	Circuit 1	Circuit 2	Circuit 3	Circuit 4
Algorithm 1	97%	87%	97%	83%
Algorithm 2	99%	98%	98.4%	97%
Shift-and-truncate	92.5%	75%	90%	68%

Table 2: Sparsification from Algorithm 1, Algorithm 2 and the shift-and-truncate method in Circuit 1, 2, 3 and 4.

The sparsifications obtained are summarized in Table 2. For a good design such as Circuit 1, Algorithm 2 achieves 99% sparsification, while the more accurate Algorithm 1 obtains 97%, which is expectedly lower. It was found that for both of our circuit-aware algorithms, the sparsifications for Circuit 2 are worse than those for Circuit 1. For Algorithm 2, if the sparsification is still relatively high at 98%, the error in the 50% delay reaches 12% and the error in the oscillation magnitude is larger than 150mV. For Algorithm 1, to obtain the 15% error in oscillation magnitude and 10% error in delay, the sparsification is found to be 87%. Under the same constraint, the sparsification for the shift-and-truncate method was even lower, at 75%. Similar trends are seen for Circuits 3 and 4.

6.3 Interpretation of the results

Several conclusions can be drawn from our results. Firstly, for accurate modeling, the influence of supply lines must be considered. Hence, Algorithm 1 provides an accurate waveform, but Algorithm 2 works well for delay estimation and produces a higher sparsification. Secondly, under the same accuracy, the shift-and-truncate method yields a lower sparsification than Algorithm 1. On an average, we found that roughly 80% of the improvements in sparsity were due to the use of circuit-aware methods, and 20% to the use of the K matrix instead of the M matrix.

The circuit-aware method also provides pointers on how to optimize inductance effects in a system. During the procedure, if two clusters are to be grouped into one cluster, the mutual inductance interactions between these two clusters, especially between ID lines in each cluster, are strong. One way to reduce these inductive interactions is to add dedicated supply lines next to ID lines. Algorithm 2 may accurately analyze such an optimized circuit, because the more localized the magnetic field of ID lines is, the more accurate the results of Algorithm 2 will be. Therefore, an interesting conclusion is that reducing inductance effects is not only useful in a circuit context but also in the ease of analysis.

7. Conclusion

Two circuit-aware based sparsification methodologies for fully coupled PEEC *K*-element representations for an inductive system are proposed by analyzing the circuit characteristics and clustering the inductances according to their relative importance to the circuit. In both algorithms, all of the switching lines are classified as ID or RD lines. Strong couplings are resolved first and weak couplings are then added to the clusters. Experimental results show the effectiveness of our method compared with the shift-and-truncate method. Algorithm 1 is designed for any signal bus and clock net circuit and provides a high accuracy but with a lower sparsification than for Algorithm 2. However, the latter works well in a design with good return paths.

References

[1] A. E. Ruehli, "Inductance Calculations in a Complex Integrated Circuit Environment," *IBM Journal of Research and Development*, pp. 470-481, vol. 16, No. 5, September 1972.

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

[3] B. Krauter and L. T. Pileggi, "Generating Sparse Inductance Matrices with Guaranteed Stability," *Proc. ICCAD*, pp. 45-52, 1995.

[4] Z. He, M. Celik and L. T. Pileggi, "SPIE: Sparse Partial Inductance Extraction," *Proc. DAC*, pp. 137-140, 1997.

[5] K. L. Shepard and Z. Tan, "Return-Limited Inductances: A Practical Approach to On-Chip Inductance Extraction," *Proc. CICC*, pp. 453-456, 1999.

[6] K. Gala *et al.*, "Inductance 101: Analysis and Design,*" Proc. DAC*, pp. 329-334, 2001.

[7] A. Devgan, H. Ji and W. Dai, "How to Efficiently Capture On-Chip Inductance Effects: Introducing a New Circuit Element K," *Proc. ICCAD*, pp. 150-155, 2000.

[8] H. Ji, A. Devgan and W. Dai, "KSPICE: Efficient and Stable RKC Simulation for Capturing On-Chip Inductance Effect," Tech. Rep. UCSC-CRL-00-10, UC Santa Cruz, Santa Cruz, CA, 2000.

[9] H. Smith *et al.*, "R(f)L(f)C Coupled Noise Evaluation of an S/390 Microprocessor Chip," *Proc. CICC*, pp. 237-240, 2001.

[10] H. Hu and S. Sapatnekar, "Circuit-Aware On-chip Inductance Extraction," *Proc. CICC*, pp. 245 - 248, 2001.

[11] A. Odabasioglu, M. Celik and L. T. Pileggi, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *Proc. ICCAD*, pp. 58-65, 1997.