A Precorrected-FFT Method for Simulating On-chip Inductance

Haitian Hu, ECE Department, University of Minnesota, Minneapolis, MN 55455
David T. Blaauw, EECS Department, University of Michigan, Ann Arbor, MI 48104
Vladimir Zolotov, Kaushik Gala, Min Zhao, Rajendran Panda, Motorola, Inc., Austin, TX 78729
Sachin S. Sapatnekar, ECE Department, University of Minnesota, Minneapolis, MN 55455

Abstract
The simulation of on-chip inductance using PEEC-based circuit analysis methods often requires the solution of a subproblem where an extracted inductance matrix must be multiplied by a current vector, an operation with a high computational cost. This paper presents a highly accurate technique, based on a precorrected-FFT approach, that speeds up this calculation. Instead of computing the inductance matrix explicitly, the method exploits the properties of the inductance calculation procedure while implicitly considering the effects of all of the inductors in the layout. An optimized implementation of the method has been applied to accurately simulate large industrial circuits with up to 121,000 inductors and nearly 7 billion mutual inductive couplings in about 20 minutes. Techniques for trading off the CPU time with the accuracy using different approximation orders and grid constructions are also illustrated. Comparisons with a block diagonal sparsification method in terms of accuracy, memory and speed demonstrate that our method is an excellent approach for simulating on-chip inductance in a large circuit.

1. Introduction

The fast and accurate simulation of on-chip inductance is a growing problem as technologies shrink further and low-k dielectrics are used to diminish capacitive effects. Inductive effects are important in determining power supply integrity and timing/noise analysis, especially for global clock networks, signal buses and supply grids for high-performance microprocessors.

One of the major problems in determining inductance has been associated with the fact that wire inductances are defined over current loops, and that the current loops are dependent on the circuit context of the switching wires. The partial element equivalent circuit (PEEC) model [1] has been developed to solve this chicken-and-egg problem and does not require the current return paths to be predetermined. The PEEC approach introduces the concept of partial inductance of a wire or a wire segment, corresponding to a return path at infinity. The partial self-inductance is defined as the inductance of a wire segment that is in its own magnetic field, while the partial mutual inductance is defined between two wire segments, each of which is in the magnetic field produced by the current through the other. For two wire segments k and m, the partial mutual inductance is given by:

$$M_{km} = \frac{1}{I_m a_k}\left(\int_{a_k}\int_{l_k} \vec{A}_{km} \bullet d\vec{l}_k da_k\right) = \frac{m_0}{4\pi a_k a_m}\int_{a_k}\int_{a_m}\int_{l_k}\int_{l_m} \frac{d\vec{l}_k \bullet d\vec{l}_m}{r_{km}} da_k da_m \quad (1)$$

where l_i and a_i (i=k or m) are the length and cross section area of wire segment i. r_{km} is the distance between any two points on segment k and m. $\vec{A}_{km}$ is the magnetic vector potential along segment k due to the current I_m in segment m, given by:

$$\vec{A}_{km} = \frac{m_0}{4\pi a_m}\left(\int_{a_m}\int_{l_m} \frac{I_m}{r_{km}} d\vec{l}_m da_m\right) \quad (2)$$

Here, simplified closed-form formulae for partial self- and mutual inductances of typical wire topologies that appear in integrated circuit environments are available in [2].

One drawback of using the PEEC method directly is that it results in a dense inductance matrix that causes a high computational overhead for a simulator. Although many entries in this matrix are small and have negligible effects, zeroing them out may cause the resulting inductance matrix to lose its desirable positive definiteness property [3], which is a necessary condition for the matrix to represent a physically realizable inductor system. Several efforts have been made to sparsify the inductance matrix while maintaining this property, such as the shift-and-truncate method [3,4], return-limited inductances [5], block diagonal method [6] and K matrix [7,8].

The shortcomings common to all of these methods are twofold. First, all these methods localize the magnetic field by a window size outside which couplings may be ignored. The principal problem is that it is difficult to definitively demarcate a region such that an aggressor wire segment outside this local interaction region is too weak to have a significant effect on a victim wire segment within it. Second, although the individual couplings that are ignored may be small, it is difficult to determine the cumulative effect of ignoring a larger set of such couplings without any knowledge of the current distributions.

FastHenry [9] is a multipole-accelerated method for inductance extraction. However, it works in frequency domain and ignores the effects of capacitance on the estimation of current return path. In order to obtain the time domain simulation, an accurate compact model has to be constructed, which is not an easy procedure.

In this paper, we propose a precorrected-FFT method that, instead of entirely dropping long-range couplings, approximates them, thereby overcoming the two shortcomings existing in the sparsification of inductance matrices. The main idea of this method is to represent the long-range part of the vector potential by point currents on a uniform grid and nearby interactions by direct calculations. The grid representation permits the use of the discrete Fast Fourier Transform (FFT) for fast potential calculations. Because of the decoupling of the short and long-range parts of the potentials, this algorithm can be applied to problems with irregular discretizations.

The basic precorrected-FFT method presented in this paper is inspired by the method in [10] for capacitance extraction, which also demonstrates that for many realistic structures, the precorrected-FFT method is faster and uses less memory compared with the multipole-accelerated method. In our work, the precorrected-FFT method is modified to be applied in a different context that is specific to the requirements of simulation of on-chip inductance, so that this work is by no means a mere incremental improvement. Unlike [10], we do not focus on extracting an inductance matrix M, but rather, directly consider how the inductance matrix is used in fast simulation algorithms.

This research was supported in part by the SRC under contract 99-TJ-714 and by the NSF under award CCR-0098117.

As described in Section 2, many simulators do not require M to be explicitly determined, but instead, require the computation of the product of M with a current vector I. The approach developed in this paper accelerates the procedure that is used to directly determine the $M \times I$ product without explicitly finding M. Several considerations are incorporated to make the algorithm efficient and applicable to large industrial circuits and complicated layouts. First, since mutually perpendicular segments do not have any inductive interactions, it is possible to apply the precorrected-FFT method to wire segments in the two perpendicular directions separately. This simplification is applicable to inductance systems and not to capacitance system. Second, different from the derivation of 2D integration in the capacitance extraction, the application of the precorrected-FFT in inductance problem involves a complicated derivation of 3D integration. Significant effort has to be made to obtain the exact and compact closed form formulae for accurate and efficient simulations. Third, since IC chips typically have much larger sizes in the two planar dimensions than in the third (i.e., they tend to be "flat"), we show that a two-dimensional grid may be used instead of a three-dimensional grid.

A comprehensive PEEC model, as described in [6], is used in this paper. We demonstrate the application of the precorrected-FFT method within a simulation flow based on PRIMA [11], on circuits of up to 121,000 inductors in PEEC model and nearly 7 billion mutual inductive couplings. It is the first implementation that incorporates accelerated PEEC approach and PRIMA to investigate industrial sized problems and give out time domain simulations. These experiments demonstrate the speed, memory consumption and accuracy of the precorrected-FFT method as compared to the block diagonal method [6], that is a heuristic sparsification technique based on a simple partition of the circuit topology, neglecting mutual inductances between partitions.. We also illustrate how tradeoffs may be made in order to obtain higher speed implementations with a small reduction in accuracy.

2. Motivation and problem formulation

It is well known that the basic PEEC model results in dense inductance matrices. The partial inductances of an n-wire segment system can be written as an $n \times n$ symmetric, positive semidefinite matrix $M \in R^{nxn}$, which may be incorporated into a circuit model of R, L, C and active elements in the circuit. If the circuit is linear, it can be solved efficiently using model order reduction techniques such as AWE [12] or PRIMA. In either method, we must calculate moments, which requires finding the product of M with a known current vector $I \in R^{nx1}$:

$$M \times I = \begin{bmatrix} M_{11} & M_{12} & \cdots & \cdots & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & \cdots & \cdots & M_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & M_{km} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ M_{n1} & M_{n2} & \cdots & \cdots & \cdots & M_{nn} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_m \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^{n} (\frac{1}{a_1} \int \vec{A}_m \bullet d\vec{l}_1 da_1) \\ \sum_{m=1}^{n} (\frac{1}{a_2} \int \vec{A}_{2m} \bullet d\vec{l}_2 da_2) \\ \vdots \\ \sum_{m=1}^{n} (\frac{1}{a_k} \int \vec{A}_{km} \bullet d\vec{l}_k da_k) \\ \vdots \\ \sum_{m=1}^{n} (\frac{1}{a_n} \int \vec{A}_{nm} \bullet d\vec{l}_n da_n) \end{bmatrix} \quad (3)$$

Here, we assume that I_m is the fictitious current in segment m and $\vec{A}_{km}$ is the magnetic vector potential on wire segment k due to I_m and can be determined by the expressions in (2). Each entry M_{km} in matrix M is the partial inductance between wire segment k and m, given by (1) and can be calculated using empirical [2] or closed form [13] formulae. The k^{th} entry in the $M \times I$ product, corresponding to the victim wire segment k, is $\sum_{m=1}^{n} M_{km} I_m = \sum_{m=1}^{n} (\frac{1}{a_k} \int \vec{A}_{km} \bullet d\vec{l}_k da_k)$. It is the summation of the integration of the magnetic vector potential over wire segment k caused by the current in each aggressor wire segment.

If the dense inductance matrix M is used, the computational cost for the matrix-vector product is very high: for a system with n variables, this is $O(n^2)$. The larger the circuit, the larger may be the number of moments and ports, and the heavier is the overhead of calculating this matrix-vector product. Therefore, methods for sparsifying the M matrix have been widely understood as being vital to solving systems with inductances in an efficient manner.

On closer examination, we observe that in order to solve the circuit, it is not M that needs to be calculated, but the product of M with a given current vector I. This motivates our work, and we present a method to efficiently find the product of M with a given current vector, using the precorrected-FFT approach to accelerate the computation. In this work, we use PRIMA as the simulation engine to test the results of the algorithm. This algorithm can also be incorporated in time domain simulators.

3. Precorrected-FFT method

The detailed explanation of the precorrected-FFT method can be found in [10] for capacitance extraction. Here, we present a simple description of this method in the magnetic field environment. The precorrected-FFT method is based on dividing the region under analysis into a grid. In the description of the algorithm, we will begin by using a three-dimensional grid, although we will show in the next section that in practice, a two-dimensional grid can also work well in an integrated circuit environment.

Consider the three-dimensional topology of wires that represents the circuit under consideration. After the wires have been cut into wire segments to be represented using the PEEC model, the circuit can be subdivided into a $k \times l \times m$ array of cells, each containing a set of wire segments. The contribution to the values of $\sum_{m=1}^{n} (\frac{1}{a_k} \int \vec{A}_{km} \bullet d\vec{l}_k da_k)$ of wire segments within a cell under consideration (the "victim cell") that is caused by wires in other cells (the "aggressor cells") can be classified into two categories: long-range interactions and short-range interactions. The central idea of the precorrected-FFT approach is to represent the current distribution in wire segments in the aggressor cell by using a small number of weighted point currents on the grid that can accurately approximate the vector potential for faraway victim cells. After this, the potential at grid points caused by the grid currents is found by a discrete convolution that can be easily performed using the FFT.

There are four steps in the precorrected-FFT approach to calculate $M \times I$:

1. **Projection:** The first step in the precorrected-FFT algorithm is to construct the grid projection operator W. Using W, the long-range part of the magnetic vector potential due to the current distribution in a given cell can be represented by a small number of currents lying on grid points throughout the volume of the cell. Thus, the real current distribution can be replaced by a set of grid point currents:

$$I_g = W I_r \quad (4)$$

where I_g and I_r are the grid current vector and real current vector, respectively. The boundary condition that is maintained during

projection is that the vector potentials at a set of test points on a sphere surrounding the cell should match the vector potentials due to the actual wires. Since the grid currents are a representation of the real current distribution, the grid can be coarser or finer than the actual problem discretization.

2. **FFT:** Once the real currents are projected to the grid, the grid potentials due to the grid currents are computed through a multi-dimensional convolution, given by:

$$A_g = HI_g \qquad (5)$$

where A_g is the grid potential and H is the contribution to the grid potential induced by unit point currents at grid points. This convolution can be calculated very fast by a multi-dimensional FFT computation, which proceeds by automatically considering all pairs of aggressor-victim combinations within the grid.

3. **Interpolation:** After the grid potential is calculated using the FFT, the values of $\sum_{m=1}^{n}(\frac{1}{a_k}\int \vec{A}_{km} \bullet d\vec{l}_k da_k)$ over victim conductors can be obtained through interpolation of the potentials on grid points throughout the cell that the victim conductor lies in. This step is basically the inverse process of the projection step, and the interpolation operator is W^T, which is the transpose of the projection operator and can be obtained by the theorem, proved in [10].

4. **Precorrection:** The grid representation of the current distribution in a cell is only accurate for potential calculations that correspond to long-range interactions. In practice, nearby interactions have the largest contribution to the total induced potentials, and therefore, these must be treated directly and accurately. Since the nearby interactions have already been included in the potential calculation after the above three steps, the last step constructs the precorrection operator $\tilde{M}$ which subtracts this inaccurate part from the result of the interpolation step before the accurate measure of nearby interactions is added in.

Combining the above steps, the induced voltages are:

$$V = MI = (\tilde{M} + W^T HW)I \qquad (6)$$

where W and $\tilde{M}$ are both sparse matrices, and H can also be constructed as a sparse matrix for an efficient implementation of FFT.

Since VLSI chips are thin and flat, we choose to use three-dimensional grid but with only one cell in the $\hat{z}$ (thickness) direction. There are three parameters that need to be determined before the precorrected-FFT algorithm is applied to a circuit: p, q and d. Here, parameter p is the number of grid points on each edge of a cell, while parameter q is the number of nearby cells which are considered in the precorrection step. The first nearest neighbors to each cell are defined as all cells that have a vertex in common with the considered cell, including the cell itself. Parameter d is the cell size, defined as the length of an edge of a cell in the $\hat{x}$ and $\hat{y}$ directions, which we will take to be equal. If n is the number of wire segments in the circuit, the computational complexity of the entire precorrected-FFT procedure is $O(n \log n)$ [10]. If p and q are fixed, there is an optimal cell size that yields the minimum value of cost. In this sense, the method for choosing the cell size is somewhat easier and more reliable than the methods used in [4][6] to find the local interaction region, because in the precorrected-FFT method, we only need to look for a minimum value of CPU or memory cost and a consideration of accuracy is relatively easier to define.

4. Experimental results

A set of experiments is carried out on a 400MHz Sun UltraSparc-II computer server to test the accuracy of the response from the precorrected-FFT method, and to compare the results with those of the block diagonal method [6] in terms of accuracy, speed and memory cost. The test circuit is a four metal layer conductor structure on layers M6, M7, M8 and M9 of a nine-layer chip, as illustrated in the top view of the structure in Figure 1. It lies within an area whose width is 330μm and thickness is 5μm. The circuit consists of three parallel signal wires, each with 0.8μm width, 0.8μm spacing and 0.5μm thickness. The power/ground wires are distributed densely in the four layers and the signal wires are on M8. The width of the test circuits is fixed throughout the experiments and the length changes along with the length of the signal wires in different experiments. The driver sizes for the three signal wires are identical and are altered with the wire length in order to set the near-end transition time to 40ps. The drivers are made to switch at the same time so that the inductance effect is maximized and the error incurred by the precorrected-FFT method can be determined for a worst-case condition. In the last part of this section, the experiments on a large industrial clock net are carried out the test the efficiency of the precorrected-FFT method in on-chip inductance simulation.

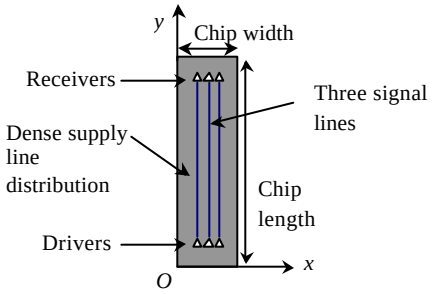

Figure 1: Top view of the test chip with three parallel signal lines on M8. The dark background represents the dense distribution of supply lines throughout the four metal layers. (Not to scale)

4.1 Accuracy of the precorrected-FFT method

In these experiments, p is set to 4, and the nearest neighbors and the next nearest neighbors are included in the direct interaction region. The cell sizes in the x and y direction are each chosen to be 15μm, while in the thickness direction, it is set to 7μm, such that the test structure is at the center of the cell. The radius of the collocation sphere is chosen to be 2.5 times the cell size. A simulation for the same circuit is also carried out with the block diagonal approximation. The partition size in the block diagonal approach is 180μm×150μm, which is much larger than the direct interaction region of 75μm×75μm. Figure 2 shows a comparison of the results from the precorrected-FFT and block diagonal methods with the accurate waveforms for 900μm long wires, showing waveforms at both the driver and receiver sides of the middle wire. The accurate waveforms are obtained by using the full inductance matrix in PRIMA without any approximation, while the approximate waveforms using the precorrected-FFT or block diagonal method with the same PRIMA simulator. In all the experiments in Section 4.1 and 4.2, there are 13 ports and the number of moments per port in PRIMA implementation is 5, as in [6], and it has been demonstrated that the response from the reduced order model converges here even if more moments are used in the simulation.

There are six waveforms in Figure 2, although only four are clearly visible since the waveforms from the precorrected-FFT almost completely overlap with those from the accurate simulation. The block diagonal waveforms at the near and far ends are marked (a) and (b), respectively. The largest error in the response from precorrected-FFT is less than 1mV. With about 100mV oscillation magnitude induced by inductance, the relative error of the oscillation magnitude is 0.1%. The relative error in the 50% delay for the response from precorrected-FFT is even smaller. Even though for a given victim line segment, more aggressor line segments are considered in the direct interaction region in the block diagonal method than in precorrected-FFT, the error in the response from the former is still larger than that of the latter. The reason for this is that the accumulated errors in the block-diagonal approach caused by the dropped mutual inductance terms is significant.

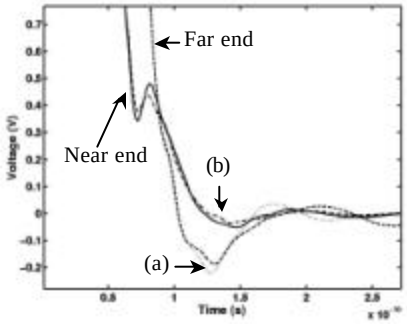

Figure 2: Comparison of waveforms from the precorrected-FFT and the accurate simulation at the driver and receiver sides of the middle wire. Waveforms from the precorrected-FFT and the accurate simulation are indistinguishable.

Because of the high accuracy that can be obtained by the precorrected-FFT method for this example, we observe that we can sacrifice some of the accuracy for higher speed. Different orders of approximation are tested to study the relation between speed, memory requirements and accuracy. The layout tested is similar to the above experiment but the length of the signal wires is extended to 5400μm, which is the largest tested wire length, so as to show the largest reduction in accuracy with the coarsening of the grid. Since there are more than 31,000 inductors in this circuit, including all of the inductors of signal wires and supply wires, nearly one billion mutual inductances are required for accurate simulation. It is therefore impossible to simulate for the accurate waveforms even in PRIMA, let alone in the time domain, and the use of an approach such as ours is essential. To simulate the response most accurately, p is set to 4, the cell size is set to be 15μm, and the first, second and third nearest neighbors are considered in the precorrection step. The response obtained from this setup can reasonably be considered as accurate.

Other precorrected-FFT simulations are carried out with lower accuracy and a coarser grid, where only the nearest neighbors are considered in the direct interaction region and the cell size is 30μm, which is double that in the above experiment. The cell size and the size of the direct interaction region are fixed in these experiments. The grids are variously chosen to be three-dimensional with p = 4, p = 3, p = 2, and two-dimensional with p = 4, p = 3, p = 2. The two-dimensional grid is in the plane that is parallel to the x-y plane and through the point that corresponds to the mid-point of the thickness of the test structure. There is only one grid point in the thickness direction. In the two-dimensional case, the collocation sphere reduces to a collocation circle in the x-y plane.

It is expected that reducing the problem to 2-D, using larger cell sizes and smaller values of p, and reducing the size of the direct interaction region will each contribute to some loss in accuracy, with an a faster computational speed and reduced memory requirements. The waveforms obtained at the driver and receiver sides of the middle wire with different levels of accuracy, corresponding to p = 2, 3 and 4 are very close and are not shown here. We find that the error in the 50% delay is insignificant for the three cases, but the relative error corresponding to the overshoot/undershoot is discernible, and is listed in the last column of Table 1. This table also lists the memory requirements and speed for each level of approximation. The setup time is the most time-consuming step in the entire algorithm, and is further divided into two parts. The first part corresponds to the calculation of the inductance values needed for the construction of the precorrection matrix, which is equal for each order of approximation, while the second relates to the time required for the calculation of the W, H and $\tilde{M}$ matrices. For p = 3 under a 3-D grid, the error at the peak is less than 1mV. The relative error in the oscillation magnitude at that point is 1%, while the speed is increased by 45% as compared with the accurate result. If p is further reduced to 2 under a 3-D grid, the error is 9mV but the speed is improved by an additional 16% compared to the p = 3 case. The 2-D grid representation with p = 2 results in the largest error of about 10mV and a similar relative error, but the speed is increased only by 6% as compared to its 3-D counterpart. The reason for this relatively low improvement is that in the case of p = 2, the precorrected-FFT is rather fast and the time consumed in the calculation of W, H and $\tilde{M}$ matrices is only a small part of the total setup time, so that even a large increase in the speed of calculation of W, H and $\tilde{M}$ matrices will not yield a significant reduction of the total run time. Another reason is that the number of grid points per cell is only reduced by half here by going from the three dimensions to two. On the other hand, if we reduce the 3-D grid to 2-D with p=4, the speed can be increased by 22% because the number of grid points per cell is reduced from 4^3=64 to 4^2=16, and the time required for the calculation of W, H and $\tilde{M}$ matrices plays a more important role in the total setup time. In this case, the accuracy is still high even under a 2-D grid. The memory requirements show similar trends.

4.2 Comparison with the block diagonal method

In this section, the memory consumption and speed of these two methods are compared for structures of different wire lengths, using a Matlab implementation. Performance results using an optimized C++ implementation are reported in Section 4.3. The lengths of the signal wires in different experiments are set to 900μm, 1800μm, 3600μm, 4500μm and 5400μm. The block diagonal partition size is chosen to be 180μm ×150μm, and for the precorrected-FFT method, a 2-D grid is imposed with p=2, and the first nearest neighbors are considered for the precorrection step. The cell size is set to 30μm. Figure 3 shows the waveforms computed by the two methods at the receiver end of the middle wire for wire lengths of 900μm and 5400μm. The accuracy, memory requirements and speed for different wire lengths for the block diagonal and precorrected-FFT methods are listed in Table 2. For the wire lengths of 900μm and 1800μm, the results of the precorrected-FFT and block diagonal methods are

similar to each other, and the block diagonal method is faster. However, as the wire length increases, the differences in the 50% delay and oscillation magnitude are larger. For example, the 50% delay calculated by the precorrected-FFT and block diagonal methods differ by about 5% for a wire length of 3600μm. The difference increases to 8% and 12.5% for lengths of 4500μm and 5400μm, respectively. For wire lengths that exceed 1800μm, the precorrected-FFT and block diagonal methods perform their computations at approximately the same speed, but the former has nearly half the memory requirements as the latter since the partition size for the block diagonal method is much larger than the direct interaction region in the precorrected-FFT, due to which the number of inductances per wire segment to be calculated by the former is much larger than that for the latter. As the circuit size increases, the setup time and memory are seen to increase at a faster rate for the block diagonal method.

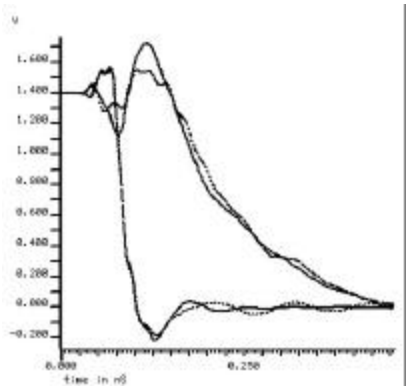

Figure 3: Simulation results at the receiver side of the middle wire from the precorrected-FFT and block diagonal methods for different wire lengths. (a) 900μm, PC-FFT (b) 900μm, block diagonal (c) 5400μm, PC-FFT (d) 5400μm, block diagonal.

Similar trends are seen for the differences in the oscillation magnitude as for 50% delay. The precorrected-FFT method predicts a more reasonable trend in the overshoot magnitude for different wire lengths: the overshoot increases as the wire length is increased from 900μm to 1800μm, and then decreases gradually as the wires grow longer and resistive effects take over. However, the trend predicted by the block diagonal method is different: the overshoot magnitude increases from 900μm to 1800μm long wires, and then decreases if the wire length increases from 1800μm to 4500μm, as in the case of the precorrected-FFT method. However, when the wire length increases from 4500μm to 5400μm, the overshoot is not reduced but is increased in the block diagonal method, which is clearly inconsistent with expectations. The differences between the results from the two methods are larger for longer wires.

Table 3 lists the overshoots and the run time of the responses at the receiver side of the 5400μm wire calculated by the precorrected-FFT and block diagonal methods, with different partition sizes of 30μm×30μm, 180μm×150μm, 330μm×150μm, 330μm×300μm, 330μm×600μm and 330μm×900μm. It is clear that the overshoots given by the block diagonal method do not converge as the block size is increased, and vary somewhat unpredictably. When the partition width increases from 180μm to 330μm, the 300mV bump disappears: the reason may be that more power/ground wires are included in each partition, and the inductance effect is greatly reduced. If the partition length is increased from 150μm to 300μm and then to 600μm and 900μm, with a 330μm partition width, the overshoot increases and nears the result from the precorrected-FFT method. It is impractical to increase the partition size further because the simulation time for 330μm×600μm partition is 6hrs, and includes 26.6M mutual inductances, while the simulation time for 330μm×900μm partition is 12hrs, and uses up about 3Gb memory. On the other hand, the precorrected-FFT method requires less than one hour and only 110MB memory. We also test the same circuit with a higher level of accuracy in the precorrected-FFT method with the fifth nearest cells included in the precorrection step and the overshoot is only 2mV different. The trends in the overshoots and run time from the two methods indicate that the precorrected-FFT converges easily, and therefore is a better candidate for fast simulation of large inductive circuits for higher accuracy.

The problem faced here by the block-diagonal method is common to most of the existing algorithms in on-chip inductance extraction. As the circuit size is increased, the local interaction region should be larger to maintain the same accuracy in the simulation. However, it is hard to predict this interaction region *a priori*, and for large circuits, increasing the interaction region gradually is impractical as it could result in very long simulation times. The precorrected-FFT method, on the other hand, overcomes this difficulty by including the calculation of far away inductance interactions using the grid representation.

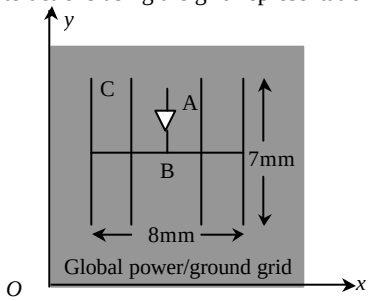

Figure 4: Top view of the layout structure of a global clock net (A: driver input, B: driver output, C: receiver input)

4.3. Application of precorrected-FFT with optimized implementation on signal lines and a large clock net

In addition to a Matlab-based implementation of the precorrected-FFT method, an optimized version using C++ was also implemented. To demonstrate the efficiency of the precorrected-FFT method, layout structures with different length of signal wires, as depicted in Section 4.2, and a large global clock net of an industrial giga-hertz microprocessor are simulated using this optimized implementation of the precorrected-FFT method.

For layout structures with different length of signal wires, the number of resistances, capacitances and inductors in circuits and the total CPU times of the simulations in the precorrected-FFT method are listed in Table 4. A three-dimensional grid is imposed with p=3, and the first nearest neighbors are considered for the precorrection step. The cell size is set to 30μm. The simulations in the precorrected-FFT method can be very fast. For the circuit with 5400μm signal wire, which includes 32.3K resistances, 64.5K capacitances and 32.3K inductors, the total CPU time is about 6 mins.

The layout of the clock net is shown in Figure 4 and has 4 ports, 12 sinks and 121065 inductors, which corresponds to 7.3G

inductance terms. Using optimized code that implements our method, the run time for PRIMA to generate the reduced order model is 21 minutes using a three-dimensional grid, and 2D precorrected-FFT is expected to be even faster. The responses from the simulation under the RC model, the precorrected-FFT and block diagonal methods are shown in Figure 5, and the layout and experimental parameters are listed in Table 5. The partition size in the block diagonal method and the direct interaction region in the precorrected-FFT procedure have approximately the same area. On-chip inductance is seen strongly affect the response. The 50% delay from the precorrected-FFT method is 130ps, compared with an 86ps delay predicted by the RC model. Relative to the 50% delay point for the far end response under an RC-only model, the precorrected-FFT method shows a shift of 17ps, while the shift from the block diagonal method is only 6ps. In addition, the differences between the 10%-90% transition time at the near and far end responses under an RC simulation and under the precorrected-FFT based simulation are 53ps and 70ps respectively, while the corresponding results from the block diagonal method are 20ps and 90ps. Therefore, in this example, compared with the precorrected-FFT results, the block diagonal method underestimates the inductance effect on the transition time at the near end by 62% and overestimates the effect on the transition time at the far end by 28.5%.

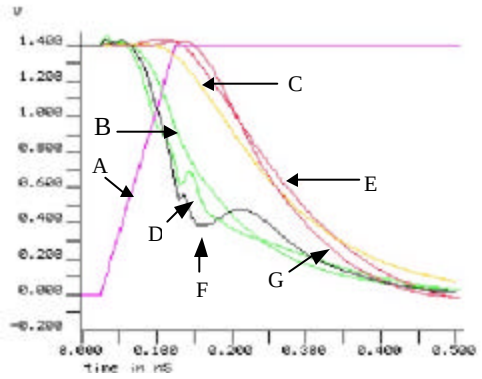

Figure 5: Responses from simulation under an RC-only model, the precorrected-FFT method and the block diagonal method for the near and far ends. A: driver input waveform, B and C: driver output and receiver input waveform, respectively, under an RC-only model, D and E: driver output and receiver input waveform, respectively, calculated using the precorrected-FFT method, F and G: driver output and receiver input waveform, respectively, calculated by the block diagonal method.

5. Conclusion

A precorrected-FFT algorithm for fast and accurate simulation of inductive systems is proposed in this paper, in which long-range components of the magnetic vector potential are approximated by grid currents, while nearby interactions are calculated directly. All inductance interactions are considered in computing the product of the inductance matrix with a given current vector, so that the induced voltages as well as the waveforms at the nodes of interest are calculated accurately. The method is demonstrated on large circuits and is shown to be faster, less memory intensive and more accurate than the block diagonal algorithm. Different approximations in the method, including using a two-dimensional grid structure, are tested and show that lowering the order of the approximation greatly improves the speed and memory consumption without a significant loss in accuracy.

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] Z. He, M. Celik and L. T. Pileggi, "SPIE: Sparse Partial Inductance Extraction," *Proc. of the ACM/IEEE Design Automation Conference*, pp. 137-140, 1997.

[4] B. Krauter and L. T. Pileggi, "Generating Sparse Inductance Matrices with Guaranteed Stability," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 45-52, November 1995.

[5] K. L. Shepard and Z. Tan, "Return-Limited Inductances: A Practical Approach to On-Chip Inductance Extraction," *Proc. of the IEEE Custom Integrated Circuits Conference*, pp. 453-456, 1999.

[6] K. Gala, V. Zolotov, R. Panda, B. Young, J. Wang and D. Blaauw, "On-Chip Inductance Modeling and Analysis," *Proc. of the ACM/IEEE Design Automation Conference*, pp. 63-68, June 2000.

[7] A. Devgan, H. Ji and W. Dai, "How to Efficiently Capture On-Chip Inductance Effects: Introducing a New Circuit Element K," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 150-155, November 2000.

[8] H. Ji, A. Devgan and W. Dai, "KSPICE: Efficient and Stable RKC Simulation for Capturing On-Chip Inductance Effect" Technical Report UCSC-CRL-00-10, University of California Santa Cruz, Santa Cruz, CA, 2000. Available at http://ftp.cse.ucsc.edu/pub/tr/ucsc-csl-00-10.ps.Z.

[9] M. Kamon, M. J. Tsuk and J. White, "FastHenry: A Multipole-Accelerated 3-D Inductance Extraction Program," *Proc. of the ACM/IEEE Design Automation Conference*, pp. 678-683, June 1993.

[10] J. R. Philips and J. K. White, "A Precorrected-FFT Method for Capacitance Extraction of Complicated 3-D Structures," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 268-271, 1994.

[11] A. Odabasioglu, M. Celik and L. T. Pileggi, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 58-65, 1997.

[12] L. Pillage and R.Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352-366, April 1990.

[13] C. Hoer and C. Love, "Exact Inductance Equations for Rectangular Conductors with Applications to More Complicated Geomotries," *J. Res. Nat. Bureau of Standards*, pp. 127-137, vol. 69C, No. 2, April-June 1965.

Table 1: A comparison of the accuracy, memory requirements and CPU time for different parameter settings for the precorrected-FFT in the simulation of three 5400μm long signal wires. Here, "2-D" and "3-D" correspond to the two-dimensional and three-dimensional cases, respectively. The total CPU time corresponds to the time required for the entire simulation, including the time required by the precorrected-FFT computations.

		Total CPU time (s)	Setup time (s)		Memory requirements (Mb)	Relative error of over/undershoot
			Inductance values	$W, H, \tilde{M}$ matrices		
p=2	2-D	2917	1060	148	110	13%
	3-D	3094	1060	302	113	12%
p=3	2-D	3118	1060	312	113	1.3%
	3-D	3682	1060	858	156	<1%
p=4	2-D	3175	1060	354	117	<1%
	3-D	4090	1060	1196	172	<1%

Table 2: A tabulation of the accuracy, memory requirements and CPU time for different circuit sizes using the block diagonal (BD) and precorrected-FFT (PCFFT) methods. The total CPU time corresponds to the time for the entire simulation, including the time required by the block diagonal or precorrected-FFT methods.

	Total CPU time (s)		Setup time (s)		Memory requirement (Mb)		Relative differences	
	BD	PCFFT	BD	PCFFT	BD	PCFFT	50% delay	Over/Undershoot
900μm	578	683	334	450	66	43	<0.1%	14%
1800μm	1056	1097	571	630	95	56	1%	0.5%
3600μm	1993	1991	1042	1010	153	89	5%	10%
4500μm	2516	2555	1285	1150	184	97	8%	19%
5400μm	3235	2917	1522	1220	210	110	12.5%	>50%

Table 3: Overshoots and run times at the receiver side of the middle wire with the length of 5400μm from the precorrected-FFT method (PCFFT) and the block diagonal method (BD) with different partition sizes varying from 30μm×30μm to 330μm×900μm.

	PCFFT	BD					
		30μm× 30μm	180μm× 150μm	330μm× 150μm	330μm× 300μm	330μm× 600μm	330μm× 900μm
Overshoot	151mV	120mV	300mV	120mV	123mV	142mV	161mV
Run time	2917s	681s	3235s	5032s	9700s	6hrs.	12hrs.

Table 4: Circuit parameters and run times for layouts with different length of signal wires from the precorrected-FFT method.

Length of signal wires	No. of resistances	No. of capacitances	No. of inductors	Total CPU time (s)
900μm	7.3K	14.7K	7.3K	~ 60
1800μm	12.3K	24.7K	12.3K	137
3600μm	22.3K	44.6K	22.3K	261
4500μm	27.3K	54.6K	27.3K	306
5400μm	32.3K	64.5K	32.3K	358

Table 5: Layout and experimental parameters (X, Y, Z: x, y and z directions in Figure 4)

No. of sinks/ports	No. of R/L/C	No. of M	No. of nodes	Runtime
12/4	160K/121K/400K	7.3G	245K	21mins
Cell size in X/Y/Z	No. of cells in direct interaction region	No.of grid points in X/Y/Z per cell	No. of cells in X/Y	No. of collocation points
74.97/74.50/4.968	9	3/3/2	64/64	144