Full-chip, three-dimensional, shapes-based RLC extraction *

K. L. Shepard, D. Sitaram[1], and Yu Zheng
Columbia Integrated Systems Laboratory (CISL), Department of Electrical Engineering
Columbia University, New York, NY 10027
[1]Cadence Design Systems, New Providence, NJ

Abstract

In this paper, we report the development of the first commercial full-chip, three-dimensional, shapes-based, RLCK extraction tool, developed as part of a university-industry collaboration. The technique of return-limited inductances is used to provide a sparse, frequency-independent inductance and resistance network with self-inductances that represent sensible "nominal" values in the absence of mutual coupling. Mutual inductances are extracted for accurate noise analysis. The tool, Assura RLCX, exploits high-capacity scan-band techniques and disk caching for inductance extraction as an extension to Cadence's existing Assura RCX extractor.

1 Introduction

With technology scaling, chips consist of more interconnect wires of smaller cross sections packed closer together. As a result, RC delays have become an important performance limitation, and capacitive coupling is a significant source of on-chip noise. These trends have resulted in significant effort to model and extract interconnect as coupled RC networks[1] for use in static timing and static noise analysis.

Most recently, however, inductance and inductive coupling have become important in the timing and noise analysis of a growing number of on-chip signal lines. Inductance must be included to accurately predict rise and fall times and delays in timing analysis. If an inductive net is overdriven, an underdamped ringing response can be observed, which can result in functional failure in receiving circuits or produce reliability problems through gate oxide stress. Moreover, inductive coupling, along with capacitive coupling, can be a significant source of noise on quiet nets due to the switching of nearby perpetrators.

A growing body of literature exists which attempts to precisely quantify when inductance effects are important[2, 3, 4, 5]. These simple relations apply to quasi-TEM propagation in a lossy transmission line. While they differ slightly in formulation, the general result is best expressed as one of two equivalent expressions. The first of these is:

$$R_{driver}Cl < \frac{\mathcal{R}\mathcal{C}l^2}{2} < \tau_f \qquad (1)$$

where $\mathcal{R}$ and $\mathcal{C}$ are the resistance and capacitance per unit length of the wire of length l. τ_f is the time-of-flight for the wire and R_{driver} is the effective resistance of the driver. In words, this equation states that the gate delay of the driver is less than the RC delay of the wire, and both are less than the time-of-flight. Recognizing that the time-of-flight is given by $\tau_f = \sqrt{\mathcal{L}\mathcal{C}}l$ and that the slew time at the driver (t_{slew}) is approximately twice the driver delay yields the equivalent relation:

$$\frac{t_{slew}}{2\sqrt{\mathcal{L}\mathcal{C}}} < l < \frac{2}{\mathcal{R}}\sqrt{\frac{\mathcal{L}}{\mathcal{C}}} \qquad (2)$$

Both sets of equations contain two important physical statements, which are probably most easily evident in Equation 2. The resistive loss of the line must be "small" and the slew time must be "fast." Specifically, from the inequality between the second and third expressions in Equation 2, the total loss ($\mathcal{R}l$) must be less than twice the characteristic impedance of the transmission line ($Z_0 = \sqrt{\mathcal{L}/\mathcal{C}}$). The slew time ($t_{slew}$) must be faster than twice the time-of-flight ($\sqrt{\mathcal{L}\mathcal{C}}l$). It is important to note that these two statements of "small" and "fast" are a function of the entire transmission line, including the driver. Furthermore, expressions involving the characteristic impedance and time-of-flight (characterized as $\sqrt{\mathcal{L}\mathcal{C}}l$) are not applicable to the very nonuniform transmission line properties that characterize on-chip interconnect, which must be practically represented as lumped-element models. Nonetheless, it does explain qualitatively why inductance is becoming important for an increasing number of nets on chip, given recent trends in technology and design.

- Designers routinely apply wide, thick upper-level metal for long signal runs to reduce inductance. Long parallel bus structures are also common, which can accentuate inductive effects particularly in the presence of simultaneous switching.
- Increasing clock frequencies and faster devices translate into faster on-chip slew times.
- Low resistivity metals (copper) are becoming increasingly common to control resistance, and low-k dielectrics are also being introduced to reduce capacitance. While these technology changes both lead to "fast" wires, they increase the relative importance of inductance dramatically.

Because whether the inductance of a given line is important depends, in general, on the entire line and its environment including the driving and receiving circuits, it is difficult to know *a priori* all of the nets that need this consideration. This is much the same problem associated with a "selective-net" approach to detailed resistance and capacitance extraction, and most modern design methodologies have embraced "full-chip" analysis and extraction for interconnect[6, 7].[1].

In this paper, we report the development of the first commercial full-chip, three-dimensional, shapes-based RLCK[2] extraction tool, developed as part of a university-industry collaboration. This tool exploits the return-limited inductance extraction approach described previously [8, 9] within the context of Cadence's Assura RLCX extraction engine. In Section 2, we describe the issues unique to

*This work was supported in part by the National Science Foundation under Grant CCR-97-34216 and by a gift from Cadence Design Systems.

[1]This does not mean that many inductance cannot be "filtered" during extraction. This will be described more in Section 4.

[2]"K" refers to mutual inductance, which corresponds to the SPICE-syntax convention. "M" was already used to refer to MOS transistors.

inductance extraction and review the return-limited inductance extraction approach. In Section 3, we describe the software architecture of Assura RLCX, including the inductance extraction capability. Section 4 describes the "filtering" approaches employed to only extract inductors where electrically significant and discusses implemented enhancements to the Standard Parasitic Format (SPEF)[10] to handle inductance. Section 5 presents some extraction results. Section 6 offers conclusions and directions for future work.

2 Return-limited inductance extraction and the halo rules

From a practical point of view, it is interesting to contrast inductance and capacitance extraction. Capacitance is very local; that is, electric field lines from a given wire tend to terminate on the nearest-neighbor conductors. This makes the capacitance matrix "automatically" sparse (that is, most of the coupling interactions can be easily discarded) but the nonzero coupling capacitances are very sensitive to precise geometries, making the actual calculation of the capacitance values difficult. By contrast, inductance tends to be much less local (although localization assumptions can be carefully applied, as discussed later in this section); that is, current loops defining flux linkages can easily extend far beyond the nearest neighbor conductors. This makes the problem of determining which mutual couplings to discard and how to discard them more difficult but makes the actual calculation of the inductance values comparatively simple, since strong geometry dependence is avoided.

In capacitance extraction, supply and ground wires as well as the substrate are regarded as equipotential ac grounds (an assumption that is verified by power-supply integrity analysis and a guarantee of adequate substrate and nwell plugs). In the "nominal" case, capacitance from the "primary" net to other signal lines ("coupling" nets) is also regarded as capacitance to ground, since those lines are assumed to be quiescent and, therefore, tied to supply or ground through an active driver. To calculate the effects of coupling-net switching activity, full simulation of the capacitance network is required.

Inductance is a property of current *loops*. As a result, the inductance of a signal line depends on the (potentially) frequency-dependent current distribution which characterizes the return-path. At frequencies high enough that inductance matters, current returns favor the lowest-inductance paths, usually the closest power or ground line, but sometimes a neighboring signal line or collection of signal lines. While power and ground lines are always-available paths for high-frequency current returns in digital CMOS integrated circuits (assuming good power-ground design and adequate decoupling), when signal lines constitute part of the return path, they usually do so conditionally in a way that depends, in part, on the switched state of driving and receiving circuits. [3]

To allow equivalent circuits to be developed in complex integrated circuit environments in which return paths are not known, the concept of partial inductances is traditionally used. The partial inductance technique, which assigns portions of the loop inductance to segments along the loop, appeared in work by Rosa in the early 1900's[11]. The book by Grover[12] provides a comprehensive tabulation of formulae for partial inductances and partial mutual inductances for different geometries. Applying partial inductances to the modelling of complex multiconductor geometries was formalized by Ruehli with the development of partial-element equivalent circuits (PEEC)[13, 14, 15]. In the partial inductance approach, the signal lines and supply and ground lines are treated equivalently, resulting in a large, densely-coupled network representation. In addition, the self-inductances do not correspond to a "physical" current return, but rather one at infinity.

[3] Displacement currents capacitively coupled into the (resistive) substrate find their way onto power and ground lines based on the proximity of substrate and nwell plugs.

Practical inductance extraction (through a set of meaningful approximations based only on geometry) must provide a sparse frequency-independent inductance and resistance network with self-inductances that represent sensible "nominal" values in the absence of mutual coupling. Return-limited inductance extraction achieves both of these goals. We refer the reader to References [8, 9] for details on the technique, but we review the aspects of the approach most important to its implementation here. In return-limited inductance extraction, the signal lines are extracted independently of the power-ground wires by modelling the power-ground wires as ideal current returns. Returns through the "closest" power-ground lines define the self-inductances.[4] Mutual inductances between signal lines are further restricted by a set of simple geometry-based decomposition rules, which we refer to as *halo rules*. These rules exploit the fact (to a good approximation) that high-frequency (that is, at frequencies at which the inductance matters) current returns will not have to extend far beyond the nearest power or ground lines.

The halo rules rely on a few definitions. A wire *segment* (or *rectangle*) is a rectangular parallelopiped defined by coordinates $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$. A *horizontal segment* is one in which the current flow is known to be horizontal (i. e., in the x direction), while a *vertical segment* is one in which the current is known to be vertical (i. e., in the y direction). The halo of a segment consists of the six semi-infinite subregions shown in Figure 1. The *horizontal halo* consists only of regions R_3, R_4, R_5, and R_6, while the *vertical halo* consists only of regions R_1, R_2, R_3, and R_4.

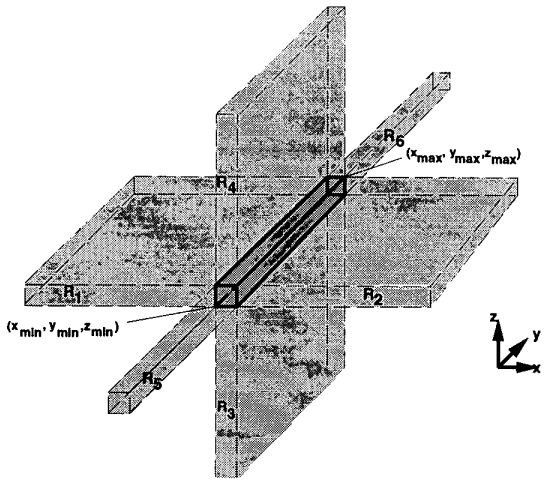

Figure 1: The *halo* of a given segment consists of six semi-infinite regions.

The halo rules are given as follows:

- Horizontal and vertical signal line segments are treated independently since they do not inductively couple to each other. Segments with horizontal currents can only couple inductively with other segments with horizontal currents. Similarly, vertical segments can only couple inductively with other vertical segments.[5]

[4] The characterization of this technique in Reference [16] is incorrect. The power-ground distribution is indeed *included* as an available current return. In effect, the power-ground wires of an interaction region represent the ground "plane" reference.

[5] This means that the extraction can be done in two separate "passes:" one to extract the horizontal segments and one to extract the vertical segments.

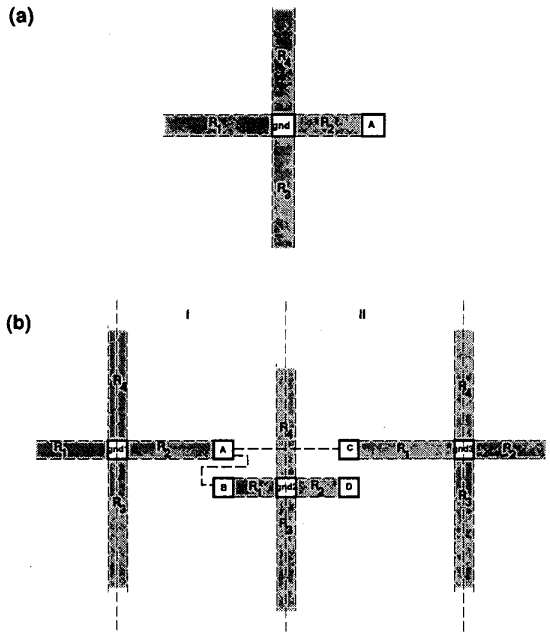

Figure 2: Properties of *halos*

- Horizontal halos of power and ground are "blocked" by horizontal signal segments while vertical halos of power and ground are "blocked" by vertical signal segments. If the halos are viewed as columnated beams emanating orthogonally from each face of a segment, then blocking occurs whenever these beams are interrupted by another segment. Figure 2(a) shows the R_2 portion of the vertical halo of a ground segment blocked by the segment of a neighboring signal wire A.
- Inductive coupling between two horizontal segments is nonzero if and only if it is possible to connect two segments by a path which does not cross the horizontal halo of any ground or supply line. Similarly, inductive coupling between two vertical segments is nonzero if and only if it is possible to connect two segments by a path which does not cross the vertical halo of any supply or ground line.[6] In Figure 2(b), signal segments A and B inductively couple, for example, because it is possible to connect them by a path (shown as a dotted line) which does not cross a halo. A and C, however, do not couple because all paths (such as the one shown as the dotted line in Figure 2(b)) between them must cross either halo region R_4 or R_5 of *gnd2*.

These halo rules divide the chip interconnect into a collection of disjoint *horizontal interaction regions* defined by the non-blocked horizontal halos of the power and ground distribution. Horizontal segments must be contained within the same *horizontal interaction region* to inductively couple. Independently, the chip is also divided into a collection of disjoint vertical interaction regions defined by the non-blocked vertical halos of the power and ground distribution. Similarly, vertical segments must be contained within the same *vertical interaction region* to inductively couple. For example, in Figure 2(b), the halo rules result in the definition of two vertical interaction regions, labelled I and II. Region I contains signal segments A and B, while region II contains signal segments C and D. In terms of the entire inductance matrix, the return-limited inductance approach results in a *block-diagonal sparsification* of the inductance matrix.

One attractive feature of this approach is that the halo rules can be easily "refined" with selective merging of interaction regions, trading extraction runtime for accuracy. This can be done while preserving all the other attractive features of this approach, in particular the passivity of the resulting network and the "physicalness" of the self-inductance values.

[6] Since when doing vertical signal line extraction, for example, we only need to consider vertical halos of ground or supply lines and these halos can only be blocked by vertical signal segments, horizontal signal segments do not have to be considered at all.

3 Extraction within the Assura RLCX shapes-processing environment

Assura RLCX is derived from the GOALIE2 shapes-processing engine described elsewhere[17, 18]. In this section, we give a brief overview of the software architecture as it has evolved in the Assura RLCX system and discuss the entensions to support inductance extraction.

The Assura RLCX system consists of a number of separate program modules which operate on geometric information (rectangles) stored in *edge files*. Other binary files are used to store elements such as transistors (*device files*), capacitors (*capacitance files*), and resistors (*resistance files*). A simplified data flow diagram of the system is shown in Figure 3. The blocks to the left show the basic system. An input GDSII file is processed through the LVS (layout-versus-schematic) engine to generate a set of edge files and device files, which contain the extracted layout of the input GDSII. The resistance extraction program (`rex`) modifies the edge files with cuts for capacitance distribution and generates the resistance files. Current directions identified by `rex` in these edge files are used to identify horizontal and vertical segments in inductance extraction. `rex` also associates *terminals* with certain rectangles or collections of rectangles in order to define the connectivity of the resulting netlist. These terminals could exist anywhere within a rectangle, but there will be at most one per rectangle. The modified edge files are then passed to the capacitance extraction module, which generates a capacitance file. The device files, resistance files and the capacitance files are then input to the netlist generators which generate either a SPICE netlist for the design or a SPEF output.

The blocks in Figure 3 enclosed in dotted lines are the modules we have added in this work to support inductance extraction. The edge files after resistance extraction are passed as input to the interaction region generation program (`lextract`) which is responsible for applying the halo rules, dividing the chip into disjoint horizontal and vertical interaction regions. The result is two *interaction region files*. The horizontal (vertical) interaction region file contains a geometric specification of the interaction region and all the horizontal (vertical) signal and power-ground lines contained in the interaction region. The interaction region geometry is specified on each metal layer with a set of rectangular *subregions*. To ensure that this specification is unique, the subregions are defined in a *vertically dominant* way. By this, we mean that every horizontal side of a subregion will constitute part of the interaction region boundary. The `lextract` module itself only produces the vertical file with a left-to-right scan algorithm. To produce the horizontal file, `lextract` is run as a second pass with the input rectangles y-sorted and transposed to effectively perform a bottom-to-top scan. `lextract` does not subdivide a signal lines or power-ground lines; this is left to a later module.

`lextract` uses a scan band to process the layout data with a scan-band width determined by twice the maximum distance between any signal line and the nearest power-ground line. This scan-band width is user-specified and if a signal line exits the scan-band with an open interaction region, the interaction region is automat-

ically closed with a fictitious power-ground return (and the user is notified with a warning message).

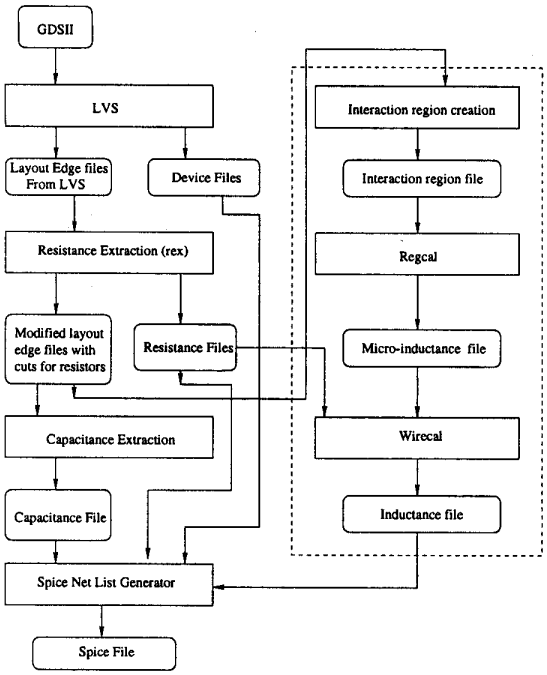

Figure 3: Assura RLCX architecture.

The inductance calculation as shown in Figure 3 is actually implemented with two engines. The first (`regcal`) reads the interaction region files. Acting one interaction region at a time, `regcal` divides nets into smaller rectangles based on a set of fracturing rules and then calculates the return-limited inductances of the constituent rectangles. The second engine (`wirecal`) "reduces" the inductance matrix to match the terminals in the resistance file. In this way, each resistance is associated with at most one self-inductance. The "breaks" created by `rex` must be controlled with a requirement on the maximum number of squares between breaks to ensure an adequate lumped-element approximation. There are various other secondary modules in the flow to sort and merge intermediate files.

We first consider some of the implementation details of `regcal`. We reference all of our discussion to vertical interaction region files. Horizontal interaction region files, which are processed independently by `regcal`, follow the vertical processing with x and y transposed. Fracturing of signal rectangle occurs in three steps. First, if a rectangle overlaps the horizontal boundary of a subregion, then it is divided into two rectangles by the boundary. Rectangles outside the current interaction region are discarded. Second, if a signal rectangle overlaps one of the ends of a power rectangle in the y-direction (in other words, if the y-value defining the end of the power rectangle lies between the y values of the two ends of the signal rectangle), then the signal rectangle will be divided into two rectangles by the y-value of the end of the power rectangle. Third, every terminal divides the associated signal rectangle into two rectangles. Figure 4 is an example illustrating the fracturing process. One signal rectangle in the original interaction region is divided into six segments, two of them discarded. Breaks a and e are created by the boundary of subregion 2. b and c are created by the ends of power rectangles. d is formed by the terminal of the net.

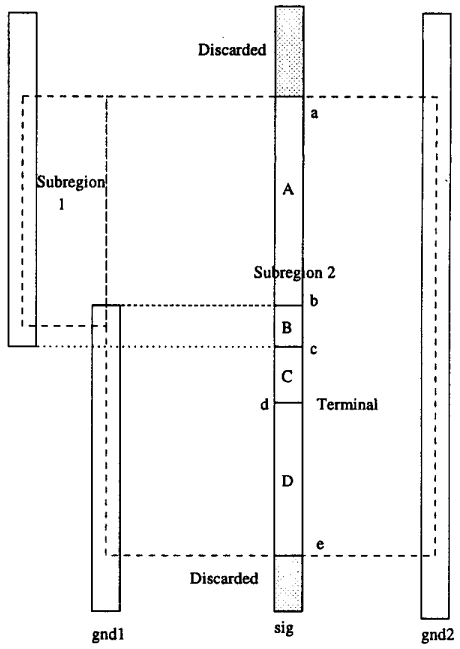

Figure 4: Example illustrating the fracturing rules.

After these fracturing operations, (open) loops are defined for each signal rectangle returning through the portion of every parallel power-ground rectangle in the interaction region overlapping it in the y-direction. In this manner, an inductance network is defined in which the inductance of each signal segment is modelled as a set of parallel loop inductances. These loop inductances are all coupled to each other, defining a *return-limited loop inductance* matrix, $\mathbf{L}'$[8, 9].

The return-limited loop inductance element between two loops i and j is given by:

$$L'_{ij} = \mathcal{L}_{ij} + \mathcal{L}_{\tilde{i}\tilde{j}} - \mathcal{L}_{i\tilde{j}} - \mathcal{L}_{j\tilde{i}} \tag{3}$$

where $\mathcal{L}_{ij}$ is the mutual partial inductance between the two signal rectangles in the two loops, $\mathcal{L}_{\tilde{i}\tilde{j}}$ is the mutual partial inductance between the two associated power segments, and $\mathcal{L}_{i\tilde{j}}$ and $\mathcal{L}_{j\tilde{i}}$ are the mutual partial inductance between the signal and power-ground segments. The partial inductance $\mathcal{L}$ can be calculated from the formulae of Grover[12], as discussed in References [8, 9].

In this way, $\mathbf{L}'$ is actually derived from a congruence transformation on the partial inductance matrix $\mathcal{L}$:

$$\mathbf{L}' = \mathbf{X}^T \mathcal{L} \mathbf{X} \tag{4}$$

where $\mathbf{X} \in \Re^{n \times (n-1)}$ is the matrix and the transformation and is of rank $n-1$. This ensures that $\mathbf{L}'$ (as $\mathcal{L}$) is positive definite. In cases in which two or more signal segments share *exactly* the same power-ground returns, the return-limited loop inductance representation should contain only one loop (chosen arbitrarily) for the signals beyond the first.

Geometric mean distances (GMD) are used in the calculation of $\mathcal{L}$ (and, therefore, $\mathbf{L}'$), effectively assuming uniform current density across the wire cross sections. At high frequencies, the skin and proximity effects can produce small changes in the inductance values (which we ingore) and large increases in resistance. The latter effect is not a concern because the skin depth almost always exceeds

the wire thickness and width at typical value of the cross-over frequency, f_c, defined by $R = 2\pi f_c L$. This means that the frequency dependence of the resistance occurs at frequencies above f_c, where the inductive response dominates. GMD is calculated through table look-up and summation rules for small separations.

We can further collapse $\mathbf{L}'$ into a *return-limited inductance matrix* $\mathbf{L}$ in which a single coupling inductance represents each segment with the transformation:

$$\mathbf{L} = (\mathbf{B}^T \mathbf{L}'^{-1} \mathbf{B})^{-1}$$

where $\mathbf{B} \in \Re^{(n-1)\times m}$ $(m \leq (n-1))$. $\mathbf{L} \in \Re^{m\times m}$. The ith column of $\mathbf{B}$ is all zero except for ones in the rows corresponding to the return-limited loop inductances associated with the given return-limited inductance[8, 9].

After the calculating of $\mathbf{L}$ for a given interaction region, it is written out to the intermediate *micro-inductance file* as shown in Figure 3. In addition to the $\mathbf{L}$-matrix elements, we store the midpoint of each associated signal rectangle and its net label, information required by the wirecal module.

The wirecal module acts, in turn, on the micro-inductance file to produce a final inductance file for netlist generation. The final inductance file associates at most one self-inductance with each resistance created by rex. As a result of the action of rex, a single terminal may be associated with several connected rectangles. The result may be an "L" shape (as in Figure 5) but will not be a "T", "+", or "U" shape. Each net has one and only one terminal, and every resistance is between the terminals of two nets. The inductance associated with each resistance is a series connection of the "micro-inductances" associated with all the segments series-connecting the two terminals. Two bounding boxes are used to determine which micro-inductances must be combined as shown in the example of Figure 5. Each bounding box is defined by the location of the terminal of the associated net and the *joint* location, the point at which the shapes associated with each of the two nets adjoin. Every rectangle labeled with the same net number as the terminal whose midpoint is contained within the associated bounding box is combined. In Figure 5, four such rectangles are associated with the connection between net1 and net2. The gray rectangles (outside the bounding boxes) are not included. The self-inductance is the sum of the self-inductances of the constituent segments and the mutuals between them. Mutual inductances between segments are also formed by the sum of the constituent mutual inductances.

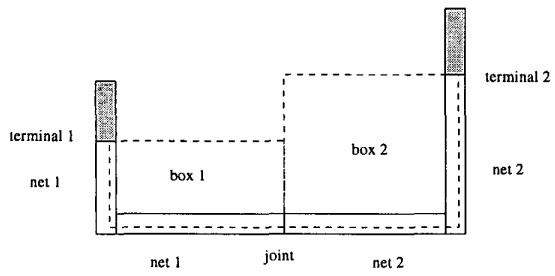

Figure 5: Example showing how "micro-inductances" are combined.

wirecal scans every resistance in the resistance file (as generated by rex) to find the associated inductances and mutual inductances. The result is an inductance file, which is combined with the resistance, capacitance, and device files for netlist generation.

4 SPEF output and inductance filtering

In addition to a SPICE netlist output, we have implemented extensions to SPEF to handle inductance. SPEF has two forms: a reduced and a detailed form. The detailed form is intended for coupling analysis. To this, we have added a mutual inductance element. In the reduced form (which does not handle coupling and is intended for "baseline" timing analysis), the driver is modelled as a pi-model (two capacitors and a resistor) as shown in Figure 6(a). The pi-model accounts for the "resistive shielding" of part of the load (C_2) with a time constant RC_2. Each receiver is modelled as a transfer function $H(s)$ between the driver and receiver characterized by one or more pole-residue pairs. To handle inductance, we allow a driver model to also be of the form shown in Figure 6(b) with two capacitors, a resistor, and an inductor. This is simpler than alternate driver models[19] and is similar to the π-match used for impedance transformation in RF circuits. Let y_i represent the moments of the driving-point admittance. If $y_3^2 \leq y_2 y_4$, then the model of Figure 6(a) is used with: $C_1 = y_2^2/y_3$, $C_2 = y_1 - C_1$, $R = -y_3^2/y_2^3$. If $y_3^2 > y_2 y_4$, then the model of Figure 6(b) applies with:

$$\begin{aligned} C_2 &= \left(y_2 y_3 + y_2\sqrt{y_3^2 - y_2 y_4}\right)/y_4 \\ C_1 &= y_1 - C_2 \\ R &= -y_2/C_2^2 \\ L &= \left(-y_3 C_2 + y_2^2\right)/C_2^3 \end{aligned}$$

In creating the reduced SPEF representation, all of the interconnect coupling are "broken"; that is, coupling capacitors are tied to ground (adding to the grounded capacitance of the net) and mutual inductances are discarded. We refer to this as the *decoupled* net.

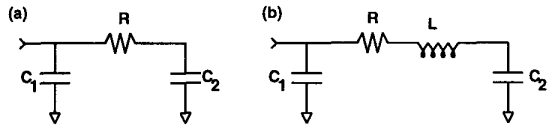

Figure 6: Two possible driver models for reduced SPEF format.

An essential aspect of inductance extraction is that it is only applied to nets where the inductance is electrically significant. To this end, we employ three levels of filtering during the extraction process. The first filter is implemented in regcal and removes "small" rectangles below a user-specified length. While this cannot be justified on electrical grounds, it is practically necessary to prevent regcal from being overwhelmed. With a conservative filter of $1\mu m$, this has no impact on the extraction accuracy. The second filter operates in wirecal, discarding the inductances associated with segments for which the cross-over frequency $f_c \gg f_{max}$. f_{max} is a user-specified "maximum" frequency, which characterizes the fastest on-chip slews possible in the given technology. We have conservatively set f_{max} to a default value of 100 GHz in our extractions.

The third filter is implemented in netlist generation (both in the SPICE netlist generator and in the SPEF generator). In SPEF generation, we calculate the equivalent driving-point model (Figure 6) for each driver of a decoupled net (usually there is only one driver, except for tristate busses). The receiver load capacitances (as determined by the standard-cell pin caps) must be included. If for all the driving-point models rendered with the C1-R-L-C2 pi-model representation, $f_{max} \ll 1/2\pi\sqrt{LC_2}$, then the inductance of the net can be ignored (that is, the inductive shielding of the load is not effective).[7] All of the self-inductances of the net and all the mutual-inductances associated with these self-inductances are discarded. In

[7] One has to be careful using the "decoupled" L in this analysis (that is, discarding

SPICE netlist generation, the analysis is similar except that transistors (rather than cells) are now connected to each net. Source-drain connections are treated as drivers and gate connections are treated as receivers (linearized as an appropriate gate capacitance).

5 Results

One of the key features of this approach is the block-diagonal sparsification of the inductance matrix produced by the halo rules. To understand the impact of these sparsification assumptions on the accuracy of the extraction (in both the time-domain and frequency-domain), we consider a small example. It is laid out, extracted, and simulated using the TSMC $0.25\mu m$ 5M1P process design rules and device models. This is a 2.5-V process with transistor saturation currents of about $600\mu A/\mu m$ for the NMOS and $300\mu A/\mu m$ for the PMOS. There are five levels of AlCu interconnect. The first four levels have sheet resistivities of 0.076 Ohms/square. metal 5 has a sheet resistivity of 0.044 Ohms/square.

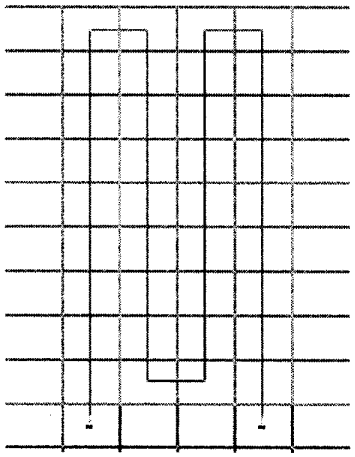

Figure 7: Layout plot of single signal wire snaking through power-ground distribution.

The testcase, shown in Figure 7, consists of a 4 μm wire snaking through a power-ground grid driven by a 100 μm/50 μm inverter and loaded on the far end by a 12 μm/6μm inverter. The signal line is routed in metal 5 vertically and metal 4 horizontally. The power ground distribution is routed in metal 5 and metal 3 horizontally and metal 4 and metal 2 vertically. The power grid spacing is about $130\mu m$ in the horizontal direction and $100\mu m$ in the vertical direction. Power and ground are routed on alternate metal layers so that the vertical metal-metal capacitance of the grid acts as decoupling.

To compare the effects of sparsification on the accuracy of the resulting network, we perform two extractions of this testcase. The first extraction is a straight return-limited extraction in which the power grid is treated as ideal and lossless and a full-interaction region decomposition is performed. In the second extraction, a full PEEC model is created for the network (also using Assura RLCX),

the mutuals), since one could falsely filter nets with inductive responses only brought to the fore in the presence of simultaneous switching (as in the second example of Section 5). One might instead choose to sum in the mutuals to produce a "worst-case" L. There are similar questions about the correct assumptions for the coupling C. We are still invertigating the correct way to implement this filter to cover all cases.

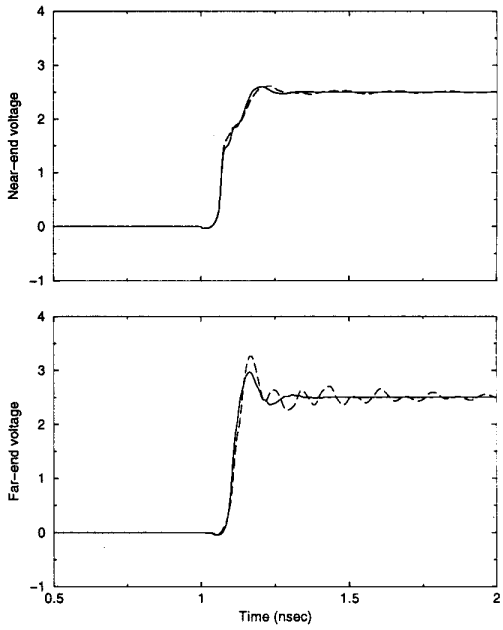

Figure 8: (a) Near-end and (b) far-end time-domain response for the switching of the signal net of Figure 7. The solid lines follow from simulation of the Assura RLCX extraction results. The dashed lines follow from simulation of the full PEEC model.

including full extraction of the power-ground network and no sparsification of the inductance matrix. Both models make the same assumptions about ignoring current returns in the substrate for inductance extraction. Neither model considers skin or proximity effect (that is, the current distribution is uniform across the wire cross-section). The results of both extractions are simulated in HSPICE (we note that the PEEC extractions and simulations are orders of magnitude slower than the return-limited extractions and simulations and are only practical on these small testcases). In Figure 8, we show the time-domain response to a switching driver at the near-end and far-end of the wire. The near-end response if nearly an identical match. The far-end responses are close but the PEEC model does show additional ringing (resonances) not present in the return-limited extraction. We expect the agreement to be better with additional decoupling capacitance in the power grid. Figure 9 compares the magnitude of the driving point impedance in the frequency domain between the return-limited and PEEC extractions. The PEEC model shows some additional structure but the agreement is generally good.

To demonstrate the problematic nature of simultaneous switching in the presence of inductive coupling, we have modified the testcase of Figure 7 to duplicate the single wire fifteen times, resulting in a 16-bit bus snaking through the power grid. The design is extracted in Assura RLCX and simulated in HSPICE. (PEEC comparison is not possible in this case because the HSPICE runtimes are prohibitively long.) The spacing between the wires of the bus is $1.3\mu m$. The drivers here are sized $24\mu m/12\mu m$. In Figure 10(a), we show the near-end response of a switching bit 7 with and without the simultaneous switching of the other 15 bits. Figure 10(b) shows the far-end response. Physically, there is little difference a wide line switching and a wide bus of narrow lines switching simultaneously. While the single line switching is effectively an RC response, all sixteen bits switching together give an inductive re-

sponse with a distinct "porch step" on the near end and ringing at the far end.

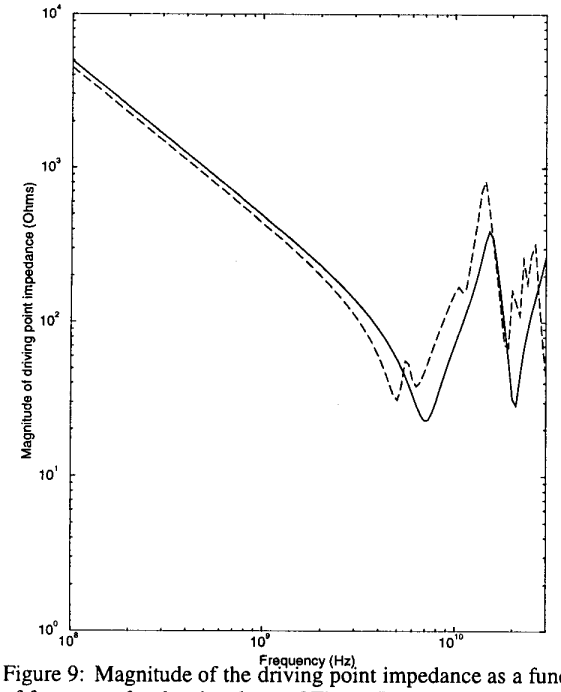

Figure 9: Magnitude of the driving point impedance as a function of frequency for the signal net of Figure 7. The solid lines follow from simulation of the Assura RLCX extraction results. The dashed lines follow from simulation of the full PEEC model.

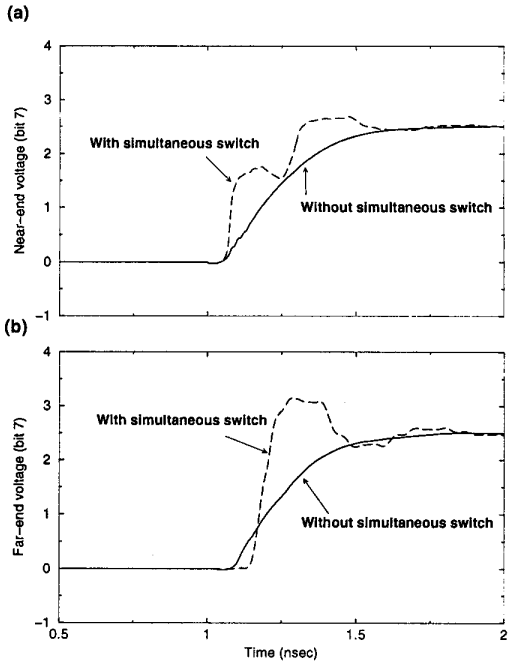

Figure 10: (a) Near-end and (b) far-end response of a switching bit 7 of a 16-bit parallel bus with and without the simultanous switching of the other 15 bits.

To indicate the capacity and runtime performance of our extractor on large data sets, we present statistics for a large testcase in Table 1. The inductance extraction adds about 25 percent to the overall extraction runtime. The careful use of disk caching and scanband techniques keeps the peak core image below 200 MB. With the extensive "blocking" mechanisms in the halo rules, there was a concern that on "real" designs, the interaction regions could grow quite large, significantly slowing down the `regcal` module (which contains a matrix factorization whose size is based on the number of segments in the interaction region). However, this has not proved to be the case. In this example, the largest interaction region is about 550 μm by 300μm. The largest number of signal rectangles in an interaction region (based on `rex` fracturing) is 1452. Because we are still implementing the third-level of filtering for flat transistor level extraction, only the first two filters described in Section 4 were applied here. The second filter filtered out 98 percent of the inductors present after the first filter. The third filter is expected (on this design) to remove the inductors on all but a few nets.

Number of nets	249028
Number of transistors	689533
Technology	0.18μm, 5-layer AlCu
RC runtime	12.5 hours
Average interaction region x extent	19.53 μm
Average interaction region y extent	40.9 μm
Maximum interaction region x extent	547.0 μm
Maximum interaction region y extent	313.5 μm
Average number of power rectangles per interaction region	7
Average number of signal rectangles per interaction region	27
Maximum number of power rectangles per interaction region	494
Maximum number of signal rectangles per interaction region	1432
`lextract` runtime/memory (Sun Ultra-5)	22m, 44.8MB
`regcal` runtime/memory (Sun Ultra-5)	2h41m, 200MB
`wirecal` runtime (Sun Ultra-5)	6.3s
Number of resistors	2450602
Number of capacitors	2125708
Number of inductors	8935
Number of mutual inductors	69039

Table 1: Statistics for "large" testcase

6 Conclusions and future work

In this paper, we have presented the first shapes-based, full-chip, three-dimensional RLCK extraction engine. Full-chip RLCK extraction will become increasingly critical with decreasing on-chip resistances and capacitances and increasing on-chip frequencies.

We intend to "solidify" this extraction engine through alpha- and beta-test on large designs. We expect user-controllable refinements to the halo rules to better trade-off accuracy and network complexity and we expect to continue refinement of the filtering criteria. Further study is also necessary to gauge the real effects of substrate

conduction. The substrate is currently excluded as an available current return path for inductance calculation under the assumption that its high loss will strongly discourage conduction relative to surface metals. Displacement currents injected into the substrate are assumed to immediately find their way onto the circuit ground or supply nodes.

References

[1] K. L. Shepard, V. Narayanan, P. C. Elmendorf, and Gutuan Zheng. Global Harmony: Coupled Noise Analysis for Full-Chip RC Interconnect Networks. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, 1997.

[2] A. Deutsch, G. V. Kopcsay, P. J. Restle, H. H. Smith, G. Katopis, W. D. Becker, P. W. Coteus, C. W. Surovic, B. J. Rubin, Jr. R. P. Dunne, T. Gallo, K. A. Jenkins, L. M. Terman, R. H. Dennard, G. A. Sai-Halasz, B. L. Krauter, and D. R. Knebel. When are transmission-line effects important for on-chip interconnections. *IEEE Transactions on Microwave Theory and Techniques*, 45(10):1836–1846, October 1997.

[3] Y. I. Ismail, E. G. Friedman, and J. L. Neves. Figures of merit to characterize the importance of on-chip inductance. In 35^{th} *ACM/IEEE Design Automation Conference*, pages 560–565, June 1998.

[4] S. Lin, N. Chang, and S. Nakagawa. Quick on-chip self- and mutual-inductance screen. In *Proceedings of the International Symposium on Quality Electronic Design*, pages 513 – 520, March 2000.

[5] B. Krauter, S. Mehrotra, and V. Chandramouli. Including inductive effects in interconnect timing analysis. In *Proceedings of the CICC*, pages 445 – 452, 1999.

[6] K. L. Shepard, S. Carey, E. Cho, B. Curran, R. Hatch, D. Hoffman, S. McCabe, G. Northrop, and R. Seigler. Design Methodology for the G4 S/390 Microprocessors. *IBM Journal of Research and Development*, 21(4/5):515 – 548, 1997.

[7] R. M. Averill and et al. Chip integration methodology for the IBM S/390 G5 and G6 custom microprocessors. *IBM Journal of Research and Development*, 45(5/6):681 – 706, 1999.

[8] K. L. Shepard and Z. Tian. Return-limited inductances: a practical approach to on-chip inductance extraction. In *Proceedings of the CICC*, pages 453 – 456, 1999.

[9] K. L. Shepard and Zhong Tian. Return-limited inductances: a practical approach to on-chp inductance extraction. *IEEE Trans. CAD*, April 2000.

[10] IEEE 1481 delay and power calculation system standard. Technical report, 1999.

[11] E. B. Rosa. The self and mutual inductance of linear conductors. *Bulletin of the National Bureau of Standards*, 4:301 – 344, 1908.

[12] F. Grover. *Inductance Calculations: Working Formulas and Tables*. Dover, New York, 1962.

[13] A. E. Ruehli. Inductance calculations in a complex integrated circuit environment. *IBM Journal of Research and Development*, 16(5):470 – 481, 1972.

[14] A. E. Ruehli. Equivalent circuit models for three-dimensional multiconductor systems. *IEEE Transactions on Microwave Theory and Techniques*, MTT-22:216 – 221, March 1974.

[15] A. E. Ruehli, N. Kulasza, and J. Pivnichny. Inductance of nonstraight conductos close to a ground return plane. *IEEE Transactions on Microwave Theory and Techniques*, pages 706 – 708, August 1975.

[16] Kaushik Gala, Vladimir Zolotov, Rajendran Panda, Brian Young, Junfeng Wang, and David Blaauw. On-chip inductance modeling and analysis. In *Proceedings of the Design Automation Conference*, pages 63 – 68, 2000.

[17] K.-W. Chiang, S. Nahar, and C.-Y. Lo. Time-efficient VLSI artwork analysis algorithms in GOALIE2. 25^{th} *ACM/IEEE Design Automation Conference*, pages 471 – 475, 1988.

[18] K.-W. Chiang. Resistance extraction and resistance calculation in GOALIE2. 26^{th} *ACM/IEEE Design Automation Conference*, pages 682 – 685, 1989.

[19] Chandromouli Kashyap and Byron Krauter. A realizable driving point model for on-chip interconnect with inductance. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 190 – 195, June 2000.