A PARASITICS EXTRACTION AND NETWORK REDUCTION ALGORITHM FOR VLSI

Teng-Sin Pong and Martin A. Brooke
Georgia Institute of Technology, School of Electrical Engineering, Atlanta, GA 30332
Telephone: (404)894-3304

Abstract

This paper discusses an algorithm for the extraction of circuit parasitics in integrated circuits using classic transmission line models. This gives a better account of the DC and AC characteristics of interconnects than models incorporating exclusively either the R or C components. We also detail a network reduction technique used to simplify the extracted RC network at user-specified accuracies to manageable complexities, especially for large VLSI circuits. The model and circuit reduction algorithms are applied to practical sample circuits. Results of simulations illustrating the reduction in circuit complexity and the degree of modeling accuracy by these methods are also given.

Introduction

With the advancement of VLSI technologies, circuit designers require more accurate prediction of propagation delays due to integrated circuit parasitics, such as gate resistances and capacitances [1]. Propagation delays are an integral part of correct timing calculation in critical paths such as clock and power distribution networks, and in all analog circuits in general. The accurate modeling of parasitics causing such propagation delays is therefore very important.

Various papers have been published on this topic. Certain algorithms [2]–[7] involve calculating interconnect resistances using general Laplace equations, conformal transformations, finite elements and differences, or geometric heuristics. They all present varying degrees of accuracy and speed of calculation. Others [8]–[13] include parasitic capacitance calculation in their algorithm as well. However, all these algorithms consider the R and C of interconnect parasitics separately and independently, which is not strictly correct.

In [14], a model is presented which considers interconnects as RC transmission line. The model gives accurate simulation results for simple lines but provides no solution to the complexity problem encountered when it is applied to large-scale networks because of their higher order.

In the extraction of complex VLSI circuits, after accurately modeling the parasitics, a network reduction algorithm is necessary to simplify them. There should also be a way to control the accuracy of the simplified network at user-specified frequencies. This will ensure that no unnecessary accuracy is used in the model when it is uncalled for.

This paper presents a transmission line model for parasitics extraction, coupled with a network simplification algorithm in order to work efficiently even with big VLSI circuit layouts. Extraction and network reduction are carried out at accuracies specified by the operator.

Modeling of Transmission Line Interconnects

It is necessary to use an accurate model to describe circuit parasitics. From transmission line theory, for a line of length D and at a given frequency ω, exact equivalent T and Pi networks can be found [3]. These networks have the form shown in Figure 1.

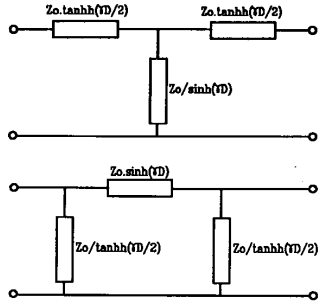


Figure 1: Exact transmission line model

In Figure 1, D =length of the line, $Z_0 = \sqrt{\frac{R+j\omega L}{G+j\omega C}}$ is the characteristic impedance of the line, $\gamma = \sqrt{(R+j\omega L)(G+j\omega C)} = \alpha + j\beta$ is the propagation function, R,L,G and C are the unit resistance, inductance, conductance and capacitance respectively.

We notice that, for $|\ \gamma D\ | \ll 1$:

$$sinh\,\gamma D \simeq \gamma D \quad (1)$$

$$cosh\,\gamma D \simeq 1 \quad (2)$$

$$tanh\,\frac{\gamma D}{2} \simeq \frac{\gamma D}{2} \quad (3)$$

So :

$$Z_0 \times tanh\,\frac{\gamma D}{2} \simeq \frac{(R+j\omega L)\times D}{2} \quad (4)$$

$$\frac{Z_0}{sinh\,\gamma D} \simeq \frac{1}{(G+j\omega C)\times D} \quad (5)$$

We can therefore replace the T and Pi networks in Figure 1 by simplified models, as illustrated in Figure 2.

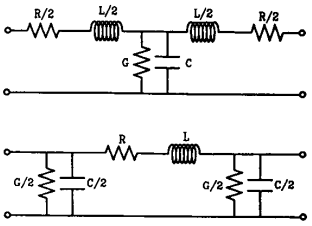


Figure 2: Simplified transmission line model

These simplified models can further be approximated by the use of only R and C elements, i.e. by neglecting the L and G components. Such approximations will incur an error of approximation, *global error*(D), which can be calculated as follows:

In VLSI, $\omega L \ll R$ and $G \ll \omega C$ are generally true. Then:

$$Z_0 = \sqrt{\frac{R}{j\omega C}} = \sqrt{\frac{R}{\omega C}} \times e^{\frac{-j\pi}{4}} \quad (6)$$

$$\gamma = \sqrt{j\omega RC} = \sqrt{\omega RC} \times e^{\frac{j\pi}{4}} \quad (7)$$

$$= \sqrt{\frac{\omega RC}{2}} \times (1+j) = \alpha + j\beta \quad (8)$$

So :

$$\alpha = \beta = \sqrt{\frac{\omega RC}{2}} \quad (9)$$

$$sinh\,\gamma D = sinh\,\beta D\,cos\,\beta D + j\,cosh\,\beta D\,sin\,\beta D \quad (10)$$

The error when approximating $sinh\,\gamma D \simeq \gamma D = \alpha D + j\beta D$ is :

$$error(R) = 1 - \frac{\beta D}{sinh\,\beta D \cos\,\beta D} \quad (11)$$

$$error(Im) = 1 - \frac{\beta D}{cosh\,\beta D \sin\,\beta D} \quad (12)$$

where $error(R)$ and $error(Im)$ are errors in the real and imaginary parts of the approximation respectively. We then define :

$$global\ error(D) = max\,(error(R), error(Im)) \quad (13)$$

Since $tanh\,\gamma D = \frac{sinh\,\gamma D}{cosh\,\gamma D}$, and $cosh\,x \simeq 1$ is a much better approximation than $sinh\,x \simeq x$ (gradient of cosh is much smaller than sinh at the origin), the error of $tanh\,\gamma D \simeq \gamma D$ is lower than that of $sinh\,\gamma D \simeq \gamma D$. The error of approximating $tanh\,\frac{\gamma D}{2} \simeq \frac{\gamma D}{2}$ is therefore limited by the expression of global error(D) derived for $sinh\,\gamma D \simeq \gamma D$ above.

We can now state, for a given error tolerance, what is the maximum permissible length, $max.(D)$, of line elements that can be lumped together : Given e = maximum error allowed, $max.(D)$ allowed is found by substituting e in the expressions above and finding (by iterating, for example) the maximum possible value of D that generates an error smaller than e.

To give an example, if $e = 0.01$ and $\omega = 2\pi \times 100\,MHz$, $R_0 = 0.01\,ohm/square$, $C_0 = 0.1\,fF/\mu m^2$, then :

$$\beta = \sqrt{\frac{\omega RC}{2}} \quad (14)$$

$$= \sqrt{\frac{\omega\,(R_0\,\frac{L}{W}\,\frac{1}{L})(C_0\,W\,L\,\frac{1}{L})}{2}} \quad (15)$$

$$= \sqrt{\frac{\omega R_0 C_0}{2}} \quad (16)$$

$$= 5.60 \times 10^4\,rad/\mu m \quad (17)$$

For $D = 50\,\mu m$, $global\ error(D) = 0.013\,\%$
For $D = 500\,\mu m$, $global\ error(D) = 1.30\,\%$

With modern VLSI circuits, certain feature sizes can easily exceed 3000 μm, which is half the size of a 250 mil chip. The values of D calculated above then indicate that any feature whose size is bigger than the maximum allowable D value should be cut into smaller pieces in order to allow accurate modeling at the accuracy chosen by the user. A way to do this is to calculate $max.(D)$ values for each different material and technology and for each stated error tolerance level. Subsequent modeling using line segments shorter than $max.(D)$ will then ensure that errors incurred by using simplified T or Pi equivalent networks shall be within manageable proportions.

In our extraction algorithm, we will therefore limit the maximum length of elementary transmission lines to $max.(D)$, calculated for each of the layers used in the technology for power lines and interconnects. Long lines will thus be segmented into shorter lengths in order to satisfy the error tolerance specified for the modeling.

As is shown in [20], both the Pi and T type transmission line models are accurate. There is no particular advantage of one of them over the other. However in our computer extraction program, the T model is used because it makes subsequent network reduction easier.

From the T model we derive starred configurations for multiple-pronged line elements. We need these variations because circuit connectivity in Manhattan designs can generate line segments connected to up to 4 other neighbors. For an n-pronged line element, each of the n horizontal resistive branches will carry one-n^{th} of R and the capacitance C will stay at the center of the equivalent star network. A few examples appear in Figure 3.

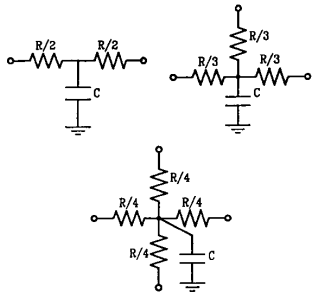


Figure 3: Multi-pronged line elements

RC Network Reduction

The transmission line model provides us with an accurate way to calculate circuit parasitics. But the network thus extracted from actual VLSI circuits would contain too many branches to be practical for any simulation purposes. So there is a need to be able to simplify the extracted network, preferably at user-controlled accuracy, as well.

Using classic network theories, we know that for any star configuration, there exists an equivalent delta configuration with one less node (the reverse is, in general, not true).

In Figure 4, we have :

$$Z_{ij} = z_i \times z_j \times \sum_{k=1}^{n} \frac{1}{z_k} = Real + j\,Imag \quad (18)$$

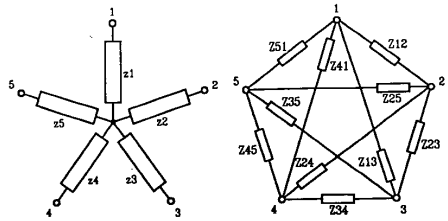


Figure 4: Star to delta conversion

This simple formula gives an *exact* transformation of star to delta networks. It is easily seen that whereas each z_i may be purely resistive or capacitive, Z_{ij} may not be so.

For our network reduction algorithm, we will recursively employ this star-delta conversion to simplify the extracted circuit. If we want an exact circuit description, this method provides whatever precision desired, limited only by the accuracy of floating-point calculations. The star-delta conversion method used recursively allows simplification of complex networks. However, the branches in the final simplified network are neither purely resistive or capacitive.

At each star-delta conversion, we will try to simplify the resulting delta network to one consisting of only purely resistive and purely capacitive branches. This in turn will speed up simulation runs. The choice of reduction to either an R or C branch

will depend on the relative amplitude of the real and imaginary parts of the impedance (complex in general) of the converted delta branch.

An example dealing with a symmetrical 4-pronged line with only R and C branches is given in Figure 5.

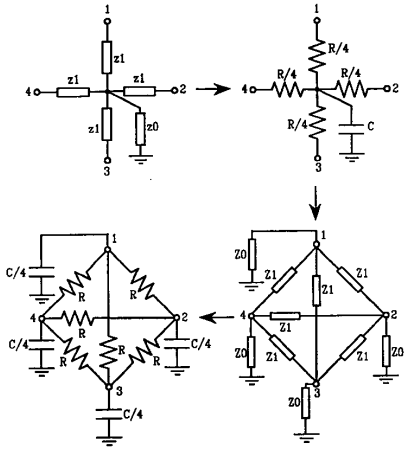


Figure 5: Simplified multi-pronged line

The simplification illustrated in Figure5 would result in an error given by :

$$error(R_{branch}) = \frac{\frac{R^2\omega C}{16}}{R} = \frac{\omega RC}{16} \quad (19)$$

$$error(C_{branch}) = \frac{\frac{R}{4}}{\frac{4}{\omega C}} = \frac{\omega RC}{16} \quad (20)$$

where $error(R)$ and $error(C)$ are errors of approximating respective branches to purely R and purely C elements.

This calculation can be generalized to all stars and would give :

$$error(R_{branch}) = (\frac{Imag(Z_{ij})}{Real(Z_{ij})})_{R_{branch}} \quad (21)$$

$$error(C_{branch}) = (\frac{Real(Z_{ij})}{Imag(Z_{ij})})_{C_{branch}} \quad (22)$$

We then define the overall error, called *maximum error*, to be :

$$maximum\ error = max.(error(R_{branch}), error(C_{branch})) \quad (23)$$

When the maximum error exceeds a specified threshold value, the conversion and subsequent approximation to purely R and C branches will em not be carried out. The maximum error should not be simply accumulated (added together) from one conversion to another. Instead a complete list of all *exact* impedance values of converted branches (including therefore real and imaginary parts) will be kept at all times and used to calculate the exact errors between them and approximated R and C values.

Algorithm for Extraction and Network Reduction

Using the transmission line and network simplification theories as explained previously, we arrive at our parasitics extraction algorithm for Manhattan designs:

a. Find all intersections of line elements and cut up the lines at the intersecting coordinates.

b. Given the working frequency and error tolerance for network reduction (these can be user inputs), calculate the maximum line segment dimensions, for each layer material, to be extracted.

c. Using the values from b., find interconnect lines and other parasitic elements whose dimensions are too big and divide them into smaller pieces whose length and width are *both* below the maximum allowable feature size.

d. Extract connectivity information and number nodes accordingly.

e. Generate line elements using the T transmission line model with exact *complex* impedance values as shown in Figure 1 and connect the R and C branches using connectivity information from d.

f. Simplify the network using the star-delta conversion formulae recursively on every node until the error threshold given by the user governing the reducibility of any star configuration prevent any further reduction.

Implementation and Results

The transmission line model and RC network reduction algorithm are implemented in a C language program of approximately 2700 lines in source code. The program assumes the input to be in the MAGIC layout file format and produces a SPICE simulation deck as output.

The program was run on the Magic layout file of several simple layouts whose responses are known. Parameters such as the error threshold and working frequency were varied to compare their influence on the accuracy of the extracted circuit. Following are results from the extraction of two typical layouts. In both cases, the transmission line model was calculated at an error tolerance level of 10 %, with the working frequency at 10 MHz. The network reduction algorithm was carried out up to a 10 % error threshold. The accuracy of the reduced network are compared to the unreduced exact model in terms of input node impedance as seen from outside the circuit at specific nodes.

Example # 1 (Figure6) shows how a complex RC network due to the many butting junctions in the circuit layout is effectively reduced to the minimum allowed by the star-delta conversion method, i.e. $\frac{n\times(n-1)}{2}$ branches for n I/O nodes. The number of RC branches and nodes was thus reduced by a factor of about 4 to 7. This example layout is typical of power or clock lines with numerous distribution points.

Example # 2 (Figure7) illustrates the effect of different characteristic impedance values of a layer type on the complexity and accuracy of the extracted SPICE file (i.e. the R and C were increased 10 folds arbitrarily to simulate the change in layer and characteristic impedance). It is worth noticing that, at 10 % error tolerance for the reduction algorithm, the extracted circuit is accurate to 90 % up to 1 GHz for all cases.

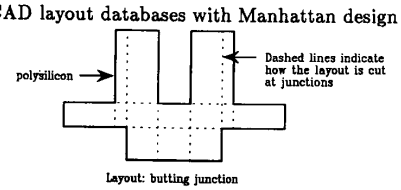
From these examples and others, it is noted that, at relatively high error thresholds for the reduction algorithm (up to even 50 % in simple cases), the extracted circuits still present remarkable accuracy at frequencies up to 1 GHz. This indicates that error thresholds can be relaxed largely in lieu of more simplicity in the extracted circuit and simulation speed.

Conclusion

The transmission line model can be used effectively and accurately in the extraction of parasitics. It affords a better account of propagation delays in integrated circuits. The net-

work reduction technique presented is attractive in that it can accommodate a compromise between speed and accuracy during simulation and produce simulation parameters accordingly.

Although this algorithm is written in the MAGIC context, the general approach in applying transmission line theory and network simplification to parasitics extraction can be extended to other CAD layout databases with Manhattan design rule.



Layout: butting junction

# of RC branches before reduction/ after reduction	# of nodes before reduction/ after reduction	Accuracy falls off at (freq.)
57/15	36/5	200 GHz

Reduction in circuit complexity and accuracy

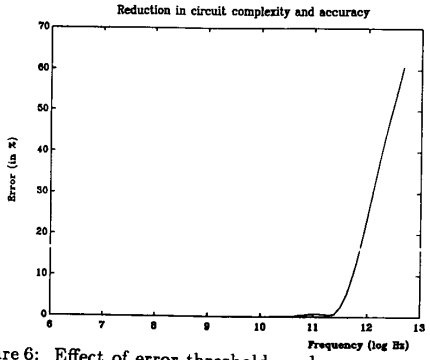
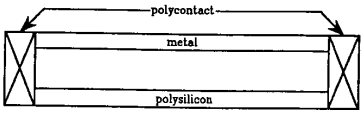


Figure 6: Effect of error threshold used on accuracy of reduced network



Layout: parallel poly-metal lines

Values of R	Values of C	# of RC branches before reduction/ after reduction	# of nodes before reduction/ after reduction	Accuracy falls off at (freq.)
x1	x1	30/2	19/1	100 MHz
x10	x1	36/8	23/4	2 GHz
x10	x10	57/15	37/7	100 MHz

Reduction in circuit complexity and accuracy

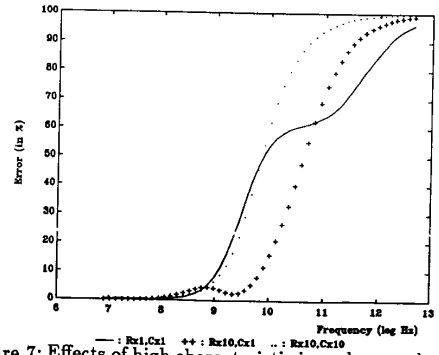


Figure 7: Effects of high characteristic impedance value of a layer type on complexity of extracted circuit

References

[1] David L. Carter, Donald F. Guise, "Effects of Interconnections on Submicron Chip Performance", *VLSI Design*, Jan 1984, pp 63-66

[2] Basant R. Chawla, Hermann K. Gummel, "A Boundary Technique for Calculation of Distributed Resistance", *IEEE Trans. Electron Devices*, vol. ED-17, no. 10, Oct. 1970, pp 915-925

[3] Takashi Mitsuhashi , Kenji Yoshida, "A Resistance Calculation Algorithm and Its Application to Circuit Extraction", *IEEE Transactions on CAD*, vol. CAD-6, no. 3, May 1987, pp 337-345

[4] Mark Horowitz, Robert W. Dutton, "Resistance Extraction from Mask Layout Data", *IEEE Trans. on Computer-Aided Design*, vol. CAD-2, no. 3, July 1983, pp 145-150

[5] Hiroshi Yoshimura, Kazuo Tansho et al., "An algorithm for Resistance Calculation from IC Mask Pattern recognition", *Proc. 1979 ISCAS*, pp 478-481

[6] M. Glez Harbour, J.M. Drake, "Calculation of Multiterminal Resistances in Integrated Circuits", *IEEE Trans. CAS*, vol. CAS-33, no. 4, April 1986, pp 462-465

[7] Albertus J. Kemp, Jacobus A.Pretorius, Willem Smit. "Generation of a Mesh for Resistance Calculation in Integrated Circuits", *IEEE Trans. Computer-Aided Design*, vol. 7, no. 10, Oct 1988, pp 1029-1037

[8] W.C.Elmore, "Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers", *IEEE J. Applied Physics*, vol. 19, Jan 1948, pp 55-63

[9] Shun-Lin Su, Basant B.Rao, Timothy N.Trick, "A simple and accurate node reduction technique for interconnect modeling in circuit extraction", *IEEE International Conf. on CAD:ICCAD-86*, Cat No. 86CH2353-1, 1986, pp 270-273

[10] Jorge Rubinstein, Paul Penfield, Jr., Mark A. Horowitz, "Signal Delay in RC Tree Networks", *IEEE Trans. on CAD*, vol. CAD-2, no. 3, July 1983, pp 202-210

[11] J.L.Wyatt,"Signal Delay in RC Mesh Networks", *IEEE Trans. on Circuits and Systems*, vol. CAS-32, no. 5, May 1985, pp 507-510

[12] William D. Smith, Jack McDonald, Ching-Tang Chang, Ron Jerdonek, "Next : A Hierarachical Layout Verification System for VLSI", *IEEE Int. Conf. on Computer Design : VLSI in Computers ICCD '84-Proceedings*, pp 820-825

[13] Takayasu Sakurai, "Approximation of Wiring Delay in MOSFET LSI", *IEEE J. Solid-State Circuits*, vol. SC-18, no. 4, August 1983, pp 418-426

[14] Robert J.Antinone, Gerald W.Brown, "The Modeling of Resistive Interconnects for Integrated Circuits", *IEEE J. of Solid-State Circuits*, vol. SC-8, no. 2, April 1983, pp 200-203