Research report

Overview

In this quarter so far, I have been mainly working on following research topics: Model order reduction, RF simulation technique investigation, Nonlinear device evaluation in frequency domain and the application of table look-up method in MOS device and in circuit simulation

Following sections will discuss each topic in details with some preliminary results and some thinking

Model reduction

For this part, I worked with guoyong and finished the journal paper. It's a wrap-up of previous research results, plus implementation of Richard's idea on approximating the inversion of symbolic matrix $(G+\Delta G)^{-1}$ and the subsequent transformation matrix to first order. Statistical simulation is also performed to verify the effectiveness of proposed reduction techniques.

RF simulation

After more comprehensive literature reading about this topic, I focused on the capacity issue of RF simulation.

Basically, there are two approaches to this problem. One is the flat matrix approach, joined with iterative solver and pre-conditioners, which is demonstrated by some researchers to be able to simulate whole receiver chains; the other is the hierarchical approach, which breaks down the circuit into sub-blocks and alleviates the big flat matrix problem.

Both approaches have received great amount of research efforts, but still lack the robustness and accuracy required for practical RF simulation purpose.

I have discussed with Karti about the adoption of hierarchical method in Harmonic Balance method: break down circuit into subcircuits according to the topology, then use HB as master simulation technique and shooting method as subcircuit simulation method. However, this idea has already been studied before, and consequently no further attempt is made along this line.

Nonlinear device evaluation in frequency domain

This is initiated by Karti's problem at BMAS workshop and also by Richard's question on how this could be accomplished as one eedesign article claimed.

Currently, major simulators (ADS and Spectre) evaluate nonlinear devices in time domain, and translate the time domain to frequency domain for HB simulation. The translation from frequency domain to time domain and vice versa, presents limitations on the accuracy of HB simulation. The ideal solution would be to evaluate nonlinear devices directly in frequency domain.

I attempted two methods. First one is the integration formula. However, this one suffers the singularity problem in the integrand. The second approach is based on spline functions, but for multi dimensional spline function, the complexity could easily shadow the benefits.

In harmonic balancing method, both voltage and current are represented by their Fourier coefficients. The KCL is satisfied at each node for each frequency. Just like the way in time domain simulation, Newton-Raphson method is adopted to solve harmonic balancing matrix, and Jacobian matrix in frequency domain is computed for each iteration.

The general form of circuit equation is given as follow [12]:

$$\int_{-\infty}^{t} y(t-\tau)x(\tau)d\tau + \frac{d}{dt}q(x(t)) + f(x(t)) - b(t) = 0 \qquad (1)$$

If re-write above equation in frequency domain, then we have:

$$YX + \Omega\Gamma q(\cdot)\Gamma^{-1}X + \Gamma f(\cdot)\Gamma^{-1}X - B = 0 \qquad (2)$$

here Γ is the time to frequency domain transform operator (it will transform time domain waveform to frequency domain spectrum, it's inverse Γ^{-1} will do the opposite transformation). Ω is the differentiation operator in frequency domain.

The Jacobian matrix is given as:

$$J = Y + \Omega\Gamma C\Gamma^{-1} + \Gamma G\Gamma^{-1} \qquad (3)$$

G and C are the conductance and impedance matrices in time domain, they are the matrix representation $\frac{\partial f}{\partial x}$ and $\frac{\partial q}{\partial x}$ at the sampling time points [t_1, t_2, … , t_N] .

$$G = \begin{bmatrix} \left.\frac{\partial f}{\partial x}\right|_{t1} & & \\ & \ldots & \\ & & \left.\frac{\partial f}{\partial x}\right|_{tN} \end{bmatrix}, \quad C = \begin{bmatrix} \left.\frac{\partial q}{\partial x}\right|_{t1} & & \\ & \ldots & \\ & & \left.\frac{\partial q}{\partial x}\right|_{tN} \end{bmatrix}$$

In order to compute the solution, q(X),f(X), C and G will have to be computed at each NR iteration. For the first approach, I tried to reduce the computation of both G and C matrices using partial integration formula, so that the differentiation computation in device load function could be removed to speed up HB analysis.

Following is a simple one-dimensional example of this idea (V is time domain node voltage, A0 is the DC component):

$$g_{00} = \frac{1}{2\pi}\int_0^{2\pi} \frac{\partial f}{\partial V}\cdot\frac{\partial V}{\partial A0}dt = \frac{1}{2\pi}\int_0^{2\pi} f\cdot\frac{V''}{V'}dt \text{ , } g_{ij} = \ldots$$

$C_{00} = \ldots$ $C_{ij} = \ldots$

By doing that, the computation of $\frac{\partial f}{\partial x}$ and $\frac{\partial q}{\partial x}$ are not necessary. However, the V' is at the denominator, and it cause singularity problem. I don't know any good method to deal with this issue.

The second method I thought about is the spline function, since it is the simplest function, which can be easily evaluated in both time domain and frequency domain; if we could express function ***q*** and ***f*** in many spline functions, and then the evaluation in frequency domain is easy.

As a simple example, let us consider a function ***f(x)***, which could be approximated by just 2 spline functions.

$$f(x) \approx f_1(x)w_1(x) + f_2(x)w_2(x) \qquad (4)$$

$f_1(x)$ and ***$f_2(x)$*** are power functions and they are continuous at the boundary up to second derivative (so that the approximation is very smooth across the boundary); w1(x) and w2(x) are window functions (take value 1 within the range, 0 otherwise).

Also, we have:

$$\frac{df(x)}{dx} \approx \frac{df_1}{dx}w_1(x) + \frac{df_2}{dx}w_2(x) \qquad (5)$$

since ***$f_1(x)$*** and ***$f_2(x)$*** are power functions, the first order derivatives would be fairly easy to compute.

If we substitute f(x) and $\frac{df(x)}{dx}$ with the spline functions, the evaluation procedure will be simpler.

The ppt file on the workspace describes the details. As mentioned, this approach could handle simple functions quite well, but for multi terminal device like MOS, the benefit of doing this will not be significant. Because in general it is quite sophisticated to approximate multivariable functions with spline functions, and the monotonic and convex property of original function would easily lost.

Table look-up MOS model and Hierarchical simulation

In the discussion with Zhao and Bo Wan 3 weeks ago, the idea of adopting table lookup method as one general technique in circuit simulation is examined. I looked at this problem afterwards.

In MOS modeling, table lookup method is commonly used along with the analytical or empirical method. This is due to the quasi-static property of MOS device, i.e. the DC current and charge only depend on the current values of terminal voltages. In those situations, the basic table is 2-dimensional, with some additional sub-tables to accommodate the multi-terminal nature of MOS device. In addition, the table lookup model for MOS has to conserve the monotonic and convex properties of the tabulated function, so that the Newton-Raphson iteration will converge to the correct solution.

Generally speaking, the table lookup method could only apply to quasi-static function, that is, the function value does not depend on the past of the inputs. Since most circuits practically designed have capacitive and inductive elements, the table lookup method will not be able to accurately capture the dynamic behavior of the circuit.

The potential speed-up with hierarchical method for circuits, which have many repeating sub-blocks, is investigated also, and a simple formula is given to estimate the time saving.

When the circuit has millions of transistors, the computation bottleneck is the matrix LU. Using the sparse matrix technique, the complexity of LU is estimated as $n^{1.5}$ (n is the number of non-zero elements). If using hierarchical approach to break the circuit into m blocks of equal size, the approximate times of computation saving is $\sqrt{m}$. Furthermore, when there are many same sub-blocks (such as memory blocks), the saving is even more.

Summary

In this quarter, I have been looking for some topics that I could work on for general exam. Several topics are attempted, with no break-through ideas to work further. However, the process gave me some general ideas about the issues in circuit simulation, which I think is very valuable guideline for next step research.