ACCELERATION OF SIMULATED ANNEALING AND ITS APPLICATION TO MICROWAVE DEVICE AND CIRCUIT OPTIMIZATION

Man-Kuan Vai, Jer-Sheng Lin, and Sheila Prasad

Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115

ABSTRACT

Simulated annealing (SA), while being successful as a general optimization technique applicable to many modeling and design problems of microwave devices and circuits, is very time consuming. To alleviate this problem, we propose the creation of a hardware-implemented SA accelerator. The design and evaluation of this SA accelerator, as well as its application to the microwave CAD area, are presented. This accelerator features a highly parallel pipeline structure with a programmable characteristic function. Acceptance prediction is used to further improve the performance. It is evident from simulation results that this accelerator has a significant (several orders of magnitude) speed advantage over an ordinary software implementation.

INTRODUCTION

Simulated annealing (SA), a probabilistic hill-climbing optimization algorithm [1], has been successfully applied to many modeling and design problems of microwave devices and circuits [2-7]. Due to the fact that the SA optimization technique is relatively independent of initial conditions and excellent results have been observed, it has aroused much interest. However, since SA is basically an iterative improvement method, it generally takes a long computational time to converge to a near-optimal solution. Comparing SA to other traditional optimization methods, its slow cooling requirement, which gives SA a leading edge in escaping from being trapped into a local minimum, also contributes to its time consuming feature.

In this paper, we will present our efforts in developing a hardware accelerator for simulated annealing. Parallel simulated annealing has been proposed elsewhere [8]; however, the emphasis was on the placement of VLSI building blocks which, as we will explain, has a quite different set of requirements when considered as an optimization problem. No attempt was made to design and build an SA hardware accelerator for microwave CAD problems.

Before the details of the design and implementation of this SA accelerator are presented, we will first briefly review the optimization method of simulated annealing and its application to microwave CAD problems.

SIMULATED ANNEALING

When an optimization method is used, the modeling and design processes of microwave devices and circuits can be treated in a common framework [5]. Both these processes try to determine the values of some model elements so that the result model is characteristically equivalent to a set of given data. The only difference lies in the source of these data -- measured data in a modeling case and required specifications in a design case. Based on this observation, we will use the modeling problem as a representative case in the following description.

The success of a modeling process depends on the capability of the optimization method which is used to fit the simulated circuit characteristics to measurements. Theoretically, the best solution of an optimization problem exists under a certain well defined objective function and can be guaranteed only by generating and evaluating all possible solutions. However, the size of the solution space (i.e., all the possible solutions) is extremely large and it grows exponentially with the number of variables in the problem. It is generally impossible to perform an exhaustive search to locate the best solution in a reasonable time.

A typical optimization process utilizes an iterative improvement strategy. First, an initial set of estimated parameters is generated as the starting point of a search process, then small variations are made to these parameters at each step to generate a new set of parameters, which is evaluated according to the objective function to be minimized. In order to guarantee the convergence of an optimization process, traditional algorithms are greedy and accept only those changes that can improve the cost of the objective function. One inherent drawback of this type of search is that it can be easily trapped into the local minima of an objective function if good initial values are not available.

CH2870-4/91/0000-1213$01.00 © 1991 IEEE

An approach called simulated annealing (SA) has been proposed and applied as a method to find a near optimal solution for combinational optimization problems [1]. SA associates the statistical mechanics, which deals with the behavior of systems with many degrees of freedom in thermal equilibrium at a finite temperature, with the combinational optimization problem, which finds the minimum of a given function depending on many parameters. In an optimization process, the perturbations of parameters to minimize an objective function is analogous to the displacements of atoms to minimize the energy in an annealing process. The excellent analogy between both processes suggested the application of an annealing process to an optimization problem, and hence the name simulated annealing. In order to apply the concept and mechanism of annealing to the optimization problem, a control parameter pseudo-temperature has to be artificially introduced in a simulated annealing process to simulate the temperature which governs the Boltzman distribution.

An SA optimization process proceeds in a way similar to the traditional iterative improvement methods except that a pseudo-temperature control parameter is set to be a large number at the beginning of the process and its is artificially decreased very slowly during the process. The success of an SA process lies in the fact that it conditionally accepts some error-increasing intermediate solutions to allow the exploration of the solution space in directions which temporarily worsen the objective function, in the hope of eventually escaping from local minima and finding a global minimum.

This technique has received much attention in real world design problems. The SA method has been applied to a variety of microwave CAD applications including device modeling [2, 7], device model simplification [3, 4], and circuit design [5, 6].

PARALLEL SIMULATED ANNEALING

Although simulated annealing is an excellent tool for CAD applications, its effectiveness is unfortunately restricted by its lengthy computational time. Hours of computer time discourages most, it not all, users.

There are generally two approaches to improve the computational speed of an algorithm: parallel processing and direct hardware implementation. Parallel processing speeds up a process by using multiple processors to perform tasks concurrently while a direct hardware implementation achieves the same goal by removing the overheads typically implied by a general-purpose computer. Both approaches are combined and applied in this implementation.

The original SA algorithm, in its current form being applied to microwave CAD problems, has to be modified before it is suitable for direct hardware implementation. Some parts of the algorithm, especially its objective function (i.e., the least squares error between measured and calculated characteristics), has to be modified for the purpose of flexibility. Unlike the original SA application to a building block placement problem which has a general objective function, microwave CAD problems require different models with different characteristic functions to be used, which are determined according to their model topologies. There is virtually no difficulty in changing a characteristic function in a software program. However, this flexibility that is commonly enjoyed in a software program no longer exists in a hardware implementation. In order for the resulting accelerator to be universally applicable, its characteristic function must be programmable.

We have decided to implement the most time consuming part of an SA process, i.e., its iterative improvement loop, into a parallel pipeline structure in this programmable hardware accelerator. The less time consuming tasks and control functions are retained in a software implementation. Figure 1 shows the relationship between software and hardware in this accelerator.

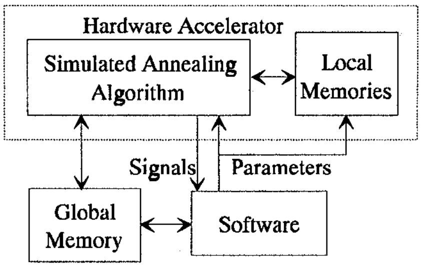

Figure 1 The software-hardware relationship of the proposed SA accelerator.

As mentioned, the original SA algorithm needs to be modified in order to be implemented effectively. The first problem to be solved is that different modeling problems have their own characteristic functions. It is not economical to design a hardware accelerator customized for only one modeling problem. The characteristic function of this accelerator is designed to be universal so that it can be programmed to suit most, if not all, modeling problems. One format of this programmable characteristic function is given below:

$$F(X) = \frac{a_0X_0+\ldots+a_7X_7+j(a_8X_8+\ldots+a_{15}X_{15})}{a_{16}X_{16}+\ldots+a_{23}X_{23}+j(a_{24}X_{24}+\ldots+a_{31}X_{31})} \quad (1)$$

where F(X) is the characteristic (e.g., measured S-parameters) to be modeled, $X = [X_0, X_1, ..., X_{31}]$ is a set of independent variables (e.g., ω, ω^2, ...) and the coefficients, a_i's, represent the model parameters to be determined. An example of using equation (1) is given here to illustrate its usage. The following characteristic function

$$F(X) = j\omega C + \frac{1}{R + j\omega L}$$

can be transformed into

$$F(X) = \frac{-\omega^2LC + j\omega RC}{R + j\omega L} = \frac{a_0X_0 + ja_8X_8}{a_{16}X_{16} + ja_{24}X_{24}.}$$

The parameters can then be specified as $X_0 = \omega^2$, $X_8 = \omega$, $X_{16} = 1$, $X_{24} = \omega$, $a_0 = -LC$, $a_8 = RC$, $a_{16} = R$, and $a_{24} = L$. The values of individual model parameters can then be found readily once a_i's have been determined.

SA continuously modifies the coefficients a_i's to generate new solutions for evaluation. A new solution is retained or discarded, depending upon whether the following Boltzman-like distribution function

$$P(\Delta E, T) = e^{-k * \Delta E/T} > R, \qquad (2)$$

where P is the probability of accepting a cost-increasing solution, ΔE is the cost increment between the current and previous solutions, k is a weighting factor, T is the pseudo-temperature, and R is a random (real) number generated between 0 and 1, is satisfied.

Since the solution used as a starting point at a certain step in the SA is determined by the acceptance of a previous proposed intermediate solution, the accelerator cannot process a new solution until its ancestor has been fully evaluated. This is very not economical since many clock cycles will be wasted. In order to fully utilize a pipeline structure, the accelerator processes intermediate solutions by means of prediction.

The objective here is to predict whether equation (2) will be satisfied by a solution before it is evaluated. Based on this prediction, the system can then continue to process new solutions by assuming whether the old or modified solution should be used for further steps. In this implementation, the assumption depends on the statistics of recent results. If most of the recent changes were accepted (e.g., at an early stage when the pseudo-temperature is relatively high), then the solution still under evaluation is assumed to satisfy equation (2). The old one is used otherwise (e.g., at a final stage when the pseudo-temperature is relatively low).

The prediction cannot, of course, always be right and an adjustment may become necessary. In order to recover from an incorrect assumption, the local memories of the accelerator (see Figure 1) store the previously processed data and parameters until they are no longer needed. Whenever a prediction is proved to be incorrect, the pipeline has to be flushed and brought back to the point where the incorrect assumption was made. The system then recovers itself by copying back the stored information and restarts the pipeline.

The operation of this accelerator is distributed among three parallel pipelines, as shown in Figure 2.

Pipeline P1 generates two random numbers and uses them to decide which coefficient a_i should be changed and its amount of change. Queues are used in this pipeline to store these newly generated parameters until their related costs and thus their acceptance have been evaluated using equation (2). This is done for the preparation of flushing the pipelines which may become necessary in the future.

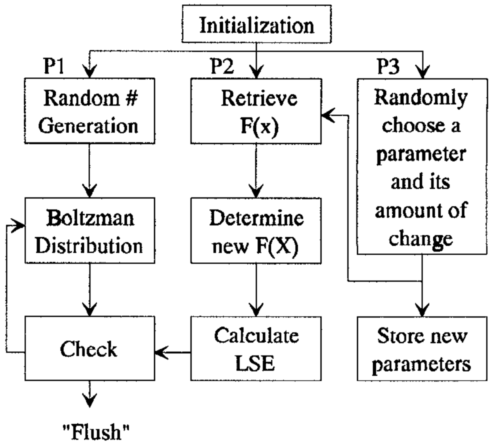

Figure 2 The parallel pipelines in the accelerator.

Based on the information provided by P1, a second pipeline, P2, calculates the characteristic of a newly generated solution and generates the least squares error (LSE) between the calculated and measured characteristics. This pipeline consists of many floating point arithmetic calculations and is responsible for a major part of calculations in the optimization process. In order to speed up the process, the following rearrangement of the cost function is performed:

$$F = \frac{a_0X_0 + \ldots + a_7X_7 + j(a_8X_8 + \ldots + a_{15}X_{15})}{a_{16}X_{16} + \ldots + a_{23}X_{23} + j(a_{24}X_{24} + \ldots + a_{31}X_{31})} = \frac{f_0 + jf_1}{f_2 + jf_3.}$$

Instead of completely recalculating F every time an a_i is changed, memories are provided for storing individual f_j's (j = 0 to 3) and only the one that is involved with the selected a_i is recalculated by

$$f_j(\text{new}) = f_j(\text{old}) + C_i * X_i,$$

where C_i is the amount of change applied to a_i. This rearrangement allows a big time saving by replacing a large number of floating point operations with a single multiplication and an addition. The LSE is then calculated.

The last pipeline P3 deals with the decision part of an SA process. This pipeline evaluates the acceptance condition, checks the results delivered by P2 and generates a "flush" signal if necessary. Two major tasks of this pipeline are to gradually lower the pseudo-temperature and to check the acceptance of new solutions. The pseudo-temperature is lowered by multiplying its current value by a factor α ($0 < \alpha < 1$). The checking operation specified in eq. (2) is rearranged to further reduce the time spent in this pipeline.

P3 receives the E(new) (LSE) from P2 and checks if the solution should be accepted according to eq. (2). Since $\Delta E = E(\text{new}) - E(\text{old})$ (i.e., the difference between LSE's), Eq.

(2) can be rewritten into

$$E(\text{new}) < E_{max} = E(\text{old}) - \frac{T}{k} * \ln(R). \qquad (3)$$

E_{max} can be interpreted as the maximum value for an acceptable new solution generated from an old one with E(old). The R. H. S. of eq. (3) can be calculated in parallel with the computation of E(new) which is performed in P2.

We have simulated and verified this accelerator using VHDL (VHSIC Hardware Description Language), an IEEE standard hardware description language. Comparing the performance of this hardware SA accelerator with a typical software implementation, several orders of magnitude of improvement in the computational speed has been observed. To give an idea of the significance of this speedup factor, the modeling of a spiral inductor to be used in microwave applications is carried out in both a Sun 3/50 workstation (a software SA implementation) and our hardware accelerator under simulation, it was found that the computation time has been reduced from about 10 minutes (CPU time, the actual elapsed time is much longer) to less than 1 second. The result of this modeling is shown in Figure 3.

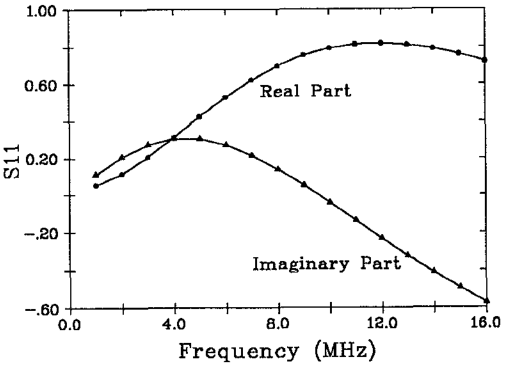

Figure 3 The modeling result of a spiral inductor.

SUMMARY

In this paper, we have proposed and described the design of a hardware SA accelerator and its application to microwave CAD modeling and design problems. The original SA process has been modified to suit a hardware implementation. This accelerator includes a parallel pipeline structure with a programmable characteristic function. A prediction of solution acceptance is used to further improve the accelerator performance. Simulation results have indicated that several orders of magnitude in speed improvement can be readily available.

REFERENCES

[1] S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi, "Optimization by Simulated Annealing," **Science**, vol. 220, pp. 671-680, 1983.

[2] M.-K. Vai, S. Prasad, N. C. Li and F. Kai, "Modeling of Microwave Semiconductior Devices Using Simulated Annealing Optimization," **IEEE Trans. Electron Devices**, vol. 36, pp. 761-762, 1989.

[3] M.-K. Vai and D. Ng, "A Technology-Independent Device Modeling Program Using Simulated Annealing Optimization," **Proc. of IEEE Custom Ingetrated Circuits Conference**, pp. 9.4.1-9.4.4, 1989.

[4] M.-K. Vai, D. Ng and s. Prasad, "Model Minimization for Electron Devices Using Simulated Annealing in Conjunction with Parameter Extraction," **IEE Electronics Letters**, vol. 26, No. 13, pp. 892-894, 1990.

[5] M.-K. Vai and S. Prasad, "Computer-Aided Design of MESFET Distributed Amplifier," **IEEE Trans. on Microwave Theory and Techniques**, vol. MTT-38, pp. 345-349, 1990.

[6] M.-K. Vai, S. Prasad and B. Meskoob, "Computer-Aided Design of Monolithic Distributed Amplifiers with Yield Optimization," **IEEE MTT-S International Microwave Symposium Digest**, pp. 347-350, 1990.

[7] G. L. Bilbro, M. B. Steer, R. J. Trew, C.-R. Chang and S. G. Skaggs, "Extraction of the Parameters of Equivalent Circuits of Microwave Transistors Using Tree Annealing," **IEEE Trans. on Microwave Theory and Techniques**, vol. MTT-38, pp. 1711-1718, 1990.

[8] M. D. Durand, "Parallel Simulated Annealing Accuracy vs. Speed in Placement," **IEEE Design & Test of Computers**, vol. 6, no. 3, pp. 8-34, June 1989.