

Parallel Genetic Algorithm for SPICE Model Parameter Extraction

Yiming Li and Yen-Yu Cho

National Chiao Tung University
Department of Communication Engineering
1001, Ta-hsueh Rd., Hsinchu 300, Taiwan
{ymli, yycho}@ymlabcad02.eic.nctu.edu.tw

Abstract

Models of simulation program with integrated circuit emphasis (SPICE) are currently playing a central role in the connection between circuit design and chip fabrication communities. An automatic model parameter extraction system that simultaneously integrates evolutionary and numerical optimization techniques for optimal characterization of very large scale integration (VLSI) devices has recently been advanced [1]. In this paper, to accelerate the extraction process, a parallelization of the genetic algorithm (GA) for VLSI device equivalent circuit model parameter extraction is developed. The GA implemented in the extraction system is mainly parallelized with a diffusion scheme on a PC-based Linux cluster with message passing interface libraries. Parallelization of GA is governed by many factors, which affect the quality of extracted parameters and its efficiency. The diffusion GA is superior to an isolated GA, and the superiority of the diffusion GA is significant when the number of devices to be optimized is increased. Theoretical estimation and preliminary implementation show that there is an optimal number of processors with respect to the number of devices to be extracted. Benchmark results, such as speedup and efficiency including accuracy of extraction are presented and discussed for different sets of realistic multiple VLSI devices to show the robustness and efficiency of the method. We believe that the practical implementation of the parallel GA approach benefits the engineering of SPICE model parameter extraction in modern electronic industry.

1. Introduction

The simulation program with integrated circuit emphasis (SPICE) models, such as BSIM, HiSIM, and

PSP models characterize very large scale integration (VLSI) device's electrical characteristics (e.g., current-voltage (I-V) curves), which are associated with a set of optimized parameters [1]-[5]. For the problem of the SPICE model parameter extraction, it usually refers to several hundred I-V points. It forms a multidimensional nonlinear optimization problem; therefore, model parameter extraction of the VLSI device is a time consuming task, and requires engineering expertise to find a set of proper parameters with reasonable physical meanings [1]-[6]. Many researches for model parameter extraction, such as pure GA or numerical optimization methods have been reported [1],[6]-[11], and most of them were performed separately. Unfortunately, such methods may not work efficiently for VLSI devices in the regime of deep-submicron. To overcome the problem above, we have recently developed a hybrid intelligent model parameter extraction technique which bases on the genetic algorithm, the monotone iterative Levenberg-Marquardt method, and the neural network algorithm [1]. A prototype was successfully implemented according to the proposed methodology. Extraction in a global sense shows good accuracy for the 90 nm n-type metal-oxide-semiconductor field effect transistors (NMOSFETs) by several testing cases. However, in order to accelerate the extraction process of the developed prototype for larger scale optimization problem, it is necessary to perform the parallelization of the system.

In this paper, we successfully implement a parallel optimization platform for VLSI device model parameter extraction on a Linux-based PC cluster with message passing interface (MPI) libraries. The GA implemented in the early developed system with 16 PCs is parallelized with a diffusion scheme which forms a 2D-grid network. When the stage of GA is performed on a processor, chromosomes are simultaneously exchanged among those results that computed by its neighbor

4 processors. Optimization process is then going to the next step according to the system configuration of the hybrid intelligent model parameter extraction technique [1]. Extraction will be terminated when the specified stopping criterion is met. Our extraction experience shows that this approach has distinguished results when the dimension of the problem is significantly large, such as parameter extraction for more than 8 VLSI. Compared with an isolated parallel GA, more than 33% difference in the evolution time is found between the two parallelization algorithms when 16 devices are optimized. In terms of several benchmarks, such as speedup, efficiency, and accuracy, results for different examples with multiple VLSI devices are examined to show the robustness and efficiency of the method. Theoretical estimation and preliminary implementation show that there are an optimal number of processors with respect to the number of devices to be extracted. For example, according to our theoretical calculation, the optimal number of units is 18 which is close to the practically obtained result (16 units), shown in the table 3, 4, and 5, respectively.

This article is organized as follow. In Sec. 2, we briefly describe our extraction system and state the architecture of parallel computing algorithms. In Sec. 3, we show the extraction results for single and multiple deep-submicron and sub-100 nm N-MOSFET devices. Finally, we draw conclusions.

2. Parallelization of the Model Parameter Extraction System

Under this section, the proposed architecture for the parallel optimization platform is described first, followed by a theoretical estimation on the optimal parallel performance of the diffusion GA.

2.1 The Parallel Architecture

Mathematically, model parameter extraction is a multidimensional nonlinear optimization problem, where the number of parameters is larger than 100. The main goal of device model parameter extraction is to minimize the error between the extracted result and the measurement, where the extracted result is obtained through the equation below:

$$I_{DS}^{ex} = I_D(\vec{p}, \vec{v}, \vec{d}), \quad (1)$$

where the I_{DS}^{ex} is the I-V functions (e.g., I-V points, shown in Fig. 4) to be optimized; the I_D is a selected compact model [1]-[4], which contains more than 40

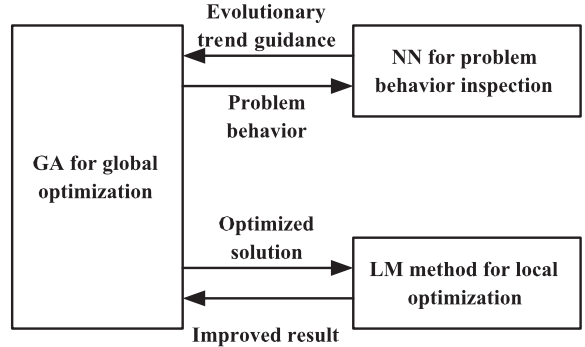


Figure 1. The architecture of the developed extraction system.

subequations in the BSIM model, for example. Vectors \vec{p} , \vec{v} , \vec{d} are the parameter sets to be extracted, the bias condition for simulation, and the device geometry, respectively. There are at least 50 I-V points forming an I-V curve, 5 I-V curves forming a set of I-V curves, and 4 sets of I-V curves to characterizing a single device behavior. Therefore, a device model parameter extraction problem can be formed as follow:

$$\min(\sqrt{\sum_d \sum_{cs} \sum_c \sum_p (I_{DS}^{ex} - I_{DS}^{me})^2}), \quad (2)$$

where I_{DS}^{me} is the measured I-V point, and d , cs , c , p refer to the number of devices, curve sets, curves, and I-V points, respectively. When perform a model parameter extraction with 16 devices as target, there are 16000 I-V points need to be minimized, and the number of parameters are more than 100. The nonlinear optimization problem is subject to proper physical constraints.

This large scale optimization problem with massive computation is performed on our early proposed extraction system. The developed hybrid optimization platform integrates the genetic algorithm, the monotone iterative Levenberg-Marquardt method, and the neural network algorithm, shown in Fig. 1. When the GA obtains a solution, the monotone iterative Levenberg-Marquardt method is activated to search for the nearby local optima, and the neural network algorithm suggests proper searching directions according to the current result and physical constrain. The detailed description of this extraction system is reported somewhere else [1].

Although the extraction system has been proposed and implemented successfully, facing a larger scale

complex optimization problem with massive computation still requires enormous amount of time, thus a parallelization technique is required. On the other hand, the time acquired by the monotone iterative Levenberg-Marquardt method and neural network algorithm can be regarded as instant compared with the time cost of GA. Therefore, only the GA is required a parallelization technique. Application of parallelization to GA provides an efficient way to reduce the computing time [12]-[15]. GA is a self-adaptive optimization strategy that mimics a living system, it usually contains five operations: encoding, fitness evaluation, selection, crossover, and mutation. We briefly state GA methods for the MOSFET device model parameter extraction. The design of gene encoding strategy depends on the property of problem. In this problem model, there are more than 100 parameters and all variables are floating-point numbers. The fitness function measures the error between simulated result and realistic measurement data. As for the reproduction issue, we adopt the tournament selection with floating point operators as the selection strategy not only this hybrid strategy selects better chromosomes but also keeps weak ones for few generations to achieve higher population diversity. For the crossover scheme, in MOSFET device model, all parameters to be optimized can be classified into several categories and each of them stands different physical characteristics [1]. Under this consideration, we take a uniform crossover scheme to preserve the physical characteristics of the parents; and based on our simulation experience, it is more effective than single and two-point crossover schemes. Finally, the mutation strategy changes the mutation rate dynamically to keep the population diversity. Such evolutionary optimization may take a long time when the dimension of investigated problem is large; in particular, for nanoscale VLSI device model parameter extraction [1]. To reduce the time cost of optimization, parallel schemes are taken into consideration.

It is known that the parallelization of GA can be classified into five different models, the isolated, the ring migration, the neighborhood migration, the unrestricted migration, and the diffusion GA [15]. Each unit in the isolated configuration performs the extraction tasks separately, and there is no data communication among units. The obvious advantage of the isolated architecture is spending less communication time in the extraction procedure; however, the isolated evolutionary environment may lead to the striking decrease of the population diversity. Contrast to the isolated GA, each extraction unit of the migration GA is treated as a separated breeding unit, and the migrations between each unit occur from time to time to pro-

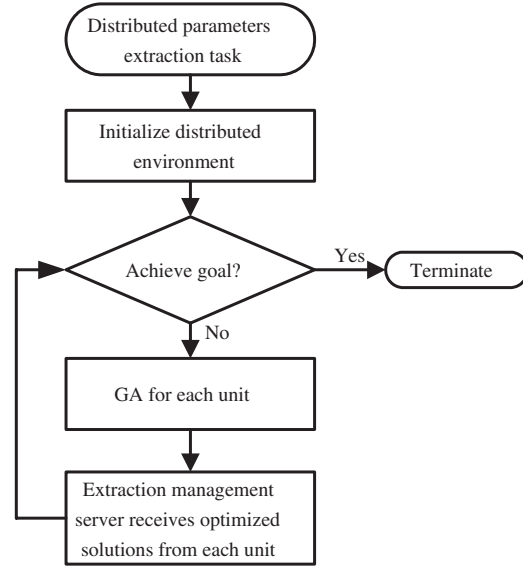


Figure 2. An execution flowchart of the parallel GA that implemented in our model parameter extraction system.

note the proliferation of good genetic building block. Most famous GA migration methods are the ring, the unrestricted, and the diffusion GA. The method of the diffusion GA implemented in this work is shown below.

Begin Diffusion GA

```

For each unit
Begin
  Initialization
  While not finished
  Begin
    Evaluation
    Send self results to 4 neighbors
    Receive results from 4 neighbors
    Selection
    Crossover
    Mutation
  End While
End For
  
```

End Diffusion GA

Each individual is assigned to a specific location, and the migration is permitted between a set of specific neighbors.

In advanced VLSI device model parameter extraction, parameters according to their engineering meanings can be classified into several groups, and each

group represent specific physical phenomenon [1]. By applying the diffusion GA, we can assign each column in the 2D-grid units to optimize different groups of parameters. This configuration also corresponds to our optimization method thus here we conclude that the diffusion GA is the most suitable distributed configuration. According to our extraction experience, the isolated GA and diffusion one are compared and focused for a series of comparison. The parallel extraction system is implemented on a PC-based Linux cluster with 16 units. Each unit is connected to a high speed network switch physically and performs automatic parameter extraction. The entire system architecture can be classified into two modules, the management server and the extraction cluster. The server controls whole extraction system. It analyzes the complexity of the problem. Based on the analysis results, the server sets the configurations of the system architecture up, and allocates proper computing resources. In the extraction process, the server monitors the extraction process, backs necessary information up, controls the extraction flow, and communicates with the other extraction modules. The extraction cluster consists many extraction units, each one can be regarded as an independent extraction entity or participate in the distributed parameters extraction process under the control of the extraction management server. Figure 2 shows the working flow of our distributed parameter extraction engine. Once the procedure starts, the environment is initialized firstly, and each unit (or processor) begins their job, and sends the current result to server if data transmission is required. This procedure loops until the fitness score is reached or the evolution time is up.

2.2 A Theoretical Estimation

Furthermore, a theoretical estimation on the optimal parallel performance of the diffusion GA is discussed for the implemented parallel extraction system. Assume that there are p processors, the communication time cost is T_c , n indicate the population size, and the total evaluation time is T_f . In our implemented diffusion GA, we set the number of neighbor of each unit as 4. Thus the entire time cost for one generation T_p is given by

$$T_p = pT_c + \frac{nT_f}{p} + 4pT_c = 5pT_c + \frac{nT_f}{p}, \quad (3)$$

where the $4pT_c$ is the extra communication cost from the diffusion GA. As more processors are used, the computation time T_p decreases as desired, but the communication time increases. This tradeoff entails the existence of an optimal number of processors that mini-

mizes the execution time. To find the optimal result, we set $\partial T_p / \partial p = 0$ and solve the corresponding equation for p :

$$p^* = \sqrt{\frac{nT_f}{5T_c}}. \quad (4)$$

The time that a sequential GA uses in one generation is $T_s = nT_f$, and to ensure that the parallel implementation has a better performance than a sequential GA the following relationship holds

$$\begin{aligned} S_p &= \frac{T_s}{T_p} \\ &= \frac{nT_f}{(nT_f/p) + 5pT_c} \\ &= \frac{nT_f/5T_c}{(nT_f/5T_cp) + p} \\ &> 1. \end{aligned} \quad (5)$$

This ratio is the parallel speedup for the diffusion GA, and it formalizes the intuitive notion that parallel does not benefit problems with very short evaluation times. Another concern when implementing parallel algorithms is to keep the processor utilization high. Formally, the efficiency of a parallel program is defined as the ratio of the parallel speedup over the number of processors:

$$E_f = \frac{T_s}{T_pp} = \frac{S_p}{p}. \quad (6)$$

Theoretically, the parallel speedup should be equal to the number of units to be used, and the efficiency equals 100%. However, the cost of communications causes the efficiency to decrease as more units are used. To set an economical number of units (p_e) that maintain a pre-estimated efficiency \hat{E}_f , we let Eq. (6) equal to \hat{E}_f and solve the corresponding equation for p . The computed p_e is given by

$$p_e = \sqrt{\frac{1 - \hat{E}_f}{\hat{E}_f} \frac{nT_f}{5T_c}}. \quad (7)$$

We note that $p_e = p^*$ when \hat{E}_f is 0.5. The maximum speedup achievable by the diffusion GA equals half optimal number of units. In our experiment, the communication time cost T_c is approximately 32 ms, and the evaluation time T_f is around 0.068 second for 16 devices simulation, and the population size is set to 800. As a result, from Eq. (4), we have

$$p^* = \sqrt{\frac{nT_f}{5T_c}} = \sqrt{\frac{800 * 0.068}{5 * 32 * 10^{-3}}} \cong 18.44. \quad (8)$$

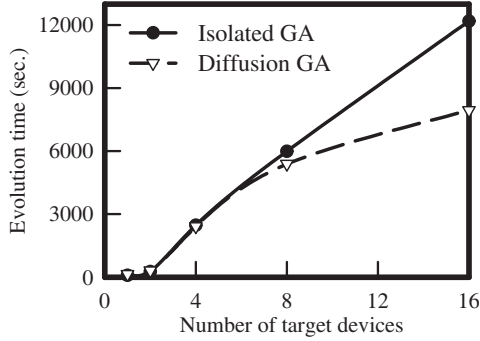


Figure 3. A comparison of the time cost versus the number of target devices for extracting multiple devices with the BSIM4 model (more than 100 parameters have to be optimized with respect to 1600 I-V points) by using the 16 extraction units with the isolated and the diffusion GAs. The root-mean-square (RMS) error is set to be 75%, 25%, and 7% for the leakage current, linear and saturation regions, respectively.

According to the point of view above, if more units are included in the parallel extraction system, the speedup will not make any further improvement; moreover, the speedup might decrease due to heavy communication in the used network. We practically implement such parallelization schemes in our hybrid optimization prototype for VLSI device model parameter extraction. Achieved results confirm the theoretical estimation.

3 Results and Discussion

In this section, three issues are examined. The first one shows the performance comparison of the isolated and diffusion GA, the second issue demonstrates the robustness of our optimization method. Finally, the parallelization configuration of this work is discussed. In our extraction experiment, the industrial standard BSIM4 SPICE model is adopted. Figure 3 shows a comparison of the amount of evolution time with respect to the number of extracted devices between the isolated and the diffusion GA. As show in this figure, the evolution time is almost the same as the search domain is small. However, when the search domain is increased, i.e., the number of devices to be extracted is greater than 4 devices, the superiority of the diffusion GA is observed. When the number

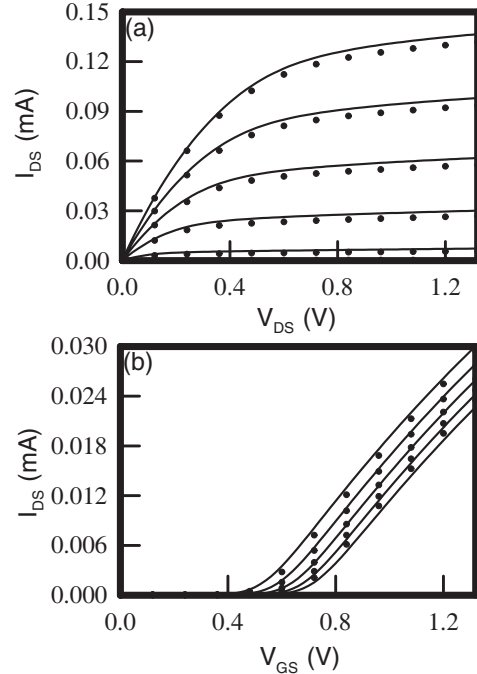


Figure 4. The extracted (solid line) and measured (dot line) for the 350 nm N-MOSFET with the BSIM4 SPICE model, where the device width is 1.2 μm . Plot (a) is the result of $I_D - V_D$, where gate bias (V_{GS}) varies from 0.4 V (the lowest curve) to 1.4 V with step = 0.2 V and bulk bias (V_{BS}) = 0 V and (b) is the result of $I_D - V_G$, where V_{BS} varies from 0 (the left curve) to -1.2 V with step = 0.3 V.

of the target devices is increased to 16, the 33% speedup of the evolution time of the diffusion GA is observed, compared with the speedup of the isolated one. However, for problem with small search domain, such as 1 or 2 devices, the difference between two parallel methods is insignificant. With this experiment, we suggest that the diffusion GA is one of suitable distributed methods in parallelization for this problem.

We further perform a series of experiments to examine the accuracy and efficiency of the proposed method. Selecting one result from 16 devices, Figs. 4 and 5 show the optimized result, where Fig. 4 is the I-V curves and Fig. 5 is the first derivatives of the corresponding original I-V curves. Comparison between the measurement data (the dot lines) and the simulation (the solid lines) with the two different sets of the extracted param-

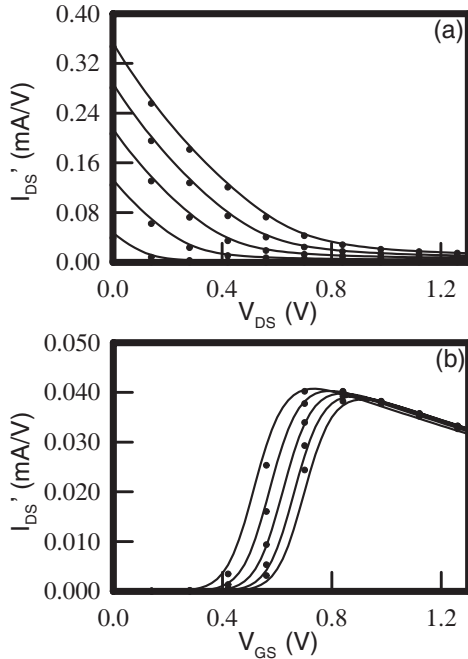


Figure 5. The derivatives of the extracted (solid line) and measured (dot line) for the same device shown in Fig. 4. Plot (a) is the result of $I_D - V_D$, where V_{GS} varies from 0.4 (the lowest curve) to 1.4 V with step = 0.2 V and $V_{BS} = 0$ V and (b) is the result of $I_D - V_G$, where V_{BS} varies from 0 (the left curve) to -1.2 V with step = 0.3 V.

ters demonstrates good accuracy of the optimization method. Table 1 and 2 shows the extraction result for 16 N-MOSFETs of 90 nm technology. As shown in this table, RMS error of curves is strictly within 3% and 6% for all original curves and the first derivative of all original curves, respectively. We note that the first derivatives with respect to the curves of $I_D - V_D$ and $I_D - V_G$ is defined by

$$I_D - V_D' = \frac{\partial I_D - V_D}{\partial V_D} \quad (9)$$

and

$$I_D - V_G' = \frac{\partial I_D - V_G}{\partial V_G}. \quad (10)$$

Figures 4 and 5, and Table 1 and 2 confirm the accuracy of the proposed method with respect to

Table 1. A RMS error list of the optimized parameter compared with the measured data for 16 CMOS devices. The device geometries are in μm . The oxide thickness of target devices are 3.36 nm and the working temperature is settled at 298.15 k.

Device Geometry ($\mu\text{m}/\mu\text{m}$)	Errors	
	$I_D - V_D$	$I_D - V_G$
L/W (0.09/0.6)	2.81%	2.41%
L/W (0.35/0.6)	2.24%	2.34%
L/W (0.80/0.6)	1.38%	1.93%
L/W (1.2/0.6)	1.34%	0.98%
L/W (0.09/1.2)	2.99%	2.74%
L/W (0.35/1.2)	2.18%	2.36%
L/W (0.80/1.2)	1.25%	2.19%
L/W (1.2/1.2)	1.07%	0.92%
L/W (0.09/10.0)	2.32%	2.45%
L/W (0.16/10.0)	2.21%	2.51%
L/W (0.18/10.0)	1.98%	2.05%
L/W (0.24/10.0)	1.79%	2.21%
L/W (0.35/10.0)	2.84%	2.63%
L/W (0.50/10.0)	2.65%	2.84%
L/W (0.80/10.0)	2.89%	2.37%
L/W (1.2/10.0)	2.59%	2.31%

different number of extracted NMOSFET devices.

As shown in Fig. 6, the experiment verifies the capability of the implemented parallel extraction system with respect to different number of working processors and different problem sizes. The accuracy for all extracted VLSI devices is strictly set to be within 3% error for all original curves and 6% error for the first derivative of all original curves. Table 3 to 5 shows the benchmark results and confirms that the speedup is increased as the number of units is increased. On the other hand, it is known that the efficiency appears to have a trend of decrease which confirms the optimal parallelization of GA [14]-[15]. We concluded that the most suitable number of processors and acceptable execution time should be 8 processors for extracting 4 and 8 devices with the BSIM4 SPICE model and 16 processors for extracting 16 devices. More detailed data are listed in Tab. 3, 4, and 5, respectively.

Table 2. A RMS error list for 16 CMOS devices. $I_D - V_D'$ and $I_D - V_G'$ refer to the first derivatives of $I_D - V_D$ and $I_D - V_G$, respectively.

Device Geometry ($\mu m/\mu m$)	Errors	
	$I_D - V_D'$	$I_D - V_G'$
L/W (0.09/0.6)	5.95%	5.79%
L/W (0.35/0.6)	5.67%	5.12%
L/W (0.80/0.6)	3.34%	2.35%
L/W (1.2/0.6)	3.38%	1.84%
L/W (0.09/1.2)	4.75%	5.41%
L/W (0.35/1.2)	3.58%	3.92%
L/W (0.80/1.2)	2.68%	4.08%
L/W (1.2/1.2)	1.08%	1.44%
L/W (0.09/10.0)	3.62%	3.56%
L/W (0.16/10.0)	2.41%	3.89%
L/W (0.18/10.0)	2.83%	2.45%
L/W (0.24/10.0)	2.59%	3.21%
L/W (0.35/10.0)	5.42%	5.84%
L/W (0.50/10.0)	5.03%	5.98%
L/W (0.80/10.0)	5.25%	5.79%
L/W (1.2/10.0)	5.82%	4.94%

Table 3. Performance comparisons of the parallelization with respect to 4 devices using the diffusion GA approach.

Units	Time (sec.)	Speed up	Efficiency
1	34581	1	100%
4	13098	2.64	66.00%
8	8214	4.21	52.62%
16	6276	5.51	34.43%

Table 4. Performance comparisons of the parallelization with respect to 8 devices using the diffusion GA approach.

Units	Time (sec.)	Speed up	Efficiency
1	90984	1	100%
4	39048	2.33	58.47%
8	23963	3.84	48.12%
16	15648	5.81	36.34%

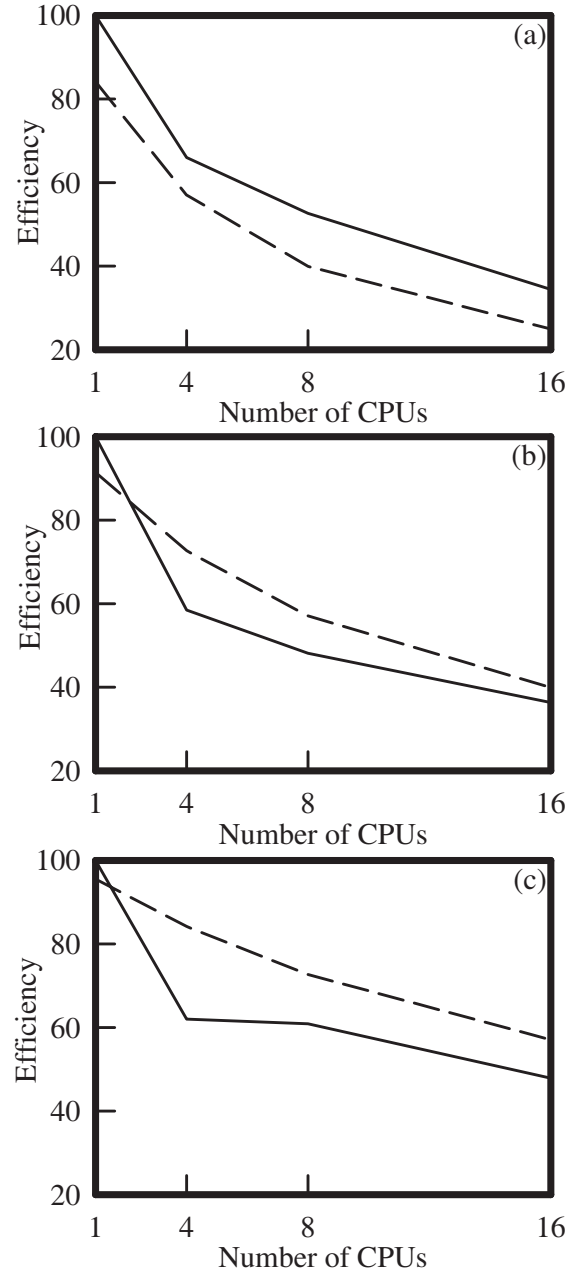


Figure 6. Efficiency comparison of the experiment. The dash lines are the theoretical predictions and the solid lines are the experimental results for (a) $T_f = 17$ ms, (b) $T_f = 34$ ms, and (c) $T_f = 68$ ms, respectively.

Table 5. Performance comparisons of the parallelization with respect to 16 devices using the diffusion GA approach.

Units	Time (sec.)	Speed up	Efficiency
1	260772	1	100%
4	105150	2.48	62.00%
8	53546	4.87	60.87%
16	34043	7.66	47.87%

4. Conclusions

In this paper, parallelization of the genetic algorithm for VLSI device equivalent circuit model parameter extraction has been developed. The GA implemented in the extraction system has mainly been parallelized with a diffusion scheme on a 16-PC-based Linux cluster with MPI libraries. Parallelization shows that the diffusion GA is superior to an isolated GA, and the superiority of the diffusion GA is significant when the number of devices to be optimized is increased. Estimation on the optimal number of processors with respect to the number of devices to be extracted was considered. Preliminary implementation has shown a good agreement with the theoretical estimation in the developed prototype. Speedup and efficiency including accuracy of extraction have been reported and discussed for different sets of realistic multiple VLSI devices. The practical implementation of parallel GA approach benefits the engineering of SPICE model parameter extraction. To validate the developed parallel intelligent model parameter extraction prototype for sub-65 nm VLSI devices, more advanced SPICE models, such as HiSIM and PSP models are currently implemented in this system. In addition, we perform the extraction on a 32-units PC-based Linux cluster for much higher performance computation.

5. Acknowledgments

This work was supported in part by Taiwan National Science Council (NSC) under Contract NSC-94-2215-E-009-084 and Contract NSC-94-2752-E-009-003-PAE, by the Ministry of Economic Affairs, Taiwan under Contract 93-EC-17-A-07-S1-0011, and by the Taiwan semiconductor manufacturing company under a 2005-2006 grant.

References

- [1] Y. Li, and Y.-Y. Cho Intelligent BSIM4 model parameter extraction for sub-100 nm MOSFET era. *Japanese Journal of Applied Physics*, 43:1717–1722, 2004.
- [2] BSIM4.2.1 MOSFET model users manual. *U.C. Berkeley, CA*, 2001.
- [3] STARC, HiSIM1.1.1 users manual. *Hiroshima University, Japan*, 2002.
- [4] G. Gildenblat, X. Li, H. Wang, and W. Wu Introduction to PSP MOSFET model. *Proceedings of the 2005 Workshop on Compact modeling, Anaheim, CA*, 19–24, 2005.
- [5] T.L. Quarles The SPICE3 implementation guide. *Technical Report No. UCB/ERL M89/44*, 1989.
- [6] P.R. Karlsson and K.O. Jeppson A direct extraction algorithm for a submicron MOS transistor model. *Proceedings of the 1993 International Conference on Microelectronic Test Structures*, 157–162, 1993.
- [7] P.R. Karlsson and K.O. Jeppson A direct method to extract effective geometries and series resistances of MOS transistors. *Proceedings of the 1994 International Conference on Microelectronic Test Structures*, 184–189, 1994.
- [8] H. Kunii and Y. Kinouchi Parameter estimation of lumped element circuit for tissue impedance. *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 20:3108–3111, 1998.
- [9] H. Arsham, M. Gradisar, and M. I. Stemberger Linearly constrained global optimization: a general solution algorithm with applications. *Applied Mathematics and Computation*, 134:345–361, 2003.
- [10] Y. Li, Y.-Y. Cho, C.-S. Wang, and K.-Y. Huang A genetic algorithm approach to InGaP/GaAs HBT parameters extraction and RF characterization. *Japanese Journal of Applied Physics*, 42:2371–2374, 2003.
- [11] John Holland Adaptation in natural and artificial systems. *University of Michigan Press, Ann Arbor, Mich.*, 1975.
- [12] D. A. Van Veldhuizen, J. B. Zydallis, and G. B. Lamont Evolutionary computing and optimization: Issues in parallelizing multiobjective evolutionary algorithms for real world applications. *Proceedings of ACM symposium on applied computing*, 595–602, 2002.
- [13] P.K. Nanda, B. Ghose, and T.N. Swain Parallel genetic algorithm based unsupervised scheme for extraction of power frequency signals in the steel industry. *IEEE Proceedings: Vision, Image and Signal Processing*, 149:204–210, 2002.
- [14] E. Cantú-Paz and D. E. Goldberg Efficient parallel genetic algorithms: theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 186:221–238, 2000.
- [15] E. Cantú-Paz Efficient and accurate parallel genetic algorithms. *Kluwer Academic Publishers, Boston*, 2000.