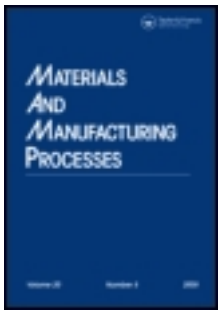

Materials and Manufacturing Processes
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/lmmp20

Parallel Genetic Algorithm for Intelligent Model Parameter Extraction of Metal-Oxide-Semiconductor Field Effect Transistors

Yiming Li [a]
[a] Department of Communication Engineering , National Chiao Tung University , Hsinchu City, Hsinchu, Taiwan
Published online: 19 Feb 2009.

To cite this article: Yiming Li (2009) Parallel Genetic Algorithm for Intelligent Model Parameter Extraction of Metal-Oxide-Semiconductor Field Effect Transistors, Materials and Manufacturing Processes, 24:3, 243-249, DOI: 10.1080/10426910802675814

To link to this article: http://dx.doi.org/10.1080/10426910802675814

Materials and Manufacturing Processes, 24: 243–249, 2009

ISSN: 1042-6914 print/1532-2475 online
DOI: 10.1080/10426910802675814

Taylor & Francis
Taylor & Francis Group

Parallel Genetic Algorithm for Intelligent Model Parameter Extraction of Metal-Oxide-Semiconductor Field Effect Transistors

YIMING LI

Department of Communication Engineering, National Chiao Tung University, Hsinchu City, Hsinchu, Taiwan

Equivalent circuit model of semiconductor devices associated with a set of optimized parameters currently plays a central role in the circuit design and semiconductor manufacturing communities. An intelligent model parameter extraction system that simultaneously integrates evolutionary and numerical optimization techniques for optimal characterization of sub-100 nm metal-oxide-semiconductor field effect transistors (MOSFETs) has recently been advanced [1]. In this article, to accelerate the extraction process, parallelization of the genetic algorithm (GA) for the intelligent model parameter extraction system of MOSFETs 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 various factors, which affect the quality of extracted parameters and its computational efficiency. The result obtained in this study shows that the diffusion GA is superior to an isolated GA, and the superiority of the diffusion GA becomes significant when the number of MOSFETs to be optimized is increased. Theoretical estimation and preliminary numerical implementation of parallel GA show that there exist 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 sub-100 nm devices to show the robustness and efficiency of the method. Practical implementation of the parallel GA approach benefits the engineering of device model parameter extraction in nowadays semiconductor manufacturing industry.

Keywords Device model; Diffusion scheme; Efficiency; Equivalent circuit; Genetic algorithm; Metal-oxide-semiconductor field effect transistors (MOSFETs); Parallelization; Parameter extraction; Speedup; Sub-100 nm.

1. INTRODUCTION

Electrical characteristics (e.g., current-voltage (I–V) curves) of metal-oxide-semiconductor field effect transistors (MOSFETs) are characterized through device models together with a set of optimized parameters [1–5]. For the problem of model parameter extraction, it in general refers to several hundred I–V points; consequently, it forms a multidimensional nonlinear optimization problem. The model parameter extraction of the MOSFETs is thus a time-consuming task, which strongly depends on engineering expertise to find a set of proper parameters with reasonable physical meanings. Applications of genetic algorithm (GA) and pure numerical optimization methods in studies of model parameter extraction problems have been reported [6–14] and have exhibited their advantages for conventional MOSFETs in the early years. Unfortunately, those methods encounter computational efficiency issues for nowadays sub-100 nm MOSFETs' technologies. Bases upon the GA, the Levenberg–Marquardt (LM) method, and the neural network (NN) algorithm, we have recently developed an intelligent model parameter extraction technique for automatic extraction of equivalent circuit model of sub-100 nm MOSFETs [1]. A prototype was further implemented according to the proposed methodology [1]. Extraction in a global sense has shown good accuracy for the 90 nm MOSFETs by several testing cases. However, in order to accelerate the extraction process of the developed intelligent prototype for the practical optimization problem resulting from semiconductor manufacturing industry, it is necessary for people to perform the parallelization of the intelligent system.

In this article, we implement a parallel optimization platform for MOSFETs model parameter extraction on a Linux-based PC cluster with message passing interface (MPI) libraries. The GA implemented in the developed intelligent system with 16 PCs is parallelized with a diffusion scheme which forms a two-dimensional (2D)-grid network. When the stage of GA is performed on a processor, chromosomes are simultaneously exchanged among those results that computed by its neighboring four 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 satisfied. Our extraction experience shows that this parallel GA approach has distinguished results when the dimension of the problem is significantly large, such as parameter extraction for more than eight devices. Compared with an isolated parallel GA, more than 33% improvement in the evolution time is found in the implemented parallelization algorithm when 16 devices are optimized simultaneously. In terms of several computational benchmarks, such as speedup, efficiency, and accuracy, results for different examples with multiple MOSFETs are examined to show the robustness and efficiency of the method. Theoretical estimation and preliminary

Received August 24, 2008; Accepted November 25, 2008
Address correspondence to Yiming Li, Department of Communication Engineering, National Chiao Tung University, 1001 Ta-Hsueh Rd., Hsinchu City, Hsinchu 300, Taiwan; E-mail: ymli@faculty.nctu.edu.tw

implementation show that there is an optimal number of processors with respect to the number of devices to be extracted. For example, according to theoretical estimation, the optimal number of units is 18 for 16 semiconductor devices to be extracted, which is close to the practically obtained result (16 units), as discussed in the section of results and discussion.

This article is organized as follows. In the next section, we briefly describe our extraction system and state the architecture of parallel computing algorithms. In the section of results and discussion, we show the extraction results for single and multiple deep-submicron and sub-100 nm MOSFETs. Finally, we draw conclusions.

2. Parallel GA for model parameter extraction

In 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 could be formed as a multidimensional nonlinear optimization problem, where the number of parameters is greater than one hundred. The main goal of device model parameter extraction is now considered to minimize the error between the model extracted result and the experimentally measured data, where the extracted result is obtained through the equation of a specified equivalent circuit model as follows:

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

where the I_{DS}^{ex} is the I–V function to be optimized; the I_D is any specified device model [1–4], which contains more than 40 mathematical equations in the BSIM model [1], for example. Vectors $\vec{p}$, $\vec{v}$, and $\vec{d}$ are the parameter sets to be extracted, the bias condition for simulation, and the device geometry, respectively. To characterize a single MOSFETs' electrical characteristic, four sets of I–V curves are required, where one set of I–V curves contains five I–V curves, an I–V curve has 50 I–V points at least. When performing an equivalent circuit model parameter extraction of 16 MOSFETs, the errors of the 16,000 I–V points are thus required to be minimized with respect to the formulated optimization problem at the same time, where the number of parameters to be optimized is more than one hundred. We notice that the nonlinear optimization problem is subject to proper physical constraints. This large-scale optimization problem with massive computation is performed on our developed intelligent extraction system [1]. The developed hybrid optimization platform integrates the GA, the LM method, and the NN algorithm, as shown in Fig. 1. When the GA obtains a solution, the LM method is activated to search for the nearby local optima, and the NN algorithm suggests proper searching directions according to the current results and physical constrain. We notice before extraction that all input measured I–V data are preprocessed by statistical reduction and sampling procedures. The GA and LM method are then applied to calculate all parameters. The NN algorithm is used to trace the errors of I–V curves and their first derivatives, and also to inspect the variations of physical quantities. Once parameters are found, the postprocess will identify a sensible searching path so that the solution engine continues evolution until parameters reach specified stopping criteria. The detail of this intelligent extraction system is reported in [1], and the essential implementation details of the GA are discussed in our recent work [14].

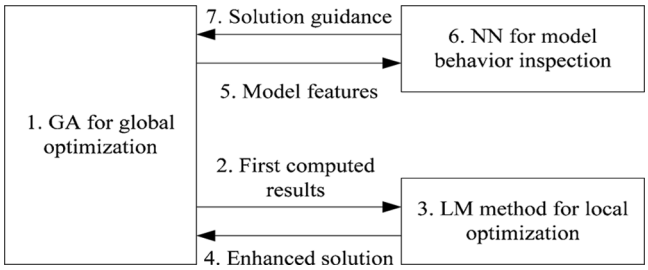

Figure 1.—A system architecture of the developed extraction system. Before extraction, all input measured I–V data are preprocessed by statistical reduction and sampling procedures. The GA and LM method are then applied to calculate all parameters. The NN algorithm is used to trace the errors of I–V curves and their first derivatives, and also to inspect the variations of physical quantities. Once parameters are found, the postprocess will identify a sensible searching path so that the solution engine continues evolution until parameters reach specified stopping criteria.

Although the intelligent extraction system has been proposed and implemented successfully, facing a larger-scale optimization problem with massive computation still requires enormous amount of CPU time, the parallel GA is thus advanced. On the other hand, we notice that the time acquired by the LM method and NN algorithm can be regarded as instant, compared with the major time cost of GA. Therefore, only the GA is parallelized in the developed intelligent system. Application of parallelization to GA provides a cost-effective way to reduce the computing time [15–19].

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 MOSFETs' model parameter extraction. The design of gene encoding strategy depends on the property of problem. In this problem model, there are more than one hundred parameters, and all variables are floating-point numbers. The fitness function measures the error between simulated result and realistic measurement data. The fitness function F used in the intelligent system is formed as

$$F = \left(\sqrt{\sum_{d}\sum_{cs}\sum_{c}\sum_{p}\left(I_{DS}^{ex} - I_{DS}^{me}\right)^2}\right), \tag{2}$$

where I_{DS}^{ex} is the extracted I–V points, I_{DS}^{me} is the measured I–V points, and d, cs, c, p refer to the number of devices, curve sets, curves, and I–V points, respectively. 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 a MOSFET device model, all parameters to be optimized can be classified into several categories and each of them stands different physical characteristics [1, 10, 12–14]. 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 sub-100 nm MOSFETs' model parameter extraction [14]. To reduce the time cost of optimization, different parallel GA schemes are taken into consideration.

The parallel extraction system is implemented on our PC-based Linux cluster with 16 units [20–22]. 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 the 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 of 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 the server if data transmission is required. This procedure loops until the fitness score reaches a specified criterion or the evolution time is up.

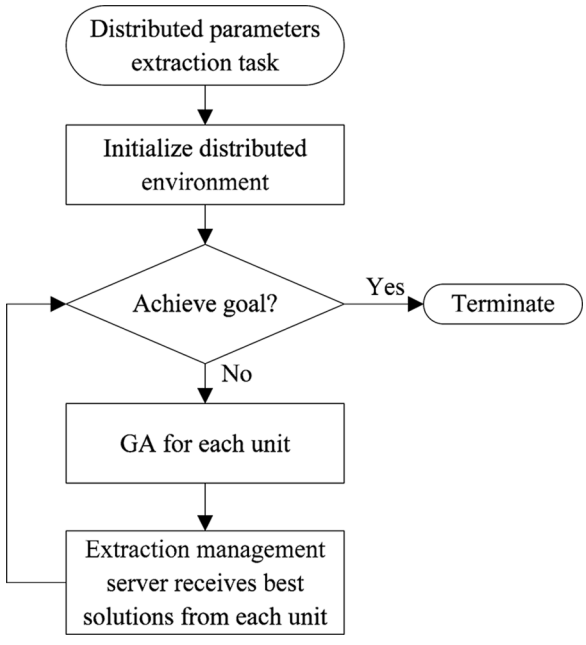

FIGURE 2.—An execution flowchart of the parallel GA that implemented in our model parameter extraction system. Once the procedure starts, the environment is initialized firstly, and each processor begins their job, and sends the current result to server if data transmission is required. This procedure loops until the fitness score is reached a specified criterion or the evolution time is up.

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 [18]. 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. In contrast 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 promote the proliferation of good genetic building blocks. The most famous migration methods of GA are the ring, the unrestricted, and the diffusion GA. Figure 3 shows the basic topologies of isolated and diffusion GA. A procedure of 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 four neighbors
        Receive results from four neighbors
        Selection
       Crossover
       Mutation
      End While
  End For
End Diffusion GA
```

In our system, each individual is assigned to a specific location, and the migration is permitted between a set of specific neighbors. In the advanced MOSFETs' model parameter extraction, parameters according to their engineering meanings can be classified into several groups, and each group represent specific physical phenomenon. 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.

2.2. Theoretical Estimation

Furthermore, a theoretical estimation on the optimal parallel performance of the diffusion GA is discussed for

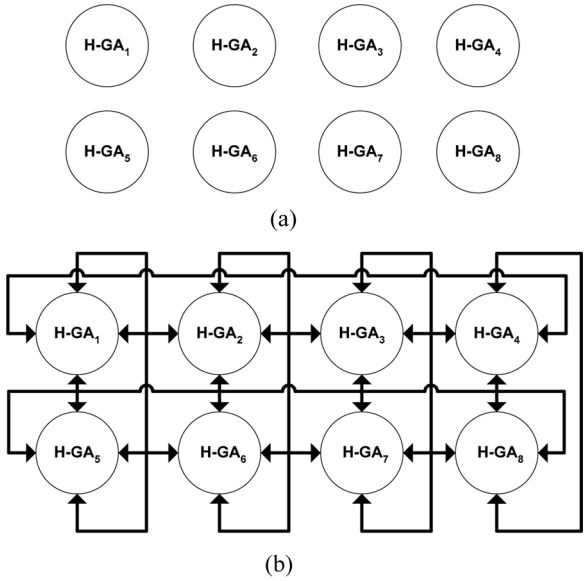

FIGURE 3.—The (a) isolated GA and (b) diffusion GA topologies for the proposed distributed hybrid-GA in this parameters extraction system.

the implemented parallel extraction system. Assume that there are p processors, the communication time cost is T_c, n denotes the population size, and the total evaluation time is T_f. In our implemented diffusion GA, we set the number of neighbors of each unit as four. Thus the entire time cost for one generation T_p is given by

$$T_p = pT_c + \frac{nT_f}{p} + 4pT_c = 5pT_c + \frac{nT_f}{p}, \tag{3}$$

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

$$p^* = \sqrt{\frac{nT_f}{5T_c}}. \tag{4}$$

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

$$S_p \equiv \frac{T_s}{T_p} = \frac{nT_f}{(nT_f/p) + 5pT_c} = \frac{nT_f/5T_c}{(nT_f/5T_cp) + p} > 1. \tag{5}$$

This ratio is the parallel speedup for the diffusion GA, and it formalizes the intuitive notion that parallel computing 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:

$$E_f = \frac{T_s}{T_p p} = \frac{S_p}{p}. \tag{6}$$

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 $\widehat{E}_f$, we let Eq. (6) equal to $\widehat{E}_f$ and solve the corresponding equation for p. The computed p_e is given by

$$p_e = \sqrt{\frac{1 - \widehat{E}_f}{\widehat{E}_f}\frac{nT_f}{5T_c}}. \tag{7}$$

We note that $p_e = p^*$ when $\widehat{E}_f$ is 0.5. The maximum speedup achievable by the diffusion GA equals half optimal number of units.

3. Results and discussion

In this section, three investigations are computationally performed. The first one demonstrates the robustness of our optimization method; the second issue shows the performance comparison between the isolated and diffusion GA. Finally, the parallelization configuration of this work is discussed. In our extraction experiment, the industrial standard BSIM4 device model is adopted [1, 2, 14]. We further perform a series of experiments to examine the accuracy and efficiency of the proposed method.

Figure 4 shows the dimension distribution (the device width versus its length) of the investigated 16 N-MOSFETs of 90 nm fabrication technology. Without loss of generality, optimized results of a N-MOSFET device among these 16 devices are shown in Fig. 5, where Figs. 5(a), (b) are the original I–V curves, and Figs. 5(c), (d) are the first derivatives of the corresponding original I–V curves.

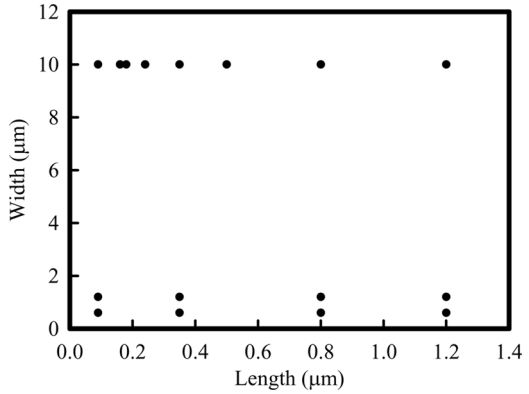

FIGURE 4.—The dimension distribution (width vs. length) of investigated 16 devices. Each symbol represents a device to be optimized, where the dimension of the smallest device is $L = 90$ nm and $W = 0.6\,\mu$m among 16 devices.

TABLE 1.—List of the root-mean-square (RMS) error of the optimized parameter compared with the measured data for 16 MOSFETs. The oxide thickness of target devices are 3.36 nm and the working temperature is settled at 298.15 K.

Device Geometry (μm/μm)	Errors between the optimization and measurement for different I–V curves			
	$I_D - V_D$	$I_D - V_G$	$I_D - V'_D$	$I_D - V'_G$
L/W (0.09/0.6)	2.81%	2.41%	5.95%	5.79%
L/W (0.35/0.6)	2.24%	2.34%	5.67%	5.12%
L/W (0.80/0.6)	1.38%	1.93%	3.34%	2.35%
L/W (1.2/0.6)	1.34%	0.98%	3.38%	1.84%
L/W (0.09/1.2)	2.99%	2.74%	4.75%	5.41%
L/W (0.35/1.2)	2.18%	2.36%	3.58%	3.92%
L/W (0.80/1.2)	1.25%	2.19%	2.68%	4.08%
L/W (1.2/1.2)	1.07%	0.92%	1.08%	1.44%
L/W (0.09/10.0)	2.32%	2.45%	3.62%	3.56%
L/W (0.16/10.0)	2.21%	2.51%	2.41%	3.89%
L/W (0.18/10.0)	1.98%	2.05%	2.83%	2.45%
L/W (0.24/10.0)	1.79%	2.21%	2.59%	3.21%
L/W (0.35/10.0)	2.84%	2.63%	5.42%	5.84%
L/W (0.50/10.0)	2.65%	2.84%	5.03%	5.98%
L/W (0.80/10.0)	2.89%	2.37%	5.25%	5.79%
L/W (1.2/10.0)	2.59%	2.31%	5.82%	4.94%

Comparison between the measurement data (the dotted lines) and the simulation results (the solid lines) with the extracted parameters significantly demonstrates good accuracy in the device model parameter extraction with the proposed optimization method. The error of extraction result for the explored 16 devices is summarized in Table 1. As shown in this table, the root-mean-square (RMS) error of curves is strictly within 3% and 6% for all original curves and the first derivative of all original curves, respectively. We notice that the first derivative of all original I–V curves is defined by

$$(I_D - V_D)' = \frac{\partial I_D}{\partial V_D} \tag{8}$$

and

$$(I_D - V_G)' = \frac{\partial I_D}{\partial V_G}. \tag{9}$$

Figure 5 and Table 1 confirm the accuracy of the proposed method with respect to different numbers of extracted N-MOSFET devices. Figure 6 correspondingly illustrates the comparison of the convergence behavior of two different optimization approaches in BSIM4 model extraction experiments. Our hybrid approach shows better convergence behavior than a pure GA method.

Figure 7 shows a comparison of the amount of evolution time with respect to the number of extracted devices between the isolated and diffusion GA. As shown 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 four devices, the superiority of the diffusion GA is observed gradually. When the number of the target devices to be optimized is increased to 16, the 33% speedup

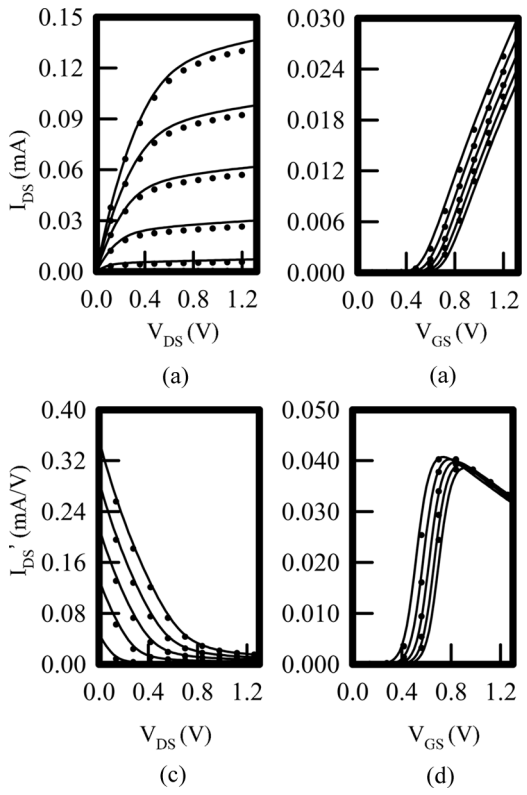

FIGURE 5.—The extracted (solid line) and measured (dot line) for the 350 nm MOSFET with the BSIM4 SPICE model, where the device width is 1.2 μm. Plot (a) is the result of $I_{DS} - V_{DS}$, where gate bias (V_{GS}) varies from 0.4 V (the lowest curve) to 1.4 V with step = 0.2 V and bulk bias (V_{BS}) = 0 V and (b) is the result of $I_{DS} - V_{GS}$, where V_{BS} varies from 0 (the left curve) to −1.2 V with step = 0.3 V. Plot (c) is the derivatives of $I_{DS} - V_{DS}$ and (d) is the derivatives of $I_{DS} - V_{GS}$ curves.

of the evolution time of the diffusion GA is achieved, compared with the speedup of the isolated one. However, for a problem with small search domain, such as only one or two devices to be optimized, the difference between two parallel methods is insignificant. With this experiment, we

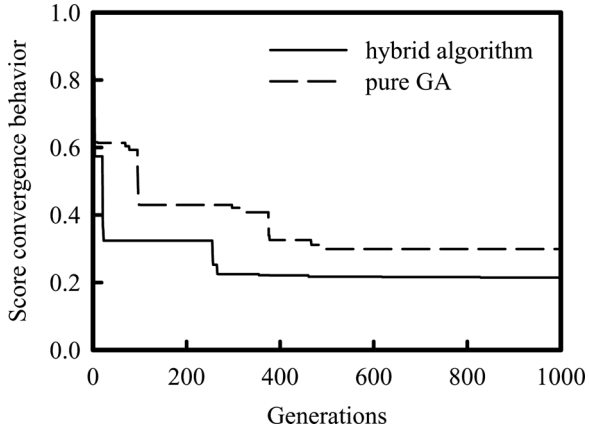

FIGURE 6.—The score convergence behavior of hybrid algorithm and pure GA approach in BSIM4 model extraction experiments.

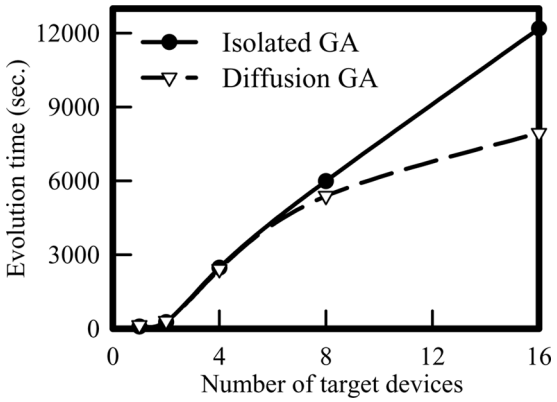

FIGURE 7.—A comparison of the time cost vs. the number of target devices for extracting multiple devices with the BSIM4 model (more than 100 parameters have to be optimized with respect to 16,000 I–V points) by using the 16 extraction units with the isolated and the diffusion GAs. The root-mean-square (RMS) error is set to be 75%, 25%, and 7% for the leakage current, linear, and saturation regions, respectively.

suggest that the diffusion GA is one of suitable distributed methods in parallelization of the explored problem. As shown in Fig. 8, 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 MOSFETs is strictly set to be within 3% error for all original curves

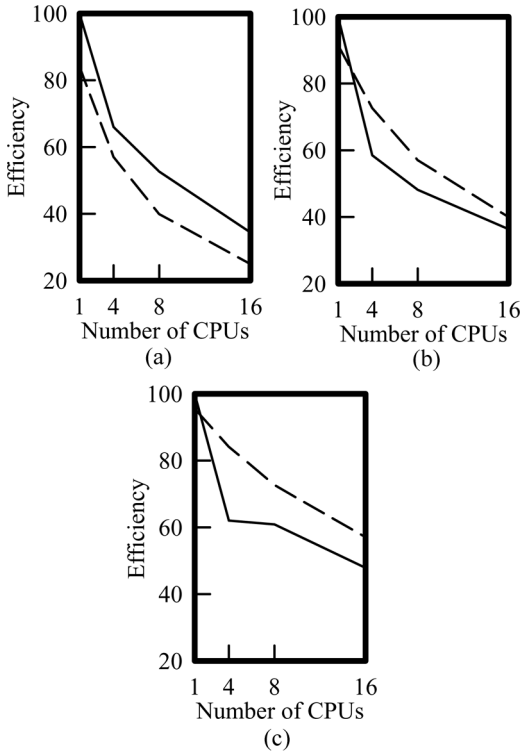

FIGURE 8.—Efficiency comparison of the experiment. The dash lines are the theoretical predictions, and the solid lines are the experimental results for (a) $T_f = 17$ msec, (b) $T_f = 34$ msec, and (c) $T_f = 68$ msec, respectively.

TABLE 2.—Performance comparison of the parallelization with respect to four, eight, and 16 devices using the diffusion GA approach.

Units	Time (sec.)	Speed up	Efficiency
		Four devices	
1	34581	1	100%
4	13098	2.64	66.00%
8	8214	4.21	**52.62%**
16	6276	5.51	34.43%
		Eight devices	
1	90984	1	100%
4	39048	2.33	58.47%
8	23963	3.84	**48.12%**
16	15648	5.81	36.34%
		Sixteen devices	
1	260772	1	100%
4	105150	2.48	62.00%
8	53546	4.87	60.87%
16	34043	7.66	**47.87%**

and 6% error for the first derivative of all original curves. The dash lines are the theoretical predictions and the solid lines are the experimental results for (a) $T_f = 17$ msec, (b) $T_f = 34$ msec, and (c) $T_f = 68$ msec, respectively. In our experiment, the communication time cost Tc 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

$$p^* = \sqrt{\frac{nT_f}{5T_c}} = \sqrt{\frac{800 \times 0.068}{5 \times 32 \times 10^{-3}}} \cong 18.44. \tag{10}$$

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, shown in Fig. 8, confirm the theoretical estimation.

Table 2 shows the benchmark results and confirms that the speedup is increased as the number of units of processors 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 [17, 18] corresponding to our estimation in Eq. (7). We concluded that the most suitable number of processors and acceptable execution time should be eight processors for extracting four and eight devices with the BSIM4 device model and 16 processors for extracting a set of optimal parameter of the explored 16 devices. Detail data of achieved speedup and efficiency are listed in Table 2.

4. CONCLUSIONS

In this article, parallelization of the GA for semiconductor device equivalent circuit model parameter extraction has been developed. The GA implemented in the intelligent 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 one, and the superiority of the diffusion GA is significant when the number of devices to be optimized is increased. Implementation 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 extraction of realistic multiple VLSI devices. The practical implementation of parallel GA approach benefits the engineering of device model parameter extraction. To validate the developed parallel intelligent model parameter extraction prototype for sub-65 nm semiconductor devices and beyond, more advanced device models, such as surface potential device models [3, 4] are currently under implemented in this system. In addition, we perform the extraction on a 32-units PC-based Linux cluster for much higher performance computation.

Acknowledgment

This work was supported in part by Taiwan National Science Council (NSC) under Contract NSC-97-2221-E-009-154-MY2 and Contract NSC-96-2221-E-009-210. The author expresses his appreciation to the referee for an exceptional in-depth reading of the manuscript.

References

1. Li, Y.; Cho, Y.-Y. Intelligent BSIM4 model parameter extraction for sub-100 nm MOSFET era. Japanese Journal of Applied Physics **2004**, *43*, 1717–1722.
2. Lu, W. *MOSFET Models for SPICE Simulation Including BSIM3v3 and BSIM4*; John Wiley & Sons Press: New York, 2001.
3. Ytterdal, T.; Cheng, Y.; Fjeldly, Tor A. *Device Modeling for Analog and RF CMOS Circuit Design*; John Wiley & Sons Press: New York, 2003.
4. Gildenblat, G.; Li, X.; Wang, H.; Wu, W.; van Langevelde, R.; Scholten, A.J.; Smit, G.D.J.; Klaassen, D.B.M. Introduction to PSP MOSFET model. Proceedings of the 2005 Workshop on Compact Modeling; Annaheim CA, USA, May 8–12, 2005; 19–24.
5. McKenzie, T.G.; Li, Y. A drain-current model for DG PMOSFETs with fabricated 35 nm device comparison. International Journal of Computational Science and Engineering **2006**, *2* (3–4), 144–147.
6. Karlsson, P.R.; Jeppson, K.O. A direct extraction algorithm for a submicron MOS transistor model. Proceedings of the 1993 International Conference on Microelectronic Test Structures, Sitges; Spain, March 22–25, 1993; Vol. 6, 157–162.
7. Karlsson, P.R.; Jeppson, K.O. A direct method to extract effective geometries and series resistances of MOS Transistors. Proceedings of the 1994 International Conference on Microelectronic Test Structures; San Diego, CA, USA, March 22–25, 1994; Vol. 7, 184–189.
8. Kunii, H.; Kinouchi, Y. 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; Hong Kong, China, Oct. 29–Nov. 1, 1998; Vol. 6, 3108–3111.
9. Arsham, H.; Gradisar, M.; Stemberger, M.I. Linearly constrained global optimization: A general solution algorithm with applications. Applied Mathematics and Computation **2003**, *134*, 345–361.
10. Li, Y.; Cho, Y.-Y.; Wang, C.-S.; Huang, K.-Y. A genetic algorithm approach to InGaP/GaAs HBT parameters extraction and RF characterization. Japanese Journal of Applied Physics **2003**, *42*, 2371–2374.
11. Holland, J. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, Michigan, 1975.
12. Li, Y.; Yu, S.-M.; Li, Y.-L. Intelligent optical proximity correction using genetic algorithm with model-and rule-based approaches. Computational Materials Science 2008, dx.doi.org/10.1016/j.commatsci.2008.04.031.
13. Li, Y.; Yu, S.-M. A novel approach to compact model parameter extraction for excimer laser annealed complementary thin film transistors. Journal of Computational Electronics **2004**, *3* (3–4), 257–261.
14. Li, Y. An automatic parameter extraction technique for advanced CMOS device modeling using genetic algorithm. Microelectronic Engineering **2007**, *84* (2), 260–272.
15. Van Veldhuizen, D.A.; Zydallis, J.B.; Lamont, G.B. Evolutionary computing and optimization: Issues in parallelizing multiobjective evolutionary algorithms for real world applications. Proceedings of ACM Symposium on Applied Computing; Madrid, Spain, March 11–14, 2002; 595–602.
16. Nanda, P.K.; Ghose, B.; Swain, T.N. Parallel genetic algorithm based unsupervised scheme for extraction of power frequency signals in the steel industry. IEE Proceedings: Vision, Image, and Signal Processing **2002**, *149* (4), 204–210.
17. Cantú-Paz, E.; Goldberg, D.E. Efficient parallel genetic algorithms: Theory and practice. Computer Methods in Applied Mechanics and Engineering **2000**, *186*, 221–238.
18. Cantú-Paz, E. *Efficient and Accurate Parallel Genetic Algorithms*; Kluwer Academic Publishers Press: Boston, 2000.
19. Rudolph, G. Deployment scenarios of parallelized code in stochastic optimization. In *Proceedings of the Second International Conference on Bioinspired Optimization Methods and Their Applications (BIOMA 2006)*; Filipic, B., Silc, J., Eds.; Josef Stefan Institute: Ljubljana, 2006; 3–11.
20. Li, Y.; Yu, S.-M. A parallel adaptive finite volume method for nanoscale double-gate MOSFETs simulation. Journal of Computational and Applied Mathematics **2005**, *175* (1), 87–99.
21. Li, Y. A parallel monotone iterative method for the numerical solution of multidimensional semiconductor Poisson equation. Computer Physics Communications **2003**, *153* (3), 359–372.
22. Li, Y.; Sze, S.M.; Chao, T.-S. A practical implementation of parallel dynamic load balancing for adaptive computing in VLSI device simulation. Engineering with Computers **2002**, *18* (2), 124–137.