Asia and South Pacific Design Automation Conference (ASP-DAC), Yokohama, Japan, 2001: 45-49

Realization of Semiconductor Device Synthesis with the Parallel Genetic Algorithm

Zhao Li Xiao-Feng Xie Wen-Jun Zhang Zhi-Lian Yang
Institute of Microelectronics, Tsinghua University, Beijing 100084, P. R. China
mail_lz@yahoo.com {xiexf, zwj}@dns.ime.tsinghua.edu.cn yangzl@tsinghua.edu.cn

Abstract – In this presented paper, to accomplish semiconductor device synthesis for TCAD application, the Parallel Genetic Algorithm is applied as the core searching algorithm for "acceptability region" of device designables, which satisfy the designed device performance. The results of some experiments on FIBMOS are shown, which indicate the Parallel Genetic Algorithm is an efficient and fast searching algorithm to fulfil device synthesis. Some potential problems related to device synthesis are also discussed.

1. INTRODUCTION

A new and promising TCAD application called technology synthesis, or process synthesis, has been explored recently [1][2]. Technology synthesis employs a top-down approach to process design, starting with the desired results (such as performance, reliability, cost/yield, etc.) and propagating this requirement downward through the process steps. Technology synthesis can be classified into several abstraction levels [Fig.1], which separate the design problem into several stages that can be potentially solved concurrently.

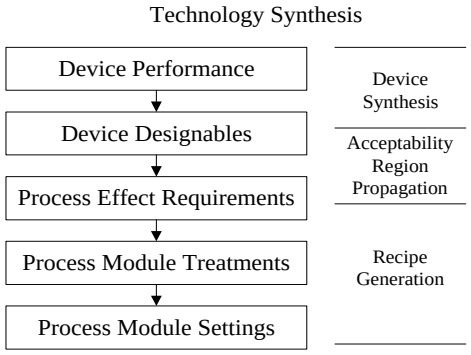

Fig.1 Five abstraction levels and the corresponding stages for Technology Synthesis

As the first step in technology synthesis, device synthesis is to determine the specification on the device designables from the requirements on the device performance. Device performance defines the electrical behavior of the device. Examples are on and off currents, threshold voltage, output resistance, etc. for MOS devices. Device designables are the parameters that specify the topography and impurity concentrations associated with a device. Typical designables are physical gate length, oxide thickness, doping profile descriptors (such as r_p and Δr_p in the Gaussian description of an implanted doping profile), etc.

Device synthesis is performed through searching over the space of device designables specified by designers [Fig.2]. With forward device models and some efficient search algorithms, the space of acceptable device designables that meet the performance specifications is determined. Instead of considering a single point solution to the device design, the region of acceptable devices in the designable space is considered, which is termed the "acceptability region".

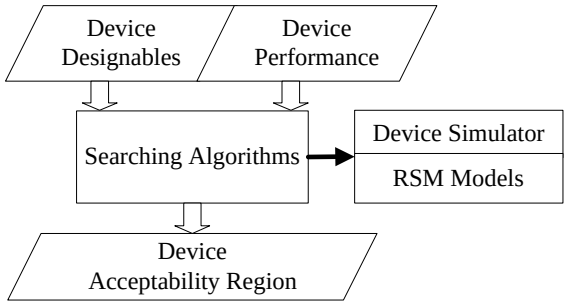

Fig.2 Flowchart of Device Synthesis

In this paper, we consider device synthesis as a problem of acquiring a solution space, so it is the most important to apply some efficient and fast searching algorithms to get multiple solution points. Thus, the core searching algorithm we employs is the Parallel Genetic Algorithm[3] (PGA) due to its efficient search and optimization ability. As an example, we have applied the PGA to synthesis the Focused-Ion-Beamed MOSFET(FIBMOS)[5][6]. The results show that the aim of device synthesis can be well satisfied. At the same time, some potential problems related to device synthesis are also discussed in this paper.

2. KEY FACTORS FOR DEVICE SYNTHESIS

1) Selection of Searching Algorithms

General searching algorithms(such as Monte Carlo methods) are short of necessary optimization directions, while general optimization algorithms(such as hill-climbing methods) are short of global searching abilities. To acquire a solution space, it is necessary to use an efficient searching algorithm, which can produce enough useful solution points to construct the solution space or its bounds while not processing too many redundant computations. Genetic Algorithms (GA)[3][4] can provide us this opportunity. GA is a kind of random algorithm which searches the space in the direction of optimizing the defined objectives without the provision of initial guesses. Further more, it is internal parallel computing, which means GA can explore many schemes through only one genetic operation. During its search process, more computations are run to reach the objective optimization, which ensures more solution points will be explored to construct a solution space or its bounds.

In our previous work[7], satisfied results have been acquired with Standard (or Sequential) Genetic Algorithms (SGA)[4] in most cases. But in some cases, more computations and time have to be taken to acquire a satisfied solution space with SGA. It is reported[3] that several competing sub-populations could be more search-effective than a wider one in which all the individuals were held together. A distributed model for GA is thus proposed, called island model after the inspiring biological observation, where the population of chromosomes is partitioned into a number of isolated sub-populations, each one of which evolves independently, trying to optimize the same function. A neighborhood structure is defined over this set of sub-populations, so that periodically each sub-population swaps its worst individuals with the best ones of its neighbors. This swapping activity is called migration of individuals. The PGA applied in this paper is based on the island model.

Experiments have shown that PGA is more efficient and faster than SGA to acquire a solution space. The fact that PGA is superior to SGA can be explained by the following two reasons : 1) During the same time interval, PGA can process more device simulations than SGA by the means of parallel device simulation, so PGA can explore a larger space than SGA; 2) According to the migration strategy, PGA is more likely to reach global optimum points while SGA is often trapped into local one.

2) Device Parameterization[2]

Numerical device simulators are commonly run based on a device description file including many command lines, which defines the grid structure, impurity distribution, electrodes, and etc. But some descriptions aren't appropriate for device synthesis because they can't be mapped directly from the description file to device structure and impurity distribution. For example, the gate length isn't an apparent parameter in PISCES description files. In our present work we build a parameterized FIBMOS[Fig. 3], which is primarily composed of two parameterized structures: a fundamental MOSFET device structure (characterized by L_{eff}, T_{ox}, X_j, N_{sd}, N_{sub}, etc) and a FIB implantation structure (characterized by FIB-X, FIB-Dose, and FIB-Energy).

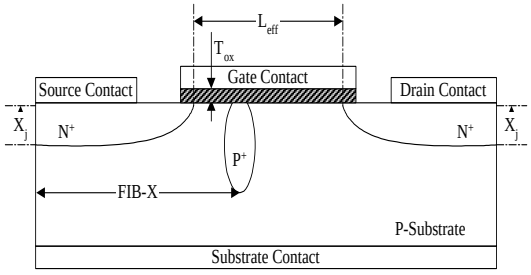

Fig.3 Device Parameterization for FIBMOS

The parameterized FIBMOS provides an easy-to-use data interface to modify device designables of FIBMOS, which is useful for device synthesis, device simulation and generation of response surface models for FIBMOS.

3) Device Simulation

A numerical device simulator costs much time to run a device simulation. Since many device simulations must be run in a device synthesis task, it is unfeasible for device synthesis to use a numerical device simulator directly. An efficient alternative is to generate response surface models(RSMs) for a specific device by design of experiments with a device simulator. With the generated RSMs, device synthesis can be run faster than ever.

3. DEVICE SYNTHESIS CASES OF FIBMOS

We applies the GA to synthesize a 0.6μm FIBMOS whose FIB-Dose is fixed at 1e14cm^{-2}. The device performance of FIBMOS includes on current (I_{on}), off current (I_{off}), and dynamic output resistance (R_{out})

[Table.1]. The device designables include lateral implantation position (x) and implantation energy [Table.2]. Here the energy is a doping profile descriptor corresponding with the vertical distance (r_p) and vertical deviation (Δr_p) of the implantation profile at the implanted point. The device simulation is done with the numerical simulator – PISCES2ET[8] in our experiments.

Table 1

Device performance of FIBMOS

Name	Objective	Unit
I_{on}	I_{on}≥1.5e-4	A/μm
I_{off}	I_{off}≤1e-12	A/μm
R_{out}	R_{out}≥1e5	Ω

Table 2

Device designables of FIBMOS

Name	Min	Max	Unit
FIB-X	0.26	0.56	μm
FIB-Energy	10	200	Kev

(Note: The channel region of FIBMOS starts from 0.26μm and stops at 0.86μm in the lateral direction)

To apply the GA efficiently, the fitness function must be provided carefully. The fitness function we designed for three objectives are listed as the following:

Fitness function for I_{on}

FF-I_{on} = 0, when I_{on}≥1.5e-4

($1.5e-4-I_{on}$)/I_{on}, when I_{on}<1.5e-4

Fitness function for R_{out}

FF-R_{out} = 0, when R_{out}≥1e5

(1e5-R_{out})/ R_{out}, when R_{out}<1e5

Fitness function for I_{off}

FF-I_{off} = 0, when I_{off}≤1e-12

(I_{off} -1e-12)/1e-12, when I_{off}>1e-12

Total fitness function (Acceptable error)

FF = FF-I_{on} + FF-R_{out} + FF-I_{off}

Each set of device designables with an acceptable error below 5% is considered as an acceptable solution point to construct the device acceptability region. It should be noted that we use a different penalty term in the lower-bound evaluation (I_{on} and R_{out}) from that in the upper-bound evaluation (I_{off}). That can ensure a faster convergence than the normal penalty term. And this kind of penalty term definition can only be used in such algorithms as GA, which can optimize functions whose first-order derivatives aren't continuous.

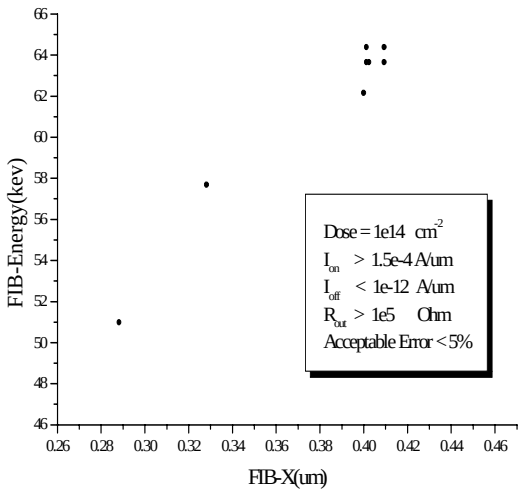

Fig.4 Results of SGA which costs 1000 device simulations in a workstation

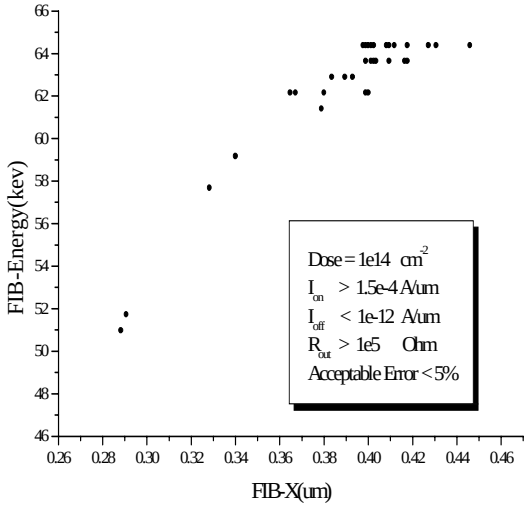

Fig.5 Results of SGA which costs 3000 device simulations in a workstation

Results with the SGA are shown as Fig.4, where 1000 device simulations are cost to finish about 18 generations. 8 results are got and the best acceptable error is 0.595%. Results with the SGA are shown as Fig.5, where 3000 device simulations are cost to finish about 55 generations. 32 results are got and the best acceptable error is 0.363%. Then a conclusion can be drawn that more device simulation time must be cost to acquire some satisfied

results with the SGA. From the following discussion, we can see that the PGA can get the same satisfied results as those of the SGA with much less simulation time.

To apply the PGA, we design the island model as Fig.6. Each island represents a UNIX workstation to run a sub-population evolution. There are totally three workstations to finish the PGA and every neighboring two islands migrate their 10% of best individuals every three generations, which is accomplished with the UNIX socket communication. One thing should be pointed out is that the initial population of each island should be generated randomly with different mechanisms.

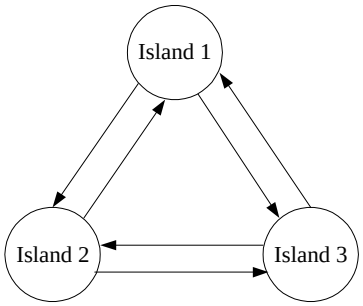

Fig.6 The island model to finish the PGA for FIBMOS

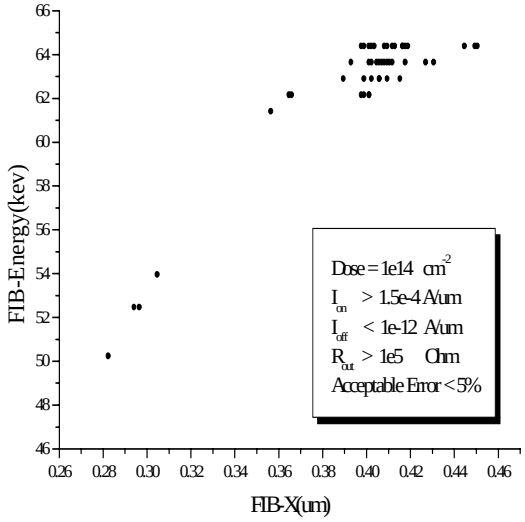

Fig.7 Results of synchronous PGA which costs 1000 device simulations in each workstation

Results with the synchronous PGA(fast workstations will wait for slow ones) are shown as Fig.7, where 1000 device simulations are cost in each workstation to finish about 18 generations. 47 results are got and the best acceptable error is 0.358%. Results with the asynchronous PGA(fast workstations will not wait for slow ones) are shown as Fig.8, where 1000 device simulations are cost in each workstation to finish about 18 generations. 34 results are got and the best acceptable error is 0.363%. The actual acceptability region is shown in Fig.9, which is got with 22800 device simulations(The precision of FIB-X is 0.005 and the precision of FIB-Energy is 0.5). 120 results are got and the best acceptable error is 0.398%. So the acceptability region acquired by the PGA matches the actual one well. It should be pointed out that pure device simulations with the assigned precision of device designbales are unfeasible, especially when the dimension of device designables is large or the actual acceptability region only occupies a very small part of the pre-assigned searching space of device designables(In this case, only 120/22800=0.526% of all simulation results are the satisfied ones.).

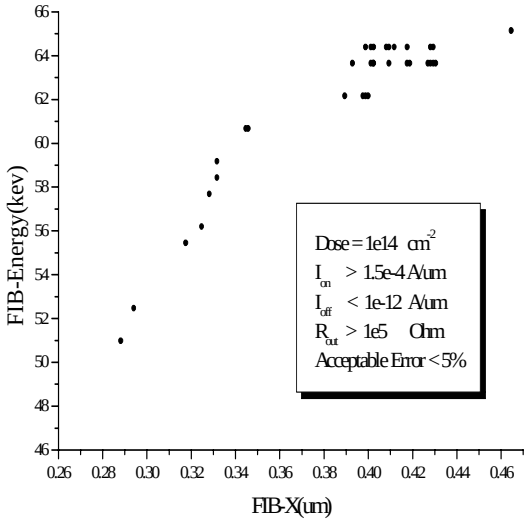

Fig.8 Results of asynchronous PGA which costs 1000 device simulations in each workstation

To compare different algorithms, we summarize above experiment results as Table.3. The conclusion can be drawn that the PGA can get more satisfied results in shorter time than the SGA. The more islands join the PGA for device synthesis, the better and the more satisfied results can be found. Especially when the dimension of device designables is large, the PGA seems to be the only efficient way to get satisfied results in reasonable time. As for the synchronous PGA and the asynchronous PGA, the latter can only be superior if more device simulations are

performed in the faster workstations than those in the slower ones.

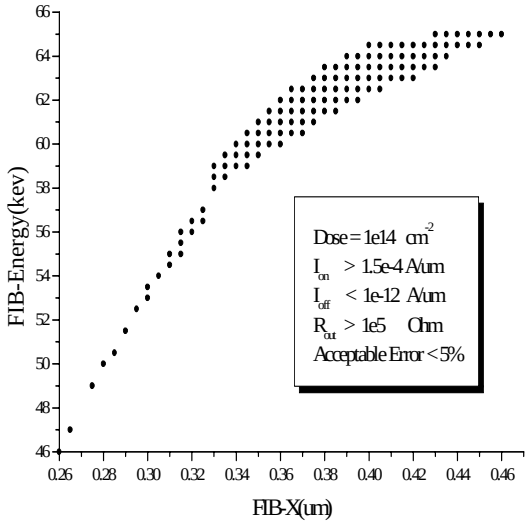

Fig.9 Results of 22800 device simulations with the assigned precision of device designables in a workstation

Table.3
Experiment results of different algorithms

Algorithm	SGA	SGA	Asyn PGA	Syn PGA
Number of device simulations in each workstation	1000	3000	1000	1000
Number of total device simulations	1000	3000	3000	3000
CPU Time (1 for 1000 serial device simulations)	1	3	1	1
Number of results	8	32	34	47
The best acceptable error(%)	0.595	0.363	0.363	0.358

4. CONCLUSION

In this paper, the Parallel Genetic Algorithm is applied to fulfil device synthesis. Device synthesis cases of FIBMOS show that the PGA is an efficient and fast searching algorithm to get the acceptability region of device designables. Based on the acceptability region, process engineers can select a set of device designables (i.e., by the sensitivity analysis of all acquired results) to meet the desired device performance and then perform next steps of technology synthesis. Further more, by analyzing the acquired acceptability region, some unknown or ignored device characteristics can be explored. So device synthesis also provides an opportunity for design engineers to study novel devices.

5. ACKNOWLEDGEMENT

This work is sponsored by MOTOROLA SPS DigitalDNA™ laboratories. We'd like to express our thanks to the experts in MOTOROLA for the helpful discussions.

REFERENCES

[1] H. H. Hosack, P. K. Mozumder, and G. P. Pollack, "Recent advances in process synthesis for semi-conductor devices", IEEE Trans. Electron Devices, vol. 45, no. 3, pp626 – 633, 1998

[2] S. Saxena et al., "An application of process synthesis methodology for first-pass fabrication success of high-performance deep-submicron CMOS", IEDM Tech. Dig., pp149-152, 1997

[3] Joachim Stender, "Parallel Genetic Algorithms: Theory & Applications", IOS Press Amsterdam Washington Tokoyo, 1993

[4] Z. Michalewicz, "Genetic algorithms + Data structures = Evolution programs", Springer-Verlag Berlin Heidelberg, 1994

[5] C.-C. Shen et al., "Use of Focused-Ion-Beam and modeling to optimize submicron MOSFET characteristics", IEEE Trans. Electron Devices, vol. 45, no. 2, pp453 - 459, 1998

[6] M. Stockinger et al., "Drive performance of an asymmetric MOSFET structure: the peak device", Microelectronics Journal, 30, pp. 229 – 233, 1999

[7] Z. Li, X. F. Xie, W. J. Zhang, Z. L. Yang, "Realization of Device Decomposition for Technology Synthesis with the Genetic Algorithm", ICDA Conference, Beijing, 2000, pp. 355-358

[8] Z. Yu, D. Chen, L. So, R. W. Dutton, "PISCES-2ET Manual", Integrated Circuits Laboratory, Stanford University, 1994