Parameter Extraction for Advanced MOSFET Model using Particle Swarm Optimization

R. A. Thakker, M. B. Patil, and K. G. Anil

Department of Electrical Engineering, IIT Bombay, India, rajesht@ee.iitb.ac.in

ABSTRACT

In this paper, parameter extraction for PSP MOSFET model is demonstrated using Particle Swarm Optimization (PSO) algorithm. *I-V* measurements are taken on 65 nm technology NMOS devices. For the purpose of comparison, parameter extraction is also carried out using Genetic Algorithm (GA). It is shown that PSO algorithm gives better agreement between measurements and model in comparison to GA and with less computational effort. A novel "memory loss (ML)" operation is introduced in the PSO algorithm for the first time, which further improves algorithm efficiency. The PSO algorithm with ML operation has taken 81 minutes on average to extract parameters of two devices of channel lengths, 70 nm and 1 μm.

Keywords: MOSFET parameter extraction, particle swarm optimization, genetic algorithm

1 INTRODUCTION

In the sub 100-nm regime, MOSFET parameter extraction has become a challenging task. Commonly used gradient based methods have many difficulties such as good initial guess requirement, singularities in objective functions, redundancy of parameters, etc. Genetic Algorithm (GA), which does not suffer from these problems, is recently reported for parameter extraction of various state-of-the-art MOSFET models [1]-[2]. Compared to GA, particle swarm optimization (PSO) algorithm [3] is reported to be more efficient for several applications and is also shown to perform better for MOSFET parameter extraction [4]. In this paper, PSO algorithm with a novel "memory loss (ML)" feature is explored for MOSFET parameter extraction. The state-of-the-art PSP MOSFET model is considered in this work. The parameter extraction is performed using PSO algorithm, PSO algorithm with ML operation, and genetic algorithm, and results are compared.

This paper is organized as follows. PSO algorithm and the memory loss operation are described in Sec. 2. Parameter extraction strategy is discussed in Sec. 3, and the results are presented in Sec. 4.

2 PSO ALGORITHM

Many variants of the PSO algorithm have appeared in the literature. In this work, we have used basic PSO algorithm with ML operation for MOSFET parameter extraction.

2.1 Basic PSO Algorithm

PSO algorithm, which mimics behavior of birds flocking in search of food, uses a cooperative approach among randomly generated "particles" to find the globally optimum solution [3]. For a problem with n variables $(x_1,x_2,...,x_n)$, a population of particles is initially generated by randomly assigning positions and velocities to each particle for each variable. In MOSFET parameter extraction, variables correspond to the model parameters to be extracted. If we denote the position and velocity of the i^{th} particle by $\mathbf{x}_i$ and $\mathbf{v}_i$ respectively, then

$$\mathbf{x}_i \equiv (x_i^1, x_i^2, ..., x_i^n), \text{and } \mathbf{v}_i \equiv (v_i^1, v_i^2, ..., v_i^n) \qquad (1)$$

Each particle in the population is a candidate for the solution, and the particles are moved towards the fittest particle (i.e., closer to the solution) in the PSO algorithm. In this process, the algorithm finds a better solution, and it is expected to reach the desired solution over time. Each particle keeps in memory the best position (denoted by $\bar{\mathbf{x}}_i$) it has attained during its trajectory. The velocity of a particle is updated on the basis of three vectors: (i) particle's own velocity, (ii) the displacement of the particle from its past best position (i.e., $\bar{\mathbf{x}}_i - \mathbf{x}_i$), (iii) the displacement of the particle from the globally best particle (i.e., $\bar{\mathbf{x}}_g - \mathbf{x}_i$). The particle moves in a direction which is a weighted addition of these three vectors. The velocity and position update of a particle at time $t + \Delta t$ is mathematically represented as follows.

$$\mathbf{v}_i(t+\Delta t) = w\,\mathbf{v}_i(t) + p_1 r_1 (\bar{\mathbf{x}}_i - \mathbf{x}_i) + p_2 r_2 (\bar{\mathbf{x}}_g - \mathbf{x}_i),$$

$$\mathbf{x}_i(t+\Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t+\Delta t)\,\Delta t, \qquad \Delta t = 1 \qquad (2)$$

Here, i is the particle index and t is the time. In actual implementation, t is generally taken to be the same as the iteration number, and therefore Δt is numerically equal to 1. The multiplicative constants w, p_1, and p_2 are parameters of the PSO algorithm, and r_1 and r_2 are random numbers uniformly distributed in the range [0, 1]. The constant w is called "inertia" of the particle, since it represents the influence of the previous velocity of the particle on its new velocity. p_1 and p_2 are acceleration coefficients. The positions $\bar{x}_i$ and $\bar{x}_g$ represent the best position attained by i^{th} particle and the globally fittest particle during their trajectory, respectively, up to time t.

The inertia parameter (w, Eq. 2), which is used to update the velocity of particles, can be either kept constant, or varied linearly or adaptively. The most commonly used linear approach to update inertia is implemented in this work and given below,

$$w(t) = (w_i - w_f)(t_{max} - t)/t_{max} + w_f \tag{3}$$

where w_i and w_f represent the initial and final values of w, respectively. t is the current iteration number and t_{max} is the maximum number of iterations. The commonly used range for w parameter is 0.9 to 0.4. The parameters p_1 and p_2 are assigned with 1.494.

2.2 Memory Loss (ML) Operation

The strength of the PSO algorithm is that each particle has a memory in the sense that it remembers the best position attained by it during its trajectory. However, in some cases, this feature makes the population vulnerable to the possibility of getting stuck in a local minimum. In the genetic algorithm, the mutation operation is generally used to bring the population out of a local minimum. In past, the mutation operator has also been applied on the particle's velocity vector in the PSO algorithm and was found to be useful. In this work, the mutation operator is used to perform a "memory loss" operation, as follows.

In each iteration, the particles are made to undergo the memory loss operation with a small probability. A random number is generated for each particle, and if it is less than certain specified probability (typically, 1%), a randomly selected component of its past best position (i.e., its memory) is replaced by a new random value. The following comments may be made about the memory loss operation.

1. It will be effective in changing the orientation of the search space in the PSO algorithm.
2. The memory loss operation is expected to continuously add diversity to the population at a small rate, thus possibly preventing the algorithm from getting stuck in a local minimum.
3. The ML operation is very easy to incorporate in the basic PSO algorithm.

In addition, it was studied by authors that the ML operation allows the inertia range to be reduced from "0.9 to 0.4" to "0.729 to 0.4" , which further makes PSO algorithm more efficient.

3 PSP MOSFET MODEL

Several advanced MOSFET models are currently in use for circuit simulation. They tend to be very complex in order to account for various short-channel effects in modern devices. These models involve generally a large number of parameters to enable accurate representation of experimental characteristics, thus making parameter extraction a challenging task. We have considered PSP MOSFET model [5], which has approximately 70 parameters to confine the behavior of one device of deep sub-micron MOSFET technology.

In the following, the procedure used for parameter extraction is described. In this context, it should be noted that it is generally not possible to extract all of the desired parameters in one optimization step because the I-V characteristics in different regions of device operation are affected by different set of parameters. For this reason, the parameters have to be extracted in several extraction steps. This step-by-step procedure will be referred to as the "parameter extraction strategy" in the following.

There are three types of parameters in PSP model.

(a) Geometry-independent (global) parameters (e.g., the oxide thickness, *tox*).

(b) Geometry-dependent (local) parameters whose values would depend on the device width W and length L (e.g. velocity saturation, *vsat*).

(c) Interpolation parameters, which can be used to compute the local parameters (in (b)) using an interpolating formula for given W and L values. For example, drain-induced barrier lowering (DIBL) local parameter (CF) in the PSP model is give by Eq. 4.

$$CF = CFL\left[\frac{L_{EN}}{L_E}\right]^{CFLEXP}\left[1 + CFW\frac{W_{EN}}{W_E}\right] \tag{4}$$

where CFL, CFLEXP, and CFW are interpolation parameters. In this work, we have restricted to parameters in (a) and (b). The extraction of the interpolation parameters (in (c)) is currently under investigation.

We have extracted 34 parameters which affect the DC current-voltage characteristics. Of these, 16 parameters are geometry-independent (i.e. type (a) above) and others are of type (b). The extraction steps used are as follows. I-V characteristics of devices with different widths (W1, W2, W3), and different lengths (L1, L2, ..., L8) are used. W1 and L1 being the smallest device, and W8 and L8 being the largest device. For a given W and L, the following steps describe the extraction procedure.

1. Doping and body bias dependent parameters are extracted from subthreshold region of $I_d - V_g$ with $V_{ds} = 0.05$, and $V_{bs} = 0.0$, -0.45, -0.9 V.

2. Mobility and series resistance parameters are extracted from $I_d - V_g$ ($V_{ds} = 0.05$ V).

3. DIBL and below-threshold channel-length modulation (CLM) parameter are extracted from subthreshold region of $I_d - V_g$ for $V_{ds} = 0.9$.

4. CLM and velocity saturation parameters are extracted from $I_d - V_d$ with $V_{gs} = 0.9$, 0.65, 0.4 and $V_{bs} = 0.0$, -0.45, -0.9 V.

5. Parameters extracted in step 3 are refined, using a narrower range for the relevant parameters.

6. Gate leakage parameters are extracted from $I_g - V_g$ ($V_{bs} = 0.0, V_{ds} = 0.0$, 0.45, 0.9 V).

We start with the device with W = W3 and L = L8, and extract parameters using steps 1-6 described above. Next, the parameters for the device with W = W3 and L = L1 (shortest channel) are extracted, keeping some of the parameters (those which do not have physical scaling, type (a)) fixed to values obtained from previous extraction step. Now, the parameters of devices with intermediate channel lengths are extracted.

4 RESULTS AND DISCUSSION

The parameter extraction for PSP MOSFET model is carried out as per the parameter extraction strategy discussed in Sec. 3. 34 parameters of PSP model are extracted from measured $I_d - V_g$, $I_d - V_d$, and $I_g - V_g$ data of NMOS devices with channel length (L) ranging from 70 nm to 1 μm and width (W) equal to 10 μm. PSP parameters are extracted using basic PSO algorithm, PSO with ML operation (PSO-ML), and GA. Population and maximum number of iterations (t_{max}) are set to 100 and 1000, respectively. Inertia (w) parameter is assigned range 0.9-0.4 in case of basic PSO and 0.729-0.4 for PSO-ML. p_1 and p_2 are equal to 1.494. For GA, crossover and mutation probabilities are set to 0.84 and 0.15, respectively.

Twenty independent runs are carried out to examine the consistency of the algorithms. The RMS error between measured and model-generated data is computed. The $I_g - V_g$ and $I_d - V_d$ characteristics of the smallest device (70 nm) are taken as benchmark to compare algorithms. The time taken by the algorithms to extract parameters of two extreme devices (70 nm and 1 m) is also noted. All runs are performed on a machine with AMD 2.2 GHz Opteron processor and 16 GB RAM. The results averaged over 20 runs are shown in Table 1.

Table 1: RMS error and CPU time averaged over 20 runs.

Characteristics	PSO	PSO-ML	GA
	RMS Error %		
$I_d - V_g$ (V_{ds} = 50 mV)	7.88	8.18	10.53
$I_d - V_g$ (V_{ds} = 0.9 V)	8.83	8.29	8.52
$I_d - V_d$ (V_{gs} = 0.4, 0.65, 0.9 V)	3.89	3.65	4.07
	CPU Time Min.		
	91	81	112

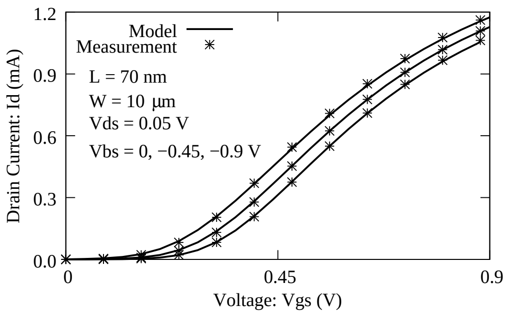

Figure 1: I_d-V_g characteristics for W/L = 10 μm/70 nm.

Both PSO algorithms give comparable results in terms of RMS error (see Table I). But, the PSO with ML operation (PSO-ML) has taken 10 minutes less. The results of parameter extraction using GA are also reported in the table. GA has taken more time and also gave larger RMS error. The comparison between experimental data and model results for the PSO-ML algorithm are shown in Figs. 1-8 and in all cases, the agreement with experimental data is found to be very good.

In conclusion, the PSO algorithm with memory loss operation is found to be very effective for MOSFET pa-

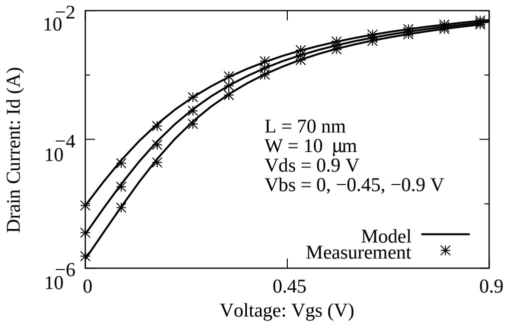

Figure 2: I_d-V_g characteristics (subthreshold) for W/L = 10 μm/70 nm.

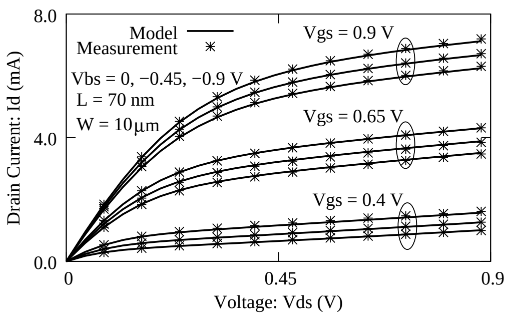

Figure 3: I_d-V_d characteristics for W/L = 10 μm/70 nm.

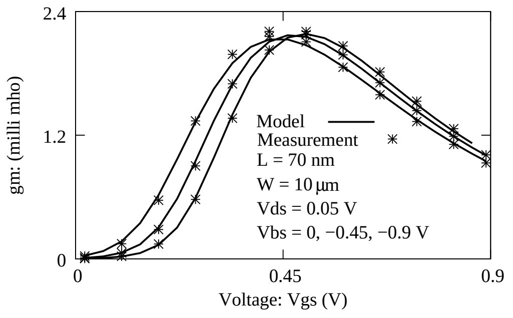

Figure 4: g_m-V_g characteristics for W/L = 10 μm/70 nm.

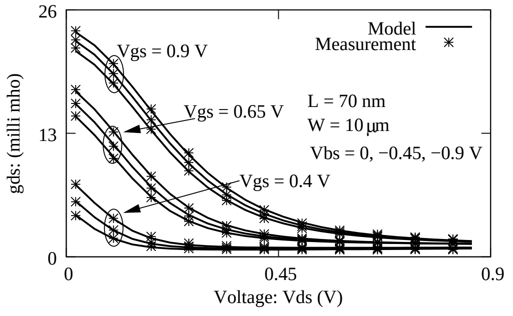

Figure 5: g_{ds}-V_d characteristics for W/L = 10 μm/70 nm.

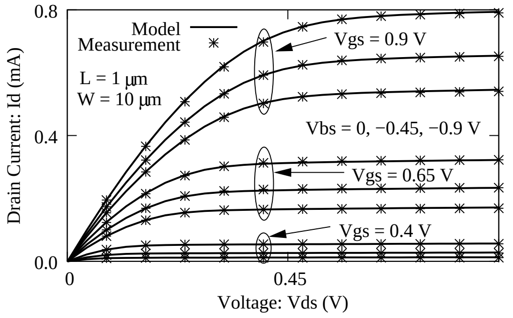

Figure 6: I_d-V_d characteristics for W/L = 10 μm/1 μm.

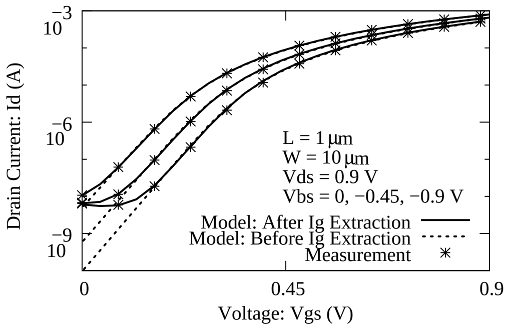

Figure 7: I_d-V_g characteristics for W/L = 10 μm/1 μm showing the effect of gate current parameter extraction.

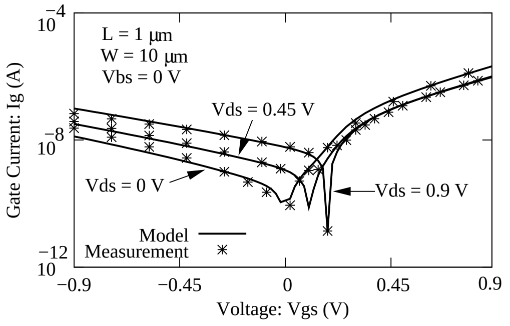

Figure 8: I_g-V_g characteristics for W/L = 10 μm/1 μm.

rameter extraction and substantially better than the genetic algorithm.

This work was performed in part at the “Centre of Excellence for Nanoelectronics (CEN), IIT Bombay” which is supported by the Ministry of Communications and Information Technology, Govt. of India.

The authors would also like to thank IMEC, Belgium for providing the devices used in this study.

REFERENCES

[1] Y. Li, *Micro. Engg.*, vol. 84, Issue 2, pp. 260-272, Feb. 2007.

[2] J. Watts, C. Bittner, D. Heaberlin, and J. Hoffman, *Proc. of Int. Conf. Mod. and Sim. of Micro.*, pp. 176-179, Apr. 1999.

[3] J. Kennedy and R. Eberhart., NJ, pp. 1942-1948, 1995

[4] R. Thakker, N. Gandhi, M. Patil, and K. Anil, *Proc. Int. Workshop Phy. of Semi. Cond. Dev.*, pp. 130-133, Dec. - 2007.

[5] G. Gildenblat et al., PSP 102.0 *User's manual.*