A hybrid MPI/OpenMP parallelization method for a quantum drift-diffusion model

Shohiro Sho and Shinji Odanaka
Computer Assisted Science Division, Cybermedia Center, Osaka University
Toyonaka, Osaka 560–0043
Email: shohiro@cas.cmc.osaka-u.ac.jp

***Abstract*—In this paper, we describe parallel domain decomposition methods based on the restricted additive Schwarz (RAS) method for a quantum drift-diffusion (QDD) model in semiconductors. We have developed a hybrid MPI/OpenMP parallelization method of the QDD system. For the inter-node parallelization, an extension of the RAS method is newly developed for the QDD model. For the intra-node parallelization, we combine a splitting-up operator method with the BiCGSTAB (SPBiCGSTAB) procedure to realize parallelization of the incomplete factorization. The parallel numerical results for three-dimensional Si bulk n-MOSFET on a multi-core parallel computer: NEC SX-ACE are demonstrated. The intra-node parallel numerical results are further evaluated on a many-core parallel computer: Cray XC40.**

I. Introduction

Numerical simulations of semiconductor devices play an important role in the analysis and design of semiconductor devices. Parallel computers having a many-core architecture are effective for the large scale device simulations and further speed up of device simulations. To make the most of such parallel computers, the development of parallel computing methods suitable for many-core architectures is an important issue. A number of authors have focused on parallelization algorithms of domain decomposition methods [1]–[4] and iterative solution methods of the linear system [5]–[7].

This paper presents a hybrid MPI/OpenMP parallelization for a quantum drift-diffusion (QDD) model in semiconductors. A restricted additive Schwarz method (RAS), which is one of the parallel domain decomposition method (DDM), has been studied for a single partial differential equation [1]. The extension of the RAS method to a system of partial differential equations is not unique. In this work, we have firstly extended the RAS method for the QDD model. For parallelization of incomplete factorization, we combine a splitting-up operator method [8] with the BiCGSTAB procedures. The performance results of three-dimensional bulk n-MOSFET on a multi-core and many-core parallel computer are demonstrated.

II. Quantum drift-diffusion model

A quantum hydrodynamic (QHD) model [9] is derived from a Chapman–Enskog expansion of the Wigner–Boltzmann equation adding a collision term. A QDD model, which is also called the density-gradient model [10], is derived from a diffusion approximation to the QHD model. The QDD model is introduced as a quantum corrected version of the classical DD model with $O(\hbar^2)$ corrections to the stress tensor. This model is viewed as one of the hierarchies of QHD models. The stationary QDD model with the unknown variables (φ, n, and u_n) is described as follows:

$$\epsilon\Delta\varphi = q(n - p - C), \tag{1}$$

$$\frac{1}{q}\mathrm{div}J_n = 0, \tag{2}$$

$$J_n = q\mu_n(\nabla(n\frac{kT}{q}) - n\nabla(\varphi + \gamma_n)), \tag{3}$$

$$b_n\nabla\cdot(\rho_n\nabla u_n) - \frac{kT}{q}\rho_n u_n = -\frac{\rho_n}{2}(\varphi - \varphi_n), \tag{4}$$

where φ, n, and p are the electrostatic potential, electron density, and hole density, respectively. φ_n is the electron quasi-Fermi-level. ϵ, q, C, and k are the the permittivity of the semiconductor, the electronic charge, the ionized impurity density, and the Boltzmann constant, respectively. T is the carrier temperature. The mobility of the electrons are denoted by μ_n. For electrons, the quantum potential γ_n is described by

$$\gamma_n = \frac{\hbar^2}{6m_n q}\frac{1}{\sqrt{n}}\frac{\partial^2}{\partial x_j^2}\sqrt{n}, \tag{5}$$

where m_n and $\hbar$ are the electron effective mass and Planck constant. From (5), the quantum potential equation is obtained as

$$2b_n\nabla^2\rho_n - \gamma_n\rho_n = 0, \tag{6}$$

where $b_n = \frac{\hbar^2}{12qm_n}$. The root-density ρ_n is written as $\rho_n = \sqrt{n} = \sqrt{n_i}\exp(u_n)$ by the variable $u_n = \frac{q}{kT}(\frac{(\varphi+\gamma_n-\varphi_n)}{2})$, where n_i is the intrinsic carrier density. As shown in [8],(6) is replaced by the equivalent form in (4). If the variable u_n is uniformly bounded, the electron density is maintained to be positive. This approach provides a numerical advantage for developing a positivity-preserving iterative solution method and high-accuracy conservative scheme [8].

III. A hybrid MPI/OpenMP parallelization algorithm

The stationary QDD model is parallelized on a parallel computer having a many-core architecture. The discretization of the QDD equations leads to the linear system of equations $Ax = b$. In the single-processor calculation, an iterative solution method for the QDD model is developed

Algorithm 1 RAS-SPBiCGSTAB

Compute $r_0 = b - Ax_0$
Choose $r^* = r_0,\ p_0 = r_0$. Compute $\rho_0 = (r^*, r_0)$.
for $i = 1, 2, ...$ **do**
  Calculate $\hat{p}$ s.t $\hat{p} = M_{RAS}^{-1}\ p_{i-1}$
  $v_i = A\hat{p}$
  $\alpha_i = \frac{\rho_{i-1}}{(r^*, v_i)}$
  $s = r_{i-1} - \alpha_i v_i$
  Calculate $\hat{s}$ s.t $\hat{s} = M_{RAS}^{-1}\ s_{i-1}$
  $t = A\hat{s}$
  $\omega_i = \frac{(t,s)}{(t,t)}$
  $x_i = x_{i-1} + \alpha_i \hat{p} + \omega_i \hat{s}$
  $r_i = s - \omega_i t$
  Check convergence; continue if necessary
  $\rho_i = (r^*, r_i)$
  $\beta_i = \frac{\rho_i}{\rho_{i-1}} \frac{\alpha_i}{\omega_i}$
  $p_i = r_i + \beta_i (p_{i-1} - \omega_i v_i)$
end for

Fig. 1. RAS-SPBiCGSTAB method.

using the Gummel's decoupled method. Each QDD equation is solved by Krylov subspace methods such as a preconditioned BiCGSTAB method.

A. Intra node parallelization

For the intra-node parallelization, we combine a splitting-up operator method [5] with the BiCGSTAB (SPBiCGSTAB) procedure to realize parallelization of the incomplete factorization. The intra-node parallelization is performed by using the OpenMP library. In the splitting-up operator method, the incomplete factorization of coefficient matrices A arising from the QDD model over a rectangular grid in three dimensions is defined as follows:

$$A \approx C = (D + A_x)D^{-1}(D + A_y)D^{-1}(D + A_z), \qquad (7)$$

where D is a diagonal matrix. A_x, A_y, and A_z are the off-diagonal matrices corresponding to the partial differences in the x-, y-, and z-directions, respectively. The incomplete factorization of (7) is one of the splitting-up operator method, which is called as the incomplete HV (IHV) decomposition [6]. In [7], this method is also called as the TF method. The parallel efficiency of the TF method is shown on a vector supercomputer. The IHV decomposition is applicable to the many-core supercomputer as a preconditioner algorithm suitable to the intra-node parallelization with shared memory. The solution of $Cz = r$ is easily calculated by solving block tridiagonal systems in the x-, y-, and z-directions:

$$(D + A_x) \cdot z_i = r, \qquad (8)$$
$$(D + A_y) \cdot z_j = D \cdot z_i, \qquad (9)$$
$$(D + A_z) \cdot z_k = D \cdot z_j. \qquad (10)$$

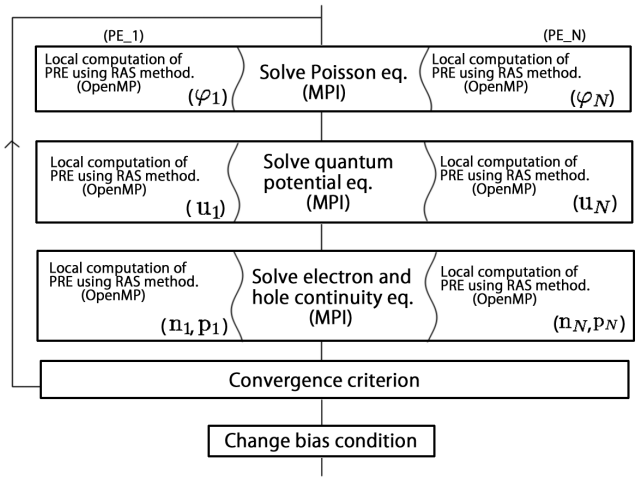

Fig. 2. The Gummel map with the RAS-SPBiCGSTAB method.

The splitting-up operator method allows parallel computation according to the natural ordering, which is realized by one-dimensional processing in the x-, y- and z-directions.

B. Inter node parallelization

For the inter-node parallelization, we apply a parallel DDM based on an overlapping Schwarz method [1]. The inter-node parallelization is performed by using the Message Passing Interface (MPI) library. We decompose the global solution domain Ω into a set of N overlapping subdomains $\left\{\Omega_i^\delta\right\}_{i=1}^N$, where δ is the number of overlaps. In this work, we have extended the restricted additive Schwarz (RAS) method [1] to the QDD model, which is described by a system of partial differential equations. The extension of the RAS method to a system of partial differential equations is not unique. In this work, we apply the RAS method as a preconditioner in the SPBiCGSTAB method shown in Fig. 1. The RAS preconditioner is defined as follow:

$$M_{RAS}^{-1} = \sum_{j=1}^{N} \tilde{R}_j^T C_j^{-1} R_j, \qquad (11)$$

where R_j is the rectangular restriction matrices from Ω to Ω_i^δ. The subdomain matrices on Ω_i^δ is defined by $C_j = R_j C R_j^T$. $\tilde{R}_j^T$ is the prolongation operator from Ω_i^0, corresponding to a non-overlapping decomposition to Ω. The algorithm contains four operations: 1. Preconditioning operations (PRE), 2. Matrix-vector products (MV), 3. Inner dot products (DOT), and 4. Vector addition and subtraction (DAXPY). In the preconditioning operations $M_{RAS}^{-1} p_{i-1}$, firstly we solve N number of local linear equations using IHV decomposition

$$C_j q_{j,i-1} = R_j p_{i-1}, j = 1, 2, \cdots, N \qquad (12)$$

to calculate local vectors $q_{j,i-1}$. Each local linear equation is allocated to each node. This procedure can be solved simultaneously. Note that we can activate intra-node parallelization

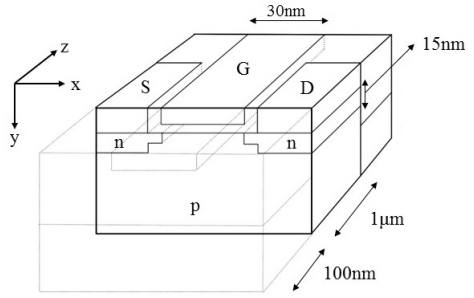

Fig. 3. Schematic of a simulated three-dimensional 30 nm Si bulk n-MOSFET.

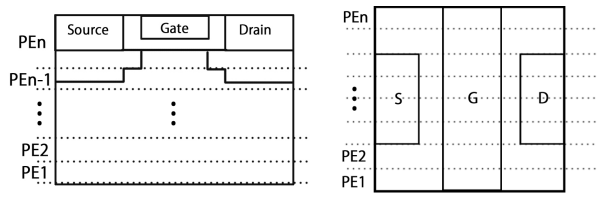

Fig. 4. One-dimensional decompositions for a three-dimensional bulk n-MOSFET: (a) y-direction and (b) z-direction decomposition.

for solving local linear equations (12). Then using local vectors $q_{j,i-1}$ to calculate global vector $\hat{p}$ by

$$\hat{p} = \sum_{j=1}^{N} \tilde{R}_j^T q_{j,i-1}. \tag{13}$$

This procedure can be realized using MPI_Isend and MPI_Irecv to exchange data in the overlapping region with adjacent subdomains. The operations MV and DAXPY can be parallelized relatively easily. The operations DOT can also be parallelized using MPI_Allreduce. This algorithm can straightforwardly be applied to the preconditioned CG method. Figure 2 shows a gummel map using this algorithm. Each QDD equation is solved using the RAS-SPBiCGSTAB method.

IV. Simulation results

A schematic of the simulated three-dimensional Si bulk n-MOSFET with high-k/metal gates for the gate length of $L_G = 30$nm is shown in Fig. 3. The number of grids used for simulations is 3,520,000 (88×200×200). The relative dielectric permittivity of gate oxide considered here is 22, and the value is known as HfO_2. The equivalent oxide thickness (EOT) is 0.8 nm. The doping concentrations of source/drain and channel are set to $N_{SD} = 1.0\times10^{20}\mathrm{cm}^{-3}$ and $1.0\times10^{18}\mathrm{cm}^{-3}$ for the Si bulk n-MOSFET. Fig. 4 shows one-dimensional decompositions in each direction. We decompose the global domain so that each subdomain contains almost the same number of grids. For the simulation, we used two grids as overlaps. The parallel computation results are obtained on a multi-core parallel computer: NEC SX-ACE. Each node of the NEC SX-ACE has four vector cores having 256 GFLOPS and a high memory bandwidth of 64GB/s per core. The intra-node parallel results are further obtained on a many-core parallel computer: Cray XC40. Each node of the Cray XC40 has 68 cores (Intel Xeon Phi KNL) having 3.05 TFLOPS. The simulation results are carried out at the bias step $V_g = 0.4$V, $V_d = 0.0\mathrm{V} \rightarrow 0.2\mathrm{V}$, which means the bias step that the gate voltage of $V_g = 0.4$V is already applied and the drain voltage of $V_d = 0.2$V is going to apply.

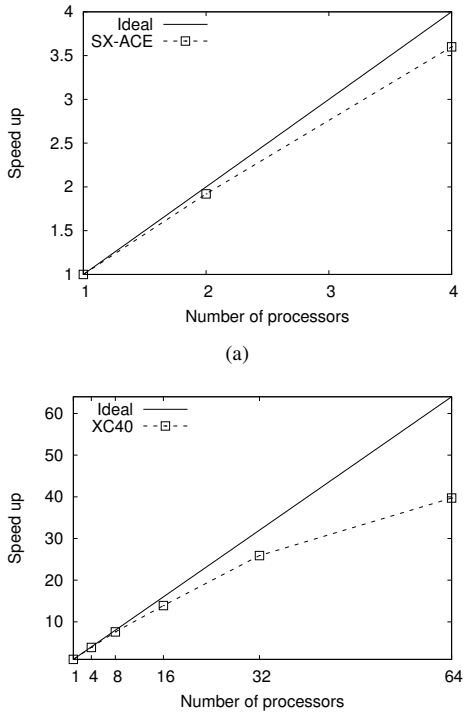

Fig. 5. The intra-node parallel speed up of the SPBiCGSTAB method. (a) NEC SX-ACE. (b) Cray XC40.

In Fig. 5, the intra-node parallel speed up of the SP-BiCGSTAB method on the NEC SX-ACE and Cray XC40 is shown. The intra-node parallel speed up is written as

$$S_{Intra}(p) = \frac{\text{CPU time using 1 core}}{\text{CPU time using p cores}}. \tag{14}$$

The parallel speed up in the intra-node parallelization strongly depends on the parallelization of the incomplete factorization by the splitting-up operator method. In the NEC SX-ACE, the parallel speed up increases almost linearly. In the Cray XC-40, we obtain a parallel speed up of 40 for 64 cores.

Figure 6 shows the inter-node parallel speed up with our hybrid MPI/OpenMP parallelization on NEC SX-ACE. The "hybrid" means an intra-node with 4 cores and inter-nodes with various N, where $N = N_y \times N_z$. The N_y and N_z are the number of y- and z-direction decomposition. The inter-node parallel speed up is written as

$$S_{Inter}(n) = \frac{\text{CPU time using 1 node (4 cores)}}{\text{CPU time using n nodes } (4 \times \text{n cores})}. \tag{15}$$

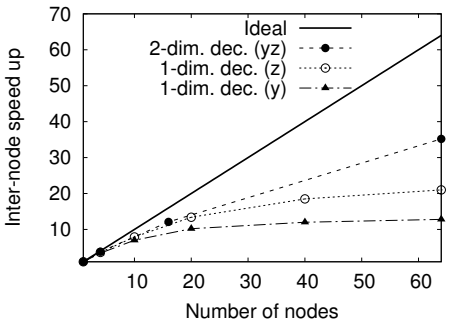

Fig. 6. The inter-node parallel speed up of the RAS-SPBiCGSTAB method in the one- and two-dimensional decompositions. The parallel speed up of 4 (2×2), 16 (4×4), 64 (8×8) nodes are plotted in the two-dimensional decomposition.

Fig. 6 shows the inter-node parallel speed up in the y- and z-direction decompositions on the NEC SX-ACE. The parallel speed up monotonically increases for both the y- and z-direction decompositions. The parallel speed up of the proposed algorithm depends on the number of RAS-SPBiCGSTAB iterations. Table I summarizes the number of RAS-SPBiCGSTAB iterations of the Poisson equation in the first Gummel loop at the bias point $V_g = 0.4$V, $V_d = 0.0$V $\rightarrow 0.2$V for the different numbers of decompositions. In the y-direction decomposition, the number of RAS-SPBiCGSTAB iterations increases as the number of decompositions increases because of the nonhomogeneity of each subdomain. In the z-direction decomposition, the number of RAS-SPBiCGSTAB iterations is almost the same as that without the DDM. This results in a smaller parallel speed up in the y-direction decomposition compared to that of the z-direction decomposition. The inter-node parallel speed up in the two-dimensional decomposition is further shown in Fig. 6. For 16 or more decompositions, the parallel speed up of the two-dimensional decomposition is larger than that of the one-dimensional decomposition. This is because the number of grids of each subdomain in the two-dimensional decomposition is smaller than that in the one-dimensional decomposition. In the case of 16 decompositions, since we use the two overlaps ($\delta = 2$) for the simulations, each subdomain has 299,200 (88×200×17) grids in one-dimensional decomposition and 256,608 (88×54×54) grids in two-dimensional (4×4) decomposition. This results in the larger speed up of the two-dimensional decomposition. A parallel speed up of 35.2 is obtained for 64 decompositions.

V. Conclusion

In this paper, we have developed parallel domain decomposition methods for a quantum drift-diffusion model in semiconductors using a hybrid MPI/OpenMP parallelization method. For the intra-node parallelization, the parallelization of incomplete factorization has been realized by the splitting-up operator method according to the natural ordering. Significant improvements in parallel performance can be achieved by the parallelization of the incomplete factorization by the splitting-up operator method. For the inter-node parallelization, we have extended the restricted additive Schwarz method to Krylov subspace methods as a preconditioner. The parallel speed up of the proposed algorithm depends on the number of RAS-SPBiCGSTAB iterations. For a one-dimensional decomposition, the number of RAS-SPBiCGSTAB iterations increases as the number of decompositions increases in y-direction decomposition, whereas that is almost the same in the z-direction decomposition. This results in a smaller parallel speed up in the y-direction decomposition compared to that of the z-direction decomposition. For a two-dimensional decomposition, a parallel speed up of 35.2 for 64 nodes is obtained.

TABLE I
The number of RAS-SPBiCGSTAB iterations.

	RAS-SPBiCGSTAB iter.	
Number of dec.	y-direction	z-direction
Without DDM	184	
4	185	192
10	200	192
20	244	189

Acknowledgment

This work was supported in part by MEXT as a social and scientific priority issue (Creation of new functional devices and high-performance materials to support next-generation industries ; CDMSI) to be tackled by using a post-K computer.

References

[1] F. N. V. Dolean, P. Jolivet, *An introduction to domain decomposition methods*. SIAM, 2015.
[2] Y. Li, "A parallel monotone iterative method for the numerical solution of multi-dimensional semiconductor poisson equation," *Computer Physics Communications*, vol. 153, pp. 359–372, 2002.
[3] A. J. Garcia-Loureiro, J. M. Lopez-Gonzalez, and T. F. Pena, "A parallel 3d semiconductor device simulator for gradual heterojunction bipolar transistors," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 16, pp. 53–66, 2003.
[4] N. Seoane and A. J. Garcia-Loureiro, "Study of parallel numerical methods for semiconductor device simulation," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 19, pp. 15–32, 2006.
[5] S. Odanaka and T. Nogi, "Massively parallel computation using a splitting-up operator method for three-dimensional device simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 824–832, 1995.
[6] T. Nogi, "Incomplete ad decomposition," *Transactions of the Research Institute for Mathematics and Science*, vol. 585, pp. 240–258 (in Japanese), 1986.
[7] S. Doi and N. Harada, "A preconditioning algorithm for solving non-symmetric linear systems suitable for supercomputers," *Proceedings of the International Conference on Supercomputing*, vol. 2, pp. 503–509, 1987.
[8] S. Odanaka, "Multidimensional discretization of the stationary quantum drift diffusion model for ultrasmall mosfet structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, pp. 837–842, 2004.
[9] C. L. Gardner, "The quantum hydrodynamic model for semiconductor devices," *SIAM Journal of Applied Mathematics*, vol. 54, pp. 409–427, 1994.
[10] M. G. Ancona, "Density-gradient theory: a macroscopic approach to quantum confinement and tunneling in semiconductor devices," *Journal of Computational Electronics*, vol. 10, pp. 65–97, 2011.