A Multi-level parallel simulation approach to electron transport in nano-scale transistors

Mathieu Luisier Purdue University - Main Campus
Purdue University - Main Campus

Gerhard Klimeck
Purdue University - Main Campus, gekco@purdue.edu

Follow this and additional works at: http://docs.lib.purdue.edu/nanodocs

http://docs.lib.purdue.edu/nanodocs/148
A multi-level parallel simulation approach to electron transport in nano-scale transistors

Abstract

Physics-based simulation of electron transport in nanoelectronic devices requires the solution of thousands of highly complex equations to obtain the output characteristics of one single input voltage. The only way to obtain a complete set of bias points within a reasonable amount of time is the recourse to supercomputers offering several hundreds to thousands of cores. To profit from the rapidly increasing availability of such machines we have developed a state-of-the-art quantum mechanical transport simulator dedicated to nanodevices and working with four levels of parallelism. Using these four levels we demonstrate that an almost ideal scaling of the walltime up to 32768 processors with a parallel efficiency of 86% is reached in the simulation of realistically extended and gated field-effect transistors. Obtaining the current characteristics of these devices is reduced to some hundreds of seconds instead of days on a small cluster or months on a single CPU.

I. Introduction

The aggressive semiconductor device scaling has led to transistors containing less than ten thousand atoms in the active region. This concerns for example the silicon-on-insulator ultra-thin-body (UTB) and multi-gate nanowire (NW) field-effect transistors (FETs) currently investigated in research labs and universities[1], [2], [3]. According to the International Road Map for Semiconductors [4] these devices will run into production in the middle of next decade. The real challenge is to make these devices operate not “in spite of” but “because of” the quantum effects inherent to the nano world.

Technology computer aided design (TCAD) can accompany this effort if classical concepts such as drift-diffusion models are replaced by quantum-mechanical approaches capturing the entire bandstructure of a crystal, the atomic granularity of the simulation domain, and the non-equilibrium condition of the problem. The semi-empirical tight-binding method fulfills these requirements and therefore has become more and more popular among the TCAD community[5]. In tight-binding each atom as well as the connections between them are taken into account and characterized by a matrix whose size reflects the complexity of the bandstructure model. We opted for the nearest-neighbor \( sp^3d^5s^* \) tight-binding model where ten (without) or twenty (with spin-orbit coupling) orbitals per atom are retained[6]. The model properly represents the behavior of the constituent baseline materials for state-of-the-art semiconductors such as Si, Ge, GaAs, InAs, AlAs with respect to all critical bandgaps, masses, and stress deformations.
Since a realistic two-dimensional device contains at least five thousand atoms the $sp^3d^5s^*$ tight-binding model leads to sparse linear systems of fifty thousand of unknowns or more that have to be solved for each injection momentum (typically 10 to 20) and energy (from 500 hundred to 2000 points). This procedure is repeated for each input voltage and eventually for several different transistor structures. Assuming that 10 seconds are needed to solve one momentum-energy combination, 14 hours is the lowest limit to treat one bias point, without even considering the self-consistency of the electrostatic potential and carrier density. Until recently it was not clear whether a multi-dimensional and quantum-mechanical simulation of nanodevices was possible without neglecting some important characteristics like the full bandstructure (effective mass approximation) or the unity of the simulation domain (mode space approximation)[7], [8].

Our quantum transport simulator, OMEN, addresses these issues. It solves two- and three-dimensional Schrödinger equations in the $sp^3d^5s^*$ atomistic basis[9]. Special attention has been paid to the calculation of the open-boundary conditions (OBCs) which may severely limit the size of the simulation domain if it is not optimized. The usual iterative solutions[10] and generalized eigenvalue problems[11] have been abandoned in favor of a shift-and-invert procedure[12]. The Non-equilibrium Green’s Function (NEGF)[13] and the Wave Function (WF)[14] formalisms have been implemented to solve the transport problem and to obtain density-of-states, transmission coefficients, carrier, and current densities. To get insight into the ballistic properties of a device where no incoherent scattering is considered the WF formalism is more suited than NEGF since it requires less operations[15].

The large supercomputer resources that are nowadays available have led us to incorporate four levels of parallelism into OMEN and to specifically design it for distributed memory machines. The highest level consists in distributing the input voltages among different processor sub-groups. This is an embarrassingly parallel workload since each bias point is independent from the others. The next two levels deal with the injection momentum and energy. Each configuration of these two quantities results in a large sparse linear system whose solution has to be integrated once all the points are calculated. Inter-processor communication is only required at the final summation stage. The fourth level of parallelization concerns spatial domain decomposition. Depending on the dimensions of the device, it is not always possible to solve the sparse linear systems on a single core. In this case the simulation domain is distributed on many processors which have to exchange information frequently.

The parallelization of momentum, energy, and spatial domain decomposition alone scales almost perfectly to 4096 cores. For a reasonable number of 50-200 bias points in transistor performance curves (output $I_d - V_{ds}$ and transfer $I_d - V_{gs}$ characteristics) OMEN would scale to at least 200’000 cores in quad-level parallelism. Here, we demonstrate the capability of the quad-level parallelism on Ranger the first NSF Track 2 system[16], [17] up to 32768 processors. This unique performance proves that supercomputers will definitively build the core of next generation device simulators.

The paper is organized as follows. In Section II the physical models that govern OMEN are reviewed and the principal equations to solve are highlighted. The implementation of the simulator is described in Section III and its computational performance is presented in Section IV. Finally, the paper is summarized in Section V and an overview of further improvement is given.
II. PHYSICAL MODELS

OMEN is a three-dimensional (3D), full-band, and atomistic simulator designed for multi-gate devices as illustrated in Fig. 1. For convenience the transport direction is assumed to be aligned with the $x$ axis, $y$ is a direction of confinement, and the confinement or periodicity of $z$ depends on the device type. For nanowires $z$ is a second direction of confinement. In 2D ultra-thin-body transistors $z$ is periodic so that the value of any function $F(x, y, z)$ representing the internal states of the device obeys the following rule

$$F(x, y, z + \Delta) = \exp(ik_z \Delta) \cdot F(x, y, z),$$

where $k_z$ is a parameter valued between $-\pi$ and $\pi$. The induced $k_z$-dependence of the internal variables is integrated to obtain carrier and current densities. Without periodicity in $z$, $k_z$ is equal to 0, the simulation domain is closed along this direction, and no structural code modification is needed.

To solve the Schrödinger equation in the WF or NEGF formalism, electrons with a momentum $k_z$ and an energy $E$ are injected and collected at the source and drain contacts of the device. There is no injection from the gate contacts. The resulting open system can be written as

$$(E - H_{ii}(k_z)) \cdot C_i(k_z, E) - H_{ii+1}(k_z) \cdot C_{i+1}(k_z, E) - H_{i-1}(k_z) \cdot C_{i-1}(k_z, E) = 0.$$  \hspace{1cm} (2)

In Eq. (2) the Schrödinger equation is expanded in the nearest-neighbor sp$^3$d$^5$s$^*$ tight-binding orbital basis so that the vector $C_i(k_z, E)$ contains all the coefficient $c_i^\sigma(y, z; k_z, E)$ associated with an orbital $\sigma \in \{s, p_x, p_y, p_z, s^*, d_{xy}, d_{yz}, d_{xz}\}$.
\(d_{x^2-y^2},d_{3z^2-r^2}\) belonging to an atom situated at the position \((y, z)\) inside a slab \(i\). Here, a slab denotes the smallest group of atomic planes (atoms with the same \(x\) coordinate) so that the block tri-diagonal form of Eq. (2) is ensured. The atomic topology in each slab and the number of layers is determined by the crystal orientation of the transport direction. For example when \(x\) is aligned with the \(<100>, <110>, \) or \(<111>\) crystal axis, a slab contains one single atomic plane, but for \(x=<112>\) a slab is composed of four atomic planes. If there are \(n_a\) atoms in the slab \(i\) the size of the vector \(C_i(E, k_z)\) is \(n_a \times t_b\), where \(t_b\) is the number of orbitals present in the bandstructure model[12].

The sparse matrices \(E - H_{ii}\) represent the on-site energy and the bond connections for all the atoms in the slab \(i\), while the \(H_{i\pm1}\) represent the coupling to the nearest-neighbor slabs. Equation (2) has to be solved for each slab index \(i\). The matrices \(H_{10}\) and \(H_{NN+1}\) couple the device to its semi-infinite contacts represented by the self-energies \(\Sigma_{11}\) and \(\Sigma_{NN}\) that are added to \(H_{11}\) and \(H_{NN}\), respectively. Different approaches exist to calculate the open-boundary conditions (OBCs). We will only mention and briefly describe three of them

- **iterative solution[10]:** an equation of type \(\Sigma = f(\Sigma)\) is iteratively solved until convergence is achieved. Full matrices have to be inverted.
- **generalized eigenvalue problem (GEVP)[11]:** given a momentum \(k_z\) and an energy \(E\) the momentum of the injected states \(k_x\) is calculated using a generalized eigenvalue problem \(AX = \lambda BX\) based on the Bloch condition that reigns in the semi-infinite contacts.
- **shift-and-invert procedure[12]:** starting from the same postulate as for the GEVP the matrix \(A\) is shifted and inverted \((A - B)^{-1}BX = 1/(\lambda - \lambda)X\). Due to its particular structure, the inversion of \((A - B)\) is straightforward and the eigenvalues of only a small block of \((A - B)^{-1}B\) need to be computed.

The shift-and-invert approach is about ten times faster than the two other methods[15] and is therefore always used in OMEN for the calculation of the open-boundary conditions in UTB and NW FETs.

After the computation of \(\Sigma_{11}\) and \(\Sigma_{NN}\), a matrix containing Eq. (2) for each slab \(i\) is assembled and solved either in the Wave Function formalism

\[
(E - H(k_z) - \Sigma(k_z, E)) \cdot C(k_z, E) = \text{Inj}(k_z, E),
\]

where the multiple right-hand-side vector \(\text{Inj}(k_z, E)\) includes the \(N_L\) (\(N_R\)) states injected from the left (right) contact, or in the NEGF formalism

\[
(E - H(k_z) - \Sigma(k_z, E)) \cdot G^R(k_z, E) = I_0,
\]

where \(I_0\) is the identity matrix[12]. If there are \(N_A\) atoms in the nanowire, the Hamiltonian \(H\) is a block-tridiagonal matrix (with sparse blocks) of size \(N_A t_b \times N_A t_b\) as the boundary self-energy \(\Sigma\). The first and last diagonal blocks of this matrix are full and contain \(\Sigma_{11}\) and \(\Sigma_{NN}\), respectively. The matrix \(\text{Inj}(k_z, E)\) is of size \(N_A t_b \times (N_L + N_R)\).

Because the symmetry properties of the boundary self-energy \(\Sigma\) and of the Hamiltonian \(H\) are different the linear system of equations above is complex, non-symmetric, and non-hermitian. OMEN is able to solve both Eq. (3) and
(4), but for ballistic transport (no incoherent scattering) the Wave Function approach is much more efficient and is preferred to NEGF[9, 15].

Equation (3) is evaluated for all the $(k_z, E)$ combinations and the resulting expansion coefficients $c_i^\alpha(y, z; k_z, E)$ are used to determine the electron $n(\mathbf{r})$ and hole $p(\mathbf{r})$ density as well as the current $I(\mathbf{r})$. The momentum and energy dependence are integrated and do no appear explicitly in the description of $n(\mathbf{r})$ and $I(\mathbf{r})$

\[
  n(\mathbf{r}) = \int dk_z \int dE \ f_n(\mathbf{r}; k_z, E) \\
  p(\mathbf{r}) = \int dk_z \int dE \ f_p(\mathbf{r}; k_z, E) \\
  I(\mathbf{r}) = \int dk_z \int dE \ f_I(\mathbf{r}; k_z, E).
\] (5)

The definitions of the functions $f_n(\mathbf{r}; k_z, E)$, $f_p(\mathbf{r}; k_z, E)$, and $f_I(\mathbf{r}; k_z, E)$ can be found in Ref. [9].

Finally, the charge density is self-consistently coupled to the electrostatic potential of the device. For that purpose the 2D and 3D Poisson equations are solved on a finite element grid and have the following form

\[
A \cdot V = S,
\] (6)

where $A$ is the stiffness matrix of the domain, $V$ the electrostatic potential at each node position, and $S$ a source term[18]. The potential $V$ is not directly calculated from Eq. (6), but by applying a Newton scheme to take care of the fact that $S$ depends on $V$[19]. Hence, the Jacobian $J$ of Eq. (6) is constructed and $V$ is iterated according to

\[
V_{i+1} = V_i + \delta V_{i+1},
\] (7)

\[
J \cdot \delta V_{i+1} = -(A \cdot V_i - S_i).
\] (8)

The size of the linear system in Eq. (8) is much larger than that of Eq. (3) and (4) because (1) additional discretization points are interleaved between the atoms and (2) the oxide layers are not taken into account in the transport calculation, but in the Poisson equation. The flow chart of OMEN for the computation of one bias point is summarized in Fig. 2.

### III. SIMULATOR IMPLEMENTATION

The quantum-mechanical simulator OMEN is written in C++ and is specifically designed for distributed memory computers using MPI[20] as the communication protocol. All algebra operations involving vectors and full matrices are performed with highly-optimized Level 1, 2, and 3 BLAS functions like (d,z)axpy, (d,z)gemm, or (d,z)gemv[21]. In its task flow depicted in Fig. 2 four loops can be identified, over the bias points (dashed blue), over the self-consistent calculation of the electrostatic potential (solid orange), over the momentum (dashed-dotted green), and over the energy (red with circles). The core of the simulator is the computation of the open-boundary conditions as described in Section II and the solution of Eq. (3) that are common to all the loops. In a typical simulation where the transfer characteristics of a device is requested, for a given drain-to-source voltage $V_{ds}$ the gate-to-source voltage $V_{gs}$ is ramped from 0.0 V to the power supply voltage $V_{dd}$ in 10 to 20 steps. For each $(V_{ds}, V_{gs})$ the calculation of
a self-consistent electrostatic potential takes 3 to 10 iterations. The momentum and energy dependence require 10 to 20 and 500 to 2000 points, respectively. Considering the lowest limit case the boundary conditions and Eq. (3) have to be solved 150,000 times (10×10×500×3). A process time of 10 seconds per iteration corresponds to a total simulation time of 17 days on a single computer.

Obviously, to reduce this time the loops over the bias points, the momentum, and the energy can be parallelized. In effect the calculation at \( V_{g1} \) does not depend on the calculation at \( V_{g2} \). Similarly, the solution of Eq. (3) for \( k_{z1} \) and \( E_1 \) is independent from the solution for \( k_{z2} \) and \( E_2 \), but the results have to be stored to evaluate Eq. (5). The loop over the self-consistent calculation cannot be parallelized since the current electrostatic potential is obtained from the previous iteration and no simultaneous distribution of the tasks is possible. As an additional fourth parallelization level the simulation domain can be spatially decomposed over different processors. The resulting simulator capabilities are shown in Fig. 3. It is a direct translation of the flow chart in Fig. 2 into a multi-level parallel hierarchy.

If a device simulation is launched on \( P_0 \) cores the distribution of the tasks happens as follows

- each of the \( N_{V_g} \) bias points defined by the user is treated independently and has \( P_{V_g} \) (user input) cores that work on it so that \( N_{V_g} \times P_{V_g} = P_0 \). During the initialization phase groups of communicators are built using
the “MPI_Group_incl” and “MPI_Comm_create” commands. Two groups $G_n$ and $G_m$ will never exchange any information during the simulation time and run in an asynchronous way (embarrassingly parallel workload). Note that the number of bias points per group is not limited to a single one.

- the $P_{V_g}$ cores dedicated to one or several bias points are subdivided into $N_{k_z}$ groups of $P_{k_z}$ cores with $N_{k_z} \times P_{k_z} = P_{V_g}$. The number of $k_z$ points is selected by the user to capture all the features of the contacts [12]. Currently, the number of processors per $k_z$ point $P_{k_z}$ is allocated by the user and not dynamically. This will be improved in the future to avoid that all the groups have to wait till the last one has completed its task. Typically, the amount of work per group has a standard deviation of 15% from its mean value.

Each member of a group of $P_{k_z}$ processors shares the same electrostatic potential, carrier density, and source, gate, and drain voltages. The integration over $k_z$ in Eq. (5) is synchronous and accomplished once the $N_{k_z}$ groups have finished their job by calling the “MPI_Allreduce” function with “MPI_SUM” as argument. Three-dimensional structures, like nanowires, need a specific treatment. The absence of periodic boundary conditions along the $z$-axis is accounted by setting $N_{k_z}=1$ and $k_z=0$. Hence, the maximum number of parallelization levels in OMEN is reduced to three for 3D nanowires FETs.

- each group of $P_{k_z}$ cores has to compute $N_E$ energy points. However, the energy grid is an adaptively refined mesh whose number of elements $N_E$ depends on the value of $k_z$. Each group member in this parallel level has a common phase $\exp(i k_z \Delta)$ and solves $n_e=N_E/P_{k_z}$ energy points if the spatial domain decomposition is switched off and $n_e=N_E/P_{k_z} \times P_E$ points if it is on. The parameter $P_E$ is the number of cores that simultaneously work on the same boundary conditions and on the same Eq. (3).

A straight-forward way to distribute the work load is to assign the $n_e$ first energy points to the first processor, the $n_e$ next ones to the second processor, and so on. This is an inefficient distribution as illustrated in Fig. 4. In effect the number of propagating states that are injected into the device from the left $N_L$ and the right $N_R$ contacts varies with energy. The time to factorize the matrix in Eq. (3) remains the same, but because the number of right-hand-sides $N_L + N_R$ increases as function of the energy the time to solve the linear system also increases. Therefore, the group dealing with the lowest energies is completed much before the group dedicated to the highest energies. For that reason the distribution mechanism has been improved to make the $m^{th}$ processor deal with the energy points $E_m$, $E_{m+n_e}$, $E_{m+2n_e}$, and so on. Due to the continuity of the transmission and density-of-states the number of right-hand-sides in the energy interval $E_m$ to $E_{m+n_e}$ is very similar and the work load is leveled better. Each processor in the group does a local energy integration of Eq. (5) over the $n_e$ points it computes.

- if the size of the matrix $(\mathbf{E} - \mathbf{H}(k_z) - \Sigma(k_z, E))$ in Eq. (3) or Eq. (4) is too large, if the memory per processor is too low, or if for a given $P_{V_g}$ $P_{k_z}=P_{V_g}/N_{k_z}$ is comprised between $N_E/4$ and $N_E$ (see Section IV) it is convenient to assign $P_E$ processors to a single energy point and to apply a spatial domain decomposition to the device. Then, the advantage of Eq. (3) over Eq. (4) becomes clear: there are several parallel direct sparse linear packages that can handle Eq. (3) like PARDISO[22] (shared memory), SuperLU_dist 2.0[23],
MUMPS 4.7.3[24], or a recently developed basis compression algorithm[25] (all distributed memory) while the NEGF problem in Eq. (4) does not lend itself to efficient parallelization[26], [27]. Extensive studies[9], [15], [25], [28] including also the sequential solver Umfpack 5.0.1[29] have shown that for ballistic transport (1) even on a single processor the solution of the NEGF problem is about 5-10 times slower than the fastest solver, (2) the basis compression algorithm is the most efficient solver in terms of performance on a single processor and parallel scalability, but only for 3D structures and under certain transport directions (z=<100> and z=<111>), and (3) in general MUMPS and Umfpack are not as sensitive as SuperLU and PARDISO to the transport direction. Other solvers like Spike[30] are currently under investigation. It is worth noting that what makes Eq. (3) computationally intensive is not really the size of the linear system, but its bandwidth. A matrix with a size \( N=2’000’000 \) and a bandwidth \( b<1000 \) is easily manageable while a matrix with \( N=500’000 \) and \( b=4000 \) as produced by 3D nanowires with a \( 5\times5 \) nm\(^2 \) cross section represents a real challenge.

Another important issue of the spatial domain decomposition is the calculation of the open-boundary conditions. As mentioned previously, two contacts are taken into account, the source and the drain and an eigenvalue problem is solved for each of them by calling the LAPACK[31] functions dgeev and zgeev. Since ScaLAPACK[32] does not provide a parallel version of these two functions (pdgeev and pzgeev) the only way to parallelize the calculation of the OBCs is to attribute the source contact to one processor and the drain to another. Only two processors can be used in the OBCs calculation, others allocated to the domain decomposition will remain idle during this phase and the parallel efficiency is reduced.

The loop over the self-consistent calculation of the electrostatic potential cannot be parallelized, but Eq. (8) can and must. The momentum, energy, and domain decomposition parallelization can reduce the time to compute charge and current densities in Eq. (5) to \( O(N^2) \) seconds. This is the order of magnitude of the Poisson equation if a sequential and direct sparse linear solver is used. To prevent OMEN from this problem Eq. (8) is solved with a parallel iterative solver, Aztec[33]. Not all the \( P_{Vg} \) processors assigned to a bias point are concerned, but a number comprised between 1 and 64 and defined by the user.

IV. Results

The performance of OMEN is analyzed for the simulation of two realistic devices run on Ranger the first NSF Track 2 Supercomputer located at the Texas Advanced Computer Center[16]. Each compute node contains four AMD Opteron Quad-Core 64-bit processors (16 cores in all, core frequency of 2.0 GHz) on a single board and 32GB of memory, i.e. approximately 2GB per core if the 16 cores per node are used. The transfer characteristics \( I_d - V_{gs} \) at a fixed drain-to-source voltage is computed. This is a very important quantity for circuit designers. According to the ITRS[4] transistors are specified among other properties by their subthreshold swing \( S \), their ON- and OFF-current \( I_{ON} \) and \( I_{OFF} \), their threshold voltage \( V_{th} \), and their channel mobility. All can be extracted from the simulated \( I_d - V_{gs} \) characteristics as illustrated in Fig. 5 for a UTB structure.

A device simulator is evaluated with respect to two criteria, the accuracy of the physical models, and the time required to obtain observable data, like the transfer characteristics. The physical models have been treated in
Fig. 4. Transmission coefficient through an UTB field-effect transistor ($V_{ds}=1.0$ V, $V_{gs}=0.4$ V) as function of the energy. A transmission $T(E)=m$ means that at least $m$ states are injected from the left and the right contacts, i.e., $N_L \geq m$ and $N_R \geq m$. A transmission equal to 0 reveals that states are injected only from one contact and no state is available in the other contact. Hence, the size of $\text{Inj}(k_z, E)$ in Eq. (3) is energy-dependent.

Section II and recognized by several publications[9], [12], [15], [34], [35], [36], [37]. The computational efficiency is discussed in this Section.

A. Gate-All-Around Nanowire Field-Effect Transistor

The first benchmark structure is a silicon gate-all-around (GAA) circular nanowire field-effect transistor as depicted in the left part of Fig. 1. Such ultrascaled devices have already been demonstrated by research groups[3]. The Si channel has a diameter of $d=3$ nm and is surrounded by oxide layers of thickness $t_{ox}=1$ nm. The source and the drain extensions measure $L_s=L_d=10$ nm, the gate $L_g=15$ nm. Since the $z$-axis is a direction of confinement no periodic boundary conditions are applied to the structure, the number of $k_z$ points $N_{k_z}$ is equal to 1, and only three levels of parallelism are possible (bias, energy, and spatial domain decomposition). In other words the number of processors per $k_z$ point is always equal to the number of processors per bias point $P_{k_z}=P_{V_g}$. Furthermore, the number of self-consistent Poisson iterations is limited to 2 in this timing experiment.

Figure 6 shows the transfer characteristics of the GAA NW transistor (inset, red curve) and the simulation time on $P_0$ cores comprised between 16 and 4096 (blue and green curves). The gate voltage is ramped from 0.0 V to 0.75 V in steps of 0.05 V and 16 bias points are simulated. The energy domain counts about 540 points. The calculation of Eq. (3) is accomplished with the basis compression algorithm[25] and requires more than 2GB of memory for high-energy points. Hence, spatial domain decomposition on $P_{E}=2$ (blue curve with squares) and $P_{E}=4$ (green curve with triangles) CPUs becomes necessary. From 16 to 256 cores $P_{V_g}$ is equal to $P_0$ (one bias point treated at the same time) then, from 512 to 4096 cores $P_{V_g}$ remains equal to 256 and $P_0/P_{V_g}$ bias points are simultaneously computed. The simulation times and relative speed-up factors measured with respect to $P_0=16$ and $P_{E}=2$ are also reported in Fig. 7.
Fig. 5. Transfer characteristics $I_d - V_{gs}$ ($V_{ds}=1.0$ V, blue line with circles) of a Si UTB FET as schematized in Fig. 1 ($a=4.9$ nm, $L_x=L_d=10$ nm, $L_y=22$ nm, $x=100$, $y=100$). Important design parameters like subthreshold swing $S$, ON-current $I_{ON}$, OFF-current $I_{OFF}$, threshold voltage $V_{th}$, or mobility upper limit $\mu_{lim}$ can be extracted from the simulation data.

Fig. 6. OMEN simulation time for the transfer characteristics $I_d - V_{gs}$ ($V_{ds}=0.6$ V, red curve with circles in inset) of a GAA NW field-effect transistor (diameter $d=3$ nm, source and drain length $L_x=L_d=10$ nm, gate length $L_y=15$ nm). Three levels of parallelism are used: bias points (16 points here), energy points ($\approx 540$ points), and spatial domain decomposition ($P_E=2$, blue curve with squares and $P_E=4$, green curve with triangles). Two self-consistent Poisson iterations are allowed per bias point.

<table>
<thead>
<tr>
<th>$P_B$</th>
<th>$P_E=2$ (s)</th>
<th>Speed-Up</th>
<th>$P_E=4$ (s)</th>
<th>Speed-Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>64373</td>
<td>1x/1x</td>
<td>85070</td>
<td>0.76x/1x</td>
</tr>
<tr>
<td>32</td>
<td>33071</td>
<td>1.9x/2x</td>
<td>43055</td>
<td>1.5x/2x</td>
</tr>
<tr>
<td>64</td>
<td>17100</td>
<td>3.8x/4x</td>
<td>21827</td>
<td>2.9x/4x</td>
</tr>
<tr>
<td>128</td>
<td>9103</td>
<td>7.1x/8x</td>
<td>11205</td>
<td>5.7x/8x</td>
</tr>
<tr>
<td>256</td>
<td>5028</td>
<td>13x/16x</td>
<td>5916</td>
<td>11x/16x</td>
</tr>
<tr>
<td>512</td>
<td>2531</td>
<td>25x/32x</td>
<td>2968</td>
<td>22x/32x</td>
</tr>
<tr>
<td>1024</td>
<td>1241</td>
<td>52x/64x</td>
<td>1470</td>
<td>44x/64x</td>
</tr>
<tr>
<td>2048</td>
<td>631</td>
<td>102x/128x</td>
<td>741</td>
<td>87x/128x</td>
</tr>
<tr>
<td>4096</td>
<td>323</td>
<td>199x/256x</td>
<td>378</td>
<td>170x/256x</td>
</tr>
</tbody>
</table>

Fig. 7. Parallel execution time (in seconds) and relative speed-up factor corresponding to the results shown in Fig. 6. The reference for the relative speed-up factor is the simulation time at $P_B=16$ and $P_E=2$. The first number in the speed-up columns is the measured speed-up, the second number the expected ideal value. The first column of the table refers to the total number of CPU used in the simulation ($P_B$), columns 2 and 3 to $P_E=2$, columns 4 and 5 to $P_E=4$. The number of CPU per bias point $P_{V_g}$ is first increased from 16 to 256 and then remains constant.
Fig. 8. Time to compute the open-boundary conditions and Eq. (3) (transport problem, left part of the table) and to solve Eq. (8) (Poisson equation, right part of the table) for the NW structures whose results are shown in Fig. 6 and 7. The basis compression algorithm[25] is employed for Eq. (3) in column 7, where the first number indicates the factorization and solve time for low-energy points and the second number for high-energy points. The variable $N_{f}$ is the size of the matrix $(\mathbf{E} - \mathbf{H} - \Sigma)$, $b$ its bandwidth, $NNZ$ the number of non-zero elements, and $\%$ the sparsity of the band. The same convention applies to the data of Eq. (8) where $N$ is the size of the Jacobian matrix $\mathbf{J}$ and $P$ the number of processors invoked by the iterative linear solver Aztec[33].

Ideally the blue and green curves in Fig. 6 would lie on top of each other since the time to compute the open-boundary conditions and to solve Eq. (3) should be two times faster with $P_E=4$ than with $P_E=2$. This is not the case as can be seen in Fig. 8, columns 6 and 7. The impossibility of parallelizing the boundary conditions on more than two cores and the non-ideal parallel efficiency of the sparse linear solvers explain this behavior. Therefore, when $P_E=4$, two processors out of four are idle during the OBCs calculation phase. Hence, on Ranger where 16 cores per node are available only half of them will work and their performance increases as shown in Fig. 8 column 6.

The scaling properties of OMEN for this realistic NW structure are very close to the ideal case. The central part of the simulator, the OBCs calculations and the solution of Eq. (3) must be performed 17280 times with an average duration of 29 seconds ($P_E=2$). This represents about 6 days on 2 cores. Note that the time to solve Poisson equation is negligible (see Fig. 8 for more details). Now the same simulation that lasts about 18 hours on 16 cores is performed in 5 minutes on 4096 cores yielding a relative speed-up factor of $199 \times (256 \times$ expected) in going from 16 to 4096 processors. To obtain a better speed-up factor less processors per point $P_{k_z}$ should be used. In effect with $P_{k_z}=256$, $P_E=2$, and $N_E \approx 540$ few cores handle five energy points while the majority deals with four energy points. If $T_0$ is the time for one energy point (assumed equal for all of them) the total time becomes $5 \times T_0$. With $P_{k_z}=128$ the total time is $9 \times T_0$. The speed-up factor in going from $P_{k_z}=128$ to $P_{k_z}=256$ is only $9/5=1.8$ instead of 2.

We note here that some devices that are subject to atomic disorder such as alloy compounds or interface roughness may require a significantly larger number of energy points ($N_E \geq 1500$). In those cases good scalability to even larger core counts can be achieved on the energy parallel level.

B. Double-Gate Ultra-Thin-Body Field-Effect Transistor

The second example is a silicon double-gate ultra-thin-body field-effect transistor designed according to the ITRS specifications for the 22 nm technology node. A schematic view of the device structure can be found on the right part of Fig. 1. The Si body has a thickness $t_{body}=4.9$ nm, the oxide layers surrounding it $t_{ox}=1.3$ nm. The length of the source and drain contacts is $L_s=L_d=10$ nm, of the gate $L_g=22$ nm. The $z$-axis of the structure is assumed periodic and is modeled by $N_{k_z}=16$ momentum points treated simultaneously. The energy grid is composed of
Fig. 9. Scaling example of OMEN for the simulation of the transfer characteristics $I_d = V_{th}$ ($V_{ds}=1.0$ V) of a UTB field-effect transistor ($t_{body}=4.9$ nm, $L_x=L_d=10$ nm, $L_y=22$ nm, $x=100$, $y=100$). The four levels of parallelism are taken into account, $N_{k_x}=16$ momentum points and $N_{k_y} \approx 1400$ energy points are computed per bias point. Three cases are investigated (1) 1 bias point with 1 Poisson iteration, $P_{k_z}$ is ramped from 1 to 256, $P_{k_y}=N_{k_x} \times P_{k_z}$ ($P_E=1$ blue curve with squares, $P_E=2$ green curve with triangles), (2) 8 bias points with 2 Poisson iterations, $P_{k_z}$ is ramped from 8 to 128 and then remains at that value, $P_{k_y}=N_{k_x} \times P_{k_z}$ ($P_E=1$ red curve with circles, $P_E=2$ cyan curve with stars), and (3) 16 bias points with 3 Poisson iterations, $16 \leq P_{k_z} \leq 128$, $P_{k_y}=N_{k_x} \times P_{k_z}$, and $P_E=2$ (violet curve with cross).

![Scaling example of OMEN](image)

Fig. 10. Parallel execution time (in seconds) and relative speed-up factor for the examples shown in Fig. 9. The relative speed-up factor refers to the simulation time obtained with $P_B=16$ and $P_E=1$ in the first test, $P_B=128$ and $P_E=1$ in the second experiment, and $P_B=256$ and $P_E=2$ in the last test. The first entry in the speed-up columns is the measured value, the second entry the expected ideal value.

![Parallel execution time](image)

<table>
<thead>
<tr>
<th>$F_0$</th>
<th>$P_E=1$ s.</th>
<th>Speed-Up</th>
<th>$P_E=2$ s.</th>
<th>Speed-Up</th>
<th>$P_E=1$ s.</th>
<th>Speed-Up</th>
<th>$P_E=2$ s.</th>
<th>Speed-Up</th>
<th>$P_E=2$ s.</th>
<th>Speed-Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>17798</td>
<td>1x/1x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>32</td>
<td>8982</td>
<td>2x/2x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>64</td>
<td>4522</td>
<td>3.9x/4x</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>128</td>
<td>2277</td>
<td>7.8x/8x</td>
<td>3010</td>
<td>5.9x/8x</td>
<td>39088</td>
<td>1x/1x</td>
<td>50324</td>
<td>0.8x/1x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>256</td>
<td>1141</td>
<td>15.6x/16x</td>
<td>1511</td>
<td>12x/16x</td>
<td>19544</td>
<td>2x/2x</td>
<td>25157</td>
<td>1.6x/2x</td>
<td>75657</td>
<td>1x/1x</td>
</tr>
<tr>
<td>512</td>
<td>583</td>
<td>30.5x/32x</td>
<td>760</td>
<td>23x/32x</td>
<td>9868</td>
<td>4x/4x</td>
<td>12672</td>
<td>3.1x/4x</td>
<td>38314</td>
<td>1.97x/2x</td>
</tr>
<tr>
<td>1024</td>
<td>291</td>
<td>61x/64x</td>
<td>385</td>
<td>46x/64x</td>
<td>5023</td>
<td>7.8x/8x</td>
<td>6383</td>
<td>6.1x/8x</td>
<td>19176</td>
<td>3.94x/4x</td>
</tr>
<tr>
<td>2048</td>
<td>150</td>
<td>119x/128x</td>
<td>198</td>
<td>90x/128x</td>
<td>2630</td>
<td>15x/16x</td>
<td>3251</td>
<td>12x/16x</td>
<td>9813</td>
<td>7.7x/8x</td>
</tr>
<tr>
<td>4096</td>
<td>85</td>
<td>210x/256x</td>
<td>100</td>
<td>178x/256x</td>
<td>1312</td>
<td>30x/32x</td>
<td>1626</td>
<td>24x/32x</td>
<td>5011</td>
<td>15x/16x</td>
</tr>
<tr>
<td>8192</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>651</td>
<td>60x/64x</td>
<td>809</td>
<td>48x/64x</td>
<td>2442</td>
<td>31x/32x</td>
</tr>
<tr>
<td>16384</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>320</td>
<td>122x/128x</td>
<td>402</td>
<td>97x/128x</td>
<td>1281</td>
<td>59x/64x</td>
</tr>
<tr>
<td>32768</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>687</td>
<td>110x/128x</td>
</tr>
</tbody>
</table>

Fig. 11. Same as Fig. 8, but for the UTB structure in Fig. 9 and 10. Eq. (3) is solved with MUMPS[24].
a maximum of about 1400 points and an average of 1160 points per momentum. Equation (3) is solved with the parallel sparse linear solver MUMPS[24] enabling spatial domain decomposition. Therefore, the four levels of parallelism of OMEN are potentially applicable.

Three different tests are run and the results are shown in Fig. 9 and summarized in Fig. 10. In the first test only one bias point is computed ($V_{gs} = 0.0$ V and $V_{gs} = 1.0$ V, three levels of parallelism) on 16 to 4096 cores with a single Poisson iteration, $P_E=1$ (blue curve with squares), and $P_E=2$ (green curve with triangles). The number of cores per $k_z$ point $P_{k_z}$ is increased from 1 to 256, $P_{Vg} = N_{k_z} \times P_{k_z}$ from 16 to 4096. Almost perfect scaling is observed on the interior three levels of parallelism with a parallel efficiency of $210/256 = 82\%$. A full sweep of $(V_{ds}, V_{gs})$ would require 50–200 independent voltage points driven by the outer voltage parallelism without loss of parallel efficiency. In the two remaining experiments we will only let the gate voltage $V_{gs}$ vary and consider 8 and 16 points up to 16384 and 32768 processors, respectively. We conclude therefore that OMEN could scale to over 200’000 cores if such a machine were available.

In the second case the gate voltage is ramped from 0.0 V to 1.0 in 8 points, all the levels of parallelism are turned on, and 128 to 16384 cores are used, once with $P_E=1$ (red curve with circles) and once with $P_E=2$ (cyan curve with stars). The parameter $P_{k_z}$ is equal to 8 for $P_0=128$, rises up to 128 for $P_0=2048$, and does not vary any more for $P_0 \geq 2048$. Finally, during the third test 16 gate voltages are computed between 0.0 V and 1.0 V, the number of processors $P_0$ increases from 256 to 32768, the simulation domain is spatially decomposed on 2 CPUs ($P_E=2$), and 3 Poisson iterations are taken into account (violet curve with cross). The resulting transfer characteristics is shown in Fig. 5 to illustrate the relevant design parameters.

The open-boundary conditions and the sparse linear system in Eq. (3) have to be solved 22’400 times for the first test, 358’400 for the second, and 1’075’200 for the last one. Each iteration takes approximately 12.5 seconds if $P_E=1$ (no spatial domain decomposition), 8 seconds if $P_E=2$, and 6 seconds if $P_E=4$ as reported in Fig. 11. On a single processor the third test would last more than five months. No one in practical device simulation can wait such a long time to obtain one curve. The fabrication and the characterization of the device would even take less time. However, on Ranger and using 32768 cores, the five months are reduced to 11 minutes. This is possible because of the quasi-ideal scaling performances of OMEN. For example in Fig. 10 one sees that the relative speed-up factor obtained in going from $P_0=256$ to $P_0=32768$ cores is equal to 110 (128 expected) which corresponds to a parallel efficiency of 86%.

C. Floating Point Operation Count

Another important measure of computing’s performance is the floating point operations per second (FLOP/s). The CRAY high performance computers offer a tool called craypat to evaluate the FLOP/s in application codes. Using the machine number 84 in the Top500 list of supercomputers[38] a rate of about 2.1 GFLOP/s on a single core is obtained to calculate the open-boundary conditions and to solve Eq. (3). The utilized processors are AMD Opteron with a core frequency of 2.6 GHz, 2 floating point operations per cycle, and a peak performance of 5.2 GFLOP/s. Hence, OMEN runs at about 40% of the machine peak performance. Assuming that our simulator keeps...
the same FLOP/s rate per core on Ranger the simulation with 32768 cores reaches a peak of 69 TFLOP/s. With the introduction of dual pipelines in the AMD Barcelona architecture the peak performance of the 2GHz cores on Ranger amounts to 8GFLOP/s per core so that we can even expect a rate higher than 69 TFLOP/s.

V. CONCLUSION AND OUTLOOK

We have presented an atomistic quantum-mechanical nanodevice simulator for 2D and 3D structures like UTB and NW field-effect transistor. An almost ideal scaling of the computational time up to 32768 cores with 86% efficiency has been achieved and has allowed the simulation of realistically extended devices in a couple of minutes instead of weeks. This performance opens now a wide range of possible applications. For instance the optimal design of a transistor can be studied by simulating various structures with different dimensions, cross sections, crystal orientations, and doping concentrations. Furthermore, random effects like interface roughness, alloy disorder, or doping position that require a large set of results to be significant can be investigated in a reasonable amount of time. This is the benefit of the four levels of parallelism.

To improve the parallelization of the open boundary-conditions calculation it is considered to compute them for all the energy points first, store them on the machine scratch, and reload them prior to solve Eq. (3). Hence, no processor remains idle in the case of spatial domain decomposition. Another issue is the efficiency of the parallel linear solvers. For the moment only direct solvers have been tested that proceed to a LU decomposition of the system of equations. As an alternative iterative solvers could be used[39]. They rely on matrix-vector multiplications that can be easily parallelized, but they are not well-adapted to multiple right-hand-side problems. Finally, an hybrid implementation of OMEN combining distributed memory and shared memory parallelization is under investigation. The relevance of such a model comes from the architecture of the modern supercomputers containing many cores per node. It is computationally very intensive to start MPI tasks on 32’000 cores. It could be more efficient to start only one MPI task per node and launch threads on the remaining cores.

VI. ACKNOWLEDGEMENT

We would like to thank the Texas Advanced Computer Center staff for helping us improving our code and for providing us a large access to Ranger, their NSF Track 2 system. This work was partially supported by NSF grant EEC-0228390 that funds the Network for Computational Nanotechnology, by NSF PetaApps grant number 0749140, and by NSF through TeraGrid resources provided by TACC.

REFERENCES


April 14, 2008 DRAFT


[38] http://www.top500.org