How far can we accelerate full-band atomistic device simulations through graphics processing units (GPUs)?

Mathieu Luisier
Integrated Systems Laboratory, ETH Zürich, 8092 Zürich, Switzerland, e-mail: mluisier@iis.ee.ethz.ch

Introduction As the size of transistors keeps shrinking, it becomes more and more important to include quantum mechanical, bandstructure, and atomistic effects to accurately simulate their properties. Hence, nano-TCAD tools that capture band non-parabolicity, atomic granularity, energy quantization, quantum confinement, and tunneling are required to design next-generation logic switches.

Advanced simulation models are computationally very intensive so they are either restricted to small devices or need some simplifications to be applied to realistic structures. For example, replacing the real-space by a mode space approach allows for the consideration of larger transistors [1]. At the algorithm level, a recursive Green's Function (RGF) solver [2-3] is faster than inverting an entire matrix. Finally, reducing the model complexity (tight-binding vs. DFT) decreases the computational burden.

Here, another approach based on novel hardware architectures is investigated to accelerate nanoelectronic device simulations: the usage of graphics processing units (GPUs). The transition from CPUs to GPUs demands for software modifications that can be significant. The purpose of this paper is to show what kind of speed-up can be obtained as function of the effort that is invested in code-rewriting.

Computational Strategy GPUs find their main application in 3D computer graphics and video cards, but can also be used as general-purpose computing units. Currently, they equip the largest supercomputer in the world called Titan and located at ORNL [4]. Usually, on each computational node, N_{GPU} GPUs are attached to N_{CPU} CPUs, as illustrated in Fig. 1(a). The total speed-up SU_{tot} that can be achieved when all the CPUs and GPUs work together, as compared to the case with CPUs only, is defined as

$$SU_{tot} = \frac{N_{GPU} \cdot SU_{GPU} + N_{CPU,working}}{N_{CPU}}, \quad (1)$$

where SU_{GPU} is the speed-up when comparing one single GPU with one single CPU and $N_{CPU,working}$ the number of CPUs that are actually working: due to the low memory of GPUs ($\leq$6 GB), $N_{CPU,working} < N_{CPU}$ so that the memory of the idle CPUs can be used to store GPU data.

To determine SU_{tot}, AMD Opteron 6272 CPUs with a frequency of 2.1 GHz and reaching a performance of P_{CPU}=16.8 GFlop/s [5] are selected. As GPUs, the Tesla K20 Kepler from NVIDIA with a peak performance of P_{GPU}=1170 GFlop/s are chosen [6]. Ideally, SU_{GPU}=P_{GPU}/P_{CPU}=70. However, even with an excellent programmer, SU_{GPU} will remain below 50 because it is difficult to fully exploit the GPU potential. Furthermore, a complete code rewrite in a GPU language is needed.

From Eq. (1) it appears that with multiple GPUs per node, the total speed-up SU_{tot} tends towards $N_{GPU} \cdot SU_{GPU}/N_{CPU}$: with 1 GPU for 4 CPUs, SU_{tot} $\approx$12.5 is possible. Many machines such as Titan at ORNL or Tödi at the Swiss National Supercomputing Centre (CSCS) [7] provide only one GPU for 16 CPUs. If two CPUs remain idle ($N_{CPU,working}$=N_{CPU}-2=14) for data storage, the maximum achievable speed-up SU_{tot}=4, obtained only after lots of code modifications. An attractive alternative when $N_{GPU} << N_{CPU}$ consists in off-loading only the code segments with heavy computations to the GPU. This hybrid approach requires less work and is tested here.

Results A full-band, atomistic quantum transport (QT) simulator based on the tight-binding model and the Non-equilibrium Green's Functions (NEGF) [8] is accelerated with GPUs. The algorithm that the QT solver uses to solve the NEGF equations [9] is slightly modified (200 lines of code) to off-load matrix operations to the GPUs.

Three applications are considered: a Si gate-all-around nanowire transistor, a non-flat graphene nanoribbon, and a Ge electron-hole bilayer tunneling transistor (EHBTFET) [10], as shown in Fig. 1(b-d). Some simulation results are depicted in Fig. 2 and 3. The time-to-solution with and without GPUs and the speed-up obtained with GPUs are reported in Fig. 4 as function of the number of CPUs (32 up to 4096) on Tödi at CSCS. An almost linear scaling of the time can be observed when the number of CPUs increases. More important, a speed up of 2 or more is obtained when GPUs are used, with a peak at 2.5 for the EHBTFET.

Conclusion In this paper, an acceleration of quantum transport simulations through GPUs has been presented. It is found that rewriting a TCAD simulator in a GPU language is only beneficial if a computer with several GPUs per node is available. Otherwise, off-loading code segments to the GPU brings useful speed-ups with much less efforts. Note that other accelerators such as the many integrated cores (MIC) from Intel are emerging with potential greater speed-up and less code modifications than GPUs.

References [1] R. Venugopal et al., J. Appl. Phys. **92**, 3730 (2002). [2] R. Lake et al., J. Appl. Phys. **81**, 7845 (1997). [3] A. Svizhenko et al., J. Appl. Phys. **91**, 2343 (2002). [4] www.olcf.ornl.gov/titan/ [5] www.amd.com/la/Documents/Opteron_6000_QRG.pdf [6] www.nvidia.com/object/tesla-servers.html [7] www.cscs.ch/ [8] M. Luisier et al., Phys. Rev. B **74** 205323 (2006). [9] T. B. boykin et al., Phys. Rev. B **77** 165318 (2008). [10] L. Lattanzio et al., IEEE Elec. Dev. Lett. **33** 167 (2012).

978-3-901578-26-7

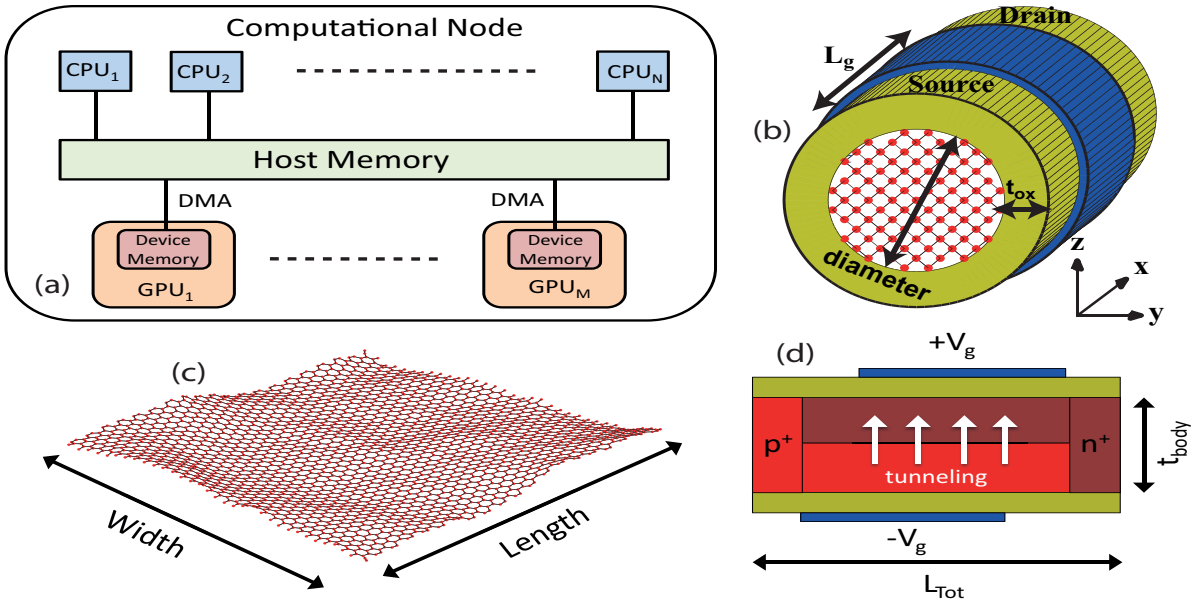

Fig. 1. (a) Schematic view of a typical computational node including N shared memory CPUs (hosts) and M GPUs (devices) connected through direct memory access (DMA). (b-d) Illustration of the nanoelectronic devices simulated in this work with either CPUs only or CPUs+GPUs. (b) Si gate-all-around nanowire field-effect transistor (GAA NW FET) with a diameter of d=4 nm, a gate length L_g=20 nm, and composed of N_A=31372 atoms. (c) Non-flat graphene nanoribbon (GNR) of width w=20 nm and length L=120 nm (N_A=97440). (d) Ge electron-hole bilayer tunneling field-effect transistor (EHBTFET) [10] made of N_A=42600 atoms with a body thickness t_{body}=10 nm and a total length L_{tot}=170 nm.

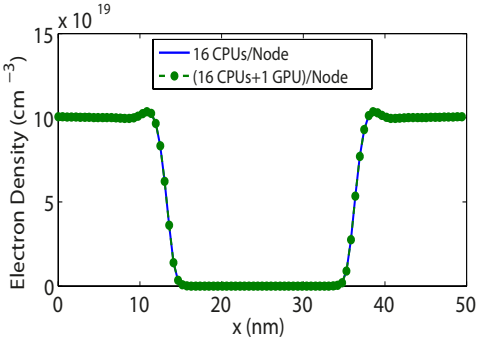

Fig. 2. Comparison of the electron density in the Si NW FET shown in Fig. 1(b) obtained with (dashed green line with circles) and without (solid blue line) the GPUs.

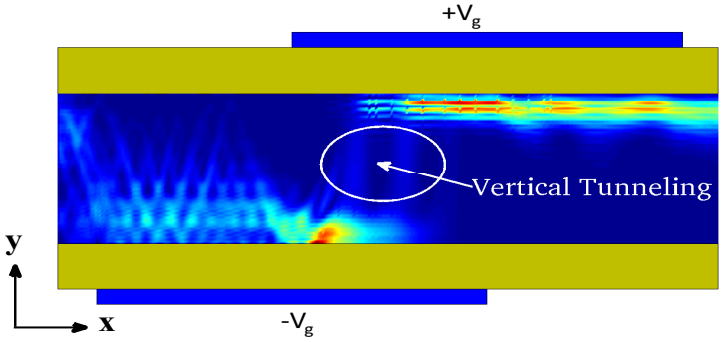

Fig. 3. Spatially-resolved ON-current flowing through the EHBTFET shown in Fig. 1(d). With GPUs, the simulation time could be reduced by a factor close to 2.5 as compared to CPUs only. Vertical tunneling paths appear clearly in the plot.

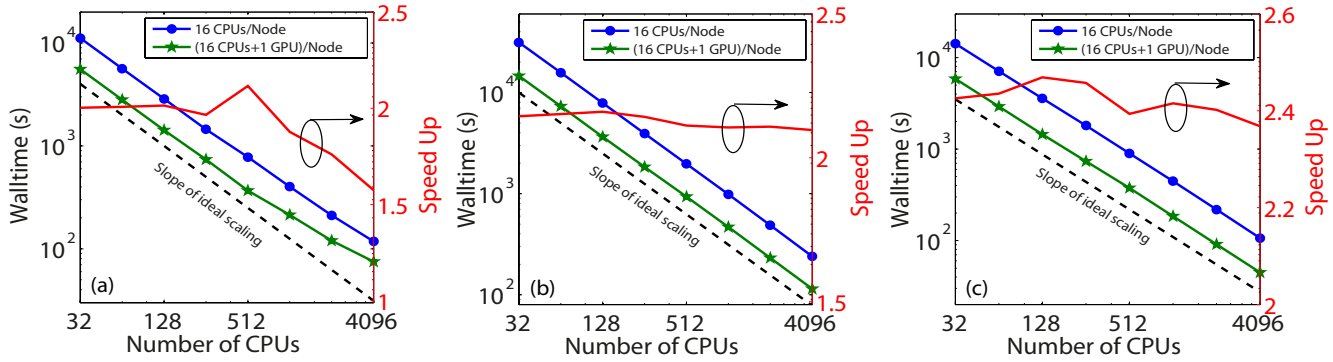

Fig. 4. Walltime vs. number of CPUs scaling curves to compute 1 Schrödinger-Poisson iteration for the (a) nanowire FET, (b) non-flat GNR, and (c) bilayer TFET shown in Fig. 1. Todi at CSCS is used: it contains 16 CPUs and 1 GPU per node. Two different types of numerical experiments are conducted: simulations with (green lines with stars) or without (blue lines with circles) the GPUs. In both cases, all the CPUs per node are utilized. The red curve indicates the speed up factor obtained when 16 CPUs and 1 GPU per node are used as compared to 16 CPUs only. Note that for the Si NW FET and Ge EHBTFET, 10 orbitals per atom are considered ($sp^3d^5s^*$ nearest-neighbor tight-binding model without spin-orbit coupling), 9 for the GNR (sp^3d^5). The largest Hamiltonian matrix amounts therefore to 876960 (GNR), then 426000 (TFET), and finally 313720 (NW).