Università degli Studi di Udine

Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica

Dottorato di Ricerca in Ingegneria Industriale e dell'Informazione

Ph.D. Thesis

Development and Optimisation of Advanced Simulation Models for Nanoscale FETs with Alternative Channel Materials

Candidate:
Patrik Osgnach

Supervisor:
Prof. Pierpaolo Palestri

Examination Committee:

Prof. Manuela De Maddis
Prof. Tomaso Erseghe
Prof. David Esseni
Dr. Kazutaka Seo
Prof. Giovanni Verzellesi
Prof. Renato Vidoni

Ciclo XXVII

Author's e-mail: patrik.osgnach@uniud.it

Author's address:

Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine
Italia

Contents

Chapter 1

Introduction

The self-fulfilled prophecy called Moore's law [1] has been driving the evolution of electron device technology over the last 50 years. This law states that the number of transistors in an integrated circuit doubles every 2 years (Fig. 1.1), a growth rate that has been possible only thanks to the improved performance at lower cost achieved by means of dimensional scaling of the planar MOSFET, which is the elementary building block of CMOS ICs [2, 3, 4]. Dimensional scaling has recently led to critical issues related to:

- static power consumption due to leakage current (gate tunnelling and sub-threshold channel current);
- short channel effects [5];
- limited or even reduced ON current when channel length is in the decananometer range;
- increased dynamic power consumption per unit of area due to the inability of lowering the supply voltage [6].

Various solutions have been proposed to mitigate these issues, which go under the general name of *technology boosters*. Some of these solutions are:

- introduction of strain in the channel region [7, 8, 9];
- replacement of the SiO_2 gate oxide with high-k materials [10, 11];
- replacement of the poly-silicon gate with a metal gate [10, 11];
- replacement of the planar bulk or SOI architecture with a 3D structure, like FinFET or gate-all-around devices [12, 13];
- replacement of the silicon channel, source or drain with alternative channel material, such as Ge, SiGe or III-V compounds as InAs, GaAs, InGaAs and GaSb [14, 15, 16, 17, 18].

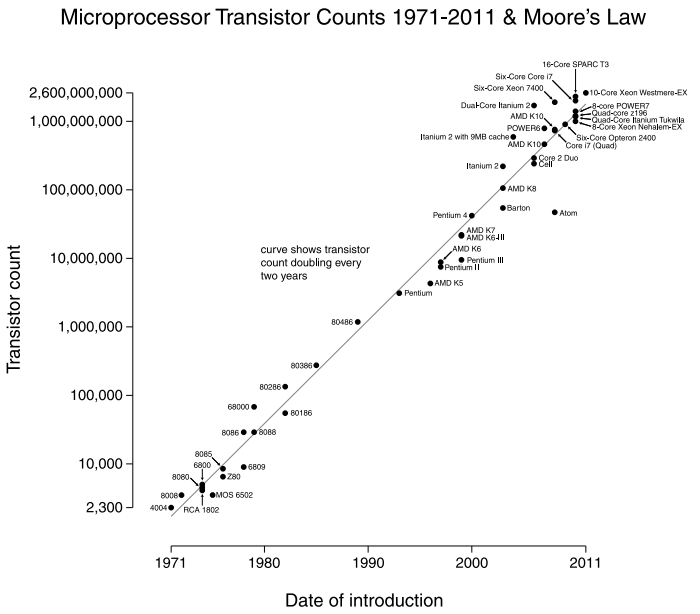


Figure 1.1: Transistor count of various CPUs with respect to the date of introduction. The count doubles approximately every two years

1.1 III-V compound semiconductor materials

III-V compounds are crystals made from elements belonging to the third and fifth group of the periodic table. The most investigated as possible replacement of silicon are InAs, GaAs, InGaAs, and GaSb. These materials have some advantages over silicon; among these: a direct band-gap, higher electron mobility, higher injection velocity[1]. Figure 1.2 reports the measured carrier velocities at the virtual source versus the gate length different III-V semiconductors. The velocity at the virtual source in III-V is more that twice the velocity in a silicon device at half the supply voltage. A supply voltage of 0.5V is lower than what the ITRS roadmap for semiconductors [19] predicts for III-V devices for the next years (0.63 V for year 2018 and 0.54 V for year 2026).

III-V materials can be very helpful when trying to reduce dynamic power consumption. In fact the dissipated power is

$$P_{dyn} = C_G \cdot n_{gate} \cdot f_{ck} \cdot V_{DD}^2 \tag{1.1}$$

where C_G is the gate capacitance, n_{gate} the average number of gate switching events in a clock period, f_{ck} the switching frequency and V_{DD} is the supply voltage. Clearly, we would like to reduce the supply voltage, but this has some negative effects. The ON current of a nanoscale MOSFET is:

$$I_{ON} = W \cdot C_G(V_{DD} - V_T)v_{VS} \tag{1.2}$$

where W is the width of the device, V_T is the threshold voltage and v_{VS} is the velocity of the carrier at the virtual source. If we reduce V_{DD} we reduce the ON current. This is bad because low ON current implies longer switching times for a load capacitance that

[1]The injection velocity is the velocity of the carriers at the virtual source, which is defined as the position of the top of the source/channel potential energy barrier.

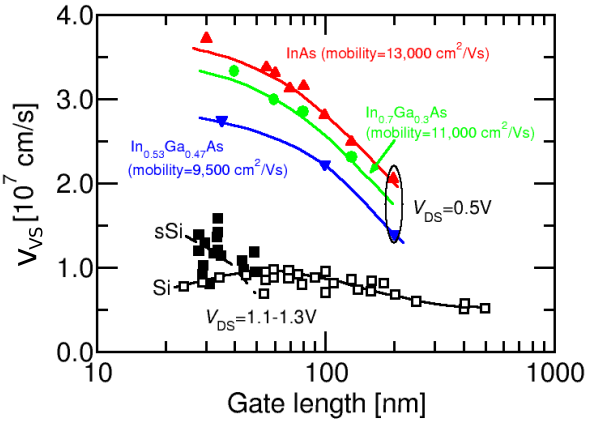


Figure 1.2: Measured carrier velocities at the virtual source of different III-V devices. Data from [20].

stays fixed or decreases less than I_{ON} itself. We can compensate this detrimental effect by reducing the threshold voltage but this causes variability issues and has an impact on the OFF current, which goes as $\exp(-eV_T/k_bT)$. The other option is having a larger v_{VS}, which is exactly what III-V materials can provide. Looking again at the ITRS roadmap [19], we can see that for year 2018 the traditional high performance devices should have an ON current of 1.61 mA/μm for a V_{DD} of 0.78V, whereas for the same year, III-V devices should have an ON current of 2.2 mA/μm for a V_{DD} of 0.68 V.

The implementation of these materials into CMOS technology however has also some disadvantages:

1. The gate stack: one big advantage of silicon is its native oxide SiO_2. Interfaces between these two materials have a low amount of defects. No such oxide exists for III-V materials. Interfaces between high-k and III-V materials have a high amount of interface states. States in the gap affect the sub-threshold slope [20] and states in the conduction band trap the free electrons and cause Fermi level pinning [20, 21]. Both effects reduce the ON current for a given OFF current. However, promising results were obtained from Al_2O_3/GaAs interfaces fabricated via atomic layer deposition [22].

2. The effective mass of the Γ valley in bulk materials is very low (0.043 m_0 for $In_{0.53}Ga_{0.47}As$) and yields a low density of states (DoS) that reduces the inversion charge, hence, the current [23, 24]. In fact, due to the low DoS, part of the applied V_{GS} is lost to move the Fermi level with respect to the conduction band edge in the semiconductor, resulting in a so-called "quantum-capacitance" [25] in series with C_{ox}.

1.2 TCAD approaches for III-V devices

To assess the possible advantages of new devices, one has to investigate a large number of architectural, geometry and material choices. Exploring experimentally all possibilities is unfeasible, due to the exceedingly large time and money. TCAD software comes then into

play by helping researchers in the R&D centres to explore the different engineering options and to select a small set to be fabricated and tested. The history of TCAD begun in the late sixties with the work of Gummel [26], Loeb [27] and Schroeder and Muller [28] and has evolved to fairly complex models and simulators of today. Among these the Drift-Diffusion model is perhaps the simplest and is still the core of many commercial TCAD products.

1.2.1 Drift Diffusion model

We take as starting point the semi-classical Boltzmann transport equation (BTE) [29]:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{R}} f + \mathbf{F} \cdot \nabla_{\mathbf{p}} f = \left.\frac{\partial f}{\partial t}\right|_{coll} + s(\mathbf{R}, \mathbf{p}, t) \tag{1.3}$$

for the distribution function f, which gives the probability of finding a particle with position $\mathbf{R}$ and momentum $\mathbf{p}$ at time t. Carriers are considered to be point particles with momentum $\mathbf{p} = m^* \cdot \mathbf{v}$ where m^* is the carrier effective mass. The electron density is expressed as:

$$n(\mathbf{R}, t) = \frac{1}{\Omega} \sum_{\mathbf{p}} f(\mathbf{R}, \mathbf{p}, t) \tag{1.4}$$

where Ω is the normalisation volume and f is the distribution function. From the zero-th order moment of the BTE we can write the continuity equations for both electrons and holes [29]:

$$\frac{\partial n}{\partial t} - \frac{1}{e} \nabla_{\mathbf{R}} \cdot \mathbf{J}_n + U_n = 0 \tag{1.5a}$$

$$\frac{\partial p}{\partial t} + \frac{1}{e} \nabla_{\mathbf{R}} \cdot \mathbf{J}_p + U_p = 0 \tag{1.5b}$$

where n and p are the electron and hole concentrations and U_n and U_p are the electrons and holes net recombination rate.

If we assume that the semiconductor is not degenerate, we can express the distribution function using Maxwell-Boltzmann statistics:

$$f(E) = \exp\left(-\frac{E - E_F}{k_b T}\right) \tag{1.6}$$

where k_b is the Boltzmann constant and T is the lattice temperature. The electron and hole current densities are then given by:

$$\mathbf{J}_n = -e\mu_n n \nabla\phi + eD_n \nabla n \tag{1.7a}$$

$$\mathbf{J}_p = -e\mu_p p \nabla\phi - eD_p \nabla p \tag{1.7b}$$

where μ_n and μ_p are the electrons and holes mobilities and D_n and D_p are the electrons and holes diffusion coefficients. The diffusion coefficients are linked to the mobilities by the Einstein relations: $D_n = k_b T \mu_n / e$ and $D_p = k_b T \mu_p / e$. The additional equation of the model is the Poisson equation:

$$\nabla \cdot \epsilon \nabla\phi = -e(p - n + N_D - N_A) \tag{1.8}$$

From the Eqs. 1.7 and Einstein relations it is clear that the mobilities are crucial parameters of the model.

The Drift-Diffusion model is well established and it has been developed to the point that commercial tools can describe with it complex 3D geometries. Sentautus Device from Synopsys is one of these tools. Finite element methods are typically used to solve the equations of the model.

The Drift-Diffusion model suffers from several limitations. It was derived for near-equilibrium conditions. Transport at high fields was addressed by including velocity saturation to the model but this entails the assumption that the velocity and field are gradually changing along the channel. Monte Carlo simulations have proven that this is not true [30, 31] in sub-micron devices. In addition, quantisation effects are not considered although they play a key role in ultra-scaled MOSFETs. Commercial TCAD tools try to overcome this limit by implementing the so called density gradient correction [32, 33] or MLDA [34]. Additional comparisons between drift-diffusion model and Monte Carlo are reported in [35].

1.2.2 Hydrodynamic models

Another transport model that can be derived from the Boltzmann equation is the hydrodynamic model [36]. In this model we retain the Poisson equation 1.8 and the continuity equations 1.7 but we also need energy-balance equations. For electrons we have [37]:

$$\frac{\partial n w_n}{\partial t} + \nabla \cdot \mathbf{S}_n = \mathbf{E} \cdot \mathbf{J}_n - U_n w_n + n \left. \frac{dw_n}{dt} \right|_{coll} \tag{1.9}$$

In the equation above, $w_n = 0.5\mathrm{tr}(k_B \hat{\mathbf{T}}_n) + 0.5 m_n^* v_n^2$ represents the electrons mean energy. $\hat{\mathbf{T}}_n$ is the electron temperature tensor defined as:

$$n k_B (T_n)_{ij} = m_n^* \int (u_{ni} - v_{ni})(u_{nj} - v_{nj}) f_n d^3 u_n \tag{1.10}$$

where $\mathrm{tr}(\hat{\mathbf{T}}_n) = T_{11} + T_{22} + T_{33}$, m_n^* is the electron effective mass, $\mathbf{u}_n$ is the electrons group velocity and $\mathbf{v}_n$ is the electrons mean velocity and i and j identify components along the axes.

$\mathbf{S}_n$ is the the energy flow, which is given by:

$$\mathbf{S}_n = -\kappa_n \nabla T_n - (w_n + k_B T_n) \frac{\mathbf{J}}{e}. \tag{1.11}$$

where κ_n is the thermal conductivity. The collision term of the balance equation is:

$$\left. \frac{dw_n}{dt} \right|_{coll} = -\frac{w_n - 1.5 k_b T_0}{\tau_{wn}} \tag{1.12}$$

where τ_{wn} is the energy relaxation time for the electrons. Finally

$$\mathbf{J}_n + n\tau_{pn} \frac{d}{dt} \left(\frac{J_n}{n} \right) = -e\mu_n n \nabla (\phi - k_B T_n / e) \tag{1.13}$$

where τ_{pn} is the electron momentum relaxation time. A similar set of equations can be written for the holes. The big difference with the drift-diffusion model is that now the electrons temperature can be different with respect to the lattice temperature. This is especially true for high electric fields, where the electrons are "hot" and is reflected by the T_n appearing in the equations above. The main drawback is that it does not work well in the near ballistic regime [38].

1.2.3 Full quantum models

Full quantum transport simulators were introduced when devices sizes became so small that quantisation effects could not be neglected anymore. Established models can be classified according to the functions on which they are based on. The Non-Equilibrium Green's Function (NEGF) [39] is one of the most widespread formalism. The non-equilibrium Green's function method solves the quantum transport problem in the most consistent and complete way, supports tunnelling through barrier and has been applied to 1D[40], 2D [41] and 3D problems [42]. It is a very general approach that can be used with many Hamiltonians, from atomistic [43] to EMA ones [44].

The main drawback of full quantum NEGF transport simulations is the heavy computational burden. In fact, if we take the Green's function $G(\mathbf{r}, t, \mathbf{r}', t')$ as introduced in [45], we can see that it depends on two vector arguments (positions $\mathbf{r}$ and $\mathbf{r}'$) and two scalar arguments (the times t and t'). A 3D discretisation would result in an enormous amount of mesh points. The inclusion of scattering mechanisms is very complicated and increases the already high computational requirements. Also, setting the proper boundary conditions requires specific calculations.

1.2.4 3D Monte Carlo

The Monte Carlo method is a statistical method which provides an exact solution of the Boltzmann transport equation without any a-priori assumption on the carrier distribution. This method solves the BTE by simulating the motion of a set of particles, motion interrupted by scattering events.

The first application of the method to the solution of the BTE was presented by Kurosawa in 1966 but it took almost 20 years before this technique got widespread use [46, 47, 48]. This method can be extended easily to support the description of new scattering mechanisms but it does not include quantisation effects in the direction normal to transport (subband formation) and along the transport direction (e.g. source/drain tunnelling). The former shortcoming is addressed by the Multi-Subband Monte Carlo.

1.2.5 Multi-Subband Monte Carlo

The traditional Monte Carlo method is an excellent tool for the simulation of a free carrier gas in far from equilibrium conditions, but the evolution of modern MOSFET devices requires a proper modelling of carrier quantisation phenomena which have important physical consequences on the device:

- the charge is displaced from the dielectric/semiconductor interface. This displacement affects the electrostatics;

- the scattering rates depend on the subband structure and on the corresponding wave-functions.

Quantum corrections have been proposed [49, 50] to address these consequences but they affect only the electrostatics and cannot capture effects such as subband splitting and modulation of the scattering rates [51, 52].

The Multi-subband Monte Carlo [53] (MSMC) method extends the Monte Carlo method for a free carrier gas by including quantisation effects, in the vertical direction (normal to transport), via the solution of the 1D Schrödinger equation. The inclusion of these effects makes this method quite demanding from a computational point of view but still much more efficient than the NEGF. The state of the art when this thesis began was that many hours were required to complete the simulation of one bias point including all the relevant scattering mechanisms.

1.3 Purpose of the work

The Multi-subband Monte Carlo method is at the foundation of the simulator developed by the nano-electronics research group of the University of Udine [30]. It is the focus of this work and will be described in Chapter 2. Considering the general remarks given so far about the MSMC method and about III-V materials, the main contributions of this work refer to three main areas:

1. the MSMC method requires lots of computational resources. Also, the computation time is increased because of the support of metal gate/high-k/III-V materials gate stacks, which requires the modelling of additional scattering mechanisms with respect to conventional silicon MOSFETs with SiO_2 dielectric. A careful optimisation work is required in order to reduce the simulation time and make it acceptable not only by academic research groups, but also by researchers in pre-industrial R& D centres of the semiconductor industries. Also, a code parallelisation work is needed to exploit properly the performance improvements provided by modern CPUs. Chapter 3 describes how these two tasks were carried out. To our knowledge this is the first implementation of parallelisation of MSMC solvers.

2. The improved version of the simulator must be tested on the field, comparing the results it delivers with experimental results and with other models. Chapter 4 reports comparisons with real devices and with two NEGF simulators, one based on an atomistic hamiltonian and one on a k.p hamiltonian.

3. III-V materials lack native oxides. The interfaces between these materials and high-k dielectrics show a much higher density of defects than the usual SiO_2/Si interfaces. These defects have an impact on device performance and must be properly modelled. Chapter 5 focuses on how we extended the MSMC simulator to support the description of interface defects for long channel device mobility simulations and to evaluate the current in short channel devices.

Finally, Chapter 6 gives some final remarks and describes possible future works.

Bibliography

[1] G.E. Moore. “Cramming More Components Onto Integrated Circuits”. In: *Proceedings of the IEEE* 86.1 (Jan. 1998), pp. 82–85.

[2] Yuan Taur, D.A. Buchanan, Wei Chen, D.J. Frank, K.E. Ismail, Shih-Hsien Lo, G.A. Sai-Halasz, R.G. Viswanathan, H.-J.C. Wann, S.J. Wind, and Hon-Sum Wong. “CMOS scaling into the nanometer regime”. In: *Proceedings of the IEEE* 85.4 (Apr. 1997), pp. 486–504.

[3] G. Baccarani, M.R. Wordeman, and R.H. Dennard. “Generalized scaling theory and its application to a 1/4 micrometer MOSFET design”. In: *Electron Devices, IEEE Transactions on* 31.4 (Apr. 1984), pp. 452–462.

[4] D.J. Frank, R.H. Dennard, E. Nowak, P.M. Solomon, Yuan Taur, and Hen-Sum Philip Wong. “Device scaling limits of Si MOSFETs and their application dependencies”. In: *Proceedings of the IEEE* 89.3 (Mar. 2001), pp. 259–288.

[5] Y. Taur and T. H. Ning. *Fundamentals of Modern VLSI Devices*. Cambridge University Press, 2013.

[6] N. Chabini and W. Wolf. “Reducing dynamic power consumption in synchronous sequential digital designs using retiming and supply voltage scaling”. In: *IEEE Symposium on VLSI Technology - Technical Digest* 12.6 (June 2004), pp. 573–589.

[7] S. Thompson, N. Anand, M. Armstrong, C. Auth, B. Arcot, M. Alavi, P. Bai, J. Bielefeld, R. Bigwood, J. Brandenburg, M. Buehler, S. Cea, V. Chikarmane, C. Choi, R. Frankovic, T. Ghani, G. Glass, W. Han, T. Hoffmann, M. Hussein, P. Jacob, A. Jain, C. Jan, S. Joshi, C. Kenyon, J. Klaus, S. Klopcic, J. Luce, Z. Ma, B. Mcintyre, K. Mistry, A. Murthy, P. Nguyen, H. Pearson, T. Sandford, R. Schweinfurth, R. Shaheed, S. Sivakumar, M. Taylor, B. Tufts, C. Wallace, P. Wang, C. Weber, and M. Bohr. “A 90 nm logic technology featuring 50 nm strained silicon channel transistors, 7 layers of Cu interconnects, low k ILD, and 1 /spl mu/m/sup 2/ SRAM cell”. In: *IEEE IEDM Technical Digest*. Dec. 2002, pp. 61–64.

[8] T. Ghani, M. Armstrong, C. Auth, M. Bost, P. Charvat, G. Glass, T. Hoffmann, K. Johnson, C. Kenyon, J. Klaus, B. McIntyre, K. Mistry, A. Murthy, J. Sandford, M. Silberstein, S. Sivakumar, P. Smith, K. Zawadzki, S. Thompson, and M. Bohr. “A 90nm high volume manufacturing logic technology featuring novel 45nm gate length strained silicon CMOS transistors”. In: *IEEE IEDM Technical Digest*. Dec. 2003.

[9] K. Mistry, M. Armstrong, C. Auth, S. Cea, T. Coan, T. Ghani, T. Hoffmann, A. Murthy, J. Sandford, R. Shaheed, K. Zawadzki, K. Zhang, S. Thompson, and M. Bohr. “Delaying forever: Uniaxial strained silicon transistors in a 90nm CMOS technology”. In: *IEEE Symposium on VLSI Technology - Technical Digest*. June 2004, pp. 50–51.

[10] C. Auth, A. Cappellani, J.-S. Chun, A. Dalis, A. Davis, T. Ghani, G. Glass, T. Glassman, M. Harper, M. Hattendorf, P. Hentges, S. Jaloviar, S. Joshi, J. Klaus, K. Kuhn, D. Lavric, M. Lu, H. Mariappan, K. Mistry, B. Norris, N. Rahhal-orabi, P. Ranade, J. Sandford, L. Shifren, V. Souw, K. Tone, F. Tambwe, A. Thompson, D. Towner, T. Troeger, P. Vandervoorn, C. Wallace, J. Wiedemer, and C. Wiegand. "45nm High-k + metal gate strain-enhanced transistors". In: *IEEE Symposium on VLSI Technology - Technical Digest*. June 2008, pp. 128–129.

[11] K. Mistry, C. Allen, C. Auth, B. Beattie, D. Bergstrom, M. Bost, M. Brazier, M. Buehler, A. Cappellani, R. Chau, C.-H. Choi, G. Ding, K. Fischer, T. Ghani, R. Grover, W. Han, D. Hanken, M. Hattendorf, J. He, J. Hicks, R. Huessner, D. Ingerly, P. Jain, R. James, L. Jong, S. Joshi, C. Kenyon, K. Kuhn, K. Lee, H. Liu, J. Maiz, B. Mcintyre, P. Moon, J. Neirynck, S. Pae, C. Parker, D. Parsons, C. Prasad, L. Pipes, M. Prince, P. Ranade, T. Reynolds, J. Sandford, L. Shifren, J. Sebastian, J. Seiple, D. Simon, S. Sivakumar, P. Smith, C. Thomas, T. Troeger, P. Vandervoorn, S. Williams, and K. Zawadzki. "A 45nm Logic Technology with High-k+Metal Gate Transistors, Strained Silicon, 9 Cu Interconnect Layers, 193nm Dry Patterning, and 100% Pb-free Packaging". In: *IEEE IEDM Technical Digest*. Dec. 2007.

[12] C-H. Lin, B. Greene, S. Narasimha, J. Cai, A. Bryant, C. Radens, V. Narayanan, B. Linder, H. Ho, A. Aiyar, E. Alptekin, J-J. An, M. Aquilino, R. Bao, V. Basker, N. Breil, M. Brodsky, W. Chang, L. Clevenger, D. Chidambarrao, C. Christiansen, D. Conklin, C. DeWan, H. Dong, L. Economikos, B. Engel, S. Fang, D. Ferrer, A. Friedman, A. Gabor, F. Guarin, X. Guan, M. Hasanuzzaman, J. Hong, D. Hoyos, B. Jagannathan, S. Jain, S-J. Jeng, J. Johnson, B. Kannan, Y. Ke, B. Khan, B. Kim, S. Koswatta, A. Kumar, T. Kwon, U. Kwon, L. Lanzerotti, H-K Lee, W-H. Lee, A. Levesque, W. Li, Z. Li, W. Liu, S. Mahajan, K. McStay, H. Nayfeh, W. Nicoll, G. Northrop, A. Ogino, C. Pei, S. Polvino, R. Ramachandran, Z. Ren, R. Robison, I. Saraf, V. Sardesai, S. Saudari, D. Schepis, C. Sheraw, S. Siddiqui, L. Song, K. Stein, C. Tran, H. Utomo, R. Vega, G. Wang, H. Wang, W. Wang, X. Wang, D. Wehelle-Gamage, E. Woodard, Y. Xu, Y. Yang, N. Zhan, K. Zhao, C. Zhu, K. Boyd, E. Engbrecht, K. Henson, E. Kaste, S. Krishnan, E. Maciejewski, H. Shang, N. Zamdmer, R. Divakaruni, J. Rice, S. Stiffler, and P. Agnello. "High performance 14nm SOI FinFET CMOS technology with 0.0174 μm^2 embedded DRAM and 15 levels of Cu metallization". In: *IEEE IEDM Technical Digest*. Dec. 2014, pp. 3.8.1–3.8.3.

[13] S. Natarajan, M. Agostinelli, S. Akbar, M. Bost, A. Bowonder, V. Chikarmane, S. Chouksey, A. Dasgupta, K. Fischer, Q. Fu, T. Ghani, M. Giles, S. Govindaraju, R. Grover, W. Han, D. Hanken, E. Haralson, M. Haran, M. Heckscher, R. Heussner, P. Jain, R. James, R. Jhaveri, I. Jin, H. Kam, E. Karl, C. Kenyon, M. Liu, Y. Luo, R. Mehandru, S. Morarka, L. Neiberg, P. Packan, A. Paliwal, C. Parker, P. Patel, R. Patel, C. Pelto, L. Pipes, P. Plekhanov, M. Prince, S. Rajamani, J. Sandford, B. Sell, S. Sivakumar, P. Smith, B. Song, K. Tone, T. Troeger, J. Wiedemer, M. Yang, and K. Zhang. "A 14nm logic technology featuring 2nd-generation FinFET, air-gapped interconnects, self-aligned double patterning and a 0.0588 μm^2 SRAM cell size". In: *IEEE IEDM Technical Digest*. Dec. 2014, pp. 3.7.1–3.7.3.

[14] M. Radosavljevic, G. Dewey, J.M. Fastenau, J. Kavalieros, R. Kotlyar, B. Chu-Kung, W.K. Liu, D. Lubyshev, M. Metz, K. Millard, N. Mukherjee, L. Pan, R. Pillarisetty,

W. Rachmady, U. Shah, and R. Chau. "Non-planar, multi-gate InGaAs quantum well field effect transistors with high-K gate dielectric and ultra-scaled gate-to-drain/gate-to-source separation for low power logic applications". In: *IEEE IEDM Technical Digest*. Dec. 2010, pp. 6.1.1–6.1.4.

[15] G. Dewey, B. Chu-Kung, R. Kotlyar, M. Metz, N. Mukherjee, and M. Radosavljevic. In: *IEEE Symposium on VLSI Technology - Technical Digest*.

[16] G. Doornbos and M. Passlack. "Benchmarking of III-V n-MOSFET Maturity and Feasibility for Future CMOS". In: *IEEE Electron Device Lett.* 31.10 (Oct. 2010), pp. 1110–1112.

[17] M. Luisier. "Performance Comparison of GaSb, Strained-Si, and InGaAs Double-Gate Ultrathin-Body n-FETs". In: *IEEE Electron Device Lett.* 32.12 (Dec. 2011), pp. 1686–1688.

[18] Seung Hyun Park, Yang Liu, N. Kharche, M.S. Jelodar, G. Klimeck, M.S. Lundstrom, and M. Luisier. "Performance Comparisons of III-V and Strained-Si in Planar FETs and Nonplanar FinFETs at Ultrashort Gate Length (12 nm)". In: *IEEE Trans. on Electron Devices* 59.8 (Aug. 2012), pp. 2107–2114.

[19] `http://www.itrs.net/Links/2013ITRS/2013Tables/PIDS_2013Tables.xlsx`.

[20] Jesus A del Alamo. "Nanometre-scale electronics with III-V compound semiconductors". In: *Nature* 479.7373 (Nov. 2011), pp. 317–323.

[21] N. Taoka, M. Yokoyama, S.H. Kim, R. Suzuki, R. Iida, S. Lee, T. Hoshii, W. Jevasuwan, T. Maeda, T. Yasuda, O. Ichikawa, N. Fukuhara, M. Hata, M. Takenaka, and S. Takagi. "Impact of Fermi level pinning inside conduction band on electron mobility of $In_xGa_{1-x}As$ MOSFETs and mobility enhancement by pinning modulation". In: *IEEE IEDM Technical Digest*. 2011, pp. 27.2.1–27.2.4.

[22] P.D. Ye, G.D. Wilk, J. Kwo, B. Yang, H.-J.L. Gossmann, M. Frei, S.N.G. Chu, J.P. Mannaerts, M. Sergent, M. Hong, K.K. Ng, and J. Bude. "GaAs MOSFET with oxide gate dielectric grown by atomic layer deposition". In: *IEEE Electron Device Lett.* 24.4 (Apr. 2003), pp. 209–211.

[23] S.R. Mehrotra, M. Povolotskyi, D.C. Elias, T. Kubis, J.J.M. Law, M.J.W. Rodwell, and G. Klimeck. "Simulation Study of Thin-Body Ballistic n-MOSFETs Involving Transport in Mixed Γ-L Valleys". In: *IEEE Electron Device Lett.* 34.9 (Sept. 2013), pp. 1196–1198.

[24] M.V. Fischetti, L. Wang, B. Yu, C. Sachs, P.M. Asbeck, Y. Taur, and M. Rodwell. "Simulation of Electron Transport in High-Mobility MOSFETs: Density of States Bottleneck and Source Starvation". In: *IEEE IEDM Technical Digest*. Dec. 2007, pp. 109–112.

[25] J. Genoe, C. Van Hoof, W. Van Roy, J.H. Smet, K. Fobelets, R.P. Mertens, and G. Borghs. "Capacitances in double-barrier tunneling structures". In: *IEEE Trans. on Electron Devices* 38.9 (Sept. 1991), pp. 2006–2012.

[26] H.K. Gummel. "A self-consistent iterative scheme for one-dimensional steady state transistor calculations". In: *IEEE Trans. on Electron Devices* 11.10 (Oct. 1964), pp. 455–465.

[27] H.W. Loeb, R. Andrew, and W. Love. "Application of 2-dimensional solutions of the Shockley-Poisson equation to inversion-layer m.o.s.t. devices". In: *Electronics Letters* 4.17 (Aug. 1968), pp. 352–354.

[28] J.E. Schroeder and R.S. Muller. "IGFET Analysis through numerical solution of Poisson's equation". In: *IEEE Trans. on Electron Devices* 15.12 (Dec. 1968), pp. 954–961.

[29] Mark Lundstrom. *Fundamentals of Carrier Transport*. Second. Cambridge University Press, 2000.

[30] L. Lucci, P. Palestri, D. Esseni, L. Bergagnini, and L. Selmi. "Multisubband Monte Carlo Study of Transport, Quantization, and Electron-Gas Degeneration in Ultrathin SOI n-MOSFETs". In: *IEEE Trans. on Electron Devices* 54.5 (2007), pp. 1156–1164.

[31] D. Lizzit, D. Esseni, P. Palestri, P. Osgnach, and L. Selmi. "Performance Benchmarking and Effective Channel Length for Nanoscale InAs, $In_{0.53}Ga_{0.47}As$, and sSi n-MOSFETs". In: *IEEE Trans. on Electron Devices* 61.6 (June 2014), pp. 2027–2034.

[32] T. Hohr, A. Schenk, A. Wettstein, and W. Fichtner. "On density-gradient modeling of tunneling through insulators". In: *Proc.SISPAD*. 2002, pp. 275–278.

[33] `http://www.synopsys.com/Tools/TCAD/CapsuleModule/tcadnews_jun2012.pdf` page 4.

[34] G. Paasch and H. Übensee. "A Modified Local Density Approximation. Electron Density in Inversion Layers". In: *physica status solidi (b)* 113.1 (1982), pp. 165–178.

[35] J.D. Bude. "MOSFET modeling into the ballistic regime". In: *Proc.SISPAD*. 2000, pp. 23–26.

[36] R. Stratton. "Diffusion of Hot and Cold Electrons in Semiconductor Barriers". In: *Phys. Rev.* 126 (6 1962), pp. 2002–2014.

[37] A. Forghieri, R. Guerrieri, P. Ciampolini, A. Gnudi, M. Rudan, and G. Baccarani. "A new discretization strategy of the semiconductor equations comprising momentum and energy balance". In: *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 7.2 (1988), pp. 231–242.

[38] Jung-Hoon Rhew, Zhibin Ren, and MarkS. Lundstrom. "Benchmarking Macroscopic Transport Models for Nanotransistor TCAD". In: *Journal of Computational Electronics* 1.3 (2002), pp. 385–388. ISSN: 1569-8025.

[39] Mathieu Luisier, Andreas Schenk, and Wolfgang Fichtner. "Quantum transport in two- and three-dimensional nanoscale transistors: Coupled mode effects in the nonequilibrium Green's function formalism". In: *Journal of Applied Physics* 100.4 (2006), p. 043713.

[40] Roger Lake, Gerhard Klimeck, R. Chris Bowen, and Dejan Jovanovic. "Single and multiband modeling of quantum electron transport through layered semiconductor devices". In: *Journal of Applied Physics* 81.12 (1997), pp. 7845–7869.

[41] A. Martinez, A. Svizhenko, M.P. Anantram, J.R. Barker, A.R. Brown, and A. Asenov. "A study of the effect of the interface roughness on a DG-MOSFET using a full 2D NEGF technique". In: *IEEE IEDM Technical Digest*. Dec. 2005, pp. 616–619.

[42] A. Martinez, J.R. Barker, A. Asenov, A. Svizhenko, and M.P. Anantram. "Developing a full 3D NEGF simulator with random dopant and interface roughness". In: *Journal of Computational Electronics* 6.1-3 (2007), pp. 215–218.

[43] Mathieu Luisier, Andreas Schenk, Wolfgang Fichtner, and Gerhard Klimeck. "Atomistic simulation of nanowires in the $sp^3d^5s^*$ tight-binding formalism: From boundary conditions to strain calculations". In: *Phys. Rev. B* 74 (20 Nov. 2006), p. 205323.

[44] S. Datta. *Quantum Transport*. Cambridge University Press, 2013.

[45] W. Schieve and L. Horwitz. *Quantum Statistical Mechanics*. Cambridge University Press, 2009.

[46] C. Jacoboni. "A new approach to Monte Carlo simulation". In: *IEEE IEDM Technical Digest*. Dec. 1989, pp. 469–472.

[47] Carlo Jacoboni and Lino Reggiani. "The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials". In: *Reviews of Modern Physics* 55 (3 July 1983), pp. 645–705.

[48] M. Fischetti and S. Laux. "Monte carlo analysis of electron transport in small semiconductor devices including band-structure and space-charge effects". In: *Phys. Rev. B* 38 (14 Nov. 1988), pp. 9721–9745.

[49] D.K. Ferry, R. Akis, and D. Vasileska. "Quantum effects in MOSFETs: use of an effective potential in 3D Monte Carlo simulation of ultra-short channel devices". In: *IEEE IEDM Technical Digest*. Dec. 2000, pp. 287–290.

[50] B. Winstead and U. Ravaioli. "A quantum correction based on Schrodinger equation applied to Monte Carlo device simulation". In: *IEEE Trans. on Electron Devices* 50.2 (Feb. 2003), pp. 440–446.

[51] D. Esseni and A. Abramo. "Modeling of electron mobility degradation by remote Coulomb scattering in ultrathin oxide MOSFETs". In: *IEEE Trans. on Electron Devices* 50.7 (July 2003), pp. 1665–1674.

[52] D. Esseni, A. Abramo, L. Selmi, and E. Sangiorgi. "Physically based modeling of low field electron mobility in ultrathin single- and double-gate SOI n-MOSFETs". In: *IEEE Trans. on Electron Devices* 50.12 (Dec. 2003), pp. 2445–2455.

[53] M. V. Fischetti and S. E. Laux. "Monte Carlo study of electron transport in silicon inversion layers". In: *Phys. Rev. B* 48 (4 July 1993), pp. 2244–2274.

Chapter 2

The Multi-subband Monte Carlo simulator

The Monte Carlo (MC) method is a powerful technique for solving the Boltzmann Transport Equation [1, 2]. This method aims to simulate the motion of a set of particles moving inside an electron device and, specifically, the channel of a MOS transistor. The motion of these particles is in general not ballistic since scattering events invariably occur while moving through the device.

The time of the simulation is divided into a discrete set of intervals called time steps. During each time step, two main phases occur [2]. During the first one, called "free flight", a particle is moved for a given time according to Newton's law. The electric field (and a possible lateral magnetic term) provides the driving force. This is a deterministic step. The second phase begins when a scattering event ends the free flight. Scattering events change the momentum of the particle and are stochastic in nature.

Two versions of the Monte Carlo method can be categorised according to how the motion of the particles is simulated during the time steps: a *single-particle* Monte Carlo computes the motion of a particle through all the time steps before considering the next particle [2, 3]; an *ensemble* Monte Carlo algorithm simulates the motion of all the particles for one time step before beginning computations of the next time step [2].

Particle statistics are collected periodically and in particular: before scattering in single particle MC and at the end of each time step in ensemble MC. A new estimate of the particle distribution is thus periodically obtained, and is then used to solve the Poisson equation to obtain an updated potential energy profile. This corresponding potential energy profile is used to compute the driving force for the next iteration of the Monte Carlo transport phase.

The iteration between the transport phase and the solution of the Poisson equation is performed until convergence is reached [4, 5] where convergence is typically evaluated in terms of time stability of the averages and reduction of variance.

The MC method in the terms described above is suited to simulate a free-particle gas, but in modern MOS transistors quantisation effects (as those discussed in the introduction) play a significant role and the method must be adapted to take into account these effects.

In this chapter we will describe and analyze the structure, models and algorithms of an advanced Multi-subband Monte Carlo simulator developed by the nano-electronics research group at the University of Udine [6].

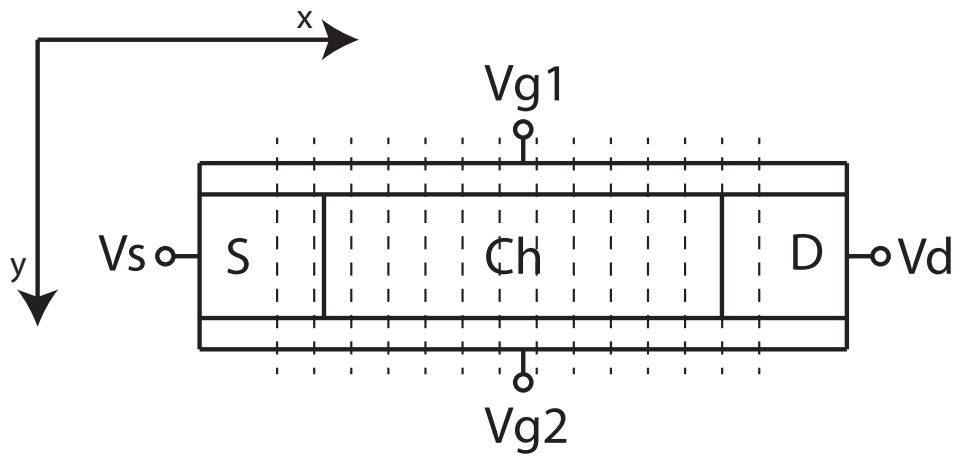


Figure 2.1: Sketch of a planar double gate MOSFET as described in the MSMC simulator of this work.

2.1 Introduction to the MSMC method

The Multi-subband Monte Carlo [6, 7, 8, 9, 10, 11, 12] method for MOSFET devices extends the Monte Carlo method for a free-electron gas by including quantisation effects via the solution of the Schrödinger equation in the direction normal to transport. Before the simulation begins, the device is partitioned in a number of sections along the transport direction x, as shown in Fig.2.1.

Since this is an iterative method, a suitable initial condition for the potential energy profile is needed. Section 2.6 describes how these initial conditions are computed. The simulator proceeds by iterating four steps (see Fig. 2.2. With reference to the planar MOSFET case of Fig. 2.1, they are:

Schrödinger equation : The Schrödinger equation is solved in each section along the quantisation direction y to obtain the subband energies $E_{\nu,n}$ and the associated wave-functions $\psi_{n,v}(y)$;

Scattering rates computation : Scattering rates are computed in each section using $E_{\nu,n}$ and $\psi_{n,v}(y)$ calculated at the previous step;

Monte Carlo transport : Particles motion is simulated using $dE_{\nu,n}/dx$ as the driving force in order to obtain the subband particle distributions for each section. The total charge density is then calculated;

Poisson equation : The new particle distribution is used to solve the 2D Poisson equation to obtain a new potential energy profile. This new profile is the input of the Schrödinger equation at the next iteration.

These four steps are carried out in sequence until convergence is reached. The flow-chart of the simulator is show in Fig.2.2.

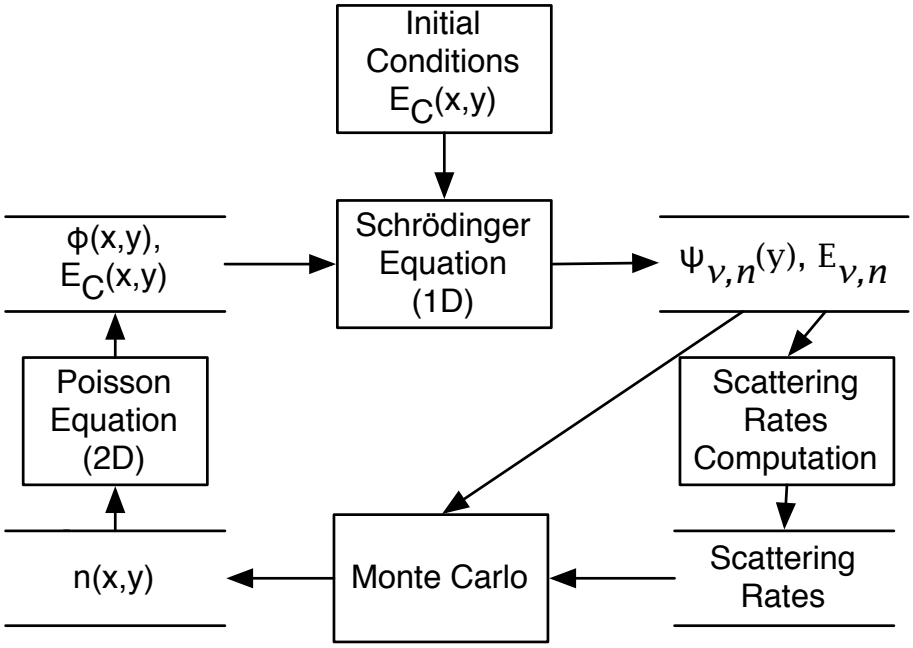


Figure 2.2: Flow-chart of a Multi-subband Monte Carlo simulator.

2.2 Solution of the Schrödinger equation

Electrons can occupy only states with a well defined energy $E_{\nu,n}$ given by

$$E_{\nu,n} = E'_{\nu,n} + E_{\nu 0}, \tag{2.1}$$

where $E_{\nu 0}$ is the conduction band minimum for the valley ν and n is the eigenvalue index, which identifies the so called *subband* of valley ν. $E'_{\nu,n}$ and the corresponding envelope wave function $\Phi_{\nu,n}$ are obtained by solving the stationary Schrödinger equation

$$[\hat{E}_{cb}^{(\nu)}(-i\nabla_R) + E_C(y)]\Phi_{\nu,n}(\mathbf{R}) = E'_{\nu,n}\Phi_{\nu,n}(\mathbf{R}) \tag{2.2}$$

where $E_C(y)$ is the confining potential energy profile. $\mathbf{R} = (\mathbf{r}, r_y) = ((r_x, r_z), r_y)$ and $\hat{E}_{cb}^{(\nu)}$ is the operator that accounts for the effects of the crystal potential. Finding the solutions for this equation can be easy or difficult, depending on the form of the $\hat{E}_{cb}^{(\nu)}(-i\nabla_R)$ operator. If we employ the parabolic effective mass approximation, we can rewrite the equation as

$$\left[-\frac{\hbar^2}{2}\left(\frac{1}{m_x}\frac{\partial^2}{\partial x^2} + \frac{1}{m_y}\frac{\partial^2}{\partial y^2} + \frac{1}{m_z}\frac{\partial^2}{\partial z^2}\right) + E_C(y)\right]\Phi_{\nu,n}(\mathbf{R}) = E'_{\nu,n}\Phi_{\nu,n}(\mathbf{R}). \tag{2.3}$$

As carriers in a MOSFET are confined in one direction (y) and can move freely in the other two ones (x and z), the wave functions can be written as

$$\Phi_{\nu,n}(\mathbf{R}) = \xi_{\nu,n}(y)\frac{\exp(i\mathbf{k}\cdot\mathbf{r})}{\sqrt{A}}, \tag{2.4}$$

where $\mathbf{k}$ is the wave vector on the transport plane ($\mathbf{k} = (k_x, k_z)$) and A is the normalisation area. By plugging 2.4 into 2.3, the Schrödinger equation can be simplified to

$$\frac{-\hbar^2}{2m_y}\frac{\partial^2 \xi_{\nu,n}}{\partial y^2} + E_C(y) = \varepsilon^P_{\nu,n}\xi_{\nu,n}. \tag{2.5}$$

Recalling Eq. 2.1, the total energy is:

$$E^P_{\nu,n}(\mathbf{k}) = E_{\nu 0} + \varepsilon^P_{\nu,n} + \frac{\hbar^2}{2}\left(\frac{k_x^2}{m_x} + \frac{k_z^2}{m_z}\right). \tag{2.6}$$

From here on we will assume that the wave functions along the quantisation direction ξ are real. The eigenvalues given by Eq. 2.6 are correct only for energies close to the conduction band valley minimum, which can be well represented by an ellipsoidal constant-energy surface. For higher energies, a more accurate approximation is obtained by applying a non-parabolicity correction as described in [13]. Non-parabolicity corrections are especially relevant for III-V semiconductors. The correction is applied to the eigenvalues $\varepsilon^P_{\nu,n}$ to obtain the non parabolic eigenvalues $\varepsilon^{NP}_{\nu,n}$:

$$\varepsilon^{NP}_{\nu,n} = U_{\nu,n} + \frac{\sqrt{1 + 4\alpha_\nu \cdot (\varepsilon^P_{\nu,n} - U_{\nu,n})} - 1}{2\alpha_\nu} \tag{2.7}$$

where $U_{\nu,n}$ is given by

$$U_{\nu,n} = \int |\xi_{\nu,n}(y)|^2 E_C(y) dy. \tag{2.8}$$

The wave-functions are unchanged with respect to the parabolic case. Recalling again Eq. 2.1 the discrete energy levels are:

$$E^{NP}_{\nu,n} = \varepsilon^{NP}_{\nu,n} + E_{\nu 0}, \tag{2.9}$$

and the total energy is:

$$E^{NP}_{\nu,n}(\mathbf{k}) = E_{\nu 0} + \varepsilon^{NP}_{\nu,n} + \frac{\sqrt{1 + 4\alpha_\nu\left(\frac{\hbar^2}{2}\left(\frac{k_x^2}{m_x} + \frac{k_z^2}{m_z}\right) + \varepsilon^P_{\nu,n} - U_{\nu,n}\right)} - \sqrt{1 + 4\alpha_\nu\left(\varepsilon^P_{\nu,n} - U_{\nu,n}\right)}}{2\alpha_\nu} \tag{2.10}$$

where α_ν is the non-parabolicity coefficient for the valley ν. Figure 2.3 shows an example of solution of Eq.2.5. The left figure shows the subband energy along the y direction. In the right figure, the corresponding subband energy levels in each section have been connected to obtain a complete contiguous profile along the transport direction x.

2.2.1 Crystal orientation

The discussion above holds for [100]/(001) silicon channel devices, where the three axis of the ellipsoidal constant energy surfaces are aligned with the three main axis of the x, y, z device coordinate system DCS (which assumes that quantisation effects are considered along the y direction and the transport plane is the xz plane). The axes of the ellipsoids are also aligned with the k_x, k_y, k_z directions of the crystal coordinate system CCS. This

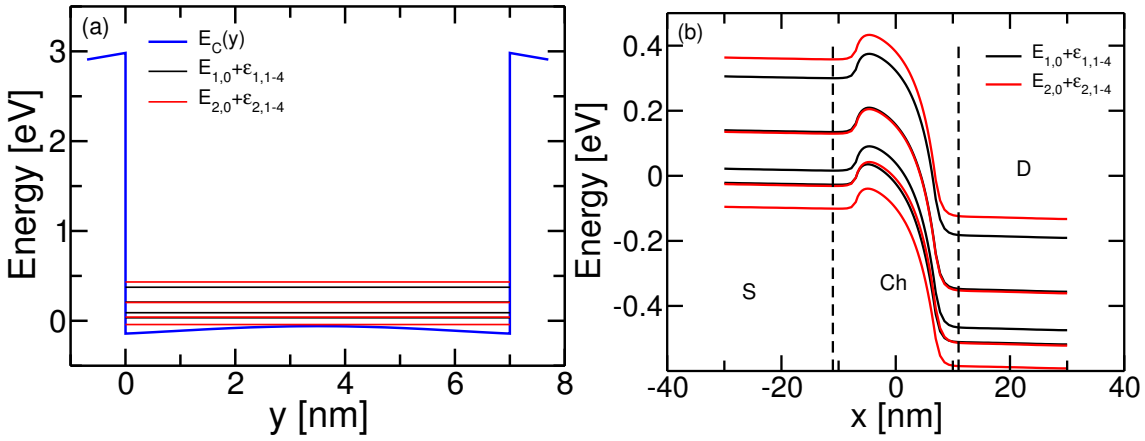


Figure 2.3: (a) Potential energy profile $E_C(y)$ used to solve Eq.2.5 in a sample slice (x = -4.5 nm) and its solutions $E_{\nu,0} + \varepsilon_{\nu,n}$. (b) Profile of the allowed energy levels along the transport direction x. The device is a 7 nm thick silicon DG-SOI with a channel 14 nm long and a 0.7 nm thick SiO_2 dielectric. $V_{GS} = 0.5V$, $V_{DS} = 0.5V$.

alignment disappears for other crystal orientations or other channel materials. To treat these cases, a new coordinates system is defined with its main axes are always aligned with the axes of the ellipsoids (ellipsoidal coordinate system, ECS). The main axes of the ECS are k_{t1}, k_{t2}, k_l and m_{t1}, m_{t2}, m_l are the three effective masses. A more complete treatment of these coordinate systems is shown in [14]. In unstrained cubic semiconductors $m_{t1} = m_{t2} = m_t$. The transformation between DCS and ECS is given by:

$$(k_{t1}, k_l, k_{t2})^T = \mathbf{R}_{D\to E} \cdot (k_x, k_y, k_z)^T, \tag{2.11}$$

where $\mathbf{R}_{D\to E}$ is a transformation matrix from DCS to ECS. In order to solve the Schrödinger equation, we must define the matrix:

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \mathbf{R}^T_{D\to E} \cdot \begin{bmatrix} 1/m_{t1} & 0 & 0 \\ 0 & 1/m_l & 0 \\ 0 & 0 & 1/m_{t2} \end{bmatrix} \cdot \mathbf{R}_{D\to E} \cdot \tag{2.12}$$

Consistent with [15], the envelope wave-function for a generic orientation is then given by:

$$\Phi_{\nu,n}(\mathbf{R}) = \frac{\xi_{\nu,n}(y)}{\sqrt{A}} \exp[i(k_x x + k_z z)] exp\Big[-i\frac{(w_{13}k_x + w_{33}k_z)y}{w_{23}}\Big]. \tag{2.13}$$

Equation 2.5 is replaced with:

$$\frac{-\hbar^2 w_{23}}{2}\frac{\partial^2 \xi_{\nu,n}}{\partial y^2} + E_C(y) = \varepsilon^P_{\nu,n}\xi_{\nu,n} \tag{2.14}$$

where the quantisation mass is now $1/w_{23}$. The longitudinal axis of the ellipsoid forms the angle α with the k_x axis of the DCS. Different valleys will form different angles [16].

2.3 Scattering rates

The motion of carriers in devices is subject to numerous collision events. These collisions, called *scatterings*, tend to restore the equilibrium conditions inside the device when external

stimuli are applied. Many are the mechanisms that can cause carrier transitions between two states and all must be properly modelled. Transitions between subbands of the same valley are called *intra-valley*, while transitions between subbands of different valleys are called *inter-valley*. Let's suppose that these mechanisms manifest themselves through a stationary scattering potential $U_{sc}(\mathbf{R})$ that adds to the potential energy in Eq. 2.3. $U_{sc}(\mathbf{R})$ is typically a rapidly varying function of $\mathbf{R}$ on the scale of the inter-atomic distance. This potential allows transitions from an initial state $(n, \mathbf{k})$ to a final state $(n, \mathbf{k}')$. Here the index n is a "global" index, which includes the valley index. The number of transitions between two states per unit of time for a given scattering mechanism m is called *scattering rate* and can be generally computed using the Fermi golden rule [16, p. 2.5.4]:

$$S^m_{n,n'}(\mathbf{k}, \mathbf{k}') = \frac{2\pi}{\hbar}|M_{n,n'}(\mathbf{k}, \mathbf{k}')|^2 \delta[E_n(\mathbf{k}) - E_{n'}(\mathbf{k}')]. \tag{2.15a}$$

For harmonic time varying scattering potentials $U_{sc}(\mathbf{R}, t)$ at angular frequency ω the Fermi golden rule reads:

$$\begin{aligned} S^m_{n,n'}(\mathbf{k}, \mathbf{k}') = &\frac{2\pi}{\hbar}|M^{(ab)}_{n,n'}(\mathbf{k}, \mathbf{k}')|^2 \delta[E_n(\mathbf{k}) - E_{n'}(\mathbf{k}') + \hbar\omega] \\ &+\frac{2\pi}{\hbar}|M^{(em)}_{n,n'}(\mathbf{k}, \mathbf{k}')|^2 \delta[E_n(\mathbf{k}) - E_{n'}(\mathbf{k}') - \hbar\omega]. \end{aligned} \tag{2.15b}$$

Note that in the first equation, the Dirac's delta function imposes that the energy of the final state must be equal to the energy of the initial state, meaning that the transitions are elastic and conserve the total energy. The second equation entails an increase (absorption) or decrease (emission) of a quantum of energy equal to $\hbar\omega$. These scattering mechanisms are called *inelastic*. The M factor appearing in both equations is called *matrix element*. The matrix element for an elastic intra-valley transition among states of a 2D electron gas is given by [16, p. 4.1.2]:

$$M_{n,n'}(\mathbf{k}, \mathbf{k}') = \frac{(2\pi)^2}{A} \int_y \xi_{\nu,n'}(y)\xi_{\nu,n}(y)U_{2T}(-\mathbf{q}, y)dy, \tag{2.16}$$

where A is the normalisation area, $\mathbf{q} = (\mathbf{k}' - \mathbf{k})$ is the wave-vector variation ($q = |\mathbf{q}|$) and U_{2T} is given by $U_{2T} = \int_A U_{sc}(\mathbf{R}) \exp(i(\mathbf{k} - \mathbf{k}')\mathbf{r})dr$. Inter-valley transitions can be neglected if the spectral components of the scattering potentials are small for wave-vectors comparable to the extension of the first Brillouin zone. If the matrix element has a very weak dependence on $\mathbf{q}$ then the mechanism is called *isotropic*.

The MSMC simulator considered in this thesis work supports various scattering mechanisms which can be divided in two broad groups. The first are isotropic phonon scattering mechanisms while the second includes anisotropic scattering mechanisms (Coulomb scattering, surface roughness scattering, alloy scattering, polar optical phonon scattering and remote phonon scattering).

As shown later in Section 2.4, the total scattering rate out of a state $(n, \mathbf{k})$ is a very important quantity for the practical implementation of Monte Carlo simulation algorithm. This quantity is obtained by summing Eqs. 2.15a or 2.15b over the final states. This sum is typically converted to an integral over all final states $\mathbf{k}'$. Generally speaking:

$$\frac{(2\pi)^2}{A} \sum_{\mathbf{k}'} S(\mathbf{k}') \approx n_{sp} \int_{\mathbf{k}'} S(\mathbf{k}')d\mathbf{k}'. \tag{2.17}$$

In our case, the spin multiplicity factor n_{sp} is set to 1 because spin is not changed by a scattering event. Also, the expression above makes the normalisation area A disappear when computing the matrix elements.

Conceptually, the SC step can be divided in two sub-steps. The first sub-step computes the matrix elements for all the scattering mechanisms under consideration including the effects of screening (see below). The second sub-step computes the scattering rates and integrates them according to Eqs 2.15a, 2.15b and 2.17.

2.3.1 Screening

The free carriers inside a device screen the scattering potential, thus reducing its impact. A larger concentration of free carriers implies a more effective screening effect. From the modelling point of view, the screening effect alters the matrix elements computed so far (which now become the unscreened matrix elements) according to:

$$M^{(unscr)}_{\nu,m,m'}(\mathbf{q}) = \sum_{w,n,n'} \epsilon^{w,n,n'}_{\nu,m,m'}(\mathbf{q}) M^{(scr)}_{w,n,n'}(\mathbf{q}) \tag{2.18}$$

where $\epsilon^{w,n,n'}_{\nu,m,m'}$ is called *dielectric function* and is given by [16]:

$$\epsilon^{w,n,n'}_{\nu,m,m'}(\mathbf{q}) = \delta_{w,\nu}\delta_{n,m}\delta_{n',m'} - \frac{e^2}{q(\epsilon_S + \epsilon_{ox})} \Pi_{w,n,n'}(\mathbf{q}) F^{w,n,n'}_{\nu,m,m'}(q) \tag{2.19}$$

where ϵ_S is the dielectric constant of the semiconductor and ϵ_{ox} is the dielectric constant of the oxide. Π is the polarisation factor and is given by [16]:

$$\Pi_{w,n,n'}(\mathbf{q}) = \frac{1}{A} \sum_{\mathbf{k}} \frac{f_{w,n'}(\mathbf{k}+\mathbf{q}) - f_{w,n'}(\mathbf{k})}{E_{w,n'}(\mathbf{k}+\mathbf{q}) - E_{w,n}(\mathbf{k})} \tag{2.20}$$

where f is the occupation function of the subband. F is the screening form factor given by

$$F^{w,n,n'}_{\nu,m,m'}(q) = \int dy \xi_{\nu,m}(y)\xi_{\nu,m'}(y) \int dy_0 \xi_{w,n}(y_0)\xi_{w,n'}(y_0)\phi_{pcN}(q,z,z_0) \tag{2.21}$$

where ϕ_{pcN} is given by

$$\phi_{pcN}(q,y,y_0) = \frac{q(\epsilon_S + \epsilon_{ox})}{e} \phi_{pc}(q,y,y_0) \tag{2.22}$$

and ϕ_{pc} is the potential produced by a point charge. This potential is given by Eq. 2.37 or 2.38 and will be further discussed in section 2.3.3.

The formulation above is known as *tensorial* screening. When $\mathbf{q}$ is small we can then employ a simpler expression, known as *scalar* screening. For inter-subband transitions the scalar formulation implies:

$$M^{(scr)}_{\nu,m,m'}(\mathbf{q}) \approx M_{\nu,m,m'}(\mathbf{q}), \quad m \neq m'. \tag{2.23}$$

For intra-subband transitions we have:

$$M^{(scr)}_{\nu,m,m}(\mathbf{q}) = \frac{M_{\nu,m,m}(\mathbf{q})}{\epsilon_D(\mathbf{q})}, \tag{2.24}$$

where

$$\epsilon_D(\mathbf{q}) = 1 - \sum_{w,n} \frac{e^2}{q(\epsilon_S + \epsilon_{ox})} \Pi_{w,n,n}(\mathbf{q}). \tag{2.25}$$

Scalar screening can be used for bulk and single-gate SOI devices, but it becomes inaccurate for double-gate SOI devices [17]. We employ tensorial screening for all the simulations of this work.

2.3.2 Non-polar Phonon scattering

If the lattice temperature is not too low, the atoms oscillate with respect to their rest positions. These vibrations perturb the otherwise perfectly periodic crystal potential and cause scattering events. The energy associated to a vibration mode ν with propagation wave-vector Q ($|\mathbf{Q}| = Q$) is quantised and is given by $E = \hbar\omega_{\nu,Q}(n_Q + 0.5)$. This energy can be interpreted as the total energy of a group of n_Q particles, called *phonons*, whose energy is $\hbar\omega_Q$. The number of phonons occupying state $(\nu, \mathbf{Q})$ is given by the Bose-Einstein statistics:

$$n_{\nu,\mathbf{Q}} = \frac{1}{\exp\left(\frac{\hbar\omega_{\nu,\mathbf{Q}}}{K_B T}\right) - 1}. \tag{2.26}$$

There are two kinds of phonon: acoustic and optical. Let's begin with the acoustic intra-valley phonons. We have two matrix elements, one for the phonon absorption and one for the phonon emission [16]:

$$|M^{(ab)}_{n,n'}(\mathbf{k}, \mathbf{k}')|^2 = \delta_{\mathbf{k}',(\mathbf{k}+\mathbf{q})} \frac{K_B T D^2_{ac}}{2\rho A v_s^2} F_{n,n'} \tag{2.27a}$$

$$|M^{(em)}_{n,n'}(\mathbf{k}, \mathbf{k}')|^2 = \delta_{\mathbf{k}',(\mathbf{k}-\mathbf{q})} \frac{K_B T D^2_{ac}}{2\rho A v_s^2} F_{n,n'} \tag{2.27b}$$

where D_{ac} is the effective deformation potential and $F_{n,n'}$ is the form factor given by

$$F_{n,n'} = \int_y |\xi_{n'}(y)\xi_n(y)|^2 dy \tag{2.28}$$

In the expression above the purpose of the Kronecker delta is to select the correct $\mathbf{q}$ and $\mathbf{Q}$ values, but otherwise the matrix element does not depend on q. Intra-valley acoustic phonon scattering is an elastic and isotropic scattering mechanism. Furthermore, the scattering rates for both absorption and emission processes are the same, so they can be simply expressed as:

$$S_{n,n'}(\mathbf{k}, \mathbf{k}') = \frac{2\pi K_B T D^2_{ac}}{\rho A \hbar v_s^2} F_{n,n'} \delta[E_n(\mathbf{k}) - E_{n'}(\mathbf{k}')]. \tag{2.29}$$

Let's consider now the intra-valley optical phonons. Their energy is practically constant ($\hbar\omega_{\nu,\mathbf{Q}} \approx \hbar\omega_0$) so we can compute the phonon number n_{op} using Eq. 2.26 and express the matrix element as:

$$|M^{(ab)}_{n,n'}(\mathbf{k}, \mathbf{k}')|^2 = \delta_{\mathbf{k}',(\mathbf{k}+\mathbf{q})} \frac{\hbar D^2_{op}}{2\omega_0 \rho A} F_{n',n} n_{op} \tag{2.30a}$$

$$|M^{(em)}_{n,n'}(\mathbf{k}, \mathbf{k}')|^2 = \delta_{\mathbf{k}',(\mathbf{k}-\mathbf{q})} \frac{\hbar D^2_{op}}{2\omega_0 \rho A} F_{n',n} (n_{op} + 1) \tag{2.30b}$$

and the scattering rate is:

$$\begin{aligned} S_{n,n'}(\mathbf{k},\mathbf{k}') = \frac{\pi D_{op}^2}{\omega_0 \rho} F_{n',n} n_{op} \delta[E_n(\mathbf{k}) - E_{n'}(\mathbf{k}') + \hbar\omega_0] \\ + \frac{\pi D_{op}^2}{\omega_0 \rho} F_{n',n} (n_{op}+1) \delta[E_n(\mathbf{k}) - E_{n'}(\mathbf{k}') - \hbar\omega_0] \end{aligned} \tag{2.31}$$

where D_{op} is the scalar optical deformation potential. Finally, for the inter-valley transitions, the scattering rate is:

$$\begin{aligned} S_{\nu,n}^{w,n'}(E_{\nu,n}(\mathbf{k})) = \frac{\pi D_{op}^2}{\omega_0} \sum_{w \neq \nu, n'} \mu_{w,v}^{(p)} F_{\nu,n}^{w,n'} \\ \times \Big[n_{op}(\hbar\omega_p) + 0.5 \mp 0.5 \Big] g_{w,n'}(E_{\nu,n}(\mathbf{k}) \pm \hbar\omega_p) \end{aligned} \tag{2.32}$$

where $\hbar\omega_p$ is the phonon energy, D_p is the deformation potential of the p type phonon, $g_{w,n'}(E)$ is the density of states of a w type valley and $\mu_{w,v}^{(p)}$ is the multiplicity of the destination valley. Finally, $F_{\nu,n}^{w,n'}$ is defined as:

$$F_{\nu,n}^{w,n'} = \int_y |\xi_{w,n'}(y)\xi_{\nu,n}(y)|^2 dy. \tag{2.33}$$

2.3.3 Coulomb scattering

Coulomb scattering is caused by a perturbation potential produced by Coulomb centres located in the semiconductor, in the dielectrics and at the interfaces between these materials. Our simulator supports two models for this kind of scattering: the local model, which is used when the the gate dielectric can be assumed to be infinitely thick and single material, and the remote model, which considers a gate stack with a high-k material lying between a metal gate and an interfacial layer. This is an elastic anisotropic scattering mechanism, and the transitions between valleys are unlikely, so we will assume that the initial and final valleys are the same [7].

Let's consider the local model first [18, 19]. Fig. 2.4(a) shows the gate stack, which is made by an infinitely thick oxide lying on top of the semiconductor. In the figure, N_{semi} is the number of Coulomb scattering centres per unit of volume located inside the semiconductor, while $N_{ox/semi}$ is the number of Coulomb scattering centres per unit of area located at the interface ($y = 0$) between the oxide and the semiconductor. In order to compute the matrix element and then the scattering rate, we must first compute the potential produced by a point charge located at $(\mathbf{r}_0, \mathrm{y}_0)$. This potential is given by the Poisson equation:

$$\nabla_{\mathbf{R}}^2 \psi_{pc}(\mathbf{r}, \mathrm{y}) = -\frac{\mathrm{e}}{\epsilon} \delta(\mathbf{r} - \mathbf{r}_0) \delta(\mathrm{y} - \mathrm{y}_0). \tag{2.34}$$

The unknown potential can be expressed as

$$\psi_{pc}(\mathbf{r}, \mathrm{y}) = \int_{\mathbf{q}} \Psi_{\mathrm{pc}}(\mathbf{q}, \mathrm{y}) \exp(-\mathrm{i}\mathbf{q} \cdot \mathbf{r}) \mathrm{d}\mathbf{q}, \tag{2.35}$$

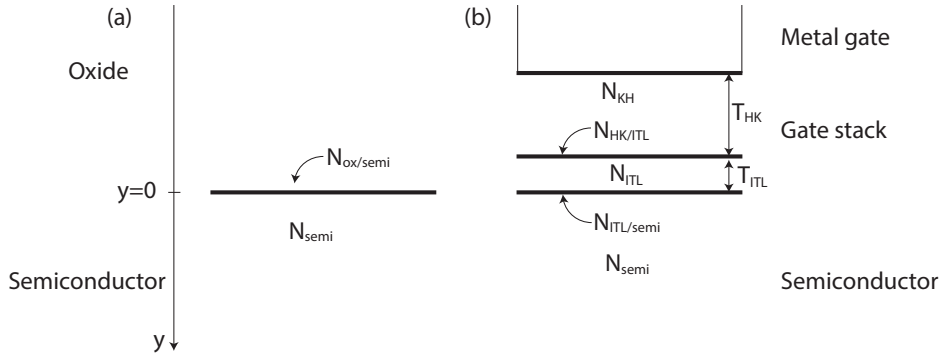


Figure 2.4: Main quantities used to compute the scattering rates for Coulomb scattering. (a) in the local model Coulomb scattering centres are located either inside the semiconductor or at the interface between the semiconductor and the dielectric (which is assumed to be infinitely thick). (b) the remote model allows to consider Coulomb centres located in the semiconductor, in the high-k layer and optionally in an interface layer. Centres can also be located at the interface between each region.

where $\Psi_{pc}(\mathbf{q}, \mathrm{y})$ must take the form

$$\Psi_{pc}(\mathbf{q}, \mathrm{y}) = \frac{\exp(\mathrm{i}\mathbf{q} \cdot \mathbf{r}_0)}{(2\pi)^2} \phi_{\mathrm{pc}}(\mathrm{q}, \mathrm{y}, \mathrm{y}_0). \tag{2.36}$$

For a bulk device, the function $\phi_{pc}(q, y, y_0)$ is [16]:

$$\phi_{pc}(q, y, y_0) = \frac{e}{2q\epsilon_S} \exp(-q|y - y_0|) + \left(\frac{\epsilon_S - \epsilon_{ox}}{\epsilon_S + \epsilon_{ox}}\right) \frac{e}{2q\epsilon_S} \exp(-q|y + y_0|), \tag{2.37}$$

where ϵ_S is the dielectric constant of the semiconductor and ϵ_{ox} is the dielectric constant of the oxide. For an SOI device with semiconductor thickness T_S, $\phi_{pc}(q, y, y_0)$ is [16]:

$$\phi_{pc}(q, y, y_0) = \frac{e}{2q\epsilon_S} \left[\exp(-q|y - y_0|) + C_1 \exp(qz) - C_2 \exp(-qy) \right], \tag{2.38}$$

where the two coefficients are:

$$C_1 = \frac{(\epsilon_S - \epsilon_{ox})^2 \exp(-q|y_0|) + (\epsilon_S^2 - \epsilon_{ox}^2) \exp(-q|T_S - y_0| - T_S)}{(\epsilon_S + \epsilon_{ox})^2 \exp(2qT_s) - (\epsilon_S - \epsilon_{ox})^2} \tag{2.39a}$$

$$C_2 = \frac{(\epsilon_S - \epsilon_{ox})(C_1 + \exp(-q|y_0|))}{\epsilon_S + \epsilon_{ox}}. \tag{2.39b}$$

Now that we have an expression for $\phi_{pc}(q, y, y_0)$, we can finally write the squared modulus of the unscreened matrix element for this scattering mechanism:

$$|M_{n,n'}(\mathbf{q})|^2 = \frac{1}{A} \left[\int_0^{y_{max}} |M^{(0)}_{n,n'}(\mathbf{q}, y_0)|^2 N_{semi}(y_0) dy_0 + |M^{(0)}_{n,n'}(\mathbf{q}, y_0)|^2 N_{ox/semi} \right] \tag{2.40a}$$

$$M^{(0)}_{n,n'}(\mathbf{q}, y_0) = \int_y \xi_{n'}(y) \xi_n(y) \phi_{pc}(q, y, y_0) dy. \tag{2.40b}$$

The equation above holds for a bulk device. For an SOI device the upper limit of the integral is replaced with T_S. The squared modulus of the screened matrix elements is instead given by:

$$|M_{w,n,n'}(\mathbf{q})|^2 = \frac{1}{A}\left[\int_0^{y_{max}} |M^{(0,scr)}_{w,n,n'}(\mathbf{q},y_0)|^2 N_{semi}(y_0)dy_0 + |M^{(0,scr)}_{w,n,n'}(\mathbf{q},y_0)|^2 N_{ox/semi}\right], \tag{2.41}$$

where $M^{(0,scr)}_{w,n,n'}(\mathbf{q},y_0)$ can be found either by solving this linear system (for tensorial screening):

$$M^{(0)}_{w,n,n'}(\mathbf{q},y_0) = \sum_{w,n,n'} \epsilon^{w,n,n'}_{\nu,m,m'}(\mathbf{q}) M^{(0,scr)}_{w,n,n'}(\mathbf{q},y_0) \tag{2.42}$$

or, for scalar screening and intra-subband transitions, simply by:

$$M^{(0,scr)}_{w,n,n'}(\mathbf{q},y_0) = \frac{M^{(0)}_{w,n,n'}(\mathbf{q},y_0)}{\epsilon_D(\mathbf{q})}. \tag{2.43}$$

For inter-subband transitions and scalar screening the screened matrix element is the same as the unscreened one. In all cases $M^{(0)}_{w,n,n'}(\mathbf{q},y_0)$ is given by Eq. 2.40b

Let's consider now the remote model [20]. This model allows the simulation of a finite high-k dielectric with the optional presence of an interfacial layer. Now, the Coulomb scattering centres can be located inside the high-k dielectric, inside the interfacial layer and inside the semiconductor. Their concentrations per unit of volume are given respectively by N_{KH}, N_{ITL} and N_{semi} (See Fig. 2.4(b)). Coulomb centres can also be located at the interface between the high-k and the interfacial layer and between the interfacial layer and the semiconductor. Their concentration per unit of area are given respectively by $N_{HK/ITL}$ and $N_{ITL/semi}$. This multi-layered stack gives a set of equations for $\phi_{pc}(q,y,y_0)$, depending on the considered region [16]:

$$\phi_{pc}(q,y,y_0)_{HK} = \frac{e}{2q\epsilon_{HK}}\exp(-q|y-y_0|) + A_1\exp(qy) + A_2\exp(-qy), \tag{2.44a}$$

$$\phi_{pc}(q,y,y_0)_{ITL} = \frac{e}{2q\epsilon_{ITL}}\exp(-q|y-y_0|) + A_3\exp(qy) + A_4\exp(-qy), \tag{2.44b}$$

$$\phi_{pc}(q,y,y_0)_{S} = \frac{e}{2q\epsilon_{S}}\exp(-q|y-y_0|) + A_5\exp(-qy). \tag{2.44c}$$

The coefficient we are most interested in is A_5 since Eq. 2.44c allows us to express the potential inside the semiconductor. All the coefficients can be found by setting five boundary conditions, namely, null potential at the metal gate and the continuity of the potential and the displacement field at both interfaces and then by solving the resulting set of equations. The squared modulus of the matrix elements for the remote model can

be computed as:

$$|M_{n,n'}(\mathbf{q})|^2 = \frac{1}{A}\Bigg[\int_{-T_{HK}-T_{ITL}}^{-T_{ITL}} |M^{(0)}_{n,n'}(q,y_0)|^2 N_{HK}(y_0)dy_0 + \int_{-T_{ITL}}^{0} |M^{(0)}_{n,n'}(q,y_0)|^2 N_{ITL}(y_0)dy_0 + \int_{0}^{y_{max}} |M^{(0)}_{n,n'}(q,y_0)|^2 N_{semi}(y_0)dy_0 + |M^{(0)}_{n,n'}(q,-T_{ITL})|^2 N_{HK/ITL} + |M^{(0)}_{n,n'}(q,0)|^2 N_{ITL/semi}\Bigg] \tag{2.45a}$$

$$M^{(0)}_{n,n'}(q,y_0) = \int_y \xi_{n'}(y)\xi_n(y)\left(\frac{e}{2q\epsilon_S}\exp(-q|y-y_0|) + A_5\exp(-qy)\right)dy. \tag{2.45b}$$

The effects of screening can be included as per the local model. However, Eq. 2.44c replaces Eq. 2.37 in Eq. 2.22.

2.3.4 Surface roughness scattering

In a real device the interface between the semiconductor and the dielectric is not perfectly flat [16, 21]. The position of the interface may vary when moving along the channel. If we assume that the interface lies at $y = 0$, then the quantity $\Delta(\mathbf{r})$ represents the distance between the true interface and the ideal interface measured along the y direction (see Fig. 2.5). This non-flatness of the interface is the origin of another scattering mechanism called surface roughness scattering. This mechanism is anisotropic and elastic. Furthermore, transitions between different valleys are negligible.

The perturbation produced by this non-ideal interface cannot be simply described by a scattering potential. We must use a perturbed hamiltonian $\hat{H}_{p,\mathbf{r}y}$. If $\hat{H}_{0,y}$ is the original unperturbed hamiltonian then the matrix element for this mechanism can be written as [16]

$$M_{n,n'}(\mathbf{q}) = \int_A \left\{\int_y \xi_{n'}(y)[\hat{H}_{p,\mathbf{r}y} - \hat{H}_{0,y}]\xi_n(y)]dy\right\}\frac{exp(-i\mathbf{q}\cdot\mathbf{r})}{A}d\mathbf{r}. \tag{2.46}$$

Since we are employing the parabolic effective mass approximation, the two hamiltonians can be written as:

$$\hat{H}_{0,y} = -\frac{\hbar}{2}\frac{d}{dy}\left(\frac{1}{m_y(y)}\frac{d}{dy}\right) - e\phi(y) + \Phi_B H_v(-y) \tag{2.47a}$$

$$\hat{H}_{p,\mathbf{r}y} = -\frac{\hbar}{2}\frac{d}{dy}\left(\frac{1}{m_y(y-\Delta(\mathbf{r}))}\frac{d}{dy}\right) - e\phi(y) + \Phi_B H_v(-y+\Delta(\mathbf{r})) \tag{2.47b}$$

where $\phi(y)$ is the electrostatic potential, Φ_B is the potential energy barrier between the semiconductor and the dielectric, H_v is the step function and the quantisation mass is the effective mass of the semiconductor, if $y \geq 0$ or the effective mass of the dielectric if

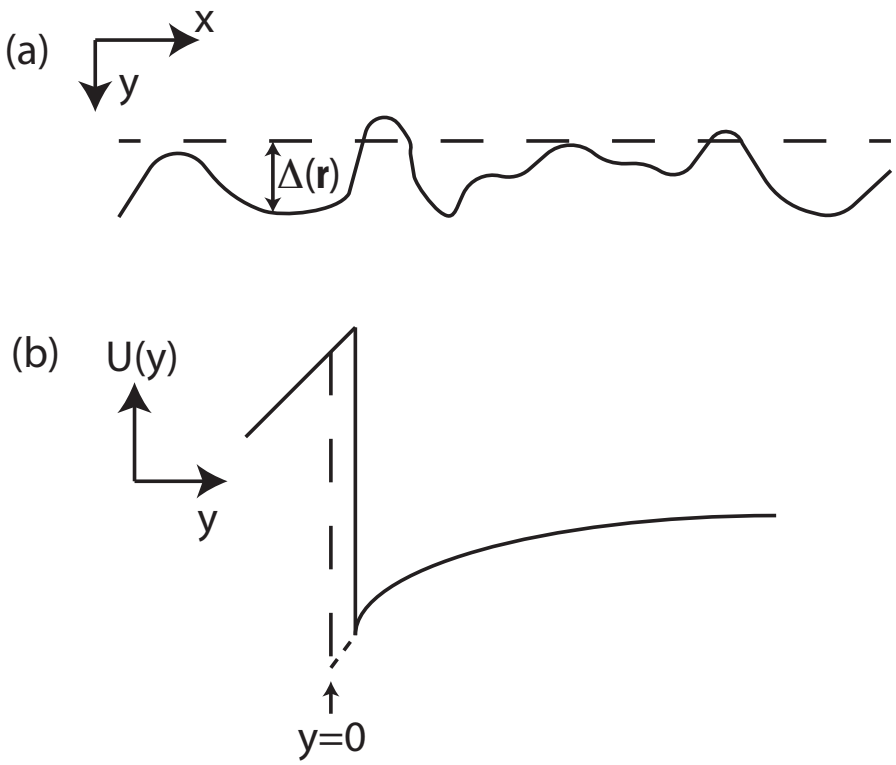


Figure 2.5: (a) The position of the true interface (solid line) can be different from the position of the ideal interface (dashed line). This causes a shift of the potential energy profile along the quantisation direction y (b).

$y < 0$. Some manipulations [22] of these equations allow us to write the unscreened matrix element as:

$$M_{n,n'}(\mathbf{q}) = \Delta(\mathbf{q}) \left[\frac{\hbar^2}{2m_y} \frac{d\xi_n}{dy}(0) \frac{d\xi_{n'}}{dy}(0) \right] \tag{2.48}$$

where

$$\Delta(\mathbf{q}) = \frac{1}{A} \int_A \Delta(\mathbf{r}) \exp(-i\mathbf{q} \cdot \mathbf{r}) d\mathbf{r}. \tag{2.49}$$

The squared modulus of the unscreened matrix element is:

$$|M_{n,n'}(\mathbf{q})|^2 = \left| \frac{\hbar^2}{2m_y} \frac{d\xi_n}{dy}(0) \frac{d\xi_{n'}}{dy}(0) \right|^2 \frac{S_R(\mathbf{q})}{A} \tag{2.50}$$

where S_R is the spectrum of the surface roughness. In literature, two different expressions have been proposed. The first if the gaussian spectrum and the second is the exponential spectrum:

$$S_R(\mathbf{q}) = \pi \Delta_{SR}^2 \lambda_{SR}^2 exp\left(-\frac{q^2 \lambda_{SR}^2}{4} \right), \tag{2.51a}$$

$$S_R(\mathbf{q}) = \frac{\pi \Delta_{SR}^2 \lambda_{SR}^2}{\left[1 + \frac{q^2 \lambda_{SR}^2}{2} \right]}, \tag{2.51b}$$

where Δ_{SR} and λ_{SR} are the r.m.s. value and the correlation length. The squared modulus of the screened matrix element for intra-subband transitions is simply:

$$|M_{w,n,n}^{(scr)}(\mathbf{q})|^2 = \frac{|M_{w,n,n}(\mathbf{q})|^2}{\epsilon_D^2(\mathbf{q})} \tag{2.52}$$

and $|M_{w,n,n}(\mathbf{q})|^2$ is computed using Eq. 2.50.

For SOI devices, we have two interfaces to model, the other one located at $y = T_S$. In this case $\Delta(\mathbf{r})$ is replaced by the two $\Delta_F(\mathbf{r})$ and $\Delta_B(\mathbf{r})$. The unscreened matrix element is:

$$M_{n,n'}(\mathbf{q}) = \Delta_F(\mathbf{q}) \frac{\hbar^2}{2m_y} \frac{d\xi_n}{dy}(0) \frac{d\xi_{n'}}{dy}(0) - \Delta_B(\mathbf{q}) \frac{\hbar^2}{2m_y} \frac{d\xi_n}{dy}(T_S) \frac{d\xi_{n'}}{dy}(T_S) \tag{2.53}$$

where Δ_F and Δ_B are given by Eq. 2.49. The corresponding squared modulus of the unscreened matrix element is written as:

$$|M_{n,n'}(\mathbf{q})|^2 = \frac{S_R^F(\mathbf{q})}{A} \left| \frac{\hbar^2}{2m_y} \frac{d\xi_n}{dy}(0) \frac{d\xi_{n'}}{dy}(0) \right|^2 + \frac{S_R^B(\mathbf{q})}{A} \left| \frac{\hbar^2}{2m_y} \frac{d\xi_n}{dy}(T_S) \frac{d\xi_{n'}}{dy}(T_S) \right|^2 \tag{2.54}$$

where S_R^F and S_R^B are computed for the front and back interfaces respectively. Roughnesses of the two interfaces are assumed to be uncorrelated. The inclusion of screening effects is slightly more involved than the bulk case. The squared modulus of the screened matrix element is obtained by summing the squared moduli of the matrix elements at both interfaces (again, we assume that the roughnesses of the two interfaces are uncorrelated):

$$|M_{w,n,n'}^{(scr)}(\mathbf{q})|^2 = |M_{w,n,n'}^{(F,scr)}(\mathbf{q})|^2 + |M_{w,n,n'}^{(B,scr)}(\mathbf{q})|^2 \tag{2.55}$$

The screened matrix element of the single interfaces are obtained by solving Eq. 2.18.

2.3.5 Alloy scattering

Semiconductors that are alloys made of two different semiconductors (like SiGe and InGaAs) present an additional scattering mechanism. This kind of scattering is caused by the random distribution of component atoms among the available lattice sites. The relative amount of each semiconductor is given by a parameter x called molar fraction, which is a number between 0 and 1. As an example, if x is 0.53, then in the alloy $In_xGa_{1-x}As$, 53% is InGas and 47% is GaAs.

Alloy scattering is an anisotropic elastic scattering mechanism. Transitions between valleys are negligible. The squared modulus of the unscreened matrix element is simply given by [23]:

$$|M_{n,n'}|^2 = \frac{\Omega_C}{A}\Delta U^2 x(1-x)\int_y |\xi_{n'}(y)|^2|\xi_n(y)|^2 dy \tag{2.56}$$

where Ω_C is the volume of the unit cell and ΔU is the difference between the electron affinites of the two semiconductors that make up the alloy. Note that the expression above does not depend on $\mathbf{q}$. The anisotropy comes into play when considering the effects of screening [24]:

$$|M^{(scr)}_{\nu,n,n'}|^2 = \frac{\Omega_C}{A}\Delta U^2 x(1-x)\int |\mathcal{M}_{\nu,n,n'}(y)|^2 dy \tag{2.57}$$

where $\mathcal{M}$ is obtained by solving:

$$\xi_{w,m}(y)\xi_{w,m'}(y) = \sum_{\nu,n,n'} \epsilon^{\nu,n,n'}_{w,m,m'}(q)\mathcal{M}_{\nu,n,n'}(y). \tag{2.58}$$

2.3.6 Polar Optical Phonons scattering

This kind of scattering dominates the phonon assisted transitions in GaAs and other III-V materials. It is caused by the polar nature of the bonding between Ga and As atoms. It is an anisotropic inelastic scattering mechanism. Again, transitions between valleys are negligible. The scattering potential for this mechanism is given by [2]:

$$U^{POP}_{ph}(\mathbf{R},t) = \frac{e\sqrt{\hbar\omega_{ph}}}{i\sqrt{2}\sqrt{\Omega}Q}\sqrt{\frac{1}{\epsilon_\infty}-\frac{1}{\epsilon_0}}\Big[a\exp(i(\mathbf{Q}\cdot\mathbf{R}-\omega_{ph}t)) + a^\dagger\exp(-i(\mathbf{Q}\cdot\mathbf{R}-\omega_{ph}t))\Big] \tag{2.59}$$

where ω_{ph} is the phonon energy, ϵ_∞ is the high-frequency dielectric constant, ϵ_0 is the static dielectric constant, Ω is a normalising volume and Q is the magnitude of the phonon wave-vector $\mathbf{Q}$. $U^{POP}_{ph}(\mathbf{R},t)$ is real since we can assume that $|a|{=}|a^\dagger|{=}\sqrt{n_{ph}+1}$. However, when computing the scattering rates, $|a|$ is set to $\sqrt{n_{ph}}$ (phonon absorption) and $|a^\dagger| = \sqrt{n_{ph}+1}$ (phonon emission). The matrix element can be written as:

$$\begin{aligned} M_{n,n'}(\mathbf{q},q_y) &= \int_{\mathbf{r}} d\mathbf{r}\int_y dy \frac{e\sqrt{\hbar\omega_{ph}}}{i\sqrt{2}\sqrt{Q}\sqrt{q^2+q_y^2}}\sqrt{\frac{1}{\epsilon_\infty}-\frac{1}{\epsilon_0}} \\ \times\sqrt{n_{ph}+\frac{1}{2}\pm\frac{1}{2}}\exp(\mp iq_y)\exp(\mp\mathbf{q}\cdot\mathbf{r})&\frac{\exp(-i\mathbf{k}'\cdot\mathbf{r})}{\sqrt{A}}\xi_{n'}\frac{\exp(-i\mathbf{k}\cdot\mathbf{r})}{\sqrt{A}}\xi_n. \end{aligned} \tag{2.60}$$

where q_y is the component in the y direction of the phonon wave-vector $\mathbf{Q}$ and the upper and lower sign are for emission or absorption, respectively. Some manipulations [16] allow

us to rewrite the matrix element into a form which is easier to evaluate:

$$|M_{n,n'}(q)|^2 = \frac{e^2\hbar\omega_{ph}}{4Aq}\left(\frac{1}{\epsilon_\infty}-\frac{1}{\epsilon_0}\right)\left(n_{ph}+\frac{1}{2}\pm\frac{1}{2}\right) \times \int_y dz \int_{z'} dz' \xi_{n'}(y)\xi_n(y)\xi_{n'}(y')\xi_n(y')\exp(-q|y-y'|). \tag{2.61}$$

The effects of screening is weak and is not considered [2][25].

2.3.7 Remote Phonons scattering

The molecules of high-k dielectrics are strongly polarised and their thermal vibration generates non-stationary electric fields which affect the semiconductor channel. These phonons are also called *soft* because the bonds with oxygen molecules are soft and the molecules can vibrate strongly. The dipoles that make up the insulator can rotate around their centroid. If the field is stationary, these dipole align against the field and their filed counteracts the effects of the external field. If the field oscillates at high frequency, the dipoles can't rotate fast enough and their field is negligible. In any case, the semiconductor will be affected by both the external and the dipole field. For simplicity, let's consider a bulk semiconductor with an infinitely thick dielectric on top and one phonon mode ω_{TO}. If we find the dispersion relationship ω_{SO} as [26]:

$$\omega_{SO} = \omega_{TO}\sqrt{\frac{\epsilon_S+\epsilon_0}{\epsilon_S+\epsilon_\infty}} \tag{2.62}$$

the matrix element for phonon emission can be written as [27, 20]:

$$M^{(em)}_{n,n'}(\mathbf{k},\mathbf{k}') = \sqrt{\frac{\hbar\omega_{SO}}{2qA}\frac{1}{\hat{\epsilon}}}a^\dagger_{SO}\int \xi_n\xi_{n'}\exp(-qy)dy\delta_{\mathbf{k}',(\mathbf{k}-\mathbf{q})}. \tag{2.63}$$

where $|a^\dagger_{SO}|^2 = n_{SO}+1$ and

$$\frac{1}{\hat{\epsilon}} = \frac{1}{\epsilon_S+\epsilon_\infty}-\frac{1}{\epsilon_S+\epsilon_0}. \tag{2.64}$$

ϵ_S is the semiconductor permittivity and ϵ_0 and ϵ_∞ are the static and high frequency permittivities of the dielectric. The absorption matrix element can be written by replacing $a^\dagger_{SO}$ with a_{SO} ($|a_{SO}|^2 = n_{SO}$) and $\delta_{\mathbf{k}',(\mathbf{k}-\mathbf{q})}$ with $\delta_{\mathbf{k}',(\mathbf{k}+\mathbf{q})}$. The effect of screening is weak and is not take into account [7].

2.3.8 From the matrix elements to the scattering rates integrals

The scattering rates computation step is divided into two sub-steps. During the first we compute the matrix elements, during the second we use the matrix elements to compute the scattering rates and we integrate them. From a computational point of view, this poses a challenge because both $\mathbf{k}$ and $\mathbf{k}'$ vectors belong to $\mathbb{R}^2$ (which is contiguous) and so is the vector $\mathbf{q}$, which is used to compute the matrix elements. In order to make the simulation feasible, during the first sub-step, we sample a set of $|\mathbf{q}|$ values and compute the matrix elements only for these values of $|\mathbf{q}|$. Therefore, in each section and for each mechanism

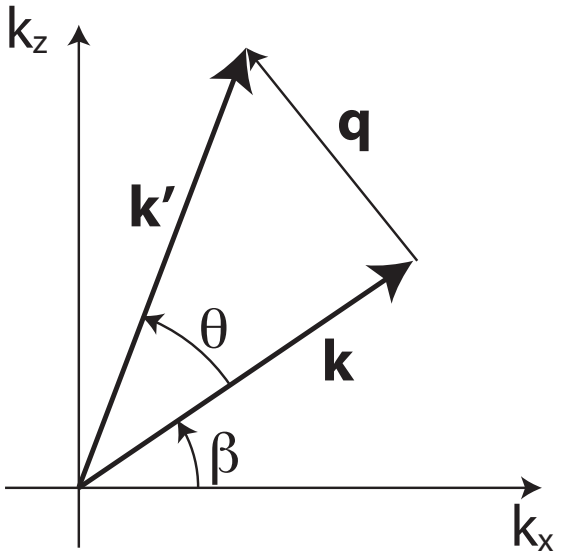


Figure 2.6: When using polar coordinates, the vector $\mathbf{k}$ forms an angle β with the k_x axis. The vector $\mathbf{q}$ represents the difference between $\mathbf{k}'$ and $\mathbf{k}$, difference that consists in an increment θ of the angle and, possibly, of the magnitude of $\mathbf{k}'$. The coordinate system is the in-plane ellipse coordinate system EpCS (see [16], chapter 8).

we must compute a matrix element for each possible combination of source valley, source subband, destination valley, destination subband and discretised $|\mathbf{q}|$ value. The maximum $|\mathbf{q}|$ is chosen so that the corresponding contribution to the integral is negligible.

During the second sub-step, the scattering rates are computed and integrated. To make this sub-step easier, we express the $\mathbf{k}$ vector using polar coordinates, so we have $\mathbf{k} = (k_x, k_z) = (k\cos(\beta), k\sin(\beta))$. The angle β is the angle between $\mathbf{k}$ and the k_x axis. The angle θ is used to represent the difference between $\mathbf{k}'$ and $\mathbf{k}$. Figure 2.6 show the relationship between $\mathbf{k}$, $\mathbf{k}'$, $\mathbf{q}$, β and θ.

k, β and θ must be discretised in order to perform a numeric integration. Since we can compute the kinetic energy from $\mathbf{k}$, the discretisation of the magnitude of $\mathbf{k}$ is a discretisation in terms of energy. Therefore $\mathbf{k}$ with similar k are represented with an energy bin E_i. Then, all the vectors that falls into the same energy bin must be further discretised according to a β-bin β_j. Due to symmetry reasons, we can restrict the β angles to the interval $[0, \pi/2]$. The triplet (n, i, j) allows us to identify every possible discretised initial state. The angle θ is discretised by dividing the interval $[0, 2\pi]$ into θ-bins θ_k. The integration procedure for a mechanism m works as follows:

```
FOR n ∈ subbands
   FOR i ∈ E-bins
      FOR j ∈ β-bins
         S^m_(n,k) ← 0
         FOR n' ∈ subbands
            IF transition from n to n' is allowed THEN
               FOR k ∈ θ-bins
                  compute the transition rate from (n, k) to (n', k')
                  using the corresponding matrix element
                  and add it to S^m_(n,k)
               END FOR
            END IF
         END FOR
    END FOR
END FOR
```

The $\mathbf{k}$ and $\mathbf{k}'$ vector used inside the procedure are computed using the discretised energy and angle bins:

$$k = \frac{\sqrt{2m_{xz}(\beta_j)[E_i - E_n^P + \alpha_n(E_i - U_n)^2]}}{\hbar} \tag{2.65a}$$

$$k' = \frac{\sqrt{2m_{xz}(\beta_j + \theta_k)[E_i + \delta - E_{n'}^P + \alpha_{n'}(E_i + \delta - U_{n'})^2]}}{\hbar} \tag{2.65b}$$

where δ accounts for a change in the final energy and the mass $m_{xz}(\beta)$ for a generic angle is:

$$m_{xz}^{-1}(\beta) = \left[\frac{\cos^2(\beta)}{m_x} + \frac{sin^2(\beta)}{m_z}\right]. \tag{2.66}$$

The magnitude of the $\mathbf{q}$ vector for such $\mathbf{k}$ and $\mathbf{k}'$ is:

$$q = \sqrt{k^2 + k'^2 - 2kk'\cos(\theta_k)}. \tag{2.67}$$

The computation of the transition rate from one state to another may require q values for which the corresponding matrix element is unknown. In such case, we must interpolate two known matrix element $M(q_a)$ and $M(q_b)$, provided that $q_a \leq q \leq q_b$ holds.

2.4 Monte Carlo transport core

The purpose of this step is to build an occupation function $f_{x,\nu,n}(\mathbf{k})$ that gives the occupation probability of a state identified by the section x, valley ν, subband n and in-plane wave-vector $\mathbf{k}$. This function is built by simulating the motion of a set of particles through the device. As said before, the particles are moved for a given amount of time steps. Also, all particles must complete the time step i before time step $i+1$ can begin, thus making our Monte Carlo an *ensemble* Monte Carlo [10]. Figure 2.7 gives a quick glance of how this step works. The method relies on the generation of random numbers r_n which are, unless otherwise stated, uniformly distributed between 0 and 1.

2.4.1 Duration of the free flight

First, we need to determine the duration of the free flight t_{FF}. In section 2.3 we have computed the scattering rate from a state $n, \mathbf{k}$ to a state $n', \mathbf{k}'$ due to a mechanism m. The total scattering rate out of a state $n, \mathbf{k}$ is given by:

$$S_{tot}(n, \mathbf{k}) = \sum_m \sum_{n'\mathbf{k}'} S^m_{n,n'}(\mathbf{k}, \mathbf{k}'). \tag{2.68}$$

From this we can compute the probability density for the free flight duration to be t_{FF} [1], that is, the probability to have a scattering event after t_{FF} seconds provided that no scattering event occurred during the time interval $(0, t_{FF})$:

$$P(t_{FF}) = S_{tot}(n, \mathbf{k}(t_{FF})) \exp(-\int_0^{t_{FF}} S_{tot}(n, \mathbf{k}(t'))dt'). \tag{2.69}$$

This is an integral equation very difficult to solve. A simpler approach involves the addition of a fake *self-scattering* mechanism. If S_{tot} in Eq. 2.69 is replaced by its upper bound

$$\Gamma = \max\ S_{tot}(n, \mathbf{k}) \tag{2.70}$$

we then have [1]

$$t_{FF} = -\frac{\ln\ r_1}{\Gamma}. \tag{2.71}$$

2.4.2 Simulation of the free flight

Next we compute the free flight and it will change both the position and the momentum of the particle. Momentum $\mathbf{k}$ will change by an amount

$$\Delta\mathbf{k} = (\Delta k_x, \Delta k_z) = \left(-\frac{F\cos(\alpha)}{\hbar} t_{FF}, -\frac{F\sin(\alpha)}{\hbar} t_{FF} \right), \tag{2.72}$$

while the position will change by an amount

$$\Delta x = \hbar \left[\frac{k_x cos(\alpha)}{m_x} - \frac{k_z sin(\alpha)}{m_z} \right] n_p\ t_{FF} - \frac{1}{2} \left[\frac{F\cos^2(\alpha)}{m_x} + \frac{F\sin^2(\alpha)}{m_z} \right] t^2_{FF}, \tag{2.73}$$

where α is the angle described in section 2.2.1, F is the driving force and n_p is the non-parabolicity correction factor given by

$$n_p = \frac{1}{1 + 2\alpha_\nu \big(E_{\nu,n}(\mathbf{k}) - U_{\nu,n} \big)}. \tag{2.74}$$

The driving force applied to a particle belonging to valley ν and subband n is computed as:

$$F = \frac{dE^{NP}_{\nu,n}}{dx}. \tag{2.75}$$

There are two special cases that must be treated appropriately:

1. if t_{FF} would make the particle move beyond the boundary of a time step, the duration of the free flight is interrupted at the boundary of the time step and the free flight will resume during the next time step;

2. if the particle moves beyond the boundary of a section then the free flight is interrupted at the boundary of the section and a new free flight will be computed using the field of the next section.

2.4.3 Determination of the scattering mechanism that interrupted the free flight

At the end of a free flight, a scattering event occurs. If $S_{tot}(\mathbf{k}(t_{FF})) < r_2\Gamma$ then a self-scattering event has interrupted the free flight and the state of the particle is left unchanged and a new free flight must be computed. If $S_{tot}(\mathbf{k}) \ll \Gamma$ then the self-scattering event happens very often which results in many short free flights instead of a one long free flight. This is a downside of the simplification introduced by Eq. 2.70. Otherwise, we must find which *true* mechanism interrupted the free flight. The probability for mechanism m to be responsible for the free flight interruption is given by

$$P_m(\mathbf{k}) = \frac{1}{S_{tot}(n,\mathbf{k})} \sum_{n',\mathbf{k}'} S^m_{n,n'}(\mathbf{k},\mathbf{k}'). \tag{2.76}$$

To select a mechanism, first we generate a random number r_3, then we find a j such that

$$\sum_{m=1}^{j-1} P_m(\mathbf{k}) < r_3 < \sum_{m=1}^{j} P_m(\mathbf{k}). \tag{2.77}$$

2.4.4 Computation of the state after scattering

To determine the state after scattering me must pick a destination valley, a destination subband and the angle θ which allows us to compute $\mathbf{k}'$ from $\mathbf{k}$. In order to find these quantities we need to perform again the integration steps described in Sec. 2.3.8 but this time the procedure is halted when we find a state $(n,\mathbf{k}')$ such that:

$$\sum_{n,\mathbf{k}'} S^m_{n,n'}(\mathbf{k},\mathbf{k}') = r_4 S_{tot}(n,\mathbf{k}). \tag{2.78}$$

If m is an anisotropic scattering mechanism, then the magnitude of $\mathbf{k}'$ is given by [13]:

$$k' = \frac{\sqrt{2m_{xz}(\beta+\theta)[E_{\nu,n'}(\mathbf{k}') - \varepsilon^P_{\nu,n'} + \alpha_\nu(E_{\nu,n'}(\mathbf{k}') - U_{\nu,n'})^2]}}{\hbar} \tag{2.79}$$

The $\beta+\theta$-dependent mass is given by Eq. 2.66 and the total final energy $E_{\nu,n'}(\mathbf{k}')$ is $E_{\nu,n'}(\mathbf{k}') = E_{\nu,n}(\mathbf{k}) + \delta$. The δ accounts for the change in the total energy provoked by some scattering mechanisms. Remember that not all anisotropic mechanisms are elastic.

If m is a phonon scattering, the angle θ is chosen randomly from $[0, 2\pi]$ and the magnitude of $\mathbf{k}'$ is given by:

$$k' = \frac{\sqrt{2m_d[E_{w,n'}(\mathbf{k}') - \varepsilon^P_{w,n'} + \alpha_w(E_{w,n'}(\mathbf{k}') - U_{w,n'})^2]}}{\hbar}, \tag{2.80}$$

where $m_d = \sqrt{m^w_x m^w_z}$ and m^w is the effective mass of the destination valley w along the x or z direction. The components of $\mathbf{k}'$ are stretched according to the effective mass along the two directions of the transport plane:

$$\mathbf{k}' = (k'_x, k'_z) = (k'\cos\theta\sqrt{m^w_x/m_d}, k'\sin\theta\sqrt{m^w_z/m_d}) \tag{2.81}$$

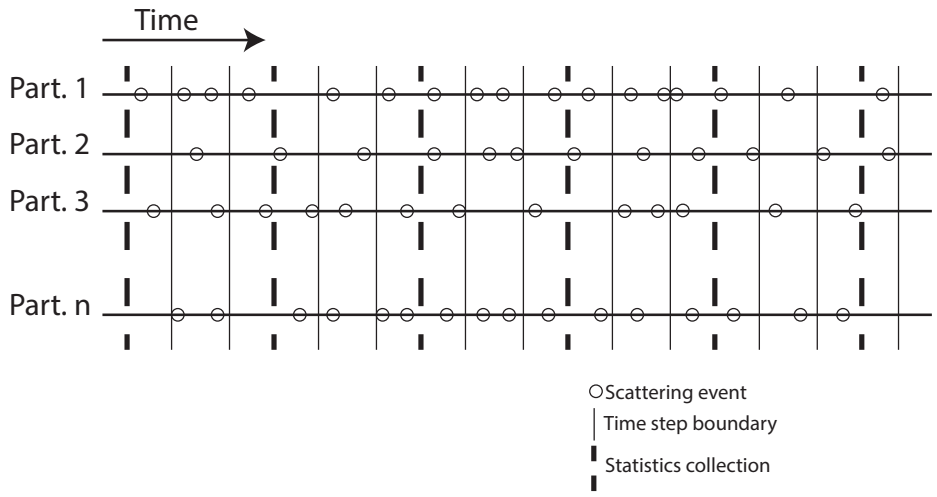


Figure 2.7: Time evolution of an ensemble Monte Carlo. Between two time steps a particle may fly freely and scatter many times. Statistics are collected periodically.

The total final energy $E_{w,n'}(\mathbf{k}')$ is $E_{w,n'}(\mathbf{k}') = E_{\nu,n}(\mathbf{k}) \pm \hbar\omega$.

We must also take into account the Pauli exclusion principle [28]. A transition to $(n', \mathbf{k}')$ is refused if $f_{x,w,n'}(\mathbf{k}') > r_5$ where r_5 is a random number uniformly distributed between 0 and 1. In this case, the state of the particle is not changed. Again, the vector $\mathbf{k}'$ must be discretised, or we cannot build numerically the function f. This time we discrete the plane (k'_x, k'_z) and assign each $\mathbf{k}'$ to one element of the grid.

2.4.5 Contacts

The last aspect to cover is how to manage device contacts, that is, what to do with particles that reach the boundaries of the device and how to add new particles. Our simulator supports mainly two kind of contacts. The first are called *looping* contacts. Looping contacts are very simple in nature. They consists of two paired contacts. Whenever one particle reaches one contact, it is immediately moved to the other contact. These contacts are used to simulate long channel devices. The other kind of contacts are called *absorbing/injecting* contacts and are used for short channel device. Particles that reach the contacts are removed from the simulation. At the beginning of each time step, both contacts adds new particles into the simulation. The number of injected particles is $W \cdot t \cdot I_{inj}/w$, where W is the width of the device, t is the absolute time of the simulation (simulation begins at $t = 0s$) and w is the statistical weight of a particle. The injection current I_{inj} is given by [29]:

$$I_{inj} = \sum_{\nu,n} \int_0^\infty \frac{\mu_\nu}{\hbar^2\pi^2} \sqrt{2m_X M L_z [E + E^{NP}_{\nu,n} - E^P_{\nu,n} + \alpha_\nu (E + E^{NP}_{\nu,n} - U_{\nu,n})^2]} \cdot \left(\frac{\cos\theta_r \sin\theta_s}{\sqrt{m_x}} + \frac{\sin\theta_r \cos\theta_s}{\sqrt{m_z}} \right) \cdot \frac{1}{1 + \exp\left(\frac{E + E^{NP}_{\nu,n} - E_F}{K_B T}\right)} dE, \tag{2.82}$$

where K_B is the Boltzmann constant, T is the lattice temperature, E_F is the Fermi level at the contact, $\theta_r = -\alpha$, $\theta_s = \arctan(\tan\theta_r \cdot \sqrt{m_z/m_x})$ and α is the angle described in section

2.2.1. Basically, we use the Fermi-Dirac distribution to fill all the states with positive group velocity along the x direction and sum their contribution. The two angles θ_r and θ_1 identify these states. The original expression from [29] has been modified to include the non-parabolicity corrections from [13]. A more detailed description of the injecting contacts is given in [30].

2.5 Solution of the 2D Poisson equation

The final step is to solve the Poisson equation:

$$\nabla\cdot\epsilon\nabla\phi^{(k+1)} = -e\left\{p^{(k)}\exp\left[e(\phi^{(k)}-\phi^{(k+1)}/kT\right]-n^{(k)}\exp\left[e(\phi^{(k+1)}-\phi^{(k)})/kT\right]+N_D-N_A\right\}. \tag{2.83}$$

The equation above is the non-linear Poisson equation. The exponential are needed to "damp" the oscillations of carrier concentrations between two iterations or stability issues will arise. These issues are due to the fact that the charge depends exponentially on the potential and the potential variation depends linearly on the charge variation. The index k refers to the current iteration of the loop in Fig.2.2. The electron concentration $n(x,y)$ is

$$n(x,y) = \frac{2}{A}\sum_{\nu}\sum_{n}\sum_{\mathbf{k}} f_{\nu,n,x}(\mathbf{k})|\psi_{\nu,n,x}(y)|^2 \tag{2.84}$$

where f is the occupation function computed during the previous step and A is the area of the device in the xz plane. The current version of the simulator supports only the transport of electrons in electron inversion layers, so the the hole transport and concentration p is computed via drift-diffusion.

Note that Eq. 2.83 is a non-linear Poisson equation. The driving force used during the Monte Carlo step is computed at the boundaries of each section while the position of a particle in real space is contiguous and is not bound to a specific mesh node. Therefore the grid spacing and the duration of particle motion Δt have a significant impact on the stability and on the accuracy of the method [4, 5]. By using the non-linear equation, we can employ a longer Δt which allows for a better statistics collection and thus the simulation requires fewer iterations to reach convergence. This has an impact on the performances because fewer iterations mean fewer scattering rates computations.

Regarding the boundary conditions, Dirichlet conditions are used for the gate contacts so the potential in this portion of the boundary is given by $\phi = V_{FG} - \Phi_{FG} + \chi_S$ where V_{FG} is the potential applied at the top gate, Φ_{FG} is the work-function of the top metal gate and χ_S is the electron affinity of the semiconductor. A similar condition is applied at the bottom gate contact for SOI devices. Neumann conditions are applied everywhere else, imposing the null derivative of the potential.

Finally, the new potential energy profile is computed from the updated potential as:

$$E_C^{(k+1)}(x,y) = -e\phi^{(k+1)}(x,y) + \chi_S - \chi(x,y). \tag{2.85}$$

2.6 Determination of the initial conditions

Looking back at the flowchart of the simulator (Fig. 2.2) we can see that there is still one block to discuss. The simulator requires a first guess of the potential energy profile so that

the first solution of the Schrödinger equation can be obtained. The quality of this first guess is very important because a good first guess means that the simulation will converge with fewer iterations. There are two ways to provide a first guess, depending on the kind of device to be simulated.

Initial conditions for long channel devices In a long channel device the charge profile does not change much while moving along the transport direction x, so it is more efficient to find an initial potential profile and keep it frozen during the simulation. To obtain such profile, we need a self-consistent solution of the coupled 1D Schrödinger-Poisson equations. The solution must be self-consistent because, due to the non-local nature of the Schrödinger equation, we cannot write a local relation between the carrier concentration and the electrostatic potential, so the two equations must be solved iteratively until convergence is reached. The Schrödinger equation is solved as described in section 2.2 but again we need a first guess for the potential energy profile. We set the reference energy level to $E_F = 0$. In the semiconductor both the electrostatic potential ($\phi(y)$) and the potential energy profile ($E_C(y)$) are set to 0. In the dielectric, ϕ is set to $V_g + \chi_S - \Phi$ at the gate contact, where V_G is the impressed gate potential, χ_S is the semiconductor affinity and Φ is the gate work-function. The potential ϕ is linear between the gate contact and 0 (the potential in the semiconductor). In the dielectric E_C is computed as $E_C(y) = -\phi(y) - \chi_{ox} + \chi_S$ where χ_{ox} is the dielectric affinity. A similar procedure is needed for the bottom dielectric if the device is an SOI.

To solve the Poisson equation we need to compute the carrier concentrations. The electron concentration $n(y)$ is computed from the solutions of the Schrödinger equation and it is given by:

$$n(y) = \sum_{\nu,n} |\xi_{\nu,n}(y)|^2 \frac{\mu_\nu m_{d,\nu} K_B T}{\pi\hbar^2} \left[ln(1 + \exp\left(\frac{E_F - E_{\nu 0} - \varepsilon_{\nu,n}^{NP}}{K_B T} \right) + 2\alpha_\nu K_B T \mathcal{F}_1 \left(\frac{E_F - E_{\nu 0} - \varepsilon_{\nu,n}^{NP}}{K_B T}) \right) \right]. \tag{2.86}$$

where μ_ν is the multiplicity of the valley ν, $m_{d,\nu} = \sqrt{m_{x,\nu} m_{z,\nu}}$ and $\mathcal{F}_1$ is the Fermi integral of order 1. Holes are not quantised, so their concentration is simply given by:

$$p(y) = N_V \mathcal{F}_{1/2} \left(\frac{E_V(y) - E_{F,p}}{K_B T} \right). \tag{2.87}$$

where N_V is the effective valence band density of states and $\mathcal{F}_{1/2}$ is the Fermi integral of order 1/2. Finally, Poisson equation is solved:

$$\epsilon_S \frac{\partial^2 \phi^{(k)}}{\partial y^2} = q\Big\{ n^{(k)} exp\Big[\frac{q(\phi^{(k)} - \phi^{(k-1)})}{K_B T} \Big] \Big\} - p^{(k)} exp\Big[\frac{q(\phi^{(k-1)} - \phi^{(k)})}{K_B T} \Big] + N_A - N_D \Big\}. \tag{2.88}$$

The Fermi integral of order 1/2 is defined as:

$$\mathcal{F}_{1/2}(\eta) = \frac{2}{\sqrt{\pi}} \int_0^\infty \frac{x^{\frac{1}{2}}}{1 + \exp(x - \eta)} dx \tag{2.89}$$

following the general definition given by [31].

Initial conditions for short channel devices With devices working close to the ballistic limit we cannot simply follow the procedure as for the long channel devices. We need to update the energy potential profile every iteration. To obtain a first guess, we follow a different procedure. First, me must set the Fermi level $E_F(x)$ through the device. $E_F(x)$ is set to 0 in the source and channel regions of the device, while it is set to $-V_{DS}$ in the drain region. Let's assume that the source and drain regions are n-doped and the channel is p-doped. The electron concentration n per unit of volume is given by:

$$n(x,y) = \begin{cases} N_D(x,y), & x < 0, 0 \leq y \leq T_S \\ N_{INV}/T_S, & 0 \leq x \leq L_{ch}, 0 \leq y \leq T_S \\ N_D(x,y), & x > L_{ch}, 0 \leq y \leq T_S \end{cases} \tag{2.90}$$

where N_D is the n-type doping concentration of the source and drain regions, N_{INV} is a user-supplied estimate of the free carrier density (per unit of area), T_S is the thickness of the semiconductor and L_{ch} is the length of the channel. Here we assume that the interface between the dielectric and the semiconductor lies at $y = 0$ and the interface between the source and the channel region lies at $x = 0$. If we assume a classic 3D carrier distribution we can write in the semiconductor the relationship:

$$n(x,y) = N_C \mathcal{F}_{1/2}\left(\frac{E_F(x) - E_C(x,y)}{K_B T}\right), \tag{2.91}$$

where N_C is the effective conduction band density of states, and solve for $E_C(x,y)$. The potential in the semiconductor is then given by:

$$\phi(x,y) = -E_C(x,y)/e + \chi_S - \chi(x,y) \tag{2.92}$$

where χ_S is the electron affinity of the semiconductor and $\chi(x,y)$ is the electron affinity of the point (x,y). The applied potentials and the potential at the boundary of the semiconductor are used to compute the potential in the dielectrics via linear interpolation. Finally, the potential energy profile in the dielectrics is computed by solving 2.92 for E_C.

Bibliography

[1] Carlo Jacoboni and Lino Reggiani. "The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials". In: *Reviews of Modern Physics* 55 (3 July 1983), pp. 645–705.

[2] Mark Lundstrom. *Fundamentals of Carrier Transport*. Second. Cambridge University Press, 2000.

[3] F.M. Bufler, A. Schenk, and Wolfgang Fichtner. "Efficient Monte Carlo device modeling". In: *IEEE Trans. on Electron Devices* 47.10 (Oct. 2000), pp. 1891–1897.

[4] P. Palestri, N. Barin, D. Esseni, and C. Fiegna. "Stability of self-consistent Monte Carlo Simulations: effects of the grid size and of the coupling scheme". In: *IEEE Trans. on Electron Devices* 53.6 (June 2006), pp. 1433–1442.

[5] P. Palestri, N. Barin, D. Esseni, and C. Fiegna. "Revised stability analysis of the nonlinear Poisson scheme in self-consistent Monte Carlo device simulations". In: *IEEE Trans. on Electron Devices* 53.6 (June 2006), pp. 1443–1451.

[6] L. Lucci, P. Palestri, D. Esseni, L. Bergagnini, and L. Selmi. "Multisubband Monte Carlo Study of Transport, Quantization, and Electron-Gas Degeneration in Ultrathin SOI n-MOSFETs". In: *IEEE Trans. on Electron Devices* 54.5 (2007), pp. 1156–1164.

[7] M. V. Fischetti and S. E. Laux. "Monte Carlo study of electron transport in silicon inversion layers". In: *Phys. Rev. B* 48 (4 July 1993), pp. 2244–2274.

[8] Chr. Jungemann, A. Emunds, and W.L. Engl. "Simulation of linear and nonlinear electron transport in homogeneous silicon inversion layers". In: *Solid State Electronics* 36.11 (1993), pp. 1529–1540.

[9] F. Gámiz, J.A. Lopéz-Villanueva, J.B. Roldan, J.E. Carceller, and P. Cartujo. "Monte Carlo simulation of electron transport properties in extremely thin SOI MOSFET's". In: *IEEE Trans. on Electron Devices* 45.5 (May 1998), pp. 1122–1126.

[10] S.C. Williams, K.W. Kim, and W.C. Holton. "Ensemble Monte Carlo study of channel quantization in a 25-nm n-MOSFET". In: *IEEE Trans. on Electron Devices* 47.10 (Oct. 2000), pp. 1864–1872.

[11] T. Ezaki, P. Werner, and M. Hane. "Self-Consistent Quantum Mechanical Monte Carlo MOSFET Device Simulation". In: *Journal of Computational Electronics* 2.2-4 (2003), pp. 97–103. ISSN: 1569-8025.

[12] J Saint-Martin, A Bournel, F Monsef, C Chassat, and P Dollfus. "Multi Sub-band Monte Carlo simulation of an ultra-thin double gate MOSFET with 2D electron gas". In: *Semiconductor Science Technology* 21.4 (2006), p. L29.

[13] Seonghoon Jin, Massimo V. Fischetti, and Tingwei Tang. "Modeling of electron mobility in gated silicon nanowires at room temperature: Surface roughness scattering, dielectric screening, and band nonparabolicity". In: *Journal of Applied Physics* 102.8 (2007), p. 083715.

[14] D. Esseni, F. Conzatti, M. De Michielis, N. Serra, P. Palestri, and L. Selmi. "Semi-classical transport modelling of CMOS transistors with arbitrary crystal orientations and strain engineering". In: *Journal of Computational Electronics* 8.3-4 (2009), pp. 209–224.

[15] Frank Stern and W. E. Howard. "Properties of Semiconductor Surface Inversion Layers in the Electric Quantum Limit". In: *Phys. Rev. B* 163 (3 Nov. 1967), pp. 816–835. DOI: 10.1103/PhysRev.163.816. URL: http://link.aps.org/doi/10.1103/PhysRev.163.816.

[16] D. Esseni, P. Palestri, and L. Selmi. *Nanoscale MOS Transistors*. Cambridge University Press, 2011.

[17] P. Toniutti, D. Esseni, and P. Palestri. "Failure of the Scalar Dielectric Function Approach for the Screening Modeling in Double-Gate SOI MOSFETs and in FinFETs". In: *IEEE Trans. on Electron Devices* 57.11 (Nov. 2010), pp. 3074–3083.

[18] F. Gámiz, J. A. López-Villanueva, J. A. Jiménez-Tejada, I. Melchor, and A. Palma. "A comprehensive model for Coulomb scattering in inversion layers". In: *Journal of Applied Physics* 75.2 (1994), pp. 924–934.

[19] D. Esseni and A. Abramo. "Modeling of electron mobility degradation by remote Coulomb scattering in ultrathin oxide MOSFETs". In: *IEEE Trans. on Electron Devices* 50.7 (July 2003), pp. 1665–1674.

[20] P. Toniutti, P. Palestri, D. Esseni, F. Driussi, M. De Michielis, and L. Selmi. "On the origin of the mobility reduction in n- and p-metal–oxide–semiconductor field effect transistors with hafnium-based/metal gate stacks". In: *Journal of Applied Physics* 112.3 (2012).

[21] D. Esseni. "On the modeling of surface roughness limited mobility in SOI MOSFETs and its correlation to the transistor effective field". In: *IEEE Trans. on Electron Devices* 51.3 (Mar. 2004), pp. 394–401.

[22] D. Lizzit, D. Esseni, P. Palestri, and L. Selmi. "Surface roughness limited mobility modeling in ultra-thin SOI and quantum well III-V MOSFETs". In: *IEEE IEDM Technical Digest*. 2013, pp. 5.2.1–5.2.4.

[23] Anh-Tuan Pham, C. Jungemann, and B. Meinerzhagen. "Physics-Based Modeling of Hole Inversion-Layer Mobility in Strained-SiGe-on-Insulator". In: *IEEE Trans. on Electron Devices* 54.9 (Sept. 2007), pp. 2174–2182.

[24] D. Lizzit, P. Palestri, D. Esseni, A. Revelant, and L. Selmi. "Analysis of the performance of n-Type FinFETs with strained SiGe Channel". In: *IEEE Trans. on Electron Devices* 60.6 (June 2013), pp. 1884–1891.

[25] Peter J. Price. "Polar-optical-mode scattering for an ideal quantum-well heterostructure". In: *Phys. Rev. B* 30 (4 Aug. 1984), pp. 2234–2235.

[26] K. Hess and P. Vogl. "Remote polar phonon scattering in silicon inversion layers". In: *Solid State Communications* 30.12 (1979), pp. 797–799. ISSN: 0038-1098. DOI: http://dx.doi.org/10.1016/0038-1098(79)90051-6. URL: http://www.sciencedirect.com/science/article/pii/0038109879900516.

[27] Massimo V. Fischetti, Deborah A. Neumayer, and Eduard A. Cartier. "Effective electron mobility in Si inversion layers in metal–oxide–semiconductor systems with a high-k insulator: The role of remote phonon scattering". In: *Journal of Applied Physics* 90.9 (2001), pp. 4587–4608.

[28] P. Lugli and D.K. Ferry. "Degeneracy in the ensemble Monte Carlo method for high-field transport in semiconductors". In: *IEEE Trans. on Electron Devices* 32.11 (Nov. 1985), pp. 2431–2437.

[29] M. De Michielis, D. Esseni, and F. Driussi. "Analytical Models for the Insight Into the Use of Alternative Channel Materials in Ballistic nano-MOSFETs". In: *IEEE Trans. on Electron Devices* 54.1 (Jan. 2007), pp. 115–123.

[30] P. Palestri, L. Lucci, S. Dei Tos, D. Esseni, and L. Selmi. "An improved empirical approach to introduce quantization effects in the transport direction in multi-subband Monte Carlo simulations". In: *Semiconductor Science Technology* 25.5 (2010), p. 055011.

[31] J.S. Blakemore. "Approximations for Fermi-Dirac integrals, especially the function $\mathcal{F}_{1/2}$ used to describe electron density in a semiconductor". In: *Solid State Electronics* 25.11 (1982), pp. 1067–1076.

Chapter 3

Improving the performance of the Multi-subband Monte Carlo

Monte Carlo (MC) techniques have been for long time regarded as excessively demanding from a computational point of view and too time consuming for the daily use in the R&D departments of semiconductor industry. Thanks to continuous increase of computing resources at decreasing costs, and to improved algorithms for efficient collection of carrier statistics [1], the use of MC transport simulators is today well accepted for device analysis and design. In fact, MC is a perfectly integrated section of standard TCAD tools [2]. The Multi-subband Monte Carlo method, while it has already demonstrated its ability to enable the understanding of complex nanoscale CMOS device physics, it is computationally heavier than conventional MC models for the 3D carrier gas and it is still mainly an academic research tool with execution times ranging from hours to tens of hours per bias point on single core architectures.

In order to bring it to the same level of acceptance that conventional 3D Monte Carlo has today, a significant reduction of the execution times is mandatory. Code optimisation is one way to achieve this goal but, as will be shown in the following, its benefits are often of modest entity and vary greatly from one simulation to another. The limitations of optimisation are evident when one considers how modern CPUs are evolving nowadays (see Fig. 3.1). Significant CPU performance improvements do not come from a more efficient micro-architecture but from the integration of multiple cores on the same die. Having many cores has a price, however. Top notch CPUs (from Intel [3]) have a maximum TDP (Thermal Design Power) of 130-150W, and an increase in the number of cores corresponds to a decrease of the clock frequency of each core. Thus, unless there are very few processes running on a given CPU, the performances of single-threaded processes are reduced. From these considerations it is clear that, in order to achieve our goal, a massive exploitation of available multi-core architectures must be sought by means of code parallelisation.

The first step of any optimisation task is to find the portions of the code where most time is spent during the program's execution, the so called hot spots, and estimate their relative contribution to the duration of the job. This is typically done using a profiler. To this purpose we used the Intel Vtune Amplifier XE 2015 [5].

The second step typically involves understanding how a change in the input will change the execution time. Based on the description of the simulator given in chapter 2.1 we can identify the dependencies of the four major steps:

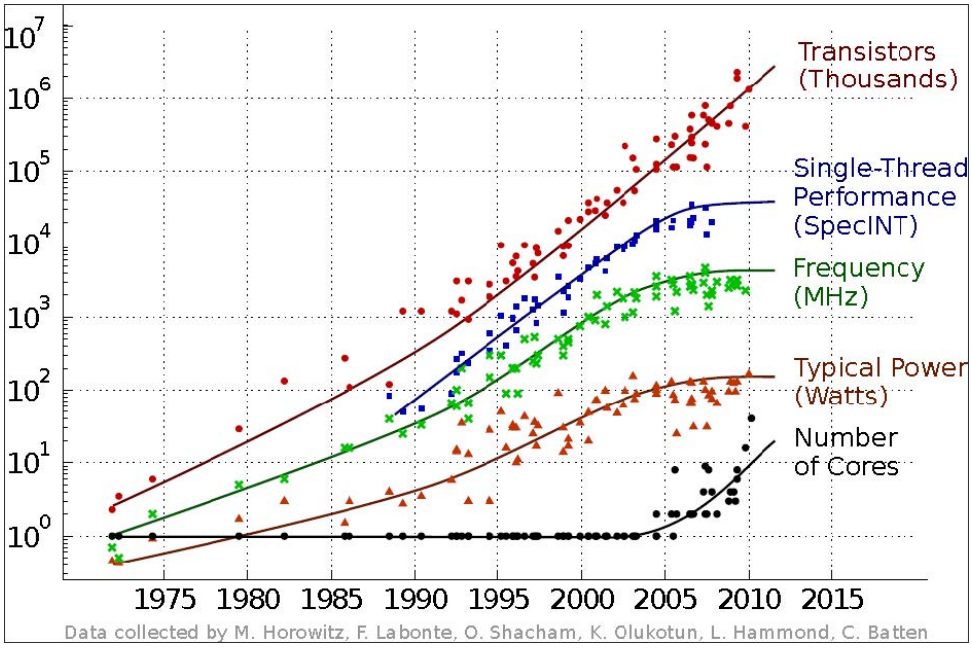


Figure 3.1: CPU evolution trends over the last 35 years [4].

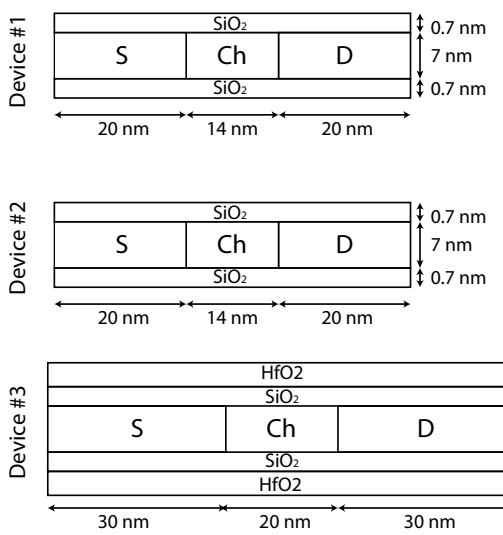
Schrödinger equation (SE) : the equation must be solved for each section, so increasing the number of sections will increase the duration of this step. Also, for each section, the equation must be solved for each valley, totalling for $sections \times valleys$ eigenvalue problem solutions. The time spent on the solution of each equation depends on the number of points of the mesh along the quantisation direction y;

Scattering rates computation (SC) : scattering rates are also computed in each section. In each section, for each mechanism, we must find the transition rate between two states. As explained in Sec. 2.3.8, this requires two steps. During the first we compute the matrix elements. The number of matrix elements to compute is $sections \times valleys \times subbands \times valleys \times subbands \times q - bins$. The time spent on computing one matrix element depends on the number of mesh points along the y direction. Note that some mechanisms forbid inter-valley transitions. During the second step we integrate the matrix elements. The duration of this step roughly goes as $sections \times valleys \times subbands \times energy - bins \times \beta - bins \times valleys \times subbands \times \theta - bins$;

Monte Carlo transport (MC) : the duration of this step depends on the number of particles to simulate, the number of time steps and the scattering rates. Higher scattering rates mean shorter free flights and more state after scattering computations;

Poisson equation (PE) : in this step we just solve a 2D differential equation, so the execution time depends on the number of mesh points.

The amount of operations performed during the first two steps is the same for all iterations, so the execution time of these steps is roughly the same during each iteration. Step 3 is a stochastic step; in principle higher scattering rates will cause longer execution times. Step 4 involves iterative steps to solve the non-linear Poisson equation, so its duration should decrease if the simulation is reaching convergence.



Semiconductor	Si
S/D Doping	10^{20} cm^{-3} n-type
Ch Doping	10^{15} cm^{-3} p-type
V_{DS}	0.5 V
Gate dielectric	0.7 nm SiO_2
Δ valley parms.	m_l=0.93 m_0, m_t=0.19 m_0, α_{np}=0.5 eV^{-1}

Semiconductor	Strained $Si_{0.5}Ge_{0.5}$
S/D Doping	10^{20} cm^{-3} n-type
Ch Doping	10^{15} cm^{-3} p-type
V_{DS}	0.5 V
Gate dielectric	0.7 nm SiO_2
Δ_y valley parms.	m_h=0.94 m_0 $m_\perp$=0.18 m0 , m_q = 0.23 m_0, α_{np}=0.5 eV^{-1}
$\Delta_{x,z}$ valley parms.	m_l=0.93 m_0, m_t=0.19 m_0, α_{np}=0.5 eV^{-1}

Semiconductor	$In_{0.53}Ga_{0.47}As$
S/D Doping	3×10^{19} cm^{-3} n-type
Ch Doping	2×10^{15} cm^{-3} p-type
V_{DS}	0.8 V
Gate dielectric	2 nm HfO_2 + 0.7 nm SiO_2-like (ε_r = 9.34)
Γ valley parms.	m_l= m_t =0.043 m_0, α_{np}=1.36 eV^{-1}

Figure 3.2: Sketch and summary of the devices used to profile the original code and measure the improvements obtained with the new code. Crystal orientation is a FinFET-like (110)/[1$\bar{1}$0].

3.1 Original code analysis

The first step of any optimisation task is to find the regions where most time is spent during the program's execution, the so called hot spots, and estimate their relative contribution to the duration of the job. We have profiled the simulation of three template devices. All of the three are FinFETs simulated as double gate SOI (we assume that the fin is high enough to make negligible the effect of the third gate) The first two are taken from [6] and were already analyzed, from a simulator performance point of view, in [7]. Figure 3.2 shows the sketches and the main features of these devices. Figure 3.3 shows the breakdown of the simulation time for the three devices using the original version of our simulator. The execution time is dominated by the MC step, followed by the SC step. Device #2 requires us to consider the alloy scattering mechanism, which increases the contribution of the SC step to the total execution time with respect to device #1. For device #3, the contribution of the SC step is further increased due to the need of computing matrix elements for the Remote Phonons and Polar Optical Phonons scattering mechanism. Computing the latter requires a huge amount of time, but for this device we consider only the Γ valley and this cuts the execution time by 3 (remember that devices #1 and #2 both have 3 Δ valleys to consider. Also, since $m_x = m_z$ and due to symmetry reasons, only one β-bin is needed (see section 2.3.8) during the matrix elements integration sub-step. Table 3.1 shows the average time required to complete one iteration for the three devices. It also shows the average time required to execute the two heavies steps of one iteration. All the optimisation work is focused on these two steps.

Device	Total time (s)	SC time (s)	MC time (s)
1	3903.9	737.5	3147.5
2	4212.3	993.4	3185.3
3	7279.4	2273.1	5000.4

Table 3.1: Average time required to complete one iteration for each device.

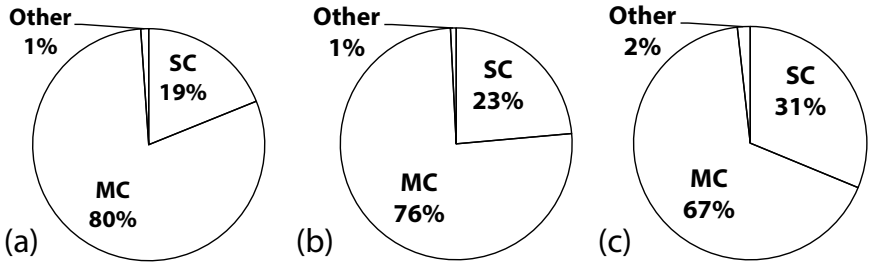


Figure 3.3: Relative simulation time breakdown obtained by profiling the original code while simulating device #1 (a), device #2 (b) and device #3 (c).

3.2 Optimisation

The analysis of the original code identified many sections of the original MSMC code where improvements were possible. Among these improvements, three are worthy of notice.

3.2.1 Optimisation of the occupation function

In section 2.4.4 we have shown that the final state can be rejected if it is too populated. This rejection is needed to take properly into account the Pauli's exclusion principle. The decision whether to accept or to reject the final state is based on the value of the occupation function f build during the MC step. Each particle contributes to the occupation function according to the device section ("x") where the particle is located, its valley (conduction band minimum), its subband (eigenvalues of the Schrödinger's equation) and its wave-vector in the transport plane ($\mathbf{k}$). This function is continuous in the $\mathbf{k}$ space, so this space must be discretised. Each particle will be assigned to a k space bin and many particles may share the same space.

Original code In the original code, a five-level tree is used to record the occupation of the electron states (Fig. 3.4a). Trees are very sparse data structures and the sparseness enforces improper memory access patterns. These patterns have a profound influence on the performance of an application. CPUs use the cache memory to reduce the costs of accessing the main memory, so the data structures must be designed in order to exploit the time and space locality principles [8].

Linearisation of the f data structure We converted the tree into an array with a fixed size record-like structure (Fig. 3.4b), thus achieving a more cache-friendly processing. By construction, each section has the same number of valleys but each valley can have a different number of subbands. The discretisation of the wave-vector requires the same

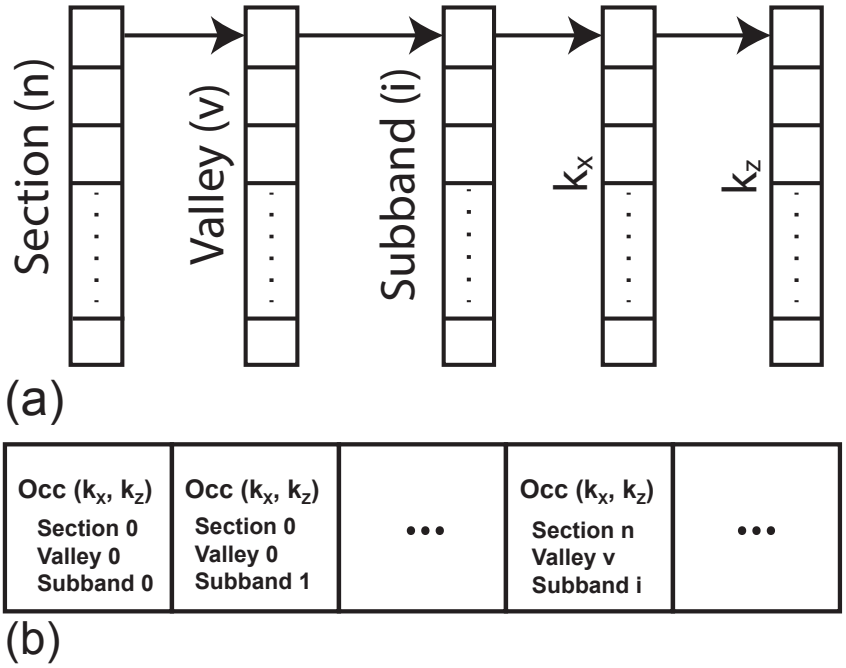


Figure 3.4: Linearisation of the branched structure of the electron states occupation. (a) represents the original tree-shaped data structure. The square matrices in (b) (one for each subband) represent the occupation in the (k_x, k_z) plane.

amount of elements for each subband. If we assume that each valley has a number of subbands equal to the maximum among all the valleys, each element can be located by performing very simple math. The same idea applies to other branched structures, such as the ones containing the scattering rates. There is, however, a price to pay. The linearisation induces some memory waste (depending on specific simulation parameters), waste partly balanced by the removal of the internal nodes of the trees, which accounted for about 1% of the tree memory occupation. At the beginning of each time step, this data structure must be cleared, so a new occupation function can be computed.

Since particles will cluster around low **k** values (as shown in Fig. 3.5), it is inefficient to clear all this data structure so a different approach must be used.

Clearing the data structure: naïve approach The simplest approach is to simply clear the whole array every time. It is a very cache friendly solution and compilers provide special and quite efficient functions for zeroing contiguous memory locations. However, particles tend to cluster around low **k** values (as shown in Fig. 3.5), so a significant portion of the array is always untouched. This unnecessary clearing is not only expensive, but it increases the amount of physical memory required to store the array due to how the operating system manages the virtual memory. All memory pages allocated with `calloc()` will point to a special zeroed memory page. When the program tries to write an address that point to this special page, the page is duplicated (a mechanism known as copy-on-write). Clearing unnecessary data can make the program to use more memory than what is strictly needed.

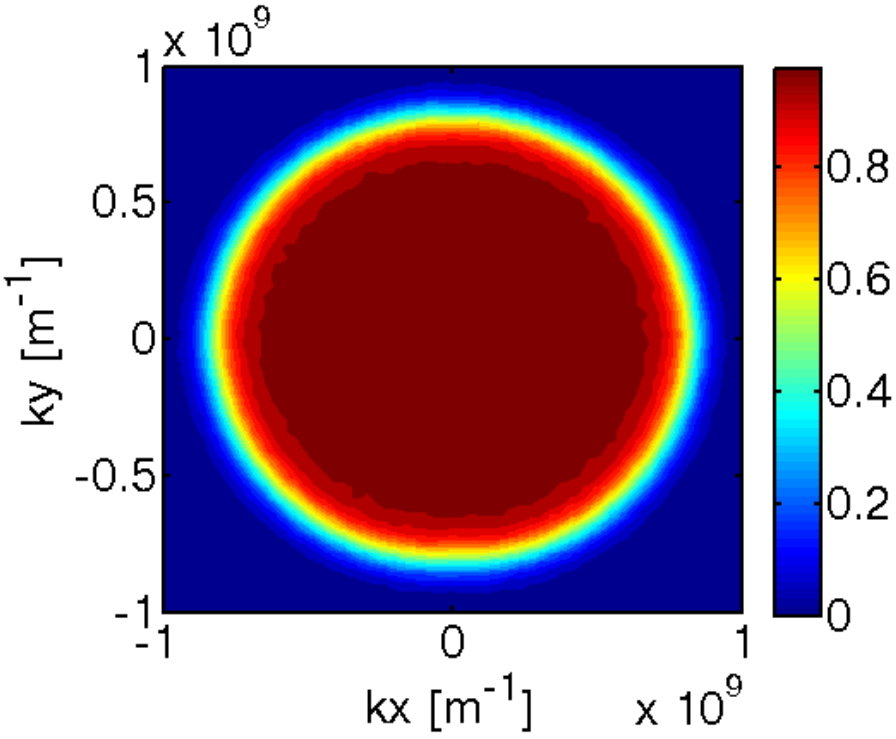


Figure 3.5: Occupation values of a sample occupation function f.

A small improvement of this scheme requires to keep two indexes pointing to first and last changed elements of the array. Only the elements between these two indexes are cleared, thus reducing the number of elements to be cleared. A further improvement would require us to store the elements that are likely to be written towards the middle of the array, but this option was not explored.

Clearing the data structure: chosen approach. The chosen approach uses an auxiliary array containing the indexes of the changed elements of the data structure. This method guarantees that only the changed elements will be cleared, albeit it is possible that an element will be cleared more than once, if more particle share the same state. The drawback is that the clearing function will jump around the array many times, thus generating some cache misses. We conclude that the time saving comes from the fact that now we access only the strictly necessary amount of elements and each element can be accessed directly instead of following a long chain of pointers. Profiler data showed that this approach can be up to three time faster than the naïve approach.

3.2.2 Optimizing the determination of the state after scattering

The second optimisation regards both the SC and MC steps (steps 2 and 3 in Fig. 2.2) when dealing with anisotropic scattering mechanisms. In section 2.3 we have shown how to compute the transition rate from a state $(n, \mathbf{k})$ to a state $(n, \mathbf{k}')$ due to a mechanism m. We have also shown in section 2.3.8 how to compute the total transition rate out of a state $(n, \mathbf{k})$ due to all considered scattering mechanisms. In section 2.4 we have shown,

in particular, how to compute the duration of free flight from the total transition rate (Eq. 2.71) and how to find the state after scattering. Section 2.3.8 introduced the main discretisation involved in the computation of the matrix elements and in their integration to compute the scattering rates. The discretisation of the vector $\mathbf{q}$ results in a set of $\mathbf{q}_i$ elements arranged to a geometric progression whose i-th element is:

$$\mathbf{q}_i = \mathbf{q}_{min} \sum_{j=0}^{i} \mathbf{q}_r^j = \mathbf{q}_{min} \frac{1 - \mathbf{q}_r^{i+1}}{1 - \mathbf{q}_r} \tag{3.1}$$

where $\mathbf{q}_{min}$ is the minimum change of the wave-vector $\mathbf{k}$ and $\mathbf{q}_r$ is the constant ratio of two consecutive elements of the series. A matrix element is computed only for these $\mathbf{q}_i$.

The discretisation of the $\mathbf{k}$ vector and of the change angle θ requires us to integrate matrix elements computed for $\mathbf{q}$ vectors that were not generated from Eq. 3.1. Therefore we need to find and index i such that $\mathbf{q}_i \leq \mathbf{q} \leq \mathbf{q}_{i+1}$. The original code performed a binary search on the set of $\mathbf{q}_i$, which is correct for an unknown but ordered set, but this approach does not exploit the relationship between the elements of the set. A more efficient approach requires us to solve for i the following inequality:

$$\mathbf{q}_{min} \frac{1 - \mathbf{q}_r^{i}}{1 - \mathbf{q}_r} \leq \mathbf{q} \leq \mathbf{q}_{min} \frac{1 - \mathbf{q}_r^{i+1}}{1 - \mathbf{q}_r}. \tag{3.2}$$

The solution is the only $i \in \mathbb{N}$ that satisfy:

$$\log_{\mathbf{q}_r} \left(\mathbf{q} \frac{1 - \mathbf{q}_r}{\mathbf{q}_{min}} - 1 \right) \leq i \leq \log_{\mathbf{q}_r} \left(\mathbf{q} \frac{1 - \mathbf{q}_r}{\mathbf{q}_{min}} \right) \tag{3.3}$$

This procedure is performed during the integration of the scattering rates, but also when computing the state after scattering. In principle, one can store all the partial results of the integration procedure and reuse them during the MC step, but this has proved to be too memory consuming and is viable only if one considers only the Γ valley and $m_x = m_z$.

3.2.3 Data caches

The last important optimisation regards data caching. Very simply, the new version of the simulator tries to store and reuse as many partial results as possible in order to avoid to compute again data that does not change very often. Results of trigonometric functions and transcendental functions (like exp and log) are prime candidates for this operation.

3.3 Parallelisation

Code parallelisation is an excellent way for improving the performances. Typically, code parallelisation is achieved by dividing a complex job into smaller pieces and by assigning each piece to a "worker", which can be a set of threads running on the local computer, a set of processes running on one or more networked computers or a combination of the two. Depending on how the workers communicate, parallel code comes in two flavours:

Shared memory : this kind of parallelism implies that a process divides the work among a team of threads that share the same memory space. This model is simple to implement but the scaling is limited by the number of CPU cores available;

Message passing : in this case, the work is divided among multiple processes, each running with its own memory space and communicating with other processes via message exchange. These processes can run on different computers connected to a network. This kind of parallelism scales better but it is more difficult to implement.

We have chosen the former approach, using OpenMP [9]. Shared memory parallelism involves new aspects that are totally absent when writing sequential code.

Race conditions When the code is parallelised, each thread will be in charge of performing part of the total work. It is possible that threads will need to communicate with each other or write a common area of memory. In these cases, the threads must access shared data according to a controlled and predictable fashion, otherwise inconsistencies may arise. A race condition is a situation where the results produced by the parallel code depend on the order by which the threads performed their jobs.

Deadlocks Race conditions can be avoided by protecting the critical memory areas with mutually exclusive locks. If these locks are not used properly, threads may wait on other threads forever and the program execution stalls.

False sharing This is a more subtle issue because it affects the performances but not the results. Data is transferred between system memory and cache memory in blocks of fixed size. These blocks are called *cache lines* and they are 64 bytes long on x86-64 compatible CPUs. In a multi-core environment, it is possible that many CPU cores have the same line stored in their respective caches. If one core modifies a line it must notify the other cores that have the same line, otherwise they might use stale data. The details of this mechanism depends on the cache coherency protocols that is implemented. The MESI protocol [10] is one of these protocols and probably is the most common. The following listing shows an example of affected code:

```
void foo(double *a, double *b, int n)
{
    for (int i = 0; i < n; ++i)
    {
        if i % nThreads == threadId
            a[i] = g(b[i]);
    }
}
```

`nThreads` is the number of threads and `threadId` is the unique thread number. Here, when thread `0` tries to write `a[0]`, it must invalidate the entire line of cache containing data from `a[0]` to `a[7]` and the CPU cores executing the other threads must update their cache. This mechanism can cause a significant overhead and loss of performances. There are two ways to avoid this issue:

Padding The array `a` is padded by inserting "dummy" elements. This means the size of `a` must be multiplied by the size of a cache line divided by the size of one element of `a`. If `a` is an array of `double` the on an x86-64 CPU the size of `a` must be multiplied by 8 and

the elements of `a` are accessed like `a[i*8]` instead of `a[i]`. This approach wastes a lot of memory and should not be used for big arrays.

Bigger chunks The array is divided into chunks and each thread processes one chunk. The code above then reads:

```
void foo(double *a, double *b, int n)
{
    int chunkSize = n / nThreads;
    for (int i = 0; i < nThreads; ++i)
    {
        for (int j = 0; j < chunkSize; ++j)
        {
            a[i * chunkSize + j] = g(b[i * chunkSize + j]);
        }
    }
}
```

There may still be some false sharing if `chunkSize` is not a multiple of the size of cache line.

If `a` contains one element for each thread then we have a third option. We can get rid of the array altogether and turn each element of `a` into a thread-private variable.

Load balance. As false sharing, this does not affect the results but affects performances. In a typical parallel program, there are strictly sequential sections interleaved with parallel section. All the threads of a parallel section must complete their work before the following sequential section can be executed. If some threads have to do more work, the faster threads will wait for the slower ones.

3.3.1 Parallel Schrödinger solver and scattering rates computation

As noted before, these two steps work on one section at a time, with no interference between neighbouring sections. This makes them embarrassingly parallel steps.

3.3.2 Parallel Monte Carlo

Recall that the Monte Carlo step implements an ensemble Monte Carlo procedure: the motion of a set of particles is simulated for a number (N) of time steps (Δt) [11]. A trivial parallel implementation will divide the set of particles in subsets and each thread will process one subset. Although simple, this approach has two issues that must be solved:

- how to divide the ensemble of particles over the various threads? A bad division will create work imbalance
- how to synchronise the different threads? The threads are almost completely independent. As described in section 2.4.4, the final state may be rejected according to the value of the f function, which depends on the state of **all** particles and not only on the particles processed by a single thread. A tighter synchronisation will affect the performance negatively, while a looser synchronisation may produce inaccurate results.

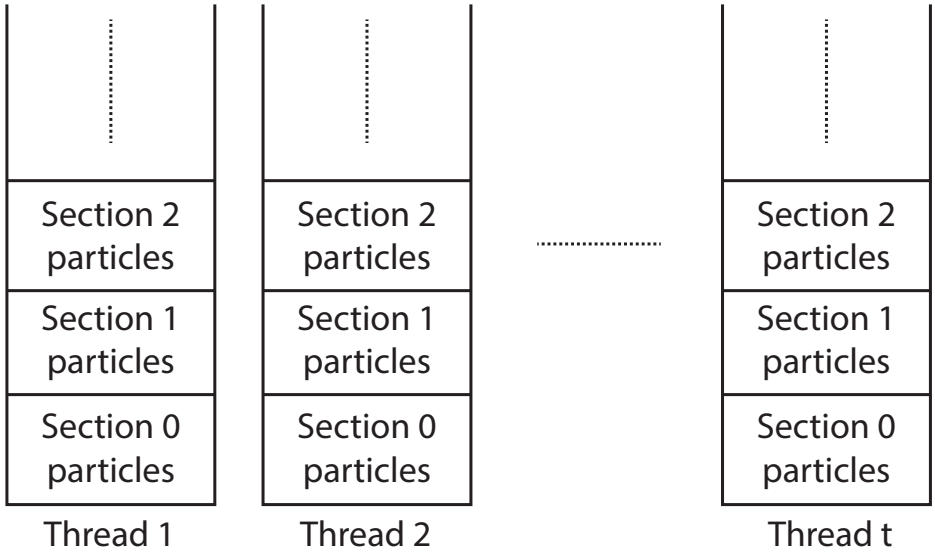


Figure 3.6: Particles are assigned to each thread according to their section and are distributed among the threads as uniformly as possible

Workload distribution We have to decide how to assign the particles to each thread. Performance scaling with the number of threads can be impaired if the number of particles processed by one thread is too different from the number of particles processed by others. Particles assignment to the threads based on the section where the particle belongs (similar to what is done in [12, 13]) minimises the amount of data structures accessed by each thread, but requires a significant overhead to trace the particles exiting the domain of one thread to enter the domain of another thread. We thus decided to evenly distribute the particles of each section to all threads (similar to what is done in [14]). This criterion applies also to the particles injected at the contacts. (see section 2.4.5 and [15] for the description of how contacts are implemented in our simulator). More precisely, a thread cannot receive a second particle until all the other threads had received their first particle. Roughly the same number of particles will leave the domain of one thread and roughly the same number will enter. This approach keeps the number of particles processed by each thread almost the same during the simulation and is sketched in Fig. 3.6. Figure 3.7 shows this technique can keep the number of particles processed by each thread roughly constant. Finally, Fig. 3.8 compares the time during which a thread was active (dark) with respect to the time during which a thread was idle (light). Since these lightly coloured are few in number, the work is well balanced.

Thread synchronisation The next problem to solve is how to synchronise the thread. Since we are interested in steady-state solutions, we can allow the threads to "drift apart", meaning that, at a given time, the motion of particles in a chunk may have been computed over a longer time with respect to the particles of other chunks. However, when enforcing the Pauli's exclusion principle, we need to know the occupation function f to reject scattering events [16] (see section 2.4.4) and f depends on the state of all the simulated

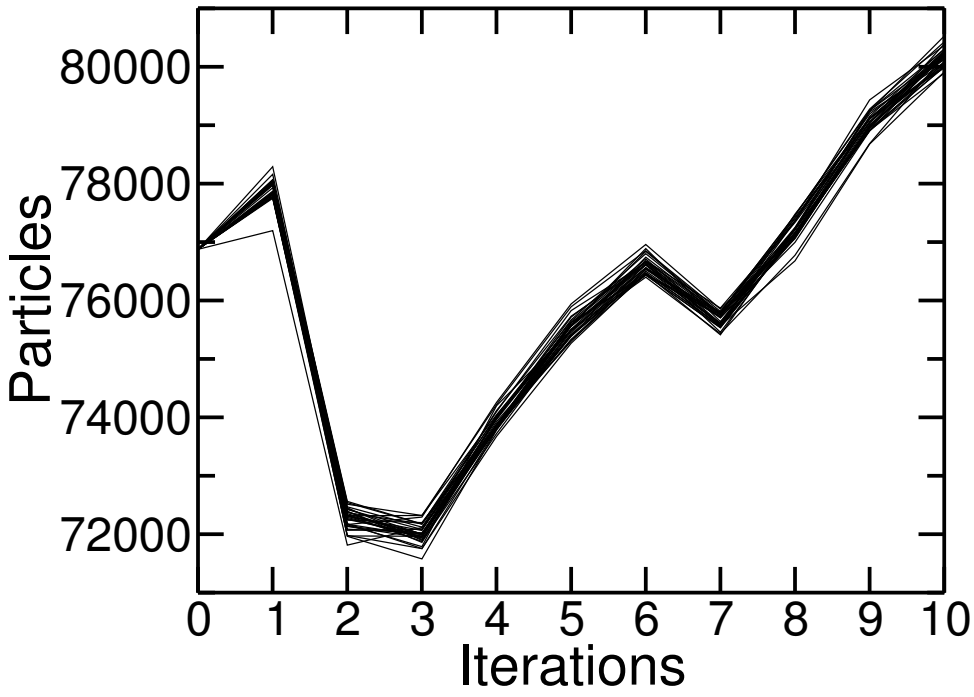


Figure 3.7: Number of particles processed by each thread with respect to the iterations. While the total number of particles will change while the simulation moves on, the number of particles processed by each thread is roughly constant. Each line represents the number of particles processed by each thread. The figure was obtained from the simulation of device #3 of Fig. 3.2. The simulation uses 32 threads.

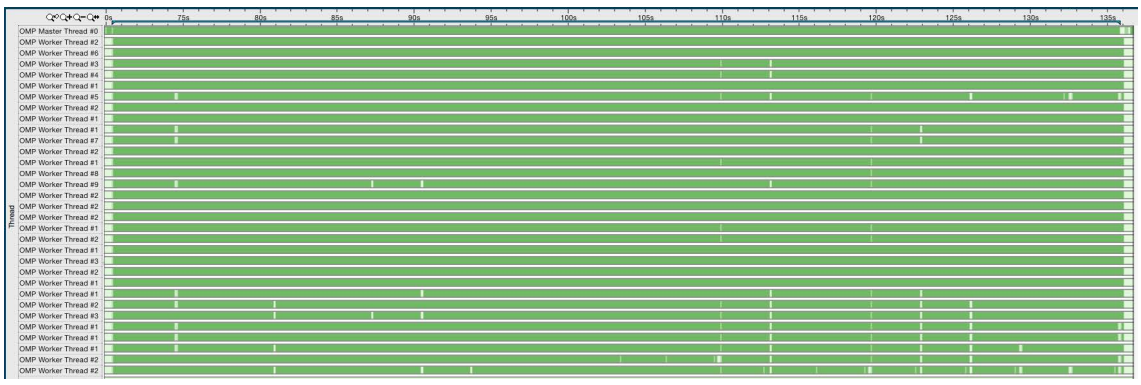


Figure 3.8: Screenshot of the profiler results showing one execution of the MC step. The darker regions represents the time during which a thread was busy doing work, while the lighter regions represents the time during which the thread was idle. Smaller amounts of the latter regions mean that the threads are well balanced.

particles, not only those processed by one thread. Also, we would like to have the least possible amount of synchronisation between threads. This requires two changes in how the data structure described in Fig 3.4b.

Distributed f function First, each thread maintains a version of f based on the particles of its chunk. Also, threads do not read the f of another thread while it is being updated. Again, to avoid the use of synchronisation constructs, each thread keeps two copies of f according to a technique called "double buffering". Other threads read from a front buffer while its owner clears and updates the back buffer at the beginning of each time step. When the update is completed, the buffers are swapped. This requires more memory but minimises the access to shared data and avoids synchronisation altogether when updating f. There is also another option: we keep only one tree but the leaves point to arrays where each thread stores its own version of f. This allows fewer jumps across the memory and better cache exploitation when computing the state after scattering, but the performances are severely decreased due to false sharing when f is updated. The false sharing results from the fact that data written by different threads are located too close in memory.

Occupation probability computation Second, the f data structure does not store the occupation probability. Instead, it store for each state the sum of the weights of the particles belonging to that state. When computing the state after scattering (see section 2.4.4), the occupation probability is computed by combining the data from all threads:

$$f_{(s,\nu,n,\mathbf{k})} = \frac{\sum_{i=0}^{t} w^i_{(s,\nu,n,\mathbf{k})}}{\frac{2W(d\mathbf{k})^2}{(2\pi)^2} \cdot dx \cdot \mu_\nu} \tag{3.4}$$

where W is the width of the device, $d\mathbf{k}$ is the discretisation width on the $\mathbf{k}$-space, dx is the width of section s, μ_ν is the multiplicity of the valley ν, t is the number of threads and $w^i_{(s,\nu,n,\mathbf{k})}$ is the sum of the weights of the particles belonging to the chunk processed by the i-th thread and belonging to the state $(s, \nu, n, \mathbf{k})$.

Explicit synchronisation Before the statistics collection phase, all threads are synchronised (by using an explicit barrier [9]). This in done in order to avoid mixing together information from particles processed for a too different amount of times when computing f. Fig. 3.9 shows that increasing the number of time steps (Δt) between two synchronisation points affects the efficiency of the parallelisation process, that is however negligible above a given number of steps. On the other hand, since we are simulating a steady-state process, the error associated to poor synchronisation is essentially negligible.

Random numbers generator and false sharing Last but not least comes the random numbers generator (RNG). Section 2.4 shows how important these numbers are. Standard library functions like `rand()` use a hidden "state" to generate the next number and this state is updated after each generation. Usually this state is implemented as a global variable and access to this variable is protected via futexes [17]. This approach prevents more than one thread from using the RNG at a given time because many threads must wait for the lock to become unlocked and this affects the performances. To solve this issue,

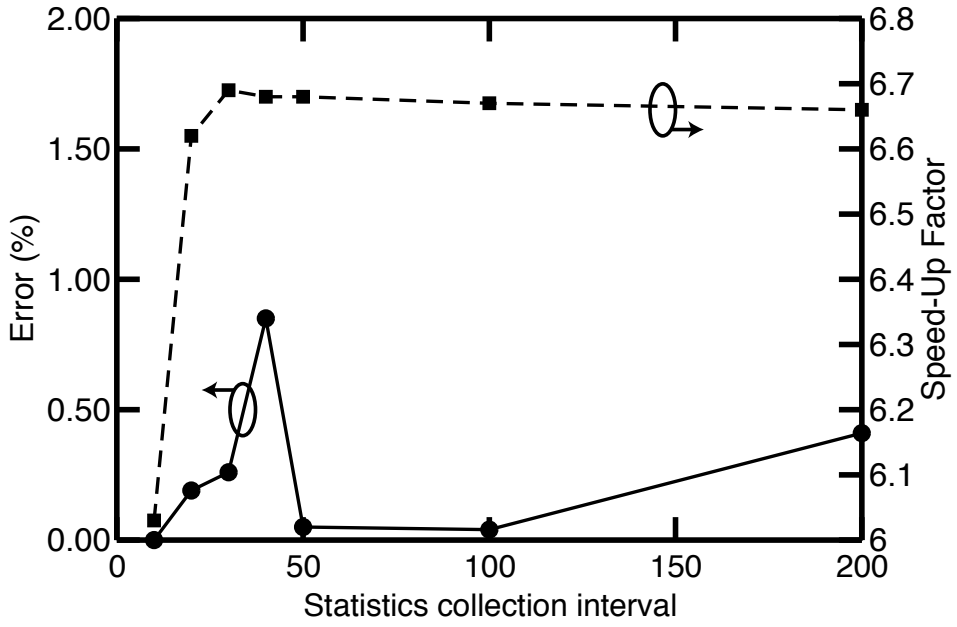


Figure 3.9: Increasing the number of time steps between thread synchronisation and statistics collection causes a negligible increase of the simulation error (see section 3.4 for the definition of the simulation error) but it improves slightly the speed-up. These measurements were performed on the simulation of device #2 of Fig.3.2 and the simulation uses 8 threads.

we use a reentrant [18, sect. 12.3.8] RNG (like `rand_r()`), otherwise some time is spent in serializing the accesses to the global state of a traditional random number generator. A reentrant RNG stores its state in a user supplied memory area. In our case, this memory area is a thread-private variable so we can avoid false sharing issues as described at the beginning in section 3.3.

3.4 Methodology and Benchmarks

To benchmark the optimised MSMC simulator we measured the execution times of the four steps in Fig. 2.2 using a profiler [5]. As introduced in section 3.1, we have simulated the three template MOSFETs described in Fig. 3.2. All benchmarks were performed on a workstation equipped with four Xeon E5-4650 and 192GiB of DDR3 main memory (1GiB=2^{30}B).

Scattering mechanisms and simulation parameters All devices have been divided in 100 sections. The mesh has 100 points along the quantisation direction and the Schrödinger equation is solved on a thinner mesh consisting of 3000 points. The MC step was configured to simulate about 1 million particles for devices #1 and #2 and about 2 millions for device #3 at each iteration. The motion of the particles is simulated for 2000 time steps of 0.1 fs each. The gate voltage is 0.5V for devices #1 and #2 and 0.8V for device #3. V_{DS} is 0.5V for devices #1 and #2 and is 0.8V for device #3. We have enabled the following scattering mechanism:

- Phonon scattering and surface roughness scattering for all of three devices. Surface roughness parameters for devices #1 and #2 are taken from [19] and for device #3 from [6];
- Alloy scattering for devices #2 and #3 since the semiconductor is an alloy of two semiconductors;
- Remote phonon and Polar optical phonons for device #3, since the semiconductor is a III-V compound and the gate dielectric a is high-k material.

Simulation error For the sake of a fair comparison, we set the appropriate number of iterations of the loop in Fig. 2.2 to reach a given relative error. The error is computed according to the procedure described in [1], which is:

1. At the end of each iteration (except the first N_{tran} ones, to discard the initial transient phase) we calculate the channel current I_D by averaging the current over the sections in the channel region;
2. then we compute $\bar{I}_D$ as the average of I_D over all the previous iterations and $\sigma_{\bar{I}_D}$ as its unbiased standard deviation. The first N_{tran} are not considered because the simulation is still in the transient phase. $N_{tran} = 10$ was deemed sufficient to avoid propagating errors from the initial transient.
3. the simulation is stopped when the coefficient of variation $r_{err} = \sigma_{\bar{I}_D}/\bar{I}_D$ falls below a chosen threshold.

Figure 3.10 shows how the error reduces while the simulation progresses. The ITRS roadmap for device modelling requires an error on the on current not greater than 5% [20]. We have chosen a more stringent error threshold of 1%. Device #1, #2 and #3 need 25, 14 and 11 iterations respectively for the error to drop below the threshold.

Metrics To evaluate the quality of the improved code we must first decide what to measure and define the appropriate metrics. Every parallel program has some strictly serial portions. These portions are defined as the code that cannot be parallelised and lies outside every parallel region and is therefore executed by only one thread. If T_s is the time spent executing this serial portion and T_p is the time spent executing the parallel portion, the total execution time can be written as:

$$T(p) = T_s + \frac{T_p}{p}, \tag{3.5}$$

where p is the chosen number of threads. Obviously, $T(1) = T_s + T_p$ is the execution time when only one thread is used. If we define the serial portion as $s = T_s/T(1)$, the equation above can be written as:

$$T(p) = T(1)s + \frac{T(1)(1-s)}{p}. \tag{3.6}$$

From this equation we can define the speedup $SU(p)$ as:

$$SU(p) = \frac{T(1)}{T(p)} = \frac{1}{s + \frac{1-s}{p}} \tag{3.7}$$

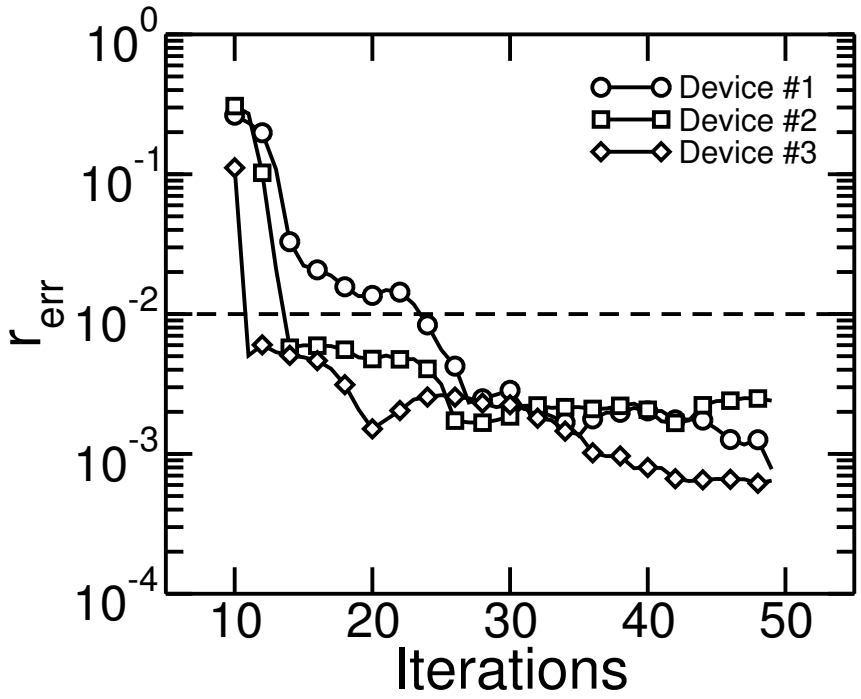


Figure 3.10: Simulation error versus iteration number. The dashed line shows the 1% error threshold

Equation 3.7 is the well known Amdahl's Law [21]. This law can be used to estimate the maximum speedup that can be achieved when the number of threads is increased while executing a program that consists of both sequential and parallel regions. The law states that when p tends to infinity, the maximum speedup saturates and is essentially limited to $1/s$. Therefore, the code must be written so that s is as small as possible. It is obvious that a good estimate of s is required in order to use Eq. 3.7. An overestimated s produces bizarre results, like a measured speedup greater than what the law predicts. An underestimated s may make people think that their parallel code is not good. The profiler can measure T_s and from this we can compute s easily. Eq. 3.7 has one big flaw: it assumes that there are no overheads and the load is perfectly balanced, that is, during the time interval T_p there are always p active threads doing useful work. Work imbalance can cause the number of active threads to drop below p, which can be seen as an increase of s.

By going the other way around, we can invert Eq. 3.7 and find an explicit expression for s:

$$s = \frac{\frac{1}{SU} - \frac{1}{p}}{1 - \frac{1}{p}}. \tag{3.8}$$

where the speedup is the measured one and not the an ideal value. As a direct consequence, s will depend not only on the strictly serial portion but also on the portion of parallel code that does not scale ideally because of work imbalance, overheads or hardware issues. Equation 3.8 is known as Karp-Flatt metric [22].

An irregular increase of s when p is increased indicates a load balance issue. In the Monte Carlo step the amount of time required to process a chunk of particles is not constant. In the scattering rates computation step, if the number of sections is not divisible by p,

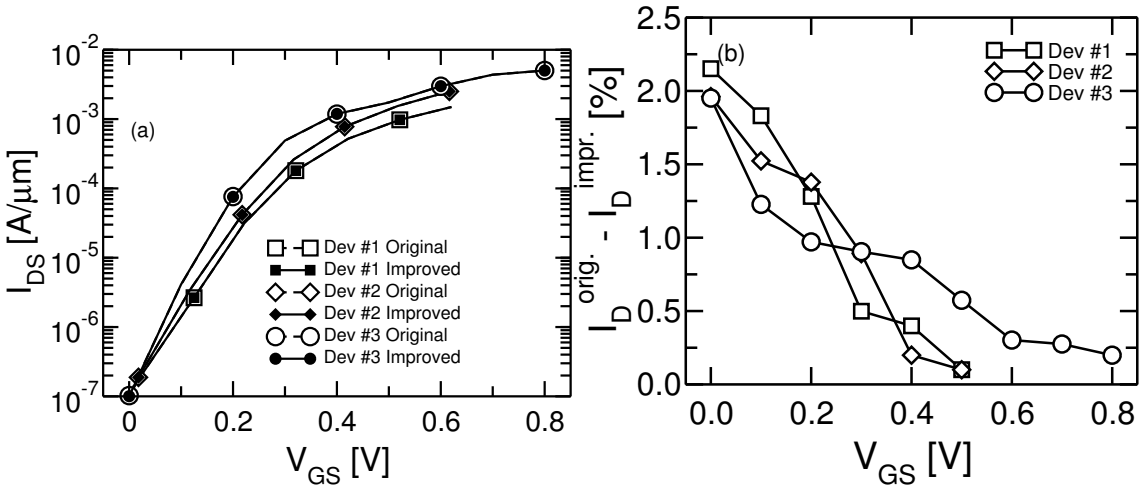


Figure 3.11: (a) Drain current versus gate voltage for the three devices. The "improved" label refers to the optimised and parallel version of the simulator. The currents are extremely similar, thus proving the correctness of our work. (b) Percentage difference between the currents computed with the original and the improved version of the simulator. Maximum difference is about 2.2% and the biggest differences are located in the subthreshold region.

sooner or later some threads will become idle. On the other hand, a smoothly increase of s with respect to the number of threads indicates overhead issues. This overhead can come from synchronisation constructs or from thread creation/destroy processes. The latter case indicates that the granularity of the parallelism is too fine. A third metric that can be employed is the efficiency e, that can be defined as

$$e = \frac{T(1)}{pT(p)} = \frac{SU(p)}{p}. \tag{3.9}$$

Clearly, the efficiency should be as close as possible to 1. Combining these metrics (serial portion and efficiency) one can assess the quality of the parallel code. As an example, if e drops rapidly but s stays constant when p is increased, the parallelism is too limited.

3.5 Results

As a first sanity check, we verified that the same currents were obtained with the original and the improved codes. The first two devices show a very similar current. We believe that the alloy scattering is compensating the improvements due to strain. These two devices are better analysed in [6]. Fig. 3.11a shows the drain currents vs the gate voltage for the three devices described in Fig. 3.2, while Fig. 3.11b shows the percentage difference between the results obtained with the two versions of the simulator. The I_D computed by the improved code has a maximum discrepancy of 2.2% with respect to the original code and the largest differences are in the subthreshold region.

3.5.1 Optimisation results

To assess the impact of the implemented optimisations, we compare the execution time of the original code with the execution time of the improved single-threaded version. We are

interested in the average execution time of one iteration. Table 3.2 reports the performance improvements due to the implemented optimisations.

Regarding the SC step when executing the original code, we can see that this step is the fastest for device #1, since the only anisotropic scattering mechanism we consider is the Surface Roughness. For device #2 this step is a bit slower because of the additional Alloy scattering mechanism. For both devices we consider the same number of valleys (see Fig. 3.2) and β-bins. The improved version performs a little better when simulating device #1 due to the optimisation of the matrix element interpolator, but the code that computes the matrix element is essentially the same. A larger improvement of the SC step when simulating device #2 is obtained due to some data caches used when computing the matrix elements for the Alloy scattering mechanism. The simulation of device #3 is the slowest when using the original code. Beside Surface Roughness and Alloy scattering, we consider also scattering due to Remote Phonons and Polar Optical Phonons. The latter is responsible for the significant increase of the computation time of the SC step and overshadows the fact that we need to compute the transition rate out of only one valley. The time spent on integrating the matrix elements and computing the scattering rates is however greatly reduced when comparing with the simulation of the other devices. This is due to the fact that we have only one source valley and we need only one β-bin because $m_x = m_z$. When analyzing the results obtained with the improved code we can see that there is a huge improvement of the time required to complete the SC step. This is due to the data caches used when computing the matrix elements for the scattering due to Polar Optical Phonons (and, to a lesser extent, due to Alloy scattering). The improvement of the matrix element integration sub-step is consistent across the simulations of all three devices.

Regarding the MC step, we can see that the management of the data structure describing the f function requires a significant amount of time when executing the original code. This amount is similar for the first two devices and is larger for the third device because we are simulating the motion of more particles. Overall, the MC step lasts longer when simulating device #2 with respect to device #1 due to additional scattering events caused by the Alloy scattering mechanism. This increased scattering rate requires us to compute more states after scattering. A similar increase of the computation time is observed when simulating device #3. Again this is due to the additional scattering mechanisms considered for this device. When moving to the improved code we can see that the execution time is greatly reduced. This is due to the improved code that manages the f function, to the matrix element interpolator used when computing the state after scattering and to some caching of the most frequently computed data. All these optimisations do not depend on any specific property of the devices, so the improvement are roughly the same for all devices, as expected.

3.5.2 Parallelisation results

Figure 3.12 summarises how the parallel code scales with the number of threads. We can immediately see that the speedup quickly deviates from the ideal (the ideal speedup is given by Eq, 3.7) when the number of threads is greater than 8. The underlying hardware is playing a role here. The system used for the benchmarks has 4 CPUs and each of them has 8 physical cores. When the simulation uses more than 8 threads, these threads will require data that are likely contained in the cache memory of the other CPUs. This is especially true for the MC step due to the rejection of the states after scattering based on

Table 3.2: Performance improvements due to optimisations alone. MC - f function refers to the time spent on clearing the data structure for the occupation function f and recomputing its value (see Sections 2.4.4 and 3.2.1 . MC - Interpolation refers to the time spent on interpolating the matrix elements when computing the state after scattering (see Section. 2.4.4. SC - Interpolation refers to the time spent on interpolating the matrix elements when integrating the scattering rates (see Section 2.3.8).

(a) Device 1

Step	Original time (s)	Improved time(s)	Difference (%)
Total	3903.9	1113.6	71.5
MC - Total	3147.5	484.3	84.6
MC - f function	1199.3	80.7	93.3
MC - Interpolation	20.4	6.7	67.2
SC - Total	737.5	615.0	16.6
SC - Interpolation	219.1	172.1	21.5

(b) Device 2

Step	Original time (s)	Improved time(s)	Difference (%)
Total	4212.3	1355.9	67.8
MC - Total	3185.3	537.2	83.1
MC - f function	1254.2	84.8	93.3
MC - Interpolation	38.2	15.9	58.4
SC - Total	993.4	807.0	18.8
SC - Interpolation	364.1	284.4	21.9

(c) Device 3

Step	Original time (s)	Improved time(s)	Difference (%)
Total	7279.4	1274.4	82.5
MC - Total	5000.4	903.8	82.0
MC - f function	1849.4	101.7	94.5
MC - Interpolation	57.3	19.6	65.8
SC - Total	2273.1	367.4	85.6
SC - Interpolation	15.6	7.1	21.9

the f function (see sections 2.4.4 and 3.3.2.

Fig. 3.13 shows the speedup of the SC and MC steps. The two blocks scale similarly up to 20 threads, but they diverge if more threads are used. Generally speaking, the SC step scales slightly better than the MC step when simulating devices #1 and #2. This is due to the fact that this step does not contain any explicit synchronisation point and the work done in each section is roughly the same. On the other hand, the parallelism of the SC step is not fine-grained because while the sections are processed concurrently, all the work needed to compute the scattering rates in one section is done sequentially. The obvious consequence is that near the end of the SC step, fewer and fewer threads are active due to the fact that the number of section still to be treated is smaller than the number of available worker threads. This effect emerges clearly from the simulation of device #3 which requires to compute the scattering rates due to Polar Optical Phonons. The computation of the matrix elements for this mechanism involves the computation of a time consuming integral (see Eq. 2.61). Preliminary work was done to implement a deeper level of parallelism via OpenMP tasks, but the proper enforcement of task dependencies has increased the overhead up to the point where there all benefits of a finer-grained parallelism are lost. In the end the SC step for device #3 exhibits the worst scaling.

Additional insight can be gained by applying the other metrics defined in section 3.4. Results of the simulations are shown in Table 3.3. The execution time is the average time required to complete one iteration.

Figure 3.14 shows the serial fractions of the three devices. Overall the serial fraction increases slowly but steadily, with the exception of device #3. The behaviour of the total serial fraction can be better understood by looking at the serial fractions of the SC and MC steps, which are the most expensive steps of the simulation. The serial fraction of the SC step increases erratically, which is consistent with the load balance issue described before. As a practical example, consider a device divided in 100 sections. If the simulation uses 33 threads then the first 99 sections will be processed concurrently in 3 groups of 33 sections, while the last section will be processed by only one thread. This can be seen as an increase of the serial fraction.

On the other hand the serial fraction of the MC step increases quite uniformly, consistent with the effect of synchronisation overhead. Remember that during this step the threads are synchronised every 100 time steps, just before the statistics are collected. There are about 20 synchronisations for each execution of the MC step.

Device #3 shows a behaviour that is different with respect to device #1 and #2: the serial fraction of the SC step is higher than MC's one. This is due to the fact that for this device we consider scattering from Polar Optical Phonons and this is the most expensive scattering mechanism to compute since Eq. 2.61 requires the evaluation of a double integral on the cross product of the eigenfunctions. The computation of this equation completely overshadows the time spent on computing the scattering rates for the other mechanisms, which results in a lower concurrency towards the end of the SC step because fewer threads are active.

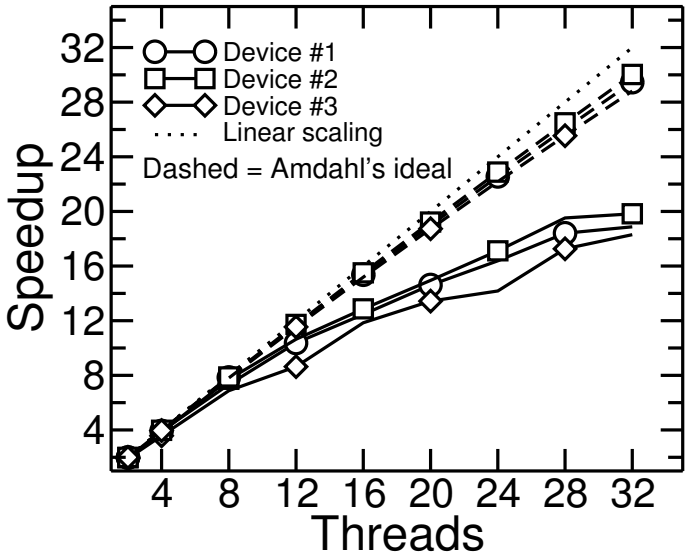


Figure 3.12: Speedup of the simulator with respect to the number of threads.

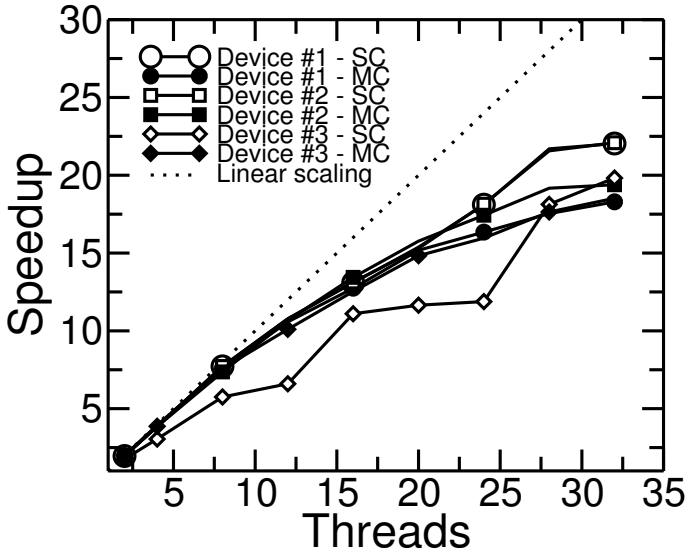


Figure 3.13: Speedup of the SC and MC steps with respect to the number of threads.

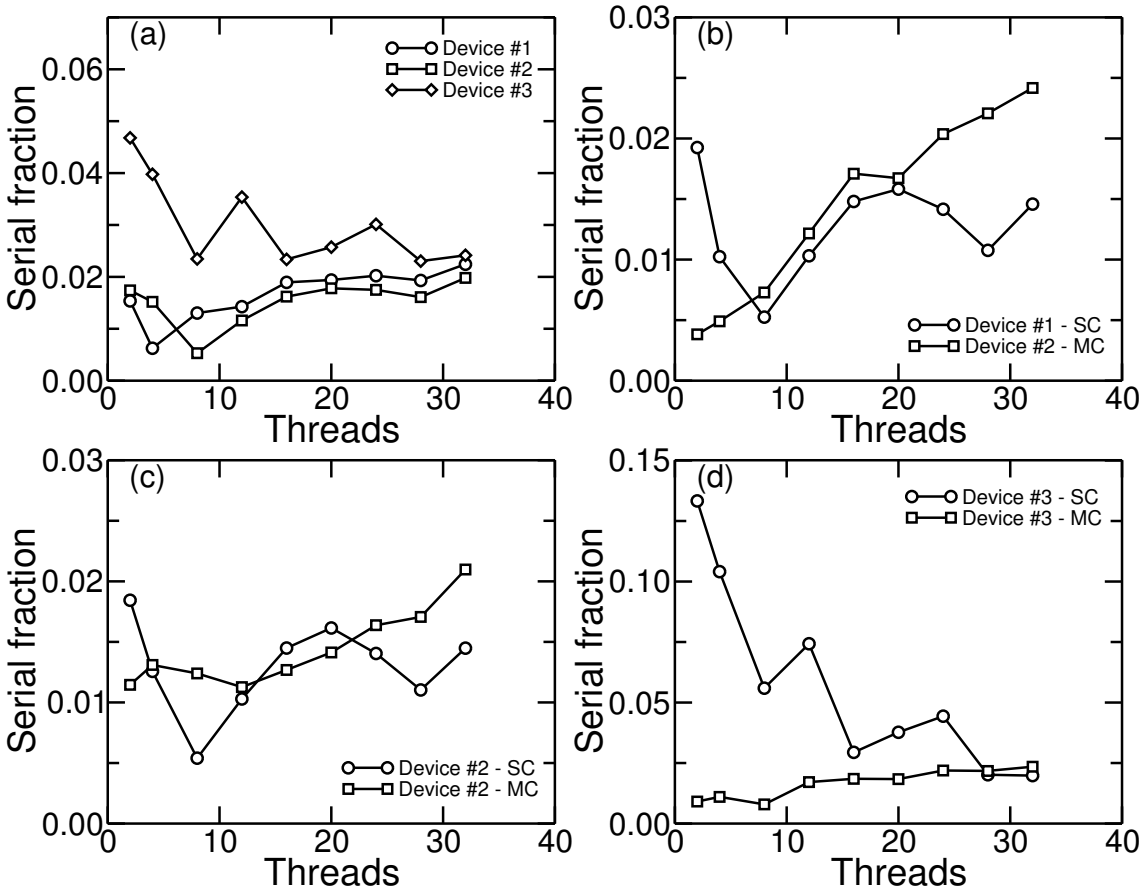


Figure 3.14: Serial fractions of the three benchmarked devices. The serial fraction is computed by using Eq. 3.8. (a) total serial fractions. (b,c,d) serial fractions of the SC and MC steps for each device.

Table 3.3: Execution time of the simulation of the three devices of Fig. 3.2 and performance metrics with respect to the number of threads. Time is the average time required to complete one iteration.

(a) Device #1

Threads	**Time (s)**	**Speedup**	**serial fraction**	**efficiency**
1	1194.3	-	-	-
2	606.3	1.97	0.0154	0.985
4	304.2	3.93	0.0062	0.982
8	162.9	7.33	0.0130	0.916
12	115.1	10.37	0.0143	0.864
16	95.8	12.46	0.0189	0.779
20	81.7	14.61	0.0194	0.731
24	72.9	16.38	0.0202	0.683
28	64.9	18.40	0.0193	0.657
32	63.3	18.87	0.0224	0.590

(b) Device #2

Threads	**Time (s)**	**Speedup**	**serial fraction**	**efficiency**
1	1412.9	-	-	-
2	718.8	1.966	0.0174	0.983
4	369.3	3.826	0.0152	0.956
8	183.2	7.711	0.0053	0.964
12	132.7	10.646	0.0116	0.887
16	109.8	12.866	0.0162	0.804
20	94.5	14.952	0.0178	0.748
24	82.5	17.117	0.0175	0.713
28	72.4	19.525	0.0161	0.697
32	71.3	19.826	0.0198	0.620

(c) Device #3

Threads	**Time (s)**	**Speedup**	**serial fraction**	**efficiency**
1	1274.4	-	-	-
2	667.0	1.911	0.0468	0.955
4	356.6	3.574	0.0398	0.893
8	185.5	6.872	0.0235	0.859
12	147.5	8.639	0.0354	0.720
16	107.6	11.847	0.0234	0.740
20	94.9	13.433	0.0257	0.672
24	89.9	14.178	0.0301	0.591
28	73.8	17.264	0.0230	0.617
32	69.6	18.299	0.0243	0.572

Bibliography

[1] C. Jungemann, S. Yamaguchi, and H. Goto. "Convergence estimation for stationary ensemble Monte Carlo simulations". In: *Proc.SISPAD*. Sept. 1997, pp. 209–212.

[2] `http://www.synopsys.com/Tools/TCAD/CapsuleModule/news_dec04.pdf` page 7.

[3] `http://ark.intel.com/`.

[4] `http://www.lanl.gov/orgs/hpc/salishan/salishan2011/3moore.pdf`.

[5] `http://software.intel.com/en-us/intel-vtune-amplifier-xe`.

[6] D. Lizzit, P. Palestri, D. Esseni, A. Revelant, and L. Selmi. "Analysis of the performance of n-Type FinFETs with strained SiGe Channel". In: *IEEE Trans. on Electron Devices* 60.6 (June 2013), pp. 1884–1891.

[7] P. Osgnach, A. Revelant, D. Lizzit, P. Palestri, D. Esseni, and L. Selmi. "Toward computationally efficient Multi-Subband Monte Carlo simulations of nanoscale MOSFETs". In: *Proc.SISPAD*. 2013, pp. 176–179.

[8] Peter J. Denning. "The Locality Principle". In: *Commun. ACM* 48.7 (July 2005), pp. 19–24.

[9] B. Chapman, G. Jost, and R. van der Pas. *Using OpenMP: Portable Shared Memory Parallel Programming*. The MIT Press, 2007.

[10] Mark S. Papamarcos and Janak H. Patel. "A Low-overhead Coherence Solution for Multiprocessors with Private Cache Memories". In: *SIGARCH Comput. Archit. News* 12.3 (Jan. 1984), pp. 348–354. ISSN: 0163-5964. DOI: `10.1145/773453.808204`. URL: `http://doi.acm.org/10.1145/773453.808204`.

[11] D. Esseni, P. Palestri, and L. Selmi. *Nanoscale MOS Transistors*. Cambridge University Press, 2011.

[12] A. Kepkep, U. Ravaioli, and B. Winstead. "Cluster-based parallel 3-D Monte Carlo device simulation". In: *International Workshop on Computational Electronics*. May 2000, pp. 21–22.

[13] Wei Zhang, Gang Du, Qiang Li, Aiqing Zhang, Zeyao Mo, Xiaoyan Liu, and Pingwen Zhang. "A 3D Parallel Monte Carlo Simulator for Semiconductor Devices". In: *International Workshop on Computational Electronics*. May 2009, pp. 1–4.

[14] A. Hiroki, S. Odanaka, and A. Goda. "Massively Parallel Computation For Monte Carlo Device Simulation". In: *Proc.Int.Workshop on VLSI Process and Device Modelling*. May 1993, pp. 18–19.

[15] P. Palestri, L. Lucci, S. Dei Tos, D. Esseni, and L. Selmi. "An improved empirical approach to introduce quantization effects in the transport direction in multi-subband Monte Carlo simulations". In: *Semiconductor Science Technology* 25.5 (2010), p. 055011.

[16] P. Lugli and D.K. Ferry. "Degeneracy in the ensemble Monte Carlo method for high-field transport in semiconductors". In: *IEEE Trans. on Electron Devices* 32.11 (Nov. 1985), pp. 2431–2437.

[17] H. Franke and E. Russell. "Fuss, Futexes and Furwocks: Fast Userlevel Locking in Linux". In: *Ottawa Linux Symposium*. June 2002, pp. 479–495.

[18] A. Tanenbaum. *Modern Operating Systems 3rd edition*. Pearson, 2007. ISBN: 978-0-13-813459-4.

[19] F. Conzatti, N. Serra, D. Esseni, M. De Michielis, A. Paussa, P. Palestri, L. Selmi, S.M. Thomas, T.E. Whall, D. Leadley, E.H.C. Parker, L. Witters, M.J. Hytch, E. Snoeck, T.J. Wang, W.-C. Lee, G. Doornbos, G. Vellianitis, M.J.H. van Dal, and R.J.P. Lander. "Investigation of Strain Engineering in FinFETs Comprising Experimental Analysis and Numerical Simulations". In: *Electron Devices, IEEE Transactions on* 58.6 (June 2011), pp. 1583–1593.

[20] `http://www.itrs.net/Links/2012ITRS/2012Tables/Modeling_2012Tables.xlsx`.

[21] Gene M. Amdahl. "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities". In: *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*. AFIPS '67 (Spring). Atlantic City, New Jersey: ACM, 1967, pp. 483–485.

[22] Alan H. Karp and Horace P. Flatt. "Measuring Parallel Processor Performance". In: *Commun. ACM* 33.5 (May 1990), pp. 539–543. ISSN: 0001-0782.

Chapter 4

Simulation of III-V devices and comparison with other models

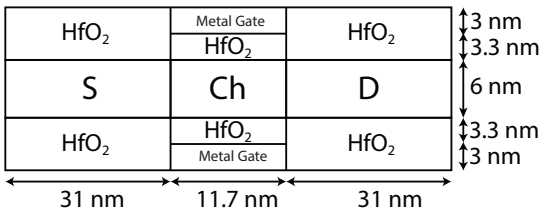
In this chapter we show the results of the simulation of three devices featuring III-V channel materials. Our goal is to compare the MSMC simulations with experimental results and simulators based on the NEGF formalism.

4.1 11.7 nm InGaAs template device

The template device is constructed based on what the ITRS Roadmap [1] foresees for year 2020. The only difference is the body thickness, which was reduced from 7 nm to 6 nm in order to achieve a better electrostatic integrity. Figure 4.1 sketches the device and its main parameters. The HfO_2 gate dielectric material was used also to cover the source and drain regions on both sides. We call these regions *spacers*.

We have simulated the drain current I_D by ramping the voltage applied to both gates from 0.0 V to 0.6 V. These simulations are ballistic and no scattering mechanism is considered. Also, we did not consider the effect of interface states, that will be discussed in Chapter 5, in order to estimate the maximum current drive and compare the MSMC with ballistic NEGF simulations. Results obtained with the MSMC simulator are shown in Fig. 4.2(a) and have been compared with those obtained by the two other models. One is a NEGF with an atomistic (tight binding) hamiltonian [2, 3] (courtesy of Prof. Mathieu Luisier from ETH Zurich), while the other is still a NEGF simulator but it uses a $\mathbf{k} \cdot \mathbf{p}$ hamiltonian [4, 5] (courtesy of Prof. Elena Gnani and Dr. Roberto Grassi from the University of Bologna) and considers only the Γ valley. We can immediately see a significant difference between the results of the three simulators. For a fair comparison we chose to match an OFF current of 100 nA/μm for a gate voltage of 0.0 V, as prescribed by the ITRS roadmap. To match the OFF current, we have changed the work-function of the two gates in the MSMC simulation until the target I_{OFF} value was obtained. The NEGF curves were just rigidly shifted to match the same I_{OFF}.

This is not the only choice, though. We can also try to match the current at the threshold voltage, which is 6.6 μA/μm for a gate voltage of $\approx$ 0.2 V, if we take the MSMC simulation as a reference. The threshold voltage is defined as the gate voltage at which the inversion carrier density in the channel is approximately equal to the channel doping multiplied by the semiconductor thickness. NEGF curves were again shifted to obtain the



Semiconductor	$In_{0.53}Ga_{0.47}As$
S/D Doping	5×10^{19} cm^{-3} n-type
Ch Doping	1×10^{17} cm^{-3} p-type
V_{DS}	0.61 V
Gate dielectric	3.3 nm HfO_2
Γ valley parms.	$m_l = m_t = 0.043\ m_0$, $\alpha_\Gamma = 1.5$ eV^{-1}
L valley parms.	$m_l = 1.57\ m_0$, $m_t = 0.23\ m_0$, $\alpha_L = 0.5$ eV^{-1}
Δ valley parms.	$m_l = 2.26\ m_0$, $m_t = 0.25\ m_0$, $\alpha_\Delta = 0.5$ eV^{-1}

Figure 4.1: Sketch of the ITRS template device for year 2020 as simulated with the MSMC simulator. Note that the body thickness is 6 nm instead of the 7 nm of the ITRS roadmap [1].

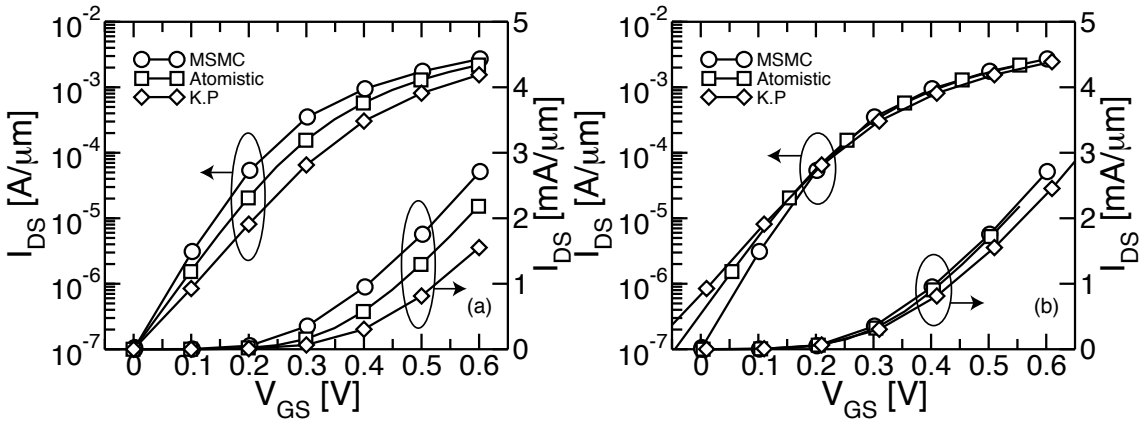


Figure 4.2: Simulated drain current of the device shown in Fig. 4.1. The results obtained by using the MSMC simulator have been compared with the results obtained with an atomistic NEGF and a $\mathbf{k}\cdot\mathbf{p}$ NEGF.(a): the simulations are matched so that the OFF current is 100 nA/μm. (b): the simulations are matched so that the same current is obtained at the threshold voltage (0.2 V as for the MSMC simulation).

same current at the threshold voltage. Results are shown in Fig. 4.2(b). Now a better agreement between the models is obtained above threshold, but there is up to one order of magnitude of difference in terms of the OFF current. This difference is due to the fact that the MSMC simulator does not model the source to drain tunnelling. This aspect will be addressed in the next section.

Fig. 4.3(a) shows the profile of the first subband along the transport direction, for the three models and for three sample gate voltages. We chose 0.5V, 0.2V and 0.0V, which correspond to the ON state, threshold state and OFF state, taking the MSMC simulation as reference. The current at the MSMC threshold voltage was used to establish a mapping with the NEGF curves. All three models agree very well, indicating that source to drain tunnelling does not affect significantly the electrostatic. On the other hand, Fig. 4.3(b) shows the inversion carrier density profile along the transport direction, and here we can see that the MSMC simulation shows a much lower density at $V_G = 0.0$ V when compared to the NEGF simulators, which is consistent with the lower OFF current. The difference is reduced when considering $V_G = 0.2$ V and is negligible at $V_G = 0.5$ V.

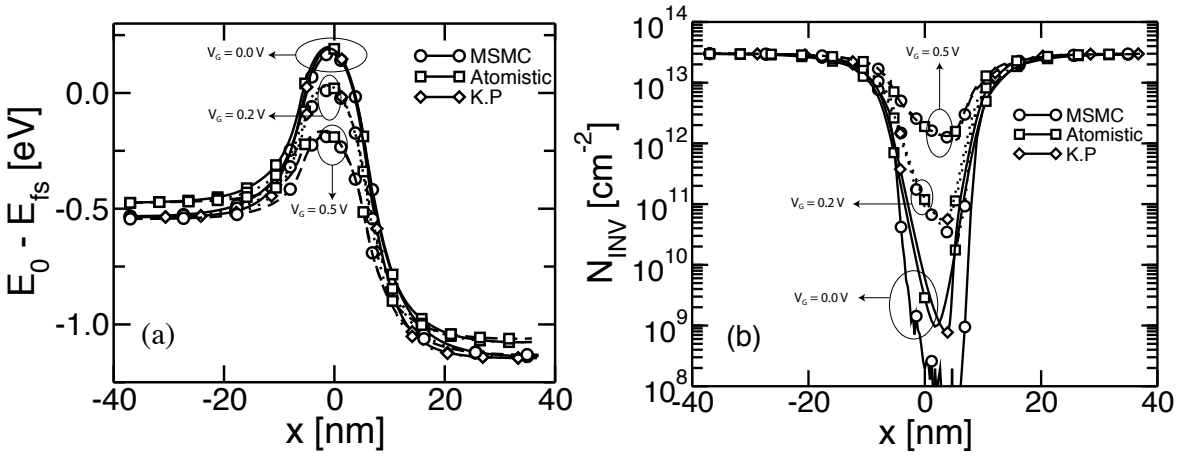
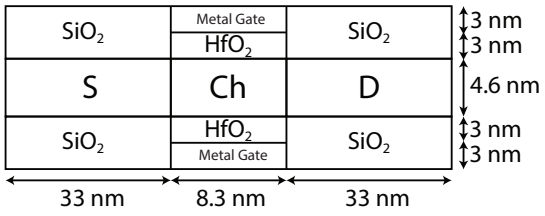


Figure 4.3: Internal quantities for the device shown in Fig. 4.1. (a): Profile of the first subband along the transport direction. (b): Profile of the inversion carrier density along the transport direction.



Semiconductor	$In_{0.53}Ga_{0.47}As$
S/D Doping	5×10^{19} cm^{-3} n-type
Ch Doping	1×10^{17} cm^{-3} p-type
V_{DS}	0.57 V
Gate dielectric	3 nm HfO_2
Γ valley parms.	$m_l = m_t = 0.043\ m_0$, $\alpha_\Gamma = 1.5$ eV^{-1}
L valley parms.	$m_l = 1.57\ m_0$, $m_t = 0.23\ m_0$, $\alpha_L = 0.5$ eV^{-1}
Δ valley parms.	$m_l = 2.26\ m_0$, $m_t = 0.25\ m_0$, $\alpha_\Delta = 0.5$ eV^{-1}

Figure 4.4: Sketch of the ITRS template device for year 2023 as simulated with the MSMC simulator.

4.2 8.3 nm InGaAs template device

This second device is also taken from the ITRS roadmap, but now we focus on what is foreseen for year 2023. Figure 4.4 sketches the device and its main parameters. For this device we study the effects of different materials for the spacers (the regions above and below the source and drain regions). We begin with SiO_2 spacers. Results are shown in Fig. 4.5(a). Again, we compare the MSMC results with the two NEGF simulators. For this device the differences between the three models are larger than in the 11.7 nm one. If we replace the SiO_2 spacer (low-k) with HfO_2 (high-k) there is a small improvement in the ON current. A slight further improvement is obtained by a gate underlap of 1.5 nm per side. As for the previous device, we have matched the current at the threshold voltage, which is 50 μA/μm for a gate voltage of ≈ 0.2 V. Results are shown in Fig. 4.5(b). The models predicts roughly the same ON current, but differ in the sub-threshold region. Again, this is due to source/drain tunnelling, which is not considered by the MSMC simulator. Fig. 4.6 shows a result similar to Fig. 4.3. The profile of the first subband (left) is almost identical in the three simulators and there are big differences in the inversion carrier density. The $\mathbf{k}\cdot\mathbf{p}$ NEGF shows a density that is lower than the one obtained with the atomistic NEGF. This may be due to the fact that the $\mathbf{k}\cdot\mathbf{p}$ NEGF includes only the Γ valley.

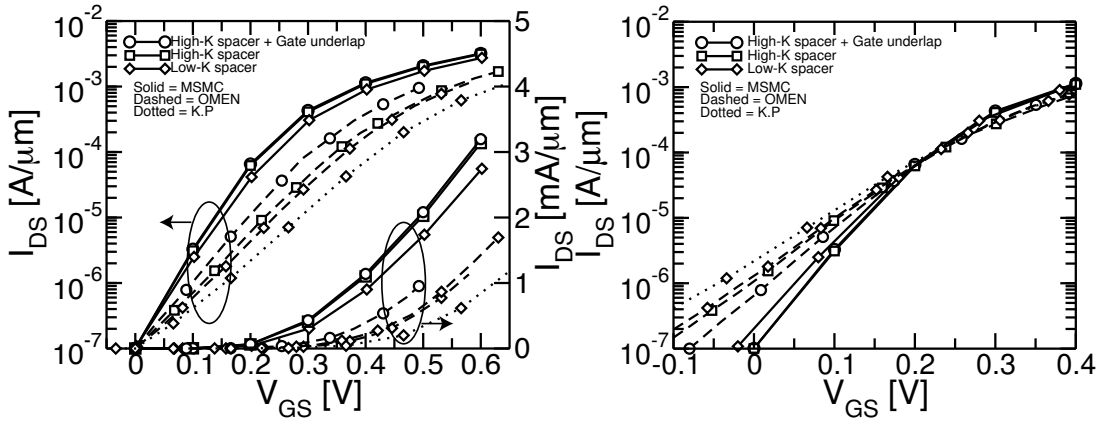


Figure 4.5: Simulated drain current of the device shown in Fig. 4.4. The results obtained by using the MSMC simulator have been compared with the results obtained with an atomistic NEGF and a $\mathbf{k} \cdot \mathbf{p}$ NEGF. (a): the simulations are matched so that the OFF current is 100 nA/μm for a gate voltage of 0.0 V. (b): the simulations are matched so that the same current is obtained at the threshold voltage (0.2 V for the MSMC simulation).

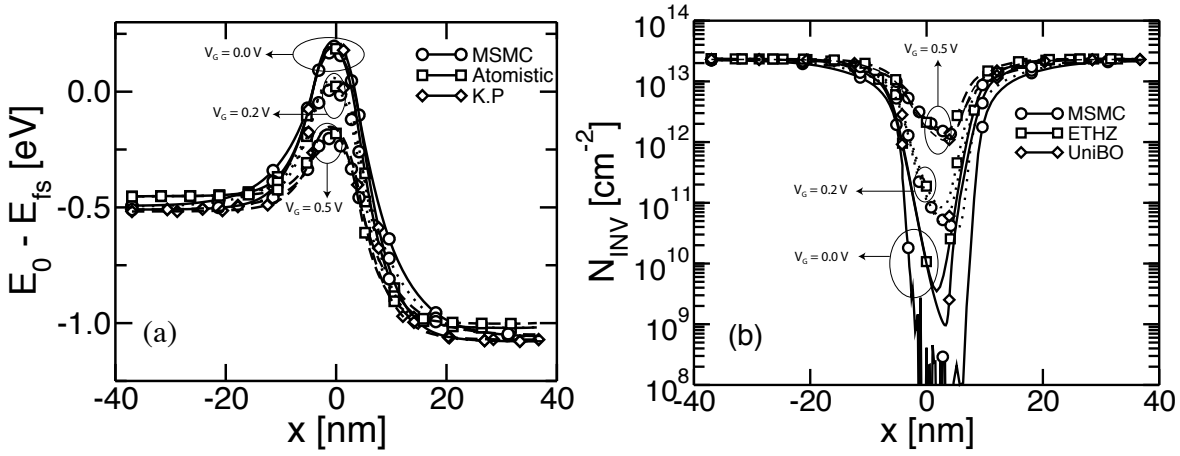


Figure 4.6: Internal quantities for the device shown in Fig. 4.4. (a): Profile of the first subband along the transport direction. (b): Profile of the inversion carrier density along the transport direction.

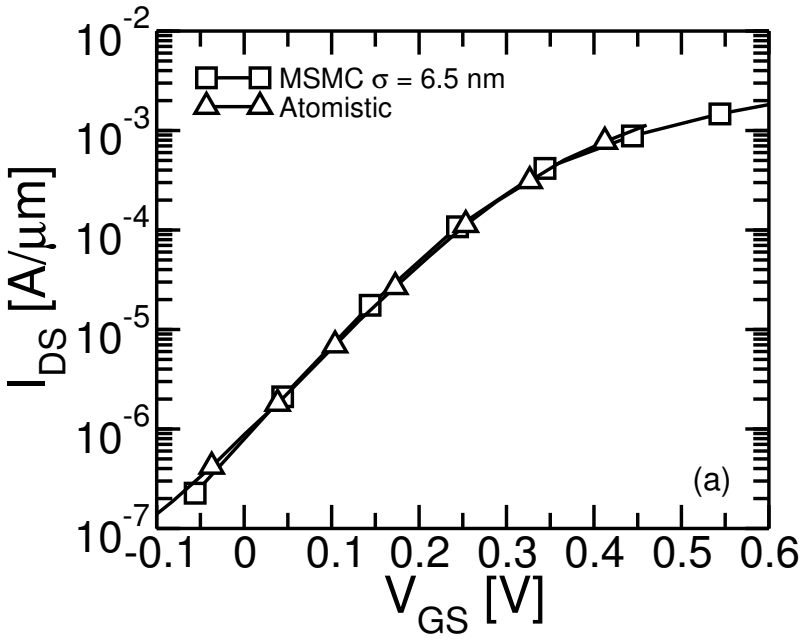


Figure 4.7: Gaussian smoothing applied to the simulation of the device sketched in Fig. 4.4. A σ value of 6.5 nm provides a good agreement with NEGF simulation. In this figure the simulation were matched in order to obtain the same current at the threshold voltage of 0.2 V.

4.3 Mimicking the source to drain tunnelling in the MSMC simulator

Figures 4.2(b) and 4.5(b) show that the source to drain tunnelling must be included in order to reproduce the current in the sub-threshold region. The MSMC simulator mimics this effect by smoothing both the subband profile and the electron concentration along the transport direction x [6]. The smoothed subbands affects the Monte Carlo transport since they are used to compute the force that moves the particles, while the smoothed electron concentration affects the solution of the Poisson equation. The smoothing is implemented as a convolution with a Gaussian function. In the case of the subbands, the smoothed subband profile is given by [6]:

$$\varepsilon_{\nu,n}^{smth}(x) = \int \varepsilon_{\nu,n}(x') \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x'-x)^2}{2\sigma^2}\right) dx' \tag{4.1}$$

σ, the r.m.s. of the gaussian is the only parameter of the model. The convolution is applied to both the parabolic eigenvalues $\varepsilon_{\nu,n}^{P}$ (solution of Eq. 2.5) and to the U factor defined by Eq. 2.8) since both are needed to compute the non-parabolic eigenvalues $\varepsilon_{\nu,n}^{NP}$ (Eq. 2.7). Figure 4.7 shows how the application of the Gaussian smoothing model affects the drain current. A value of 6.5 nm for the σ parameter allows the MSMC to attain good agreement with the atomistic NEGF simulator.

A σ of 1.1 nm was used in [7] for silicon devices, while for $In_{0.53}Ga_{0.47}As$ a value between 6 nm and 6.5 nm is needed. This would suggest that σ depends mostly on the

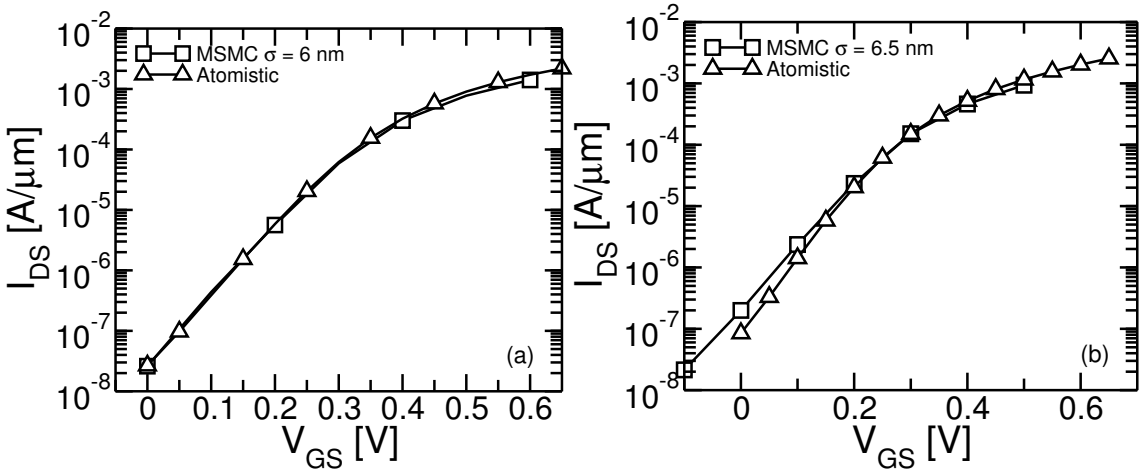
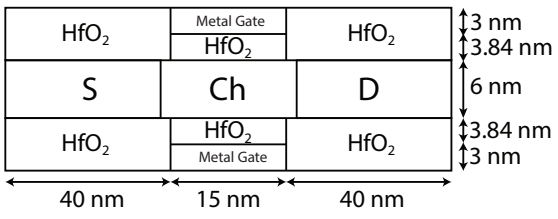


Figure 4.8: Gaussian smoothing applied to the simulation of the devices sketched in Figs. 4.1 (left) and 4.9 (right). A σ value between 6 and 6.5 nm provides a good agreement with NEGF simulation. In this figure the simulation were matched in order to obtain the same current at the threshold voltage of 0.2 V.



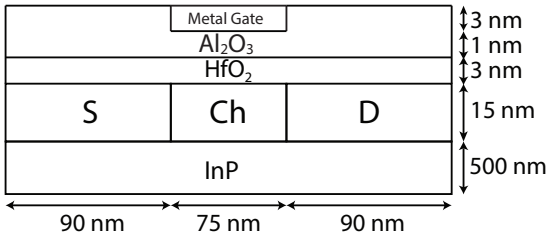
Semiconductor	$In_{0.53}Ga_{0.47}As$
S/D Doping	5×10^{19} cm^{-3} n-type
Ch Doping	1×10^{17} cm^{-3} p-type
V_{DS}	0.63 V
Gate dielectric	3.84 nm HfO_2
Γ valley parms.	$m_l = m_t = 0.043\ m_0$, $\alpha_\Gamma = 1.5$ eV^{-1}
L valley parms.	$m_l = 1.57\ m_0$, $m_t = 0.23\ m_0$, $\alpha_L = 0.5$ eV^{-1}
Δ valley parms.	$m_l = 2.26\ m_0$, $m_t = 0.25\ m_0$, $\alpha_\Delta = 0.5$ eV^{-1}

Figure 4.9: Sketch and parameters of the device with $L_G = 15$ nm used to test the σ dependance on device geometry. There is also a 2 nm gate underlap.

semiconductor material and weakly on the device geometry. To confirm this, we have also applied the smoothing to the simulation of the devices of Figs. 4.1 and 4.9. Results are shown in Fig. 4.8. A σ between 6 and 6.5 nm allows to attain a very good agreement with the atomistic NEGF simulator.

4.4 Realistic InGaAs device with $L_G = 75$ nm

This device is a realistic long channel device, for which experimental data are reported in [8]. Figure 4.10 sketches the device and its main parameters. In this simulation we have considered scattering from polar and non polar phonons, remote phonons, surface roughness, alloy and coulomb centres (only due to doping impurities as interface charges will be addressed in the next chapter). Parameters for these mechanisms were calibrated in [9]. Fig. 4.11 show the simulation results compared with the experimental measurements. The gate work-function has been calibrated in order to attain the same behaviour in the sub-threshold region when considering measurements at $V_{DS} = 0.05V$. There is a divergence in the above-threshold region, which may be due to the lack of series resistances



Semiconductor	$In_{0.53}Ga_{0.47}As$
S/D Doping	2×10^{19} cm^{-3} n-type
Ch Doping	1×10^{15} cm^{-3} p-type
V_{DS}	0.5 V
Gate dielectric	3 nm HfO_2 + 1 nm Al_2O_3
Γ valley parms.	$m_l = m_t = 0.043\ m_0$, $\alpha_{nf} = 1.36\ eV^{-1}$

Figure 4.10: Sketch of the device from [8] as simulated in this work.

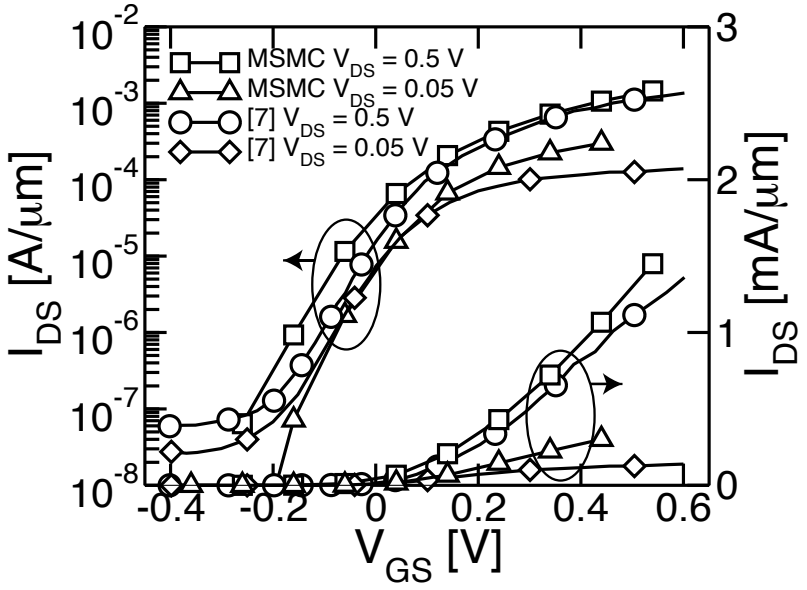


Figure 4.11: Simulation results for the device shown in Fig. 4.10.

in our simulator. The simulation for the $V_{DS} = 0.5V$ case was performed using the same work-function calibrated for the $V_{DS} = 0.05V$ case.

Bibliography

[1] `http://www.itrs.net/Links/2013ITRS/2013Tables/PIDS_2013Tables.xlsx`.

[2] Mathieu Luisier, Andreas Schenk, and Wolfgang Fichtner. "Quantum transport in two- and three-dimensional nanoscale transistors: Coupled mode effects in the nonequilibrium Green's function formalism". In: *Journal of Applied Physics* 100.4 (2006), p. 043713.

[3] Mathieu Luisier, Andreas Schenk, Wolfgang Fichtner, and Gerhard Klimeck. "Atomistic simulation of nanowires in the $sp^3d^5s^*$ tight-binding formalism: From boundary conditions to strain calculations". In: *Phys. Rev. B* 74 (20 Nov. 2006), p. 205323.

[4] E. Baravelli, E. Gnani, R. Grassi, A. Gnudi, S. Reggiani, and G. Baccarani. "Optimization of n- and p-type TFETs Integrated on the Same InAs/$Al_xGa_{1-x}Sb$ Technology Platform". In: *IEEE Trans. on Electron Devices* 61.1 (Jan. 2014), pp. 178–185.

[5] Bradley Foreman. "Elimination of spurious solutions from eight-band $\mathbf{k}\cdot\mathbf{p}$ theory". In: *Phys. Rev. B* 56 (20 Nov. 1997), R12748–R12751.

[6] P Palestri, L Lucci, S Dei Tos, D Esseni, and L Selmi. "An improved empirical approach to introduce quantization effects in the transport direction in multi-subband Monte Carlo simulations". In: *Semiconductor Science Technology* 25.5 (2010), p. 055011.

[7] P. Palestri, L. Lucci, S. Dei Tos, D. Esseni, and L. Selmi. "An improved empirical approach to introduce quantization effects in the transport direction in multi-subband Monte Carlo simulations". In: *Semiconductor Science Technology* 25.5 (2010), p. 055011.

[8] X. Zhou, A. Alian, Y. Mols, R. Rooyackers, G. Eneman, D. Lin, T. Ivanov, A. Pourghaderi, N. Collaert, and A. Thean. "In0.53Ga0.47As quantum-well MOSFET with source/drain regrowth for low power logic applications". In: *IEEE Symposium on VLSI Technology - Technical Digest*. June 2014, pp. 1–2.

[9] D. Lizzit, D. Esseni, P. Palestri, P. Osgnach, and L. Selmi. "Performance Benchmarking and Effective Channel Length for Nanoscale InAs, $In_{0.53}Ga_{0.47}As$, and sSi n-MOSFETs". In: *IEEE Trans. on Electron Devices* 61.6 (June 2014), pp. 2027–2034.

Chapter 5

Modelling the Effects of Interface States

The replacement of silicon with III-V compound semiconductors as channel materials in advanced MOSFETs has been widely investigated over the last years [1, 2]. The surface trap-state density of III-V compounds is much larger than that of state of the art silicon/SiO_2 interfaces [3, 4], and Hall mobility measurements have shown that the charging of these states results in a remarkable Fermi level pinning which precludes attaining a free carrier density larger than $N_{INV} \approx 5 \cdot 10^{12} cm^{-2}$ [5, 6]. This large trapped charge affects the electrostatics but it does not contribute to the drain current. Consequently, one of the basic assumptions at the foundation of split-CV mobility extraction techniques is violated [7].

In this chapter, a self-consistent solution of the Schrödinger and Poisson equations in the presence of interface charge is used to extract the energy profile $D_{it}(E)$ of interface states. The extracted charge is then introduced in our Multi-Subband Monte Carlo (MSMC) simulator (described in chapter 2), both as a source of Coulomb scattering and as a contribution to device electrostatics, to asses its effect on low field electrical mobility and on the drive current, I_{ON}, of short channel devices.

5.1 Interface traps model

Interface states have been introduced in the equilibrium solution of the coupled Schrödinger and Poisson equations as a sheet of charge at the interface between the channel and the gate dielectric. The Schrödinger equation is solved as described in section 2.2 The model is appropriate for near equilibrium conditions to investigate, for instance, MOSFETs biased at low V_{DS} as for low-field mobility measurements. The solution of the Schrödinger equation considers wave-function penetration in the dielectrics, which can be relevant in III-V materials [8].

The interface charge per unit area $Q_{it} = -qN_{it}$ is computed under the following assumptions: a) traps below E_{Cn} are donor-like, i.e., they contribute with a positive charge when empty; b) traps above E_{Cn} are acceptor-like, i.e., they contribute with a negative charge when occupied by an electron; c) the occupation probability $f(E, E_F)$ follows the equilibrium Fermi-Dirac statistics. d) E_{Cn} is assumed to be close to the midgap [7]. Figure

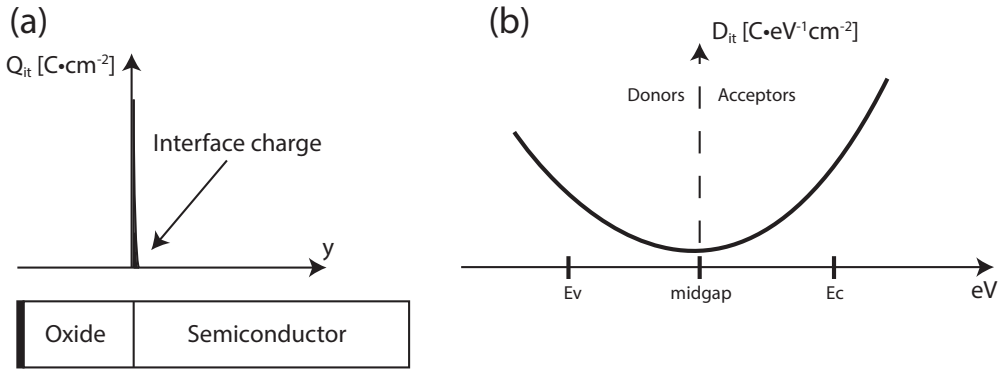


Figure 5.1: (a) Interface charge per unit of area modelled as a Dirac's delta sheet of charge lying at the interface between the dielectric and the semiconductor. (b) Interface states concentration as a function of energy. States below the mid-gap are positively charged when empty, while the states above the mid-gap are negatively charged when occupied. The conduction band minima (E_C) is taken as the reference value.

5.1 shows a sketch of this model. We further choose to express D_{it} as:

$$D_{it}(E) = D_{it}(E_C) \cdot \left(10^{\sum_{i=1}^{n} a_i (E-E_C)^i}\right) \tag{5.1}$$

where E is the state energy. Eq 5.1 is polynomial on a semi-logarithmic scale. The choice of this shape was inspired by the D_{it} profiles shown in [5]. The N_{it} is then given by:

$$N_{it} = -\int_{-\infty}^{E_{Cn}} D_{it}(E) \cdot [1 - f(E, E_F)]dE + \int_{E_{Cn}}^{\infty} D_{it}(E) \cdot f(E, E_F)dE \tag{5.2}$$

where E_F is the equilibrium Fermi level and E_{Cn} is the energy level that separates donor-like from acceptor-like traps. The shape of the D_{it} energy distribution is set by the coefficients a_i. $D_{it}(E_C)$ is the trap concentration at the conduction band edge (E_C) of the semiconductor in units of $eV^{-1}cm^{-2}$.

A correct choice of the $D_{it}(E_C)$, the polynomial degree, n, and coefficients, a_i, is necessary to reproduce experimental N_{INV} and mobility curves [6, 5]. Figure 5.2 compares simulated $N_{INV}(V_{GS})$ curves with experimental data from [6]. Polynomials of different degree can be used to fit the measurements but, after choosing the appropriate coefficients, the trap distributions are very similar as can be seen by looking at the D_{it} energy profiles in the inset. In the following, we have opted for the lowest polynomial degree $n = 2$ that results in a good agreement with the experiments. Also, this choice makes easier the task of finding the optimal coefficients.

To this end, we note that for the sole purpose of finding the best coefficients in Eq. 5.1, traps do not necessarily need to enter the coupled Schrödinger-Poisson problem explicitly; in fact, since wave-function penetration beyond the surface charge layer is modest, self-consistent calculations with traps can be accurately reproduced if the abscissa of a simulated $N_{INV}(V_{GS})$ curve without traps is "stretched" by an amount equal to $-Q_{it}(V_G)/C_{OX}$. Figure 5.3 illustrates the use of this technique. The stretching implies that the same N_{INV} is found at a higher gate voltage in the case with traps.

To extract the D_{it} energy spectrum and determine the fitting parameter values we set up a global optimisation problem whose solution is the set of a_i and $D_{it}(E_C)$ coefficients that

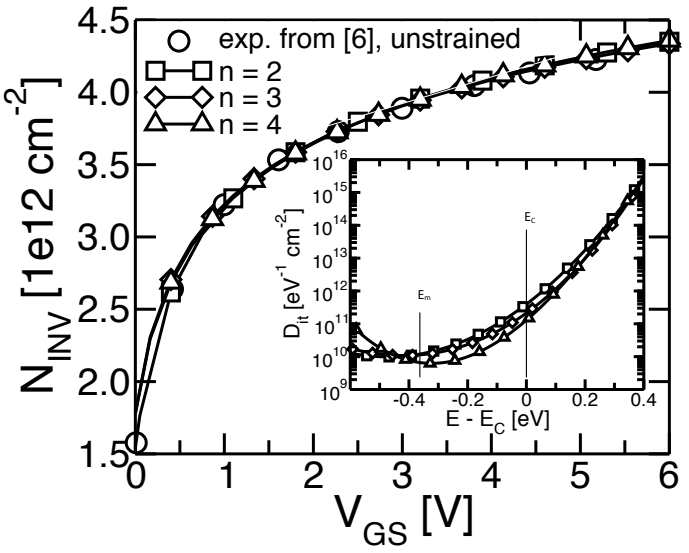


Figure 5.2: Simulated $N_{INV}(V_{GS})$ for different polynomial degree n in Eq. 5.1. The inset shows the extracted D_{it} corresponding to each polynomial function.

yields the best agreement between measured and simulated $N_{INV}(V_{GS})$ curves. In other words, we have to find the correction term $Q_{it}(V_{Gsim})$ that minimises the difference between the experimental and simulated $N_{INV}(V_{GS})$. In formula, the problem is to minimise:

$$\left\| V_{Gexp}(N_{Sexp}) - \left(V_{Gsim}(N_{Ssim}) - \frac{Q_{it}(V_{Gsim})}{C_{OX}} \right) \right\|_2^2 \tag{5.3}$$

where $V_{Gsim}(N_{Ssim})$ is obtained from simulations without interface traps. $Q_{it}(V_{Gsim})$ is obtained from Eq. 5.2 by replacing E_F with the Fermi level obtained from the simulation without interface traps. The search for the solution of the minimisation problem (that is, the sought set of $D_{it}(E_C)$ and $a_1, \ldots, a_n$) must be adequately constrained, otherwise the result may still fit the experiments but with an unrealistic D_{it} profile. In particular, based on results in [5], we expect D_{it} profiles with exactly one minimum at a specific energy (e.g. the midgap) and no maximum. It is not always straightforward to satisfy these requirements, but if the polynomial is a second order one, we just need to enforce a_2 to be positive. Note that, since the D_{it} is fitted on Fermi level pinning experiments, it may end up being inaccurate in the gap. We will return later on this point.

Note that the stretching technique described so far is used only for the purpose of finding the optimal coefficients of Eq. 5.1, whereas in the other calculation of this chapter, the full self-consistent problem is solved.

5.2 Results: Fermi level pinning and D_{it} profiles

We define the Fermi level pinning as the condition at which an increase of the gate voltage corresponds to a very small increase of the Fermi level with respect to the minimum of the conduction band, and we consider MOSFETs at equilibrium ($V_{DS} = 0V$), consistently with the bias used during Hall mobility measurements and $N_{INV}(V_{GS})$ extraction. Fig. 5.5 reports a few of the $N_{INV}(V_{GS})$ as measured by different groups [5, 6, 9]. Figure 5.4 shows

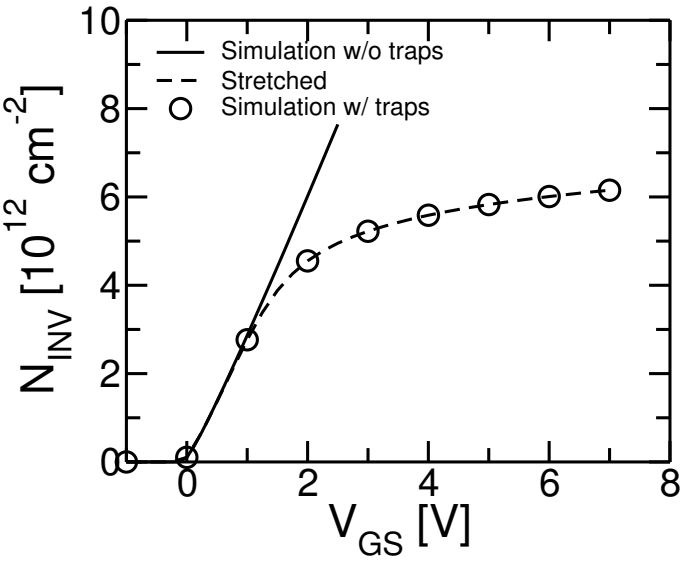


Figure 5.3: Illustration of the stretching technique used to fit the free carrier density versus gate bias curves. First, an $N_{INV}(V_{GS})$ curve is simulated without considering the effect of interface traps (solid line); then, this curve is stretched by $-Q_{it}(V_{Gsim})/C_{OX}$ (dashed line). The circles represent $N_{INV}(V_{GS})$ simulations with the full self-consistent loop considering the effect of interface traps. The agreement between the dashed curve and the symbols proves the validity of the proposed extraction algorithm.

the sketches and data for the devices we have simulate in order to replicate the results of these measurements. The curves saturate at high gate voltage, indicating Fermi level pinning. We express D_{it} as in Eq. 5.1 and determine the coefficients $D_{it}(E_C)$, a_1 and a_2 by solving the minimisation problem of Eq. 5.3 obtaining a good match with the experiments.

Each data set saturates at a different free carrier density, therefore in principle each set has a different D_{it} profile. Fig. 5.6 reports the $D_{it}(E)$ over the energy range spanned by the Fermi level when V_{GS} spans from approximately 0V to the maximum V_{GS} in the experiments (see Fig. 5.5). Interestingly, the spread between the various D_{it} profiles is not too large for the considered $In_{0.53}Ga_{0.47}As/Al_2O_3$ devices, demonstrating a comparable degree of maturity in the fabrication process. However, given the exponential increase of $D_{it}(E)$, small horizontal shifts of energy have non-negligible consequences on the simulation results. As expected, higher N_{INV} saturation values, as observed for instance in strained samples, correspond to lower trap densities (triangles up in Fig. 5.6). We also see that the D_{it} which reproduces the data in [5] are lower than for the other cases; this may be related to the much thicker oxide used in [5] (16 nm of Al_2O_3) with respect to the other works [6, 9].

The D_{it} profiles diverge significantly from the distributions shown in [5] and [10] when looking inside the band-gap. In fact, our extraction method based on the free carrier density above threshold is not so accurate in the gap region because those traps have a negligible effect on the Fermi level pinning. Therefore, Fig. 5.6 shows the D_{it} in the energy range actually covered by the experiments as solid lines, while the dashed lines are just an extrapolation that follows the functional form given by Eq. 5.1. However, trap density as low as extracted in Fig. 5.6 is not completely unrealistic. D_{it} measurements reported in

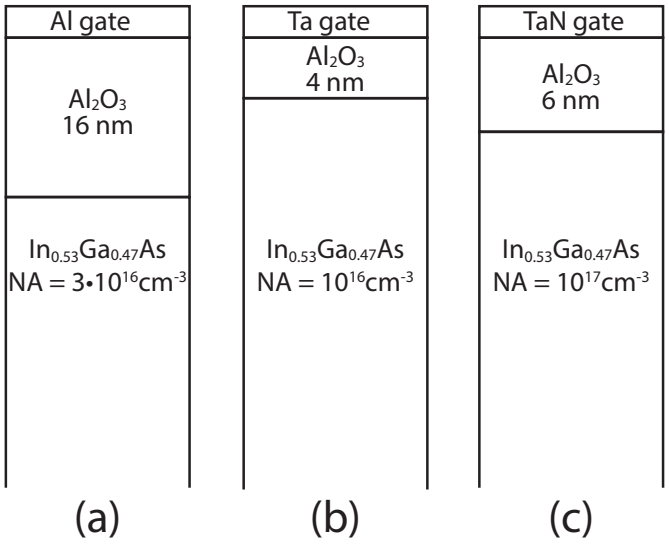


Figure 5.4: Sketches of the devices we have simulated in order to replicate the Fermi level pinning effects on N_{INV} measurements from [5](a), [6](b) and [9](c).

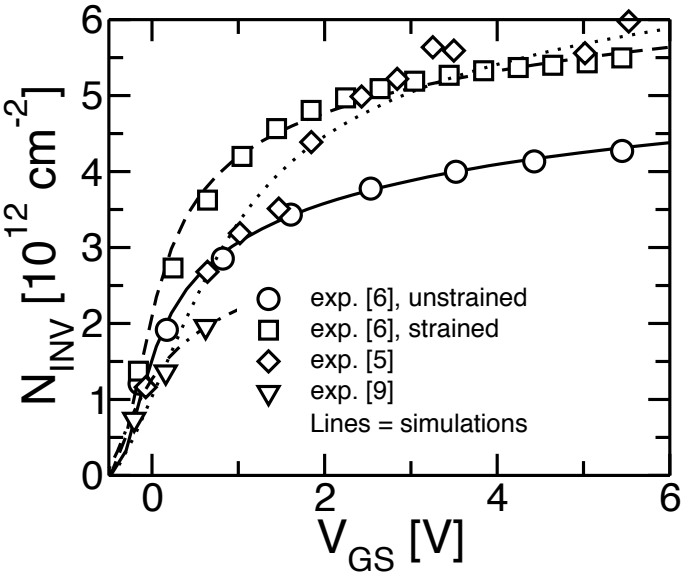


Figure 5.5: Simulated (lines) and experimental (symbols) free carrier density versus gate voltage. The corresponding D_{it} profiles are reported in Fig. 5.6.

[11] show values around $10^{11} eV^{-1} cm^{-2}$ for an $HfO_2/In_{0.53}Ga_{0.47}As$ interface. It has been shown that in III-V semiconductors E_{Cn} is closer to E_C [10] rather than the midgap. Since the D_{it} is so low in the gap, the choice of E_{Cn} is not so critical and had a negligible effect on the results.

5.2.1 Capacitance computation

The capacitance is computed during the solution of the coupled Schrödinger equation-Poisson problem described in section 2.6 according to:

$$C = -\frac{d(Q_S + Q_{it})}{dV_G} \tag{5.4}$$

where $Q_S = Q_n + Q_p$ is the total semiconductor charge. Q_n and Q_p are respectively the electrons and holes charges and are computed as:

$$Q_n = -e \int n(y) dy \tag{5.5a}$$

$$Q_p = e \int p(y) dy \tag{5.5b}$$

where the position-dependant electron and hole concentrations are given by Eqs 2.86 and 2.87, respectively.

The traps in the gap have a much larger impact on the depletion region of the CV curve. This is illustrated in Fig. 5.7 which compares our model with the CV measurements in [10]. Good agreement is obtained by using the corresponding D_{it} profiles indicated by squares in Fig. 5.6. However, if we use a distribution with lower D_{it} in the gap, for example the trap profile given by diamonds of Fig. 5.6 on the one hand, the low trap density in the band-gap prevents us from reproducing the experimental CV in the 0V-1V region; on the other hand, since the profile indicated by diamonds in Fig. 5.6 has a larger D_{it} inside the conduction band than the one indicated by squares, the former gives larger capacitance closer to Cox in strong inversion ($V_{GS} > 1V$). In fact, we are assuming that all traps respond to the AC probing signal and a large trap density short-circuits the inversion charge capacitance, making the total capacitance approaching Cox.

5.2.2 Effects of strain

Data in [6] show also the effects of strain on the free carrier density. From $\mathbf{k}\cdot\mathbf{p}$ calculations we have found that a tensile biaxial strain of 0.46% shifts down the conduction band by approximately 33 meV, similar to the 30 meV shift reported in [6]. Figure 5.8(a) shows the conduction band shift as a E-k plot while Fig. 5.8(b) shows the shift with respect the y direction.

It is assumed in [6] that the energy position of the D_{it} profile with respect to the vacuum level and the profile itself do not change with strain. Here we embrace the same assumption and furthermore we keep the same D_{it} used to fit the unstrained N_{INV} results (diamonds of Fig. 5.6). As can be seen in Fig. 5.9(a) the simulated N_{INV} comes very close to the measurements but an additional modification of the D_{it} profile (Fig. 5.9(b)) has been necessary to better reproduce the experimental $N_{INV}(V_{GS})$ results.

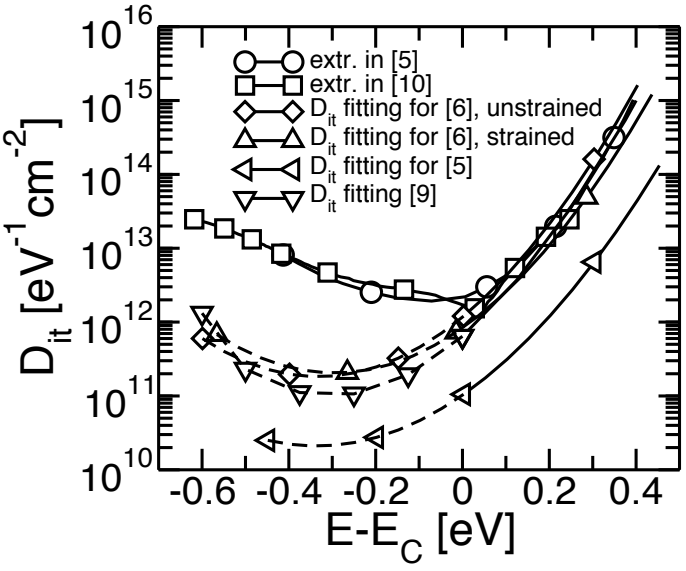


Figure 5.6: Interface trap densities extracted by fitting the experimental $N_{INV}(V_{GS})$ data in Fig. 5.5. Circles and squares indicate the D_{it} extracted in [5] by comparing simulated CV and $N_{INV}(V_{GS})$ curves with experiments and in [10] via the conductance method, respectively. Solid lines are used for the range of energies actually explored by Hall measurements (i.e. the range of $E_F - E_c$ corresponding to the V_{GS} range where Fermi level pinning is observed), dashed lines are extrapolations of solid lines following the functional form given by Eq.5.1.

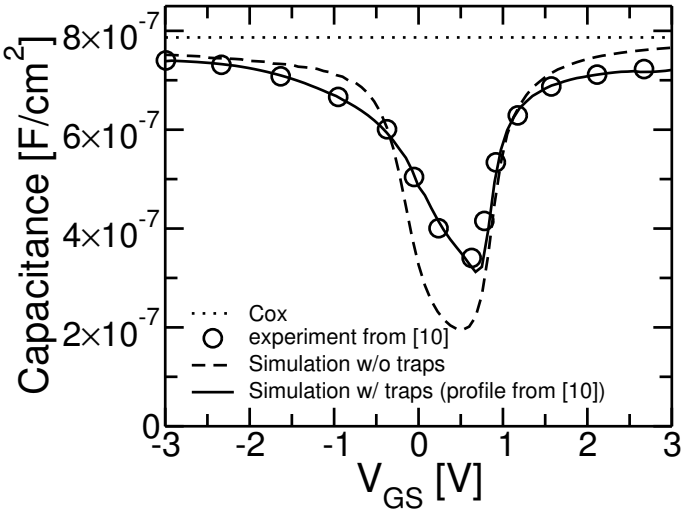


Figure 5.7: CV measurements from [10] compared with simulations with the D_{it} energy profile shown by squares in Fig. 5.6. A lower trap profile in the band-gap (diamonds in Fig. 5.6) fails to reproduce the experiments. The difference at high V_{GS} is due to fact that the trap profiles have been calibrated for different devices and are different inside the conduction band.

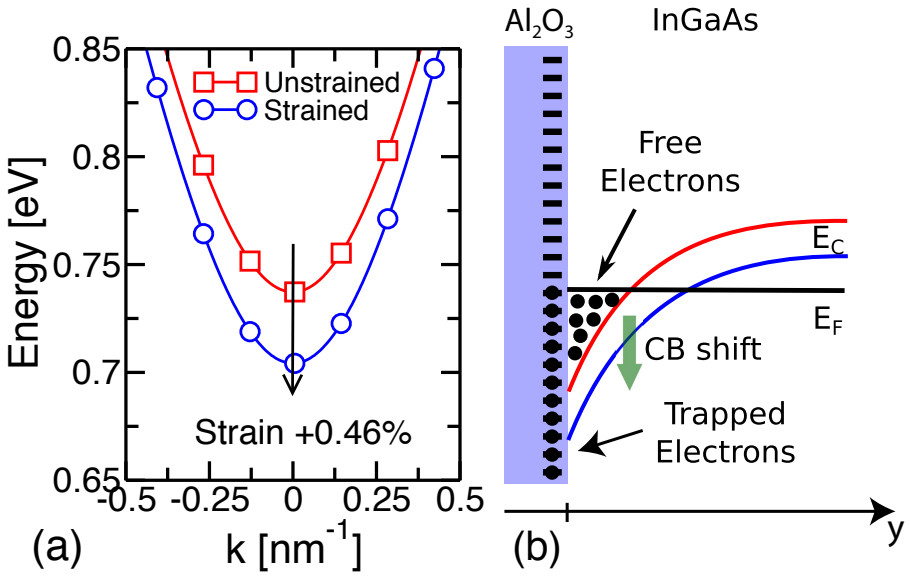


Figure 5.8: Effects of stain on device (b) of Fig. 5.4. (a) Conduction band shift caused by a 0.46% biaxial tensile strain and computed using the **k**·**p** method. (b) Conduction band minimum shift for the same D_{it} will cause an increase of the free electrons concentration.

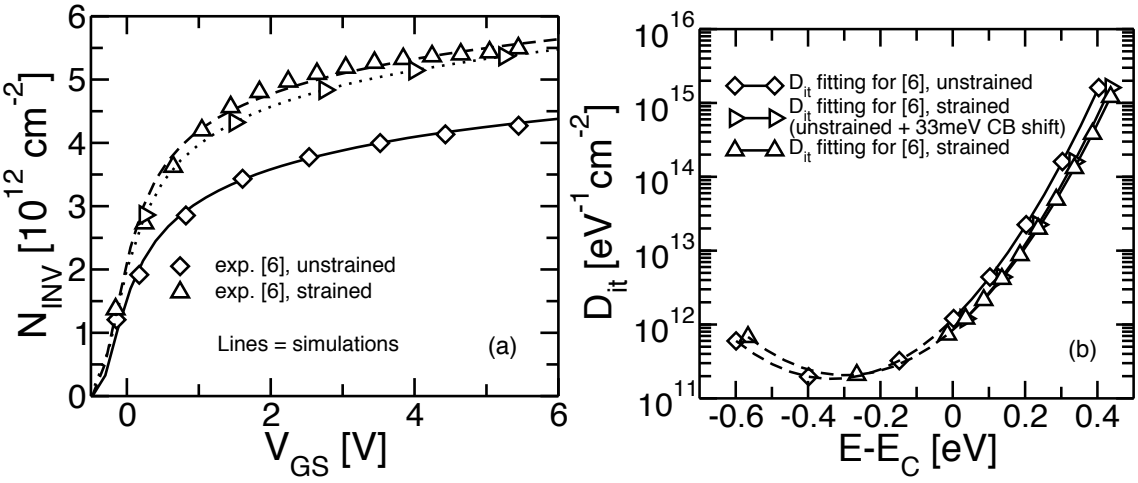


Figure 5.9: (a) Simulated N_{INV} of the devices from [6]. The dotted curve show the simulation of the strained device using the same D_{it} for the unstrained device, but with the conduction band shifted by 33 meV. The energy shift alone is not enough to reproduce the experiment (triangles-up). Dashed line: beside the 33 meV conduction band shift, the shape of D_{it} is modified too. (b) D_{it} profiles used for Fig. (a).

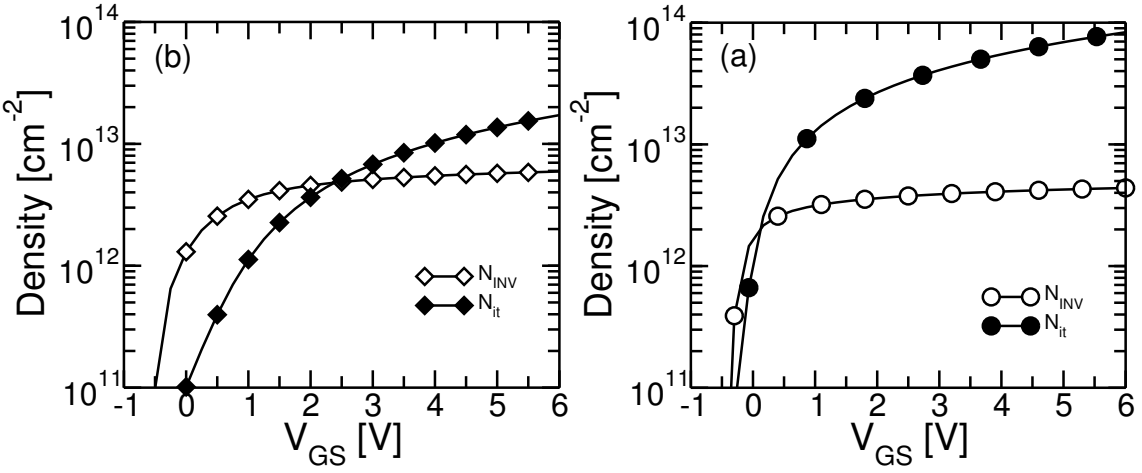


Figure 5.10: Simulated N_{INV} compared to interface trap density N_{it}. Plot (a) is obtained from the simulation of the device (a) of Fig. 5.4 (unstrained case), while plot (b) is obtained from the simulation of the device (b) of Fig. 5.4. Both N_{INV} profiles are the same as in Fig. 5.5.

5.2.3 Trapped charge versus Free charge

Figure 5.10 compares N_{INV} to the (signed) trapped carrier density N_{it}. Note that N_{it} is positive throughout the whole V_{GS} range, indicating that the charge due to occupied acceptor states is dominating. Figure 5.11 shows the position of the Fermi level and the first two subbands versus the gate voltage. As V_{GS} increases, the surface potential is pinned and N_{it} becomes very large, and eventually overcomes N_{INV}. The most effective traps in pinning the potential are those with energy below the Fermi level but above the lowest subband energy E_0. These states are occupied but are also expected to respond very rapidly to the time dependent voltages at the device terminals and thus they may affect the CV curves even at high-frequency. If this is the case, the validity of techniques proposed to compensate the effect of interface states on the mobility is challenged [12, 13] because they assume that interface traps will not respond to high frequency AC probe signal used for CV measurements.

5.3 Mobility model

To assess the impact of interface states on the mobility and on the drive current of short channel devices, I_{ON}, we used the Multi-Subband Monte Carlo simulator described in chapter 2.

For mobility simulations, low field conditions are assumed. The potential energy profile in the quantisation direction does not change much along the transport direction, so only one section is considered. The potential energy profile is obtained as described in section 2.6 and is kept frozen through the simulation. The scattering rate parameters for both mobility and I_{ON} simulations are reported in [14] and have been calibrated on

[0] N_{it} is a signed quantity as a direct consequence of Eq. 5.2. Positive N_{it} indicates that there are more occupied acceptor states than free donor states. A simple multiplication with $-e$ yields the correct Q_{it}.

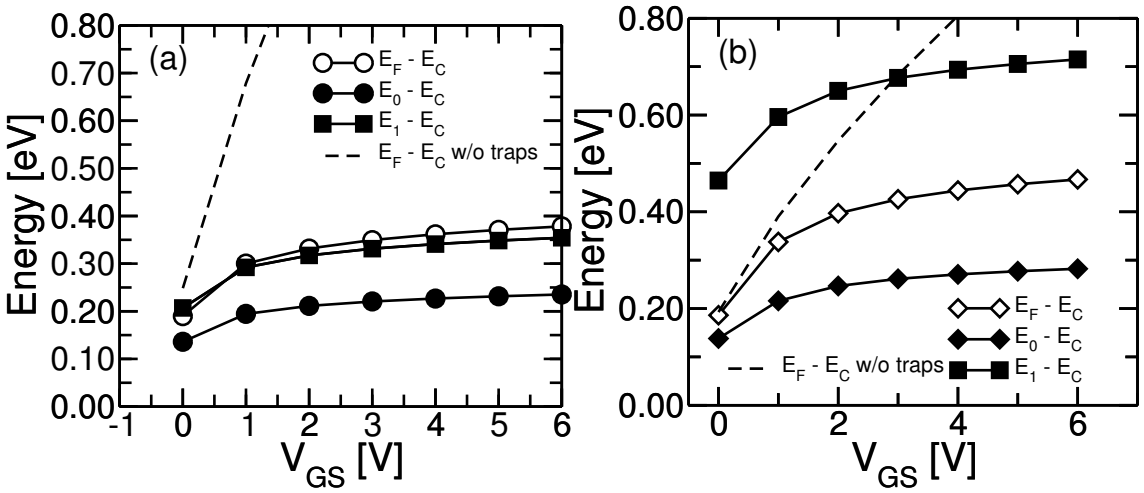


Figure 5.11: Fermi level (E_F), first two subband (E_0 and E_1) referred to the conduction band edge (E_C). Plot (a) is obtained with the D_{it} profile that fits the $N_{INV}(V_{GS})$ from [6] (unstrained case), while plot (b) with the one that fits the $N_{INV}(V_{GS})$ from [5]. Dashed line shows the Fermi level with respect to the conduction band edge for a simulation without traps.

experimental mobility data for $In_{0.53}Ga_{0.47}As$. Fig. 5.12 compares our mobility simulations with experimental data from [12] and [13] and shows the great impact that surface roughness scattering has on the mobility of UTB devices. In the surface roughness scattering model employed by the MSMC simulator the matrix element is proportional to the wave function derivative at the oxide interfaces and the wave function is allowed to penetrate inside the oxides. The simulator uses the non-parabolic Effective Mass Approximation model for the $In_{0.53}Ga_{0.47}As$ band structure, whose parameters are: $m_\Gamma = 0.043m_0$ and $\alpha_\Gamma = 1.36\ eV^{-1}$ for the unstrained case, m_Γ (transport plane) $= 0.0421m_0$, m_Γ (quantisation direction) $= 0.0384m_0$ and $\alpha_\Gamma = 1.4\ eV^{-1}$ for the strained case.

For short device simulations, the effect of interface states poses an additional modelling challenge. In fact, equilibrium or near-equilibrium conditions were assumed so far, as can be seen from the use of f in Eq. 5.2. This model must be adapted for an out of equilibrium condition. For sake of simplicity, we have used the same expression as in the previous analysis, but replacing the equilibrium Fermi level E_F with an "effective" Fermi level E_{Feff}. This effective Fermi level is computed by solving Eq. 5.6 for E_{Feff} in each section:

$$N_S(x) = \sum_{i \in subband} \int_{E_i}^{\infty} Dos(E) \cdot f(E, E_{Feff}(x)) dE \tag{5.6}$$

which means finding an effective Fermi energy such that the right hand side of Eq. 5.6 yields the same N_{INV} as the one computed by the Monte Carlo transport model. Once $E_{Feff}(x)$ has been computed, N_{it} is obtained as:

$$N_{it}(x) = -\int_{-\infty}^{E_{Cn}} D_{it}(E) \cdot [1 - f(E, E_{Feff}(x))] dE + \int_{E_{Cn}}^{\infty} D_{it}(E) \cdot f(E, E_{Feff}(x)) dE \tag{5.7}$$

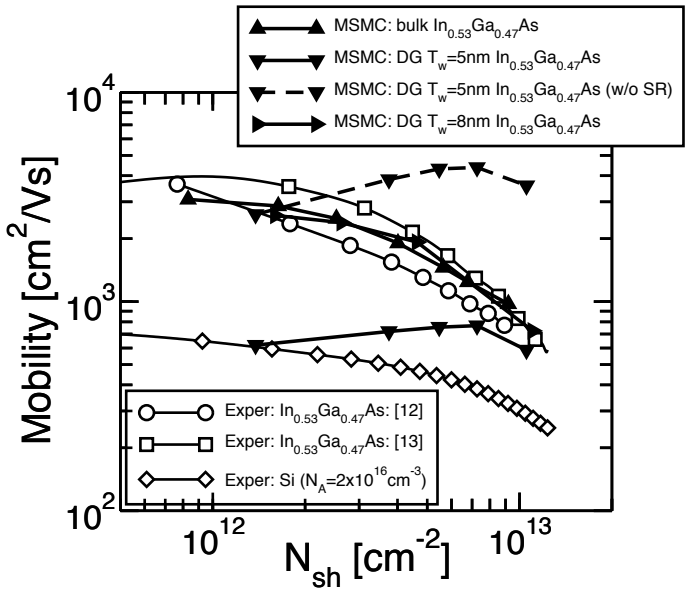


Figure 5.12: Measured and simulated mobility versus N_{INV}. Good agreement is obtained after careful calibration of scattering parameters.

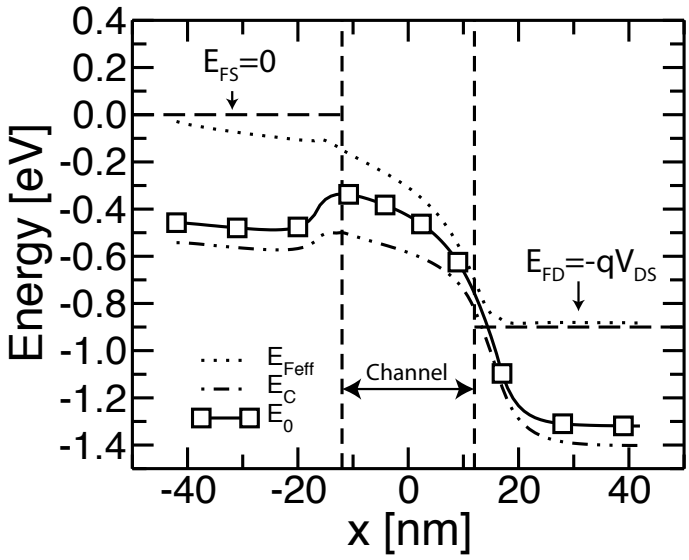


Figure 5.13: Effective Fermi level computed for the template device of Fig. 5.14, profile of the conduction band at the semiconductor/dielectric interface and the lowest subband along the transport direction. $V_{DS} = 0.9V$, $V_{GS} = 0.9V$. The effects of interface traps is considered only in the channel.

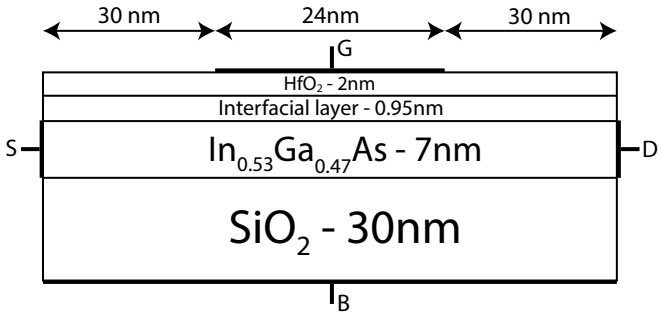


Figure 5.14: Sketch of the simulated short channel device. The device is a single gate fully depleted SOI MOSFET. Channel is p-doped with a doping density of $2 \cdot 10^{16} cm^{-3}$. Source and drain regions are n-doped with a doping density of $3 \cdot 10^{19} cm^{-3}$. Interfacial layer is made of SiO_n with $\epsilon_r = 7.0$. Figure 5.13 shows the effective Fermi level obtained from the simulation of this device.

Fig. 5.13 reports an example of effective Fermi level plotted along the channel. As expected, it changes smoothly from the source to the drain equilibrium Fermi levels. The difference between E_{Feff} and E_C along the channel implies that, under our modeling assumptions, the concentration of occupied traps is higher near the source side of the channel. The figure also shows the profile of the lowest subband that is always well below the Fermi level. This means that most of the traps are exposed to a large concentration of free electrons with the same energy. Exchanges between traps and free electrons in this energy range are expected to be very fast and the energy distribution of the trapped electrons may deviate from an equilibrium distribution and get close to the distribution of the free electrons in the channel. These exchanges are not modelled in our simulator.

5.4 Results: Mobility

In the following, we study the effects of the interface trapped charge on the Hall mobility, and on the effective mobility extracted from split-CV measurements. In particular, we focus on the data in [5] for an $In_{0.53}Ga_{0.47}As$ MOSFET with a 16 nm Al_2O_3 gate dielectric. Besides the scattering mechanisms described in [15], we include here Coulomb scattering with charged centres of opposite signs due to traps in the high-k dielectric and corresponding to an equivalent interface density of centres N_{fix}= $9 \cdot 10^{12}$ cm^{-2}. In addition we consider Coulomb scattering with the bias-dependent (see Fig. 5.10) charged interface states $-qN_{it}$ (see the bias dependence of N_{it} in Fig. 5.10). The model for Coulomb scattering is strictly valid only if the density of interface states is not too high, so that each Coulomb center acts as an independent source of scattering [16]. However, at large density of Coulomb centres, the single trap scattering potentials overlap, resulting in a less severe mobility degradation [16] than predicted by the model (see section 2.3.3). Concerning N_{fix}, this charge is a fitting parameter to obtain a better agreement with experiments and is not included in the solution of the Poisson equation based on the assumption that positive and negative centres in random positions essentially compensate each other. The same problem (i.e. need to introduce high density of Coulomb centres to match the experimental mobility)

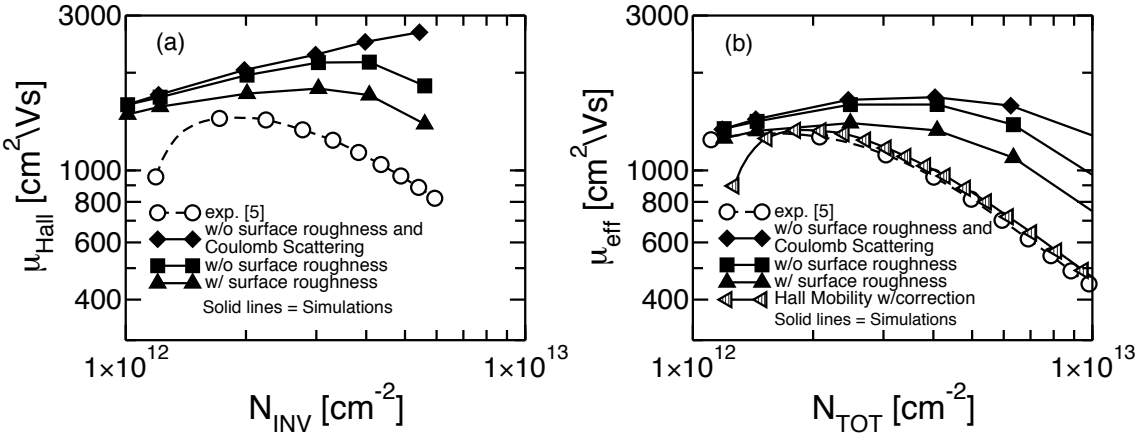


Figure 5.15: (a): Hall mobility μ_{Hall}: comparison between the MSMC model and the experimental data in [5]. (b) Like (a) but for the effective mobility. The slanted triangles are the experimental mobility modified according to Eq. 5.9.

has been reported in previous simulation studies of MOSFETs featuring high-k dielectrics [17]. A detailed discussion of its formulation and implications for high-k dielectrics on top of silicon channels is reported in [18]. We believe that this additional charge is playing the role of neutral defects in the channel or at the interface that are not included in the model and requires us to mimic their effect by increasing the ionised impurity scattering.

5.4.1 Hall mobility

The Hall mobility differs from the effective mobility even in the absence of interface states [7] but the exact calculation of μ_{Hall} according to its definition [7] requires 3D simulations in real space whereas our MSMC is 2D in real space [19]. To overcome this difficulty, in this paper we denote as simulated *Hall mobility* the mobility computed assuming that N_{INV} is known. This assumption is not always verified because the N_{INV} obtained from split-CV measurement may include a contribution from the interface traps. Traps partially responds to the AC signal and therefore they contributes to the capacitance, but negligibly to the current. Consistently with the discussion above, in the MSMC simulations of the Hall mobility we set a lateral field F_x, we compute the average free carrier velocity $\langle v_x \rangle$ and then derive:

$$\mu_{Hall} = \frac{\langle v_x \rangle}{F_x} \tag{5.8}$$

Comparison between measured and simulated Hall mobility according to this definition is reported in Fig.5.15a. As anticipated when discussing Fig. 5.10, due to the increased N_{it} at large bias, we observe a limited mobility roll-off even without surface roughness scattering despite the counteracting effect of screening. When surface roughness mechanism is active instead, the main trends and the value of the experimental mobility are reproduced with reasonable accuracy.

5.4.2 Effective mobility

As for the effective mobility, we adopt an empirical correction to mimic the limitations of the split-CV method in extracting $\langle v_x \rangle$ from the ratio of the drain current per unit of width to the gate charge per unit of area. In particular, since for mobility extraction from split-CV measurements the charge is obtained by integration of the gate differential capacitance, which includes the contribution of the traps that can respond to the AC signal, we multiply the MSMC value of $\langle v_x \rangle$ by the ratio between the free charge and the total charge N_{TOT}. Thus, we evaluate

$$\mu_{eff} = \frac{\langle v_x \rangle}{F_x} \cdot \frac{N_{INV}}{N_{INV} + N_{it}^*} = \mu_{Hall} \cdot \frac{N_{INV}}{N_{INV} + N_{it}^*} \tag{5.9}$$

where N_{it}^* is the trap population that responds to high frequency AC signals in the CV measurements and $\langle v_x \rangle$ is the same as in the Hall mobility calculations. Following a similar reasoning we transform the x-axis of the $\mu - N_{INV}$ plot from N_{INV} to $N_{TOT} = N_{it}^* + N_{INV}$. N_{it}^* is estimated as the fraction of traps with energy above the lowest subband energy but different physically reasonable choices yield essentially the same qualitative results (Fig. 5.16) [20]. Figure 5.17 shows the ratio between N_{INV} and the total charge $N_{INV}+N_{it}^*$ as obtained from our model and as extracted by comparing Hall and CV experiments in [5]. The triangles of Fig. 5.17 were obtained by following a simple procedure: 1) choose an experimental N_{INV} obtained by split-CV measurements, 2) find the corresponding gate voltage, 3) find the N_{INV} obtained by hall measurements for that gate voltage, 4) divide the two N_{INV}. The mutual agreement is more than satisfactory. The comparison between measured [5] and simulated μ_{eff} is then reported in Fig. 5.15b. The mutual agreements is essentially as good as for the Hall mobility. We notice a small roll-off at high N_{INV} even when both Coulomb scattering due to trapped charge and surface roughness scattering are off. This is an artefact due to the charge in the traps that in our model is assumed to respond to the CV. In the experiments this charge cannot be distinguished from free charge and this results in an underestimation of the mobility at high N_{INV}.

It is also interesting to note that if the same correction used for the simulated Hall mobility is applied to the measured Hall mobility (slanted triangles in Fig. 5.15b), the resulting curve gets very close to the experimental effective mobility. The corresponding Hall coefficient $r_H \simeq 1.05$ is close to expectations. The analysis in Fig. 5.15b supports the view in [5] that the traps in the conduction band respond to the split-CV measures introducing an error in the experimental extraction of the effective mobility at high N_{INV}.

5.4.3 Interface vs. border traps: effect of trap position

So far, we have assumed that the trapped charge is placed at the interface between the semiconductor and the dielectric. However, border traps are present in the dielectric and may contribute to Fermi level pinning as well as respond to fast CV [21, 22]. To check the impact of the trap position on the model results, we have considered a limiting case where the sheet of charge is at the position y_{trap} with respect to the semiconductor/dielectric interface (set at $y = 0$). This displacement cannot be too large, otherwise traps do not respond to the high frequency CV experiments used for mobility measurements; we therefore tentatively set $y_{trap} = -0.5$nm, a reasonable value to represent the combination of a sheet of charge that is the combination of fast interface and border traps. Traps deeper

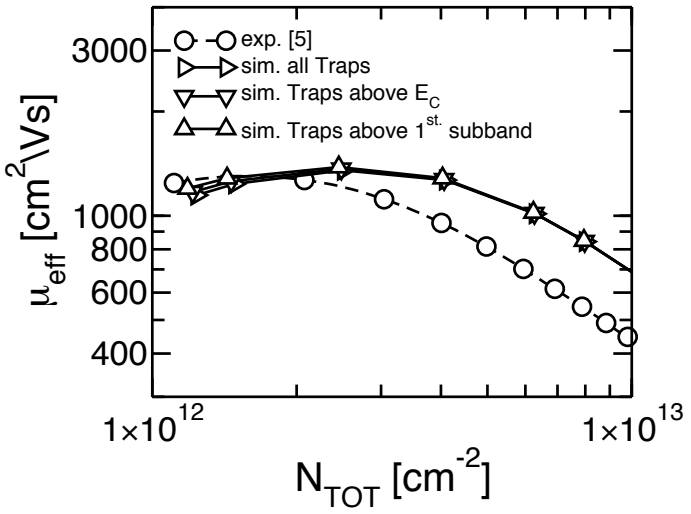


Figure 5.16: Effective mobility μ_{eff}: comparison between the MSMC model calculations according to different assumptions on the fraction of interface traps responding to the AC signal during CV measurements and the experimental data in [5]. Surface roughness is activated.

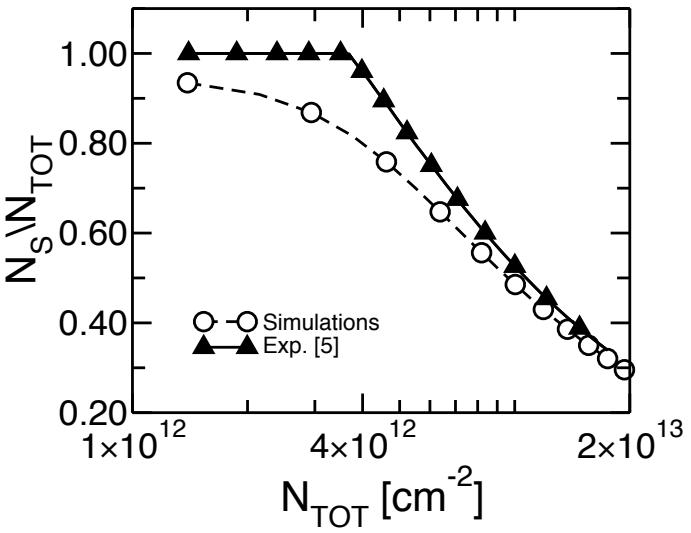


Figure 5.17: Ratio between the free carriers density N_{INV} and the total carrier density (free plus trapped charge) as obtained with our model for the same device as in [5] (circles) and as extracted from the comparison between Hall and CV experiments in [5]. The D_{it} profile is the one indicated by left pointing triangles in Fig. 5.6.

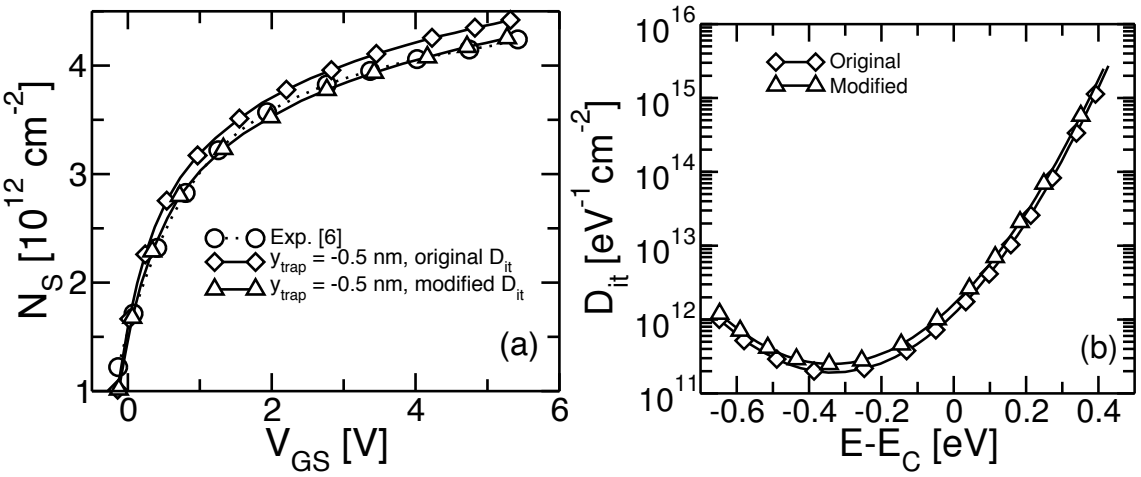


Figure 5.18: (a) If the traps are positioned 0.5 nm away from the interface and inside the dielectric, the resulting N_{INV} (diamonds) does not match the experiment (circles) anymore if the original D_{it} is used (diamonds of Fig. b). A small change of the D_{it} (triangles-up of Fig. b) is needed in order to fin the experiment again (triangles-up). (b) D_{it} profiles used in Fig. a.

into the dielectric are not considered here because they are not expected to affect the split-CV used used in mobility measurements, although they are important when trying to reproduce CV experiments at different frequencies [23].

The effect of a charge displacement by y_{trap} on the electrostatics is modest. In fact, traps shift the V_{GS} by $qN_{it}(t_{ox} - |y_{trap}|)/\epsilon_{ox}$ [1]. This shift has an impact only for small t_{ox}. For a 4 nm dielectric as in [6], a modest adjustment of the D_{it} profile with respect to the case $y_{trap} = 0nm$ is sufficient to reproduce the experimental $N_{INV}(V_{GS})$ (Fig. 5.18). This results in a slightly larger N_{it} (about 10%) for given N_{INV} with respect to the case $y_{trap} = 0nm$ (Fig 5.19). For a 16 nm dielectric as in [5], we found that the D_{it} profile and the resulting $N_{INV}(V_{GS})$ curve are essentially the same for $y_{trap} = 0nm$ and $y_{trap} = -0.5$ nm (Fig. 5.20).

Although the trap position has a modest influence the electrostatics, it affects the Coulomb-limited mobility. In Fig. 5.21 we show the mobility without surface roughness when the trapped charge is located at $y_{trap} = -0.5$ nm. The N_{it} values are the same as in the case $y_{trap} = 0nm$. As one can see (compare filled and slanted squares), the mobility roll-off at high N_{INV} caused by the trapped charge is slightly smaller when traps are at -0.5 nm. Results at low N_{INV} are not affected by y_{trap}, because N_{it} in this range is small and N_{fix} is right at the interface ($z = 0$ nm) in both cases.

5.5 Impact of traps on the I_D of short channel devices

In this Section we investigate the effect of trapped charges on the static drain current of a short channel device. The carrier distribution in the semiconductor significantly deviates from the equilibrium Fermi-Dirac distribution and one cannot simply replace the Fermi-Dirac distribution in Eq. 5.7 with the distribution function computed by the MSMC

[1] This shift is obtained by applying Gauss' law

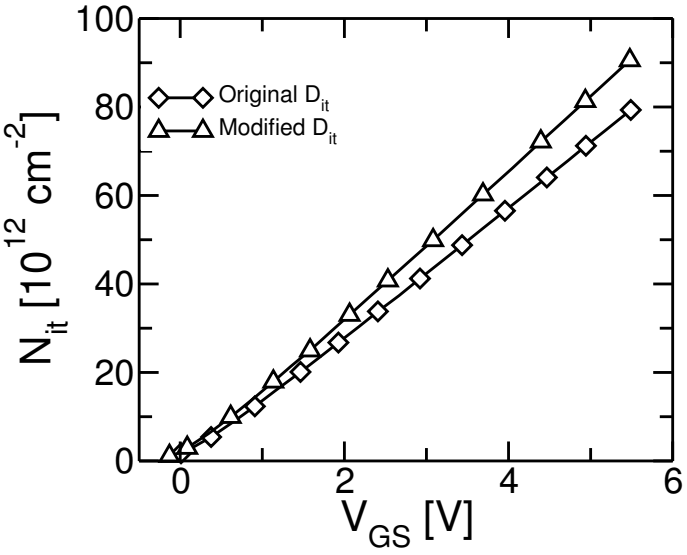


Figure 5.19: Moving the traps 0.5 nm away from the interface and inside the dielectric requires a change of the D_{it} in order to fit again the experiment. This causes a change of the N_{it}. The D_{it} are shown in Fig. 5.18b.

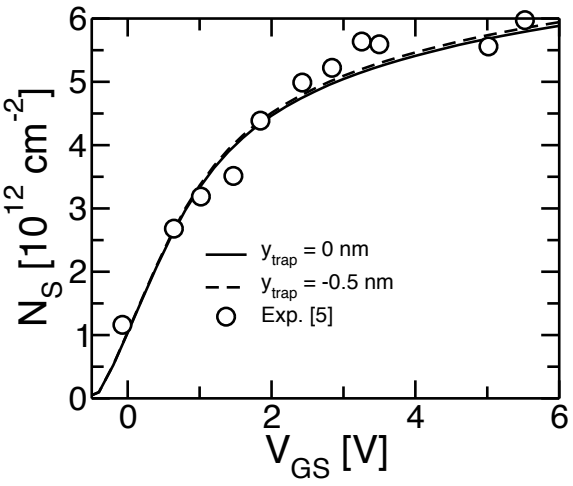


Figure 5.20: If we consider the experimental data from [5], a displacement of the traps of 0.5 nm away from the interface and inside the dielectric has a negligible effect on the N_{INV}.

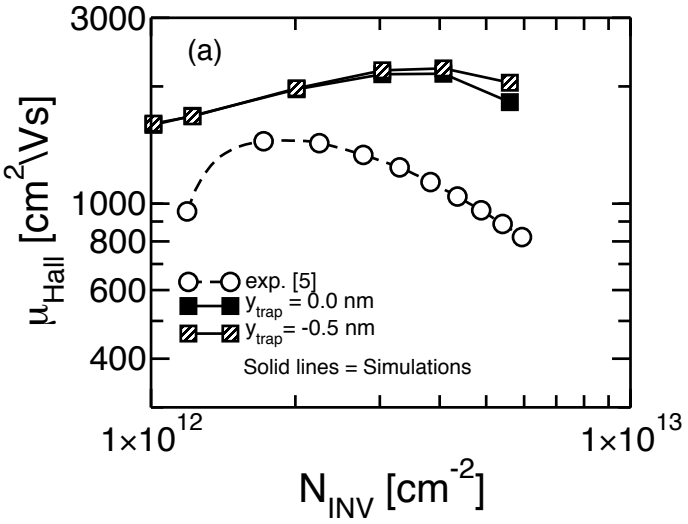


Figure 5.21: Hall mobility simulations without considering the surface roughness scattering. If traps are moved 0.5 nm away from the interface and inside the dielectric, the mobility roll-off caused by Coulomb scattering is reduced.

(see Eq. 3.4) unless elastic trapping-detrapping processes are much more efficient than capture-emission processes among traps. Traps occupancy out of equilibrium has thus been modelled as described in Section 5.3; in particular, N_{it} is given by Eq. 5.7. Note that the occupation function computed by the MSMC is, generally speaking, different from the f used in Eq. 5.7. The two occupation functions coincide only at equilibrium. To asses the impact of interface traps on the I_D, we have simulated two devices: the first is sketched in Fig. 5.14 while the second is device #3 of Fig. 3.2. Having in mind the limits of this analysis, Fig. 5.22 reports the drain current versus the gate voltage for the first device at $V_{DS} = 0.9V$.

First, a trap distribution that pins the Fermi level at a free carrier density of about $6 \cdot 10^{12}$ cm^{-2} is considered. This is the same kind of pinning reported in [6] for the strained device (squares of Fig. 5.5). Since the device is not exactly the same of Fig. 5.5, the trap distribution is slightly different from the one used to reproduce the data of [6] and is shown by circles of Fig. 5.25). To determine the correct D_{it} we have performed a set of N_{INV} vs V_{GS} simulations at $V_{DS} = 0.0$V and modified the D_{it} until the chosen N_{INV} pinning was obtained. The N_{INV} was measured at the center of the channel. Results of this operation for the first device (Fig. 5.14) are shown in Fig. 5.24(a). The interface states have a marginal effect on the current (compare circles in Fig. 5.22 to the result without traps). The reason for the small effect is that there are many interface states at high energy, but the gate voltage is not large enough to populate them. In fact, Fig. 5.13 shows that $E_{Feff} - E_C$ in the channel is small and this difference decreases towards the drain region. Also, the impact of Coulomb scattering with trapped charge is limited since the "deflection" angles are small and have a small impact on back-scattering [24]. A more pessimistic trap profile (triangles-up of Fig. 5.25) that pins the Fermi level at a free carrier density of about $3 \cdot 10^{12}$ cm^{-2} produces a more significant saturation that limits the drain current to $1.7mA/\mu m$. Since the D_{it} profiles used in these simulations fall down to very low values inside the gap (as the ones in Fig. 5.6) the subthreshold slope is not affected by

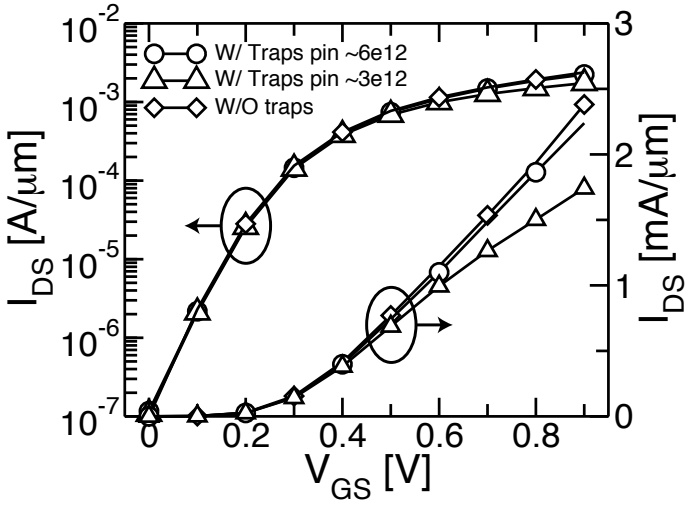


Figure 5.22: Drain current versus gate voltage curves for the device in Fig. 5.14 using trap profiles that produce different amount of Fermi level pinning. $V_{DS} = 0.9V$.

the traps. To investigate this point, we report in Fig. 5.23 the simulated IV characteristics of the same device in Fig. 5.14 using the trap distribution given by the squares of Fig. 5.25, consistent with the values from [10] and [5]. We see that the subthreshold slope is degraded from 73.3 mV/dec to 90.8 mV/dec as expected. Figure 5.26 shows, for this device, the free carrier densities along the transport direction x for various gate voltages. Fig. 5.26b shows a slight decrease of the free carrier density with respect to the simulation without traps (Fig. 5.26a). A more aggressive trap profile pushes the free carrier density further down (Fig. 5.26c) causing a larger decrease of the drain current. Finally, trap profile from [10] (squares of Fig. 5.25) limits the free carrier density even at lower gate voltages, which has an impact on the sub-threshold slope.

Let's consider now the second device (device #3 of Fig. 3.2). The drain currents versus the gate voltage are shown in Fig. 5.27. First, we consider a trap distribution that pins the Fermi level at a free carrier density of about $1 \cdot 10^{13}$ cm^{-2} (squares of Fig. 5.28). To find the appropriate D_{it} we have followed the same procedure performed for the first device and the results are shown in Fig. 5.24(b). Since this is a DG-SOI, we assume that half of this carrier concentration is located near each interface. This kind of pinning has a negligible effect on the drain current (squares of Fig. 5.27). This is confirmed by Fig. 5.29b, which shows a free carrier density far from the pinned concentration. If we increase the trap density in order to pin the free carrier density at about $6 \cdot 10^{12}$ cm^{-2} (circles of Fig. 5.28) we notice a more significant impact on the drain current (circles of Fig. 5.27). Fig. 5.29c shows that the free carrier density is lower with respect to the previous case. This limiting effect is even more pronounced (triangles-up of Fig. 5.27) if we set the trap density to pin the free carrier density at about $3 \cdot 10^{12}$ cm^{-2} (triangles-uo of Fig. 5.28). The effects on the free carrier concentration is shown in Fig. 5.29d.

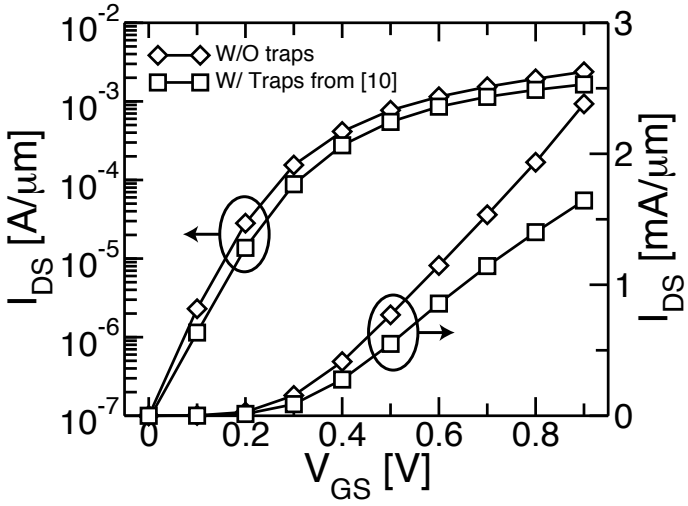


Figure 5.23: Drain current versus gate voltage curves for the device in Fig. 5.14 using the trap profile from [10] (squares of Fig. 5.25). $V_{DS} = 0.9V$. Gate workfunctions have been chosen so that the I_{OFF} is 100 nA/μm for both simulations.

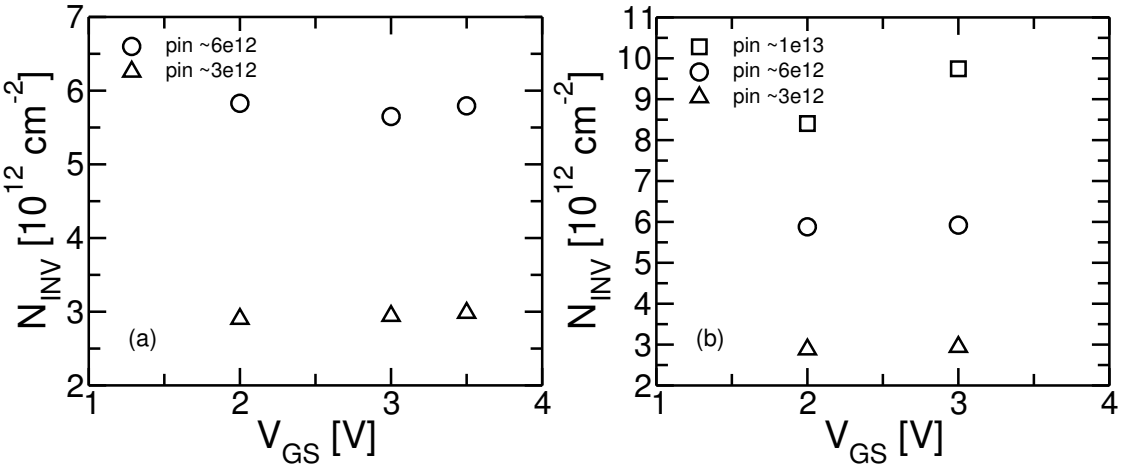


Figure 5.24: (a) N_{INV} pinning for the device of Fig. 5.14 obtained from simulations at $V_{DS} = 0.0$V. The N_{INV} is measured at the center of the channel. (b) Same as (a) but for device #3 of Fig. 3.2.

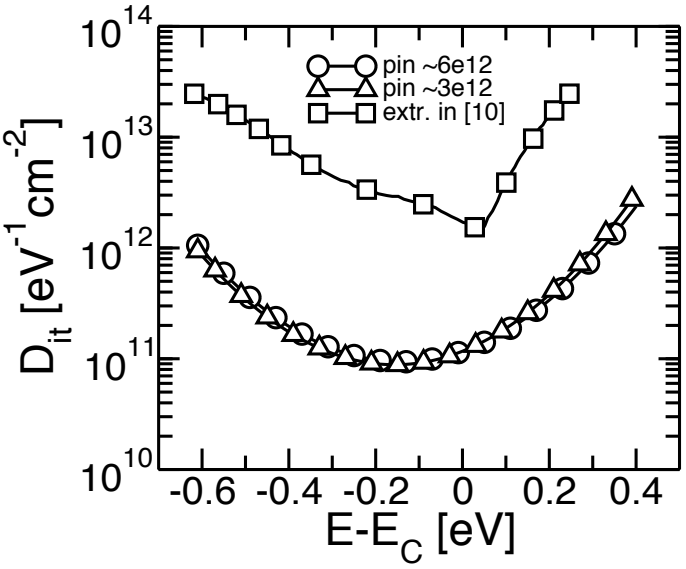


Figure 5.25: Traps energy profiles used for the simulation of the device shown in Fig. 5.14. Simulation results are shown in Figs. 5.22 and 5.23.

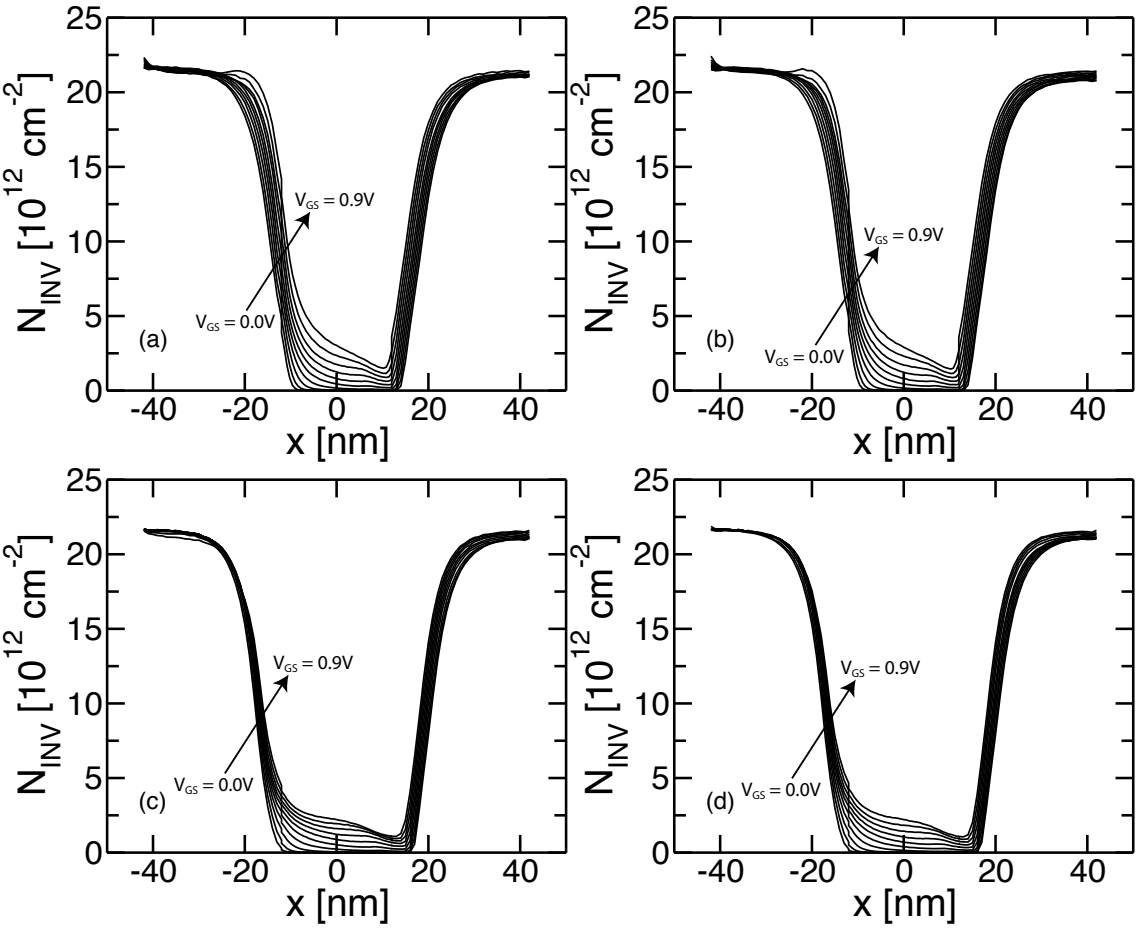


Figure 5.26: Free carrier densities for the device shown in Fig. 5.14. (a) Simulation without traps. (b) Simulation with traps that pin the N_{INV} at $6 \cdot 10^{12}$ cm^{-2}. (c) Simulation with traps that pin the N_{INV} at $3 \cdot 10^{12}$ cm^{-2}. (d) Simulation using the traps profile shown by the squares of Fig 5.25.

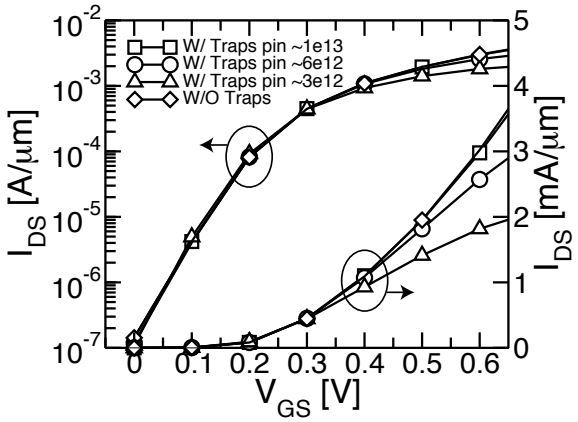


Figure 5.27: Drain current versus gate voltage curves for the device #3 in Fig. 3.2 using trap profiles that produce different amount of Fermi level pinning. $V_{DS} = 0.8V$.

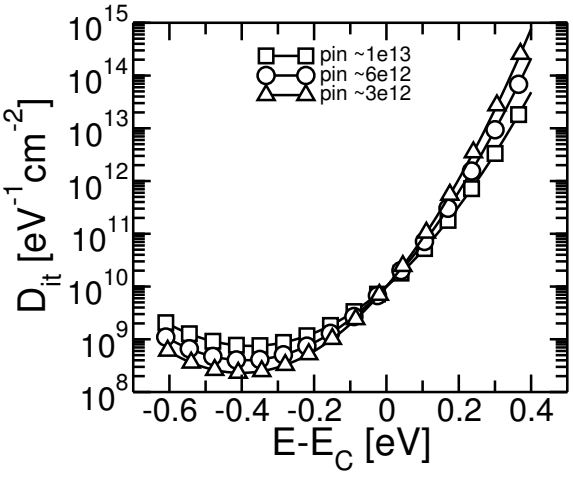


Figure 5.28: Traps energy profiles used for the simulation of the device #3 of Fig. 3.2. Simulation results are shown in Fig. 5.27 and 5.23.

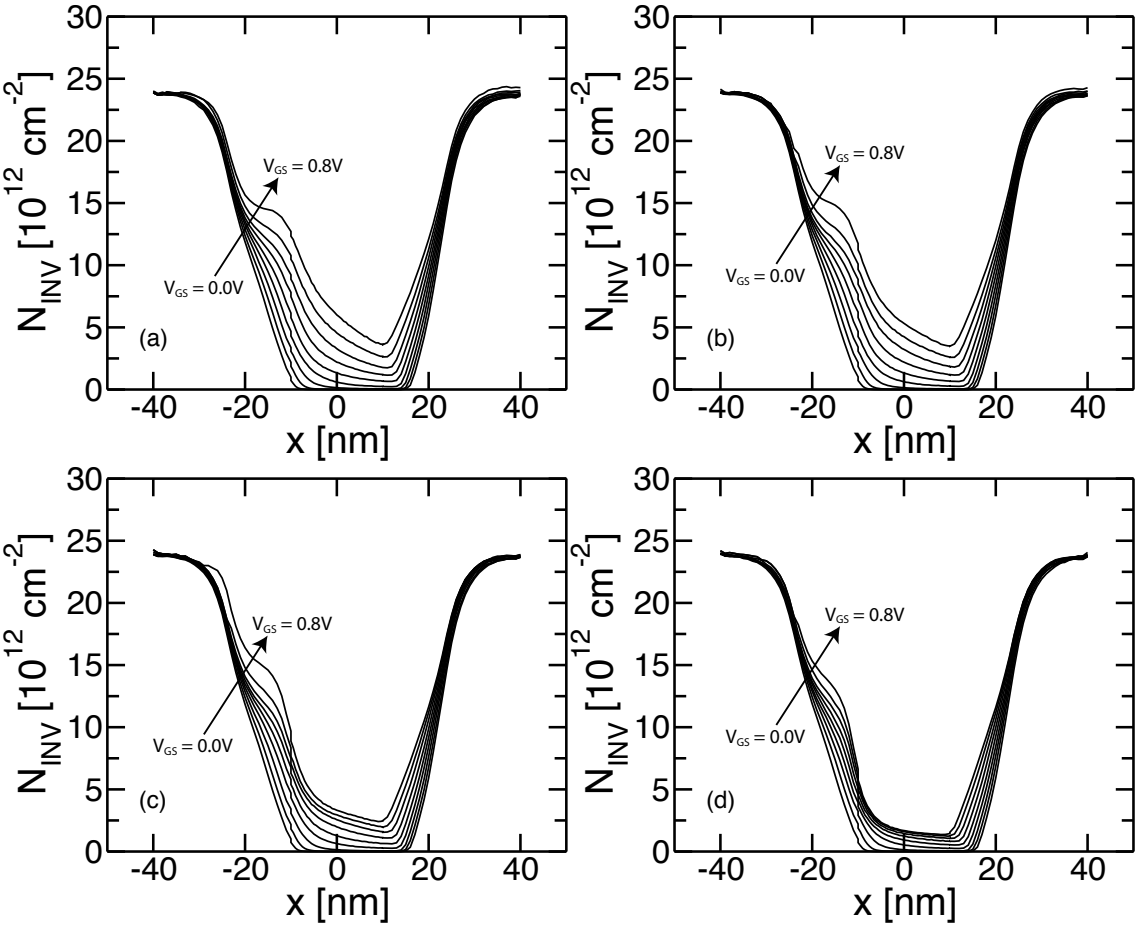


Figure 5.29: Free carrier densities for the device #3 shown in Fig. 3.2. (a) Simulation without traps. (b) Simulation with traps that pin the N_{INV} at $1 \cdot 10^{13}$ cm^{-2}. (c) Simulation with traps that pin the N_{INV} at $6 \cdot 10^{12}$ cm^{-2}. (d) Simulation with traps that pin the N_{INV} at $3 \cdot 10^{12}$ cm^{-2}.

Bibliography

[1] S.H. Kim, M. Yokoyama, N. Taoka, R. Nakane, T. Yasuda, O. Ichikawa, N. Fukuhara, M. Hata, M. Takenaka, and S. Takagi. "Enhancement technologies and physical understanding of electron mobility in III-V n-MOSFETs with strain and MOS interface buffer engineering". In: *IEEE IEDM Technical Digest*. 2011, pp. 13.4.1–13.4.4.

[2] S.W. Chang, X. Li, R. Oxland, S.W. Wang, C.H. Wang, R. Contreras-Guerrero, K.K. Bhuwalka, G. Doornbos, T. Vasen, M.C. Holland, G. Vellianitis, M.J.H. van Dal, B. Duriez, M. Edirisooriya, J.S. Rojas-Ramirez, P. Ramvall, S. Thoms, U. Peralagu, C.H. Hsieh, Y.S. Chang, K.M. Yin, E. Lind, L.-E. Wernersson, R. Droopad, I. Thayne, M. Passlack, and C.H. Diaz. "InAs n-MOSFETs with record performance of I_{ON} = 600 μA/μm at I_{OFF} = 100 nA/μm (Vd = 0.5 V)". In: *IEEE IEDM Technical Digest*. 2013, pp. 16.1.1–16.1.4.

[3] P. Weigele, L. Czornomaz, D. Caimi, N. Daix, M. Sousa, J. Fompeyrine, and C. Rossel. "III-V heterostructure-on-insulator for strain studies in n-InGaAs channels". In: *Proc. ULIS*. 2013, pp. 45–48.

[4] A.M. Sonnet, R.V. Galatage, P.K. Hurley, E. Pelucchi, K. Thomas, A. Gocalinska, J. Huang, N. Goel, G. Bersuker, W.P. Kirk, C.L. Hinkle, and E.M. Vogel. "Remote phonon and surface roughness limited universal electron mobility of $In_{0.53}Ga_{0.47}As$ surface channel MOSFETs". In: *Microelectronic Engineering* 88.7 (2011), pp. 1083–1086.

[5] N. Taoka, M. Yokoyama, S.H. Kim, R. Suzuki, R. Iida, S. Lee, T. Hoshii, W. Jevasuwan, T. Maeda, T. Yasuda, O. Ichikawa, N. Fukuhara, M. Hata, M. Takenaka, and S. Takagi. "Impact of Fermi level pinning inside conduction band on electron mobility of $In_xGa_{1-x}As$ MOSFETs and mobility enhancement by pinning modulation". In: *IEEE IEDM Technical Digest*. 2011, pp. 27.2.1–27.2.4.

[6] SangHyeon Kim, Masafumi Yokoyama, Noriyuki Taoka, Ryosho Nakane, Tetsuji Yasuda, Osamu Ichikawa, Noboru Fukuhara, Masahiko Hata, Mitsuru Takenaka, and Shinichi Takagi. "Strained $In_{0.53}Ga_{0.47}As$ metal-oxide-semiconductor field-effect transistors with epitaxial based biaxial strain". In: *Applied Physics Letters* 100.19 (2012).

[7] D. K. Schroder. *Semiconductor material and device characterization*. Hoboken, NJ, USA: John Wiley & Sons, Inc., Oct. 2005. ISBN: 9780471749097.

[8] D. Lizzit, D. Esseni, P. Palestri, and L. Selmi. "Surface roughness limited mobility modeling in ultra-thin SOI and quantum well III-V MOSFETs". In: *IEEE IEDM Technical Digest*. 2013, pp. 5.2.1–5.2.4.

[9] D. Veksler, P. Nagaiah, T. Chidambaram, R. Cammarere, V. Tokranov, M. Yakimov, Y.-T. Chen, J. Huang, N. Goel, J. Oh, G. Bersuker, C. Hobbs, P. D. Kirsch, and S. Oktyabrsky. "Quantification of interfacial state density (D_{it}) at the high-k/III-V interface based on Hall effect measurements". In: *Journal of Applied Physics* 112.5 (2012).

[10] G. Brammertz, H.-C. Lin, M. Caymax, M. Meuris, M. Heyns, and M. Passlack. "On the interface state density at $In_{0.53}Ga_{0.47}As$/oxide interfaces". In: *Applied Physics Letters* 95.20 (2009).

[11] M. Heyns, A. Alian, G. Brammertz, M. Caymax, Y. C. Chang, L. K. Chu, B. De Jaeger, G. Eneman, F. Gencarelli, G. Groeseneken, G. Hellings, A. Hikavyy, T. -Y Hoffmann, M. Houssa, C. Huyghebaert, D. Leonelli, D. Lin, R. Loo, W. Magnus, C. Merckling, M. Meuris, J. Mitard, L. Nyns, T. Orzali, R. Rooyackers, S. Sioncke, B. Soree, X. Sun, A. Vandooren, A.S. Verhulst, B. Vincent, N. Waldron, G. Wang, W. -E Wang, and L. Witters. "Advancing CMOS beyond the Si roadmap with Ge and III-V devices". In: *IEEE IEDM Technical Digest.* 2011, pp. 13.1.1–13.1.4.

[12] C.L. Hinkle, A.M. Sonnet, R.A. Chapman, and Eric M. Vogel. "Extraction of the Effective Mobility of $In_xGa_{1-x}As$ MOSFETs". In: *IEEE Electron Device Lett.* 30.4 (2009), pp. 316–318.

[13] Y. Xuan, Y.Q. Wu, T. Shen, T. Yang, and P.D. Ye. "High performance submicron inversion-type enhancement-mode InGaAs MOSFETs with ALD Al_2O_3, HfO_2 and HfAlO as gate dielectrics". In: *IEEE IEDM Technical Digest.* 2007, pp. 637–640.

[14] D. Lizzit, D. Esseni, P. Palestri, P. Osgnach, and L. Selmi. "Performance Benchmarking and Effective Channel Length for Nanoscale InAs, $In_{0.53}Ga_{0.47}As$, and sSi n-MOSFETs". In: *IEEE Trans. on Electron Devices* 61.6 (June 2014), pp. 2027–2034.

[15] D. Lizzit, P. Palestri, D. Esseni, A. Revelant, and L. Selmi. "Analysis of the performance of n-Type FinFETs with strained SiGe Channel". In: *IEEE Trans. on Electron Devices* 60.6 (June 2013), pp. 1884–1891.

[16] D. Esseni, P. Palestri, and L. Selmi. *Nanoscale MOS Transistors.* Cambridge University Press, 2011.

[17] Sylvain Barraud, Olivier Bonno, and Mikaël Cassé. "The influence of Coulomb centers located in HfO_2/SiO_2 gate stacks on the effective electron mobility". In: *Journal of Applied Physics* 104.7 (2008), p. 073725.

[18] P. Toniutti, P. Palestri, D. Esseni, F. Driussi, M. De Michielis, and L. Selmi. "On the origin of the mobility reduction in n- and p-metal–oxide–semiconductor field effect transistors with hafnium-based/metal gate stacks". In: *Journal of Applied Physics* 112.3 (2012).

[19] L. Lucci, P. Palestri, D. Esseni, L. Bergagnini, and L. Selmi. "Multisubband Monte Carlo Study of Transport, Quantization, and Electron-Gas Degeneration in Ultrathin SOI n-MOSFETs". In: *IEEE Trans. on Electron Devices* 54.5 (2007), pp. 1156–1164.

[20] P. Osgnach, E. Caruso, D. Lizzit, P. Palestri, D. Esseni, and L. Selmi. "The impact of interface states on the mobility and the drive current of III-V MOSFETs". In: *Proc. ULIS.* Apr. 2014, pp. 21–24.

[21] Eun Ji Kim, Lingquan Wang, Peter M. Asbeck, Krishna C. Saraswat, and Paul C. McIntyre. "Border traps in $Al_2O_3/In_{0.53}Ga_{0.47}As$ (100) gate stacks and their passivation by hydrogen anneals". In: *Applied Physics Letters* 96.1 (2010), p. 012906.

[22] Hideki Hasegawa and Takayuki Sawada. "Electrical modeling of compound semiconductor interface for FET device assessment". In: *IEEE Trans. on Electron Devices* 27.6 (June 1980), pp. 1055–1061.

[23] G. Brammertz, A. Alian, D.H.-C. Lin, M. Meuris, M. Caymax, and W.-E. Wang. "A Combined Interface and Border Trap Model for High-Mobility Substrate Metal-Oxide-Semiconductor Devices Applied to $In_{0.53}Ga_{0.47}As$ and InP Capacitors". In: *Electron Devices, IEEE Transactions on* 58.11 (Nov. 2011), pp. 3890–3897.

[24] P. Palestri, D. Esseni, S.. Eminente, C. Fiegna, E. Sangiorgi, and L. Selmi. "Understanding quasi-ballistic transport in nano-MOSFETs: part I-scattering in the channel and in the drain". In: *IEEE Trans. on Electron Devices* 52.12 (Dec. 2005), pp. 2727–2735.

Chapter 6

Conclusions

The analysis carried out during my PhD have shown that the Multi-subband Monte Carlo is a valid TCAD tool for the simulation of III-V devices. Its major drawback is the computational burden since not optimised implementations can require 90 minutes on average to complete a single iteration. This means that one must wait around half a day (or even a whole day) to complete the simulation of one bias point. Biases above the threshold voltage are faster to simulate [1].

An optimised and parallel implementation can deliver the same results much faster. If enough hardware resources are available, on can expect to be able to complete one iteration in one or two minutes and a full simulation can be completed in less than one hour. The Monte Carlo and scattering rates computation steps (see the flowchart of Fig. 2.2) are the ones that consume most of the simulation time so any further performance improvement work must still be focused on these two steps. Chapter 3 has shown that the performance scaling with respect to the number of threads are far from ideal. The MC step scaling is limited by thread synchronisation enforced during the statistics collection phase. Minimizing the impact of this synchronisation is challenging because the MC step is not deterministic. The time required to process one particle depends on how many scattering events it experiences, which in turn depends on the state of the particle. Another source of imbalance come from particle absorption/injection at the contacts which cause a variation of the number of particles processed by each thread. While the impact of the latter issue can be reduced as described in section 3.3.2, there is no implemented solution for the former. The scattering rates computation step also shows scaling issues. These issues depends mostly on the shallow level of parallelism, which operates only at the sections-level. Towards the end of this step, many threads will become idle. A possible future work involves the implementation of a deeper level of parallelism.

The Multi-subband Monte Carlo method still operates in the semi-classical modelling framework, and considers quantisation effects only in the direction normal to the dielectric/semiconductor interface. Chapter 4 has shown that for gate lengths around 10 nm and below, quantisation effects along the transport direction start to play a role. One of these effect is the source to drain tunnelling, which increases the OFF current and degrades the sub-threshold slope. The MSMC simulator implements a simple approach to model this effect by means of subband profile smoothing via convolution with a Gaussian function [2]. A proper calibration of the parameter of the Gaussian allows the MSMC to attain a good agreement with the results obtained from simulators based on NEGF formalism. Values of

the σ parameter around 6.5 nm can be effectively used to replicate the results of NEGF simulators for various device geometries, suggesting that this parameter depends mainly on the semiconductor material rather than on the device geometry.

Regarding the interface states, the most delicate part of the models is certainly the D_{it} trap distribution profile. This profile can be easily calibrated by matching the inversion carrier density versus bias voltage obtained from Hall measurements. The profile can be then used for mobility simulations. The proper replication of the effects of Fermi level pinning is required when comparing measured and simulated mobilities, otherwise one may overestimate the impact of the scattering mechanisms on mobility since an artificial degradation appears. Chapter 5 showed that the implemented model allows to obtain mobility results fairly close to the experiments. A possible future work involves further studies about the applicability of this model to the simulation of the current of short devices. The simulations showed that the interface states have a low impact on the current of short channel devices because the gate voltage is not high enough to allow the population of the higher energy states. So far, interface states are populated assuming equilibrium statistics associated with an effective Fermi level, but further work is required in order to prove the validity of this assumption.

Bibliography

[1] P. Osgnach, A. Revelant, D. Lizzit, P. Palestri, D. Esseni, and L. Selmi. "Toward computationally efficient Multi-Subband Monte Carlo simulations of nanoscale MOSFETs". In: *Proc.SISPAD*. 2013, pp. 176–179.

[2] P Palestri, L Lucci, S Dei Tos, D Esseni, and L Selmi. "An improved empirical approach to introduce quantization effects in the transport direction in multi-subband Monte Carlo simulations". In: *Semiconductor Science Technology* 25.5 (2010), p. 055011.

Publications of the Author

Journal Papers

A. Revelant, P. Osgnach, P. Palestri and L. Selmi, "An Improved Semi-Classical Model to Investigate Tunnel-FET performance", ECS Transactions, Vol. 54, issue 1, pp. 77-82, 2013

A. Revelant, P. Palestri, P. Osgnach, L. Selmi, "Calibrated multi-subband Monte Carlo modeling of tunnel-FETs in silicon and III–V channel materials", Solid-State Electronics, vol. 88, 2013, pp. 54-60

D. Lizzit, D.Esseni, P.Palestri, P. Osgnach and L.Selmi, "Performance benchmarking and Effective Channel Length for nanoscale InAs, $In_{0.53}Ga_{0.47}As$ and sSi n-MOSFETs", Electron Devices, IEEE Transactions on, vol.61, no.6, pp.2027-2034, June 2014

P. Osgnach, E. Caruso, D. Lizzit, P. Palestri, D. Esseni, and L. Selmi "The impact of interface states on the mobility and drive current of $In_{0.53}Ga_{0.47}As$ semiconductor n-MOSFETs", Solid-State Electronics, 2015

Conference Papers

A. Revelant, P. Osgnach, P. Palestri, L. Selmi, "An Improved Semicalssical Approach for Simulating Tunnel-FETs", Proocedings of GE2012, 44th Conference, Marina di Carrara, June 20-22, 2012 p. 33-34

D. Lizzit, P. Osgnach, D. Esseni, P. Palestri and L. Selmi, "Performance of III-V nanoscale MOSFETs: a simulation study", Proocedings of GE2013, 45th Conference, Udine, June 19-21, 2013, p.33-34

P. Osgnach, A. Revelant, D. Lizzit, P. Palestri, D. Esseni, L. Selmi, "Toward computationally efficient Multi-Subband Monte Carlo Simulations of Nanoscale MOSFETs", Proceedings International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), Glasgow (UK), settembre 2013, pag. 176-179

A. Revelant, P. Palestri, P. Osgnach, D. Lizzit, L. Selmi "On the Optimisation of SiGe and III-V Compound Hetero-Junction Tunnel FET Devices", European Solid- State Device Research Conference (ESSDERC), Bucharest (ROMANIA), 16-20 September 2013, pag.

49-52

P. Osgnach, E. Caruso, D. Lizzit, P. Palestri, D. Esseni and L. Selmi, "The impact of interface states on the mobility and the drive current of III-V MOSFETs", Proceeding of the International Conference on Ultimate Integration on Silicon (ULIS), pp. 21-24, Apr 2014

E. Caruso, D. Lizzit, P. Osgnach, D. Esseni, P. Palestri, L. Selmi, "Simulation analysis of III-V n-MOSFETs: channel materials, Fermi level pinning and biaxial strain", Proceedings of IEDM 2014