Multilevel Parallelization Approach to Estimate Spin Lifetime in Silicon: Performance Analysis

Joydeep Ghosh[1], Dmitry Osintsev[2], Viktor Sverdlov[2], and Swaroop Ganguly[1]
[1]Department of Electrical Engineering, Indian Institute of Technology, Mumbai, India
[2]Institute for Microelectronics, TU Wien, Gußhausstraße 27–29/E360, 1040 Wien, Austria
e-mail: {joydeepghosh}@ee.iitb.ac.in

***Abstract*— Evaluation of the spin lifetime in thin Silicon films is a challenge because of the necessity of performing appropriate averaging of the strongly scattering momenta depending spin relaxation rates. Here we discuss a two-level highly parallelized algorithm to calculate the spin lifetime. This algorithm is based on a hybrid parallelization approach, using the message passing interface MPI as well as OpenMP. Most efficient way to maximally utilize the computational resources is described. Finally, how an application of shear strain can dramatically increase the spin lifetime is shown.**

Keywords*— Spin lifetime; MPI; OpenMP; Multi-Level Parallel Computing

I. INTRODUCTION

In modern days, supercomputers play a significant role in the field of computational sciences. It is becoming feasible to solve more and more complex problems, because high performance computational resources are accessible for practical calculations. MPI (Message Passing Interface) is a popular standard for writing message passing programs [1]. MPI provides the user with a programming model where processes communicate with each other by calling library routines to send and receive messages. OpenMP (Open Multi-Processing) is an application programming interface (API) which supports multi-platform shared memory programming [2,3]. Moreover, the MPI and OpenMP programming models can be combined into a hybrid paradigm to exploit parallelism. This hybrid approach is suitable for clusters of symmetric multi-processor (SMP) nodes where MPI is needed for parallelism across nodes and OpenMP is used to implement loop level parallelism within a node. A lot of efforts have been directed to utilize the computational power in the most effective way [3,4]. Practically, all systems that belong to TOP500 supercomputers are based on multi-core CPUs [5]. A considerable part of algorithm in any computational problem can be parallelized by dividing the domain into independent parts, known as domain decomposition [6]. Each part can be calculated in a single MPI process without much efforts devoted to communication between separate MPI processes. Unfortunately, such an approach gets limited if each MPI process requires a high amount of memory or intensive communication. Indeed, need of a large memory requirement would depend on the problem being addressed as well as the algorithm that should be chosen. Fortunately, in some cases, the memory requirements are significantly reduced if the calculations are performed by sharing memory. However, as the number of cores per node is limited, the reduction of memory requirements can lead to an unacceptable increase of the total calculation time. Nevertheless, for the class of problems for which shared memory can significantly reduce the total amount of memory requirements, a combination between MPI and OpenMP approach is promising [3,4,6].

Here, our problem is to find the spin lifetime (τ) in (001) ultra-thin Silicon films subjected to [110] uniaxial tensile stress (ε_{xy}). We take into account the main mechanisms determining the mobility in thin Silicon films, namely surface roughness (SR) and electron-phonon scattering (namely, longitudinal acoustic LA and transversal acoustic TA phonons), and analyze their role in spin relaxation. It belongs to a class of problems that use shared memory to reduce on-node memory requirements in the combination with weakly coupled domain decomposition.

Big efforts of researchers and workers involved in a high-performance computing are directed to properly load balancing of the MPI jobs [7]. Since the domain of the considered problem can be divided to equivalent sub-domains, the amount of computational time for each MPI job differs insignificantly. Another important point pertaining to the load balancing is that all calculations are performed on the symmetric multiprocessor system (SMP). Thus, the problem of dynamic load balancing can be ignored. In the next section, a brief introduction of the problem statement and our solution approach is discussed.

II. MODEL AND SIMULATION

The unprimed electron subband energies and the wave functions are obtained with the two-band **k·p** Hamiltonian describing the [001] valley dispersion including spin. After that, the spin relaxation matrix elements is calculated. The SR and phonon mediated spin flip rates are then computed. Based on that, one can calculate the spin relaxation time by thermal averaging [8-12]:

$$\frac{1}{\tau_m[\tau_s]} = \frac{\sum_i \int \frac{1}{\tau_i(\mathbf{k_1})} \cdot f(E)(1-f(E))d\mathbf{k_1}}{\sum_i \int f(E)d\mathbf{k_1}}, \quad (1)$$

$$\int d\mathbf{k_1} = \int_0^{2\pi} d\varphi \cdot \int_{E_i^{(0)}}^{\infty} \frac{|\mathbf{k_1}|}{\left|\frac{\partial E(\mathbf{k_1})}{\partial \mathbf{k_1}}\right|_{\mathbf{k_1}}} dE.f(E) = \frac{1}{\left[1+\exp\left(\frac{E-E_F}{K_B T}\right)\right]}. \quad (2)$$

E is the electron energy, $\boldsymbol{k_1}$ is the in-plane wave vector before scattering, K_B is the Boltzmann constant, T is the temperature, and E_F is the Fermi level. The expression of SR mediated spin relaxation rate is given below.

978-1-5386-4811-7/18/$31.00 ©2018 IEEE

$$\frac{1}{\tau_{i,s,SR}(\mathbf{k_1})} = \frac{4\pi}{\hbar(2\pi)^2}\sum_{j=1,2}\int_0^{2\pi} d\varphi \cdot \pi\triangle^2 L^2 \cdot \frac{1}{\epsilon_{ij}^2|\mathbf{k_2}-\mathbf{k_1}|}\cdot\frac{\hbar^4}{4m_l^2}\cdot\frac{|\mathbf{k_2}|}{\left|\frac{\partial E(\mathbf{k_2})}{\partial \mathbf{k_2}}\right|}$$
$$\cdot\left[\left(\frac{d\Psi_{i\mathbf{k}_{1\sigma}}}{dz}\right)^*\left(\frac{d\Psi_{j\mathbf{k}_{2-\sigma}}}{dz}\right)\right]^2_{z=\pm\frac{t}{2}}\cdot\exp\left(\frac{-|\mathbf{k_2}-\mathbf{k_1}|^2L^2}{4}\right)$$
$$\cdot\theta(E_j(\mathbf{k_2})-E_j^{(0)}). \quad (3)$$

Here, $\Psi_{i\mathbf{k}\sigma}$ denotes the i^{th} subband wave functions, $\boldsymbol{k_2}$ is the in-plane wave vector of the electron after scattering, φ is the angle between $\boldsymbol{k_1}$ and $\boldsymbol{k_2}$, $\epsilon_{i,j}$ is the dielectric permittivity, L is the autocorrelation length, and Δ is the mean square value of the SR-fluctuations. $\sigma=\pm1$ is the spin projection to the [001] axis, and θ(x) is the Heaviside function. During the computational process, a multi-dimensional integral over the energy E and φ is to be computed. The spin relaxation matrix elements are characterized by very narrow and sharp peaks which are resolved by using a very fine mesh. Additionally, for accurate calculations, the energy step value ΔE is estimated to be upper-bounded by 0.5meV. The lower limit of the integral over E is zero, and we also identify that it is sufficient to set the corresponding upper limit to be 0.7eV. Thus, this particular simulation setup requires around 1400 points. The step value for φ, or $\Delta\varphi$, needs to be set smaller than 0.5°, where the lower and upper limits of the inner integral over φ are 0° and 360° respectively. Thus, this integral on before- and after-scattering directions at fixed energies requires almost 1000 points each. All these cause the numerical spin relaxation time calculation prohibitively expensive. We have implemented a two-level parallelization algorithm (ref: Algorithm 1) [13]. At Level 1, all static wave functions and energy data are calculated and archived in a binary file as a file-based cache technique. This is known as the serialization process. At the second stage (Level 2), the spin lifetime is calculated by deserializing the cache and calculating the spin relaxation rates. Serialization and deserialization processes are performed by using the Boost Serialization library [14]. Next, we scrutinize the performance of the algorithm in every level and determine the most efficient way to utilize the resources. It is measured on the Vienna Scientific Cluster (VSC-2) which consists of 1314 nodes having high performance InfiniBand (IB) for network communications [15]. Each node of the cluster has 2 processors (AMD Opteron 6132 HE, 2.2 GHz and 8 cores) and 8×4GB main memory.

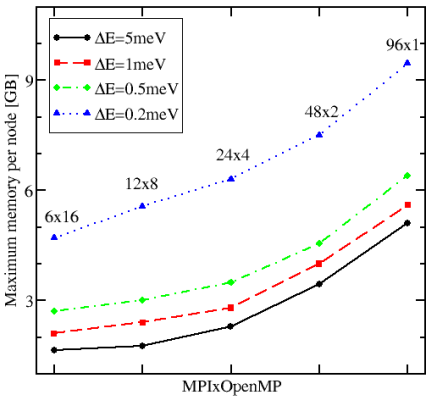

Fig. 1. Dependence of the maximum memory/node for different values of ΔE and different MPI×OpenMP configurations is shown.

Algorithm 1: Spin relaxation rate

Level 1:

1 Divide the whole range of angle φ into sub-domains for each MPI process.

1.1 Divide the whole range of energy E into sub-domains for each OpenMP thread.

1.1.1 Calculate wave functions' derivatives at the interface $(\frac{\partial\psi}{\partial z})_{z=\pm1/2}$ and $\frac{|\boldsymbol{k}|}{|\frac{\partial E(k)}{\partial z}|}$.

2 Collect all cached values at the first MPI process.

3 Archive serialized cache to the binary file.

Level 2:

4 Load serialized cache by the master MPI process.

5 Divide the range of φ into sub-domains for each MPI process.

5.1 Divide the range of E into sub-domains for each OpenMP thread.

5.1.1 Calculate (1) for a given range of values.

6 Collect all calculated relaxation rates into the final relaxation rate.

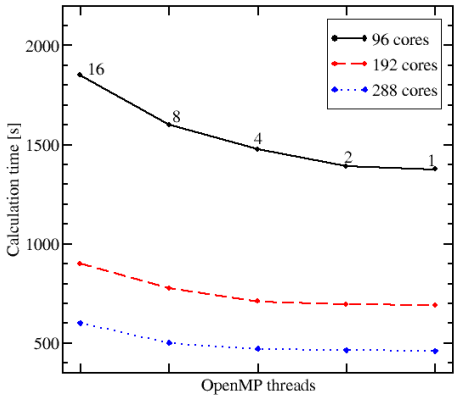

Fig. 2. Dependence of the total cache calculation time on different number of threads for fixed total core numbers is shown.

A. Level 1 Performance:

We begin with different configurations of MPI and OpenMP with certain number of cores, say 96. Three configurations are compared; 6 MPI jobs with 16 threads, 24 MPI jobs with 4 threads, and finally 96 MPI jobs with a single thread (known as a pure MPI configuration). Since the first MPI process is responsible for the collecting of all results and serializing the cache, it executes longer than other processes in all three cases. We have examined in all three cases that the fluctuations in the whole load figure remain insignificant, and the time spent for collecting and serialization is around 2 seconds. The IB throughput is much higher than the message size, and this is why the number of communication points does not influence the collection and serialization time.

Fig. 1 shows the dependence of the maximum memory per node required for different values of ΔE and different

configurations of MPI×OpenMP. Since the load is distributed equivalently for different MPI×OpenMP configurations, load balancing is not required. We observe that the memory requirement per node increases when number of threads is reduced, because each thread needs to make more calculations in a fixed direction, thereby increasing the wave vectors' cache. If a single thread is used (i.e. full MPI configuration) the memory requirement becomes 5GB by considering ΔE=5meV, which is around three times larger as compared to maximum threaded case (i.e. 6×16 MPI×OpenMP configuration). More realistic requirements are shown in Fig. 1 when ΔE=0.5meV. In the full MPI configuration scheme it requires 6.3GB memory, whereas in 6×16 MPI×OpenMP configuration it demands 2.6GB. It is further observed that even very accurate calculations (when ΔE=0.2meV) require less than 10GB of memory. Thus, memory limitations are not an issue considering any modern supercomputer with this particular simulation setup. Now we investigate the calculation speed.

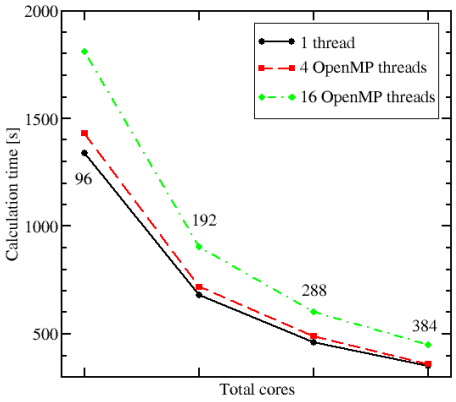

Fig. 3. Dependence of the calculation time for a fixed threads count on total number of cores is shown.

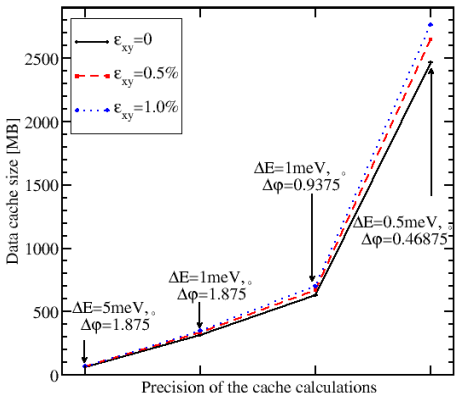

Fig. 4. Dependence of the size of cache on the precision of the calculations fixed by energy and angle steps for different ε_{xy} values is shown. Sample thickness t=1.5nm.

Fig. 2 demonstrates the dependence of the total cache calculation time on a different number of the parallel threads. For a 96 core application, the total calculation time reduces by 30% while the number of OpenMP threads reduces from 16 to 1. The performance decrease of the hybrid approach is due to data locality issues arising in shared-memory techniques. The profile of the calculation time with respect to the number of OpenMP threads for a larger number of cores remains similar. Fig. 3 shows the dependence of the calculation time on the total number of cores for a fixed threads count. As the MPI processes sub-domains are not correlated, the calculations in one domain cannot influence on the other. Therefore, an increment of the number of cores leads to the lossless reduction of the total calculation time.

B. Level 2 Performance:

Fig. 4 shows the size of the cache calculated in *Level 1* for different values of the energy and angle steps. This cache size depends significantly on the precision of calculation but weakly on the stress point. For example, when ΔE=1meV and $\Delta\varphi$=1.875°, the cache size is close to 500MB. However, accurate calculations demand at least ΔE=0.5meV and $\Delta\varphi$=0.5º. For such discretization parameters, the size of the serialized cache grows up to 3GB, causing restrictions on the number of parallel executed MPI jobs on a single node. Even smaller energy step (ΔE=0.2meV) makes the cache as big as 7GB, and theoretically only three processes can work together on a single node causing a significantly loss of the computational resources. Fig. 5 shows maximum memory per node for 8 and 16 threaded MPI application. We find that the memory requirements are mainly determined by the size of the serialized cache, and the memory footprint of the algorithm itself can be neglected. Thus, doubling the MPI jobs per node requires double memory space.

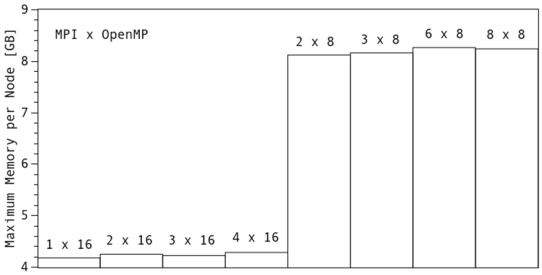

Fig. 5. Maximum required memory/node used by VSC-2 for different configurations is shown. Each MPI process loads 4GB cache.

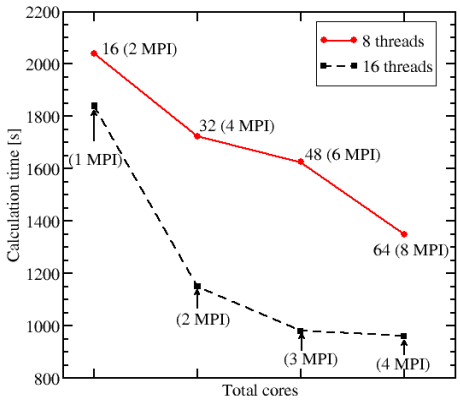

Fig. 6. Dependence of the total calculation time of the spin relaxation on total cores for 8 and 16 threads per MPI process.

Fig. 6 shows the dependence of the total calculation time on the number of cores as well as the number of threads. This demonstrates that increasing the total number of cores at a fixed number of threads decreases the demand on computing time, which is further reduced when the number of threads is increased. Moreover, in contrast to Fig. 2 and Fig. 3, Fig. 6 shows how increasing the thread numbers from 8 to 16 leads to decrease of the total calculation time for all values of the cores numbers. This approach is tested with 416 cores and requires only around 40 min for a single relaxation time data point (around 280 core-hours).

III. SPIN LIFETIME

Finally, we find an order of increment of the spin lifetime under application of shear strain ε_{xy} in Fig. 7. We show the spin-flip caused by the intra- and inter-unprimed subband scattering. The major contribution to spin relaxation comes from the intersubband processes due to the presence of the spin hot spots characterized by the peaks of the intersubband spin relaxation matrix elements. It is further noticed that at higher stress the intrasubband component also becomes significant. The subband splitting between the unprimed valley pair is lifted by ε_{xy}, and this lifting of the degeneracy is the crucial factor for spin lifetime enhancement [12,16-18]. Additionally, we observe that the spin lifetime is a factor of two longer for spins injected in-plane as compared to the time for spins injected perpendicular to the film. We point out that this factor remains preserved for both inter- and intra-valley scattering processes, and is independent of the considered scattering mechanisms [11,19].

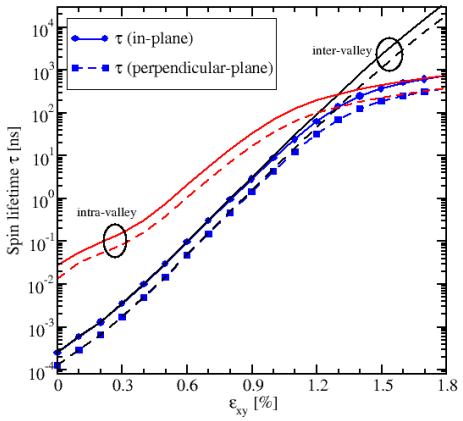

Fig. 7. The variation of the spin lifetime with its inter- and intrasubband components with ε_{xy} is shown. The film thickness is t=1.36nm, T=300K, and the electron concentration is N_S=10^{12}cm^{-2}.

IV. CONCLUSION

We have described a two-level parallelization scheme to calculate the Silicon spin lifetime. The computational trade-off with respect to accuracy with our simulation set up, memory consumption, and calculation time are analyzed in each level. We conclude that the cache calculation effectively can be performed in a pure MPI approach, however for the relaxation time calculations, combination of MPI and OpenMP approach solves the problem in the most efficient way. Finally, we have shown how shear strain can boost Silicon spin lifetime, and spin injection orientation has an additional impact.

ACKNOWLEDGMENT

The computational results presented have been partly achieved using the Vienna Scientific Cluster (VSC).

REFERENCES

[1] MPI 1.1 Standard, http://www-unix.mcs.anl.gov/mpi/mpich.

[2] L. Dagum, R. Menon, "OpenMP: an industry standard API for shared-memory programming", proc. in IEEE Computational Science and Engineering, pp. 46-55 (1998).

[3] G. Jost, H. Jin, D. an Mey, and F. F. Hatay, "Comparing the OpenMP, MPI, and Hybrid Programming Paradigms on an SMP Cluster", NAS Technical Report NAS-03-019, pp. 1-10 (2003).

[4] R Rabenseifner, G. Hager, and G. Jost, "Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-Core SMP Nodes", proc. in Euromicro International Conference on Parallel, Distributed and Network-based Processing, pp. 427-436 (2009).

[5] Top500 supercomputers, http://www.top500.org/lists/2017/11/.

[6] S. Tang, B.-S. Lee, and B. He, "Speedup for Multi-Level Parallel Computing", proc. in International Parallel and Distributed Processing Symposium Workshops & PhD Forum, pp. 537-546 (2012).

[7] O. Pearce, T. Gamblin, B. R. de Supinski, M. Shulz, and N. M. Amato, "Quantifying the Effectiveness of Load Balance Algorithms", proc. in 26th ACM international conference on Supercomputing, pp. 185-194 (2012).

[8] D. Osintsev, V. Sverdlov, and S. Selberherr, "Reduction of Momentum and Spin Relaxation Rate in Strained Thin Silicon Films", proc. in European Solid-State Device Research Conference, pp. 334-337 (2013).

[9] J. Ghosh, D. Osintsev, V. Sverdlov, and S. Selberherr, "Variation of Spin Lifetime with Spin Injection Orientation in Strained Thin Silicon Films", ECS Transactions, vol.66:5, pp. 233-240 (2015).

[10] J. Ghosh, D. Osintsev, V. Sverdlov, and S. Selberherr, "Enhancement of Electron Spin Relaxation Time in Thin SOI Films by Spin Injection Orientation and Uniaxial Stress", Journal of Nano Research, vol.39, pp. 34-42 (2016).

[11] J. Ghosh, D. Osintsev, V. Sverdlov, and S. Selberherr, "Dependence of Spin Lifetime on Spin Injection Orientation in Strained Silicon Films", proc. in Joint International EUROSOI Workshop and International Conference on Ultimate Integration on Silicon, pp. 285-288 (2015).

[12] V. Sverdlov and S. Selberherr, "Silicon Spintronics: Progress and Challenges", Physics Reports, vol.585, pp. 1-40 (2015).

[13] J. Ghosh, D. Osintsev, V. Sverdlov, J. Weinbub, and S. Selberherr, "Evaluation of Spin Lifetime in Thin-Body FETs: A High Performance Computing Approach", Lecture Notes in Computer Science, vol.9374, pp. 285-292 (2015).

[14] "Boost Serialization Library", www.boost.org/libs/serialization/.

[15] Vienna Scientific Cluster, http://www.vsc.ac.at/systems/vsc-2/.

[16] J. Ghosh, D. Osintsev, V. Sverdlov, and S. Selberherr, "Intersubband Spin Relaxation Reduction and Spin Lifetime Enhancement by Strain in SOI Structures", Microelectronic Engineering, vol.147, 89 (2015).

[17] J. Ghosh, V. Sverdlov, and S. Selberherr, "Influence of Valley Splitting on Spin Relaxation Time in a Strained Thin Silicon Film", proc. in International Workshop on Computational Electronics, pp. 1-4 (2015).

[18] V. Sverdlov, J. Ghosh, D. Osintsev, and S. Selberherr, "Modeling silicon spintronics", proc. in Recent Advances in Mathematical Methods in Applied Sciences, pp. 195-198 (2014).

[19] J. Ghosh, D. Osintsev, V. Sverdlov, and S. Selberherr, "Injection Direction Sensitive Spin Lifetime Model in a Strained Thin Silicon Film", proc. in Simulation of Semiconductor Processes and Devices, pp. 277-280 (2015).