

Acceleration of 3D numerical simulation of silicon solar cell using thread parallelism

B. Min, S. Suckow, U. Yusufoglu, T. M. Pletzer and H. Kurz
 Institute of Semiconductor Electronics, RWTH-Aachen University
 Sommerfeldstraße 24, D-52074 Aachen, Germany
 E-mail: min@iht.rwth-aachen.de

Abstract— We have investigated the potential to accelerate three-dimensional numeric simulation of silicon solar cell using thread parallelism. The device simulated is a rear side passivated cell with rear point contacts (PERC). The optical and electrical behaviour of the device was simulated with Sentaurus Device (formerly desso). We show that the simulation run-time on a four socket Opteron 6168 machine is reduced down to 6% compared to the run-time without thread parallelism. Furthermore, limits of time reduction by varying the number of threads up to 48 are studied. Thereby, the number of threads for the optimum use of the hardware resources is determined.

Keywords: simulation, PERC, silicon solar cell, high-performance computing

I. INTRODUCTION

Advanced cell concepts such as PERC, MWT (metal wrap through), EWT (emitter wrap through) or combinations of these concepts enable cell manufacturers to reduce the production cost per output power by increasing cell efficiency and reducing the cell thickness.

For this kind of complex cell concepts which include point like elements in the device structure, a true 3D device simulation is necessary, since the 2D simulation leads to errors in resistive and recombination losses [1]. Although simulations have become a very useful tool in PV research, 3D simulations are not widely carried out in comparison to 2D simulations because of time constrictions.

This work focuses on the effect of thread parallelism on the run-time of 3D device simulations of silicon solar cells. In this paper, a solar cell with rear point contact is chosen as an application example of thread parallelism. By varying the number of threads up to 48, we observed a significant reduction of simulation time.

II. APPROACH

A. Simulation model

As the simulation domain, a quarter of a symmetric element of a silicon solar cell with PERC is used. The device symmetry is illustrated in Fig. 1, where geometrical parameters are listed in Table 1.

Uniformly boron doped p-type silicon with doping of $1 \times 10^{16} \text{ cm}^{-3}$ ($\rho = 1.47 \text{ } \Omega \text{cm}$) is chosen as the substrate. The emitter of the simulated solar cells is selective, we used the

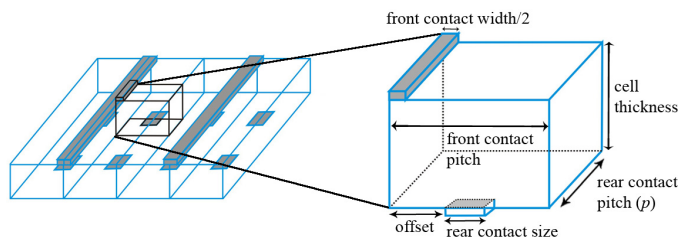


Fig. 1. The geometrical structure of the simulation domain.

TABLE I. THE GEOMETRICAL PARAMETERS OF THE SIMULATION DOMAIN.

Parameter	Value [μm]
cell thickness	180
front contact pitch	2
front contact width	80
front contact thickness	20
rear contact pitch (p)	350
rear contact size	85
offset	160

measured Electrochemical Capacitance-Voltage profiles of 45 /sq and 110 /sq phosphorus emitters from the work of Rudolph et al. [2]. The aluminum diffusion of the local back surface field is described by a gaussian profile with peak doping density of $2 \times 10^{19} \text{ cm}^{-3}$ and a junction depth of $5 \mu\text{m}$. The minority carrier lifetime in the SRH recombination model is equal to $100 \mu\text{s}$. The physical models adopted in these simulations include Schenk band gap narrowing [3] as well the Philips unified mobility model [4]. Fermi-Dirac statistics is adopted for the precise simulation at high dopant densities in emitter.

B. Thread parallelization

Sentaurus Device creates one process to run a simulation. The process itself can create multiple threads. Each thread occupies one processor core and is executed in parallel. The data communication between threads is established via shared memory. As solver for the computation we used the direct solver 'PARDISO', which is a high performance software package for solving large sparse symmetric or nonsymmetric systems of linear equations in parallel [5]. The software runs on a quad socket machine with four Opteron 6168 CPUs at 1.9 GHz. The main memory of 256 GB is distributed equally so that all 16 memory controllers are used.

III. RESULT

In the first part of this work the effect of the thread number variation on simulation run-time is investigated. In this case, there is only one process creating multiple threads. The number of threads varies between 1 and 48. As shown in Fig. 2 the simulation run-time drops almost linearly by increasing the number of threads up to 16, but levels off at higher numbers of threads. In comparison to the simulation run-time without thread parallelism (637 min.), 89% reduction of computation time (68 min.) was achieved at this point.

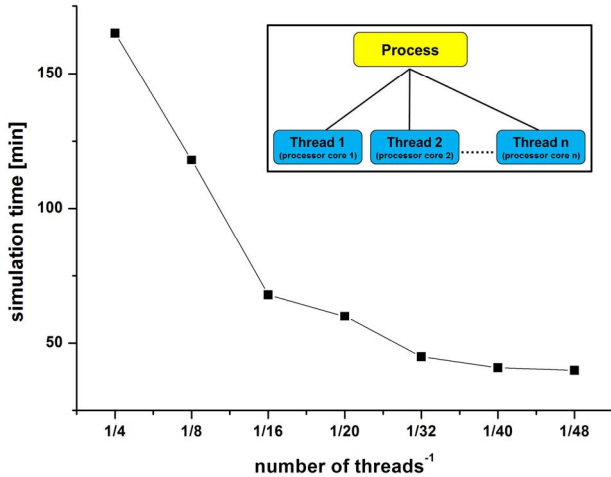


FIG. 2. Effect of thread number variation on simulation run-time; for one process.

With the assignment of all available process cores using 48 threads, 94% reduction of computation time (39 min.) was achieved in comparison to the run-time without thread parallelism. But relative to the case of using 16 threads, a time reduction of only 57% more is achieved by use of 200% additional process cores. Therefore another way to achieve optimal throughput from the machine is followed in the second part of this work. Since the acceleration due to thread parallelism saturates, the simultaneous execution of multiple processes with multiple threads has been tried for maximum throughput.

The optimum number of processes n_{pr}^{opt} and its corresponding number of threads n_{th}^{opt} can be determined by considering the maximum number of processor cores n_{core}^{max} and the number of parameter variations which can be run in parallel n_{var} with $n_{th}^{opt} = n_{core}^{max} / n_{var}$ and $n_{pr}^{opt} = n_{var}$. As soon as this occurs, the performance will decrease dramatically.

In Fig. 3 the simulation time is plotted versus the number of simultaneously running processes. From the negligible increment of simulation time with increasing number of running processes follows that additional reduction of simulation time can be achieved by executing multiple processes. In case of $n_{var} = 3$, the run-time of 71 minutes by parallel execution of processes with 16 threads each is much less than 117 minutes by sequential execution with 48 threads. Each process can be assigned for the analysis of a specific parameter variation such as cell thickness or base resistivity.

One may argue that parallel execution produces different rounding errors that influence the number of iterations of numerical computations. The investigations performed in this study, however, show no effect on the value of any simulation results of solar cell parameters.

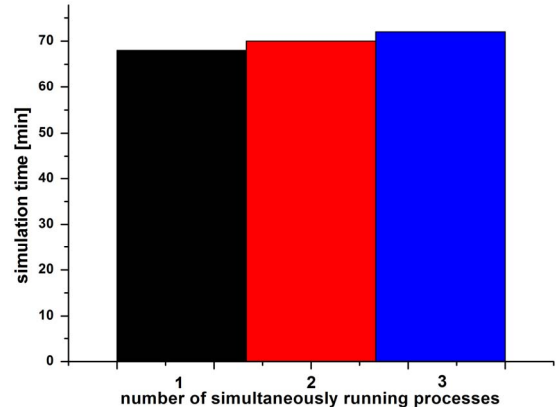


Fig. 3. Effect of simultaneous execution of multiple processes. Each process has 16 threads.

IV. CONCLUSION

The effect of parallelization through multi-threading technique on computation speed in the device simulator Sentaurus Device by 3D simulation is significant. We demonstrated for the optimization of a PERC cell that the simulation time can be reduced to 6% using all available cores. The gain in computation speed provides a significant reduction of the “time to result”. However, the scaling turned out to be less than ideal for more than 16 concurrent threads per simulation. The data accumulation in Fig. 2 provides important information on limit of maximum throughput, if several simulations need to be run and the amount of main memory is sufficient. This way three simulations with 16 threads each can be completed in 71 minutes whereas three subsequent simulations with 48 threads would require 117 minutes.

ACKNOWLEDGMENT

This work is part of the project “Kompetenzzentrum für innovative Photovoltaik-Modultechnik NRW” and has been supported by the European Union – European Regional Development Fund and by the Ministry of Economic Affairs and Energy of the State of North Rhine-Westphalia, Germany.

REFERENCES

- [1] P. Altermatt, “Models for numerical device simulations of crystalline silicon solar cells – a review”, *Journal of Computational Electronics*, 314-330, 2011.
- [2] D. Rudolph et. al., “Etch back selective emitter process with single POCl_3 diffusion“, 26th European Photovoltaic Solar Energy Conference and Exhibition, Hamburg, Germany, 2011.
- [3] A. Schenk, *J. Appl. Phys.* 84, 3684, 1998.
- [4] D. B. M. Klaassen, “A Unified Mobility Model for Device Simulation-I. Model Equations and Concentration Dependence”, *Solid.-State Electronics*, vol. 35, no. 7, pp. 953-959, 1992.
- [5] O. Schenk, “Scalable Parallel Sparse LU Factorization Methods on Shared Memory Multiprocessors”, *Series in Microelectronics*, vol. 89, Konstanz, Germany: Hartung-Gorre, 2000.