Numerical Simulation of a GaAs MESFET Device using Parallel Processing Techniques with Adaptive Meshing and Dynamic Load Balancing on a Transputer Network

C.S. Tsang-Ping, D.M. Barry, C.M. Snowden

Department of Electronic & Electrical Engineering, Leeds University, Leeds, LS2 9JT, UK

Summary

A parallel implementation of the numerical simulation of a GaAs MESFET device using the finite difference discretisation scheme and solved by a point iterative method is presented. Parallel techniques targeted at Multiple-Instructions Multiple-Data (MIMD) message-passing distributed memory architectures, in particular, *Transputers*, are described. Efficient parallelism is achieved by the geometric decomposition of the problem domain. Issues on the convergence and efficiency of the solution with novel strategies such as iteration ordering techniques and communication protocols in a parallel environment are discussed. The parallel implementation of the adaptive grid refinement also requires load balancing techniques for optimum efficiency. Performance results of the parallel simulation using these techniques are presented.

1. Introduction

Physical models are now widely used for simulating complex semiconductor devices. The increasing complexity of the device models requires high performance computers particularly for *interactive* device characterisation. With the advent of relatively cheap parallelism in the form of *Transputers* [1], a building block for a multiprocessor MIMD Distributed Memory Parallel System, the computational requirements for fast device characterisation can theoretically be met. Parallel processing offers attractive advantages such as scalable performance and superior cost/performance ratio. While parallel hardware is readily available, there still is a need for robust and efficient parallel software for semiconductor device simulation.

2. The Device Simulation Problem

This work covers the numerical simulation of a typical n-channel Metal Semiconductor Field-Effect Transistor (MESFET) using a physical modelling approach. A simplified *Drift / Diffusion Transport* model with the *Scharfetter-Gummel* formulation for current density has been used. The numerical solution was achieved using the finite difference discretisation scheme. The basic semiconductor equations consisting of the closely coupled poisson and continuity equations are solved by the Gauss-Seidel point iteration method with successive relaxation for time-dependent solution [2]. An adaptive refinement strategy is implemented for numerical accuracy and optimising computer resources.

3. The Parallel Solution - System and Algorithms

The parallel system used consists of an array of 16 TRAnsputer Modules (TRAMs) which forms a General Purpose MIMD [3] (GP-MIMD) message-passing distributed memory parallel system. A high level portable language, *Parallel ANSI C*, was used to code the algorithms. The device simulation problem has been parallelised using a one dimensional geometric decomposition [4]. Each processor is assigned a slice of the domain which is stored locally on each *TRAM* module's memory (e.g. Figure 1 shows 4 transputer modules and 4 subdomains). This is easily scalable to *n* transputers where n is less than the no of grid points horizontally.

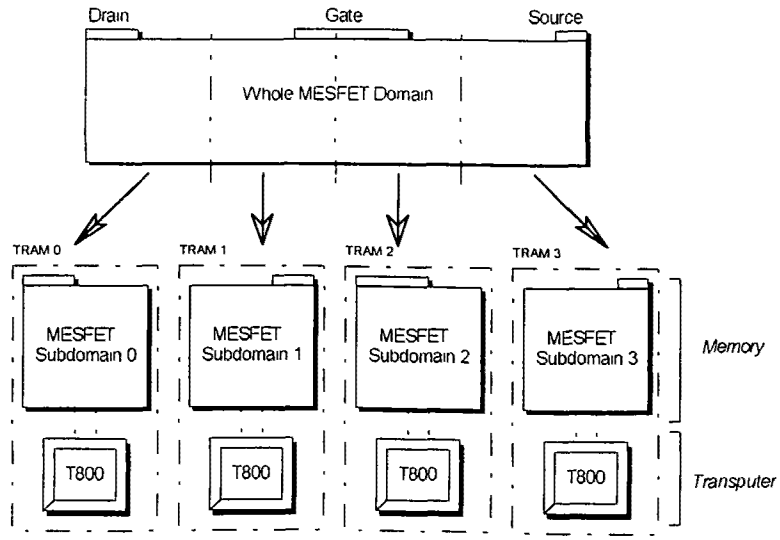

Figure 1. *Parallel Decomposition of the Device Simulation Problem*

The parallel system configuration of both hardware and software and associated communication channels is shown in figure 2. The transputers are connected in a ring network topology. A graphics process enables the real-time visualisation of the solution .

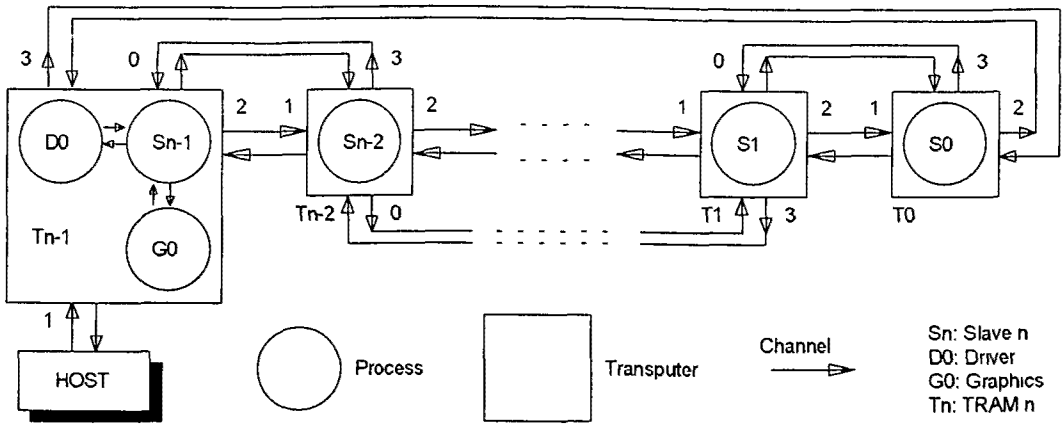

Figure 2. *Parallel Hardware and Software Configuration*

We now introduce the major overheads in the parallel simulator as opposed to a sequential implementation. The *slave* processes are controlled via channels 1 & 2 (control protocol) by a *driver* process. The latter instructs the slaves to perform specific task and monitors the state of

the simulation such as the global convergence of the iterative solvers and refinement decisions. The five-point formula used for solving the poisson and continuity equations requires four neighbouring data. However, at subdomain boundaries, a column of node data is missing and has to be transmitted via channels 0 & 3 (data exchange protocol).

The decomposition of the *MESFET* domain introduces convergence and stability problems of the numerical solution. For the poisson and continuity iterative solver, the update ordering is modified when domain is partitioned. A novel red/black checker-board updating technique named as the *RB SOR 1D* [5] partitioning method that ensures the stability and provides optimal convergence of the numerical solution was developed.

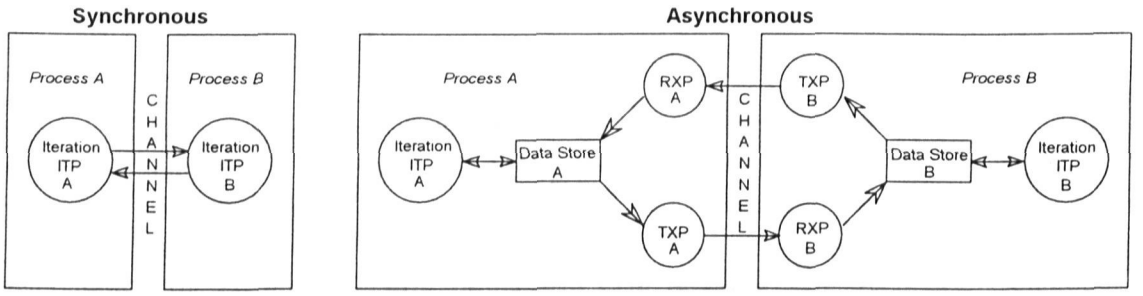

Figure 3. ***Data Exchange Communication Protocols***

Data exchange communication protocol which ensures data consistency across boundaries can broadly be categorised as synchronous and asynchronous [6]. Figure 3 depicts the two types of protocols implemented. Synchronous communication ensures that iteration processes are synchronised and boundary data are updated in a orderly fashion. Asynchronous communications, on the other hand, enable the communication processes (TXP & RXP) to be completely independent of the computation process (ITP). This protocol provides less synchronisation overheads but data consistency at each iteration cannot be predicted.

4. Parallel Adaptive Meshing and Load Balancing

A refinement algorithm based on the potential difference criteria was implemented on the parallel simulator. The dynamic refinement of the domain results in an imbalance of work loads on each processor. Load balancing techniques based on a quasi-dynamic [7] strategy were designed to achieve optimum performance. Figure 4 shows the allocation of the sub-domains when a network of 4 transputers has been load-balanced.

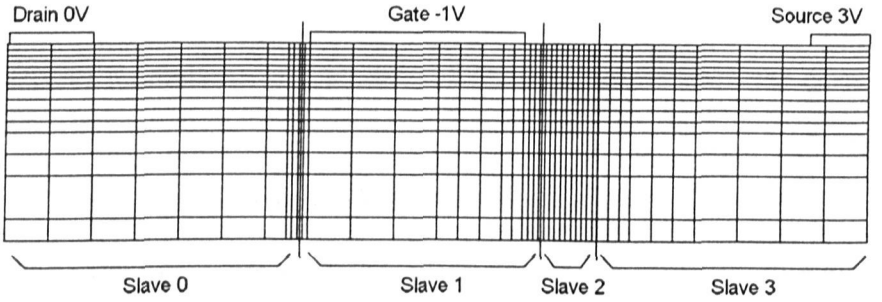

Figure 4. ***Adaptive meshing and Load Balancing***

5. Results and Performance

These results and performance data were obtained from the parallel system by simulating a *GaAs MESFET* with 0.15μm active channel (1.5e23 m^{-3}) and 0.55μm gate length using dynamic adaptive grid meshing. The solution for the electron concentration profile is shown in figure 5 for a bias condition shown in figure 4.

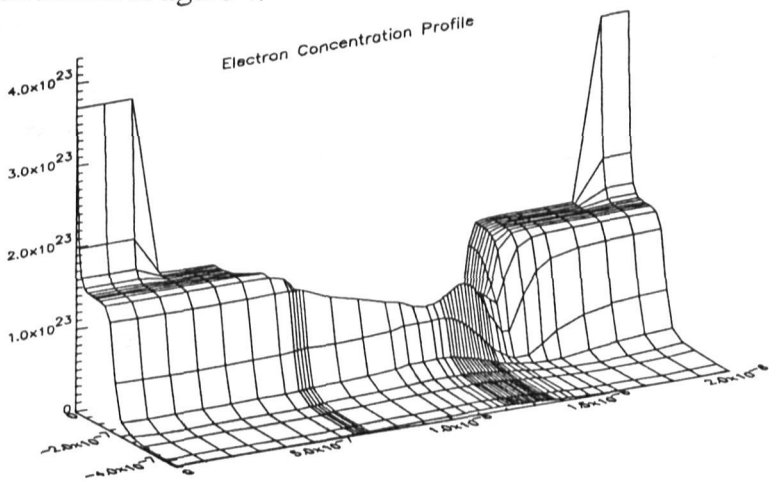

Figure 5. *Electron Concentration Profile for GaAs MESFET*

An overview of the overall performance of the parallel system is shown in figure 6. The drop in speed up is attributed to the communication overheads in the parallel system. The graph shows the parallel performance of the solution for a fixed grid of 128x32. A speed increase of up to a factor of 12.5 for synchronous protocol on a 16 transputer network has been achieved. For the asynchronous protocol depicted in figure 3, lower performance is observed due to excessive communication overheads.

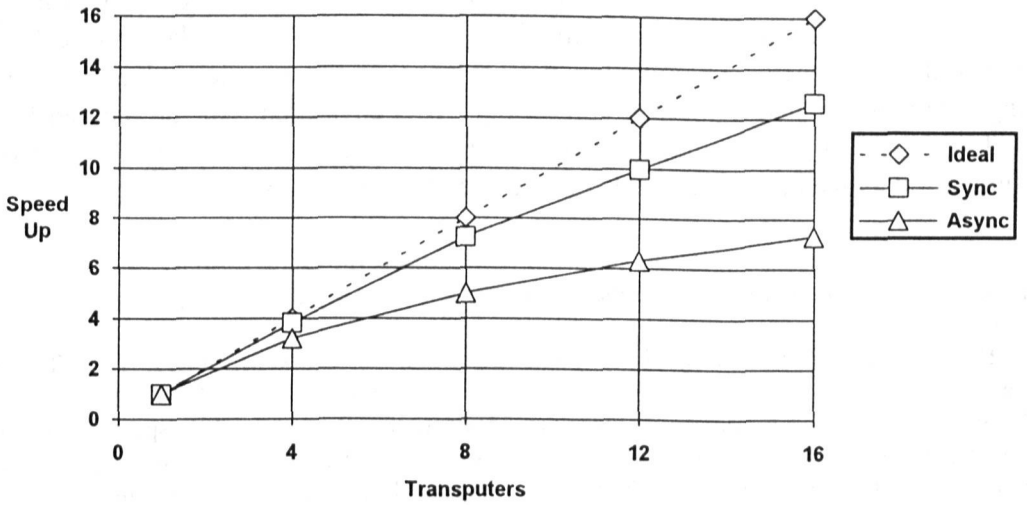

Figure 6. *Overall Parallel Performance Results for fixed grid*

Figure 7 shows the parallel performance for the parallel adaptive meshing algorithm of the *GaAs MESFET* device (from figure 4 & 5) starting with a grid of 41x10 to 62x14 over a time of 50fs. Due to the small grid size, the communication overheads are significant leading to

decreasing performance as more processors are used. The importance of dynamic load balancing is clearly shown from these results (using synchronous and RB SOR 1D techniques).

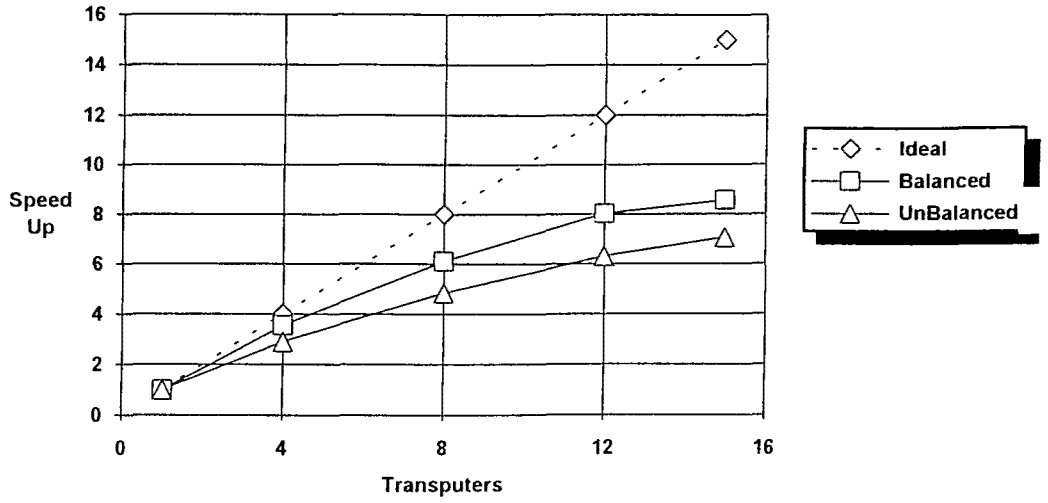

Figure 7. *Overall Parallel Performance Results for Adaptive Grid*

6. Conclusion

A parallel numerical solver for an iterative GaAs MESFET device solver has been presented. Geometric domain decomposition proves to be the natural way of parallelising the simulation enabling a logical map on transputer-based systems. Combined with Parallel C, this provides a good platform for developing parallel algorithms. However the overall code complexity and size increases in the parallel implementation. The convergence and stability behaviour of the solution is also found to be affected in a parallel environment. This led to the development of suitable iteration update ordering techniques which ensure fast convergence and stability of the solution. Communication overheads remain the major factor limiting the efficiency of the parallel simulator. The parallel grid refinement provided very efficient computation, with the optimum grid structure automatically generated but requires load balancing techniques to improve parallel efficiency. Overall, the performance of the parallel solver is scalable for increasing problem size and numerical complexity. It is envisaged that the parallel techniques developed will have a wide application on exceptionally computer intensive semiconductor device simulation as well as on fast/real-time device characterisation.

References

[1] A. Trew, G. Wilson, *Past, Present, Parallel: A survey of available parallel computing systems*, Springer-Verlag, 1991.

[2] T.M. Barton, *Characterisation of the physical behaviour of GaAs MESFETs*, PhD Thesis, Leeds University, 1988.

[3] M.R. Jane, R.J.Fawcett, T.P. Mawby, "Transputer Applications - Progress and Prospects", *Proceedings of the closing symposium of the SERC/DTI initiative in the engineering apps. of transputers*, IOS Press, 1992.

[4] C.G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, D.Walker, *Solving problems on concurrent processors: General techniques and regular problems*, Prentice-Hall, 1988.

[5] C.S. Tsang-Ping, D.M. Barry, C.M. Snowden, "Application of Parallel Processing Techniques to GaAs MESFET Device Simulation", *Proceedings of the 7th GaAs Simulation Group Meeting*, Harrogate, UK

[6] D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation*, Prentice Hall Inc., 1989

[7] R.D. Williams, "Performance of dynamic load balancing algorithms for unstructured mesh calculations", *Concurrency: Practice and Experience*, Vol.3(5), pp. 457-481, Oct 1991.