

Acceleration Techniques for Analysis of Microstrip Structures

R. Pomarnacki¹, A. Krukoniš¹, V. Urbanavicius¹

¹*Department of Electronic Systems, Vilnius Gediminas Technical University,
Naugarduko St. 41–427, LT-03227 Vilnius, Lithuania
raimondas.pomarnacki@vgtu.lt*

Abstract—This paper discusses the acceleration techniques for analysis of microstrip structures. Accurate calculation of parameters of such structures with numerical techniques requires the solution of dense matrix equations involving thousands of unknowns. Solution of this large problem takes long time. In this paper we present three techniques for such computations acceleration: parallel algorithm implemented in computer cluster, sparse bound-matrix technique, and graphic processing unit in conjunction with CUDA technology. The execution time and speed-up of proposed techniques are evaluated through comparing of different numbers of processors and unknowns. The results indicate that all presented techniques can significantly reduce computation time.

Index Terms—Microstrip structures, parallel algorithm, sparse bound-matrix, GPU, CUDA technology.

I. INTRODUCTION

Microstrip devices are widely used in modern microwave systems [1]–[5]. Microstrip transmission line, coupled lines, as well as multiconductor lines (Fig. 1) are used as basic elements in the design of such devices as filters [1], couplers [2], antennas [3], delay lines [4], [5], etc. Despite the fact that the microstrip lines have been known and used for more than 50 years, it is necessary to pay much attention to their analysis when new microstrip devices are designed. Most accurately microstrip structures may be analysed by numerical techniques such as: finite difference method (FDM) [6], finite elements method (FEM) [7], method of moments (MoM) [8], finite difference time domain (FDTD) method [9], as well as hybrid methods [10] and simulators [11].

The main drawback of numerical methods is their significant demand of computer resources, one of the main of which is the computation time, which in some cases can reach tens of hours [12]. Achievements of computer technology allow different ways to speed up calculation of electromagnetic problems. For example Cui *et al.* in [13] and Jobava *et al.* in [14] applying MoM to PC clusters for calculation correspondently of scattering by large 3D objects and currents distribution. Ergul and Gurel in [15] also use computer cluster to solve scattering problems. Angeli *et al.* in [16] demonstrated the implementation of FDM on 64 processors cluster. Yu *et al.* in [17] and Geterud *et al.* in

[12] realized FDTD method on computer clusters. There are also examples of the use of graphic processing unit (GPU) instead of a CPU to solve electromagnetic problems: Potratz *et al.* in [18] use FEM in conjunction with GPU to calculate scattering parameters of waveguide structures, and Livesey *et al.* in [19] apply GPU and CUDA technology to accelerate FDTD calculations. Motuk *et al.* in [20] presented implementing of FDM on a multiprocessor architecture on a FPGA device.

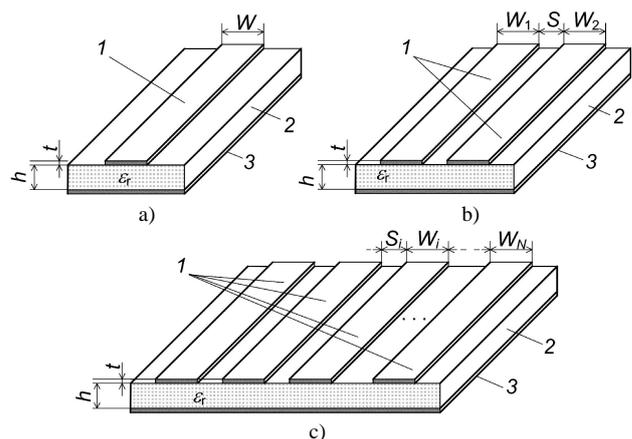


Fig. 1. Design of the microstrip structures: (a) transmission line, (b) coupled lines, (c) multiconductor lines, where 1 are microstrips; 2 is dielectric substrate; 3 is conductive shield.

Overview of open publications [12]–[20] reveals that computer hardware devices and other computations accelerating techniques are not used to analyse microstrip structures and we will try to do it in this paper.

The paper is organized as follows. In Section II, the parallel calculation of general parameters of microstrip structures using a computer cluster is given. Sparse band-matrices technique to accelerate calculation of microstrip structures parameters briefly described in Section III. The general principle of organizing calculations using GPU and CUDA technology and its application to the analysis of microstrip structures is presented in Section IV. Conclusions are discussed in Section V.

II. PARALLEL ALGORITHM AND COMPUTER CLUSTER

Almost every calculation process, especially cyclic calculations, can be organized in parallel manner, when calculations are distributed among more when one computers. Computation of a problem in parallel computer

network (cluster) can significantly reduce calculation time, however increased data transfer between computers is inevitable and must be taken into account.

In our previous work [6], we proposed a parallel algorithm for the analysis of coupled microstrip structures, i.e. calculation of dependency of electrical parameters of these structures on their design parameters.

Microstrip structures main electrical parameters – the effective permittivity $\nu_{\text{reff}ic,\pi}$ and characteristic impedance $Z_{0ic,\pi}$ for c- and π -normal waves can be found from corresponding capacities per-unit-length:

$$\nu_{\text{reff}ic} = \frac{C_{ic}}{C_{ic}^{(a)}}, \quad (1)$$

$$Z_{0ic} = \frac{1}{c_0 \sqrt{C_{ic} C_{ic}^{(a)}}}, \quad (2)$$

where c_0 is speed of light in vacuum; $C_{ic,\pi}$ is the i -th microstrip capacity per-unit-length correspondently for c- or π -normal wave; $C_{ic}^{(a)}$ is the capacity of the same microstrip when the substrate dielectric constant is changed to $\nu_r = 1$. According to (1) and (2) electrical parameters of coupled or the multiconductor microstrip lines for c- and π -normal waves in the line are calculated two times: first time with dielectric substrate and second, when substrate substituted with air ($\nu_r = 1$).

From that follows that the analysis of coupled microstrip structures can be arranged in a parallel fashion, combining 5 computers in the cluster (Fig. 2).

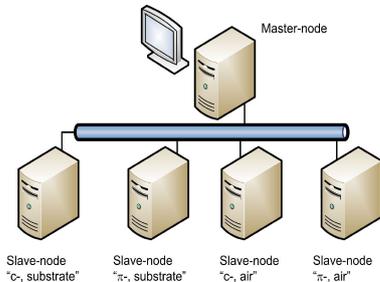


Fig. 2. Organization of a computer cluster for the analysis of coupled microstrip structures.

Cyclic calculations in the case of analysis of microstrip structures are necessary to perform when the influence of design parameters of these structures on their electrical parameters is investigated. In this case the master-node (Fig. 2) sends range of possible variations of design parameters and variation steps to slave-nodes. Slave-nodes, operating in a given cycle, calculate capacitances per-unit-length: Slave-node “c- substrate” – C_{ic} capacitance; Slave-node “ π - substrate” – $C_{i\pi}$; Slave-node “c- air” – $C_{ic}^{(a)}$, and Slave-node “ π - air” – $C_{i\pi}^{(a)}$ capacitance. After slave-nodes finish their calculations, they send the results to the master-node. The masternode sorts the received data and calculates the effective permittivity and impedance according to (1) and (2).

It should be noted, that any numerical method and quasi-TEM approach could be used as an analysis method in the proposed parallel algorithm. We have implemented the algorithm in [6] using FDM and solving the problem by iterative technique. Exploring the performance of the proposed parallel analysis algorithm we calculated in [6] the electrical parameters of a multiconductor microstrip line. The analysis area of 100×500 unknowns was used. It was found that the parallel algorithm execution time on the 5 computers cluster 3.4 times less than execution time on a single computer. This means that the increase in computing performance exceeds 250 percent.

III. BOUND SPARSE MATRIXES

Solution of partial derivative equations in finite difference method leads to large systems of algebraic equations and computation of these equations is done in two main techniques:

- Iterative;
- Coupled matrices.

Calculation time using iterative technique depends on the desired accuracy and problem area and can be very long. For calculation speed up a coupled matrices technique can be successfully used. According to this technique finite difference solution is found by composing and solving linear equation system.

According to finite difference method problem area is divided into square nodes mesh. Value of each node depends on mean of other nodes closest to it

$$\phi(i, j) = \frac{\phi(i+1, j) + \phi(i-1, j) + \phi(i, j+1) + \phi(i, j-1)}{4}, \quad (3)$$

where ϕ – electric field potential; i, j – indexes indicating the position of potential in 2D. So remote nodes have not influence on the calculated potential (i, j) . Therefore analysis of all nodes in the problem area can be found in resolving the equation

$$[A] \times [X] = [B], \quad (4)$$

where $[A]$ is coefficient matrix with many zero elements; $[X]$ is a vector consisting of unknown node values; $[B]$ is a vector consisting of known node values. Unknown nodes vector $[X]$ can be calculated using for example this equation

$$[X] = [A]^{-1} \times [B], \quad (5)$$

where $[A]^{-1}$ is the inverse matrix of coefficients. By solving (5), unknown potential vector can be obtained and recomposed to potential distribution matrix – the problem area. Potential distribution can be farther analysed to find the device electrical parameters e.g.: electric charge density, capacity per-unit-length and so on.

Since calculated potential depends only on the neighbouring potentials – coefficient matrix $[A]$ consist mostly of zero elements, which takes significant amount of memory – each “double precision” type value occupies 8 bytes. It is possible to reduce the memory space occupied by the coefficient matrix and to speed up the calculations using sparse matrices. In sparse matrices only non-zero elements

are stored in memory. Unknown nodes vector $[X]$ can also be found through various elimination methods (Gaussian, Gaussian and Jordan et al.).

In order to evaluate the speedup of the FDM calculations the coupled microstrip lines will be analysed. Their constructive parameters are as follows: substrate dielectric constant $\epsilon_r = 6.0$, normalized microstrips width $W_1/h = W_2/h = 0.5$, normalized space between microstrips $S/h = 0.5$.

Electrical parameters and potential distribution calculation speeds by different techniques are represented in Fig 3. Two electrical parameters were calculated in the process: characteristic impedance Z_0 and effective permittivity ϵ_{eff} (Table I). The investigation area chosen square and one side varied from 52 to 122.

TABLE I. ELECTRICAL PARAMETERS OF COUPLED MICROSTRIP LINES* CALCULATED BY BOUND SPARSE MATRICES TECHNIQUE.

$Z_{0c}[\Omega]$	$Z_{0f}[\Omega]$	ϵ_{effc}	ϵ_{efff}
101.770	61.204	3.996	3.563

Note: Design parameters are the follows: $\epsilon_r = 6.0$, $W_1/h = W_2/h = S/h = 0.5$.

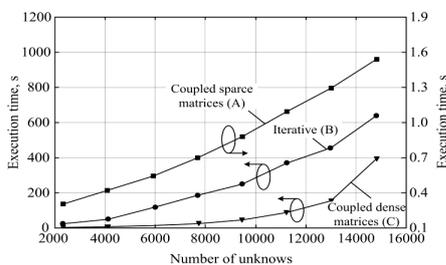


Fig. 3. Execution times of the implemented algorithms, where A – algorithm using coupled sparse matrices technique, B – iterative algorithm and C – algorithm using coupled dense matrices technique.

Figure 3 shows execution times of different implemented algorithms with different number of unknowns. Comparison was done with implemented code in Matlab and as A curve shows sparse matrix implementation vastly reduces calculation time.

IV. GPGPU & CUDA TECHNOLOGY

Analysing of the microstrip devices can be done using a general-purpose computing on graphics processing units (GPGPU). These processors can have up to 512 and even more processor cores (so-called general-purpose streaming multiprocessors) it means they have 100 times more cores than usual general-purpose CPU has. Their advantage is also that it is not additional specialized computing device. All the GPGPU are embedded on all desktop computers and laptops manufactured from a couple of years ago. Also they are extremely fast and efficient to perform operations with real numbers and with a high degree of data parallelism. In this way, computing performance increases many times comparing with a general-purpose CPU. It is becoming increasingly prevalent to develop and investigate techniques to allow using of these computing capabilities.

There are two competing GPGPU programming platforms. A patented CUDA technology developed by NVIDIA Company [21], which integrating technology only in company produced GPGPU's. However, with the using of NVIDIA manufactured GPGPU video card the developing programs in CUDA is free. It should be also noted that

CUDA technology appeared a little earlier than the second – OpenCL technology [22], so, at this time, it is more developed and designed scientific and engineering solutions specifically for CUDA technology. On the other hand, it is becoming now more pervasive technology – OpenCL.

OpenCL programming technology was created a bit later, who not only supports GPGPU's, but also the general purpose CPU and special accelerators, they are used in mobile phones and embedded systems. This technology is completely free, so it can be integrated into any microprocessor or accelerator by any company and any scientist who wishes to build applications. The main problem of this technology, where is no many created or modified mathematical functions library for OpenCL technology yet. Therefore, in order to perform vector and matrix operations, it is needed to self-create the desired function, or settle for a lower calculation speed compared with CUDA technology.

By solving electromagnetic problems the iterative calculations are applied mostly because iterations could reduce the space occupied by variables in main memory. But iterative calculations limits the accuracy of the results, because of a given accuracy level for the iterative calculation. On the other side, the direct linear solvers allow to instantly get the correct result. Downside for direct linear solvers is that usage amount of main memory is significantly greater, what was recently simply impossible. Also iterative calculations are more complicated to split into smaller tasks in order to distribute them to parallel computing systems, than solving a system of linear equations using the direct methods. Undoubtedly solving linear equations also apply iteration calculations, but in this case the system of linear equations with special methods is decomposed into blocks those facilitate the distinction between linear calculations of parallel systems.

In order to evaluate the speedup of the FDM calculations the coupled microstrip lines will be analysed. Their design parameters are the same as described in Section III.

To solve linear equations system two libraries CULA and ViennaCL [23] will be used. Gaussian elimination technique will be used for execution time comparison. These libraries designed to solve linear equations system using dense matrices and sparse matrices, but for sparse matrices created coefficient matrix must be converted into matrix storage format. CULA library is optimized and works only with CUDA technology, ViennaCL can operate with OpenCL technology also.

Curves in Fig. 4 show implemented algorithms execution time with different number of unknowns to find (problem area). It is seen that, comparing execution time of CULA library curve and authors implemented Gaussian elimination technique curve, when 2500 unknowns were found, differ 120 times, and when 14400 unknowns were calculated – these curves differ, more than 1000 times. Comparing curves corresponded to ViennaCL library and Gaussian elimination technique it is evidence that execution time in both case practically not differs for low number of unknowns – 6400 and at 14400 number of unknowns differ only 1.24 times. Such negligible difference between calculations using ViennaCL library and Gaussian elimination technique can be

explained by the fact that the larger set of features and hardware support in ViennaCL library case typically come at the cost of lower performance comparing with CUDA based implementations.

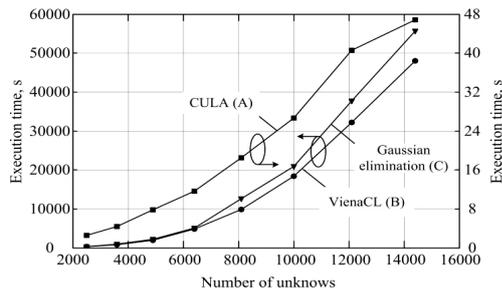


Fig. 4. Execution times of the implemented algorithms.

This is also partly due to the fact that CUDA is tailored to the architecture of NVIDIA products, while OpenCL represents in some sense a reasonable compromise between different many-core architectures. Also one of the reasons is the different focus of ViennaCL – solvers for sparse instead of dense linear algebra.

Calculated electrical parameters of coupled microstrip lines analysed by GPGPU & CUDA technology are presented in Table II.

TABLE II. ELECTRICAL PARAMETERS OF COUPLED MICROSTRIP LINES* CALCULATED BY GPGPU & CUDA TECHNOLOGY.

$Z_{0e} = h $	$Z_{0f} = h $	V_{reffc}	V_{refff}
101.204	61.195	4.091	3.581

Note: Design parameters are the same as in Table I.

V. CONCLUSIONS

Accurate calculation of parameters of microstrip structures with numerical techniques requires the solution of dense matrix equations involving thousands of unknowns. Solution of this large problem takes long time. We present three techniques for such computations acceleration: parallel algorithm implemented in computer cluster, sparse bound-matrix technique, and graphic processing unit (GPU) in conjunction with CUDA technology. The execution time and speed-up of proposed techniques are evaluated through comparing of different numbers of processors and unknowns. The results indicate that all presented techniques can significantly reduce computation time: reduction of the parallel algorithm execution time is inversely proportional to the number of computers in the cluster, sparse bound-matrix is capable of hundreds of times to reduce the computation time compared with the iterative technique, GPUs reduce computation time in thousands of times compared with conventional mathematical techniques.

REFERENCES

- [1] C. J. Kikkert, "A design technique for microstrip filters", in *Proc. 2nd Intern. Conf. on Signal Processing and Communication Systems, (ICSPCS 2008)*, Gold Coast, 2008, pp. 1–5.
- [2] X. Tang, K. Mouthaan, "Analysis and design of compact two-way Wilkinson power dividers using coupled lines", in *Proc. Microwave Conf., (APMC 2009)*, Asia Pacific, Singapore, 2009, pp. 1319–1322.
- [3] N. Apaydin, K. Sertel, J.-L. Volakis, "Nonreciprocal leaky-wave antenna based on coupled microstrip lines on a non-uniformly biased ferrite substrate", *IEEE Trans. on Antennas and Propagation*, vol. 61, pp. 3458–3465, 2013. [Online]. Available: <http://dx.doi.org/10.1109/TAP.2013.2257646>

- [4] E. Metlevskis, R. Martavicius, "Computer models of meander slow-wave systems with additional shields", *Elektronika ir elektrotechnika*, no. 3, pp. 61–64, 2012. [Online]. Available: <http://dx.doi.org/10.5755/j01.eee.119.3.1365>
- [5] A. Lujambio, et al., "Dispersive delay line with effective transmission-type operation in coupled-line technology", *Microwave and Wireless Components Letters*, vol. 21, pp. 459–461, 2011. [Online]. Available: <http://dx.doi.org/10.1109/LMWC.2011.2162822>
- [6] R. Pomarnacki, A. Krukoni, V. Urbanavicius, "Parallel algorithm for the Quasi-TEM analysis of microstrip multiconductor line", *Elektronika ir Elektrotechnika*, no. 5, pp. 83–86, 2010.
- [7] Y. Yan, P. Pramanick, "Finite-element analysis of generalized V- and W-shaped edge and broadside-edge-coupled shielded microstrip lines on anisotropic medium", *IEEE Trans. on MTT*, vol. 49, pp. 1649–1657, 2001. [Online]. Available: <http://dx.doi.org/10.1109/22.942579>
- [8] M. Farina, A. Morini, T. Rozzi, "On the derivation of coupled-line models from EM simulators and application to MoM analysis", *IEEE Trans. on MTT*, vol. 53, pp. 3272–3280, 2005. [Online]. Available: <http://dx.doi.org/10.1109/TMTT.2005.857125>
- [9] S. Ahmed, A. Schuchinsky, "Full-wave FDTD analysis of UWB pulses on printed coupled lines", in *Proc. of the 36th European Microwave Conf.*, Manchester, 2006, pp. 9–12.
- [10] M. Khalaj-Amirhosseini, "Analysis of coupled or single nonuniform transmission lines using the equivalent sources method", in *Proc. Int. Symposium on Microwave, Antenna, Propagation, and EMC Technologies for Wireless Communications*, Hangzhou, 2007, pp. 1247–1250.
- [11] K.-H. Tsai, C.-K. C. Tzuang, "Mode symmetry analysis and design of CMOS synthetic coupled transmission lines", *IEEE Trans. on MTT*, vol. 59, pp. 1947–1954, 2011. [Online]. Available: <http://dx.doi.org/10.1109/TMTT.2011.2155666>
- [12] E. Geterud, M. Hjelm, T. Ciamulski, M. Sypniewski, "Simulation of a lens antenna using a parallelized version of an FDTD simulator", in *Proc. 3rd European Conf. Antennas and Propagation, (EuCAP 2009)*, Berlin, 2009, pp. 3457–3461.
- [13] Z.-W. Cui, et al., "Parallel MoM solution of JMCIE for scattering by 3-D electrically large dielectric objects", *Progress in Electromagnetics Research M*, vol. 12, pp. 217–228, 2010. [Online]. Available: <http://dx.doi.org/10.2528/PIERM10042607>
- [14] R. Jobava, et al., "Solving large scale EMC problems using Linux cluster and parallel MoM", in *Proc. 9th Int. Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory, (DIPED 2004)*, Tbilisi, 2004, pp. 83–86.
- [15] O. Ergul, L. Gurel, "Rigorous solutions of electromagnetic problems involving hundreds of millions of unknowns", *IEEE Antennas and Propagation Magazine*, vol. 53, pp. 18–27, 2011. [Online]. Available: <http://dx.doi.org/10.1109/MAP.2011.5773562>
- [16] J. P. De Angeli, A. M. P. Valli, N. C. Reis, A. F. De Souza, "Finite difference simulations of the Navier-Stokes equations using parallel distributed computing", in *Proc. 15th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2003)*, Los Alamitos, 2003, pp. 1–8.
- [17] W. Yu, et al., "High-performance conformal FDTD techniques", *IEEE Microwave Magazine*, vol. 11, pp. 42–55, 2010. [Online]. Available: <http://dx.doi.org/10.1109/MMM.2010.936496>
- [18] C. Potratz, H.-W. Glock, U. Rienen, "Time-domain field and scattering parameter computation in waveguide structures by GPU-accelerated discontinuous-Galerkin method", *IEEE Trans. on MTT*, vol. 59, pp. 2788–2797, 2011. [Online]. Available: <http://dx.doi.org/10.1109/TMTT.2011.2166163>
- [19] M. Livesey, et al., "Development of a CUDA implementation of the 3D FDTD Method", *IEEE Antennas and Propagation Magazine*, vol. 54, pp. 186–195, 2012. [Online]. Available: <http://dx.doi.org/10.1109/MAP.2012.6348145>
- [20] E. Motuk, R. Woods, S. Bilbao, "Parallel implementation of finite difference schemes for the plate equation on a FPGA-based multi-processor array", in *Proc. 13th European Signal Processing Conf. (EUSIPCO 2010)*, Antalya, 2010, pp. 1–4.
- [21] J. Sanders, E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Upper Saddle River: Addison-Wesley, 2011, pp. 1–20.
- [22] P. O. Jaaskelainen, C. S. et al., "OpenCL-based design methodology for application-specific processors", in *Proc. Embedded Computer Systems (SAMOS 2010), Inter. Conf.*, Samos, 2010, pp. 223–230.
- [23] K. Rupp, "ViennaCL and PETSc tutorial", Mathematics and Computer Science Division Argonne National Laboratory, pp. 1–31, 2013. [Online]. Available: <http://www.karlsruhp.net/wp-content/uploads/2013/05/FEMTEC2013-tutorial.pdf>