# CHAMELEON-RF PROTOTYPE CODE VALIDATION ACTIVITIES AND COMPUTATIONAL SPEEDUP THROUGH PARALLELIZATION

Sebastian GIM[1]

*Continua miniaturizare prin scalare a dispozitivelor electronice în zona nanometrilor, cât şi sporirea frecvenţei lor de lucru în zona multi-GHz, determină efecte noi al căror impact anterior a fost unul neglijabil, dar care acum prezintă provocări mari şi o complexitate fără precedent în ceea ce priveşte proiectarea circuitelor integrate cu semnal mixt analog-digital. Proiectul CAMELEON-RF [1] a fost conceput pentru a răspunde acestor provocări. Lucrarea de faţă prezintă rezultate recente obţinute în validarea codului prototip de modelare elctromagnetică numit Chamy, dezvoltat în acest proiect în cadrul UPB-CIEAC, prin comparaţie cu comportarea structurilor de test fabricate de AustriaMicroSystems, un partener în consorţiul proiectului. Validarea este un pas important în ceea ce priveşte verificarea funcţionării corecte a programului. Procesul de validare utilizează recent finalizata platformă Chamy, care permite extragerea modelelor compacte ale structurilor pasive din circuitele integrate, parametrizate multiscalar, utilizând metoda descompunerii în subdomenii şi conceptele de element electormagnetic de circuit şi cel de conector pentru interacţiune cu mediul elctromagnetic. Cu toate acestea Chamy este un cod serial, conceput să se execute secvenţial, pe un singur procesor. Această lucrare prezintă, de asemenea, activitatea desfăşurată în vederea paralelizării unor părţi ale platformei Chamy şi discută diversele strategii de paralelizare, prezentând accelerarea obţinută prin acestea.*

*The continuous scaling of electronic devices into the nanometer region and high frequencies of operation well into the multi-GHz region has given rise to new effects that previously had negligible impact but now present greater challenges and unprecedented complexity to designing successful mixed-signal silicon. The CHAMELEON-RF project [1] was conceived to address these challenges. This paper presents recent validation results of the prototype code called Chamy, which was developed at CIEAC, against significant contemporary industrial benchmarks fabricated at AustriaMicroSystems, a consortium partner. Validation is an important milestone in the verification of correct program operation. The validation effort uses the recently completed Chamy platform demonstrator which enables multi-scale parameterized compact modeling of passive integrated structures using the domain decomposition approach and the concept of electromagnetic circuit elements and hooks. Chamy however is serial code. This paper also presents the work done in parallelizing some parts of Chamy, discusses parallelization strategies and presents the computational speedup accomplished.*

[1] PhD student, Center of Computer-Assisted Electrical Engineering (CIEAC), University POLITEHNICA of Bucharest, Romania, e-mail: simona@lmn.pub.ro

**Keywords:** Compact modeling, Numerical Simulation, Coupled Problems

## 1. Introduction

The shrinking transistor according to Moore's Law has yielded many improvements in microprocessor performance [2]. However, when VLSI devices are scaled into the nanometer region and operates in the multi-GHz region, there are new effects that were previously negligible but now present challenges to continued scaling. Modelling and simulating these nano electronic devices are a challenge. In an effort to mitigate design complexity, compact models of on-chip passive integrated structures [3] are needed. At lower frequencies, simplifying assumptions can be made and traditional lumped models of circuits offer sufficient accuracy in modeling on-chip interconnect passives. However at higher frequencies, wave propagation along these interconnects are greatly influenced by changes in the material properties of the environment. This is further exacerbated by complex routing scenarios in modern day VLSI designs. This leads to a transition to a completely new level of complexity where accurate solution of Maxwell equations instead of traditional lumped circuit approach are required which often necessitates computationally intensive 3D simulations which has resulted in an increase in engineering resources essential for a successful design and lengthy iterations of product prototyping. The ITRS roadmap [4] projects extreme scaling of CMOS technology until the 10 nm region and operating frequencies of up to 60 GHz in future generation devices. At such close proximity coupled with fabrication process variations, substrate noise and EM coupling between circuit components make mixed-signal RF silicon designs extremely challenging [5]. Hence, in order to mitigate electronic design challenges in the nanometer scale, the development of adequate mathematical models and numerical methods that is implemented within a coherent electronic design automation (EDA) software framework is sorely needed [6].

Because of this, the CHAMELEON-RF project was conceived as part of an initiative to address these issues [1]. The project is a research platform for the development of prototype tools and methodologies for comprehensive high accuracy modeling of on-chip electromagnetic effects without simplifying apriori assumptions in order to manage the unprecedented complexity faced when designing next generation highly integrated mixed-signal architectures [7].

This paper describes the work done, insight gained and results when validating Chamy with measurements of fabricated benchmark structures from a consortium industrial partner's site (AustriaMicroSystems). The RFPAD is a significant contemporary structures widely used within the semiconductor industry. The benchmark is modelled using the domain decomposition (DD) approach through the use of electromagnetic circuit elements (EMCE) concept.

Magnetic coupling between domain decomposed EMCEs are described by means of electromagnetic interconnectors or hooks that describe bidirectional electrical or magnetic coupling. Domain decomposition with EMCEs open the doorway towards tractable multiscale simulation of complex mixed signal designs faced by industry today [8]. The remaining sections of this paper are organized as follows. Section 2 presents the key points of the CHAMELEON-RF framework and workflow which enables tractable multi-scale parameterized compact modeling of passive integrated structures. Section 3 presents the CHAMELEON-RF system state space matrices and modeling techniques used. Section 4 examines the RFPAD benchmark test case using the recently completed CHAMELEON-RF software prototype Chamy written using Matlab. Validation is an important milestone in the verification of correct program operation. Section 5 then describes the work done in parallelizing some parts of Chamy, discusses parallelization strategies and presents the computational speedup accomplished. The paper is finally concluded in Section 6.

## 2. Chameleon-RF Workflow

Contemporary mixed signal design workflows involve ad-hoc iterations of incremental design improvements based upon tweaking a baseline circuit using a gamut of discrete software tools, some analytic methods and plenty of heuristics. It is an iterative cycle of SPICE circuit simulation, silicon fabrication, measurement and parameter tweaking to meet design performance targets. Many fabrication spins and prior experience is often needed to isolate coupled effects before tweaking SPICE parameters in order to compensate for these undesirable parasitic effects and variations in fabrication processes [9].

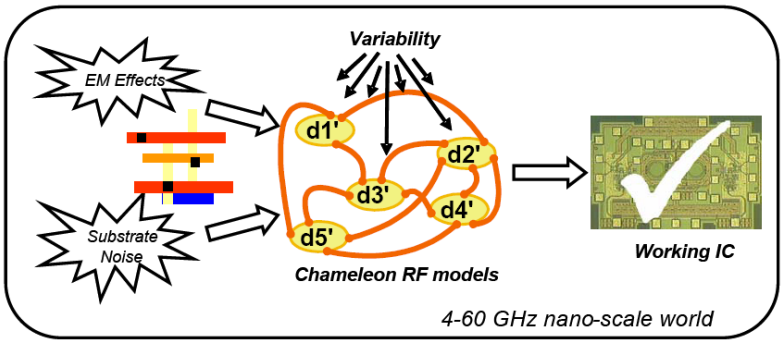



Fig. 1. Chameleon-RF concept

The CHAMELEON-RF nano-EDA research platform (**Fig. 1**) incorporates a novel dual Finite Integration Technique (dFIT) EM field simulator with possibility for systematic All Level Reduced Order Modeling (ALROM) [10-12].

This approach has advantages over alternative approaches in full 3D field simulators - such as FEM, BEM [13] etc. - which, although enable greater accuracy, are intractable for most practical real world designs and difficult to integrate with existing EDA suites. On the contrary, the CHAMELEON-RF workflow proposes a direct path from design layout file to an accurate parameterized compact model [14] that is ready for design verification [1].

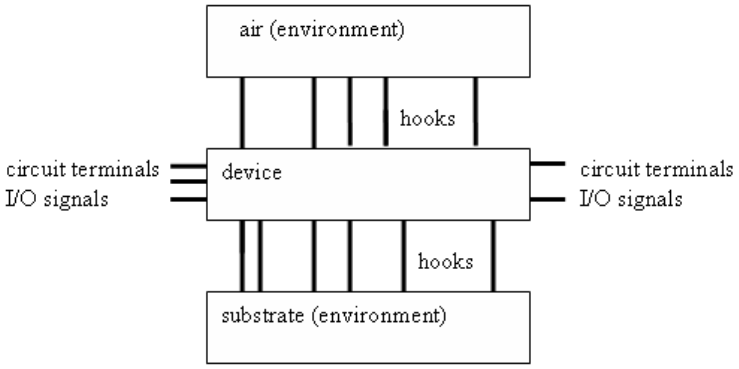



Fig. 2. Domain decomposition within integrated circuit using EMCE + hooks

Complexity is managed through the use of domain decomposition and hierarchical substrate structuring [15]. When the modeling domain is decomposed into distinct domains connected by hooks (**Fig. 2**), computational savings can be obtained. Non essential domains, such as the substrate or air layers, can be simulated just in the simplified field regime such as electro static (ES), magneto static (MS), electro quasi static (EQS), magneto quasi static (MQS) regimes instead of a full wave (FW) analysis. Only critical structures of interest need to be simulated in FW which results in computational savings and a more tractable model. A more detailed treatment of the system state space matrices used and EMCE boundary conditions is given in the next section.

### 3. System State Space Matrices

Numerical methods are employed in computational electromagnetics (CEM) to efficiently approximate solutions to real world problems of electromagnetic interaction between objects in the real world environment where analytical or closed form solutions to the classical Maxwell equations (1-4) are not readily derivable [16].

$$curl\ \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \tag{1}$$

$$curl\ \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2}$$

$$div\,\mathbf{D} = \rho \tag{3}$$

$$div\,\mathbf{B} = 0 \tag{4}$$

$$\mathbf{n}\,curl\,\mathbf{E}(P,t) = 0 \quad for \quad \forall P \in \Sigma - S_k'' \tag{5}$$

$$\mathbf{n}\,curl\,\mathbf{H}(P,t) = 0 \quad for \quad \forall P \in \Sigma - S_k'' \tag{6}$$

$$\mathbf{n} \times \mathbf{E}(P,t) = 0 \; for \; \forall P \in \bigcup S_k' \tag{7}$$

$$\mathbf{n} \times \mathbf{H}(P,t) = 0 \; for \; \forall P \in \bigcup S_k'' \tag{8}$$

Generally, various approaches within CEM involve the discretization of a continuous domain of interest by a grid (with appropriate boundary conditions) before iteratively or otherwise solving the Maxwell equations for each point in the grid [17]. Both orthogonal and non orthogonal grids are possible depending on application scenario. Likewise, multigrid and flexible grid methods for structured and unstructured grids are also available and either the integral [18] or differential [19, 20] form could be used for solutions in time or frequency domain over n dimensions.

$$\begin{cases} (2) \\ (4) \end{cases} \Rightarrow \begin{cases} \oint \mathbf{E} d\mathbf{r} = -\iint \frac{\partial \mathbf{B}}{\partial t} d\mathbf{A} \\ \oiint \mathbf{B} d\mathbf{A} = 0 \end{cases} \Rightarrow \begin{cases} \mathbf{C}v = -\frac{d\varphi}{dt} \\ \mathbf{D}'\varphi = 0 \end{cases}$$

$$\begin{cases} (1) \\ (3) \end{cases} \Rightarrow \begin{cases} \oint \mathbf{H} d\mathbf{r} = \iint (\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}) d\mathbf{A} \\ \oiint \mathbf{D} d\mathbf{A} = \iiint \boldsymbol{\rho} dv \end{cases} \Rightarrow \begin{cases} \mathbf{C}'\mathbf{u} = \mathbf{i} + \frac{d\boldsymbol{\psi}}{dt} \\ \mathbf{D}\boldsymbol{\psi} = \mathbf{q} \end{cases} \tag{9-13}$$

$$\Rightarrow div\mathbf{J} = -\frac{\partial \rho}{\partial t} \Rightarrow \iint \mathbf{J} d\mathbf{A} = -\iiint \frac{\partial \rho}{\partial t} dv \Rightarrow \mathbf{D}\mathbf{i} = -\frac{d\mathbf{q}}{dt}$$

$$\begin{cases} \mathbf{B} = \mu \mathbf{H} \\ \mathbf{D} = \varepsilon \mathbf{E} \\ \mathbf{J} = \sigma \mathbf{E} \end{cases} \Rightarrow \begin{cases} \varphi = \mathbf{M}_{\mu}\mathbf{u} = \mathbf{M}_{\nu}^{-1}\mathbf{u} \\ \boldsymbol{\psi} = \mathbf{M}_{\varepsilon}\mathbf{v} \\ \mathbf{i} = \mathbf{M}_{\sigma}\mathbf{v} \end{cases} \tag{14-16}$$

Chamy uses the concept of the EMCE [21] ()which imposes the boundary conditions (5-8). FIT [18] is then used to obtain a set of discrete algebraic equations (9-13) with associated material properties (14-16). State space matrices to describe the system of the form (17-18) can then be constructed and solved using an appropriate direct or iterative solver. The solution of such a system consisting of large, sparse, unsymmetrical matrixes is computationally the most costly step of the entire process.

$$\begin{cases} C\frac{d\mathbf{x}}{dt} = -G\mathbf{x} + B\mathbf{u} \\ \mathbf{y} = L\mathbf{x} \end{cases} \tag{17-18}$$

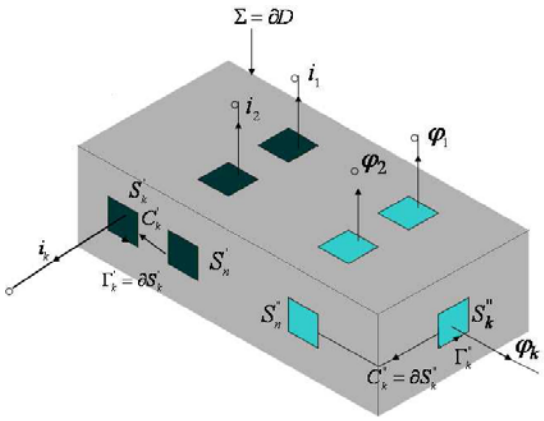



Fig. 3. Electromagnetic Circuit Element (EMCE)

Creative use of domain decomposition, multi grid techniques or reduced order modelling techniques (ROM) can be selectively applied at all levels of the process to efficiently prune down DoFs. However, the simulation of complex systems within a reasonable amount of time remains a computational challenge.

## 4. Validation of RFPAD Benchmark

The Chameleon-RF prototype software called Chamy that was developed at CIEAC was validated against measurements of fabricated benchmark structures from industrial partners in the consortium network. The benchmark structure that was validated by the author is an RFPAD which illustrates the part of the Chameleon-RF workflow which deals with compact circuit model extraction. This compact circuit model, once extracted, can then be reused by design engineers in an existing linear circuit simulator such as PSPICE, Cadence SPECTRE or Mentor Graphics ELDO to rapidly simulate complex mixed signal real world designs within a certain degree of accuracy.

This benchmark structure is a full stack (four metal layers) RF pad placed over a 3 μm thick nwell layer that acts as shielding. RF pads are a widely used structure in the semiconductor industry for off-chip packaging connections via wire-bond leads or flip chip technology and also functions as a filter for high frequency components of the signal.

### 4.1 Model Creation

For benchmark validation purposes, a simple model (Fig. 4) was constructed using the Chameleon prototype software.

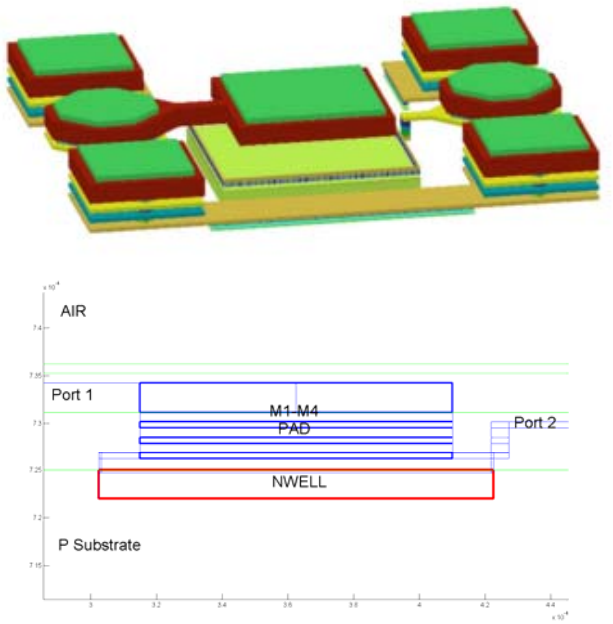



Fig. 4. Fabricated benchmark structure (Top). Cross section view of RFPAD Chamy model (Bottom)

It consists of all the metal layers of the RFPAD suspended over an nwell. Port 1 is voltage excited on the xmin face and the port 2 is excited on the xmax face. The p-substrate is grounded. In the measurements, the signal is injected via port 1 and the output energy is collected in port 2 via a ring at the periphery of the nwell. Measurements are carried out over the frequency range of 50MHz to 40GHz.

### 4.2 Validation

First using a minimal grid of nx=10 ny=15 nz=10, a system matrix size of 6983 Degrees of Freedom (DOF) was generated and simulated. **Fig. 5** illustrates the comparison between simulation results and measurement.

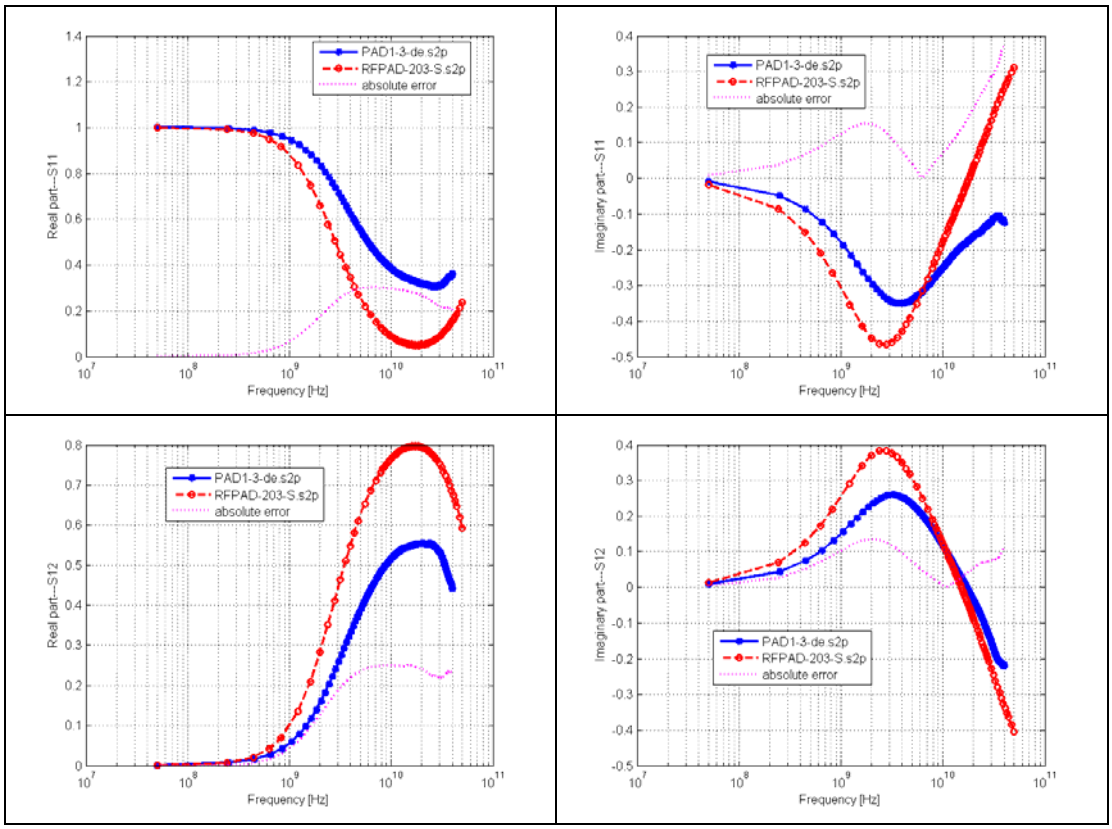
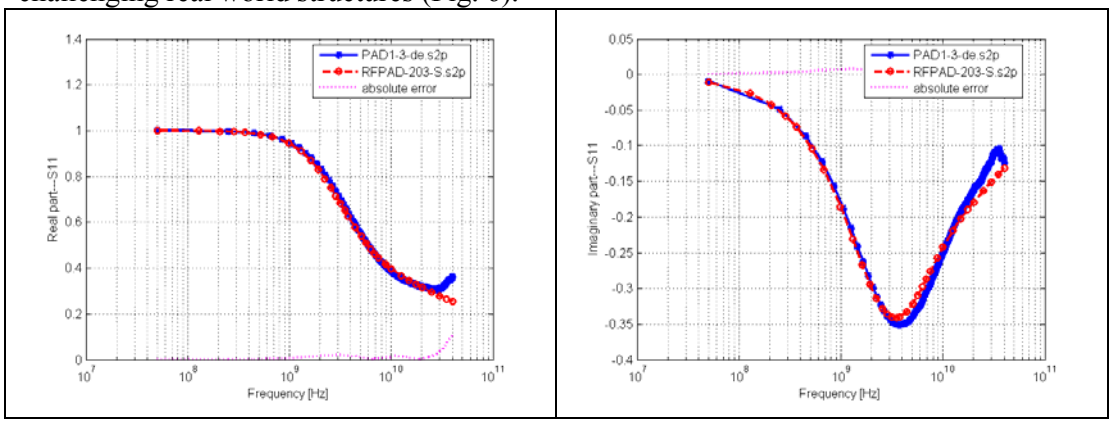



Fig. 5. Minimal grid S11 and S21. Measurement (Solid line) Simulation (Broken Line)

After refining the grid to nx=11 ny=16 nz=13 and a matrix size of 11245 DOFs, a better correspondence between the measurement and simulated graphs can be seen and illustrates the effectiveness of Chamy in handling relevant challenging real world structures (Fig. 6).



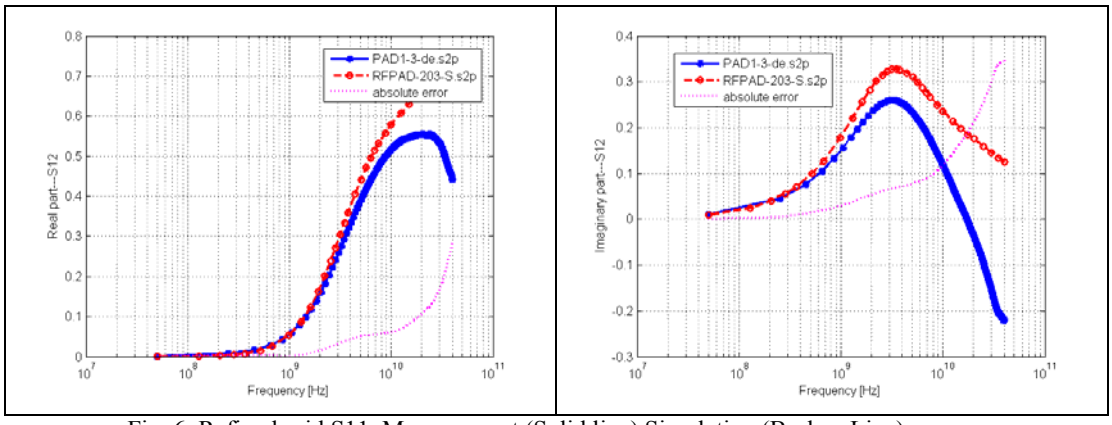



Fig. 6. Refined grid S11. Measurement (Solid line) Simulation (Broken Line)

The key observation to obtaining proper model fit is keep discretization meshes as square as possible and determining an appropriate boundary for simulation domain. DOFs can be reduced by adding denser discretization meshes only at critical places where higher resolution is required. Once the model has been tuned to an arbitrary desired accuracy, then a compact circuit model (SPICE netlist) can be extracted which is stored in an appropriate RF-block reusable design library. It can then be reused by design engineers in an existing linear circuit simulator.

## 5. Parallelization

The Chameleon-RF workflow consists of creating the extensible markup language (XML) device description file and associated materials file, generating the grid over the defined domain, generating the state space matrices (XML → SYS) before finally simulating the system over a frequency range and writing the results into the Touchstone SNP format (SYS → SNP). XML → SYS and SYS → SNP basically create and solve the state space matrices of the described system (17, 18). In the XML→SYS part, an appropriate discretization mesh is first created for a structure. After generating the grid using the grid_generate function, the matrices C, G, B, L, F are generated using the generate_matrices_parametricFIT_v4 function before being written into a matrix file. In the SYS→SNP part of Chamy, the MIMO state space system has been previously constructed can then be solved by an appropriate numerical method (direct or iterative).

Original Chamy however is serial code and simulation of complex systems took a long time. Hence, the ToK4nEDA project was proposed to reinforce the competence in the area of High Performance Computing (HPC) hardware, software, and grid solutions at 'Politehnica" University of Bucharest Romania that

complement the ongoing nano electronic design automation (nEDA) project activities through the recruitment of external experienced researchers. A Beowulf cluster named Advanced Technology Laboratory Server (ATLAS) was setup at CIEAC under the ToK4nEDA project for the purposes of research, training and transfer of knowledge in high performance computing. The ATLAS cluster is a hybrid cluster consisting of state of the art technologies in high performance computing. It contains several 4U accelerated compute nodes, either 2 socket AMD Opteron hosts or 2 socket INTEL Nehalem hosts with either ATI Firestream or NVIDIA Tesla GPGPU floating point accelerators. Accelerated nodes can also be used as normal nodes by ignoring the additional GPGPU capability. This configuration gives the most flexibility for coexistence of existing application using conventional cluster technology plus room for expansion, research and experimentation with new technologies such as floating point accelerators (GPGPUs) or integer accelerators (FPGAs). Recently with the advent of GPGPUs, unsurpassed additional floating point performance has become available to the high performance computing scientific community. GPGPUs have been identified as the enabling technology for peta-scale computing. With the use of GPGPUs, high accuracy simulations of complex systems can be obtained within a tractable amount of time. This is an emerging area of research and has been identified by the European Commission as a FP7 Future and Emerging Technology (FET) which is actively researched. The inclusion of GPGPUs in the cluster will enable CIEAC researchers to participate in research in this area also. All nodes (Master, Compute) are connected by a high bandwidth low latency Infiniband 20 GBps switch fabric with RDMA capability. A separate 1 GBps Ethernet network is maintained as backup and also for off band signalling and maintenance purposes.

In parallelizing Chamy, it was observed that the calculation of the state space matrix solution for a particular frequency can be done independently in parallel. A clean task parallel version (64 bit) of SYS → SNP was written which can be run on the ATLAS Beowulf cluster. The new function of parallel SYS → SNP is called ComputeSYS2SNP_parallel64.m. ComputeSYS2SNP_parallel64 accepts as input the SysFilename, SnpInFilename, AFSinfo, SnpOutFilename which follows the original serial Chamy input parameters. The pseudo code of parallel SYS→SNP is as follows:

```
Function
Parallel_sys2snp (SysFilename,  SnpInFilename,  AFSinfo,
SnpOutFilename)

Looking for cluster job managers or creating local
manager();

[C G B L F] = Loading Matrices(SysFilename);
freq_list   = LoadSnpFrequencies(SnpInFilename);
```

```
if (AFSinfo.flag == false)
      Distributed calculation using each given
      frequencies(freq_list);
else
      Distributed calculation of each frequency using
      AFS(freq_list);
end

WriteSnpFile(SnpOutFilename);
```

### 5.1 Non Adaptive Frequency Sampling

Non Adaptive Frequency Sampling (AFS) distributed calculation is done by creating n independent tasks for each given frequency. This will then be submitted to the cluster for evaluation by worker nodes as they become available. Both the AFS and non AFS distributed cases have been written in such a way as to use the cluster that could have multiple simultaneous users on it in a neutral manner. The pseudo code of Distributed non AFS calculation using each given frequency (freq_list) is as follows:

```
Function
Distributed calculation using each given
frequencies(freq_list)
% Create a job on the JobManager Eg. Diana-JMgr
nonAFSjob = createNonAFSjob(jobmanager);
% Create distributed tasks and attach to job
for idx_f = 1:length(freq_list)
      createTask(nonAFSjob, @Compute_One_Frequency, 1,
      {freq_list(idx_f), SysFilename} );
end
% Displaying job manager statistics
jmgr.get();
% Submitting job to active queue
submit(nonAFSjob);
% Waiting for completion of jobs
waitForState(nonAFSjob, 'finished');
% Getting results from JobManager
results = getAllOutputArguments(nonAFSjob);
% Displaying job statistics
nonAFSjob.get();
% Destroying job on JobManager to free up resources.
destroy(nonAFSjob);
```

### 5.2 Adaptive Frequency Sampling

AFS distributed calculation is done by first dividing the given frequency range into sub bands for distributed processing. A distributed calculation is then performed for the frequencies at the band edges. Then a distributed calculation of the frequencies within each sub band is performed. This is essentially a parallel

implementation of a depth first search with branch and bound (DFBB) whereby each processor asynchronously searches a disjoint sub-tree without communication. The first pass is a data partitioning phase which performs a breath first search (BFS) partitioning of the first few levels of the tree. The second pass consists of an asynchronous search phase where each of the sub-trees are searched using DFBB. Finally, the results of the two distributed calculations are assembled. The code has been written in a way such that results are calculated in correct order so no further sorting is required, thus saving some computation. Each frequency computation is performed only once and overlap computations are avoided. The pseudo code of the distributed calculation of each frequency using AFS( freq_list) is as follows:

```
Function
Distributed calculation of each frequency using
AFS(freq_list)
% Creating AFS job and AFS sub-band job
AFSsub = createJob(jmgr, 'Name', 'AFS sub-band job');
AFSjob = createJob(jmgr, 'Name', 'AFSJob');
% Dividing frequency range into sub bands for distributed
processing
AFS_freq_dist =
AFS_GetFrequencyDistribution(fmin,fmax,AFSinfo,n_tasks);
AFS_freq_dist = [fmin; AFS_freq_dist; fmax];
% Distributed calculation for each band edge frequency
for idx_f = 1:length(AFS_freq_dist)
      createTask(AFSsub, @Compute_One_Frequency, 1,
      {AFS_freq_dist(idx_f), SysFilename} );
end
% Submitting job to active queue
submit(AFSsub);
% Waiting for completion of jobs
waitForState(AFSsub, 'finished');
% Getting results from JobManager
AFS_freq_dist_yc = getAllOutputArguments(AFSsub);
% Distributed calculation of each sub band
      for idx_f = 1:(length(AFS_freq_dist) - 1)
            createTask(AFSjob, @AFS_Compute_Frequencies, 1,
            {AFS_freq_dist(idx_f), AFS_freq_dist(idx_f + 1),
            AFS_freq_dist_yc{idx_f}, AFS_freq_dist_yc{idx_f
            + 1}, SysFilename, AFSinfo});
      end
% Submitting job to active queue
submit(AFSjob);
% Waiting for completion of jobs
waitForState(AFSjob, 'finished');
% Getting results from JobManager
AFS_subband_results = getAllOutputArguments(AFSjob);

% Formatting results from completed jobs.
```

```
% Destroying job on JobManager to free up resources.
destroy(AFSsub);
destroy(AFSjob);
```

### 5.3 Utilizing Out-of-Core Solvers

Usage of the parallel module is simple and selection of either AFS or non-AFS mode is done through a flag in the AFSinfo structure. Conceptually, the parallel version of SYS → SNP creates and distributes amongst the available processors of the cluster, n tasks of computing the frequency response at a particular frequency. The task of computing the frequency response at each frequency comprises of first reading the semi state space matrices from the Matlab matfile, constructing the system of linear equation, solving the system and filtering the output as shown below.

```
Function
out = compute_one_frequency(f, C, G, B, L, F)
A = j*2*pi*f*C + G      % f is in Hz
x = A\B                 % linear solver
out = Lx + F            % compute output at that frequency
return y
```

When the system state space matrices generated are huge and cannot be handled efficiently in memory, out of core (OOC) solvers and techniques must be used. Thus, instead of utilizing Matlab's default solver (mldivide), external solvers can be utilized using the Matlab MEX C & Fortran interface. The MEX application programming interface (API) contains functions to access matrices stored in Matlab .MAT file format, access and create variables in the Matlab workspace (MX Array Manipulation) and Matlab compute engine manipulation. This MEX interface allows the seamless inclusion of an external solver within the framework previously described by simply replacing x = A\B with x = MEX_External_Solver(A,B). Efficient C/C++ or Fortran code can be called as if they were 'built-in' Matlab functions. Depending on the problem to be solved, some solvers are more suitable to the problem at hand which is not available from Matlab. For example, when the matrices encountered are huge and do not fit into main memory or excessive fill in of factors is encountered, OOC solvers [22-25] need to be employed where the relevant data structures are stored on disk instead of main memory. Hence OOC solvers are more suited to be run as a batch job in the background which runs over an extended period of time. A MEX interface to the TAUCS OOC solver [24] was written as a demonstration of this concept. It was also used in the teaching of a module 'Introduction to MEX file interfaces' by the author as part of the ToK DOW commitment and used as a mini-project teaching aid for ESRs.

### 5.3 Computation Speedup

A series of benchmarks were run to test the computational speedup on the ATLAS Beowulf cluster.

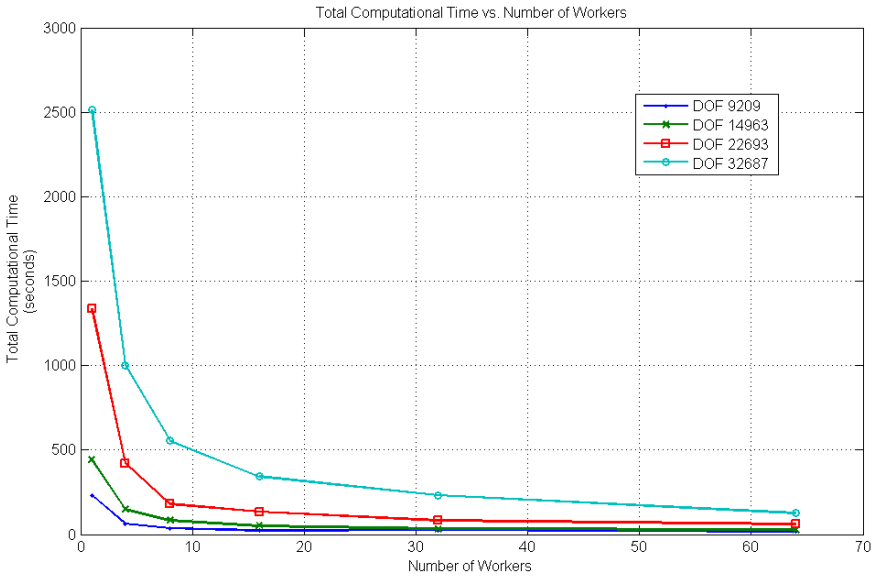



Fig. 7. Total computational time (seconds) versus number of workers used for several matrix sizes (degrees of freedom) using ComputeSYS2SNP_parallel64 with Distributed Adaptive Frequency Selection (AFS)

The computational time was recorded for test runs of different matrix sizes (DOF: degrees of freedom) of the RFPAD benchmark. **Fig. 7** illustrates the total computational time taken by ComputeSYS2SNP_parallel64 with Distributed Adaptive Frequency Selection enabled. It was observed that parallelization impressively decreases the time taken for computing a solution. For different benchmarks, the response of the system will yield different curves. Depending on the nature of the curve, different speedup values can be expected. For the RFPAD benchmark used, speedups of up to 21 X were observed. The test was run using 1, 4, 8, 16, 32 and 64 workers and the output of the parallel run was compared with the corresponding serial run. Both serial and parallel outputs were found to be equivalent and hence it confirms correctness of operation.

### 6. Conclusion

Denser integration of devices inevitably leads to greater impact of EM coupling between circuits. Because of this, the CHAMELEON-RF project was conceived as part of an initiative to address these issues [1]. The validation of the

Chamy prototype software against real world industrial benchmark structures has demonstrated that the domain decomposition approach with electromagnetic circuit elements is an effective method for managing the complexity of modeling these effects in contemporary mixed signal design. Computational speedup can be gained through parallelization. Often however, parallelization of serial code is not a straight forward case and depends on the underlying architecture and distributed programming model used. A novel parallel module was developed for the Chameleon-RF demonstration platform which yielded up to 21 X speedup compared with the original serial version.

**Acknowledgement**

The author would like to gratefully acknowledge support and insightful discussion with Prof. Daniel Ioan, Asc.Prof. Gabriela Ciuprina and the Chameleon-RF project team. The authors would also like to acknowledge the financial support offered by the European Commission under the Marie Curie Fellowship (FP6) programme. I am grateful to Mălureanu Emilia-Simona for her help. The views stated herein reflect only the author's views and the European Commission is not liable for any use that may be made of the information contained.

## R E F E R E N C E S

[1] *J. Niehof, H. Janssen, and W. Schilders*, "Comprehensive High-Accuracy Modeling of ELectromagnetic Effects in Complete Nanoscale RF blocks: CHAMELEON RF," presented at Signal Propagation on Interconnects (SPI). IEEE Workshop on, 2006.

[2] *E. Mollick*, "Establishing Moore's Law," IEEE Annals of the History of Computing, **vol. 28**, pp. 62-75, 2006.

[3] *D. Ioan, G. Ciuprina, M. Radulescu, and E. Seebacher*, "Compact modeling and fast simulation of on-chip interconnect lines," Magnetics, IEEE Transactions on, **vol. 42**, pp. 547-550, 2006.

[4] "ITRS Roadmap," in www.itrs.net.

[5] *T. McConaqhy and G. Gielen*, "Automation in Mixed-Signal Design: Challenges and Solutions in the Wake of the Nano Era," Computer-Aided Design, International Conference on, **vol. 23**, pp. 1092-1152, 2006.

[6] *G. Q. Zhang, M. Graef, and F. van Roosmalen*, "Strategic Research Agenda of "More than Moore"," presented at Thermal, Mechanical and Multiphysics Simulation and Experiments in Micro-Electronics and Micro-Systems (EuroSime 2006). 7th International Conference on, 2006.

[7] *D. Leenaerts, G. Gielen, and R. Rutenbar*, "CAD Solutions and Outstanding Challenges for Mixed-Signal and RF IC Design," Computer-Aided Design, International Conference on, **vol. 10**, pp. 744-984, 2001.

[8] J. *Niehof, H. H. J. Janssen, W. H. A. Schilders, W. Schoenmaker, D. Ioan, G. Ciuprina, and W. Pflanzl*, "Domain Decomposition via Electromagnetic Hooks for the Modeling of Complete RF blocks," presented at Signal Propagation on Interconnects, 2008. SPI 2008. 12th IEEE Workshop on, 2008.

[9] *D. Ioan, G. Ciuprina, and M. Radulescu*, "Theorems of parameter variations applied for the extraction of compact models of on-chip passive structures.," presented at Signals, Circuits and Systems (ISSCS 2005). International Symposium on, 2005.

[10] *D. Ioan, G. Ciuprina, and D. Mihalache,* "Reduced Order Electromagnetic Models for On-Chip Passives Based on Dual Finite Integrals Technique," Scientific Computing in Electrical Engineering (SCEE), Mathematics in Industry, vol. ISBN 978-3-540-71979-3, pp. 287-294, 2006.

[11] *D. Ioan, G. Ciuprina, M. Radulescu, and M. Piper*, "All levels models strategy to reduce the model order of on-chip passive components," IEEE Conference on Electromagnetic Field Computation CEFC 2004 vol. Digest Book, pp. 345, 2004.

[12] *D. Ioan, G. Ciuprina, and M. Radulescu*, "Absorbing boundary conditions for compact modelling of on-chip passive structures," COMPEL, **vol. 25**, pp. 652, 2006.

[13] *L. Yunn-Shiuan, C. Shiang-Woei, and C. Jeng-Tzong*, "FEM versus BEM," Circuits and Devices Magazine, IEEE, vol. 20, pp. 25-34, 2004.

[14] *G. Ciuprina, D. Ioan, D. Niculae, J. Villena, and L. Silveira*, "Parametric Models Based on Sensitivity Analysis for Passive Components," in Intelligent Computer Techniques in Applied Electromagnetics, 2008, pp. 231-239.

[15] *D. Ioan, G. Ciuprina, and L. M. Silveira*, "Effective Domain Partitioning With Electric and Magnetic Hooks," Magnetics, IEEE Transactions on, **vol. 45**, pp. 1328-1331, 2009.

[16] *M. Clemens*, "Large systems of equations in a discrete electromagnetism: formulations and numerical algorithms," Science, Measurement and Technology, IEE Proceedings -, **vol. 152**, pp. 50-72, 2005.

[17] *J.M. Dohlus, P. Hahne, X. Du, B. Wagner, T. Weiland, and S.G. Wipf*, "Using the Maxwell grid equations to solve large problems," Magnetics, IEEE Transactions on, vol. 29, pp. 1914-1917, 1993.

[18] *T. Weiland*, "A Discretization Method for the Solution of Maxwell's Equations for Six-Component Fields," Electronics and Communication., **vol. 31**, pp. 116, 1977.

[19] *K.L. Shlager and J.B. Schneider*, "A selective survey of the finite-difference time-domain literature," Antennas and Propagation Magazine, IEEE, **vol. 37**, pp. 39-57, 1995.

[20] *Abd-El-Raouf and H.E.*, "A Class of Finite Difference Time Domain (FDTD) Techniques for Solving Large Electromagnetic Structures," presented at RF and Microwave Conference, 2006. RFM 2006. International, 2006.

[21] *I.M.D. Ioan*, "Missing Link Rediscovered: The Electromagnetic Circuit Element Concept," JSAEM Studies in Applied Electromagnetics and Mechanics, **vol. 8**, pp. pp. 302-320, 1999.

[22] *M. Omer, I. Dror, and T. Sivan*, "An out-of-core sparse symmetric-indefinite factorization method," ACM Trans. Math. Softw., **vol. 32**, pp. 445-471, 2006.

[23] *J.R. Gilbert and S. Toledo*, "High-Performance out-of-core sparse LU factorization," Proceedings of the 9th SIAM Conference on Parallel Processing for Scientific Computing San Antonio, Texas., 1999.

[24] *S. Toledo, D. Chen, and V. Rotkin*, "TAUCS: A Library of Sparse Linear Solvers Version 2.2. ," Tel-Aviv Univ., http://www.tau.ac.il/stoledotaucs/, 2003.

[25] *E. Agullo, A. Guermouche, J.-Y. L'Excellent, and et al*., "Towards a Parallel Out-of-core Multifrontal Solver: Preliminary Study," 2007.