Finite Element Emulation-based Solver for Electromagnetic Computations

M. Tarek Ibn Ziad*, Mohamed Hossam*, Mohamad A. Masoud*, Mohamed Nagy*, Hesham A. Adel*, Yousra Alkabani*, M. Watheq El-Kharashi*, Khaled Salah**, Mohamed AbdelSalam**

* Computer and Systems Engineering Department, Ain Shams University, Cairo, Egypt

** Mentor Graphics, Cairo, Egypt

International Symposium on Circuits and Systems (ISCAS)

24-27 May 2015, Lisbon, Portugal

Motivation

- Electromagnetic (EM) computations are the cornerstone in the design process of several real-world applications.
- These computations require solving millions of simultaneous equations.
- Software-based solvers do not scale well as the number of equations-to-solve increases.
- FPGAs are limited by memory and area constraints.
- Emulation technology provides a solution to the memory and area constraints encountered by FPGAs.
- A scalable design is introduced to accelerate the finite element solver of an EM simulator on a hardware emulation platform.

Mentor Graphics Veloce Emulator

Background

A. Finite Element Method (FEM)

FEM is a numerical method, which is used to solve boundary-value problems defined by a partial differential equation (PDE) and a set of boundary conditions.

Discretization of the solution domain

↓

Calculation of elemental matrices

↓

Assembly of a global matrix

↓

Solution of the system equations

↓

Post-processing of the results

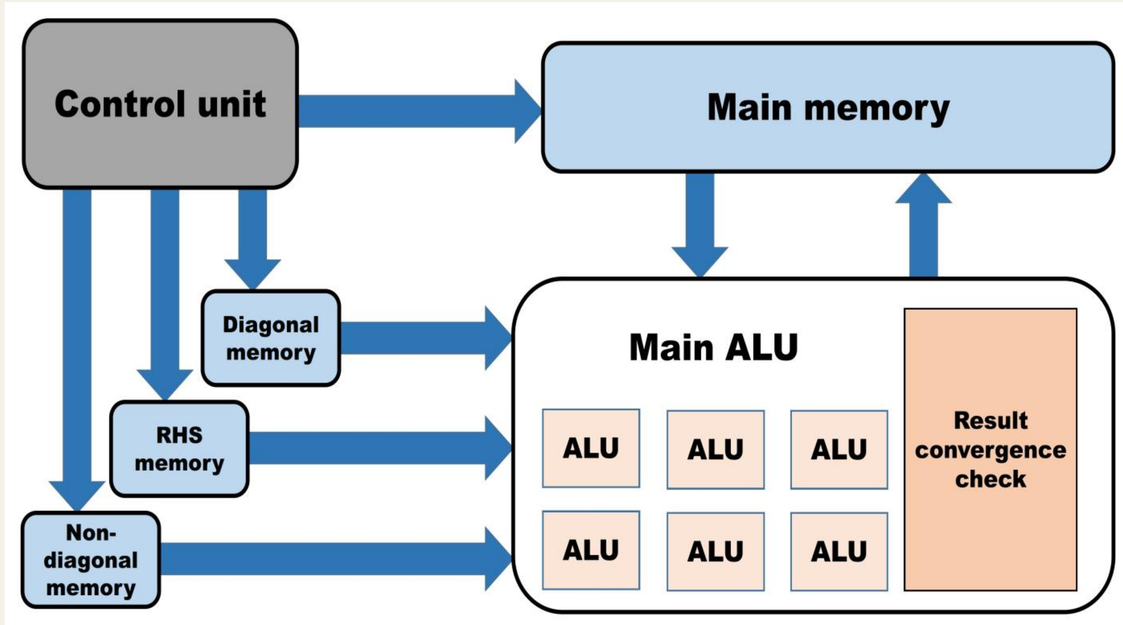
B. Jacobi Iterative Method

Jacobi is an Iterative method used to solve the system of linear equations generated from the FEM.

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, i\neq j}^{n} a_{ij}x_j^{(k)}\right)$$

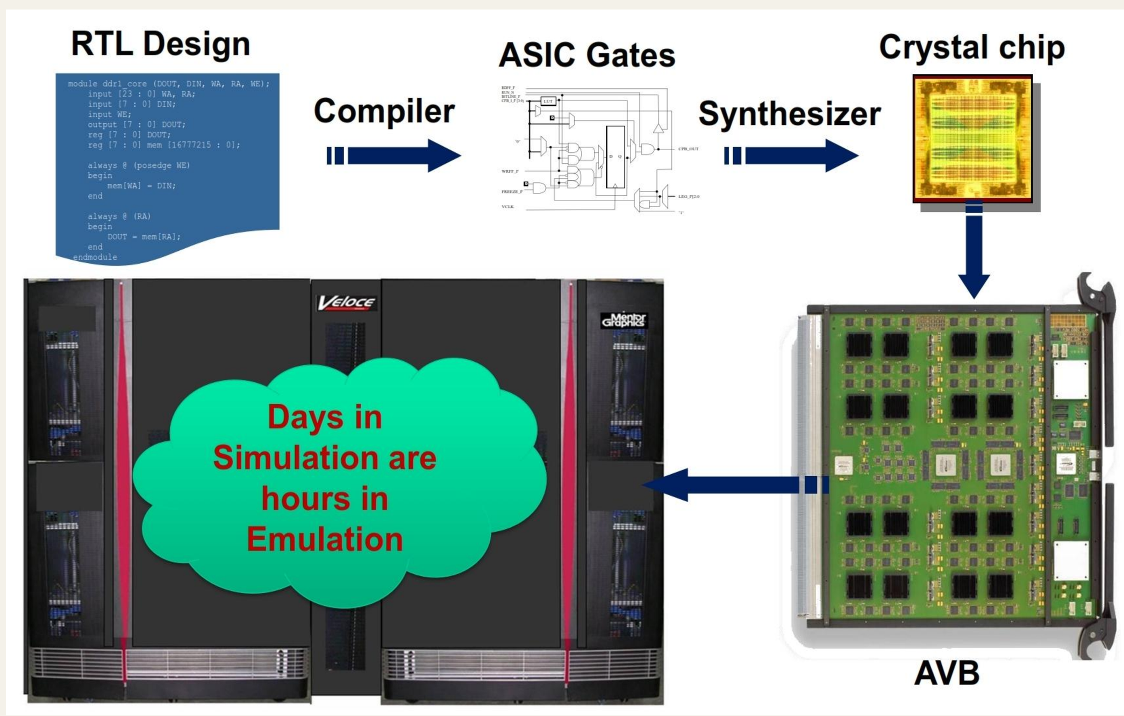
Hardware Implementation

- Our design splits the coefficient matrix into clusters, where each cluster contains a number of rows (equations).
- Each cluster is independent of the others, so all of them can be operated on in parallel using simple ALUs.
- Four memories are used to store 32-bit floating-point elements of the coefficient matrix, right hand side (RHS) vector, and solution vector.
- The design is fully pipelined. It could be configured with a pre-defined tolerance to control the accuracy of the final solution.



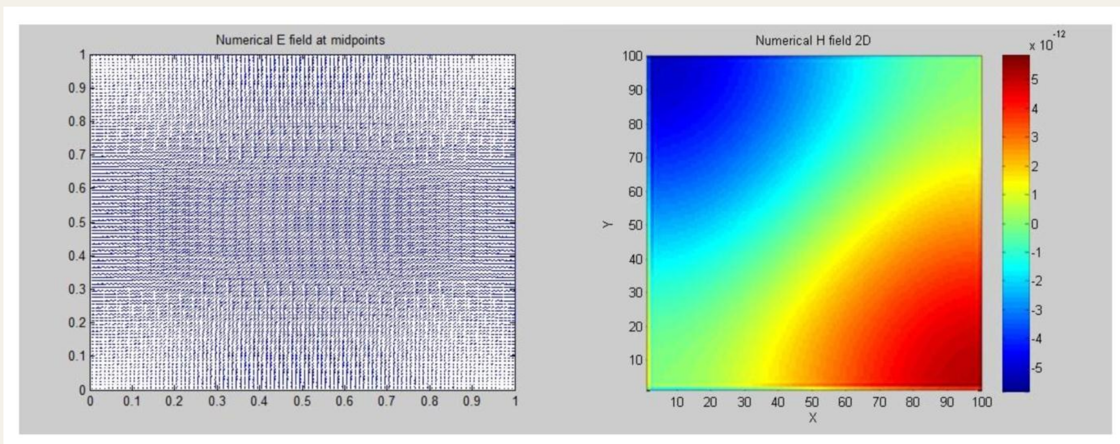
Experimental Setup

Our architecture was modeled using Verilog. Then, the design was compiled and run on a physical hardware emulation platform with 8 advanced verification boards (AVBs). That platform provides a total capacity of 128 crystal chips (64 M logic gates) with 4 GB of memory.



Electromagnetic (EM) Solver

We demonstrated the efficiency of our proposed solution by implementing a two-dimensional (2D) edge element code for solving Maxwell's equations for metamaterials using FEM. The outputs are the numerical Electric and Magnetic field graphs.



The code consists of three parts; pre-processing, solver, and post-processing. Pre-processing and post-processing calculations are performed on MATLAB while the solver part, which consumes most of time, is accelerated using the hardware emulation platform.

Experimental Results

A. Resource Utilization

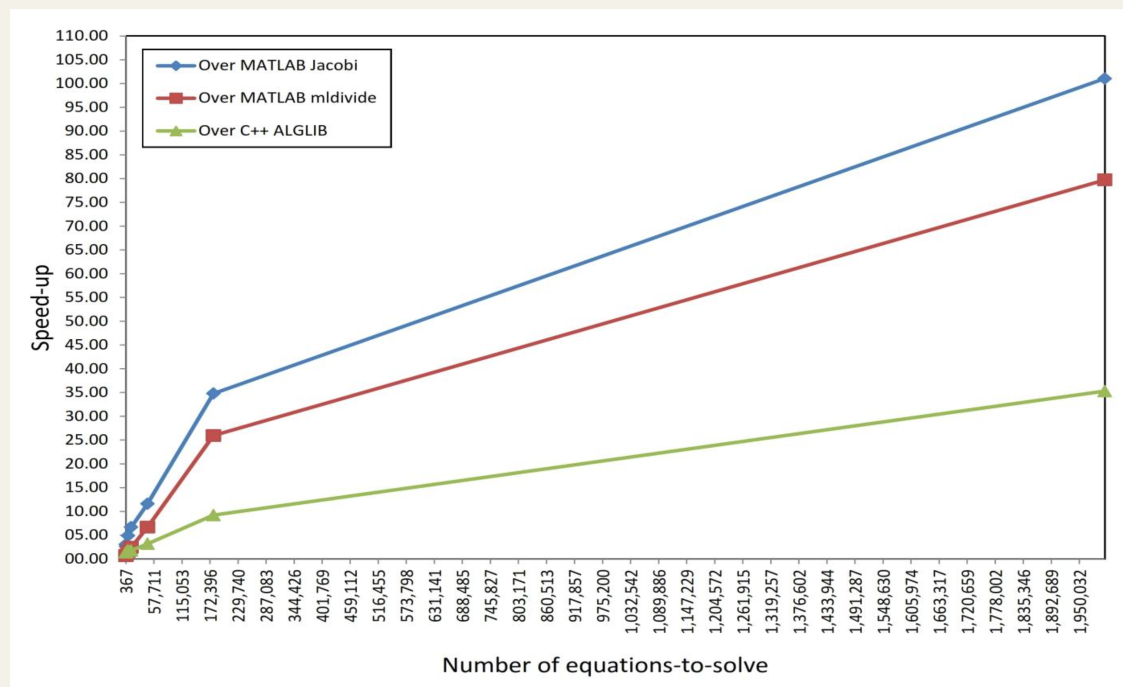
The given Table evaluates our 32-bit floating-point Jacobi design using different test cases.

	Our Jacobi Design Test Cases				
Number of equations	420	11,100	44,700	179,400	2,002,000
Number of ALUs	14	74	149	299	1,000
Frequency (KHz)	1666.7	1754.4	1587.3	1754.4	1333.3
Number of iterations	21	22	22	22	22
Number of flip-flops	18,752	85,490	168,896	335,702	1,115,220
Number of LUTs	106,401	535,566	1,071,976	2,144,796	7,158,357
Memory bytes	8,704	376,832	1,521,664	6,115,328	40,943,616
Number of FPGAs	2	8	15	29	98

B. Speed-up

The design speed-up is evaluated against three different software-based solvers on a 2.00 GHz Core i7-2630QM CPU. The first solver is a standard Jacobi iterative implementation on MATLAB. The second is the MATLAB special operator for solving systems of equations, mldivide. Finally, the third solver is an iterative solver from ALGLIB, an open-source numerical analysis library.

No. of equations	Our Jacobi	MATLAB Jacobi		MATLAB mldivide		C++ ALGLIB	
	Time (S)	Time (S)	Speed-up	Time (S)	Speed-up	Time(S)	Speed-up
420	0.0003	0.0009	3.00	0.0002	0.66	0.0004	1.33
1,200	0.0007	0.0018	2.57	0.0005	0.71	0.0010	1.43
4,900	0.0011	0.0054	4.90	0.0017	1.55	0.0020	1.82
11,100	0.0017	0.0113	6.65	0.0041	2.41	0.0030	1.76
44,700	0.0038	0.0440	11.58	0.0253	6.66	0.0120	3.16
179,400	0.0062	0.2157	34.79	0.1608	25.94	0.0570	9.19
2,002,000	0.0240	2.4252	101.05	1.9138	79.74	0.8470	35.29



Conclusion

- Our optimized Jacobi design has achieved a remarkable improvement in matrix calculations run-time over the software-based solvers for an EM problem of solving Maxwell's equations in a 2D metamaterial edge element using FEM.
- Higher speed-up could be obtained upon using more emulator area as the design is fully parallelized.

Future Work

- Discussing more complex designs to evaluate the performance of the emulation-based implementation against large scale problems.
- Implementing hardware solutions for Maxwell's equations using methods other than FEM.

Poster template by ResearchPosters.co.za