A FEASIBILITY STUDY ABOUT A CUSTOM HARDWARE IMPLEMENTATION OF THE FDTD ALGORITHM

L. Verducci [(1)], P. Placidi [(1)], G. Matrella [(2)], L. Roselli [(1)], F. Alimenti [(1)], P. Ciampolini [(2)], A. Scorzoni [(1)]

[(1)] DIEI, University of Perugia, Via G. Duranti, 93, 06125 Perugia (Italy)
E-mail: verducci@diei.unipg.it

[(2)] DII, University of Parma, PARCO AREA delle SCIENZE 181/A, 43100 Parma (Italy)

ABSTRACT

A number of inherent characteristics makes the Finite Difference Time Domain (FDTD) algorithm almost ideal for the analysis of a wide class of microwave and high-frequency circuits. In this paper, the architecture of a digital system dedicated to the fast execution of this algorithm is presented. Such system is conceived as a module communicating with a host personal computer via a PCI bus, and is based on a VLSI ASIC, which implements the "field-update" engine. The system structure is defined by means of a hardware description language, allowing for keeping high-level system specification independent of the actual fabrication technology. Simulations show that significant speed-up can be achieved, with respect to state-of-the-art software implementations of the same algorithm.

INTRODUCTION

In recent years, the FDTD method has become one of the most widely used computational techniques for the full-wave analysis of electromagnetic phenomena [1,2,3,4]. However, its application to practical three-dimensional problems, is often limited by the demand of very large computational resources. Thanks to microelectronic technologies, inexpensive RAMs and high-speed processors have become available and practical engineering application of the FDTD method have been introduced. However, the actual performance which can be achieved by means of high-level programming is very much dependent on the operating system, the language compiler and the program structure [5]. On the other hand, the FDTD algorithm, taking advantage of the simplicity and symmetry of discretized Maxwell's equations, exhibits some features which make an hardware implementation appealing. Maxwell's equations in an isotropic medium can be written as follows:

$$\begin{aligned} &\frac{\partial \vec{B}}{\partial t} + \nabla \times \vec{E} = 0 \\ &\frac{\partial \vec{D}}{\partial t} - \nabla \times \vec{H} = -\vec{J} \\ &\vec{B} = \mu \vec{H} \\ &\vec{D} = \varepsilon \vec{E} \end{aligned} \tag{1}$$

where the symbols have their usual meaning, and $\vec{J}$, μ, and ε, are known functions of space and time.
According to the FDTD discretization method, the propagation domain is divided into cells, each cell being independently characterized in terms of material properties. Electromagnetic field components are mapped at cell edges and eqs. (1) are discretized, in both time and space, at each cell [1]. Referring to a given magnetic-field component (for instance H_x) and following the standard Yee's notation the update equation reads:

$$H_x\Big|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = H_x\Big|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n-\frac{1}{2}} + c_1\left(E_z\Big|_{i,j+1,k+\frac{1}{2}}^{n} - E_z\Big|_{i,j,k+\frac{1}{2}}^{n}\right) + c_2\left(E_y\Big|_{i,j+\frac{1}{2},k}^{n} - E_y\Big|_{i,j+\frac{1}{2},k+1}^{n}\right) \tag{2a}$$

whereas the update equation for an electric field component (for instance E_x) becomes:

$$E_x\Big|_{i+\frac{1}{2},j,k}^{n+1} = E_x\Big|_{i+\frac{1}{2},j,k}^{n} + c_3\left(H_z\Big|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z\Big|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}\right) + c_4\left(H_y\Big|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} - H_y\Big|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}}\right) + H_z\Big|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \quad (2b)$$

In eqs. (2a) and (2b), the coefficients c_1, c_2 ,c_3 and c_4 are kept constant throughout the simulation: their value depends only on the actual material, on the discretization mesh-size and on the time step adopted [1]. Basically, such a scheme involves the evaluation of time derivatives of electric and magnetic-field components at alternate time intervals, denoted by integer and fractional superscripts, respectively, in (2a) and (2b). According to such a scheme, field updates depend only on quantities computed at previous time iterations, so that update equations pertaining to a given time step can be decoupled and independently solved (i.e., sequentially, in any arbitrary order). Then six scalar equations per cell are eventually obtained, which share the same structure:

$$y = a + k_1(b - c) + k_2(d - e) + j_s \quad (3)$$

In the above equation, y represents the field component to be updated, a is its current value (i.e., that computed at the previous time step), k_1 and k_2 are material-related constant coefficients, b, c, d and e are known field components (i.e., previously computed at neighboring grid points) and j_s represents the field source, if any.

Moreover, due to the intrinsic nature of a finite-difference algorithm, updating a field component at a given location only involves a limited number of variables, located at nearest-neighboring positions. By using suitable numbering techniques, simple and regular coefficient patterns can be obtained, so that the data-communication bandwidth can be kept under control. Vector and massively-parallel computers can thus be exploited to achieve fast implementation of the FDTD algorithm [5]. The basic drawback of such an approach consists of the huge hardware costs, which limits the development and diffusion of supercomputing facilities.

The same characteristics recalled above, however, can be exploited to implement an efficient (yet scalar), custom processor, at a small fraction of the cost of a general-purpose supercomputer. In this paper, we describe a custom VLSI chip implementing the "field-update"' engine of the FDTD algorithm on a 3D domain. This system is conceived as a module communicating with a host personal computer via a PCI bus.

SYSTEM ARCHITECTURE

The system architecture has been chosen in order to achieve three main goals: optimization of the Floating Point Unit (FPU) pipeline, good trade-off between parallelism in the structure and system input/output (I/O) pin count and efficient communication with the PC host. The main functional blocks of this architecture are: a dedicated FPU, some registers banks ($CACHE_X$, $CACHE_Y$, $CACHE_Z$ and $CACHE_C$), a set of control units (*CONTROL UNIT*$_A$, *CONTROL UNIT*$_B$, CTR_X, CTR_Y, CTR_Z, CTR_C, CTR_V and CTR_P), a PCI Target Interface Circuit (*PCI INTERFACE*) and several external SDRAM memories (Fig.1).

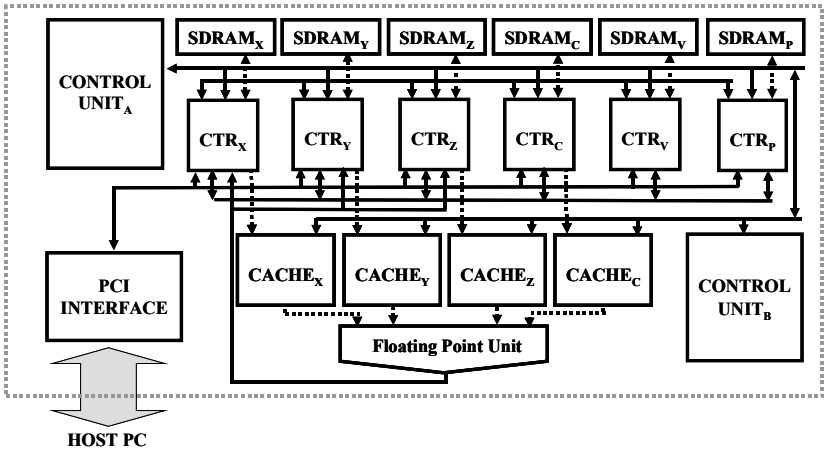

Fig. 1. System architecture

The functionality of the blocks has been described in detail in [6] and therefore in this paper we will focus on the data storage organization.
To store and to manage the data required to solve the FDTD algorithm several external SDRAM banks have been used. This choice improves the system modularity: in fact it is possible to consider a bigger amount of data (which enables the system to manage a bigger mesh-size) just using bigger SDRAM and introducing very small modifications in the system pin I/O count. In addition an accurate use of the SDRAM chips allow us to improve the performance of the FPU pipeline (Fig.2). In fact in order to avoid the stalls of the pipelined FPU, for every clock cycle useful data should be accessed from the SDRAM. This goal can be achieved by exploiting streaming mode ("burst") access of the SDRAM but requires an optimised data storage in the memories itself. By fixing two out of three coordinates in the space, a straight row of cells is selected: for each cell in the row, and due to the translational symmetry of the FDTD mesh, field coefficients appearing in homologous position within eq. (3) refer to aligned mesh cell as well (Fig.2). Therefore it is convenient to store field components relative to aligned cells into the same memory row (i.e.: $H_y\big|_{i,j-1,k}^{n+\frac{1}{2}}, H_y\big|_{i,j,k}^{n+\frac{1}{2}}, H_y\big|_{i,j+1,k}^{n+\frac{1}{2}}$).
Referring to Fig.2 we should notice that to update the H_y (i, j, k) component, four other electromagnetic field components are required: two component for the E_x and two components for the E_z fields which share the same y coordinates. In the same way all the components required to update each of the H_y components share the same y coordinate. Therefore we can use a very long valid data burst using full page read and write SDRAM burst operations.
A fully optimized memory organization, i.e. yielding a valid data in every clock cycle, should require infinite burst operations and so a very big amount of SDRAM busses would be necessary, making the I/O count not manageable. A satisfying trade-off of these two requirements was obtained using six SDRAM busses: one for each electromagnetic field space coordinates (x, y, z), one for material related coefficients, one for memory pointers and one for field source points values.
With this data organization, to update a field component burst, two field component bursts must be read from other SDRAM chips and therefore a temporary storage of data read in the first burst-read operation is required ($CACHE_X$, $CACHE_Y$, $CACHE_Z$ and $CACHE_C$). The second burst read from SDRAM will be directly provided at the input of the FPU and then the updated burst values (the output of the FPU) will be written in the same SDRAM of the data coming from the burst to be updated.

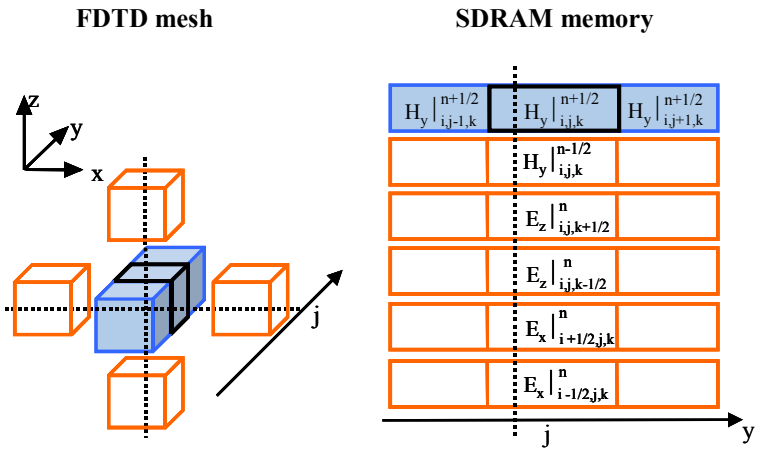

Fig.2. Storage of the field components into SDRAM memories.

PERFORMANCE ESTIMATION

A virtual implementation of the system has been carried out by specifying architectures through an hardware description language, and an estimate of the performance has been obtained by means of simulation [7]. To evaluate the performance of the proposed architecture, and to allow for comparison with software implementation on a general purpose architecture, a fairly simple case has been considered and a cubic resonant cavity has been simulated. After a pulsed stimulus is injected at a given location (accounting for a non-zero initial field) the transient simulation is carried out. Here we have limited our investigation to check for consistency between the “hardware” and the “software” simulation results by direct inspection. Such a check has been completely successful: negligible differences were found between the output vectors of the two solutions. These small differences are due to the different impact of round off, being the sequence of arithmetic operations not necessarily the same in the hardware and software implementation. Performance has been estimated by looking at the time required to compute a given amount of time iterations. To compute 10,000 time steps on a structure of 21×21×21 cells, an estimate of 15 seconds has been obtained for the "hardware" solver, whereas 64 seconds were needed by the “software” one. The latter figure was obtained by using a highly-optimized FORTRAN code, running on a 500 MHz Pentium-class machine, equipped with 128 Mbyte RAM (sufficient to avoid virtual-memory page swapping, given the size of the problem at hand) under Linux operating system. Such a rough estimate therefore indicates that a significant speed-up factor seems achievable.

CONCLUSIONS

In this paper, a hardware architecture for the solution of the FDTD algorithm on a 3D domain has been presented. This architecture is based on a custom VLSI chip implementing the “field-update” engine, and is conceived as a module communicating with a host personal computer via a PCI bus. A virtual implementation of the system has been carried out by means of a VHDL description of its architecture, and the performance estimate has been extracted from simulation. From these results it emerges that, even at relatively low system clock frequencies, the time required for the solution of FDTD equations positively compares with a software implementation on general-purpose microprocessors.

REFERENCES

[1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equation in isotropic media", *IEEE Trans. Antennas Propagat.*, vol. 14, pp.302-307, 1966.

[2] S. D. Gedney, "Efficient Implementation of the FDTD Algorithm on High Performance Computers", Time-Domain Methods For Microwave Structures, *IEEE Press*, pp.313-322, 1997.

[3] M. Picket-May, A. Taflove, and J. Baron, "FD-TD modeling of digital signal propagation in 3-D circuits with passive and active loads", *IEEE Trans. on Microwave Theory and Techniques*, vol.42, pp.1514-1523, 1994.

[4] P. Ciampolini, P. Mezzanotte, L. Roselli, and R. Sorrentino, "Efficient simulation of high-speed digital circuits using time adaptive FD-TD technique", *25th European Microwave Conf.*, (Bologna, IT), pp.636-640, sep 1995.

[5] S. D. Gedney, "Finite-Difference Time-Domain Analysis of Microwave Circuit Device on High Performance Vector/Parallel Computers", *Time-Domain Methods For Microwave Structures*, IEEE Press, pp.344-344, 1997.

[6] P. Placidi, L. Verducci, G. Matrella, L. Roselli, P. Ciampolini, "A custom VLSI architecture for the solution of FDTD equations", *Special Issue of IEICE Transactions*, March 2002.

[7] "IEEE Standard VHDL Reference Manual", *IEEE Standard 1076*, 1987.