FPGA-Based Real-Time Digital Simulation

Mahmoud Matar, *Student member, IEEE*, Mohamed Abdel-Rahman, *member, IEEE,* Abdel-Mohaimen. Soliman

Abstract – This paper is an effort to develop an FPGA-based real-time digital simulator. FPGA is used instead of parallel processors to simulate electromagnetic transients in power systems. A sample single-phase test systems was used to examine the principle. The mathematical model for the power system under consideration is implemented and simulated using Xilinx ISE 4.2 software and ModelSim simulator.

Keywords: Power system modeling, Real-time digital simulation, Field Programmable Gate Array (FPGA).

I. INTRODUCTION

The use of simulation programs have become largely accepted in simulating electromagnetic transients. However, one of the main disadvantages of these programs is that they operate in non-real-time. That is, the CPU time required to compute the modeled system's response might take several seconds or minutes, orders of magnitude longer than the actual response. This non-real-time operation precludes the interfacing of external equipment to the simulator and therefore limits the application of such simulators for the testing of physical control and protection devices. Real-time simulators are the only choice when it is required to perform a closed loop test on a certain device as prudent utility practice forbids experimenting with the actual systems. To close the loop, it is necessary that the simulator is able to accept input information from the device under test and to incorporate this information into the ongoing simulation run. The system's equations need to be solved fast enough, so that the outputs are available before the arrival of the next sample, or mathematically

$$t_c \le \Delta t$$

Where t_c is the computation time Δt and is the simulation time step

M. Matar is with Department of Electrical Engineering, Ain Shams University, Abbassia, Cairo, Egypt. (e-mail: mahmatar@ieee.org).
M. Abdel-Rahman is with Department of Electrical Engineering, Ain Shams University, Abbassia, Cairo, Egypt. (e-mail: m_abdel_rahmanca@yahoo.com).
A. Soliman is with Department of Electrical Engineering, Ain Shams University, Abbassia, Cairo, Egypt. (e-mail: abdelmohymen@hotmail.com).

Presented at the International Conference on Power Systems Transients (IPST'05) in Montreal, Canada on June 19-23, 2005

Paper No. IPST05 - 149

The hardware of a real-time digital simulator is usually based on computers with parallel processors [1,2]. Each processor may represent a different section in the power system under study and a master processor is used to control the communication between the other processors.

Field Programmable Gate Arrays (FPGAs) can be used instead of parallel processors as FPGAs provide a cheap alternative for parallel processors. The required hardware to solve the power system equations can be implemented on an FPGA chip and be processed in series, parallel or a combination of them as required.

Section II compares between FPGAs and Digital signal processors (DSP). Section III describes the process of FPGA implementation. Section IV is an implementation example and results. Section V is the conclusions.

II. Choosing between FPGAs and DSPs

For the purpose of real time digital simulation, we are looking for a programmable device capable of solving all the power system equations in a time less than the sampling period. i.e., we need a programmable device that can handle heavy mathematical operations in a very short time. This fact narrows our search for two alternatives:

1. Digital Signal Processor (DSP)
2. Field Programmable Gate Array (FPGA).

DSP processors use special architectures to accelerate repetitive, numerically intensive calculations. The most important feature that is especially useful in power system simulation is the ability to perform one or more multiplication operations in a single instruction cycle. In addition, DSP processors often contain multiple-access memory architectures that allow the processor to simultaneously load multiple operands, such as a data sample and a coefficient, in parallel with loading an instruction. In addition, DSP processors often include a variety of special memory addressing modes and program-flow control features designed to accelerate the execution of repetitive operations. Lastly, most DSP processors include specialized on-chip peripherals or I/O interfaces that allow the processor to efficiently interface with other system components, such as analog-to-digital converters and host processors [3, 4].

However, while the DSP processor may perform multiple instructions per clock cycle, the overall process is performed in a three-step series of

1. Memory-read

2. Process
3. Memory-write

This process is adequate for independent sequential algorithms. However, the DSP processor becomes less efficient when an algorithm is dependent on two or more of the past and/or present state conditions. The algorithm developed by DOMMEL to simulate the power system represents each of the power system components with a set of difference equations that is of recursive nature, i.e. the present value depends on the past values [5].

FPGA is an integrated circuit that contains many identical logic cells that can be viewed as standard components. The individual cells are interconnected by a matrix of wires and programmable switches. A user's design is implemented by specifying the simple logic function for each cell and selectively closing the switches in the interconnect matrix. Complex designs are created by combining these basic blocks to create the desired circuit.

FPGA, unlike a DSP, is a parallel device and its hardware is custom configured for the job at hand. FPGA have two main advantages over DSP chips when it comes to mathematical operations:

- They allow many different operations to be implemented on the same device meaning that the answers can be evaluated simultaneously.
- Buses are required to effectively carry data. Most DSP chips have several buses but an FPGA can contain hundreds, if not thousands of them. Furthermore, the width of each bus can be exactly the number of bits needed; i.e., precision is not lost and no extra memory is consumed. This makes FPGAs very cost effective when multiple channels of data must be processed.

Although a DSP is a specialized mathematical processor, however, an FPGA can outperform a DSP by many orders of magnitude as long as data is processed in an algebraically unvarying manner. This is because the FPGA can be configured in parallel giving it an advantage over a DSP, which is a sequential device like a microprocessor. Even though some modern DSPs are adding parallel units, most DSPs can handle only up to two channels of data.

To sum up, math is fastest when implemented on FPGAs; since the algorithm and technology can be optimized to the flow of data. Thus, implementation of the network equations on an FPGA chip and taking advantage of its parallel structures results in a very short execution time [6].

For the previously stated reasons, we have chosen to implement the power system equations on a single FPGA chip.

III. FPGA IMPLEMENTATION

Separate units will be implemented on the FPGA chip, each one of these units will be dedicated for the evaluation of a single equation. The proposed method for FPGA implementation is based on the generalized algorithm developed by Dommel for accurate time domain simulation of transients in networks containing both lumped and distributed parameters elements. In this algorithm each of the different elements of the network is represented by equivalent resistances and current sources standing for the previous states of the element.

The inductance is represented by the equivalent circuit shown in Figure.1 [5]

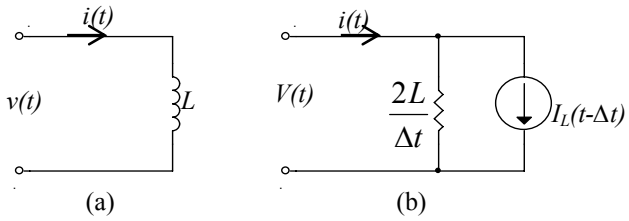

Figure 1
(a) Inductance (b) Equivalent Model

The value of the current source is updated in every simulation step by the equation

$$I_L(t) = \frac{2v(t)}{(2L/\Delta t)} + I_L(t-\Delta t)$$

The capacitance is represented by the equivalent circuit shown in Figure.2 [5]

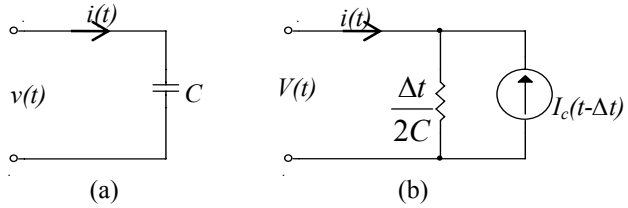

Figure 2
(a) Capacitance (b) Equivalent Model

The value of the current source is updated in every simulation step by the equation

$$I_c(t) = \frac{2v(t)}{(\Delta t/2C)} - I_c(t-\Delta t)$$

The lossless transmission line is represented by the equivalent circuit shown in Figure 3 [5]

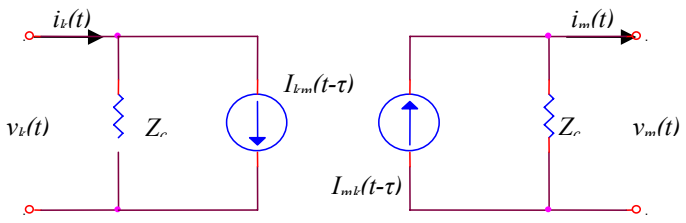

Figure 3
Lossless Transmission line equivalent

The values of the current sources are updated in every simulation step by the equations

$$I_{km}(t) = I_{mk}(t-\tau) - \frac{2}{Z_c} v_m(t)$$

$$I_{mk}(t) = I_{km}(t-\tau) + \frac{2}{Z_c} v_k(t)$$

The time domain representation of a practical transmission line is very similar to that of the lossless transmission line, except that the characteristic impedance is represented by a rational function and the traveling waves are convoluted with the shaping function. Figure 4

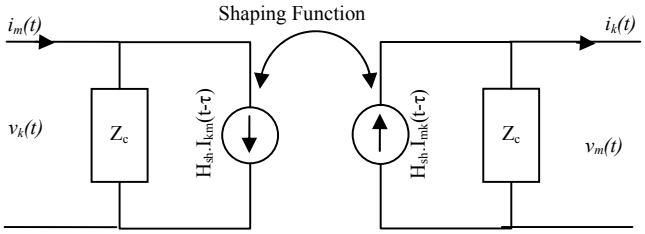

Figure 4
Practical Transmission line equivalent

The values of the current sources are updated in every simulation step by the equations

$$I_{km}(t) = H_{sh}.I_{mk}(t-\tau) - \frac{2}{Z_c} v_m(t)$$

$$I_{mk}(t) = H_{sh} I_{km}(t-\tau) + \frac{2}{Z_c} v_k(t)$$

As can be seen, the equations for updating the values of the current sources for any element comprise mainly multiplication and addition operations. Therefore; we can implement on the FPGA a number of modules, each containing hardware multipliers and adders. Each one of these units is responsible for calculating the new value of each of the current sources.

The node voltages at all nodes of the power system are calculated by the equation

$$[Y].[v(t)] = [i(t)] - [I]$$

where [*Y*] is the nodal conductance matrix, [*v(t)*] is the column vector of the node voltages, [*i(t)*] is the column vector of current sources and [*I*] is the column vector of history current sources.

Therefore; we can implement on the FPGA a number of modules, each containing hardware multipliers and adders. The number of the voltage calculating modules is equal to the number of nodes of the system under study.

After the completion of the implementation phase, the following units should exist on the chip:

- N modules dedicated for calculating the voltage at different nodes of the system.
- M modules dedicated for updating the values of equivalent current sources representing the history states of the system.

During each clock cycle, the previous states of the system, which is represented by the current sources, are used to calculate the node voltages at different nodes, and then the new values obtained for the voltages is used to update the values of the current sources. To perform these dependant operations in one clock cycle, the calculations are split into two stages, the first stage is triggered by the rising clock edge and the second is triggered by the falling clock edge. Figure 5

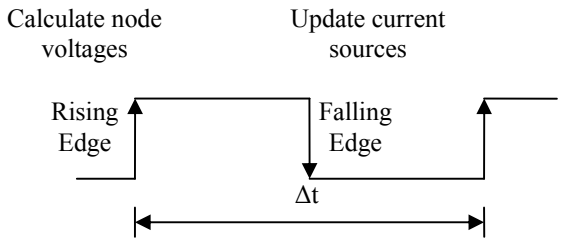

Figure 5
Unit triggering timing

Each of the implemented units is equipped with an edge detecting unit that triggers the unit when the proper edge is detected such that

- The rising clock edge triggers the voltage-calculating modules.
- The falling clock edge triggers the Current sources updating modules.

If the network contains switches, the current calculating modules of the elements connected to switches are equipped with a module enable/disable function to disable the module if the switch is opened, also implemented in the voltage calculating modules a function to enforce the admittance of the disconnected element to zero.

To reduce the resources required, the power system under study can be divided into two zones, a study zone and

an external network. The study zone is that section of the network that is significantly affected and has the primary impact on the behavior of the phenomenon under consideration. The study zone usually contains a relatively small section of the power system. The study zone must be represented in details. The remaining part of the power system is replaced by an equivalent. This remainder is called the external equivalent. The external equivalent is obtained based on the frequency response of the external network, as any network model that matches the frequency characteristics of a given network will duplicate its transient response.

The admittance frequency response of the external network is approximated by rational functions [7]. Based on the bilinear transformation, $s = \frac{2}{\Delta t}\left[\frac{z-1}{z+1}\right]$ the rational function approximating the admittance frequency response can be directly simulated in the time domain [8].

The model of the power system under study is described using the VHDL (VHSIC Hardware Description Language). VHDL is a high level programming language that has been specially designed for describing the behavior of digital electronic systems. It allows the translation of the design into actual working hardware [9, 10].

IV. Implementation Example

This paper contains two single phase case studies to verify the validity of the approach. The first case study shows the voltage at the receiving end of a single phase 300 Km transmission line terminated by an inductive load when energized with a 100 V d.c. Figure 6,7.

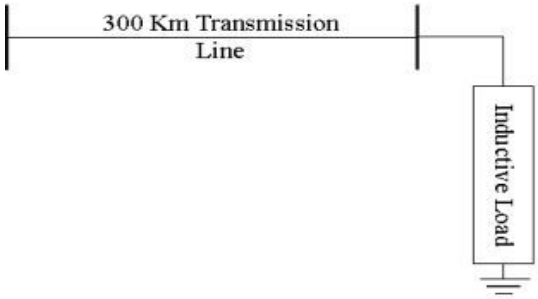

Figure 6
Test system for the first case study

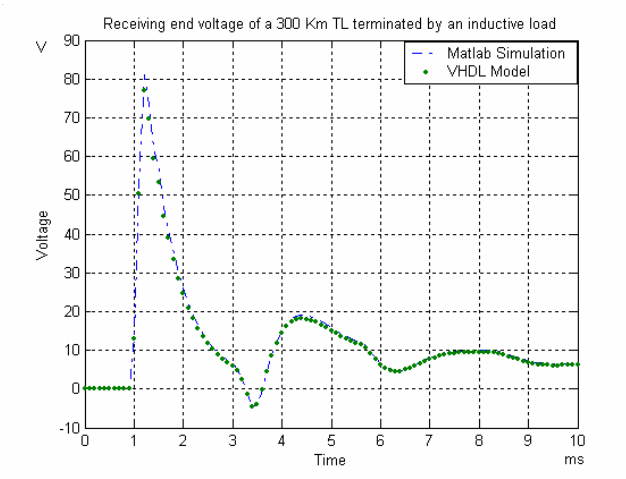

Figure 7
Comparison of the receiving end voltage of the test system and the VHDL simulation when energized with 100 V DC source

The second case shows the implementation of a network equivalent on an FPGA chip. The original system is composed of six transmission lines and six loads. Figure 8.

The parameters of the system are given in table I. We will perform two energization scenarios and compare the results obtained from the original system and that obtained from the network equivalent. The equivalent is implemented and simulated using Xilinx Foundation ISE 4.2 and ModelSim simulator.

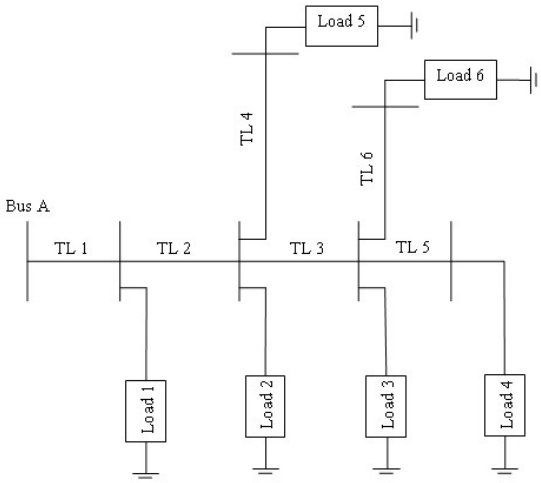

Figure 8
Test system for the second case study

Table I
Test System Parameters

Transmission Lines					
TL 1	TL 2	TL 3	TL 4	TL 5	TL 6
400 (Km)	100 (Km)	60 (Km)	200 (Km)	300 (Km)	150 (Km)
Loads					
Load 1	Load 2	Load 3	Load 4	Load 5	Load 6
R = 350 Ω L = 50 mH	R = 350 Ω L = 50 mH	R = 350 Ω L = 5mH	R = 250 Ω L = 10 mH	R = 250 Ω L = 10 mH	R = 250 Ω L = 10 mH

The test system is best implemented on a Xilinx Spartan II 2s300eft256-7 FPGA chip, which is a lower end product. This chip is capable of producing the output in less than 100 ns, which is much smaller than the desired time step.

Figure 9, shows a schematic of the implemented units for the network equivalent on the FPGA chip. The connections between these units are also shown.

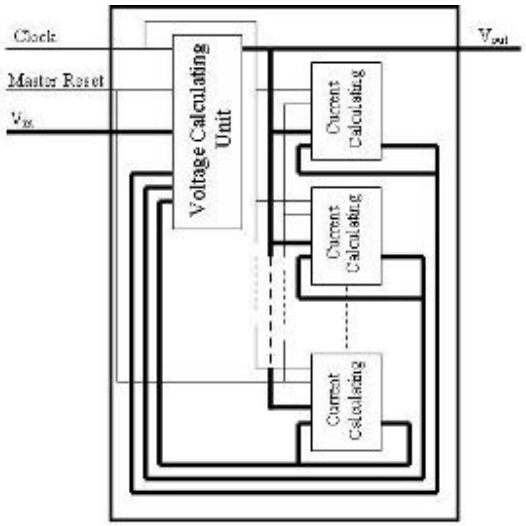

Figure 9
Schematic of the implemented network equivalent

Figure 10 compares between the voltage at the sending end Va of the test system and the VHDL simulation for a 100V dc energization scenario. Figure 11 shows the energization current of the test system and that of the VHDL simulation when energized by a 100 KV 50Hz ac voltage source.. Simulation step is 50μs for both cases. Both figures show that the simulated output of the VHDL implementation is within acceptable accuracy with respect to the test system response.

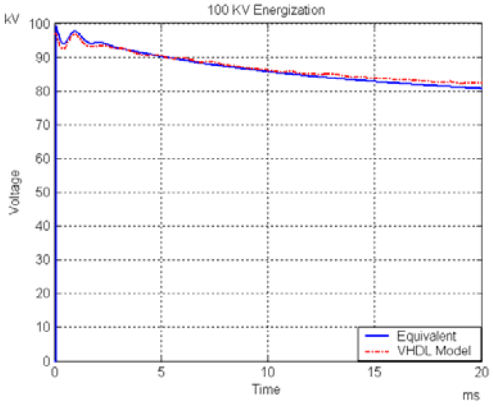

Figure 10
Comparison of the sending end voltage of the test system and the VHDL simulation when energized with 100 V DC source

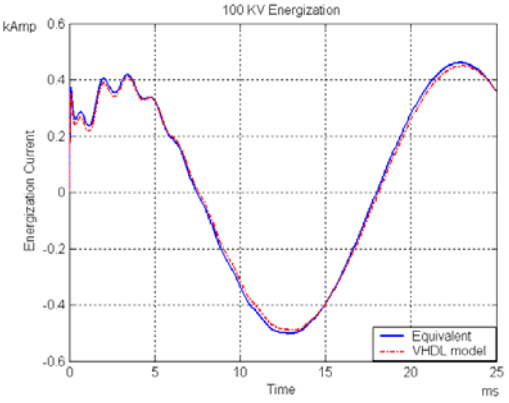

Figure 11
Comparison of the energization current of the external equivalent and the VHDL simulation for 100 KV, 50Hz AC energization

V. Conclusions

Implementing the mathematical model for the power system to be simulated on an FPGA chip and taking advantage of its parallel structure results in a very high performance that exceeds the performance of a single or multiple processors.

When using an FPGA chip a calculation time of 100ns which is less than 1/500 of the desired simulation time step (50μs to 100 μs) has been achieved.

VI. References

[1] J.R. Marti and L.R. Linares, "Real-Time EMTP-Based Transients Simulation", IEEE Trans. on Power Systems, Vol. 9, No. 3, August 1994.

[2] O. Devaux, L. Levacher and O. Huet, “An Advanced And Powerful Real-Time Transient Network Analyser”, IEEE Transactions on Power Delivery, Vol. 13, No. 2, April 1998.

[3] Berkeley Design Technology, Inc., “Choosing a DSP Processor”, Berkeley Design Technology, Inc., 2000.

[4] “The Evolution of DSP Processors”, Signal Processing Magazine, IEEE, Volume: 17, Issue: 2, March 2000

[5] H.W. Dommel, “Digital Computer Solutions of Electromagnetic Transients in Single and Multiphase Networks”, IEEE Trans. On Power App. and Sys. Vol. 88. April 1969.

[6] R. C. Restle, “Choosing between DSPs, FPGAs, µPs, and ASICs to implement digital signal processing ”, Conference proceedings of ICSPAT: DSP world, Autumn 2000.

[7] Bjørn Gustavsen and Adam semlyen, “Rational Approximation of Frequency Domain Responses by Vector Fitting”, IEEE TRANSACTIONS ON POWER DELIVERY, VOL. 14, NO. 3, July 1999

[8] M. Matar, M. Abd-El-Rahman and A. Soliman, “Developing an FPGA coprocessor for Real Time Simulation of Power Systems,” ICEEC’04, Egypt, August 2004

[9] J. R. Armstrong and F. G. Gray, “VHDL Design Representation and Synthesis” second edition, Prentice Hall PTR 2000.

[10] David Pellerin and Douglas Taylor, “VHDL made easy”, Prentice Hall PTR 1997.