WWW based Virtual SPICE Simulator

Tania Vassileva, Vassiliy Tchoumatchenko, Miroslav Mladenov, Ilario Asitinov
Department of Electronics, Technical University – Sofia
tkv@vmei.acad.bg

Abstract. The paper describes the developed network-based virtual laboratory that makes PSPICE, HSPICE and Berkeley SPICE simulators readily available over the Internet. Java-based distributed WebSpice allows for platform independence and provides users with consistent on-demand access regardless of time and location thus reducing start-up and training investments and allowing wide-area collaboration.

1. Introduction

VLSI system design typically involves design specification, optimisation and synthesis, generation of physical design and verification. In today's complex design process, designers require multiple CAD tools in order to perform the various tasks that make up a design process. The design of future high-performance VLSI systems will require a distributed design and verification methodology due to the diverse expertise required at various level of abstraction. These systems will need tools and generators that allow exploration of the design space at all level of abstraction and a design environment that allows distributed access to libraries, models and design tools. A particular problem within the university research community is limited resources of equipment and expertise available to researchers near their facilities. Advanced research requires access to resources that are scarce and may reside across the nation. Increasing size and geographical separation of design data and teams has created a need for a network-based cost-effective electronic design environment.

The advent of Internet has opened new opportunity in the areas of distributed design and the World Wide Web has emerged as the most desirable platform for distributed access to information. One objective of the distributed design environment should be to bring these tools to the designer in a simple transparent fashion, making it possible to run them in remote locations on demand. Designers could have access to the latest developments in computational resources from universities and CAD vendors and perform benchmarking on various design tools before committing to a specific tool purchase. A distributed design will facilitate access to cell libraries and VLSI CAD tools like synthesisers, generators, optimisers and simulators distributed over the Internet. Through the Internet, collaboration among geographically distributed researchers can be made possible.

To address these issues we aim to construct an operational prototype of a CAD design environment enabling Internet-wide IC design for the electronics industry. The goals are dual. In the small – to empower individual electronics designers by affording them efficient desktop access to the numerous, heterogeneous design resources. In the large – to reduce electronics industry market entry barriers to new entrepreneurs by providing a streamlined cost-effective design development environment. In reducing the costs, and shortening the time-to-market of new intellectual capital, we expect to stimulate the electronics industry to new growth.

The principal focus will be to leverage networking, web technology, visualisation and collaborative infrastructures to develop environment for distributed design of complex systems. One way of organising the major aspects of network and visualisation technologies are shown in Figure 1 [1].

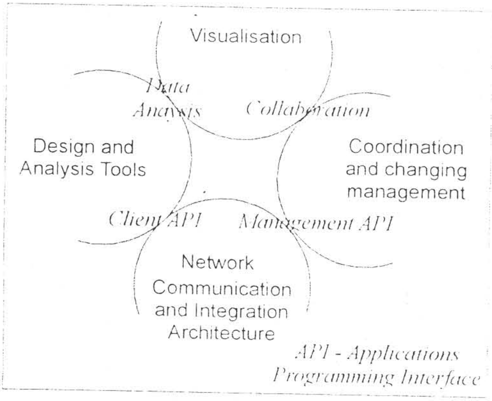

Figure 1. Major aspects of integration architecture

Our goal is to provide the distributed CAD tools and environment to support the design and verification of complex electronic systems. Initially we will focus on simulation, synthesis and physical design. The first prototype of distributed CAD tool, concerning VHDL simulation over the Internet was reported in [2-4].

The requirements to integrate more and more transistors on a single chip using deep sub-micron technology will introduce new challenges which include the management of power dissipation, dealing with interconnect, sizing for required delay, etc. All this tasks will need intensive performance verification by analogue simulation.

Spice [5] is a general-purpose electronic circuit simulation program. It has become the industry standard for analogue simulation, thanks in part to the free distribution of the original text based program by the University of California at Berkeley, where it was developed. The benefits of using Spice include [6]: cost savings, faster prototyping, more advanced analysis.

To answer enhanced interest to analogue simulation we have developed virtual laboratory that allows geographically distributed users (researchers and students) to access and run existing SPICE simulators via Web browser.

This paper describes the implementation model and user interface of the developed network-based virtual laboratory that makes PSPICE, HSPICE and Berkeley SPICE simulators readily available over the Internet.

2. Enabling Technologies for Web based CAD

In this section, we describe the technologies, mechanisms and protocols available that support distributed applications with the standard Web interface. Current tools support execution of applications at the server side using Hyper Text Transfer Protocol (HTTP) and Common Gateway Interface (CGI) and use Java technology, scripts and plug-ins to create applications that are executed locally on the user's machine. Another approach is to use object technologies to define, locate and request computational services from participating applications, both remotely and locally. In this approach, the Web takes the role of providing uniform access and presentation mechanisms.

CGI programs are usually referred to as scripts, which run on the server machine and produce the output to be displayed on the client's browser. The HTTP and CGI are the protocols that govern interactions between the client, server and script.

CGI programs and forms lack the interactivity and complex user interface. Java allows us to do complex client-side processing in a platform independent manner. Java applets can be embedded in HTML (Hyper Text Mark-up Language) pages, which are loaded from the Web server and run on the client-side browser as a mini application.

The Common Object Request Broker Architecture (CORBA) [7] provides an infrastructure, which enables invocations of operations on objects located anywhere in the network as if they were local to the application using them. CORBA simplifies heterogeneous distributed computing and enable location transparency, activation transparency, language independence, and platform neutrality. CORBA is object oriented, enabling many potential benefits such as reuse.

The object-Web architecture is a three-tier architecture [8] consisting of client tier, the application and data tier and the middle tier providing the middleware services. In this architecture, the client tier is build on Web browsers to provide a standard graphical interface, through which users can access tools and information via the Internet and the intranets. Lightweight Java client applications can run in Java-enabled browsers. Application can also be accepted using the HTML form based interface.

The application and data tier consist of different application tools distributed across the network. The tools are heterogeneous and run on diverse platforms. The application tools have well defined interfaces and they cooperate to build distributed applications through the middleware services.

3. Implementation Model of the Virtual SPICE Simulator

The virtual SPICE simulator is based on the three-tier client/server-computing model. Figure 2 shows a conceptual view of the simulator. On the client side the front-end software allows the user to prepare input files, upload them to the server, set simulation parameters and run the simulator as well as view and download output – all via standard Web browsers.

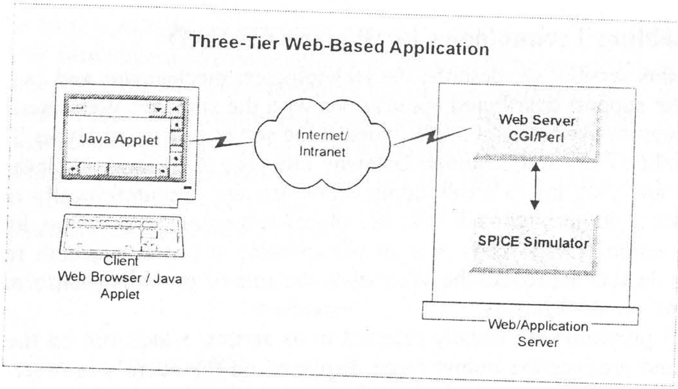

Fig.2. Implementation Model

On the server side the Web server handle access control, directory and file manipulation and runs the simulator via CGI scripts. In the current implementation the simulation program and the web server coexist on the same computer.

The front-end software is implemented as a Java applet and provides a Graphics User Interface (GUI) to the simulation environment (Fig.3). Once the model is loaded and the simulation parameters set the simulation job can be submitted to the server by pressing *Simulate* button.

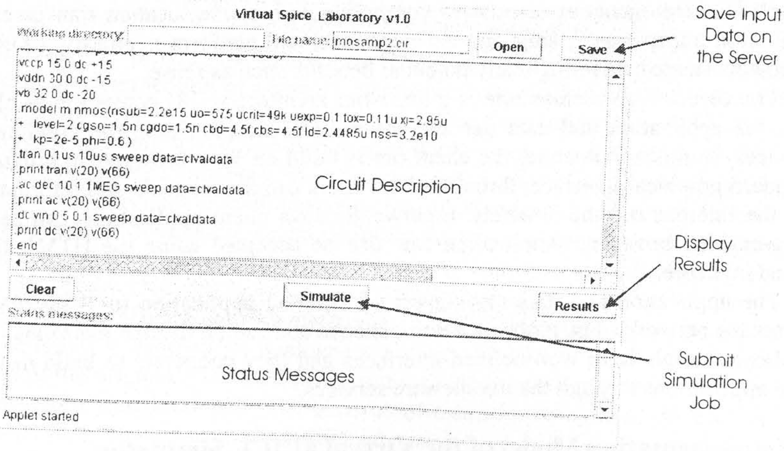

Fig. 3 Virtual SPICE – input window

To submit data to the server the applet opens http connection to the Web server and invokes the CGI script to create a working directory and SPICE model file on the server. We propose to use the POST method as standard for describing the interface specification of the Web based CAD applications. This is because the

POST method is capable of handling any arbitrary data transfer between client and server. The POST method is used to talk to the HTTP server, which would execute the CGI program.

After the applet invoke a second script, which use *system*() function to spawn a sub-process and to run sequentially the SPICE simulator. During the simulation all messages issued by the CGI scripts and by the simulator are send back to the client and displayed into the applet's *system messages* area.

Simulation results are stored in a file, which is read back by the applet. The results are visualized both in textual and graphical formats. The applet displays the output data in the browser.

4. User interface for data displaying

Virtual simulator performs DC, AC, transient and parametric analysis. The user interface for data visualisation is arranged in the manner similar to those in the real simulators. It permits to select analysis type as well as the variables for visualisation as shown in Figure 4.

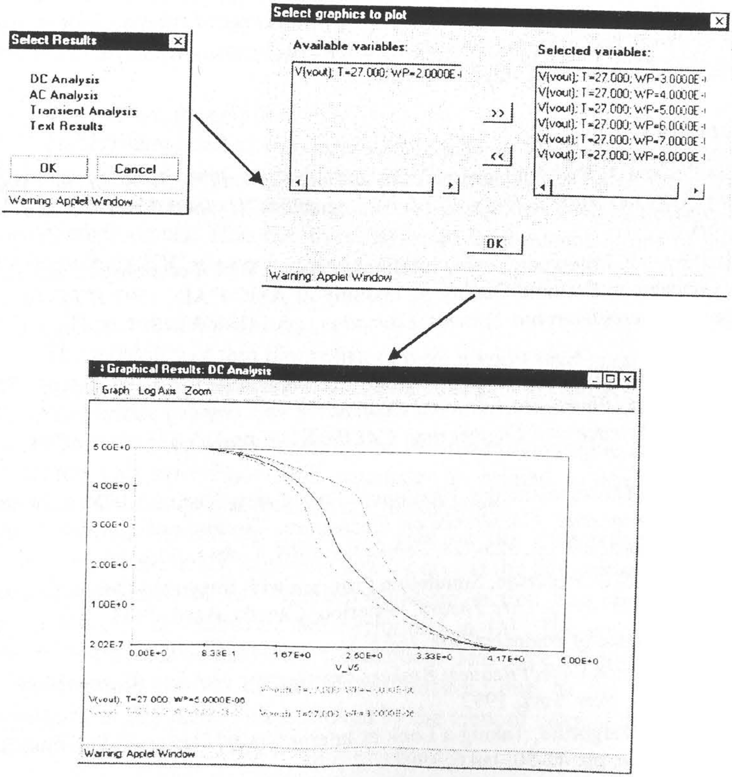

Fig.4 Dialog for selecting data for visualisation

Results from DC analysis of a CMOS inverter with transistor widths ratio as a parameter are displayed in the same figure.

Graphical data can be manipulated, by adding or delaying some curves, by changing axis in linear or logarithmic scales. Zoom facilities are also provided.

SPICE output file is displayed in text window.

Conclusions

The increasing complexity of electronics design mandates a distributed and collaborative design environment and the World Wide Web serves as a desirable platform realising these goals.

We have developed a Web based tool WebSpice, which provides Web access to the circuit simulator PSPICE for WindowsNT and HSPICE for Sun/Solaris. WebSpice is an independent Web tool and can be accessed through the HTML form. WebSpice provides input data entry through Web browser, simulates the SPICE netlist and sends back the results of simulation to the applet. The applet then displays the resulting graphics and text dàta in the browser.

Java-based distributed WebSpice allows for platform independence and provides users with consistent on-demand access regardless of time and location thus reducing start-up and training investments and allowing wide-area collaboration among geographically distributed researchers.

References

1. M.D.Spiller and A. Richard Newton, EDA and the Network, *IEEE/ACM Int. Conference on Computer Aided Desidn, ICCAD'97*, pp.470-476, Novrmber.1997, San Jose, CA, USA
2. T. Vassileva , V.Tchoumatchenko, I.Astinov, Mixing Web Technologies and Educational Concepts to Promote Quality of Training in ASIC CAD, 1997 *IEEE Int. Conference on Microelectronic Systems Education*, pp.149-150, 1997, Arlington, VA, USA
3. T.Vassileva, V.Tchoumatchenko, I.Astinov, I.Furnadziev, WWW based VHDL Training System, *Fourth International Conference on Computer Aided Learning and Instruction in Science and Engineering, CALISCE'98*, pp.125-131, June, 1998, Goteborg, Sweden
4. T.Vassileva, V.Tchoumatchenko, I.Astinov, I.Furnadziev, Virtual VHDL Laboratory, *5-th IEEE International Conference on Electronics, Circuits and Systems,* Instituto Superior Techniko, v.3, pp. 325-328, September 1998, Lisboa, Portugal
5. L.W.Nagel and D.O. Pederson, Simulation Program with Integrated Circuit Emphasis, *Proc. 16th MidWest Symp. Circ.Theory*, Waterloo, Canada, April, 1973
6. D.Harris, The Spice of engineers' life, *IEEE Spectrum*, pp. 27, June 1998
7. OMG, *The Common Object Request Broker. Architecture and Specification*, John Willey and Sons, New York, 1993
8. D.Alles and G.Vergottini, Taking a Look at Internet-based Design in the Year 2001, *Electronic Design*, pp. 42-50, Jan.6, 1997