Letters to the Editor

Internet as a New Graphical User Interface for the SPICE Circuit Simulator

Bogdan Wilamowski, Aleksander Malinowski, and John Regnier

***Abstract*—The Spice Internet Package (SIP) was developed using an Internet browser as a platform-independent graphical user interface. The SIP application has many options that include simulation of SPICE files, graphical postprocessing data, and online editing of SPICE files. It can run remotely through a network on any operating system.**

***Index Terms*—Design automation, Internet, SPICE.**

I. Introduction

Some of the most frequently used electronic design automation (EDA) tools by electronics engineers are SPICE programs [1]. Differences between various SPICE programs are described in [2]. SPICE programs are used not only for electronic circuit analysis and design, but are also used for analysis of electronic drives [3] or gaseous discharge lamps [4]. The common problem faced by many electronic engineers in the industry is that their design tools often work only on one or two operating platforms. Server-side software uses a common gateway interface (CGI) script. CGI scripts utilize PERL, PHP, or other scripting languages. The usual dilemma is to decide what programming language should be used for what part of the software package and how to partition its components between server and remote clients [5].

Network programming uses distributed resources. Certain information must be frequently sent both ways between client and server. It is also important to develop methods which take advantage of computer networks and platform-independent browsers. This would require solving several issues, such as: minimization of the amount of data to send by a network, task partitioning between the server and client, selection of proper programming tools used for various tasks, development of special user interfaces, security and access handling, and portability of software used on servers and clients.

For example, should graphics be generated on the server and sent to a client as a compressed image, or should only text and binary data be sent to the client and a Java or ActiveX applet used to generate the graphics there? In the first case, little data are sent from client to server, and much more data is sent back to the client as images. In the latter case, all data are transferred to the client machine together with a Java applet.

II. SPICE Internet Package (SIP)

In the case of the SIP [6], [7], it only makes sense to use CGI for the SPICE simulation, because it would be impossible to use applet technology and send the SPICE engine through the network every time it was requested. The SIP program currently incorporates CGI with

Manuscript received December 19, 2000; revised June 1, 2001. Abstract published on the Internet October 24, 2001.

B. Wilamowski is with the University of Idaho, Boise, ID 83712 USA.

A. Malinowski is with the Department of Electrical and Computer Engineering, Bradley University, Peoria, IL 61625 USA.

J. Regnier is with Micron Technology Inc., Boise, ID 83707 USA.

Publisher Item Identifier S 0278-0046(01)10284-4.

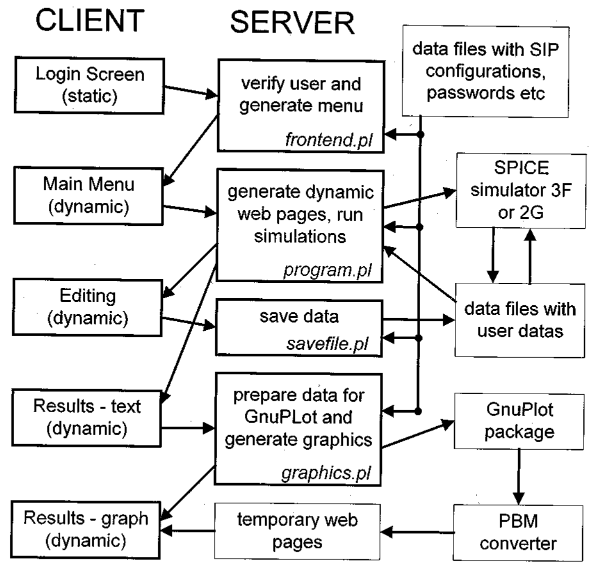

Fig. 1. Block diagram of the SIP.

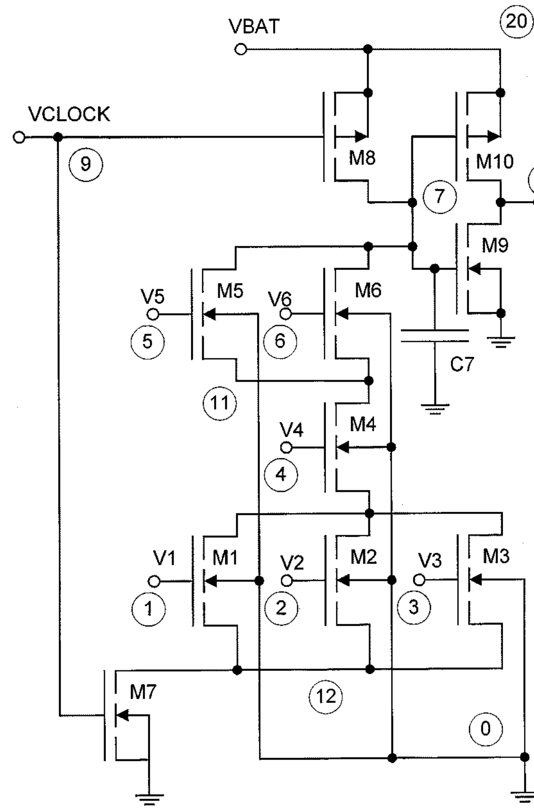

Fig. 2. Circuit of a domino gate used in SIP simulations.

PERL, and front-end with HTML and JavaScript. A unique feature of the SIP, versus other SPICE simulators, is that it is operating system independent. Anyone that has access to the Internet and a web browser can run a SPICE simulation and view the results graphically from anywhere in the world using any operating system.

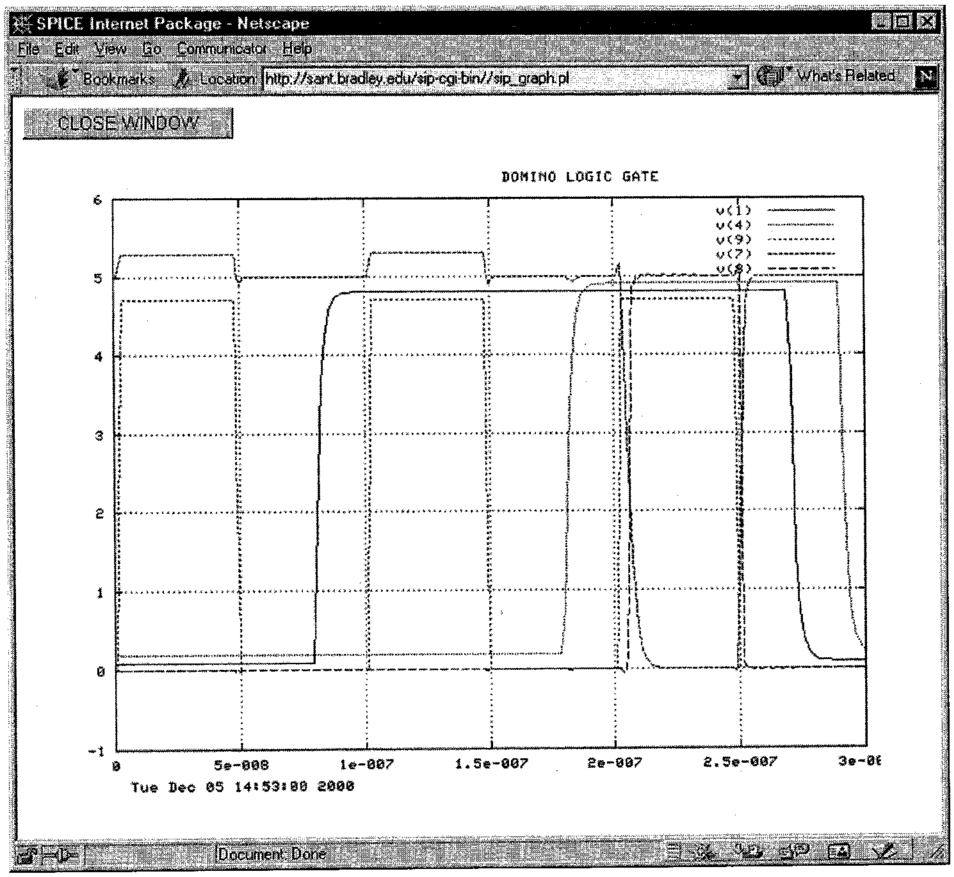

Fig. 3. Results of a transient analysis for the dynamic domino gate.

Fig. 1 shows the flow of information and the distinction between the client (to the left) and the server (to the right). Login information entered by the user is passed to the first CGI script, frontend.pl. The purpose of that script is to generate a custom front-end page that allows for the user's account management.

The current version of SIP accepts input data written in the SPICE net list format. The input data file can be entered in two ways: 1) by editing the file on the client's machine or 2) by uploading the data file. Each request triggers the execution of the second CGI script, program.pl. That script controls file access, processes text data, executes the SPICE engine, and generates a Web page with results.

The major SPICE engines used in the package are the SPICE3F5 and the SPICE2G versions for UNIX/VAX systems developed at the University of California at Berkeley [1]. Authors are using these software packages both in UNIX and Windows environments.

Most of the SPICE packages have system-dependent graphical postprocessors. In the SIP, these postprocessors were replaced with a portable code, which runs on every system. In the SIP the standard .PRINT function in the netlist is used to communicate between the SPICE engine and the graphical postprocessor. The required information from the output file is extracted using a text-processing unit written in PERL. The process of producing the graphical version of the simulation results uses the script graphics.pl. This script utilizes the GnuPlot package to generate plots. The GnuPlot package generates an output file in bitmap format, which cannot be directly displayed on web pages. Therefore, other software (NETPBMP) is used to convert the large bitmap into a highly compressed GIF format that can be handled by a Web browser. An example with the CMOS domino dynamic gate is shown in Fig. 2, while the corresponding SIP graphical output is shown in Fig. 3.

III. Conclusion

Remote access to the SIP allows users to run SPICE simulations from any computer on the network. The current SPICE engine allows simulating circuits with an unlimited number of transistors. The SIP can be currently accessed on the following URLs: http://gdansk.bradley.edu/~sip/ (UNIX server), http://nn.uidaho.edu/sip/, and http://sant.bradley.edu/sip/ (Windows servers).

Another example of using the Internet as a graphical user interface is ICP [8] where computer programs written in C, C++, PASCAL, FORTRAN, and JAVA could be compiled on web pages. Availability of design tools via the Internet could boost design processes in many new communities, small businesses, and improve our education processes at universities by allowing students to use the same sophisticated software as is used by the leading industries.

References

[1] T. Quarles, A. R. Newton, D. O. Pederson, and A. S. Vincentelli, *SPICE3 Version 3F5 User's Manual*. Berkeley, CA: Univ. of California Press, 1994.

[2] B. Wilamowski and R. C. Jaeger, *Computerized Circuit Analysis Using Spice Programs*. New York: McGraw-Hill, 1997.

[3] M. A. Inerfield, S. L. Garverick, W. S. Newman, and Y. Zhang, "A SPICE model for a novel brushless adjustable-speed drive," *IEEE Trans. Ind. Electron.*, vol. 47, pp. 1307–1318, Dec. 2000.

[4] T.-F. Wu, J.-C. Hung, and T.-H. Yu, "A PSPICE circuit model for low pressure gaseous discharge lamps operating at high frequency," *IEEE Trans. Ind. Electron.*, vol. 44, pp. 428–431, June 1997.

[5] "Can Internet-based applications be engineered?," *IEEE Softw.*, vol. 15, pp. 104–110, Sept./Oct. 1998.

[6] B. Wilamowski, J. W. Regnier, and A. Malinowski, "SIP—Spice internet package," in *Proc. IEEE Int. Symp. Industrial Electronics (ISIE'98)*, Pretoria, South Africa, July 7–10, 1998, pp. 493–496.

[7] J. W. Regnier and B. Wilamowski, "SPICE simulation and analysis through Internet and Intranet networks," *IEEE Circuits Devices Mag.*, vol. 14, pp. 9–12, May 1998.

[8] A. Malinowski and B. Wilamowski, "Internet accessible compilers in software and computer engineering," in *Proc. ICSEE/Western Multiconf.*, San Francisco, CA, January 17–20, 1999, pp. 221–224.

Six Self-Lift DC–DC Converters, Voltage Lift Technique

Fang Lin Luo

***Abstract*—Voltage lift technique is a popular method widely applied in electronic circuit design. Applying it has created six new dc–dc step-up converters, namely, self-lift dc–dc converters, which possess high output voltage with smooth ripples. Therefore, these converters can be used in computer peripheral equipment and industrial applications.**

***Index Terms*—Self-lift dc–dc converter, voltage lift technique.**

I. Introduction

Voltage lift technique is a popular method widely applied in electronic circuit design. It has been successfully applied in dc–dc converters [1]–[4]. This paper introduces the skills to design new self-lift dc–dc converters using voltage lift technique. The six self-lift converters are a group of new dc–dc step-up converters, which are developed from the basic prototypes [1]–[7]. For all circuits, the load is usually resistive, i.e.,

$$R = V_O/I_O.$$

The normalized load is

$$z_N = \frac{R}{fL_{\text{eq}}}$$

where L_{eq} is the equivalent inductance [7].

We concentrate on the absolute values of all voltages and currents in the following description and calculations. Their direction (polarity) is defined and shown in the corresponding figures. We also assume that the semiconductor switch and the passive components are all ideal. For any component X (e.g., C, L and so on): its instantaneous current and voltage are expressed as i_X and v_X. Its average current and voltage values are expressed as I_x and V_x. The input voltage and current are V_O and I_O; the output voltage and current are V_I and I_I. T and f are the switching period and frequency. The six self-lift dc–dc converters are shown in Fig. 1. They are: Fig. 1(a) self-lift Cuk converter; Fig. 1(b) self-lift positive output Luo-converter; Fig. 1(c) reverse self-lift positive output Luo-converter; Fig. 1(d) self-lift negative output Luo-converter; Fig. 1(e) reverse self-lift negative output Luo-converter; and Fig. 1(f) self-lift Sepic converter.

Manuscript received January 2, 2000; revised June 1, 2001. Abstract published on the Internet October 24, 2001.

The author is with Nanyang Technological University, Singapore 639798 (e-mail: eflluo@ntu.edu.sg).

Publisher Item Identifier S 0278-0046(01)10285-6.

The voltage transfer gain of all circuits in continuous mode is

$$M = \frac{V_O}{V_I} = \frac{I_I}{I_O} = \frac{1}{1-k}.$$

Detailed analysis is given in the following sections. Due to the length limit of this letter, only the simulation and experimental results of the self-lift Cuk converter are given. However, the results and conclusions of other self-lift converters should be quite similar to those of the self-lift Cuk converter.

II. Self-Lift Cuk Converter

The self-lift Cuk converter and its equivalent circuits during switch-on and switch-off periods are shown in Figs. 1(a) and 2(a). It is derived from the Cuk converter. S and D_1 are on, and D is off during the switch-on period in Fig. 2(b). D is on, and S and D_1 are off during the switch-off period in Fig. 2(c).

A. Continuous Conduction Mode (CCM)

In steady state, the average inductor voltages over a period are zero. Thus, $V_{C1} = V_{CO} = V_O$. During the switch-on period, the voltages across capacitors C and C_1 are equal, so that

$$V_C = V_{C1} = V_O. \tag{1}$$

The inductor current i_L increases during switch-on and decreases during switch-off. The corresponding voltages across L are V_I and $-(V_C - V_I)$. Therefore,

$$kTV_I = (1-k)T(V_C - V_I).$$

Hence,

$$V_O = V_C = V_{C1} = V_{CO} = \frac{1}{1-k}V. \tag{2}$$

The voltage transfer gain in the CCM is

$$M = \frac{V_O}{V_I} = \frac{I_I}{I_O} = \frac{1}{1-k}. \tag{3}$$

The characteristics of M versus conduction duty cycle k are shown in Fig. 2(d).

Since all the components are considered ideal, the power loss associated with all the circuit elements are neglected. Therefore, the output power P_O is considered to be equal to the input power P_{IN}

$$V_O I_O = V_I I_I.$$

Thus,

$$I_L = I_I = \frac{1}{1-k} I_O.$$

During switch-off,

$$\begin{aligned} i_D &= i_L \\ I_D &= \frac{1}{1-k} I_O. \end{aligned} \tag{4}$$

The capacitor C_O acts as a low-pass filter, so that

$$I_{LO} = I_O.$$

The peak-to-peak variation of current i_L is

$$\Delta i_L = \frac{kTV_I}{L}.$$