

## Virtual Lab for Electronics

Pranali Ninawe<sup>1</sup>, Roshan Kumar<sup>2</sup>, Priyanka<sup>3</sup>, Ashish Durgude<sup>4</sup>

<sup>1,2,3,4</sup>Students, Computer Engineering, Pune Institute of Computer Technology, Maharashtra, India

\*\*\*

**Abstract** - The major problem with the student of any university dealing with the hardware related assignment is, either the hardware is not available or else the ratio of student to experiments kit per batch is not appropriate. So, the solution is virtualization. Now-a-days so many students are currently using the online labs under the virtual labs project. Many of these labs are being accessed outside the regular lab hours. Hence, by performing a virtual lab before a physical lab, it helps students to learn from their mistakes and finally doing their experiments correctly in their physical labs. Thus, we are concentrating on the experiments being performed as per the university syllabus in electronics of all years.

**Keywords:** Learning environments, tools, e-learning, e-Assessment, online education, virtual labs, computer and education, simulation.

### 1. INTRODUCTION

Electronic circuit design requires an accurate method of assessing circuit performance. For the design of discrete circuits, the traditional "breadboard" is a convenient method of measuring the electrical characteristics of a circuit. The circuit can be modified and design improvements can be made immediately. Since a breadboard closely resembles the circuit that finally will be built, the laboratory measurements yield an accurate characterization of the final circuit performance.

A computer program that simulates the electrical performance of an electronic circuit circumvents many of the practical problems that are encountered in circuit characterization. The circuit is represented in mathematical terms, and numerical analysis procedures that correspond to typical laboratory measurements are performed. The output of the simulation program therefore simulates the results of laboratory measurements. Moreover, circuit simulation can provide information about circuit performance that virtually is impossible to obtain with laboratory measurements.

This project is a digital program that simulates the electrical performance of electronic circuits. It contains models for the common circuit components and is capable of simulating most electronic circuits. We have extended the concept further to allow students to carry out the experiments on the web browsers and devices of their choice. The simulations themselves are carried out utilizing cloud computing infrastructure such as those provided by Amazon and Microsoft.

Our project is completely web-based, which not only means it's cross-platform (Windows/Mac/Linux) and requires no installation or plugins.

### 2. CIRCUIT DEFINITION

The program input defines the circuit to be simulated on an element by element basis. This project uses elements like resistors, capacitors, inductors, independent voltage and current sources, and the four most common semiconductor devices: the diode, the junction field-effect transistor (JFET), and the insulated-gate field-effect transistor (IGFET or MOSFET).

A. Linear Elements :

1. Resistor (R)
2. Capacitor (C)
3. Inductor (L)
4. Independent Voltage Source (V)
5. Independent Current Source (I)

B. Non-linear Elements :

1. Nonlinear Voltage-Controlled Current Source (N)
2. Diode (D)
3. Junction Field-Effect Transistor (J)
4. Insulated-Gate Field-Effect Transistor (M)

The linear elements require only one or two parameter values to specify completely the electrical characteristics of the element. However, the models for the semiconductor devices, which includes the MOSFET, are more complicated and contain many parameters.

In addition to defining the circuit, the program input specifies the analysis to be performed and the output to be generated. The simulation of an electronic circuit usually requires a combination of three basic analysis: DC analysis, time-domain transient analysis, and small-signal AC analysis. In addition, several sub-analysis capabilities also are incorporated.

DC Analysis:

- DC operating point
- Linearized Device Model Parameterization
- Small-Signal Transfer Function
- Small-Signal Sensitivities
- DC Transfer Curves

Transient Analysis:

- Time-Domain response

AC Analysis:

- Small-Signal Frequency-Domain Response

### 2.1. DC ANALYSIS

It determines the idle operating point of the circuit. All energy-storage elements in the circuits are ignored in a DC analysis by treating capacitors as open circuits and inductors as short circuits and inductors as short circuits, the program prints the circuit node voltages.

### 2.2. TRANSIENT ANALYSIS

It determines the time-domain response of the circuit to specified time-domain inputs. The initial time point, arbitrarily defined as time zero, is determined by a previous DC operating point solution. The time interval (0, T) that is specified by the user is divided into discrete time points and the program determines the circuit solution at each successive time point starting from time zero.

Voltage or current output variables are stored at each time point, and can be listed in plotted form at the conclusion of the analysis.

### 2.3. AC ANALYSIS

The small signal linear equivalent circuit is analyzed in the frequency domain with the phasor method. All circuit voltages and currents are complex variables that usually are expressed in terms of magnitude and phase.

## 3. DESIGN OF CIRCUIT

The input subprogram reads the input file, constructs a data structure from this input, and checks the data structure for obvious user error. The data structure, after the input phase, contains a complete, self-consistent description of the circuit. After the input subprogram has executed successfully, the setup subprogram constructs additional data structure that are required by the analysis subprogram. The setup subprogram constructs the sparse matrix subroutines for the analysis subprogram.

The input subprogram reads the input file, constructs a data structure from this input, and checks the data structure for obvious user error. The data structure, after the input phase, contains a complete, self-consistent description of the circuit. After the input subprogram has executed successfully, the setup subprogram constructs additional data structure that are required by the analysis subprogram. The setup subprogram constructs the sparse matrix subroutines for the analysis subprogram.

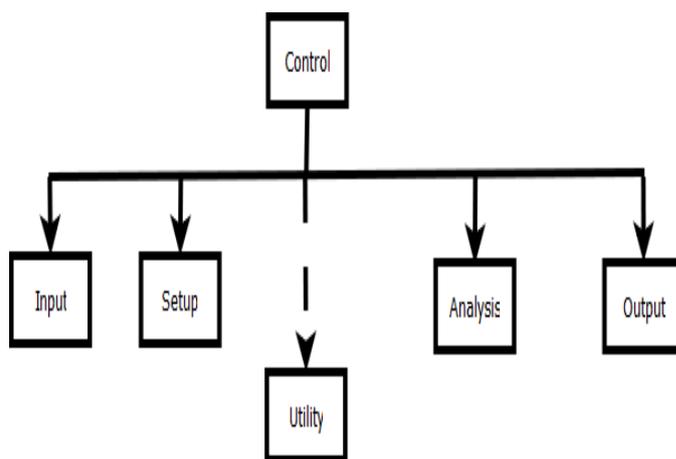


Fig -1: Circuit Design chart

## 4. CIRCUIT ANALYSIS

The analysis portion of a circuit simulation program determines the numerical solution of a mathematical representation of the circuit. To accomplish the transition from the physical circuit to a mathematical system of equations, each element in the circuit is represented by a mathematical model.

After each element in the circuit is modeled, the system of equations that describe the complete circuit is determined by the model equations for each element and topological constraints that are determined by the interconnection of the elements. The topological constraints reflect Kirchhoff's Current Law (KCL) and Kirchhoff's Voltage Law (KVL). Kirchhoff's Current Law states that the net incoming and outgoing current is zero. Hence, for any component that has k nodes, it will contribute to k KCL equations in the circuit. All the equations should be solved simultaneously, which implies to solving a big matrix.

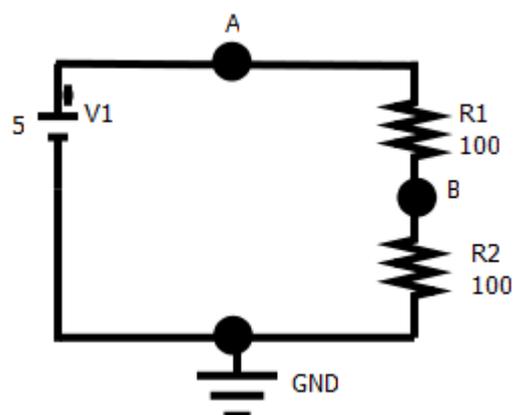


Fig -2: Sample Circuit-1

The voltage at A, B, GND is denoted as  $v(A)$ ,  $v(B)$  and  $v(G)$  respectively.

The KCL at B is:

$$\frac{v(A) - v(B)}{100} - \frac{v(B) - v(G)}{100} = 0$$

The KCL at GND is:

$$v(G) = 0$$

The KCL at A is:

$$i(V1) - \frac{v(A) - v(B)}{100} = 0$$

The KCL at i(V1) is:

$$v(A) - v(G) - 5 = 0$$

The above equations can be written in matrix form ( $Ax = B$ ):

$$\begin{pmatrix} 1 & -2 & 1 & 0 \\ 100 & 100 & 100 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \\ 100 & 100 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} v(A) \\ v(B) \\ v(G) \\ i(V1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 5 \end{pmatrix}$$

The above matrix is solved using linear solver and we get the values for the unknown variables as

$$v(A) = 5 \text{ volts}, v(B) = 2.5 \text{ volts}, v(G) =$$

$$0 \text{ volts and } i(V1) = 0.025 \text{ A.}$$

#### 4.1. DC ANALYSIS

The DC analysis portion determines the DC operating point of the circuit with inductors shorted and capacitors open. A DC analysis is automatically performed prior to transient analysis to determine the transient conditions, and prior to AC small-signal analysis to determine the linearized, small signal models for nonlinear devices.

The DC analysis of a linear circuit therefore requires only a method of formulating the circuit equations and a linear solution algorithm to solve the equations. Different methods for formulating the circuit equations are available. Among those, Nodal Analysis is the simplest one. The circuit equations are determined by writing KCL equations for different loops within the circuit.

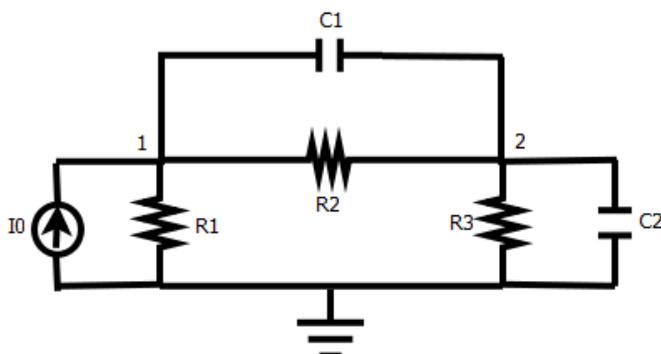


Fig-3: Sample Circuit-2

If the capacitors in the circuit are ignored, Nodal Analysis produces the following system of equations:

$$\begin{bmatrix} \frac{1}{R1} + \frac{1}{R2} & -\frac{1}{R2} \\ -\frac{1}{R2} & \frac{1}{R2} + \frac{1}{R3} \end{bmatrix} \begin{bmatrix} v1 \\ v2 \end{bmatrix} = \begin{bmatrix} I0 \\ 0 \end{bmatrix}$$

#### 4.2. AC ANALYSIS

The AC small-signal portion computes the AC output variables as a function of frequency. The program first computes the DC operating point of the circuit and determines linearized, small-signal models for all of the nonlinear devices in the circuit. The resultant linear circuit is then analyzed over a user-specified range of frequencies. The desired output of an AC small-signal analysis is usually a transfer function. If the circuit has only one AC input, it is convenient to set that input to unity and zero phase, so that output variables have the same value as the transfer function of the output variable with respect to the input.

#### 4.3. TRANSIENT ANALYSIS

The transient analysis portion computes the transient output variables as a function of time over a user-specified time interval. The initial conditions are automatically determined by a DC analysis. All sources which are not time dependent (for example, power supplies) are set to their DC value.

### 5. RESULTS

The following is a screenshot of our application with its respective circuit and graph:

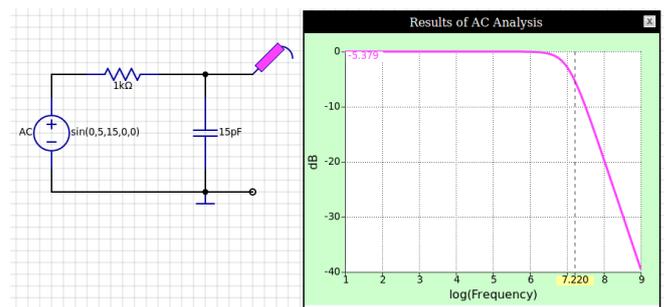


Fig-4: Result

### 6. REFERENCES

1. Ranjan Bose, "Virtual labs Project: A Paradigm Shift in Internet -based Remote Experimentation", IEEE, October 18, 2013.
2. Debabrata Bagchi, Kapil kaushik and Bhanu Kapoor, "Virtual Labs for Electronics Engineering using cloud

computing”, IEEE interdisciplinary engineering design education conference, 2013.

3. Anita Diwakar, Sushant Poojary and S.B Noronha, “Virtual labs in engineering education: Implementation using free and open source resources”, IIT Bombay Powai, 2012.