

# MNA Generation and Circuit Simulation

Carleton University  
ELEC3999A-Work Term Report 1

Seyed-Ali Nouri  
100923591  
[alinouri@cmail.carleton.ca](mailto:alinouri@cmail.carleton.ca)

PROFESSOR R. ACHAR  
DEPARTMENT OF ELECTRONICS

## **Executive Summary**

Computer Aided Design has become a very important part of electrical engineering. As systems are becoming larger and more complex, it is no longer possible to solve them by hand. Computer programs are required to help in the design and the simulation of these systems. This significantly reduces costs for companies as systems can be virtually built and tested with high accuracy before fabrication. An important concept when trying to convert circuits into mathematical models, as covered in this report, is Modified Nodal Analysis. This stamps circuit components into matrix equations that can then be solved using mathematical tools such as MATLAB. Another very important tool when simulating circuits is HSPICE. This programs take circuit data and outputs the simulation results in one step. Due to the number of different tools used in the industry, it becomes vital to compare results obtained using different methods. This can be done by extracting the data and formatting it correctly. Once the formatting is complete, the data can be imported into MATLAB where the results can be plotted into a single graph for easy comparison. Being able to use and understand the above mentioned programs will prove beneficial when looking for projects or jobs in the future. By teaching me about Computer Aided Design and circuit simulation it has opened many new doors and has motivated me to learn and explore new ideas.

## **Acknowledgements**

I would like to thank my supervisor, Professor Ram Acharya for giving me the opportunity to expand my knowledge during this work term. This was truly an eye opening experience for a first year student such as myself. I would like to thank Professor M. Nakla for allowing me to use his lecture notes to better understand Modified Nodal Analysis. I must also thank everyone else who helped me throughout the work term.

## Table of Contents

<b>1</b>	<b>Introduction</b> .....	1
1.1	Report Structure .....	1
1.2	Organizational Context .....	1
1.3	Technical Background .....	1
1.3.1	MNA and Stamping .....	1
1.4	Motivation.....	4
1.5	Objectives .....	4
<b>2</b>	<b>Development</b> .....	5
2.1	Creating an MNA Simulator .....	5
2.1.1	MNA Generator .....	5
2.1.2	Frequency Domain Solver .....	5
2.2	HSPICE.....	5
2.2.1	Netlist.....	5
2.2.2	Listing File .....	6
2.3	Importing Simulation Results Into MATLAB .....	6
2.3.1	MATLAB simulator results .....	6
2.3.2	HSPICE results .....	7
2.4	Comparing the Results .....	8
2.5	Writing a small handbook .....	8
<b>3</b>	<b>Reflection on Work Experience</b> .....	9
3.1	Relation to Academic Studies .....	9
3.2	Career Development .....	9
3.3	Overall Experience.....	10
<b>4</b>	<b>Conclusion</b> .....	11
4.1	Future Work .....	11
<b>5</b>	<b>List of Abbreviations</b> .....	12
<b>6</b>	<b>References</b> .....	13

# 1 Introduction

## 1.1 Report Structure

This technical report is organized into 4 major components. The introduction provides background information on the employer and technical information behind the work. It also discusses the motivations behind the work and the objectives that were planned before beginning the project. The Next major section will go over the tasks completed during the work term. It will provide a technical summary for each portion of the project. The next section will be a reflection on the work completed and how I have benefited from this opportunity. The last component will be a conclusion that will summarise the report and explain future work that can be done. Together, the components will create a comprehensive overview of the work completed during my first Co-op term.

## 1.2 Organizational Context

I was privileged to work for the Department of Electronic at the University of Carleton. The Department of Electronics is active in many areas of research such as Very Large Scale Integration (VLSI) Design, Computer Aided Design (CAD), and Photonics [1]. Most of my project was based on CAD and for this reason I worked with professor Ram Achar who is very knowledgeable in this field. His research is based on adapting VLSI design/verification methodologies and CAD tools to deal with higher-density, lower-power, higher speed, and more complex designs [2]. The head of the Faculty of Engineering and Design is Dean Rafik Goubran [3]. The head of the Department of Electronics, is Prof. Niall Trait [4] and the president of the university is Dr. Roseann Runte [5].

## 1.3 Technical Background

### 1.3.1 MNA and Stamping

Modified Nodal Analysis (MNA) is a technique often used to automate the setup of a system of equations [6]. The system of equations can then be used to determine the voltage at any node in the system. MNA is based on Kirchhoff's Current Law (KCL) which states that the algebraic sum of the current exiting and entering a node is zero [7]. The general process of analysing these systems involves creating a matrix equation and solving for the unknowns using linear algebra. Generating the matrix equation can be done in two ways. The first method involves writing out the KCL equation for each node and then manually inputting it into a matrix equation. The second method is more automated because it skips the initial step of writing the equations. It works by using stamping to import the given information into an NxN matrix, where N is equivalent to the sum of the number of nodes, the number of dependent/independent voltage sources, and the number of inductors. Stamping requires a set of rules that need to be followed in order to achieve the correct results. The rules vary with each type of component. For example, stamping resistors is different from stamping capacitors. To better understand this concept, the following example

should be followed. Please note that the following example only contains resistors and a single current source for simplification purposes. This method still stand with more complex circuits.

**Example 1:**

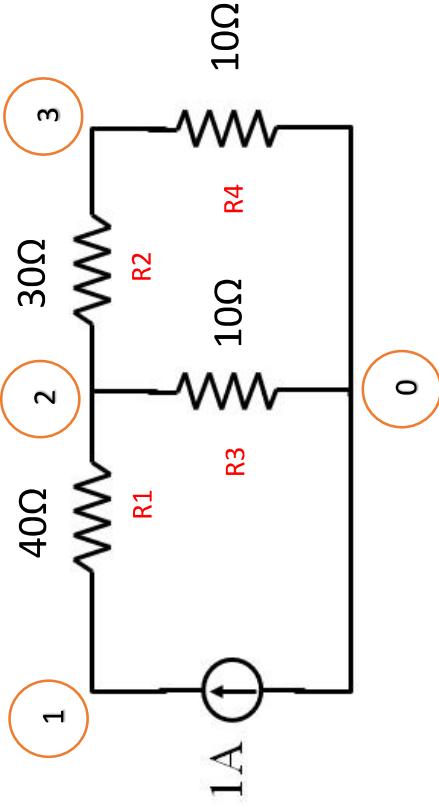


FIGURE 1: Sample Circuit To Help Explain MNA

The following steps need to be followed in order to understand MNA :

1. Number all nodes as seen in figure 1.
2. Identify a reference node that will be known as ground (i.e. 0 Volts) and label it as node 0.
3. Label all the resistors as seen in figure 1.
4. Organize the given information as follows.

TABLE 1: Given Information

Label	Value	Node (i)	Node (j)
R1	40	1	2
R2	30	2	3
R3	10	2	0
R4	10	3	0
Current source	1	1	0

5. Stamping is now used to import the resistors into a matrix. This is done by adding the conductance (g) to  $G_{ii}$  and  $G_{jj}$ . One must also subtract g from  $G_{ij}$  and  $G_{ji}$  [8].

$$GX=b \quad (\text{Eq. 1})$$

$$G = \begin{bmatrix} \frac{1}{40} & -\frac{1}{40} & 0 \\ -\frac{1}{40} & \frac{1}{40} + \frac{1}{30} + \frac{1}{10} & -\frac{1}{30} \\ 0 & -\frac{1}{30} & \frac{1}{30} + \frac{1}{10} \end{bmatrix} \quad (\text{Eq. 2})$$

6. The current source must be taken into account by stamping it into matrix  $b$ . This is done by adding the magnitude of current in row I and subtracting it from row j [8]. The result is equation 3.

$$b = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (\text{Eq. 3})$$

7. Now the system can be solved using row reduction or other mathematical tools such as MATLAB. The following results are obtained:

TABLE 2: MNA Results

Voltage at node 1:	8V
Voltage at node 2:	8V
Voltage at node 3:	2V

Please note that this example only included resistors and a current source. The same stamping procedure is used with other components such as capacitors, inductors, and voltage sources. The exact stamping technique for these components will not be covered since they are out of the scope of this document. For the proper stamping procedure of each component please refer to [8].

Since this process has a predetermined algorithm, it becomes possible to automate the process of creating the matrices using computer simulation programs. These programs accept netlists similar to table 1 and they go through the entire algorithm to stamp the components and solve the equations. This makes it possible to simulate large circuits much more efficiently.

## 1.4 Motivation

There are many tools that exist for circuit simulation. A frequently used, industrial-grade, simulation tool is HSPICE. A common problem is being able to import the simulation results from the simulator into another software (e.g. in MATLAB) to undertake the post-processing and comparison stages. A solution was required that would import the data obtained using the HPICE and MATLAB simulations into a single MATLAB file. Once this was completed, the results could then be plotted into a single graph.

## 1.5 Objectives

There were two main objectives for this work term. The first objective was to implement Modified Nodal Analysis (MNA) and to complete the required simulations using MATLAB. To accomplish this, a good understanding of matrix construction using MNA was required. The second objective was to create a small guideline for students to follow in order to plot and compare, on to the same graph, the results obtained using multiple simulation tools. To accomplish this, multiple methods were required to import data generated from both MATLAB and HPSICE. With this tool, the students will be able to compare results more easily and with better accuracy.

## 2 Development

The following section presents the work done during the work term. It will divide the work into subsections which will be explained in the order of completion.

### 2.1 Creating a Circuit Simulator

After learning how to implement MNA by hand, I wrote a program in MATLAB that automates the process. The purpose of this program is to read circuit data and to output results as seen in table 2. To accomplish this task, the program had to be divided into two parts. The following subsections will go over the function of each part of the program.

#### 2.1.1 MNA Generator

This part of the program is designed to read in the data and to generate the matrix equation. There is one main file that calls upon multiple function files. Each function is designed to stamp a specific type of component to the appropriate matrix. Global variables are needed to make this system work. This type of variable allows the program to add on to the matrix that is in memory every time it is stamping a new component. Once all the components are incorporated into the equation, the program then prints the results into a separate file for reading in the future. In conclusion, the sole purpose of this program is to generate the matrix equation that will be used by another program to solve the system.

#### 2.1.2 Frequency Domain Solver

Using the MNA generator, I am able to generate the appropriate equation that can be solved using LU factorization. LU factorisation is technique commonly used to solve matrix equations. Its mathematical properties allow this technique to be highly efficient for problems with varying b matrices (refer to Eq. 1) [9]. Following the completion of the calculations, the results are printed and saved as a MATLAB data file (\*.mat). This saved file will prove useful when comparing the obtained results with the outcome of the HSPICE simulation.

### 2.2 HSPICE

Another way to solve these problems is to use commercially available simulators such as HSPICE. These simulators are very complex and much more flexible than the simulator I wrote using MATLAB. This is mainly due to the immense resources that these companies dedicate to perfect their software. Nevertheless, it is important to be able to complete simulations using both methods. Completing simulations using HSPICE requires a good understanding of the file formats used by the software. The following are the two main file format and how I used them.

#### 2.2.1 Netlist

HSPICE reads in Netlist files. These are a special type of text file (\*.sp) that can be created using Notepad. They contain both the circuit data and the instructions for HSPICE to follow. In order for HSPICE to understand the Netlist, a predetermined syntax needs to be used.

Most Netlist files begin with an asterisk to signify a comment. HSPICE will ignore all lines beginning with either an asterisk or a dollar sign. To signify the end of the document .END must be used. Example 2 is a well formatted piece of data in a Netlist.

**Example 2:**

```
*sample1
Vin 1 0 3
R1 1 2 10
R2 2 0 20
.END
```

As seen in example 2, the first column of every line specifies the component being described (i.e. resistor). The second and third columns specify the nodes before and after the component, respectively. The third column is then used to describe the values. For instance, as seen in example 2, Vin is between nodes 1 and Ground. It has a magnitude of 3 Volts.

For the purposes of my work, I wrote an HSPICE Netlist file (\*.sp) with the proper simulation options and the same data that I gave to my MATLAB simulator. I then ran the simulation which printed the results in a Listing file.

### 2.2.2 Listing File

Once the simulation was completed the results were outputted into a Listing file. This file contained a lot of information. It had the results, any errors that were encountered, and the simulation statistics. The statistics are always written at the bottom of the document and the results of the simulation are printed right before the statistics. The results are formatted in labeled columns that are easy to identify. Unfortunately, having the results printed in this way made it difficult to compare them with the MATLAB simulation results. This was because the output data is in a different format and very complex. To create a better comparison, a program had to be written to plot the result of both simulations on the same graph. This would have to be done on MATLAB.

## 2.3 Importing Simulation Results Into MATLAB

Since MATLAB is to be used to compare the results, the obtained simulation results need to be in a format that MATLAB comprehends. MATLAB is capable of importing any text file containing numbers arranged in columns and thus all files need to be reformatted accordingly. The following are the methods used.

### 2.3.1 MATLAB simulator results

The results generated using my simulator in section 2.1.2 are created by MATLAB and thus they do not need to be reformatted when importing back into MATLAB for plotting.

### 2.3.2 HSPICE results

As explained in section 2.2.2, the listing file generated by HSPICE contains the results. However, the results are sandwiched between compiling messages, error message and simulation statistics. MATLAB is unable to comprehend this when importing the file. To fix this issue, the raw data needed to be extracted from the listing file. The following three methods were tested by me.

#### 2.3.2.1 *Manual Formatting*

This method was very straightforward. I opened the listing file and manually copy pasted the simulation data into a separate \*.txt file. By eliminating the unnecessary text in the listing file, I was able to import the data into MATLAB. The downfall of this method was the time it took to manually identify the list of data and select it. This could prove to be impractical when dealing with extremely large and complex circuits.

#### 2.3.2.2 *Automatic Formatting*

To speed up the process, I chose to create a program using C++ that would independently identify the relevant information and extract the data into a separate text files. The program first works by asking the user for the address of the Listing file generated by HSPICE. It then reads through the file looking for the pattern that signifies the beginning of the data. Subsequently, it reads through the data, line by line, and copies it into a new text file. Once all the data has been read, the program exits and saves the new text file that it has created. This program makes it possible to extract any length of data, rapidly and without losing accuracy. Please note that even though this method obtains the same result as manual formatting, it is well suited for the nature of automation of the process and proves efficient in the case of repetitive simulations. This translates into saved time during the design and verification phase of new systems.

#### 2.3.2.3 *HSPICE Toolbox*

Another method of extracting data form a Listing file is to use the HSPICE Toolbox. This is a free addition to MATLAB that does the same thing as the automatic formatting program created by me. The difference between the two programs lies in the fact that the toolbox is more complex, limited to specific cases, and lacks support. Using this tool, I was able to extract the data and save it as a MATLAB data file for later use.

## 2.4 Comparing the Results

The result of the previous steps is three data files. One was generated using the MATLAB simulator and the other was made using either manual formatting (section 2.3.2.1), automatic formatting (section 2.3.2.2) or using the HSPICE Toolbox (section 2.3.2.3). I chose to compare the data files obtained using MATLAB, my C++ program, and the HSPICE Toolbox. I plotted frequency versus the voltage, at node 2, for all three data files and placed them on the same graph. The resulting graph is illustrated in figure 2. The blue line is the result of the MATLAB simulation, the red line represents the outcome of using the HSPICE Toolbox, and the yellow dashed line is the result of using the data formatted by my C++ program. As illustrated, all three method obtained the same results.

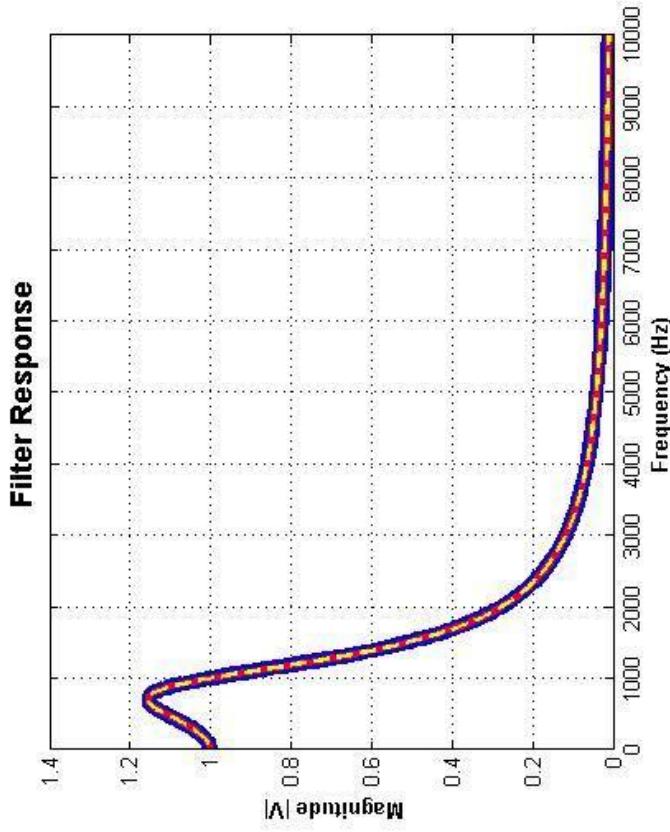


FIGURE 2: comparing the results of each method on the same graph

In conclusion, any of these three method is valid for completing simulations and extracting the obtained data. It is up to the user to choose which method to use based on the situation.

## 2.5 Writing a small handbook

The outcome of section 2.4 can be very useful for comparing the result obtained using different methods. This can prove helpful when multiple members of a project want to compare the results that they obtained independently. Being able to compare and collaborate on projects can prove beneficial for everyone and for this reason I was asked to write a small handbook on the steps to make these comparisons. The document was made using Latex and is a step by step guide for completing simulations, extracting the data, and plotting the result for comparison.

### 3 Reflection of Work Experience

#### 3.1 Relation to Academic Studies

As an electrical engineering student, it is extremely important to have a good understanding of circuits. The work done this work term, provided an excellent opportunity to become familiar with circuits and the different characteristic that they possess. Unfortunately, as a first year student I had not had the opportunity to become very familiar with circuits due to the general nature of the courses that I have taken so far. Although my courses lacked the application opportunity that this work term provided, they gave me a good set of skills to use. For instance taking ECOR 1606 taught me how to program using C++. My understanding of programming was limited but it was a good foundation to build upon during this work term. Creating the automatic data formatter (section 2.3.2.2) encouraged me to fine tune my programming skills by researching new syntax and capabilities of the C++ language. Another course that helped me during this work term was ECOR 2606. This class was based on MATLAB and, through the labs, it taught me to use MATLAB as a tool to solve mathematical problems. These skills included LU factorisation and a variety of MATLAB syntax. Once again this created a good foundation for me to use as I worked on the MNA program written in MATLAB. These two courses provided the basic skills, but they failed to show the application of these tools in real life situations. By participating in this work term, I got the opportunity to learn new things and to bring together the content I have learnt over the last year to create a solution for a real life problem.

#### 3.2 Career Development

The project completed for my first work term taught me how to use a number of industry standard programs. After this work term, I feel much more confident in using MATLAB which is a program used by engineers and mathematicians to solve complex mathematical problems. As a future engineer, knowing how to use this program can be very beneficial as it reduces my learning curve when working on projects. This can be desirable for many companies. Another important software that I have become comfortable using is HSPICE. This is a state-of-the-art program that allows for accurate circuit simulation. Being familiar with his program allows me to look for future positions that require experience with this piece of software. Like learning MATLAB, this opens the door to more job opportunities. I also had the opportunity to work on my C++ programming skills. This is very important as C++ is a popular and powerful programming language. Being able to write C++ programs allows me to bring an important skill into any project. As seen during this work term, C++ programs can be used to make tools to improve project efficiency by automatically completing tedious tasks. Becoming familiar with MATLAB, HSPICE, and C++ has added new tools to my skill set. When entering the work force in the future, I will have the ability to better contribute to my project teams by being familiar with the tools of the industry.

### 3.3 Overall Experience

This work term was a great opportunity to become familiar with my chosen program. It was a good way of getting a taste of the work ahead of me. Being familiar with my program gives me more confidence moving forward. I feel confident in choosing electrical engineering as my career path as I enjoyed the project that I completed. I know that the tools I have learnt over the last 10 weeks will open many doors when looking for careers or projects in the future. This work term was an excellent opportunity to build a bridge between my knowledge and its application in the real world. In the future I wish I had a better idea of what engineering concepts I was expected to know, so I could have had a chance to learn the concept on my own before the work term. This would allow me to spend more time learning and using the tools rather than focusing on learning new engineering concepts during the limited length of the work term. Over all, this was an excellent experience and because of it I feel greater purpose for everything I have and will learn during my university career.

## 4 Conclusion

I was handed a great opportunity as a first year student to learn and explore the career path that I have chosen. Over the last couple of weeks I have learnt new electrical engineering concepts such as Modified Nodal Analysis. I now understand how circuit simulators used this concept to analyses circuits. In addition to this, I am able to complete these simulations using my own MATLAB program or by using another well-known program called HSPICE. Once I obtain the results of these tools I am able to extract, format, and compare them. The comparison is done in a single MATLAB graph. To make this possible, I wrote my own C++ program to extract the HSPICE results and to format them in a way that MATLAB would recognise. For future reference, I documented the instruction in a small manual. This will help others to compare the simulation results that they obtained using different tools. I am grateful to have had the opportunity to learn such a great deal and to help other students in the future.

### 4.1 Future Work

This section will act as a note to myself or anyone wanting to continue on this work in the future. Like all pieces of software, the C++ program that I made can be further improved upon. The current program does not check for any errors in the simulation. It assumes that the results are error free. The next iteration could actually find the errors and inform the user. The way the program deals with data for more than 4 nodes can also be improved. When there are more than 4 nodes printed, HSPICE prints the values below the first 4 nodes. The current program cannot properly handle this possibility. In the future the code must be adjusted so that the program can read the data correctly no matter the number of nodes printed or the errors encountered.

## 5 List of Abbreviations

CAD	Computer Aided Design
MNA	Modified Nodal Analysis
VLSI	Very-Large-Scale Integration
KCL	Kirchhoff's Current Law

## 6 References

- [1] Carleton University, "Department of Electronics: Research Areas," 2013. [Online]. Available: <https://www.doe.carleton.ca/content/research-areas>. [Accessed 11 August 2014].
- [2] Carleton University, "Department of Electronics: Ram Achar," December 2012. [Online]. Available: <http://www.doe.carleton.ca/~achar/>. [Accessed 11 August 2014].
- [3] Carleton University, "About:Faculty of Engineering and Design," [Online]. Available: <http://carleton.ca/engineering-design/about/>. [Accessed 11 August 2014].
- [4] Carleton University, "DOE: About the Department," 2013. [Online]. Available: <https://www.doe.carleton.ca/content/about-department>. [Accessed 11 August 2014].
- [5] Carleton University, "Carleton University: About," [Online]. Available: <http://carleton.ca/about/>. [Accessed 11 August 2014].
- [6] University of Kentucky, "A Modified Nodal Analysis (MNA) Method," [Online]. Available: [http://www.engr.uky.edu/~gedney/courses/ee211/slides/MNA\\_Notes.pdf](http://www.engr.uky.edu/~gedney/courses/ee211/slides/MNA_Notes.pdf). [Accessed 11 August 2014].
- [7] "U of Guelph: Kirchhoff's Current Law," [Online]. Available: <http://www.physics.uoguelph.ca/tutorials/ohm/Q.ohm.KCL.html>. [Accessed 11 August 2014].
- [8] M. S. Nakha, "Computer methods for analysis and design of vlsi and communication," Carleton University, Ottawa, Canada, Lecture Notes ELEC-4506, 2013.
- [9] University Of Cambridge, "Numerical Methods: University of Cambridge," 13 May 2014. [Online]. Available: <http://www.cl.cam.ac.uk/teaching/1314/NumMethods/supporting/mcmaster-kirubaludecomp.pdf>. [Accessed 11 August 2014].