

CMOS Circuit SPICE Generator



spring 2019

the american university in cairo

Nada Badawy – Eman Asem

900171975 - 900172070

Contents

[Description 2](#_Toc8986105)

[Coding and assumptions 2](#_Toc8986106)

[Coding: 2](#_Toc8986107)

[Assumptions: 2](#_Toc8986108)

[Testing the program 2](#_Toc8986109)

[How to run and compile the program 2](#_Toc8986110)

[Contribution of each member: 3](#_Toc8986111)

# Description

The program generates the data statements part of the SPICE deck describing a digital CMOS circuit that realizes the Boolean expression entered by the user. This output only consists of a listing of interconnected NMOS transistors and PMOS transistors Each element statement has the following syntax:

Mname drain gate source body type

Moreover, after implementing the core part of the project which is the data section of a SPICE generator; we also add two parts of the bonus. The first feature that we enable the user to use brackets in the input Boolean expression to override original operator precedence. The second feature is that the code can generate more than Boolean expression at time.

# Coding and assumptions

## Coding:

The program begins by two options to the user to choose from. The first option is to write a single expression without any spaces. In this mood the user can write the expression with brackets as well. Then the second mood is to write multiple expressions separated by semicolon. This part is very sensitive for the spaces. So, the user should enter the expressions only separated by semicolon without any spaces.

To generate the data part of the spice generator we first read the Boolean expression from the user and put it in a string. Then coping the original string to vector of type string to deal with it easily in the further operations. After copping it the code will pass the vector to a function that traverse the vector and indicates the operation needed. We have three main functions which generates all the combinations needed. The AND, OR, NOT functions are the main skeleton for our generator. The function traverse the Boolean expression then indicates the operation needed then it passes the inputs and empty string to the output.

## Assumptions:

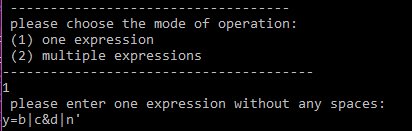
We made a specific assumption in order to get the proper format which will be translated by the program into the needed output. So, our assumptions are:

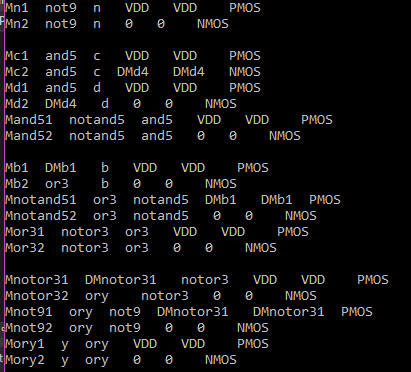
1. Spaces are not allowed in the input expressions
2. Parentheses are allowed in the input expressions in the first mode only
3. Multiple expressions are allowed but the user must not to put spaces after the expression “each expression are ended by “;” except the final one.
4. The described circuit has single or multiple semicolon-separated output symbol.
5. Symbols in the input expression are case-sensitive. X is a different symbol from x.
6. Assume that there is always a net called vdd which represents the positive terminal of an independent DC voltage source.

# Testing the program:

in order to test each feature in the program we will conduct a test for each feature separated and compares our results with a real SPICE generator.

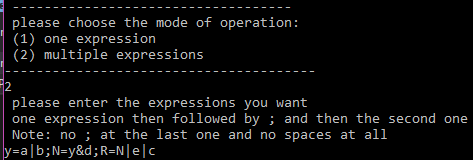
**First general Boolean expression:**

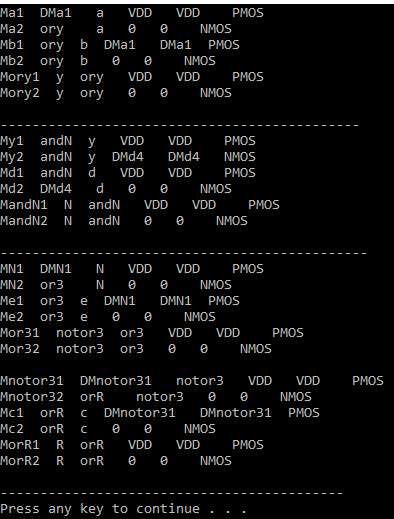




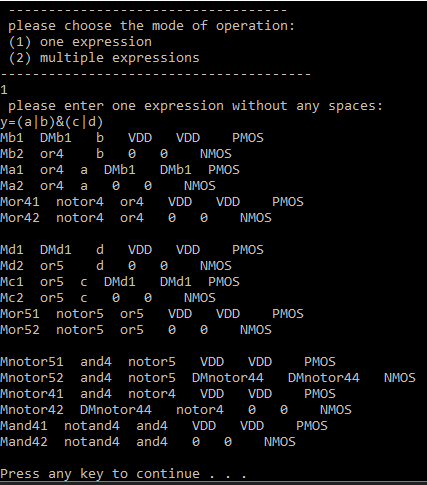
It starts with the negotiation and then the AND, and finally the OR with the write combinations.

**Second the multiple expressions:**





**Third the test for bracket:**



# How to run and compile the program

The program needs a specific format to be able to run in appropriate way. For instance, the required format should be written without spaces and symbols in the input expression are case-sensitive where X is a different symbol from x. Moreover, the user can input any Boolean expression using any symbols; however, the output symbol cannot be used as an input symbol. The &, |, and ’ operators are used for AND, OR, and NOT respectively. NOT has the highest precedence while OR has the least precedence. For instence:

**to run the program:**

* Step1: the program will ask the user to enter a string with the format explained before.
* Step2: the program will output the output string according to the user input

# Contribution of each member:

* The project consisted of 3 main parts:

1. Designing the program
2. Coding the program
3. Testing the program

* Firstly, we all contributed in designing the project. We sat together and came up with the most suitable design that would be efficient, and we decided the functions which will be needed and the structure of the code.
* Secondly, coding the program was a collaborative work. Each member was responsible for coding a part of the program:
  + Nada was responsible for making the function of OR, NOT and AND functions.
  + For the bonus features, we both did the first bonus feature which is allowing multiple semicolon-separated output symbol. In addition to the second bonus feature which is allowing parentheses in the input expressions
  + Eman was responsible for reading the string from the user and divide it to be able to be passed to the function.
* Testing and debugging: both members contributed in the testing of the program. We started by trying different inputs and applied various functions on them to confirm that all the program’s functions work properly.