# HSPICE Tutorial
# University of Missouri-Kansas City

Marouf Khan

## 1 Introduction

This tutorial was created for students taking VLSI courses at UMKC. It will outline how to run HSPICE installed on a linux server, how to create and simulate netlists and how to view the results using CosmosScope. This document is not meant to be a comprehensive user guide on HSPICE. It is meant to be an introductory tutorial.

SPICE is an acronym which stands for ***S***imulation ***P***rogram with ***I***ntegrated ***C***ircuit ***E***mphasis. Many different EDA vendors have their own versions of SPICE. We will be using HSPICE, a tool from Synopsys.

## 2 Running HSPICE

Log onto any SCE lab computer using your SSO. Then click on *Start* → *Programs* → *Specialized Software* → *Cadence*. Enter your password when prompted. This will bring up the terminal window. You will type in HSPICE commands in this window.

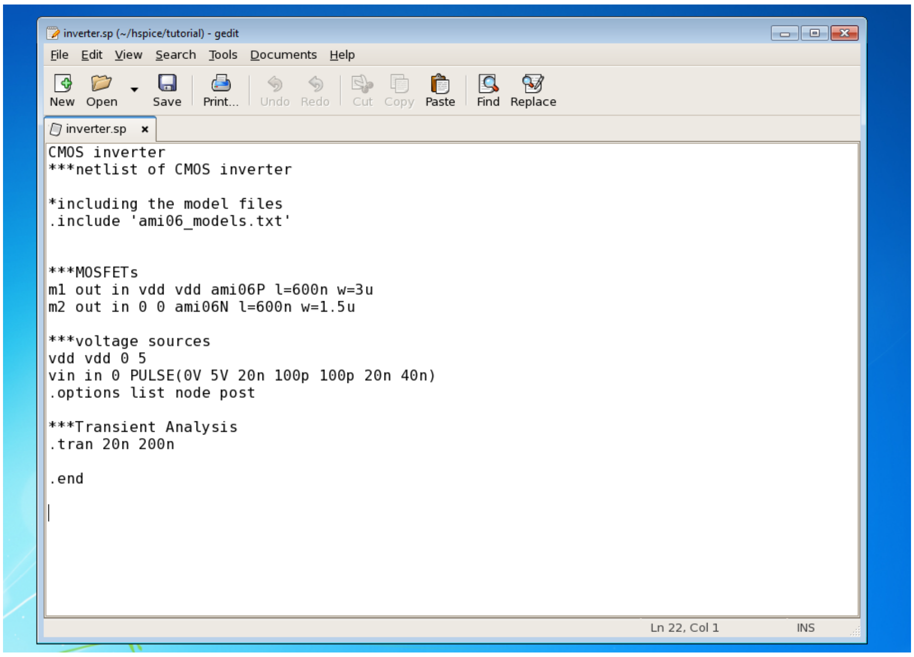
HSPICE takes the **netlist** of a circuit as its input. This file is usually a text file with the extension **'.sp'**. The first step is to create this file using any text editor of your choice. For this tutorial we will use **gedit**.

In the following commands we make a directory called hspice to save your HSPICE files and then we navigate to it. We then launch **gedit** to type our netlist (the dollar sign is part of the prompt):

```
$ mkdir hspice
$ cd hspice
$ gedit &
```

This is will open up the gedit text editor. Type up the spice netlist as given below:

```
CMOS inverter
*netlist of CMOS inverter

*including the model files
.include ‘ami06_models.txt’

*MOSFETs
m1 out in vdd vdd ami06P l=600n w=3u
m2 out in 0 0 ami06N l=600n w=1.5u

*voltage sources
vdd vdd 0 5
vin in 0 PULSE(0V 5V 20n 100p 100p 20n 40n)

*always include this line.It specifies the output files
.options list node post

*Transient Analysis
.tran 20n 200n
.end
```


inverter.sp (~/hspice/tutorial) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
inverter.sp
Ln 22, Col 1
INS

```
CMOS inverter
***netlist of CMOS inverter

*including the model files
.include 'ami06_models.txt'


***MOSFETs
m1 out in vdd vdd ami06P l=600n w=3u
m2 out in 0 0 ami06N l=600n w=1.5u

***voltage sources
vdd vdd 0 5
vin in 0 PULSE(0V 5V 20n 100p 100p 20n 40n)
.options list node post

***Transient Analysis
.tran 20n 200n

.end
```

Figure 1: Inverter netlist

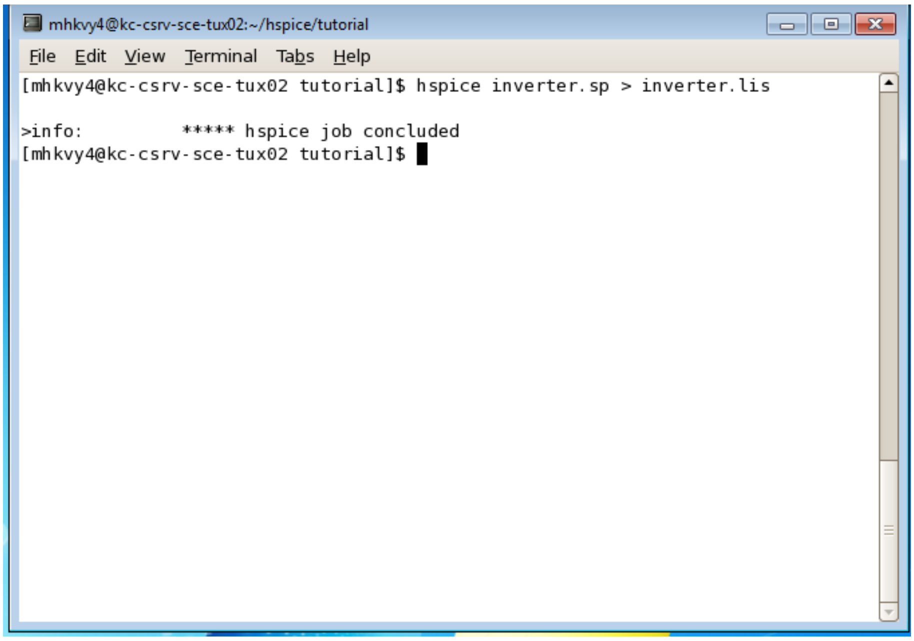
Save your file as ***inverter.sp***. In the above netlist all the lines beginning with '∗' are comments. The first line is just the title and is ignored by the simulator. Check the appendix for the syntax needed to write netlists.

Before you can run HSPICE you need to include the model files for both MOSFETs. Open the **'ami06_models.txt'** file provided on Blackboard. Launch **gedit** again from the terminal window and copy-paste the model file. Save this file as 'ami06_models.txt'. Your HSPICE simulation will need this model file to run.

Next we will run HSPICE to perform a transient analysis on our inverter circuit. Type in the following command.

```
$ hspice inverter.sp > inverter.lis
```

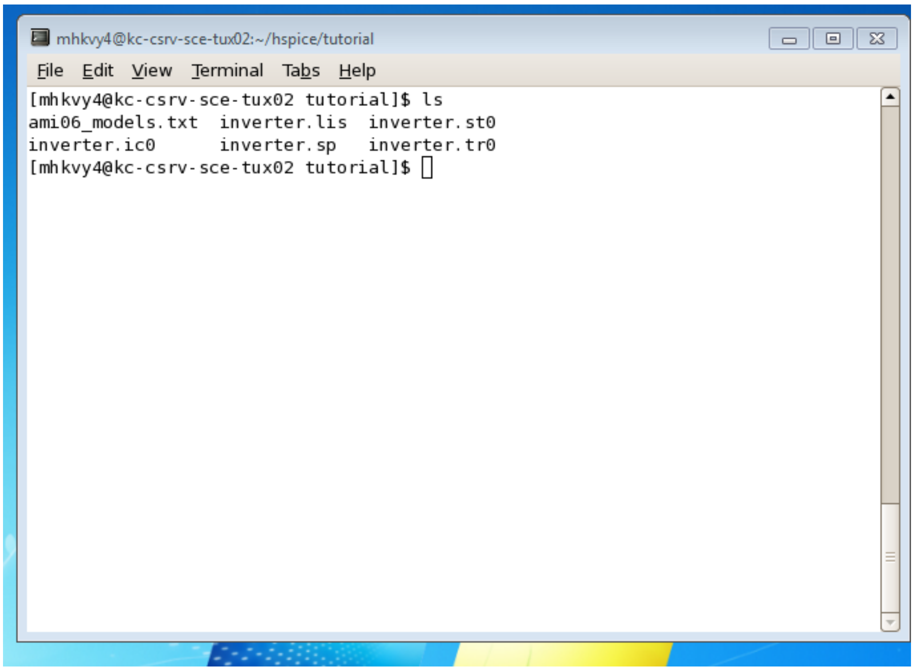
This command does the actual simulation. The syntax for the command is *hspice 'input_file_name.sp'* > *'output_file_name.lis'*. The output file above is ***inverter.lis*** and it is a log file of sorts for our simulation. It is useful for printing results and troubleshooting if HSPICE reports an error. The following screen shot represents a successful simulation.


mhkvy4@kc-csrv-sce-tux02:~/hspice/tutorial
File Edit View Terminal Tabs Help

```
[mhkvy4@kc-csrv-sce-tux02 tutorial]$ hspice inverter.sp > inverter.lis

>info:          ***** hspice job concluded
[mhkvy4@kc-csrv-sce-tux02 tutorial]$ 
```

Figure 2: HSPICE simulation

Now type in the **ls** in your terminal. **ls** will list the files in your directory.



mhkvy4@kc-csrv-sce-tux02:~/hspice/tutorial
File Edit View Terminal Tabs Help

```
[mhkvy4@kc-csrv-sce-tux02 tutorial]$ ls
ami06_models.txt  inverter.lis  inverter.st0
inverter.ic0      inverter.sp   inverter.tr0
[mhkvy4@kc-csrv-sce-tux02 tutorial]$ 
```

Figure 3: HSPICE files

Along with the models file, the input file and the output file, you will see three other output files that our HSPICE example generates. Depending upon the analysis performed, HSPICE generates different result files.

The following table provides a summary of common types ( The # symbol will be an integer and depends on the number of sweeps or iterations. Default for one sweep is zero. It is not a complete list of all output files generated from different analyses):

| **Output File Type** | **Extension** |
|---|---|
| Output listing | *.lis*, or user-specified |
| Transient analysis Plot file | *.tr#* (requires **.options post** line) |
| DC analysis Plot file | *.sw#* (requires **.options post** line) |
| AC analysis Plot file | *.ac#* (requires **.options post** line) |
| Output Status | *.st#* |
| Operating point node voltages (initial conditions) | *.ic#* |

Table 1: HSPICE output files and results

The file we need to see the waveforms for our transient analysis is the ***inverter.tr0***

## 3 Using CosmosScope

The wave viewer we will be using to look at our output wave forms is CosmosScope. It is another tool from Synopsys. CosmosScope is used to view the plot files mentioned in the above table. We will use it to open the ***inverter.tr0*** file. Type the following command in your terminal window.

```
$ cscope &
```

This will launch CosmosScope. Click on *File* → *Open* → *Plotfiles*. Select ***inverter.tr0***. Select the signals **v(in)** and **v(out)**. The screenshot below shows the results you should see.

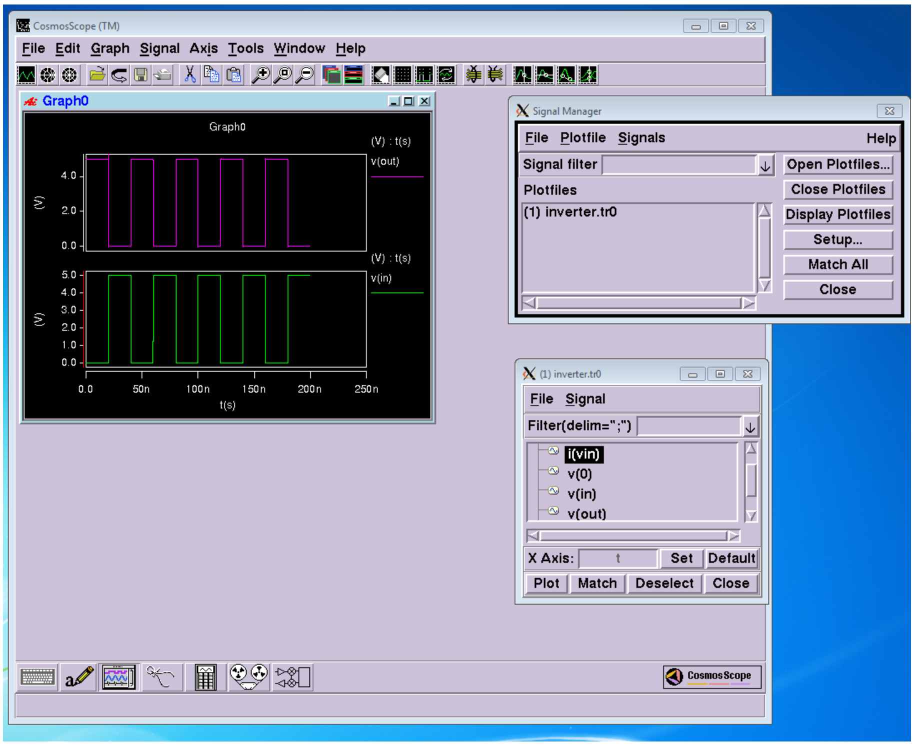


Figure 4: CosmosScope

This is the end of this basic tutorial. The goal was to familiarize students with HSPICE and CosmosScope. The appendix will now discuss some of the syntactic features of SPICE. The appendix is based on a SPICE tutorial by Jonathan Roderick, Onder Oz and Tyler Rather at USC.

# Appendices

## A Circuit Element Statements

The SPICE netlist describes how each circuit element is connected in a circuit. It is usually helpful to sketch a schematic before attempting to write a netlist. The element statement contains the element name, the nodes which tell its location, and the physical characteristics of an element.

The first letter in the element statement identifies what type of element the statement describes. A user can give any name to an element. The next two (for two terminal device), three (for three terminal device), or four (for four terminal device) characters represent the nodes where an element is connected to in a circuit. The last part contains the physical model. For example, the line below describes a 10KΩ resistor **R1** between the nodes 0 and 1.

```
R1 0 1 10K
```

The table below gives the device letter abbreviations used to identify them in SPICE.

| First Letter | Element |
|---|---|
| C | Capacitor |
| D | Diode |
| E | Voltage-controlled voltage source (VCVS) |
| F | Current-controlled current source (CCCS) |
| G | Voltage-controlled current source (VCCS) |
| H | Current-controlled voltage source (CCVS) |
| I | Independent current source |
| L | Inductor |
| M | MOSFET |
| Q | BJT |
| R | Resistor |
| V | Independent voltage source |

Table 2: SPICE device letter abbreviations

The following two tables show the scaling and unit abbreviations used in SPICE.

| Spice Abbreviation | Metric Prefix |
|---|---|
| T | tera |
| G | giga |
| Meg | mega |
| K | kilo |
| M | mili |
| U | micro |
| N | nano |
| P | pico |
| F | femto |

Table 3: SPICE scaling abbreviations

| Spice Abbreviation | Units |
|---|---|
| A | amps |
| Degree | Degree |
| F | Farad |
| H | Henry |
| Hz | Hertz |
| Ohm | Ohms(Ω) |
| V | Volts |

Table 4: SPICE Unit abbreviations

## A.1 Active and Passive device elements

This section gives you the syntax for some of the more commonly used active and passive devices. All the bracket labels represent user defined names, node numbers and values.

- Resistor
  `r<name> <terminal 1> <terminal 2> <value>`

- Inductor
  `i<name> <terminal 1> <terminal 2> <value>`

- Capacitor
  `c<name> <terminal 1> <terminal 2> <value>`
- Model (type can be NMOS, PMOS, NPN, PNP, or for Diode)
  `.model <name> <type> (<parameter list>)`
- Diode
  `d<name> <+ terminal> <- terminal> <model> <parameter list>`
- BJT
  `q<name> <collector> <base> <emitter> <model> <parameter list>`
- MOSFET (W and L values are placed in the parameter list)
  `m<name> <drain> <gate> <source> <body> <model> <parameter list>`

## A.2 Voltage and Current sources

This section list the syntax for some of commonly used voltage and current sources. All the bracket labels represent user defined names, node numbers and values.

- Independent Current source
  `i<name> <+ terminal> <- terminal> <value>`
- Independent voltage source
  `v<name> <+ terminal> <- terminal> <value>`
- Sinusoidal source (used as a <value>)
  `sin(<offset> <amplitude> <frequency> <delay> <damping> <phase>)`
- Square wave source (used as a <value>)
  `pulse(<vmin> <vmax> <delay> <rise time> <fall time> <pulse width> <period>)`
- Piece-wise linear source (used as a <value>)
  `pwl(<t0> <v0> <t1> <v1> <t2> <v2> ...)`

# B Circuit Analysis Statements

This section list the syntax for some of the analysis you will perform on your circuits.

- AC analysis (pick either lin, dec or oct scale)
  ```
  .ac <lin|dec|oct> <number of samples> <freq start> <freq stop>
  ```
- DC analysis
  ```
  .dc <source> <start> <stop> <step>
  ```
- Transient analysis
  ```
  .tran <t step> <t stop>
  ```