CMOS VLSI Lab

Instructor: Dr. Srinivas Katkoori

Handout on SPICE Circuit Simulator

SPICE (Simulation Program for Integrated Circuits Emphasis) is a powerful circuit simulator that is widely used in industry to verify circuit designs and to predict the circuit behavior. SPICE was originally developed in 1975 at University of California, Berkeley. Many versions of SPICE exist some of which are: PSPICE (a PC version of SPICE), HSPICE (a popular industry strength version), and Berkeley SPICE (that is continuously under development at Univ. of California, Berkeley). The primary reason why SPICE is so popular is that it mimics the circuit behavior accurately (within 10-15% range) compared to the real implementation.

SPICE is run from the command line (using the HSPICE simulator), accepting a netlist describing the circuit to be simulated. All circuit designs in this class will be verified by running the SPICE simulator on the circuits extracted from the layouts. This handout explains how to use HSPICE for circuit simulations when an input netlist is available.

The input netlist provided to HSPICE is a text file describing the interconnections among the circuit elements (transistors, resistors, capacitors etc.), the device models used for the circuit elements, and the input voltages/ signals applied to the circuit. The transistor models are in the "*mosistsmc180.sp.txt*" file.

A typical SPICE netlist for an inverter gate in Fig 2 could be written as:

```
**---------------------------------------------------------
 *         Inverter SPICE deck
**---------------------------------------------------------
 *         Parameters and models
**---------------------------------------------------------
 .include mosistsmc180.sp.txt
 .options post list scale=1n
**---------------------------------------------------------
 *         Simulation netlist
**---------------------------------------------------------
 Vdd Vdd 0 1.8V
 Vin in  0 PULSE 0 1.8V 0.5ns 0.1ns 0.1ns 10ns 20ns
 m0  out in Vdd Vdd PMOS W=360 L=180
 m1  out in  0   0  NMOS W=180 L=180
 Cload out 0 0.01pF
**---------------------------------------------------------
*        Stimulus and plot statements
**---------------------------------------------------------
 .tran   1n  40n
 .plot   V(in) V(out)
 .end
```

Fig. 1: SPICE description of a CMOS Inverter

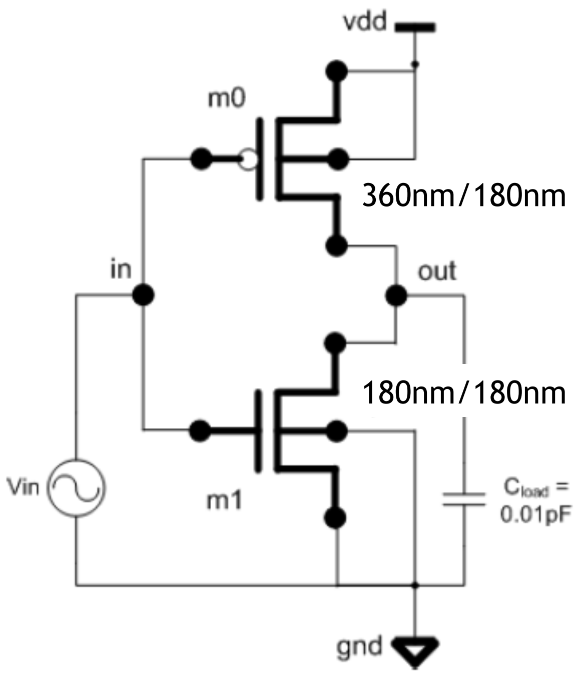

Fig.2: Transistor Schematic of a CMOS Inverter

SPICE Input Description

A SPICE input file consists of three main parts:

1. **Data statements**: describe the components and the interconnections of the circuit simulated. (for example, lines 11-15 in Fig.1).
2. **Control statements**: specify the type of analysis performed on the circuit (for example, lines 19-23 in Fig.1).
3. **Output statements**: specifies what outputs are to be printed or plotted (for example, line 25 in Fig.1).

These statements can appear in any order, but the above sequence is most commonly used. Two other statements are required: the **title statement** and the **end statement**. The title statement is the first line and can contain any information, while the end statement is always **.end**. This statement must be a line by itself, followed *by a carriage return*!
In addition, **comment statements**, can be included in a SPICE input file. Comment lines always begin with an asterisk (*) and are ignored by SPICE. The first statement in a SPICE netlist should always be a comment statement.

A typical SPICE input file has the following structure:

```
Title Statement
Element Statements
.
.
.
Command/Control Statements
Output Statements
.end <CR>
```

Format: The statements have a free format and consist of fields separated by a blank. If one wants to continue a statement to the next line, one uses a "+" sign (continuation sign) at the beginning of the next line.

The Spice input file shown in Fig.1 models the inverter shown in Fig. 2. The first line typically contains a brief description of the circuit.

Data Statements and Component Specification

(1) Independent DC Sources

Voltage source: `Vname N1 N2 Type Value`
Current source: `Iname N1 N2 Type Value`

where,

N1 is the positive terminal node,
N2 is the negative terminal node,
Type can be DC, AC or TRAN, depending on the type of analysis performed,
Value gives the value of the source.

The positive current direction through the *current or voltage source* is from the positive (N1) node to the negative (N2) node. ***Note***: the name of a voltage and current source must start with V and I, respectively.

Examples:

```
V1 in gnd DC 2.5    *** DC voltage source

Isrc 1 5 DC 1.5     *** DC current source
```

In Fig.1, lines 11 and 12 describe independent voltage sources. The voltage source type can be omitted if it is a constant DC voltage source (for example, line 11 in Fig.1).

(2) Transistors : The general format of a MOS transistor description is:

```
m[name] Drain  Gate  Source  Substrate  ModelName  W=  L=
```

The MOS transistor name (`m[name]`) has to start with an `m`; `Drain, Gate, Source` and `Substrate` are the node names of the Drain, Gate, Source and Substrate terminals, respectively. By default, the node number 0 represents the GND node. `ModelName` is the name of the transistor model. L and W are the length and width of the gate (in meters). Lines 13 and 14 in Fig.1 are examples of transistor descriptions.

(3) Resistors and Capacitors: The general format for resistor and capacitors are:

```
Resistor:  Rname N1 N2 ResistanceValue  < IC >
Capacitor: Cname N1 N2 CapacitanceValue < IC >
```

where N1 is the positive node, N2 is the negative node. IC is the initial condition (DC voltage or current). The symbol <> means that the field is optional. If not specified, it is assumed to be zero.
Line 15 in Fig.1 describes a capacitor.

(4) Piecewise linear voltage source (PWL)

```
Vname N1 N2 PWL(T1 V1 T2 V2 T3 V3 ...)
```

in which (Ti Vi) specifies the voltage value Vi of the source at time Ti

Example:

```
    Vin in gnd PWL(0 0 10ns 5 100ns 5 110ns 0)
```

(5) Pulse voltage source (PULSE)

The general format of a PULSE description (see Figure2) is:

```
Vname N1 N2 PULSE V1 V2 TD Tr Tf PW Period
```

where V1 - initial voltage; V2 - peak voltage; TD - initial delay time; Tr - rise time; Tf - fall time; PW - pulse-width; and Period - total time period. See Figure.3 for an illustration of these parameters. Line 12 in Fig.1 is an example of a Pulse voltage source.

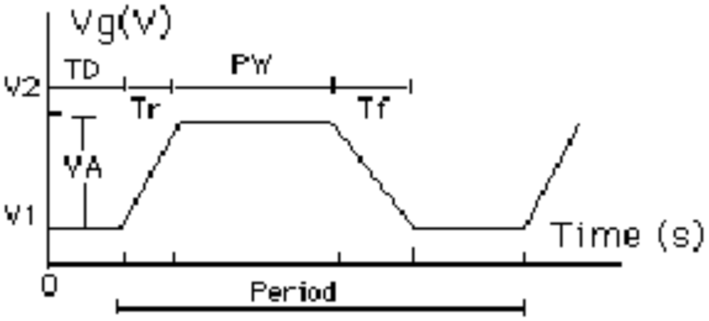

Fig.3: Pulse Waveform

Control Statements in SPICE

SPICE provides a large number of control statements to specify the type of analysis to be performed. The commands that you will most frequently use in your assignments are described below.

(1) .TRAN Statement

This statement specifies the time interval over which the transient analysis takes place, and the time increments. The format is as follows:

```
    .tran TSTEP TSTOP <TSTART <TMAX>> <UIC>
```

where,

TSTEP is the printing increment.
TSTOP is the final time
TSTART (optional) is the starting time (if omitted, TSTART is assumed to be zero)
TMAX (optional) is the maximum step size.
UIC (optional) stands for Use Initial Condition and instructs SPICE not to do the quiescent operating point before beginning the transient analysis. If UIC is specified, SPICE will use the initial conditions specified in the element statements (see data statement) IC = value.

For example, in Fig.1, the command `.tran` in line 19 asks the SPICE to perform a transient analysis from 0ns to 10ns in increments of 0.1 nanoseconds.

(2) .DC Statement

This statement allows you to increment (sweep) an independent source over a certain range with a specified step. The format is as follows:

```
.DC SRCname START STOP STEP
```

in which `SRCname` is the name of the source you want to vary; START and STOP are the starting and ending value, respectively; and STEP is the size of the increment.

Example: `.DC Vin 0 10 0.1`

When the `START` and `STOP` values are identical (and the `STEP` is non-zero), the .DC command produces only one value. This may be useful in HSpice when you do not want all the DC voltages and currents to be printed, but are interested in a limited number of DC voltages and currents.

You can nest the DC sweep command which is often used to plot transistor characteristics, such as the Drain current I_{DS} versus the Drain-Source voltage, Vds for different gate voltages Vgs. This can be done as follows:

```
.DC SRCname1 START STOP STEP SRCname2 START STOP STEP
```

Example: `.DC Vds 0 5 0.5 Vgs 0 5 1`

In the example above, the voltage Vds will be swept from 0 to 5V in steps of 1V for every value of Vgs.

(3) ***.OPTIONS*** Statement

• Control options are set in .options statement. Any number of options can be set in one statement.
• If there is more than one .options statement the settings of the last one are taken.
Format:

```
.OPTIONS opt1 opt2…
```

Example:

```
.OPTIONS GMINDC=1.0000E-12 plot
```

.OPTION KEYWORDS (i.e opt1…)
General .OPTION Keywords

- LIST (use this in your .options statement): Produces an element summary listing of the input data to be printed.
- NODE (use this in your .options statement): Causes a node cross reference table to be printed. The table lists each node and all elements connected to it.
- MEASOUT Outputs: Measure statement values and sweep values into the <design>.mt# file; where # depends on the run specified by the .ALTER statement. If .ALTER is not used then the file is <design>.mt0
- PROBE: By default Hspice reports all node voltages and currents. The PROBE statement in .options along with the .PROBE / .PLOT/ .PRINT / .GRAPH statements restricts the recorded values to those specified in these statements.
- POST: Enables saving of results to be later analyzed using Spice explorer.

In Fig.1, line 7 specifies various options that we would like to use for simulation. The argument "LIST" produces an element summary of list of the input data to be printed. The argument "NODE" causes a node cross-reference table to be printed. The argument "POST" enables storing of simulation results for analysis.

(4) .MEASURE statement (USER DEFINED ANALYSIS)

The .MEASURE statement is used to print ***user specified*** electrical specifications. It can be used for DC, AC and Transient analysis. Used to measure delay, power and other such parameters over the data points produced as a result of transient analysis. Result of measurement placed in **.mt0** file.

Delay Measurement
* Calculate delay between two events occurring during transient analysis
Syntax:
.**measure tran** result
+ **trig** trig_var **val**=trig_val <**td**=delay> <**rise/fall**=n>
+ **targ** targ_var **val**=targ_val <**td**=delay> <**rise/fall**=n>

where,
- trig_var – variable from which measurement will start
- targ_var – variable at which measurement will end
- trig_val – value of trig_var at which measurement will start
- targ_val – value of targ_var at which measurement will end
- delay – time that must elapse before starting (or stopping) measurement
- n – number of rising (or falling) transitions before starting (or stopping) measurement

Example:

```
.measure tran tr TRIG v(out) val=0.1 RISE=1 TARG  v(out) val=0.9 RISE=1
```

In this example, the `.measure` statement in lines specifies the *rise time* of `v(out)`.

Power calculation
Syntax:
.**measure tran** result **AVG POWER** from=start_time to=end_time
- Returns the average power dissipated in the circuit during the transient analysis between simulation times start_time and end_time.

Example:

```
.measure tran avgpow AVG POWER from=1ns to=10ns
```

(5) .END statement

Every input file must end with a .END statement.
A carriage return after .END is also required.
Format:
.END

Output Statements in SPICE

(1) .Print and .Plot Statements

These statements are used to specify output to be generated by SPICE. If no outputs are specified in the input file, SPICE will always calculate the DC operating points. The two types of output statements are `.print` and `.plot`. A print is a table of data points and a plot is a graphical representation. The formats for these are as follows.

```
.PRINT TYPE OV1 OV2 OV3 ...
.PLOT  TYPE OV1 OV2 OV3 …
```

where TYPE specifies the type of analysis to be printed or plotted and can be DC, AC, or TRAN. The output variables are OV1, OV2 and can be voltage or currents in voltage sources.

Examples:

```
        .PLOT DC V(in) V(out) I(Vmeas)
        .PRINT TRAN V(in) I(Vmeas)
```

<u>Running HSPICE from the command line:</u>

If you wish to run HSPICE from the command line, then type:

```
grad%  hspice inv.sp > inv.lis
```

When the job finishes, HSPICE displays:

```
>info:                    ***** hspice job concluded
```

When you are done with the simulation, you can check for correct operation of the circuit by looking at the output waveforms. Spice Explorer is a waveform viewer that can be invoked by typing “sx” at the command line. You should invoke it in the invsp.run1 directory, i.e., the same directory in which .lis file resides.

```
grad%  sx
```

You can now plot the required signal on the spice explorer window. You can analyze the results by viewing the "**inv.lis**" and "**inv.st0**" files in your favorite text editor!

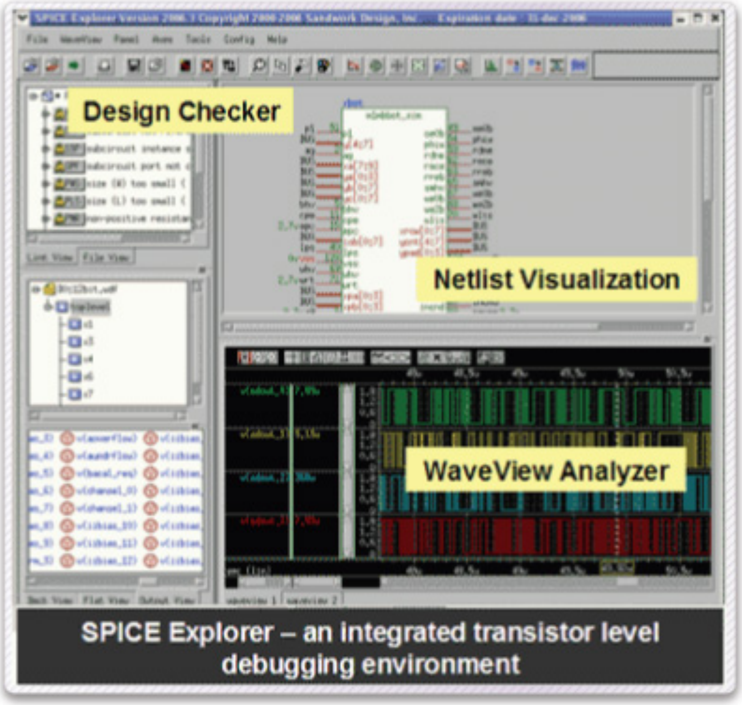
Design Checker
Netlist Visualization
WaveView Analyzer
SPICE Explorer – an integrated transistor level debugging environment

Fig.4: Spice Explorer Window

When you run HSPICE on the input file, the following new files will be in your directory: inv.lis, inv.ic, inv.st0, and inv.mt0. Familiarize yourself with the contents of these files.

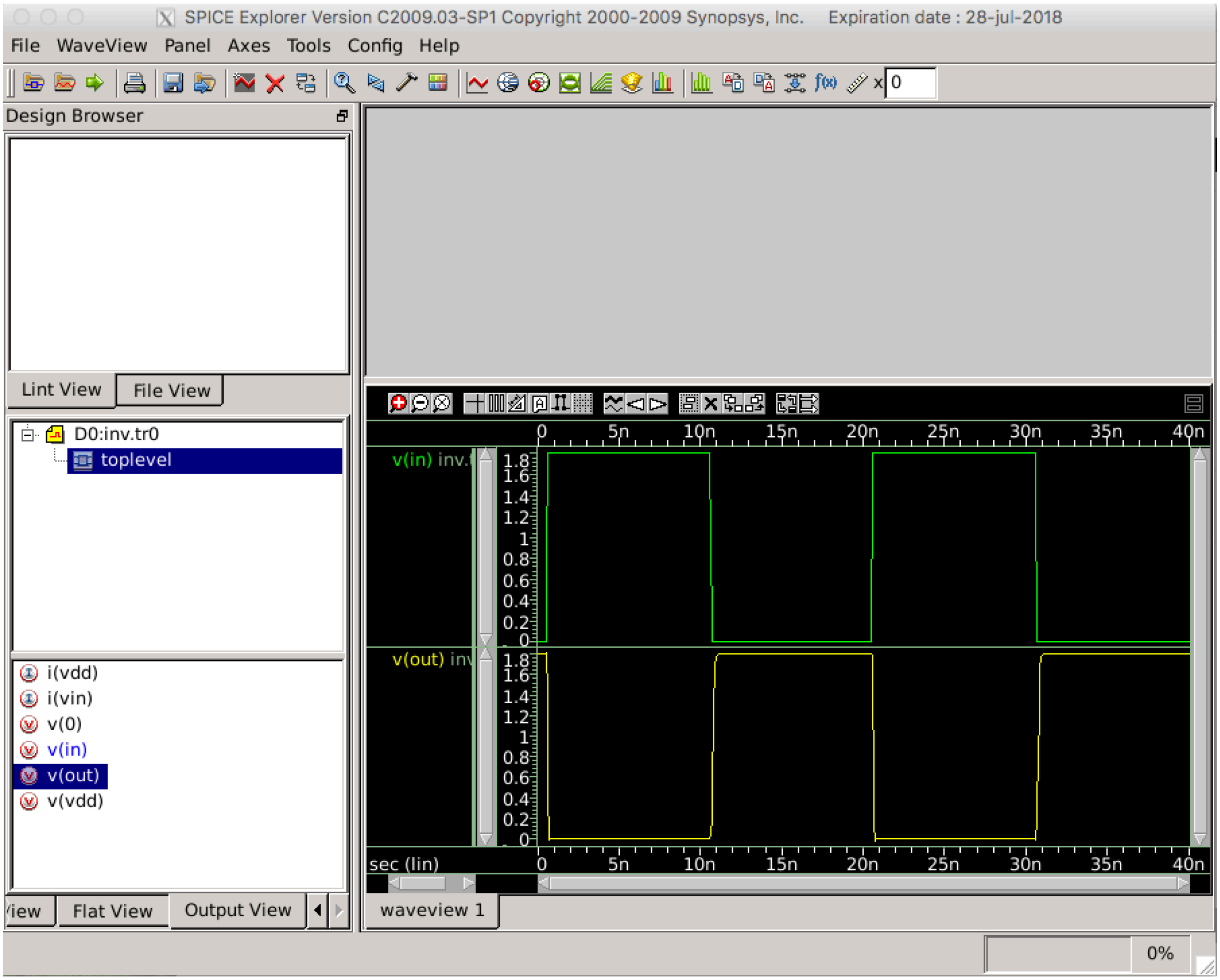

Fig.5: Inverter waveforms

Working with the SPICE EXPLORER waveform viewer:

1. **Importing Spice Netlist**. In the SPICE EXPLORER window, select File → Import Spice Netlist → Filename.sp. This will list the input file "inv.sp" in the dialog box. Press Ok.
2. **Importing Waveform File**. In the SPICE EXPLORER window, select File → Import Waveform File → Filename.lis. This will list the input file "inv.lis" in the dialog box. Press Ok. Now the current waveform file is inv.lis.
3. **Importing Netlist and Waveform through command line option**.

```
grad% sx -s inv.sp inv.lis
```

4. **Viewing Signal List**. In the Design Browser click on Output View. Click on the Top level of the waveform file. This will list down the signals present in the design.
5. **Viewing Waves**. Left click on the signal names and drag them into the waveview on the right panel of the Spice explorer. This will add the signals in the wave viewer.

6. **Dumping Screen to Image**. Right Click on the Waveview. Click on the Dump Screen option. Select image format in the save screen dump options menu.
7. **Familiarize** yourself with various options available over the Wave View. These will help in debugging the signal waveforms. The buttons are. Zoom-in, Zoom-out, Unzoom, Add Cursor, Monitor, Dynamic meter, Text Remark, Data Point, Grid, Signal Mode, Select all, Scan next, Delete signal, group panel, ungroup panel, dock/undock etc.

A good resource for SPICE:
`http://www.seas.upenn.edu/~jan/spice/spice.overview.html`