



Spectre Basics (IEC Lab – 1)

V0 : 06/09/2011

What is Spice ?

- Simulation Program with Integrated Circuit Emphasis
- Simulation → Act of imitating behavior of situation.
- Announced in 1973 by Professor Donald O. Pederson of the University of California, Berkeley.
- Simulating a circuit with SPICE is the industry-wide standard way to verify operation at the transistor level before manufacturing an Integrated Circuit.
- Simulating a circuit involves using mathematical models to replicate its behaviour.
- Impact of spice
 - Within a few years, SPICE had achieved acceptance at almost all electrical engineering schools
 - SPICE became an industry standard without the usual help of standards bodies, committees, meetings, position papers, and bureaucracy.

Need For Circuit Simulator

- The process of making an IC is expensive, time taking, and probing the behaviour of internal signals is difficult. Circuit Simulator checks the integrity of circuits and predicts how they will behave before anything is manufactured.
- Simulation can determine problems in the design early on, leading to a better final design and significant cost savings. Almost all IC design relies heavily on simulation.
- IC designers aren't the only ones using though. It is part of engineering curricula to help teach students the fundamentals of circuit design.
- For highly non linear complex circuits, it is very difficult to solve the equations exactly. So use approximate model to design the circuit then verify with the simulator.

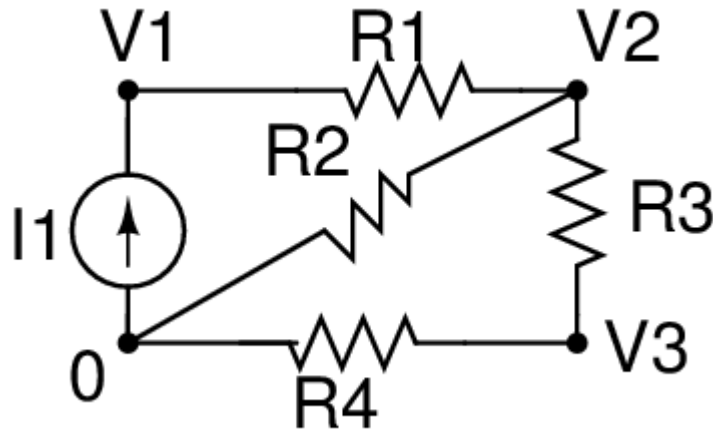
Circuit Simulator

- Ngspice, LT spice, Pspice, Hspice, Eldo and Spectre .
- We will learn Spectre simulator in this course.
- Cadence was able to make Spectre 3-5 times faster than traditional versions of SPICE, while improving its accuracy and reliability.
- We need to describe the circuit in a way simulator understands using basic constructs, keywords and so on.
- Netlist is an interconnection of *primitive circuit elements*.
- *primitive circuit elements* include MOS devices, BJT's, Diodes, resistor, caps and so on..
- To understand the behavior of the circuit / system, we need to analyze the circuit in transient, which is more close to real time scenario.
- The best way to understand / design the system is looking from DC, ac (small signal) and transient response of the system one after the other.

Different Analysis's in Spectre

- Different analysis available in spectre like ac, dc, transientnoise, stb, pz, , pss, pac, pnoise .
- We will briefly go through the following
 - DC analysis: Configure all independent sources to be constant valued and replace all inductors with shorts and capacitors with open for DC analysis.
 - Linear DC analysis : For linear circuits
 - Non Linear DC analysis : For non linear circuits
 - AC analysis :
 - Spectre calculates the operating point of the circuit using DC analysis and replaces all non linear devices with small signal model. Solves set of equations.
 - Solutions computed by the AC analysis contain only sinusoids at the same frequency as the input signal. Thus, each signal is represented with only two numbers, one that gives the magnitude, and another that gives the phase.
 - Transient analysis

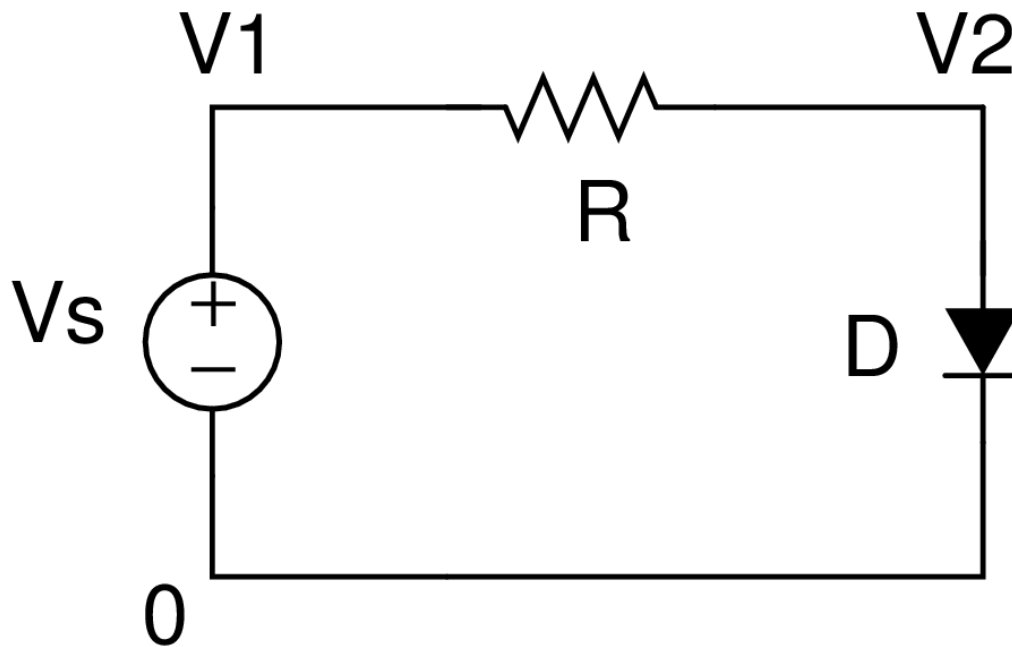
Linear DC Analysis



- Apply Nodal analysis.
- Using matrix methods, simulator can solve for different node voltages.

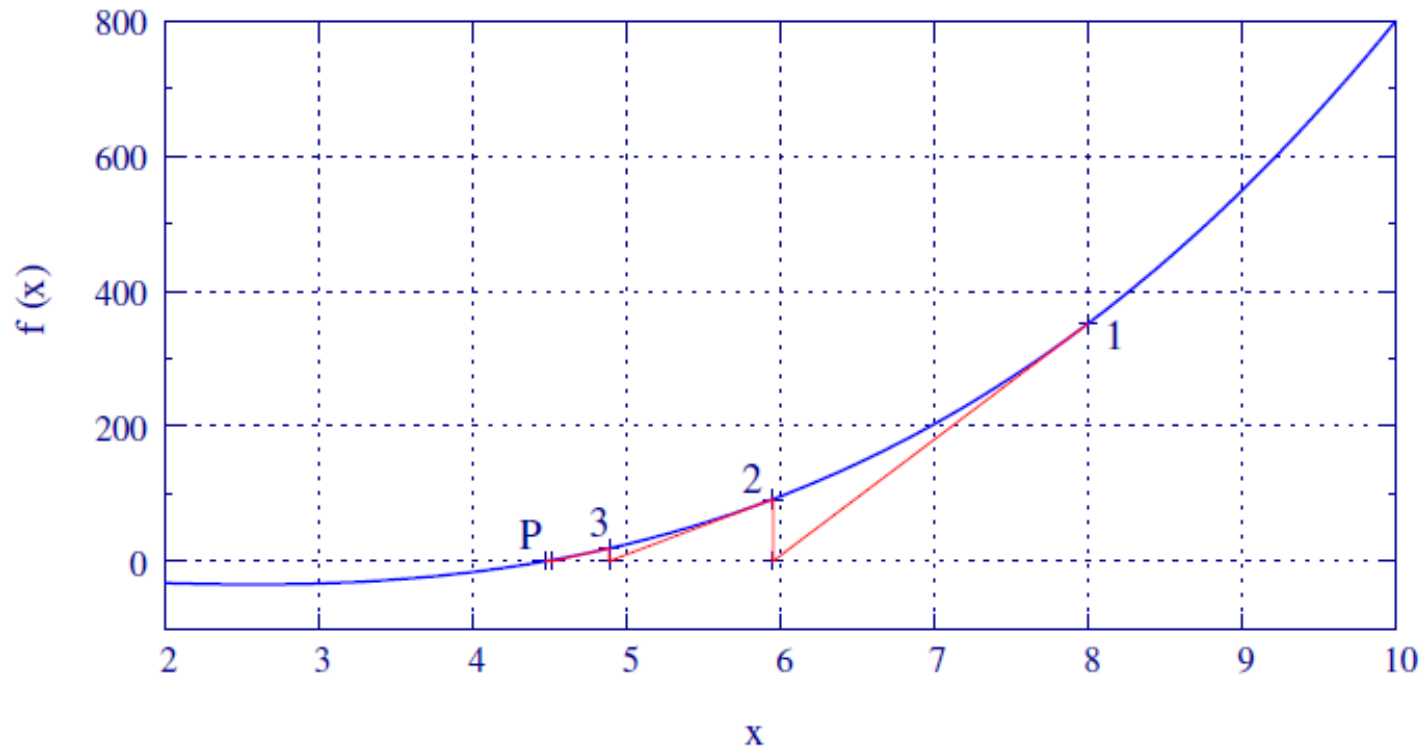
$$\begin{bmatrix} G1 & -G1 & 0 \\ -G1 & G1+G2+G3 & -G3 \\ 0 & G3+G4 & 0 \end{bmatrix} \begin{bmatrix} V1 \\ V2 \\ V3 \end{bmatrix} = \begin{bmatrix} I1 \\ 0 \\ 0 \end{bmatrix}$$

Non Linear DC Analysis



- Apply Nodal analysis. $\frac{V_s - V_2}{R} - I_0(e^{\frac{V_2}{V_T}} - 1) = 0$
- System of nonlinear equations, solve using Newton Raphson method.

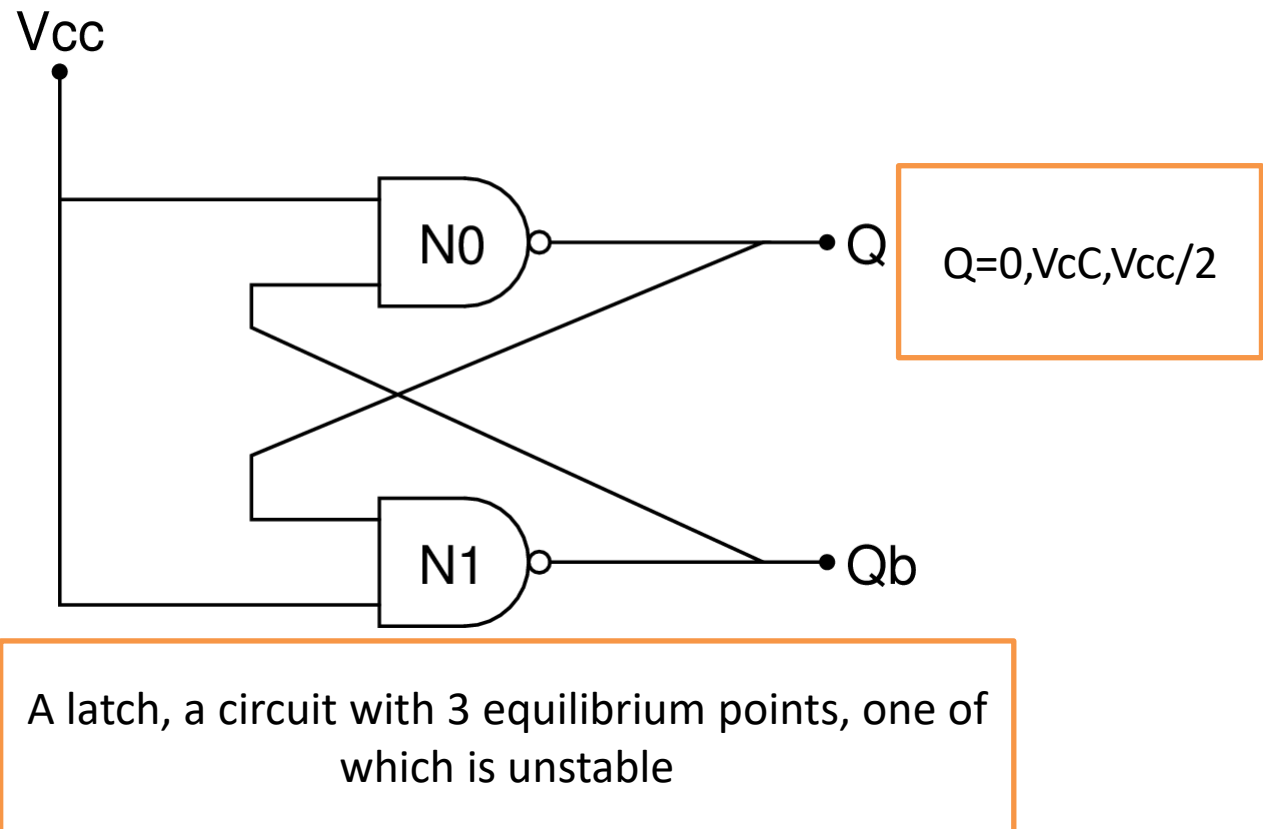
Non Linear DC Analysis - 2



Solution of $x^3 - 20x = 0$ with $x = 8$ as a initial guess.

DC Analysis - Stability

- It is important to understand that:
 1. Circuits sometimes have more than one DC solution.
 2. The DC solution computed by the circuit simulator may be unstable.
- Circuit simulators do not really distinguish between stable and unstable solutions.



Convergence

- **Method-1:**

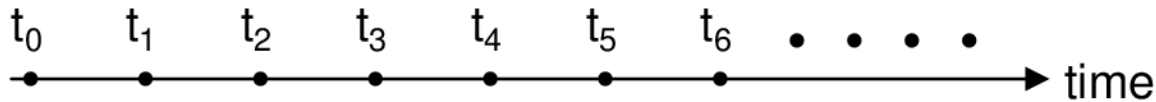
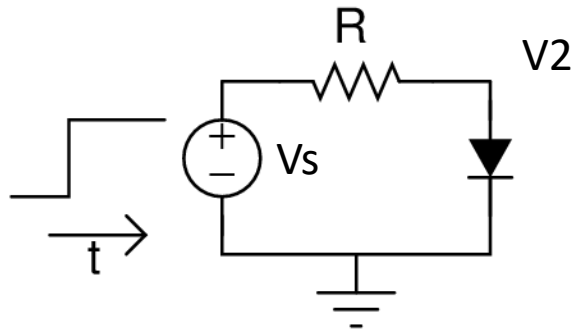
- **1st condition:** $|f_n(v^k)| < \varepsilon_f$
- **2nd condition :** $|v_n^k - v_n^{(k-1)}| < \varepsilon_x$ Where $f_n(v)$

System of non linear equation obtained by nodal analysis, ε_x and ε_f are small positive numbers.

- **Method-2: (Implemented in Spectre)**

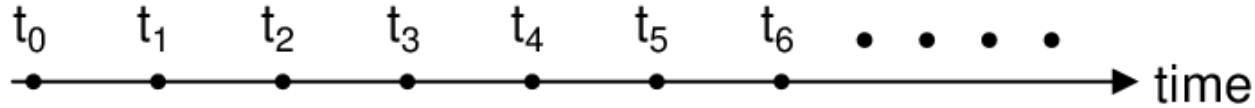
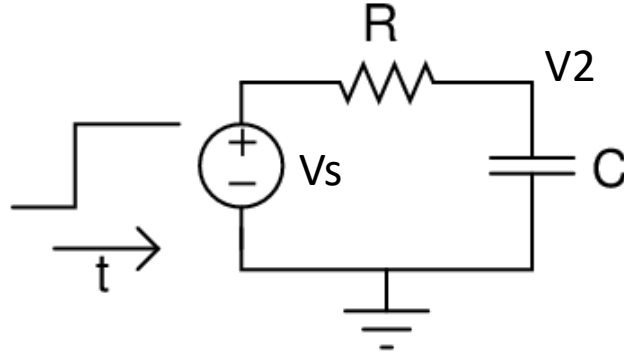
- **1st condition:** $|v_n^k - v_n^{(k-1)}| < reltol v_{n\ max} + vabstol$, By default *reltol* is 1m and *vabstol* is 1uV .
- **2nd Condition:** $|f_n(v^{(k)})| < reltol f_{n\ max} + iabstol$, where $f_{n\ max}$ is the absolute value of the largest current entering node n from any branch and *iabstol* is 1pA.

Transient Analysis - 1



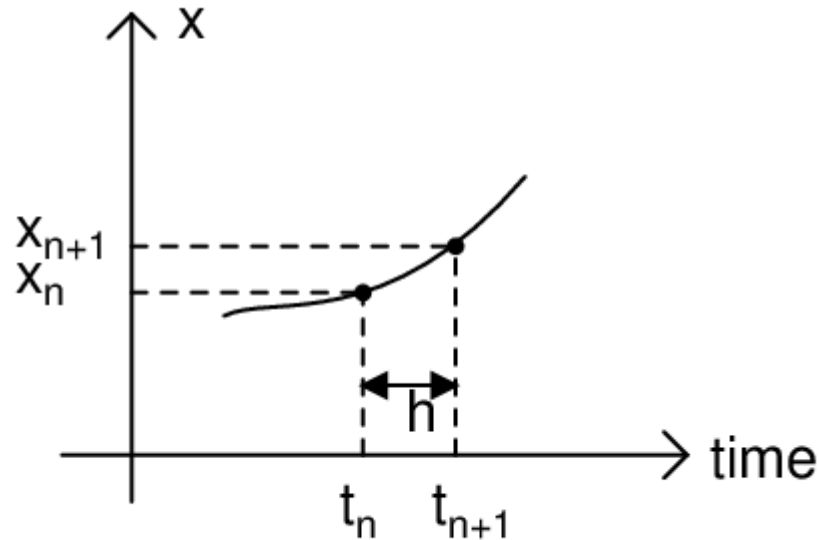
- Apply Nodal analysis.
$$\frac{V_s - V_2}{R} - I_0 \left(e^{\frac{V_2}{V_T}} - 1 \right) = 0$$
- Discretize the time, solve the above equation using Newton Raphson method.

Transient Analysis - 2



- Apply Nodal analysis,
$$\frac{V_s - V_2}{R} - C \frac{dV_2}{dt} = 0$$
$$\frac{dV_2}{dt} = \frac{V_s - V_2}{RC} \Rightarrow \frac{df}{dt} = f(t, x)$$
- Solve the above equation using Forward Euler, Backward Euler or Trapezoidal method.

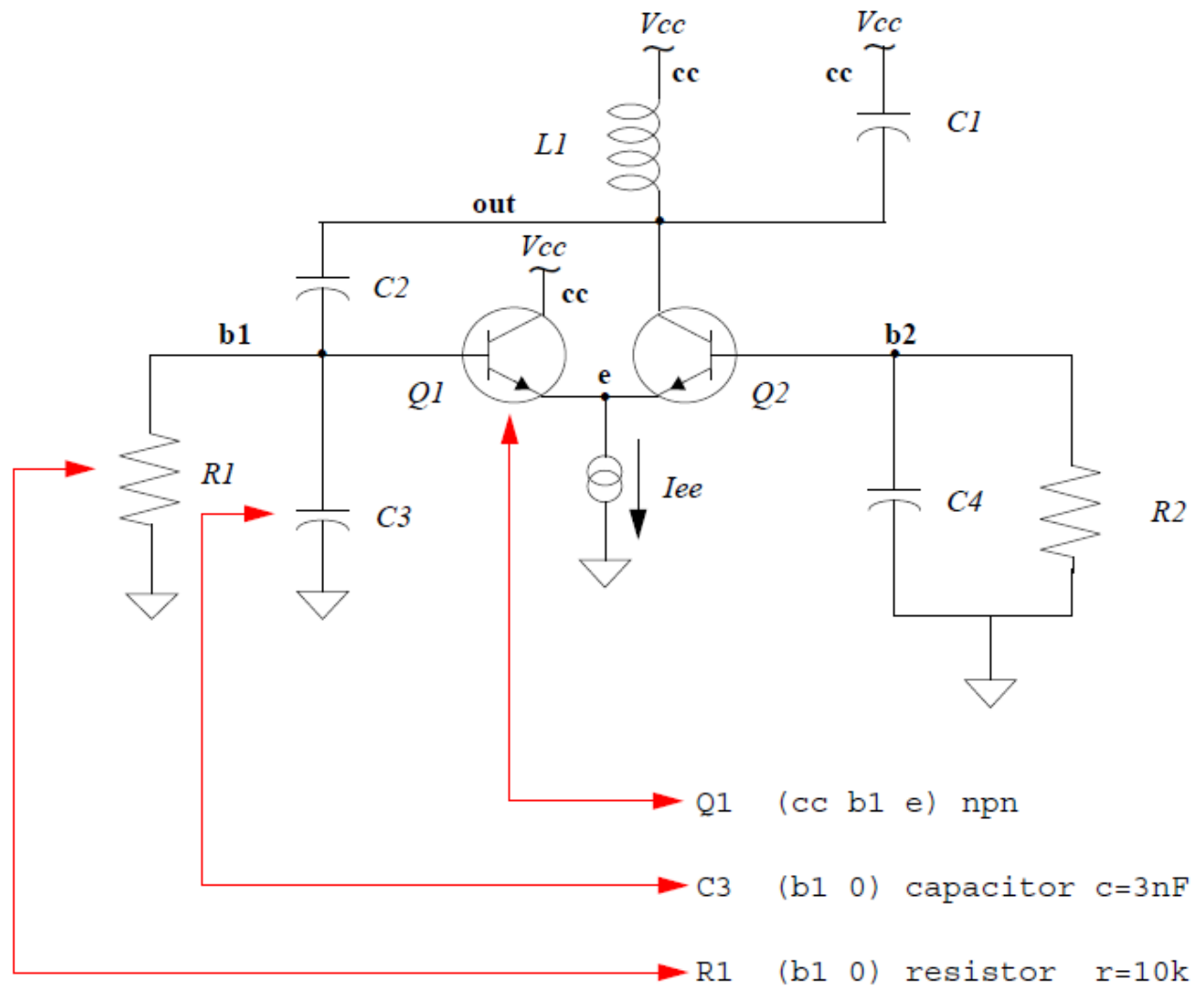
Transient Analysis - 3



Method	Value
Forward Euler	$(x_{n+1} - x_n)/h = f(t_{n+1}, x_{n+1})$
Backward Euler	$(x_n - x_{n+1})/h = f(t_n, x_n)$
Trapezoidal	$(x_{n+1} - x_n)/h = 0.5 (f(t_n, x_n) + f(t_{n+1}, x_{n+1}))$

Trapezoidal method gives minimum error for given h.
Accuracy and stability are major concern for a given method.

Netlist Example -1



Netlist Example - 2

```
// BJT ECP Oscillator
simulator lang=spectre
Vcc (cc 0) vsource dc=5
Iee (e 0) isource dc=1m
Q1 (cc b1 e) npn
Q2 (out b2 e) npn
L1 (cc out) inductor l=1u
C1 (cc out) capacitor c=1p
C2 (out b1) capacitor c=272.7p
C3 (b1 0) capacitor c=3n
C4 (b2 0) capacitor c=3n
R1 (b1 0) resistor r=10k
R2 (b2 0) resistor r=10k
ic b1=1

model npn bjt type=npn bf=80 rb=100 vaf=50 \
cjs=2pf tf=0.3n tr=6n cje=3p cjc=2p

OscResp tran stop=80u maxstep=10n
```

Comment (indicated by //)

Indicates the file contains a Spectre netlist

Instance statements

Control statement (sets initial conditions)

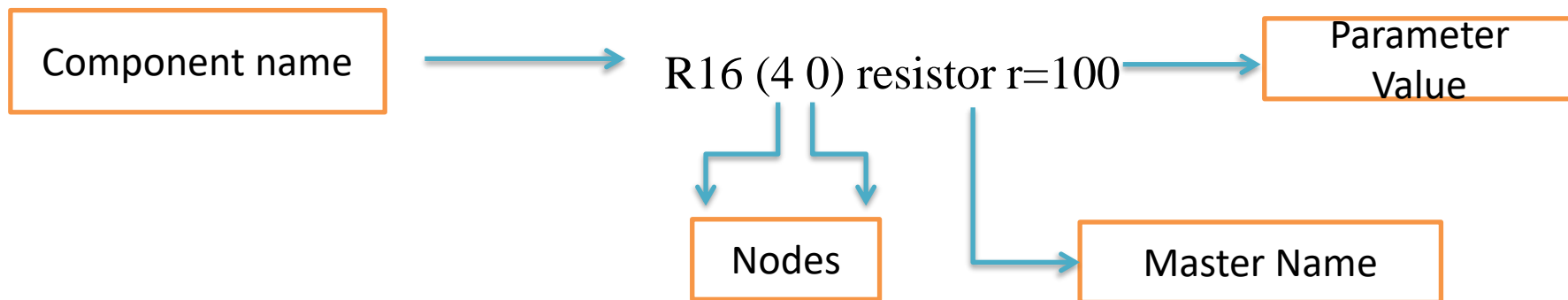
Model statement

Analysis statement

Netlist Example - 3

To specify single components within a circuit, you must provide the following information:

- A unique component name for the component
- The names of nodes to which the component is connected
- The master name of the component (identifies the type of component)
- The parameter values associated with the component
- Typical Spectre instance statement looks like this:



Netlist -5

Component Names :

Unlike SPICE, the first character of the component name has no special meaning. You can use any character to start the component name.

Master Names :

The type of a component depends on the name of the master, not on the first letter of the component name (as in SPICE); this feature gives you more flexibility in naming components.

Parameter Values :

Real numbers can be specified using scientific notation or common engineering scale factors. For example, you can specify a 1 pF capacitor value either as $c=1p$ or $c=1e-12$.

Control Statements :

To set initial conditions.

Model Statements :

Some components allow you to specify parameters common to many instances using the model statement. The only parameters you need to specify in the instance statement are those that are generally unique for a given instance of a component.

Netlist -5

Analysis Statements

Analysis statement has the same syntax as an instance statement, except that the analysis type name replaces the master name. To specify an analysis, you must include the following information in a netlist statement:

- A unique name for the analysis statement
- Possibly a set of node names
- The name of the type of analysis you want
- Any additional parameter values associated with the analysis

Spectre `–help <option>` can be used to get the help for any options/analysis. For example if you are looking for tran then use `spectre –help tran`

Running The Simulation

you can run the simulation with the spectre command. To run a simulation for the circuit we have described, type the following at the command line:






spectre osc.cir

Note: osc.cir is the file that contains the netlist.


Following Simulation Progress

- As the simulation runs, the Spectre simulator sends messages to your screen that show the progress of the simulation and provide statistical information.
- Spectre simulator prints some warnings and notifications. When you see a Spectre warning or notification, you must decide whether the information is significant for your simulation.

Extracting The Results - 1

- Once the simulation is completed, Spectre saves the output data in <Designname>.raw. Where Design name in our case is osc.
- Now you can open ocean, to print the results.
- To invoke the ocean , you need to type ocean  at the command prompt.
- You need to open the results directory by using
 - openResults("./osc.raw")  , you have to use complete path if the results are not in current directory
- Now use the following command to know about all the results available:
 - results 
- Select the result. In our example to select transient results, use following command :
 - selectResult('tran') 
- To view all the outputs (nodes and currents from different sources) saved use
 - outputs() 

Extracting The Results - 2

- To plot particular node use
 - `plot(v("<node_name>"))`, to plot output node voltage in our example, use `plot(v("out"))`
- To plot the current from the particular source
 - `plot(i("<source_name>p"))` , to plot current from source VCC in our example, use `plot(i("Vcc:p"))`
- You can do lot of operations on the outputs. For example, to extract the maximum output voltage in our case use `ymin(v("out"))`.
- To know more options and operations in ocean, use `OcnHelp()` 
- `ocnHelp()` prints the options and commands available in Ocean.
- For example, to know about the `ymin` function, use `ocnHelp('ymin')`

References

1. Ken Kundert, “THE DESIGNER’S GUIDE TO SPICE AND SPECTRE”
2. “Specter Circuit Simulator User Guide”