Behavioral Sources, Parameters and Expression Evaluation

Presented by: Thomas Mosteller ADI FAE
Expression Evaluation

- When curly braces { } are encountered, the enclosed expression is evaluated on the basis of all relations available at the scope and reduced to a floating point value (evaluated before simulation begins).
Behavioral Sources (BV, BI)

- Behavioral sources are used when the user would like to define a source with an arbitrary expression.
- Expressions can contain the following:
  - Node voltages, e.g., \( V(n001) \)
  - Node voltage differences, e.g., \( V(n001, n002) \)
  - Circuit element currents; for example, \( I(S1) \), the current through switch S1 or \( I_b(Q1) \), the base current of Q1.
    - It is assumed that the circuit element current is varying quasi-statically, that is, there is no instantaneous feedback between the current through the referenced device and the behavioral source output.
    - Similarly, any ac component of such a device current is assumed to be zero in a small signal linear .AC analysis.
  - Various functions and operations as defined in the help file.
    - Search “Arbitrary Behavioral Voltage or Current Sources” in help
  - The only difference between Bi1 and Bi2 is the direction of current.
Places to use Expressions & Behavioral Sources

- Without { }, the expression is not reduced to a value before simulation, but is calculated during simulation in real time. Below it is used within a behavioral source.

- In this example, we will create a load that dissipates a constant power regardless of the voltage across it.
Places to use Expressions & Behavioral Sources

- Expressions for component values can also be calculated in real time.

```
V1
pulse(5 19 0 .5m .5m 0 1m)

R1
R=V(X)/2
```
Places to use Expressions & Behavioral Sources

- Within the waveform editor (right click on trace name or via the ‘Plot Settings’ menu then ‘Add Trace’)

[Diagram showing waveform editor and expression editor]

Right-Click

Enter expression
Places to use Expressions & Behavioral Sources

- Behavioral sources can be used to do on the fly calculations during simulations. One example is calculating efficiency.

\[ V = (out_1)I(R_{load_1}) + (out_2)I(R_{load_2}) \]

\[ V = -V(in)i(V1) \]

\[ V = \text{limit}(v(pout)/v(pin), 0, 1) \]
Places to use Expressions & Behavioral Sources

- Try to create this waveform using behavioral sources.
Behavioral Source with Delay

- The “Delay” function can be used to insert a time delay into the behavioral source.

```
.sine 0 1 100k 0 0 0 1000meg
.tran 20u
```

```
V4 B1
V = delay(V(x)/2, 2u)
```
User-Defined Functions/Parameters

- The .func directive allows the creation of user-defined functions for use with user parameterized circuits and behavioral sources
- `.func <name>([args]) {<expression>}`
  - Ex. `.func myfunc(x,y) {sqrt(x*x+y*y)}`

- This is useful for associating a name with a function for the sake of clarity and parameterizing subcircuits so that abstract circuits can be saved in libraries.

- The .func statement can be included inside a subcircuit definition to limit the scope of the function to that subcircuit and the subcircuits invoked by that subcircuit.
User-Defined Functions/Parameters

- In this example the source B1 calls the function `constantpower` and sends it the parameters of “10” and “V(X)” (the voltage at the node labeled “X”).
- `.func constantpower` calculates 10 divided by the voltage at “X” and returns the result to source B1 in real time.
User-Defined Functions/Parameters

- The `.param` directive allows the creation of user defined variables
- Useful for varying component values without actually editing component properties
- All parameter substitution evaluation is done before the simulation begins.
- Example syntax: `.param x=y y=z z=1k*tan(pi/4+.1)`
User-Defined Functions/Parameters

- Parameters can be used within components
User-Defined Functions/Parameters

- Parameters can also be used within sources.
- Multiple parameters can be used simultaneously.

Multiple parameters

Within sources
User-Defined Functions/Parameters

- Supercap example:
Parameter Sweeps

- The `.step` command causes the analysis to be repeatedly performed while stepping a model parameter.
- Multiple back-to-back simulation results are kept instead of being discarded.
- Steps may be linear, logarithmic, or specified as a list of values:
  - Linear: `.step <stepped element> <start> <stop> <increment>`
  - Octave: `.step oct <element> <start> <stop> <#pts per octave>`
  - Decade: `.step dec <element> <start> <stop> <#pts per decade>`
  - List: `.step <element> list <value1> <value2> … <value_n>`
Parameter Sweeps example syntax

- Example: `.step I1 10u 100u 10u`
  - Step independent current source I1 from 10u to 100u in step increments of 10u (Linear).

- Example: `.step oct v1 1 20 5`
  - Step independent voltage source V1 from 1 to 20 logarithmically with 5 points per octave.

- Example: `.step dec param X 10k 1Meg 10`
  - Step global parameter X from 10k to 1Meg logarithmically with 10 points per decade.

- Example: `.step NPN 2N2222(VAF) LIST 50 75 100`
  - Perform the simulation three times with NPN model parameter VAF being 50, 75, and 100.

- Example: `.step temp -55 125 10`
  - Step the temperature from -55°C to 125°C in 10-degree step (Linear).
Parameter Sweeps

- Example: RC network and stepping a list of values

1. Edit → Spice Directive
2. Type in .step directive
3. Place on schematic
Parameter Sweep – Identifying Runs

Right-click on cursor to display which run is associated with each waveform:

Up/Down arrow keys will toggle between waveforms:
Parameter Sweep – Choosing Runs

- Use the “Select Steps” option to choose which runs are shown (Right click on the Plot Pane)
Parameter Sweeps

- Example: Stepping a source directly
Stepping Multiple Parameters

- If you have multiple stepped parameters, all the combinations will be stepped (Step sweeps may be nested up to three levels deep)

```
.step param C list .1u .5u 1u
.step param R list 100 500k 1k
```

Diagram showing the stepped parameters with a waveform graph.
Stepping Multiple Parameters

- The table function can be used to step multiple parameters simultaneously using a table format (ex. pairs of values can be defined and simulated)

- `table(x,a,b,c,d,...)` function interpolates a value for `x` based on a look up table given as a set of pairs of points.

```
.step param n 1 3 1
.param C=table (n, 1, 100n, 2, 200n, 3, 300n)
.param R=table (n, 1, 10K, 2, 15K, 3, 20K)
```
Stepping Multiple Parameters

- Example: Stepping compensation components
User-Defined Functions/Parameters

- Model parameters can be stepped.
Monte Carlo Examples

- Example using the built in MC function
  - `mc(val, tol)` is a function that uses a random number generator to return a value between `val-tol*val` and `val+tol*val`

- Example passing variables to a function using a flat or Gaussian distribution
  - `flat(x)`: a function that uses a random number generator to return a value between `-x` and `x`.
  - `{Val * (1 + FLAT(TOL) )}` is the same as `mc(val, tol)`
  - `gauss(x)`: a function that uses a random number generator to return a value with a Gaussian distribution and sigma `x`. 
“Worst Case” Examples

- Example that randomly selects the max or min value based on tolerance
  - Be careful and understand the sensitivities of your circuit. Worst case values doesn’t always give you worst case operation. Ex. imagine a circuit that unintentionally resonates at nominal values, but is fine at “worst case” values.

- Example that exercises every combination of worst case values without repeating any (minimum number of runs used)
Saving the values for multiple runs

- Example that prints the values chosen to a log file

- When doing a .AC simulation, the values are shown in dB by default (we don’t normally think of component values in dB). This example converts them back.
Thermistor Simulations: Plotting Temperature and Resistance
Plotting Temperature and Resistance

- Voltage and/or current are typically plotted on the vertical axis and time is typically plotted on the horizontal axis.
- It is possible to plot resistance, temperature, and other parameters on the horizontal and vertical axes.
- Thermistor simulation example: navigate to the NTCCircuit.asc simulation file and follow the instructions.
Plotting Temperature and Resistance

Important items to note for the NTCCircuit.asc simulation:

- The DC operating point “.op” simulation command must be used (see LTspice help regarding DC operating point definition)
- The SPICE model for the thermistor is included in the simulation file
- A two terminal thermistor schematic symbol with the appropriate device parameters is required
- Additional instructions / information is included in the simulation file.
- Voltages can be labeled and in this case the voltage across thermistor R1 is labeled Vtherm
Plotting Temperature and Resistance

Important items to note for the NTCCircuit.asc simulation (cont.):

- Currents cannot be labeled, thus we must determine what LTspice has called the current flowing into thermistor R1.
- Probing the top terminal of R1 we see the current has been labeled by LTspice as “Ix(R1:A)
- Plotting the expression V(vtherm)/Ix(R1:A) therefore plots resistance of R1.
- Note that probing the bottom terminal of R1 we see that the current has been labeled Ix(R1:B) by LTspice even though in this case the current is the same as the top terminal (but reversed)!
Examples

- Time domain capacitor-based crystal test fixture
- Freq domain capacitor-based crystal test fixture
- LTC1696 fuse crowbar
- Intrinsically safe circuit breaker
- Intrinsically safe circuit breaker (plot expressions)
- Stepping a resistor current limited boost converter
- Gear vs. Trapezoidal vs. Modified Trapezoidal Comparison
- Arbitrary capacitance: write an expression for the charge
- Arbitrary inductor: write an expression for the flux
- On the fly RMS calculation
- Charge/Energy calculation
- Worst Case/Monte Carlo comparison for an amplifier circuit (downloaded from the LTspice Yahoo! User Group)