ENGR-434

Spice Netlist Syntax Details

Rev 5/25/11

Introduction

As you may know, circuit simulators come in several types. They can be broadly grouped into those that simulate a circuit in an analog way, a digital way, or a combination of analog and digital. This last category are often called mixed mode simulators since analog simulation is used for part of the circuit and digital simulation for the rest. The most common analog simulator, SPICE (Simulation Program with Integrated Circuit Emphasis), grew out of a class project and subsequent doctoral dissertation at Berkeley. It continues to be developed there and elsewhere. It is available as freeware (at least in demo versions) and as a commercial product from several companies. In this class Eldo, a commercial spice like simulator, will be used via direct invocation and via the DAIC user interface.

Mentor Analog Simulators

There are two user views of analog simulation with our mentor tools. The first is starting simulation from Design Architect - IC (DA-IC). The second is direct invocation of the simulator. When started from within DA-IC there are two parts to the analog simulation environment: the simulation kernel (or engine), which is Eldo, and the user interface within DA-IC. Eldo uses the same netlist format and has many of the same options as SPICE. The user interface creates a netlist file and a control file and starts the simulation engine (Eldo) which writes output files to disc. After simulation is finished, DA-IC launches Ezwave to view results as specified by the user. DA-IC is customized for IC design and thus when drawing schematics you won't find libraries of "off the shelf" logic or transistor components and the palette has been tuned for actions needed in IC design. It has a simulator user interface for creating and editing the netlist and control files file sent to Eldo.

SPICE

SPICE simulates a circuit by solving simultaneous differential equations that describe voltages or currents in a circuit network. The equations come from mathematical models for components such as resistors, capacitors, bi-polar transistors (BJT), MOSFET transistors, transmission lines, etc. While precision limited numbers are used in these calculations, numerical resolution is adequate to typically consider the nodal solutions to be continuous (watch out though, I have experienced circuits whose simulation showed behavior, caused by numerical limitations, not observed in a real circuit). For a SPICE simulation, the following are needed:

- A mathematical model for each type of component in the circuit. For example a BJT.
- Parameters for the component models that tailor the model to emulate a specific component. For example, a 2n2222 NPN BJT. For IC creation, MOS transistors are constructed using a specific companies fabrication process and thus the parameters from a specific fabrication process are needed.
- A description of how components are connected to form the circuit. This is called a Netlist. A Netlist contains a list of components, interconnections, and various commands to the simulator.

SPICE Netlist

Historically, circuit simulation programs did not have graphical user interfaces. Rather, a set of commands and statements were written using a text editor and saved in a netlist file to describe the circuit, inputs to the circuit, and information to be kept for later display or analysis. The simulator was then started and it read the netlist file to know what to simulate.

While many component models and simulator commands exist, a short list will enable us to simulate our VLSI circuits. Below is a netlist for simulating an inverter:
Notes:

- netlist files for Eldo are not case sensitive. Lower and upper case can be mixed.
- any line beginning with an asterisk * is a comment
- command statements begin with a period
- blank lines, i.e. white space, is ok
- circuit element names have first letters that identify the type of component
- in the example below, my comments are in {} brackets. These brackets are **not** recognized as comment delimiters by Eldo or other SPICE versions.
- this netlist combines the contents of a .cir file and a .spi file

```
* Component: inverter                              {Informational.  Complete path name of}
                                                   {  the design is placed here by Mentor}
.lib /apps/adk-2.0/technology/accusim/ami05.mod    {a file that defines transistor parameters}
.connect GND 0                                     {SPICE considers node 0 to be ground}
.global VDD GND

v1 VDD GND dc 5                                    {a DC voltage source for power}
v2 in GND pulse(0 5 5N .2N .2N 25N 50N)            {a pulsed input voltage}

m1 out in VDD VDD p L=0.6u W=1.5u                  {p channel MOSFET}
m2 out in GND GND n L=0.6u W=1.5u                  {n channel MOSFET}
c1 in GND 1.14f                                    {capacitor}
c2 out GND 2.17f                                   {capacitor}

.probe tran                                        {save all node voltages from tran analysis}
.plot tran v(in) v(out)                            {plot the specified voltages from tran anal}
.op                                                {find DC operating point}
.tran 0 50N                                        {parameter 2 is the simulation stop time}
```

Circuit components shown in the sample netlist are detailed below. Note that node names typically come from the schematic or follow names given by Design Architect when a schematic is drawn.

Voltage source, DC

```
      V1 VDD GND DC 5
       |  |   |   |  \______voltage
       |  |   |   \_____dc voltage specified
       |  |   \____negative terminal
       |  \____positive terminal
       \____voltage name, first letter V indicates voltage source
```

Voltage source, pulsed

```
V2 IN 0 PULSE (0 5 5N .2N .2N 25N 50N)
 |  | |        | |  |   |   |  |    \__Pulse period
 |  | |        | |  |   |   |  \_______Pulse length
 |  | |        | |  |   |   \__________falling edge
 |  | |        | |  |   \______________rising edge time
 |  | |        | |  \__________________ delay from sim start
 |  | |        | \_____________________ pulsed voltage
 |  | |        \_______________________ initial voltage
 |  | \_______ net that connects to the negative terminal
 |   \______  net that connects to the positive terminal
  \____voltage name. first letter V indicates voltage source
```

Capacitor

```
C1 IN GND 1p
 |  |  |   \______ capacitor value
 |  |  \_____ negative terminal
 |  \_____ positive terminal
  \____component name. leading C identifies it as a capacitor
```

MOSFET transistor

```
M1 out in VDD VDD p L=0.6u W=1.5u AD=2.475p AS=2.475p
 |  |  |   |   |  |   |      |     |         \__source area
 |  |  |   |   |  |   |      |     \____drain area (m sq)
 |  |  |   |   |  |   |      \____width (meters)
 |  |  |   |   |  |   \____length (meters)
 |  |  |   |   |  \_____Model#
 |  |  |   |   \_____Bulk
 |  |  |   \_____source
 |  |  \_____gate
 |  \____drain
  \____component identifier. leading M identifies a MOSFET
```

Here is an alternative way to specify a pulsating (digital like) voltage source. Note that the string of bit values allow creation of an arbitrary waveform. The example here will produce the same pulsed input as the example pulse statement above.

```
V2 in gnd PATTERN 5 0 5N .2N .2N 5N 1111100000
 |  |  |          | |  |   |   |  |       \____ bit pattern
 |  |  |          | |  |   |   |  \____time spent at each
 |  |  |          | |  |   |   |       1 or 0 (bit time)
 |  |  |          | |  |   |   \___ fall time
 |  |  |          | |  |   \___ rise time
 |  |  |          | |  \___delay until pattern starts
 |  |  |          | \___voltage at logic 0
 |  |  |          \___voltage at logic 1
 |  |  \____net connection to negative terminal
 |  \____net connection to positive terminal
  \___component identifier. leading V means voltage source
```

Note that to have the most accurate simulation of your IC design, transistor source and drain area should be specified on each MOSFET and there should be parasitic capacitance on each node of the circuit.

Sub Circuits

Parts in a circuit can be grouped into what are called sub circuits. Extractors often do that for the transistors in a leaf cell or even the whole circuit that it is extracting.

statements, derived from the schematic, that define subcircuits, transistors, or instances including the following:

.subckt name port_names (this begins a group of statements that define a sub circuit)
component instances - such as transistors or capacitors
..... (more components)
.ends statements (this ends a subcircuit definition)

A subcircuit is conceptually similar to a procedure in a program and a Subcircuit Instance Statement is like a call to the procedure.

Syntax to create a Subcircuit Instance Statement:

```
X1 port1 port2 port3 port4 ... portx sub_circuit_name
 |   |     |     |     |          |          |
 |   |     |     |     |          |          \___component name
 |   |     |     |     |          \___last port name
 |   \ ___ \ ___ \____ \___Port names in the order listed in
 |                              the .subckt statement
 \_Subcircuit instance identifier. Starts with X. Must be unique.
```

Example:

The following shows how a netlist file for an inverter named min_inverter3 is created. Note the .subckt statement:

```
* File: $MGC_WD/ee434/spring11/min_inverter3

.subckt min_inverter3 GND IN OUT VDD
* devices:
m0 OUT IN VDD VDD p l=0.6u w=1.5u ad=2.475p as=2.475p
m1 OUT IN GND GND n l=0.6u w=1.5u ad=2.475p as=2.475p
* lumped capacitances:
cp1 IN GND 1.1787f
cp3 OUT GND 2.1672f
.ends min_inverter3
```

Here is the syntax for a Subcircuit Instance statement that may be part of a much larger design.

```
X1 gnd in out vdd min_inverter3
```

Notes:

1) For an inverter, the input capacitance reported by the extractor does not include the gate capacitances, just the wiring parasitic capacitance. During simulation, the gate capacitance is accounted for by the MOSFET model. Recall that the gate capacitance depends on the gate-to-channel voltage and also frequency and thus the capacitance calculated using gate area and oxide thickness is only an estimate. The MOSFET model provides a more accurate determination of the capacitance.

(notes continued)

2) Additional documentation about spice netlist format and the commands (i.e. control language) used to control Spice can be found in the Eldo Users Manual. You will find this manual as a pdf file in this directory:

/home/classes/engr434/docs/eldo_ur.pdf

Chapter 3 (starting pg 61) describes the control language and chapter 4 describes the device models (i.e. things like resistors, capacitors, transistors, etc).

Another reference book in the same directory describes Ezwave (ezwave_ur.pdf)