

Using Verilog-A to Simplify a SPICE Netlist

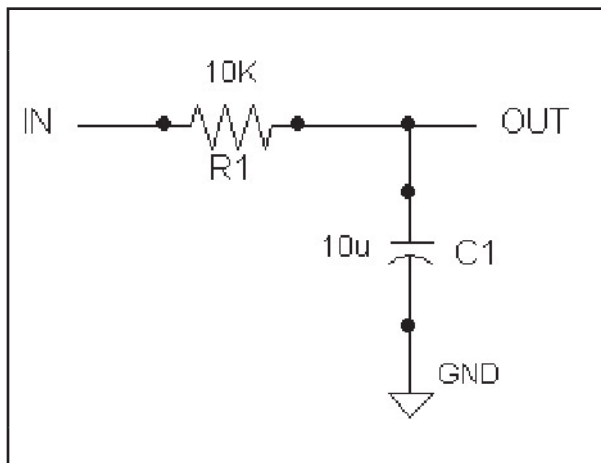
The SPICE netlist format is often a complex way of describing a circuit topology.

The Verilog-A language provides designers with an alternative method for describing analog circuit blocks. With Verilog-A rich C like syntax and clear growth path, Verilog-A is a suitable successor to a method of describing circuit topologies.

The Verilog-A language is supported by both SmartSpice and Harmony. In SmartSpice, the number of equations to solve a circuit condition can be simplified by replacing certain netlist blocks with Verilog-A equivalent netlists and therefore reducing simulation time. When SmartSpice encounters an active device it must solve a larger number of equations to determine the transistor terminal currents and then use them to solve for the currents in the circuit. By using Verilog-A description a single calculation step gives the currents for the circuit thus avoiding the expensive computation. Verilog-A description allows savings in calculation complexity, system resources and the simulation time.

The following examples show how to represent a SPICE netlist as a Verilog-A module.

Example 1: An RC Circuit



SPICE:

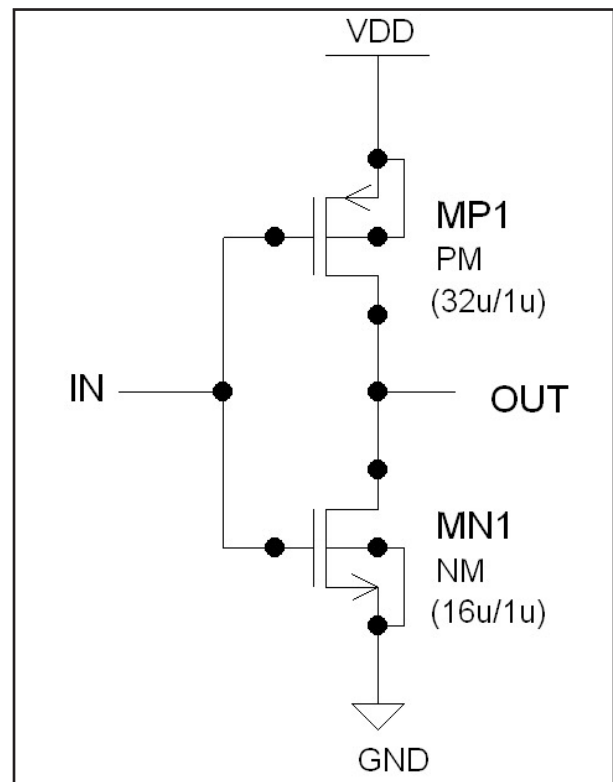
```
*RC Circuit
R1 in out 10k
C1 out gnd 10u
```

Verilog-A

```
// RC Circuit
module RC(in, out);
  inout in;
  inout out;
  electrical in;
  electrical out;
  ground gnd;

  resistor #(r(10k)) r1 (in, out);
  capacitor #(c(10u)) c1 (out, gnd);
endmodule
```

Example 2: A CMOS Inverter



An Inverter SPICE Netlist

```
*CMOS Inverter
MP1 out in vdd vdd
+ pch L=1u W=32u
MN1 out in gnd gnd
+ nch L=1u W=16u

V1 in gnd
+ pwl( 0, 0, 10e-6, 5 )

.model nch NMOS
+ level=49
.model pch PMOS
+ level=49

.tran 1n 100u
.save v(in) v(out)
.end
```

Verilog-A Equivalent Netlist

```
//CMOS Inverter
module INVERTER(in, out);
input in;
output out;

electrical in;
electrical out;
ground gnd;

PM#(.1(1e-6),.w(32e-6))
mp1(out,in,vdd,vdd);
NM#(.1(1e-6),.w(16e-6))
mn1(out,in,gnd,gnd);

endmodule
```

Pulse Generator Description Using Verilog-A Netlist

```
"include" discipline.h"

module pwlggen (pwIOut);
inout pwIOut;
electrical pwIOut;
ground gnd;

vsource #(.pwl ([0,0 10e-6, 5]) vpwlggen() (pwIOut, gnd);

endmodule

Toplevel Testbench
"include" discipline.h"

`timescale 1ns/1ns

module testbend();

electrical in;
electrical out;

pwlggen pwlggen1 (in);
rc rc1 (in, out);

endmodule
```

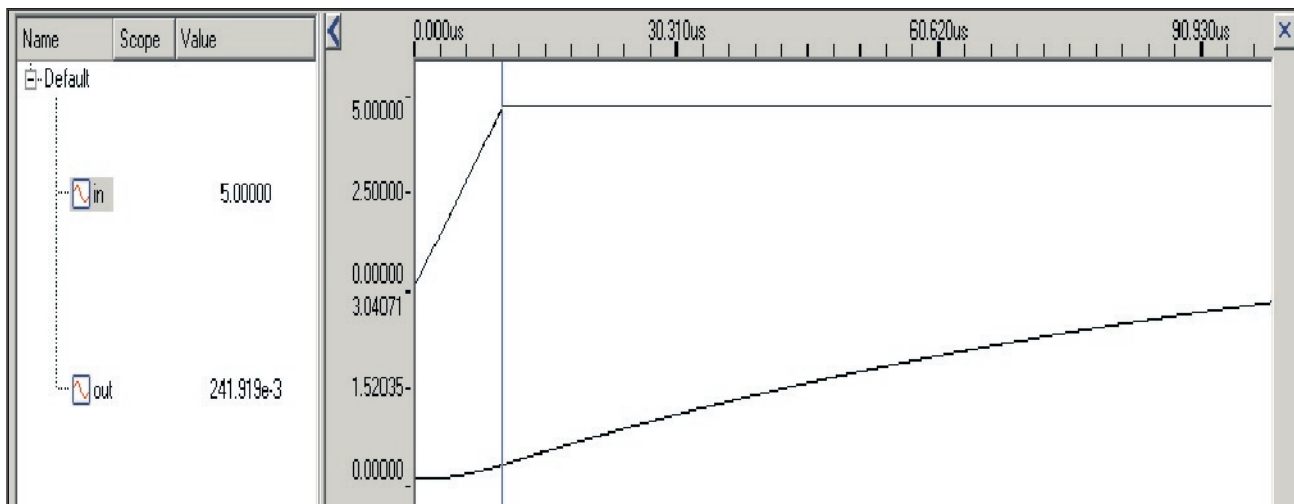


Figure 1. Transient simulation of a pulsed inverter in Harmony of - original SPICE netlist and Verilog-A equivalent netlist show identical results.