

Verilog-A를 활용하여 SPICE 넷리스트를 단순화

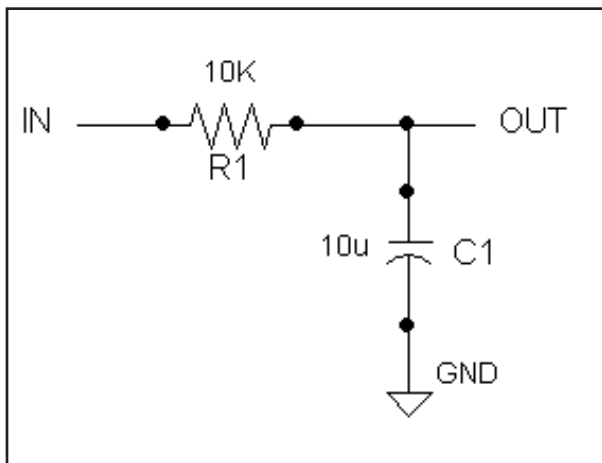
SPICE 넷리스트 형식은 회로의 토폴로지를 설명하는 정교한 방식입니다.

Verilog-A 언어는 아날로그 회로 블록을 설명하기 위한 다른 방법을 제시합니다. 구문과 같은 Verilog-A rich C와 명확한 성장 경로를 통해, Verilog-A는 회로의 토폴로지를 설명합니다.

Verilog-A 언어는 SmartSpice와 Harmony에서 모두 지원됩니다. SmartSpice에서, 특정 넷리스트 블록을 동일한 Verilog-A 넷리스트로 대체하여 시뮬레이션 시간을 단축함으로써 회로 조건에 대한 방정식의 갯수를 단순하게 할 수 있습니다. SmartSpice가 능동 소자를 만나면 더 많은 방정식을 풀어 트랜지스터 단자 전류를 결정하고, 이를 통해 회로의 전류를 풀어야 합니다. Verilog-A 설명을 활용하면, 한 번의 계산으로 회로에 대한 전류를 풀어서 힘든 계산 과정을 피할 수 있습니다. Verilog-A 서술을 사용하여 계산의 복잡성을 줄이고, 시스템 리소스 및 시뮬레이션 시간을 절약할 수 있습니다.

다음 예는 Verilog-A 모듈로 SPICE 넷리스트를 나타내는 방법을 소개합니다.

예 1. RC 회로



SPICE:

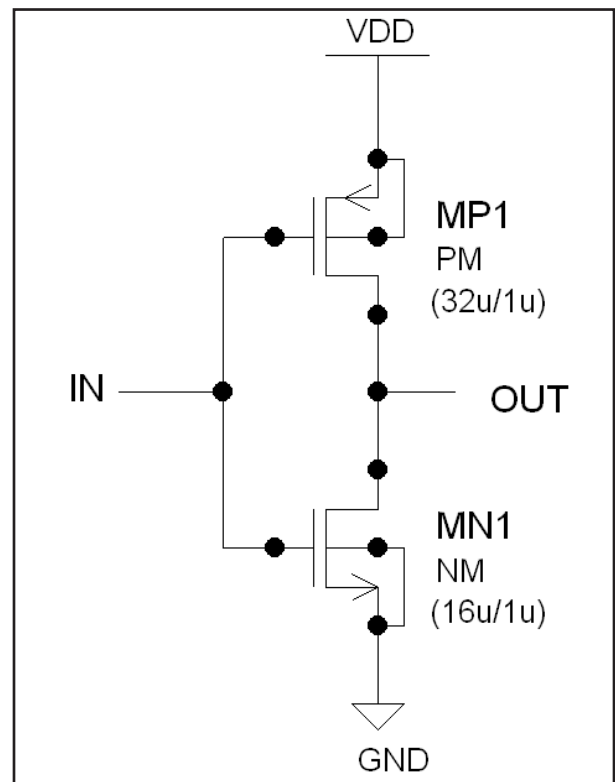
```
*RC Circuit
R1 in out 10k
C1 out gnd 10u
```

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
```

예 2: CMOS 인버터



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
```

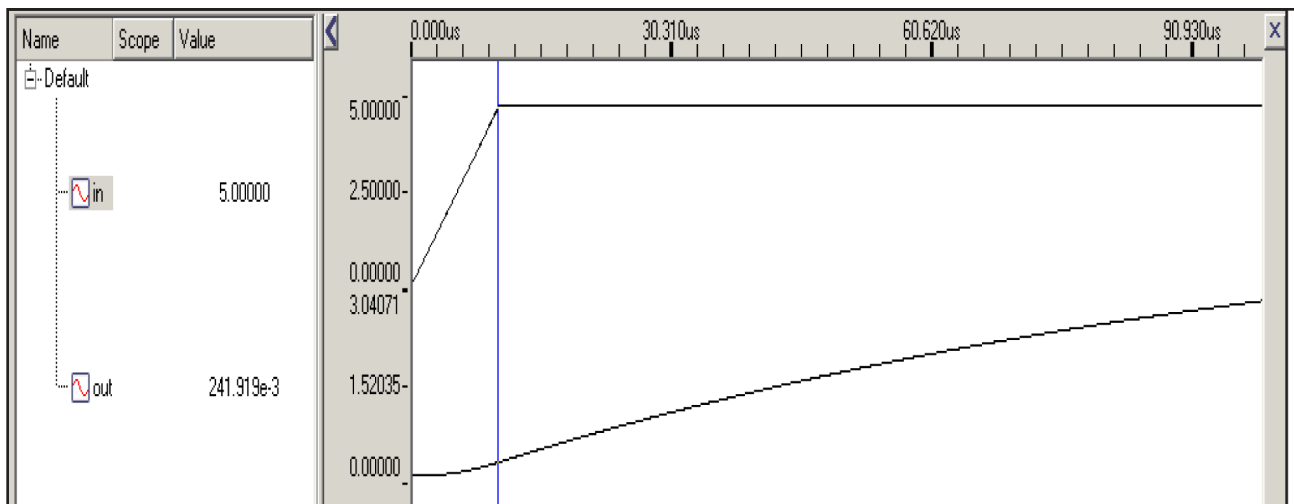


그림 1. 원본 SPICE 넷리스트와 Verilog-A 등가 넷리스트에 의해, Harmony에서 펄스 인버터의 과도 시뮬레이션을 수행하면 동일한 결과를 나타냅니다.