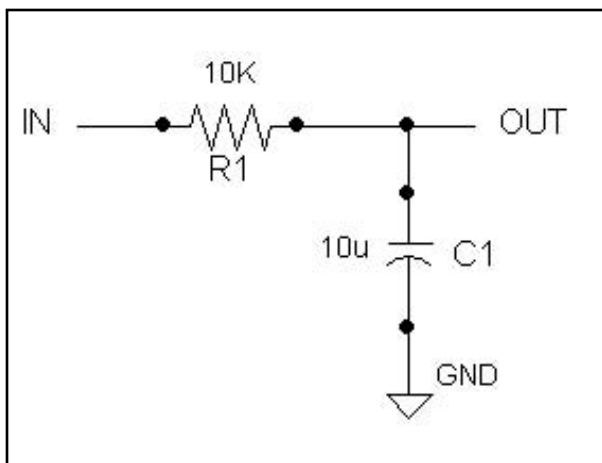


Verilog-A による SPICE ネットリストの簡略化

SPICE ネットリスト形式は、回路トポロジの記述法として、複雑であることが多々あります。アナログ回路ブロックを記述する他の方法として、Verilog-A 言語があります。充実した C 言語ライクなシンタックスを持ち、さらなる進化が予想される Verilog-A は、これからの時代にふさわしい回路トポロジ記述法です。

SmartSpice と Harmony の両シミュレータが Verilog-A 言語に対応しています。SmartSpice では、特定のネットリスト・ブロックを、Verilog-A の等価なネットリストに置き換えることで、回路条件を解くための方程式の数を少なくし、シミュレーション時間を短縮できます。SmartSpice は、能動素子を 1 つ検出することにより、より多くの方程式を解かなければなりません。まず、各トランジスタの端子電流を求め、そこから回路中の電流を求めるためです。Verilog-A を使用した記述では、回路電流を 1 回の計算ステップで求めることができるため、計算コストを削減できます。Verilog-A 記述を採用することにより、計算の複雑性を軽減し、システム・リソースを節約し、シミュレーション時間を短縮することが可能となります。以下に、SPICE ネットリストを Verilog-A モジュールとして記述する方法の例を示します。

例 1: RC 回路



SPICE:

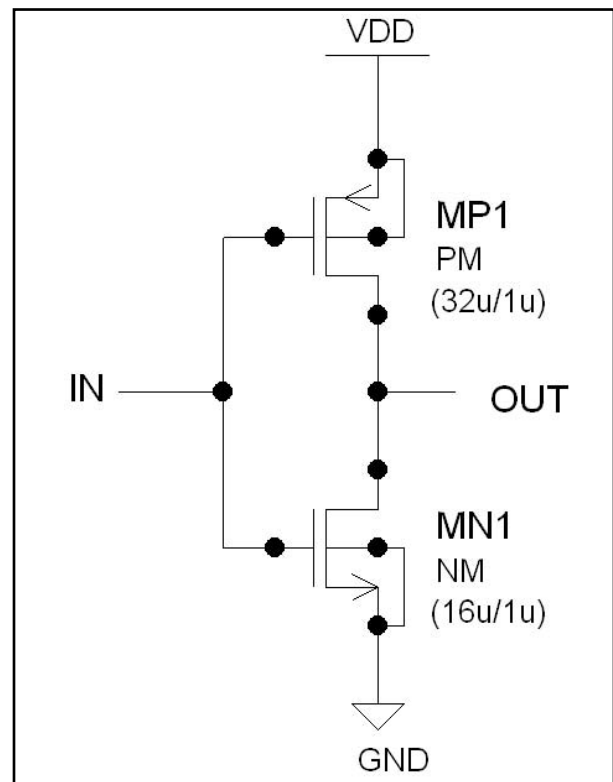
```
*RC Circuit
R1 in out 10k
C1 out gnd 10u
```

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
```

例 2: CMOS インバータ



インバータの SPICE
ネットリスト

```
*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の
等価なネットリスト

```
//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
```

パルス発生器の記述 (Verilog-A ネットリストを使用)

```
'include "discipline.h"

module pwlgen (pwlOut);
inout pwlOut;
electrical pwlOut;
ground gnd;

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

endmodule

Toplevel Testbench
'include "discipline.h"

'timescale 1ns/1ns

module testbend();

electrical in;
electrical out;

pwlgen pwlgen1 (in);
rc rc1 (in, out);

endmodule
```

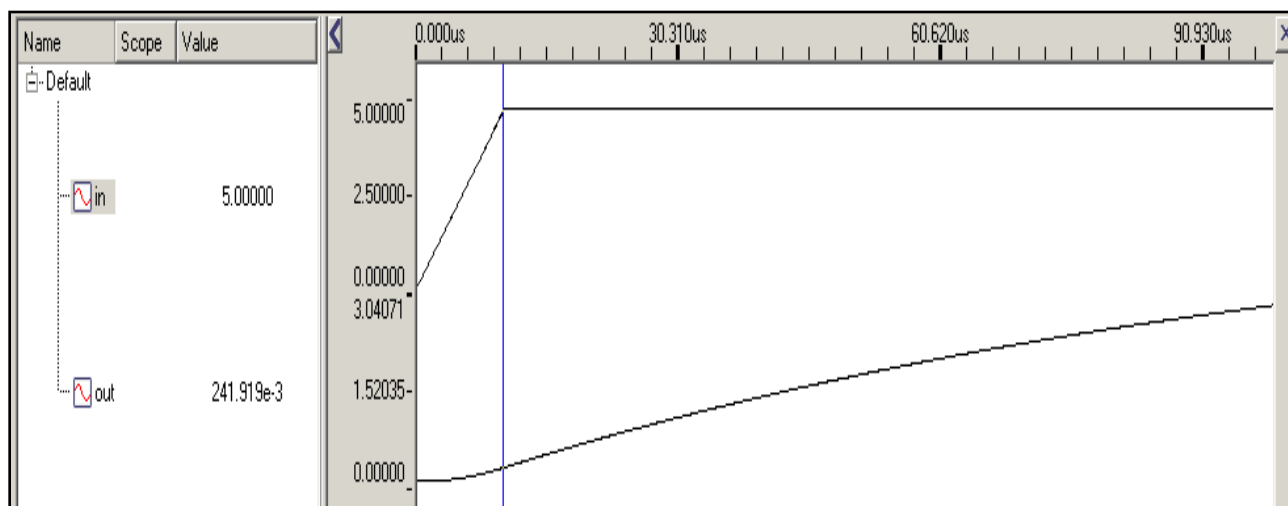


図1. Harmonyによるパルスが掃印されたインバータの過渡シミュレーション
- SPICEネットリストと、Verilog-Aの等価なネットリストで同一の結果