

# HSPICE Tutorial

by Yousof Mortazavi (Oct. 2004)

Define parameters with .param  
Can define parameter based on other parameters or expressions.

Use '+' to continue long lines on the proceeding line.

Subcircuits are SPICE's way of defining modules repeated in your design.

You can even parametrize values in your module and select these values when you instantiate your module.

```

* Tutorial: CMOS NAND Gate Characterization
***** define parameters *****
.param vdd = 3.3
.param Wn  = 1.0u
+   Wp    = Wn
+   Lmin  = 0.35u
+   delay = 100p

***** define stimuli *****
vdd vdd 0 vdd
Va   a_   0      pulse (0 vdd 20n delay delay 40n 60n)
Vb   b_   0      pulse (0 vdd 10n delay delay 20n 40n)

***** define global nodes for use in subcircuits *****
.global vdd

*** define subcircuits (modules) ***

.subckt nand    x      y      out
Mp1  out      x      vdd    vdd    P      W=Wp    L=Lmin
+   PD='Wp + 6*Lmin'    PS='Wp + 6*Lmin'
+   AD='3*Wp*Lmin'      AS='3*Wp*Lmin'
Mp2  out      y      vdd    vdd    P      W=Wp    L=Lmin
+   PD='Wp + 6*Lmin'    PS='Wp + 6*Lmin'
+   AD='3*Wp*Lmin'      AS='3*Wp*Lmin'
Mn1  out      x      node1  0      N      W=Wn    L=Lmin
+   PD='Wn + 6*Lmin'    PS='Lmin'
+   AD='3*Wn*Lmin'      AS='Wn*Lmin'
Mn2  node1    y      0      0      N      W=Wn    L=Lmin
+   PD='Wn + 6*Lmin'    PS='Wn + 6*Lmin'
+   AD='3*Wn*Lmin'      AS='3*Wn*Lmin'
.ends

.subckt inverter  in      out      strength=Wn      ratio='Wp/Wn'
Mn1  out      in      0      0      N      W=strength      L=Lmin
+   PD='strength + 6*Lmin'    PS='strength * 2 + Lmin'
+   AD='3*strength*Lmin'      AS='3*strength * Lmin'
Mp1  out      in      vdd    vdd    P      W='strength*ratio'      L=Lmin
+   PD='strength*ratio + 6*Lmin'    PS='ratio*strength + 6*Lmin'
+   AD='3*ratio*strength*Lmin'      AS='3*ratio*strength*Lmin'
.ends

```

First line in a SPICE deck is always a comment (with or without \*).

First vdd is the voltage source.  
Second vdd is the node.  
Third vdd is the parameter (=3.3 V).

Here's how you can do arithmetic on your parameters.

These are the subcircuit parameters and their default values, in case you don't specify them during instantiation.

Instantiate modules like so. (Xnand1: doesn't uses default parameter values).

Measure propagation delays accurately using the '.meas' statement.

Outputs are written to .mtn files, where  $n=0,1,\dots$  (alteration number)

.ALTER statements allows us to modify the circuit and run again.

They must be before the final .end statement.

Note: ALTER blocks are incremental!!!

```
**** A buffer module consisting of two inverters
.subckt buffer in      out      strength=Wn  ratio='Wp/Wn'
Xinv1 in  mid  inverter  strength=strength  ratio=ratio
Xinv2 mid  out  inverter  strength=strength  ratio=ratio
.ends

***** define main circuit *****
Xbuff1 a_  a  buffer  strength='Wn'  ratio=1
Xbuff2 b_  b  buffer  strength='Wn'  ratio=1
Xnand1 a  b  out  nand
* Load capacitance
Cl  out  0  0.1p

***** Analysis Options *****
.tran STEP=5p STOP=80n sweep wn lin 5 1.0u 5.0u

.meas tran tplh_blar trig v(a)  td=20ns val='vdd/2' cross=1
+      targ v(out) td=20ns val='vdd/2' cross=1

.meas tran tplh_albf trig v(b)  td=30ns val='vdd/2' cross=1
+      targ v(out) td=30ns val='vdd/2' cross=1
.meas tran tplh_albr trig v(b)  td=50ns val='vdd/2' cross=1
+      targ v(out) td=50ns val='vdd/2' cross=1
.meas tran tplh_blaf trig v(a)  td=60ns val='vdd/2' cross=1
+      targ v(out) td=60ns val='vdd/2' cross=1

.meas tran avgpower AVG power from=1ns to=80ns
.meas tran peakpower MAX power from=1ns to=80ns

***** load 0.35u library *****
.prot
.lib 'logs355v.1' TT
.unprot

***** Alter the parameters and run again
.alter case 2: Wp=2Wn
.param Wp = '2*Wn'

.alter case 3: increase Vdd by 10%
.param vdd = '3.3*1.1'

.alter case 4: use fast corner
.del lib TT
.lib 'logs355v.1' FF

.alter case 5: change load capacitance
Cl  out  0  0.01p

.alter case 6: change temperature
.TEMP 70
.end
```

We can sweep parametrized values using 'sweep'.

Here 'lin' means linear. Alternatives are: 'dec' and 'oct'.

Temperature may be swept by using *temp* instead of *sweep par*

Here we trigger when the voltage at node 'a' crosses vdd/2, and measure the time until the output crosses vdd/2. Td = 20 ns makes sure were on the right part of the waveform.

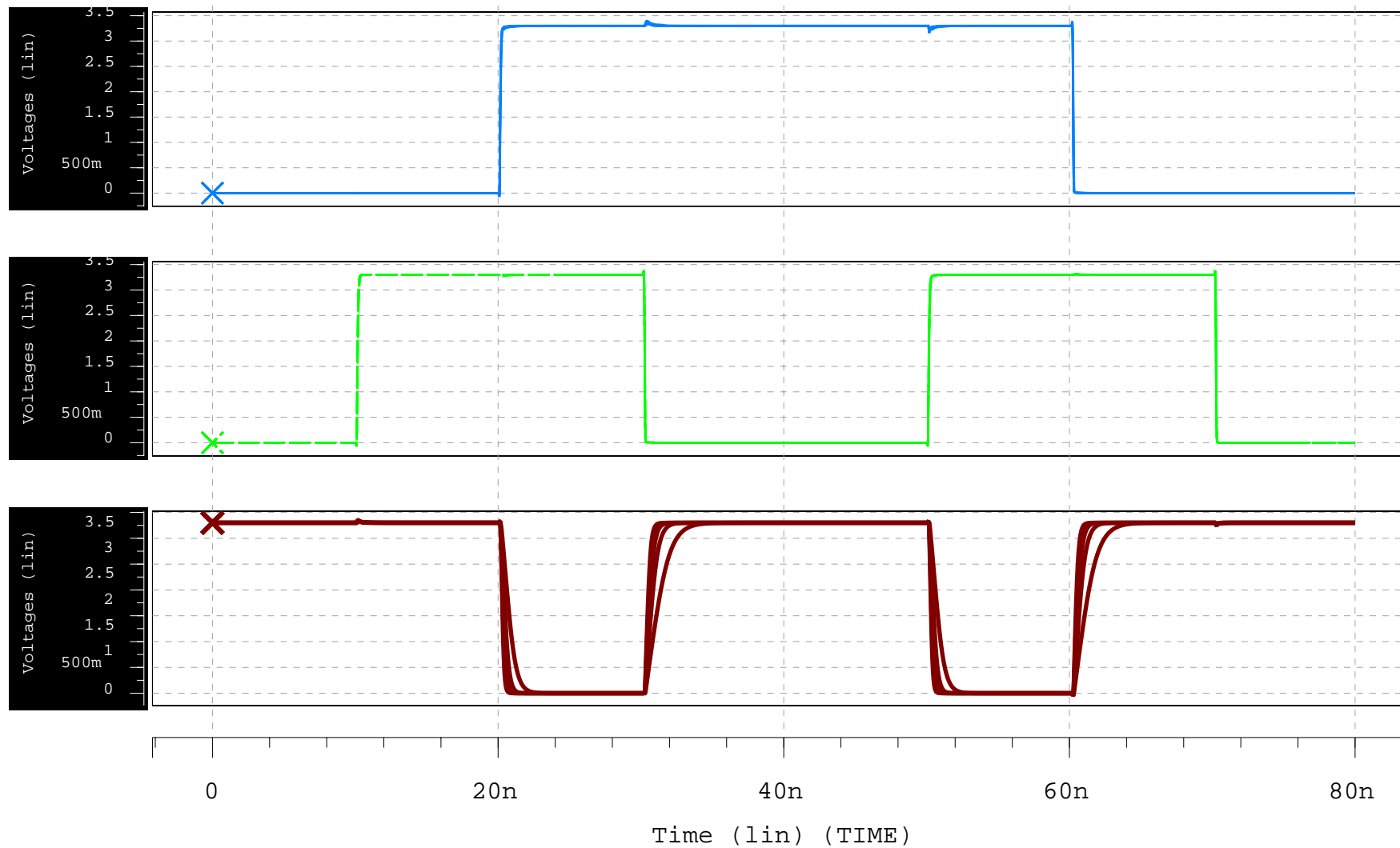
Power measurements

We can even change the models.

Elements may also be replaced.

Temperature may be altered as well.

\* tutorial: cmos nand gate characterization



Design	Type	Wave	Symbol
D0: NAND	Transient	D0:tr0:v(a)	<span style="color: blue;">×</span> —
D0: NAND	Transient	D0:tr0:v(b)	<span style="color: green;">×</span> ---
D0: NAND	Transient	D0:tr0:v(out)	<span style="color: darkred;">×</span> —