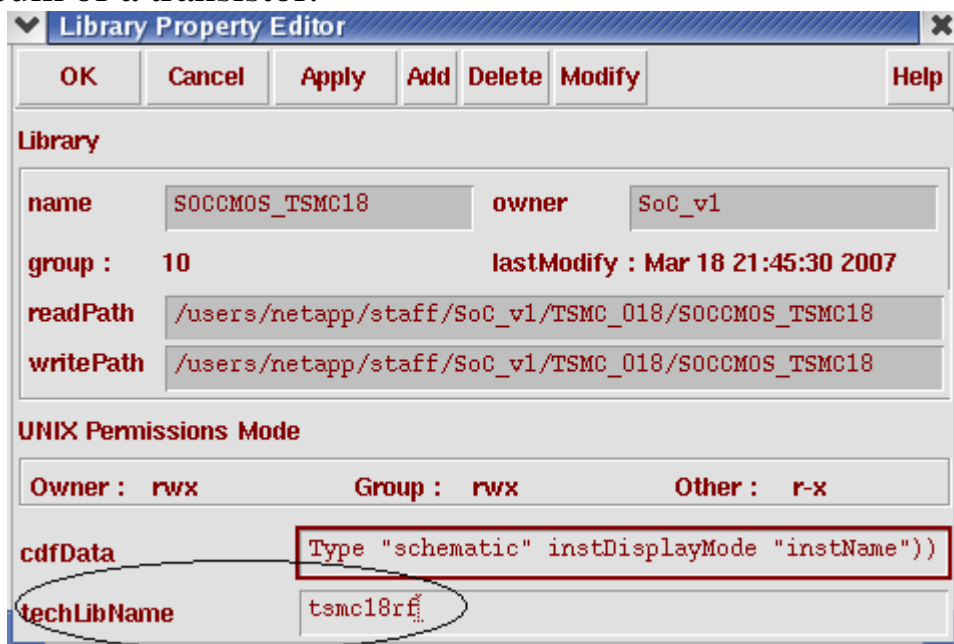


# Sized Inverter Drive net with RC Parasitic

For executing this manual we used the following program:



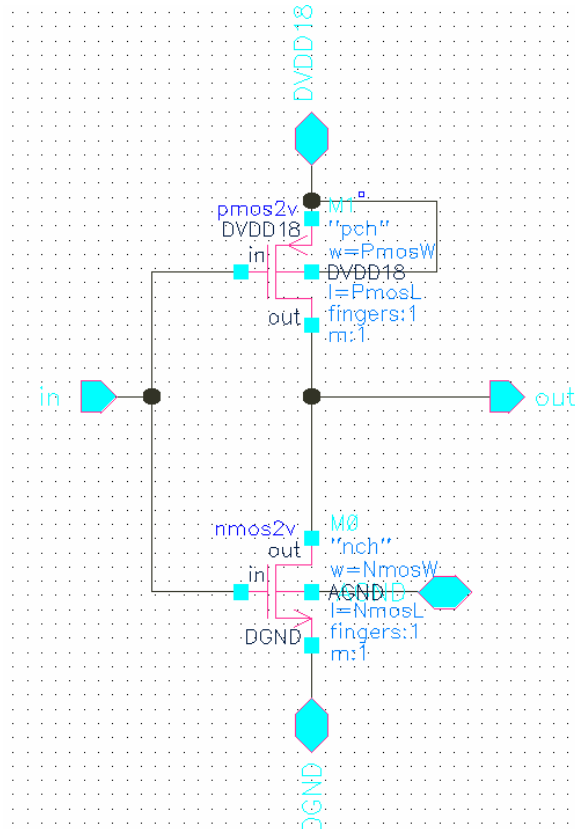
The technology library we used to was created for length 0.18um of a transistor.



The work was done in hierarchical way.

At the beginning we created the **inv\_1** cell in schematic view  
And afterwards we created a symbol for the cell. The same for **rc\_line** cell.

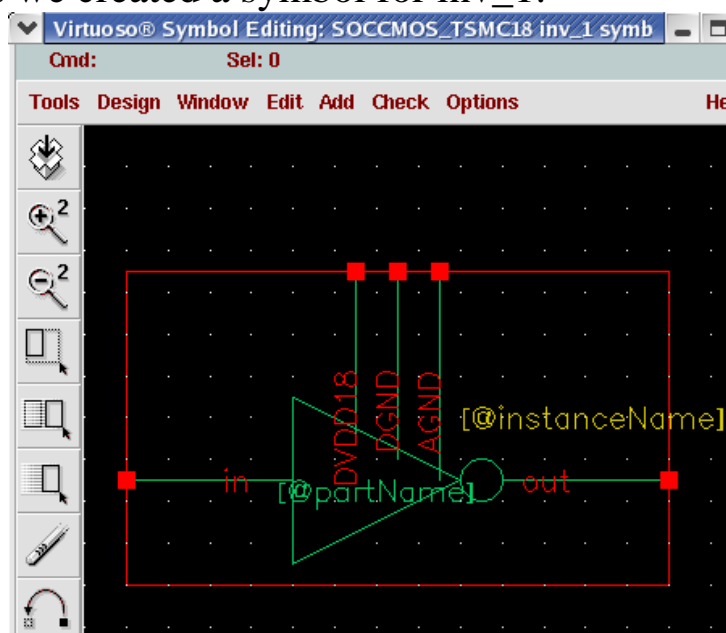
## inv\_1 cell



As you can see we splitted the bulk of nmos and connected different gnd for source in order to reduce the noise.

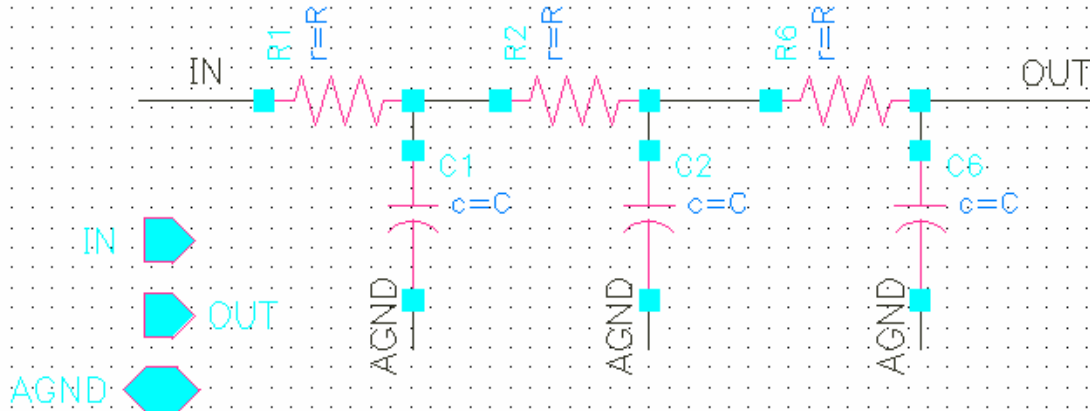
Also we defined design variables (PmosW/ PmosL / NmosW/ NmosL) which later will be used for simulation.

Afterwards we created a symbol for inv\_1:

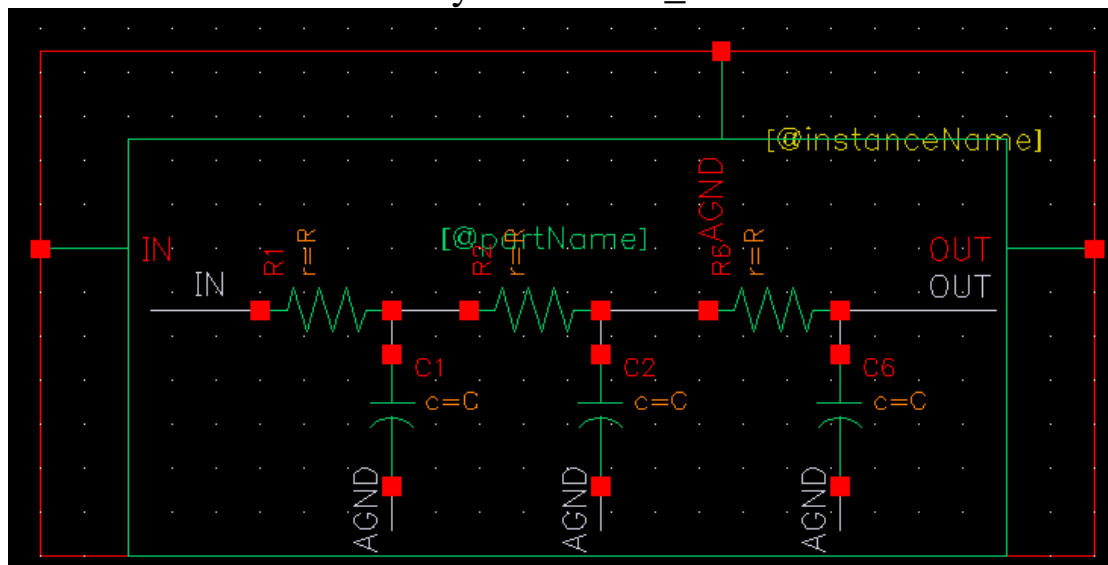


## rc\_line cell

We created one unit of rc\_line from 3 resistors and capacitors. It will save a lot of job and place in paragraph 2 (instead of creating 30 units we will copy 10 symbols of the following cells). Also we defined R/C as design variables.



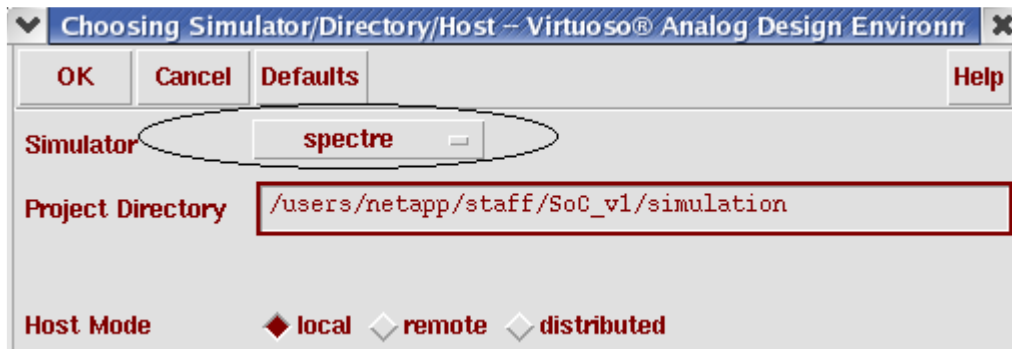
Afterwards we created a symbol for rc\_line:



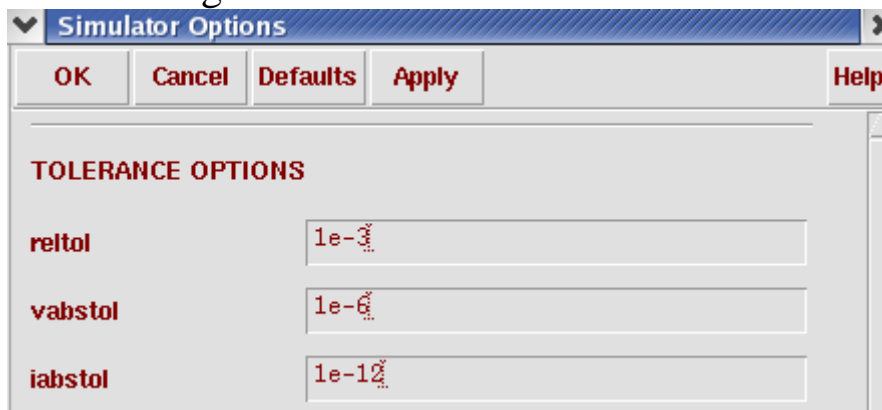
The next stage was to build a state of simulation.

## Simulation of paragraph 1

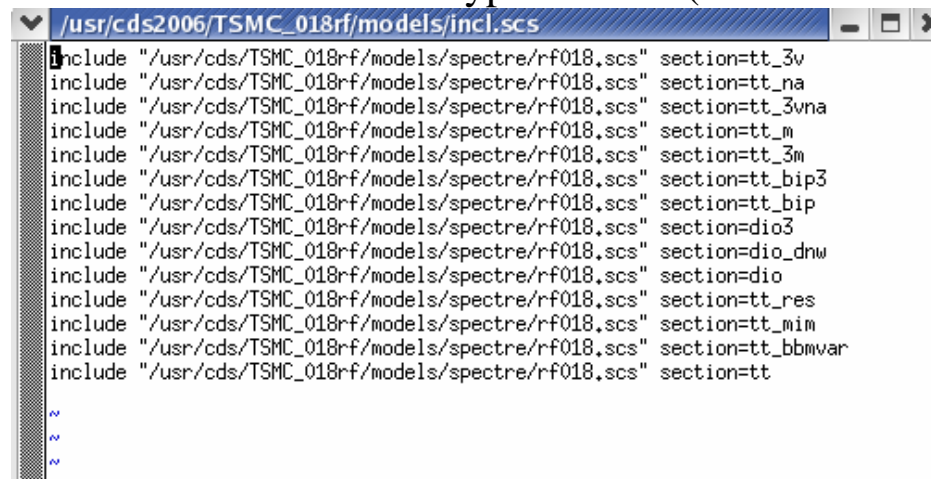
The simulator is:



With the following resolution:



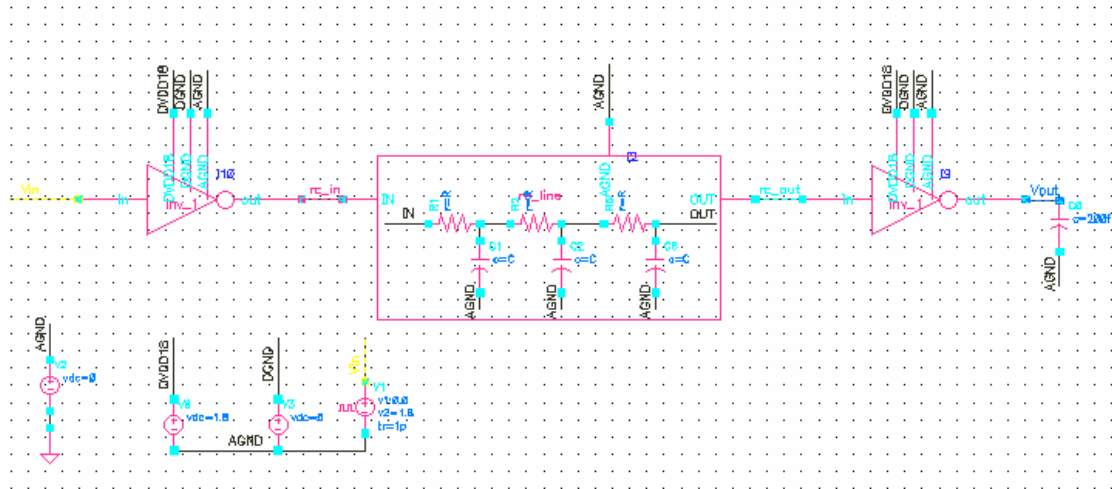
All the simulations are in typical mode (no corners required).



Therefore the temperature for simulation:



The following test was created for the simulation:



We generated an input signal with vpulse source at 25MHZ.

<b>Voltage 1</b>	0.0 V <sub>dc</sub>	<b>off</b>
<b>Voltage 2</b>	1.8 V <sub>dc</sub>	<b>off</b>
<b>Delay time</b>	0 ns	<b>off</b>
<b>Rise time</b>	1p ns	<b>off</b>
<b>Fall time</b>	1p ns	<b>off</b>
<b>Pulse width</b>	20n ns	<b>off</b>
<b>Period</b>	40n ns	<b>off</b>

## The netlist which was created by simulation:

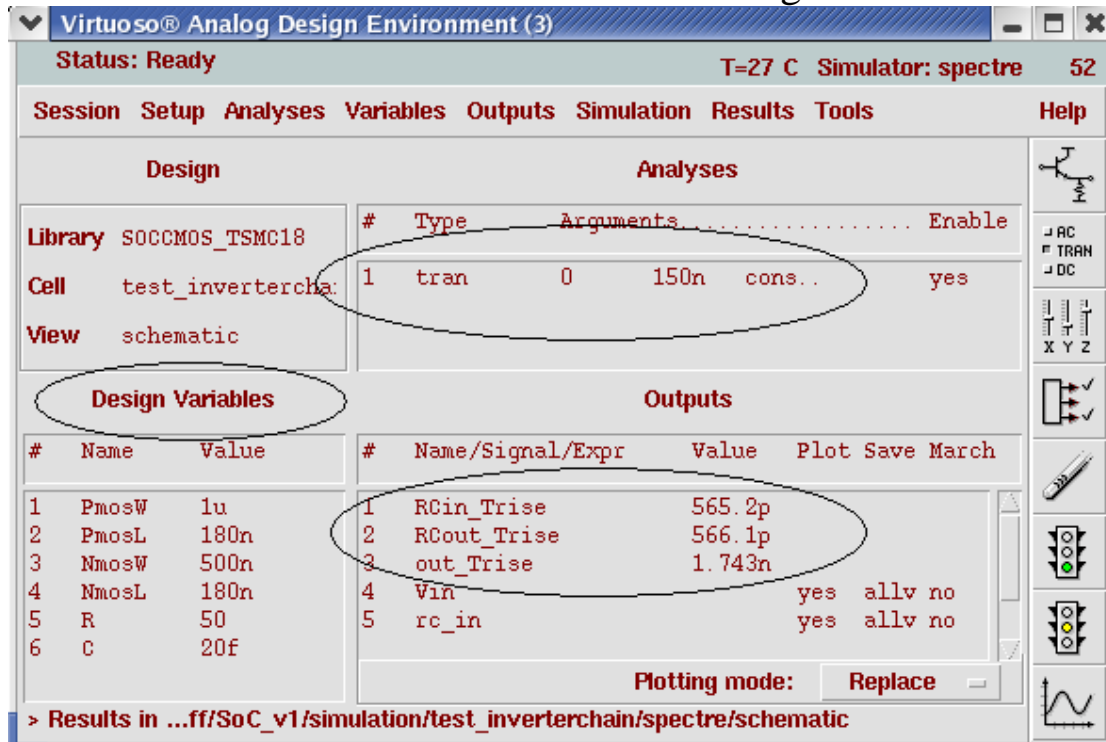
```
// Generated for: spectre
// Generated on: Mar 22 21:29:51 2007
// Design library name: SOCCMOS_TSMC18
// Design cell name: test_inverterchain
// Design view name: schematic
simulator lang=spectre
global 0
include "/usr/cds2006/IC_5.1.41/tools/dftII/samples/artist/ahdLib/quantity.spectre"
parameters PmosW=1u PmosL=180n NmosW=500n NmosL=180n R=50 C=20f
include "/usr/cds2006/TSMC_018rf/models/incl.scs"

// Library name: SOCCMOS_TSMC18
// Cell name: rc_line
// View name: schematic
subckt rc_line AGND IN OUT
  R6 (net16 OUT) resistor r=R
  R2 (net18 net16) resistor r=R
  R1 (IN net18) resistor r=R
  C6 (OUT AGND) capacitor c=C
  C2 (net16 AGND) capacitor c=C
  C1 (net18 AGND) capacitor c=C
ends rc_line
// End of subcircuit definition.

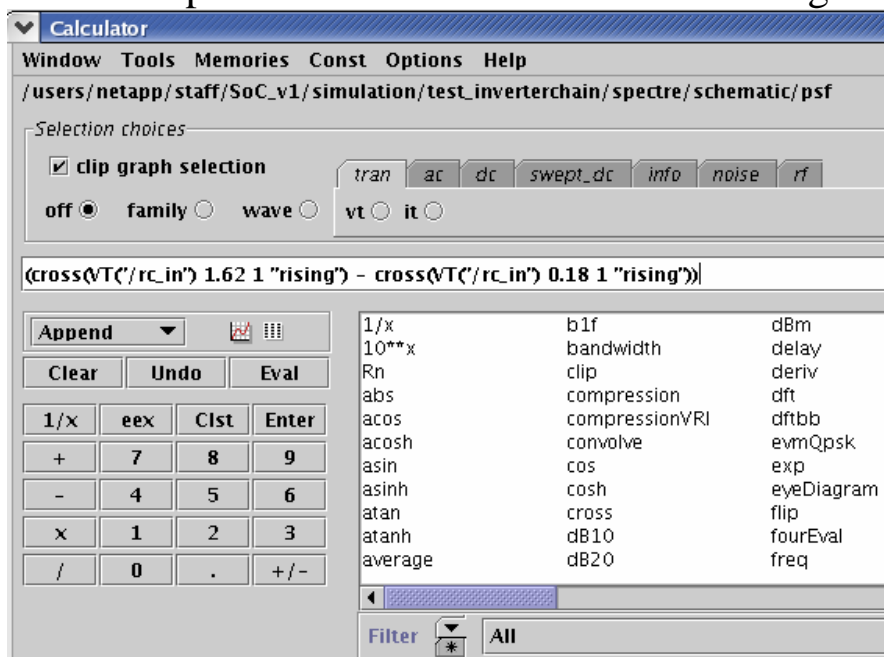
// Library name: SOCCMOS_TSMC18
// Cell name: inv_1
// View name: schematic
subckt inv_1 AGND DGND DVDD18 in out
  M1 (out in DVDD18 DVDD18) pch l=PmosL w=PmosW m=1 \
    ad=(PmosW*((4.8e-07+((1-1)+5.4e-07)/2+0)*(int((1+1)/2)-int(1/2)))+(1/2)+5.4e-07*(int((1+2)/2)-int((1+1)/2)))+(0+0*((1-1)/2+1-0))*(int((1+1)/2)-int(1/2))+(0+0*int(1/2))*
    as=(PmosW*((4.8e-07+((1-1)+5.4e-07)/2+0)*(int((1+1)/2)-int(1/2)))+(4.8e-07+4.8e-07*(int(1/2)-1)+5.4e-07+0+0)*(int((1+2)/2)-int((1+1)/2)))+(0+0*((1-1)/2+1-0))*(int((1+1)
    pd=((4.8e-07+((1-1)+5.4e-07)/2+0)*(int((1+1)/2)-int(1/2)))+(1/2)+5.4e-07*(int((1+2)/2)-int((1+1)/2)))+2*((1+1)*int((1+1)/2)-int(1/2))+1+0*(int((1+2)/2)-int((1+1)/2)
    ps=((4.8e-07+((1-1)+5.4e-07)/2+0)*(int((1+1)/2)-int(1/2)))+(4.8e-07+4.8e-07*(int(1/2)-1)+5.4e-07+0+0)*(int((1+2)/2)-int((1+1)/2)))+2*((1+1)*int((1+1)/2)-int(1/2))+(1+
    nrd=(5.4e-07+2.7e-07*(1+2-2))/1/(PmosW*2) \
    nrs=(5.4e-07+2.7e-07*(1+2-2))/1/(PmosW*2)
  M0 (out in DGND AGND) nch l=NmosL w=NmosW m=1 \
    ad=(NmosW*((4.8e-07+((1-1)+5.4e-07)/2+0)*(int((1+1)/2)-int(1/2)))+(1/2)+5.4e-07*(int((1+2)/2)-int((1+1)/2)))+(0+0*((1-1)/2+1-0))*(int((1+1)/2)-int(1/2))+(0+0*int(1/2))*
    as=(NmosW*((4.8e-07+((1-1)+5.4e-07)/2+0)*(int((1+1)/2)-int(1/2)))+(4.8e-07+4.8e-07*(int(1/2)-1)+5.4e-07+0+0)*(int((1+2)/2)-int((1+1)/2)))+(0+0*((1-1)/2+1-0))*(int((1+1)
    pd=((4.8e-07+((1-1)+5.4e-07)/2+0)*(int((1+1)/2)-int(1/2)))+(1/2)+5.4e-07*(int((1+2)/2)-int((1+1)/2)))+2*((1+1)*int((1+1)/2)-int(1/2))+1+0*(int((1+2)/2)-int((1+1)/2)
    ps=((4.8e-07+((1-1)+5.4e-07)/2+0)*(int((1+1)/2)-int(1/2)))+(4.8e-07+4.8e-07*(int(1/2)-1)+5.4e-07+0+0)*(int((1+2)/2)-int((1+1)/2)))+2*((1+1)*int((1+1)/2)-int(1/2))+(1+
    nrd=(5.4e-07+2.7e-07*(1+2-2))/1/(NmosW*2) \
    nrs=(5.4e-07+2.7e-07*(1+2-2))/1/(NmosW*2)
ends inv_1
// End of subcircuit definition.

// Library name: SOCCMOS_TSMC18
// Cell name: test_inverterchain
// View name: schematic
I2 (AGND rc_in rc_out) rc_line
V1 (Vin AGND) vsource type=pulse val0=0.0 val1=1.8 period=40n delay=0 \
  rise=1p fall=1p width=20n
V0 (DVDD18 AGND) vsource dc=1.8 type=dc
V2 (AGND 0) vsource dc=0 type=dc
V3 (DGND AGND) vsource dc=0 type=dc
C0 (Vout AGND) capacitor c=200f
I9 (AGND DGND DVDD18 rc_out Vout) inv_1
I10 (AGND DGND DVDD18 Vin rc_in) inv_1
simulatorOptions options reltol=1e-3 vabstol=1e-6 iabstol=1e-12 temp=27 \
  tnom=27 scalen=1.0 scale=1.0 gmin=1e-12 rforce=1 maxnotes=5 maxwarns=5 \
  digits=5 cols=80 pivrel=1e-3 ckptclock=1800 \
  sensfile=".../psf/sens.output" checklimitdest=psf
```

For the simulation we created a state in analog environment

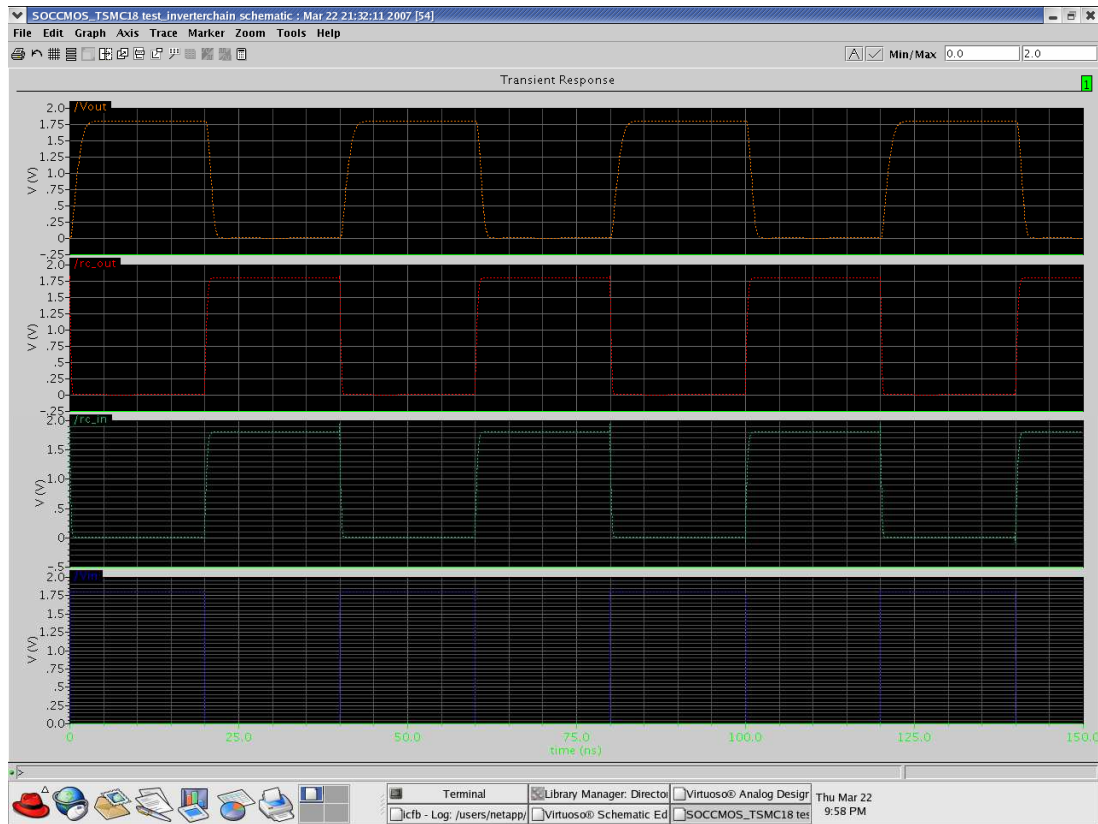


In design variables are all the parameters of transistors and RC network parameters. We defined transient for 150ns run. We also build a function to calculate manually rise time at 3 points. For example calculation of rise time at RC stage in:

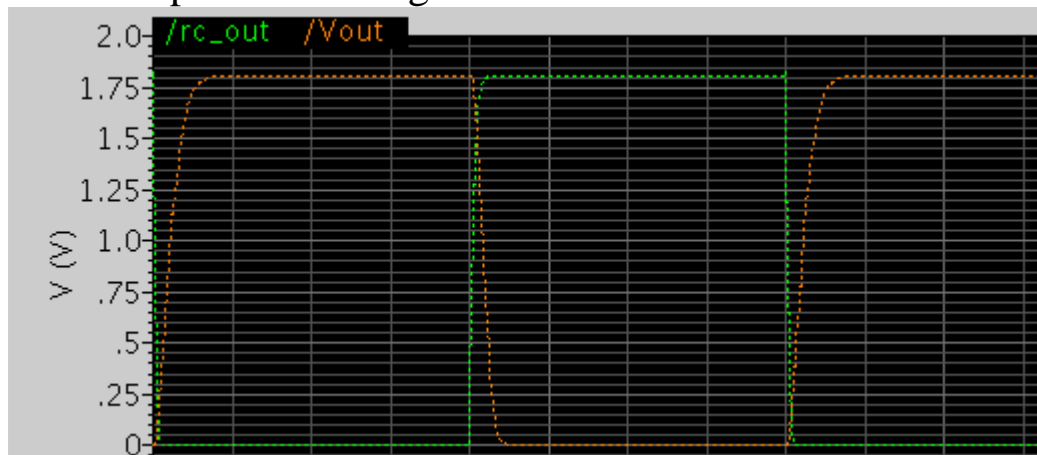


And we defined the expression as RCin\_Trise (look at previous illustration).

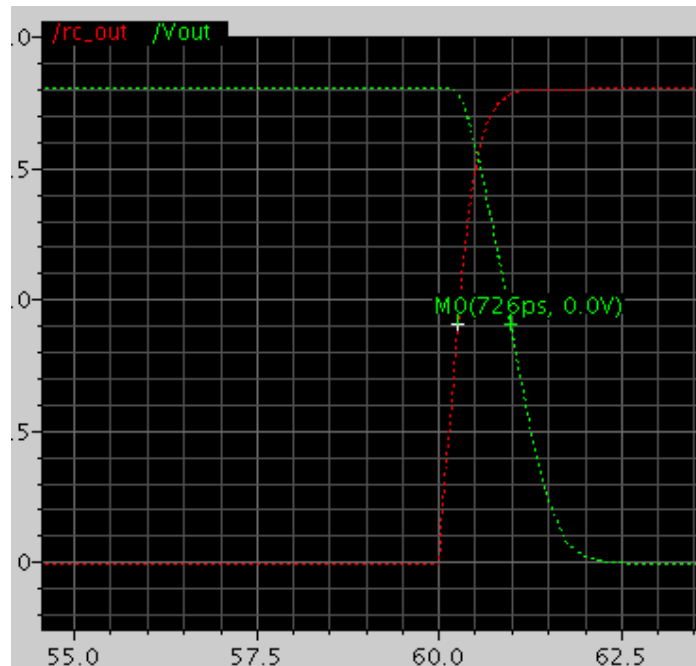
The delay time however we measured manually from the signals at waveform:



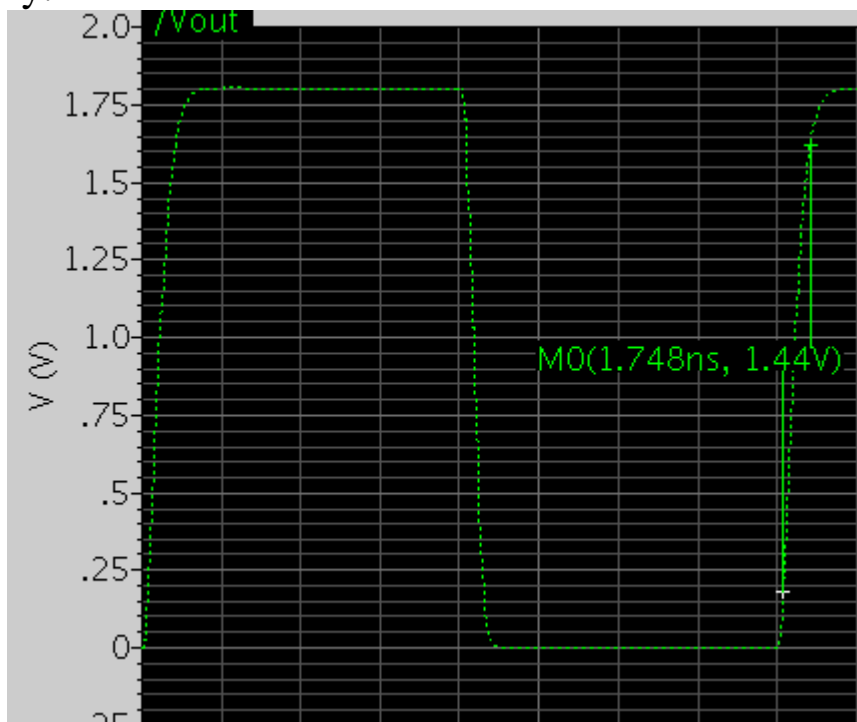
For example for RC stage out rise:



The measurement from time  $rc\_out$  reached 50% (0.9v) until  $V_{out}$  will go down to 50% of it's value.



Just to prove that the expression which are calculating rise time (from 10% -0.18v to 90%- 1.62v), we did one measurement manually.

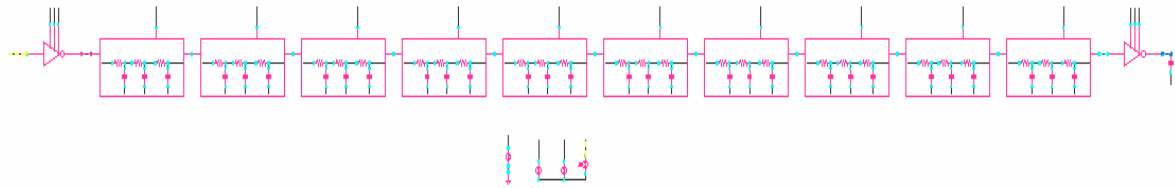


As you can the value matches at 99.5% the value of  $out\_Trise$  in analog environment (5 illustrations back).

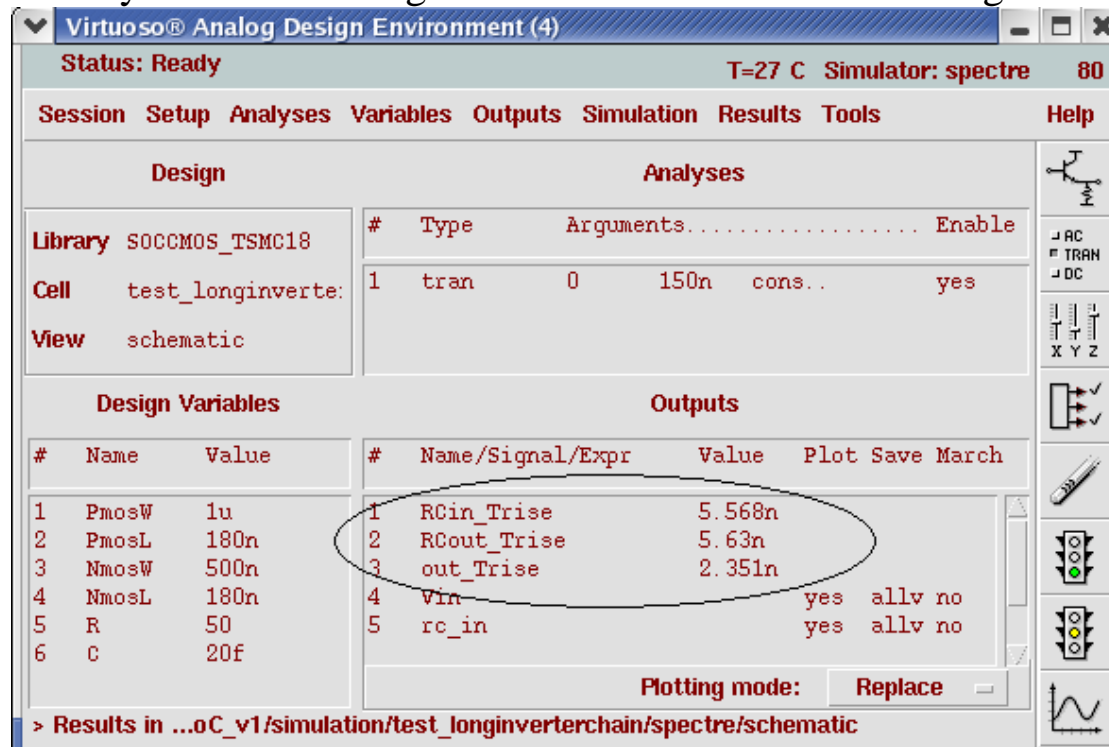
So the measurements of paragraph 1 are as following:

1 <sup>st</sup> inverter size	2 <sup>nd</sup> inverter size	RC stage in-delay	RC stage in-rise	RC stage out-delay	RC stage out-rise	out-delay	out-rise
w_nmos=0.5um	w_nmos=0.5um	264ps	565.2ps	6.4ps	566.1ps	726ps	1.743ns
w_pmos=1um	w_pmos=1um						

For **Paragraph 2** all we had to do is to multiply rc\_line 10 times. The following test was built:



Exactly the same analog environment was used for testing



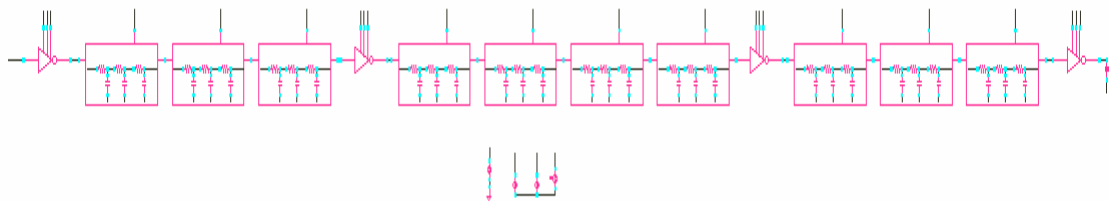
The following results were achieved:

1 <sup>st</sup> inverter size	2 <sup>nd</sup> inverter size	RC stage in-delay	RC stage in-rise	RC stage out-delay	RC stage out-rise	out-delay	out-rise
w_nmos=0.5um	w_nmos=0.5um	2.17ns	5.568ns	481ps	5.63ns	1.3ns	2.351ns
w_pmos=1um	w_pmos=1um						

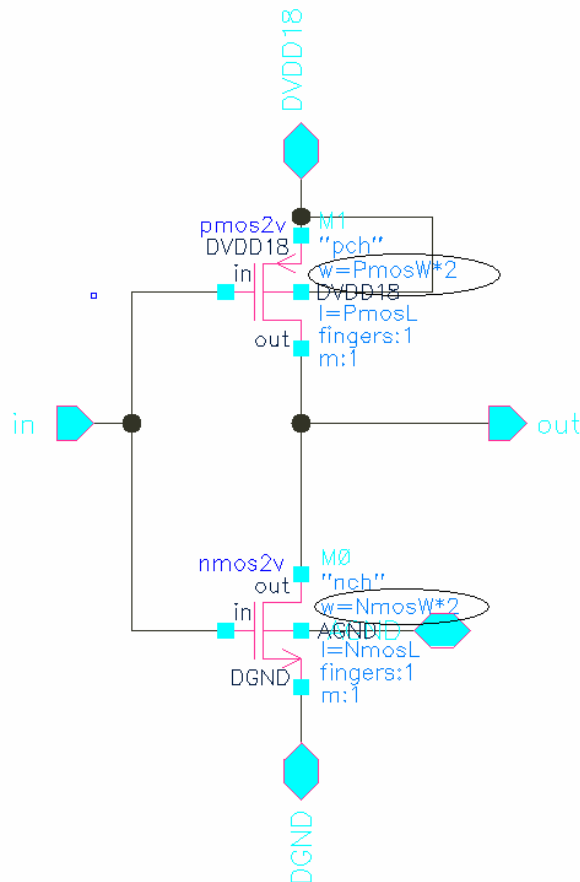
### Paragraph 3:

In order to decrease the delay and rise time from paragraph 2 we build inv\_2 ,inv\_4 ,inv\_8 besides inv\_1. The length of all inverters is minimal and defined by technology. We placed the inverters equally between the rc chains from smallest to biggest.

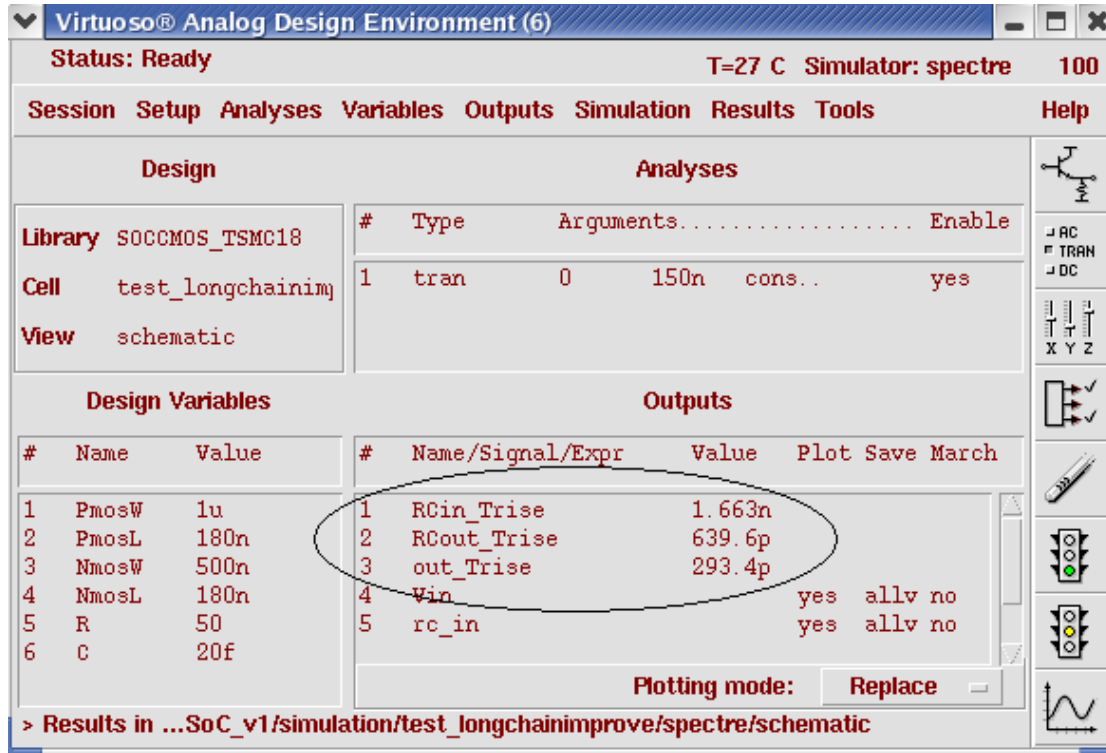
1 <sup>st</sup> inverter size	2 <sup>st</sup> inverter size	3 <sup>st</sup> inverter size	4 <sup>nd</sup> inverter size
w_nmos=0.5um	w_nmos=1um	w_nmos=2um	w_nmos=4um
w_pmos=1um	w_pmos=2um	w_pmos=4um	w_pmos=8um



We created cells inv\_2, inv\_4, inv\_8 the same way as inv\_1 the only difference was the definition of width.



Exactly the same analog environment was used for testing. Notice that we didn't have to add design variables to define the width of other inverters.



The following results were achieved:

RC stage in-delay	RC stage in-rise	RC stage out-delay	RC stage out-rise	out-delay	out-rise
735ps	1.663ns	1.25ns	639.6ps	173ps	293.4ps

As you see the results were improved compare to paragraph 2.