High Speed Bipolar Mixed Signal and Communication ICs
Exercise in Running the Simulation Tools and Introductory Circuits

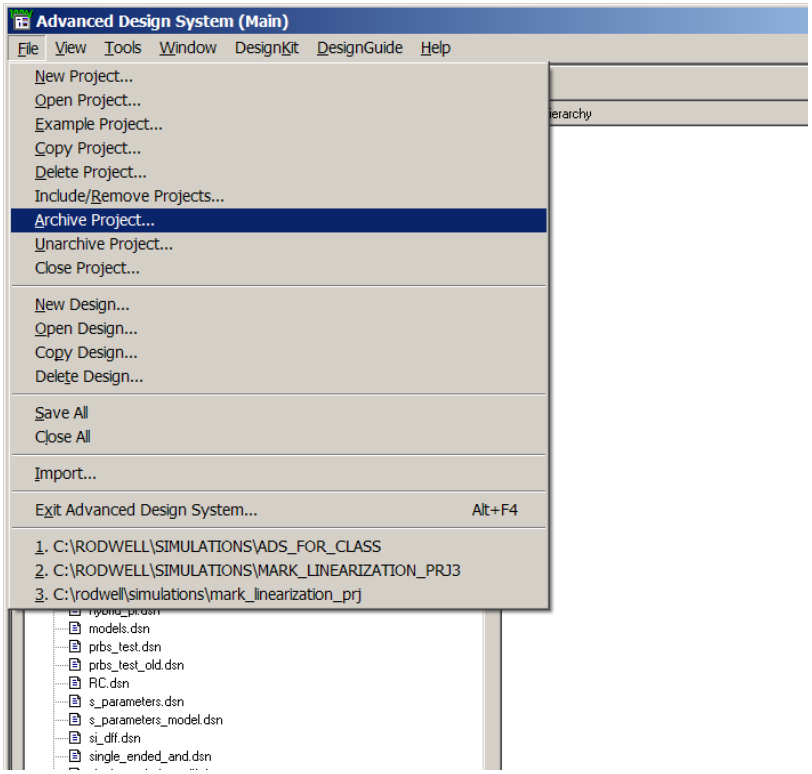
The exercises below are designed to **complement* your running the ADS tutorials (in ADS documentation), which are highly recommended.

First: Mechanics of Accessing the program.

I have set up some example ADS directories to aid in getting the tools running quickly.

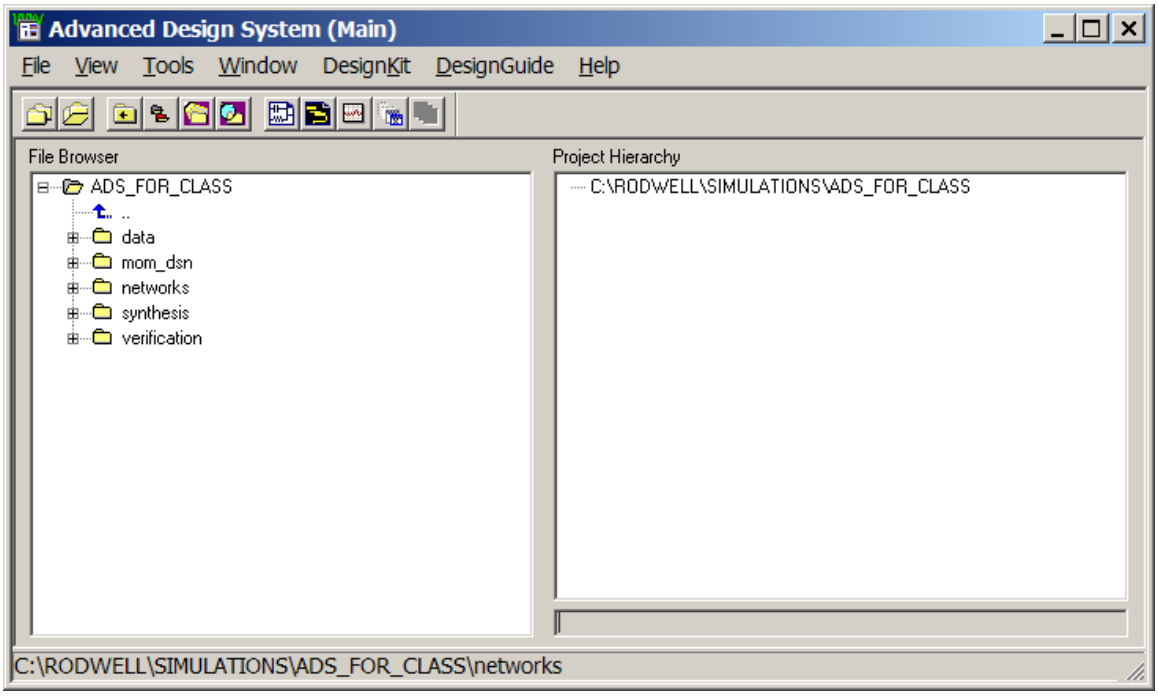
Downloading the ADS project directory

On my web page will be a compressed ADS project directory. This is in a ZAPPED format, which sounds like, but is not, a Zipped format. Download it using a web browser (save to disc) and then *unarchive project* to decompress it:



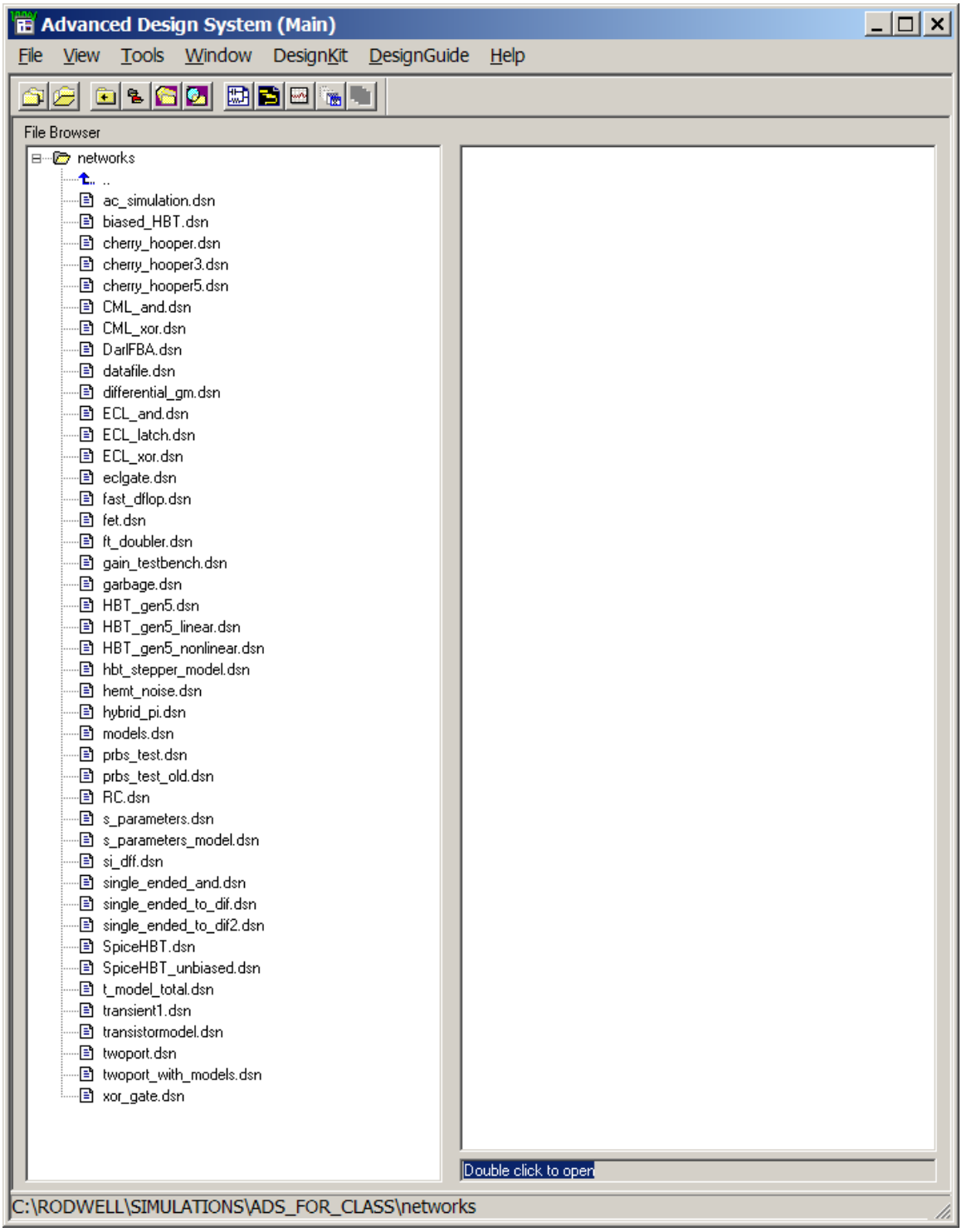
Examples of ADS simulations:

Download the project, Unarchive it as illustrated above, and open it. You should then see the following:



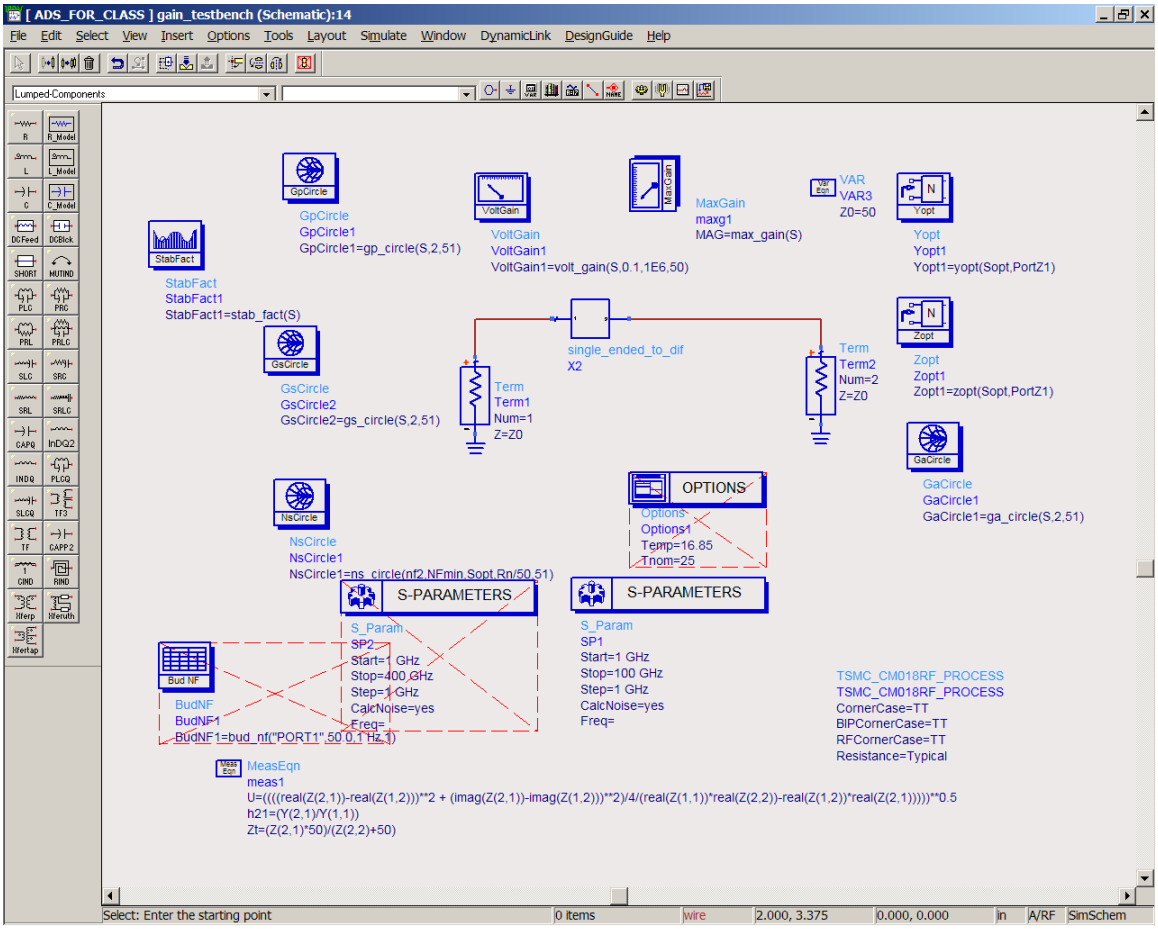
…you can open schematic editor window, by clicking on the button which looks like a schematic….but you may not need to: often the program opens itself in the state it was last saved.

You can expand the file browser window to see available circuits (I have given you lots):


Advanced Design System (Main)
File View Tools Window DesignKit DesignGuide Help
File Browser
networks
..
ac_simulation.dsn
biased_HBT.dsn
cherry_hooper.dsn
cherry_hooper3.dsn
cherry_hooper5.dsn
CML_and.dsn
CML_xor.dsn
DarlFBA.dsn
datafile.dsn
differential_gm.dsn
ECL_and.dsn
ECL_latch.dsn
ECL_xor.dsn
eclgate.dsn
fast_dflop.dsn
fet.dsn
ft_doubler.dsn
gain_testbench.dsn
garbage.dsn
HBT_gen5.dsn
HBT_gen5_linear.dsn
HBT_gen5_nonlinear.dsn
hbt_stepper_model.dsn
hemt_noise.dsn
hybrid_pi.dsn
models.dsn
prbs_test.dsn
prbs_test_old.dsn
RC.dsn
s_parameters.dsn
s_parameters_model.dsn
si_dff.dsn
single_ended_and.dsn
single_ended_to_dif.dsn
single_ended_to_dif2.dsn
SpiceHBT.dsn
SpiceHBT_unbiased.dsn
t_model_total.dsn
transient1.dsn
transistormodel.dsn
twoport.dsn
twoport_with_models.dsn
xor_gate.dsn
Double click to open
C:\RODWELL\SIMULATIONS\ADS_FOR_CLASS\networks

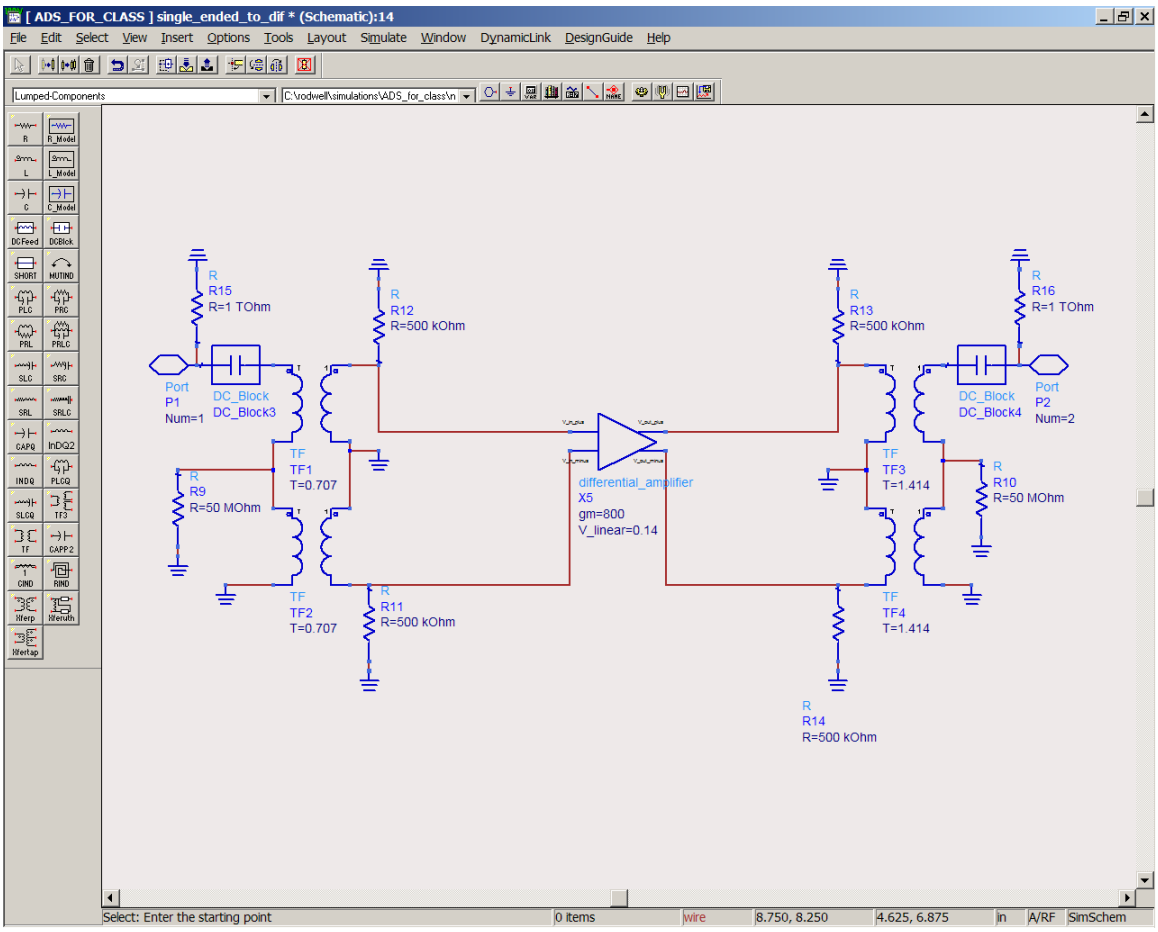
Differential S-parameter Simulation

Open up the file gain_testbench which I have created to calculate S-parameters:

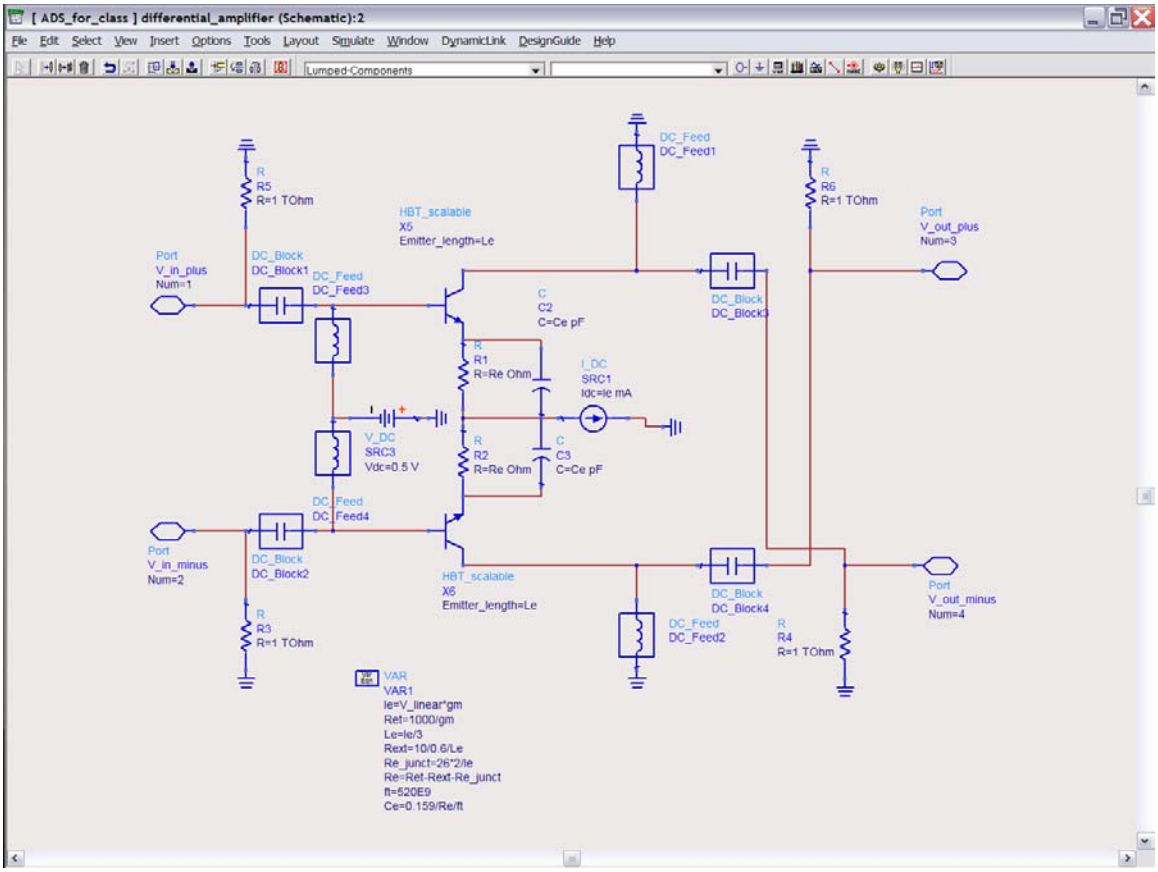


This will test for Sparameters of the subcircuit "single_ended_to_dif". You can type the name of other subcircuits there to test them instead.

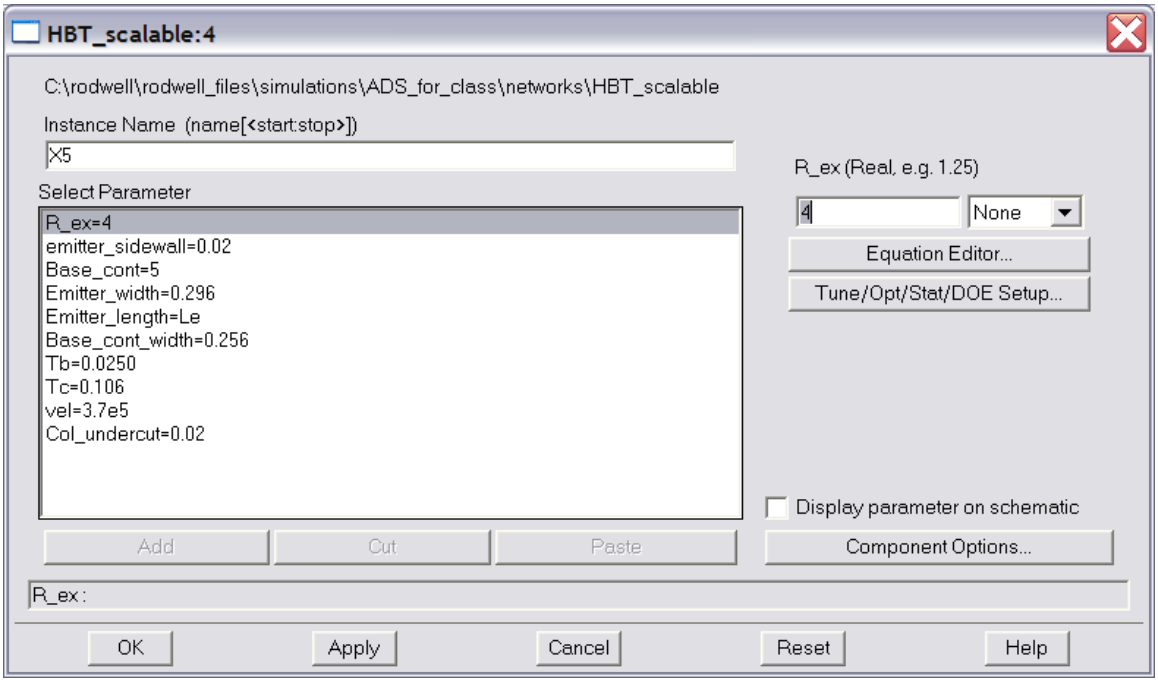
You will see a "down arrow" button which is used to pop into the network hierarchy. Click on it, then on the "single_ended_to_dif" subcircuit to move into the hierarchy.



This is a utility block I created, which allows one to simulate for S parameters of a differential circuit (in this case the circuit is differential_amplifier). This is a trick to overcome a shortcoming in ADS: it is otherwise hard to make it compute S parameters of differential circuits. Pop into it, and you see a common form of high frequency amplifier:


[ADS_for_class] differential_amplifier (Schematic):2
File Edit Select View Insert Options Tools Layout Simulate Window DynamicLink DesignGuide Help
Lumped-Components
DC_Feed
DC_Feed1
R
R5
R=1 TOhm
R
R6
R=1 TOhm
HBT_scalable
X5
Emitter_length=Le
Port
V_out_plus
Num=3
Port
V_in_plus
Num=1
DC_Block
DC_Block1
DC_Feed
DC_Feed3
C
C2
C=Ce pF
DC_Block
DC_Block3
R
R1
R=Re Ohm
I_DC
SRC1
Idc=Ie mA
V_DC
SRC3
Vdc=0.5 V
R
R2
R=Re Ohm
C
C3
C=Ce pF
DC_Feed
DC_Feed4
Port
V_in_minus
Num=2
DC_Block
DC_Block2
HBT_scalable
X6
Emitter_length=Le
DC_Block
DC_Block4
Port
V_out_minus
Num=4
DC_Feed
DC_Feed2
R
R4
R=1 TOhm
R
R3
R=1 TOhm
VAR
VAR1
Ie=V_linear*gm
Ret=1000/gm
Le=le/3
Rext=10/0.6/Le
Re_junct=26*2/Ie
Re=Ret-Rext-Re_junct
ft=520E9
Ce=0.159/Re/ft

IMPORTANT: select the transistor, and double click on it and make sure that the following model parameters are set for *all* devices which you use in your circuit:

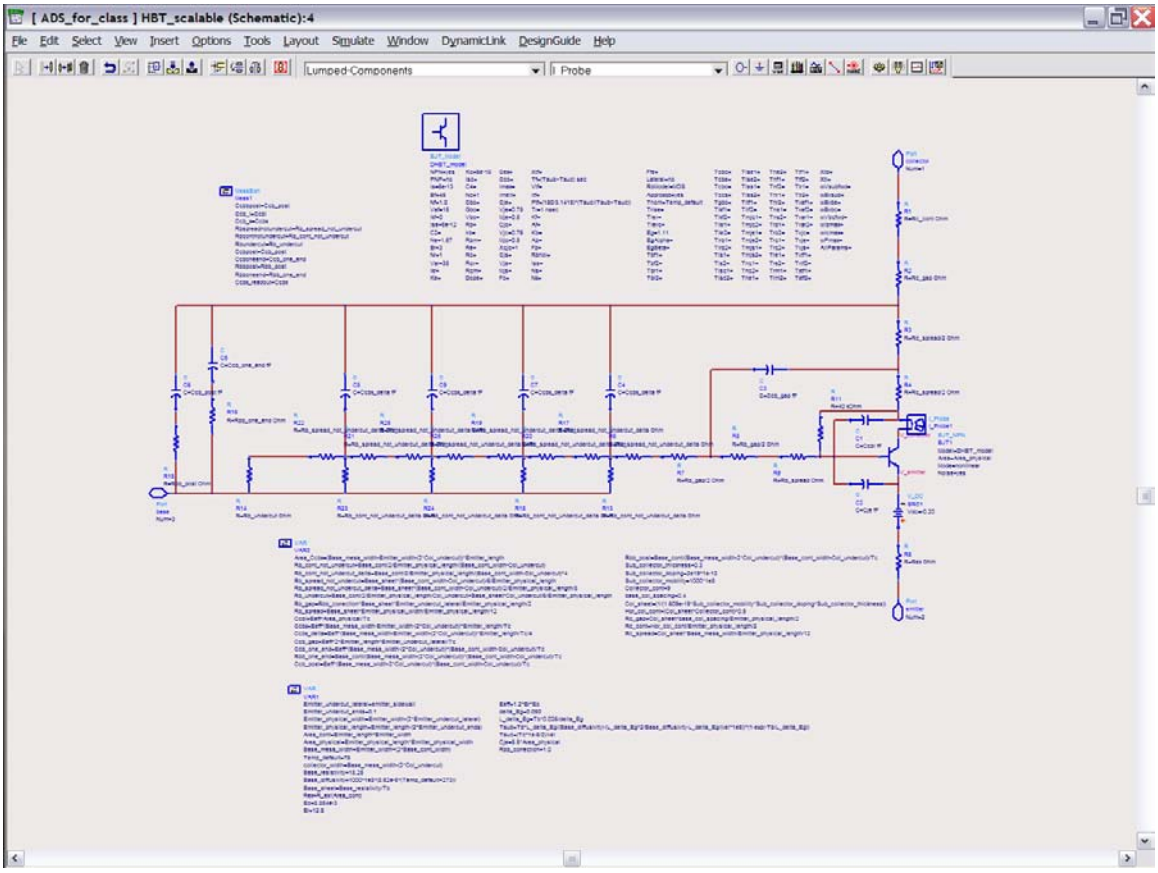


In this list
Rex = 4 Ohm-micron^2 is the parasitic emitter resistance per unit area
emitter_sidewall=0.02 micron is the emitter sidewall thickness
base_cont= 5 Ohm-micron^2 is the base contact resistance per unit base contact area
emitter width=0.296 is the emitter junction width*
the physical emitter junction width is 0.296-2(emitter sidewall)=0.256 um
emitter length, here set = to the variable Le, is the emitter length in microns
base_cont_width=0.256 micron is the base contact width
Tb=0.025 is the base thickness in microns
Tc=0.106 is the collector depletion layer thickness in microns
vel=3.7e5 is the collector electron saturation drift velocity in m/s
Col_undercut =0.02 micron is the base-collector junction undercut
the width of the base-collector

In all of your circuits, **all** of the parameters should be set to the above values, except the emitter length Le, which you may set to 1, 2 , 4, or 8 microns. For convenience, and clarity in the circuit diagram, only the emitter length is shown by default. This has the danger that you may be inadvertently setting these transistor parameters to undesired values !

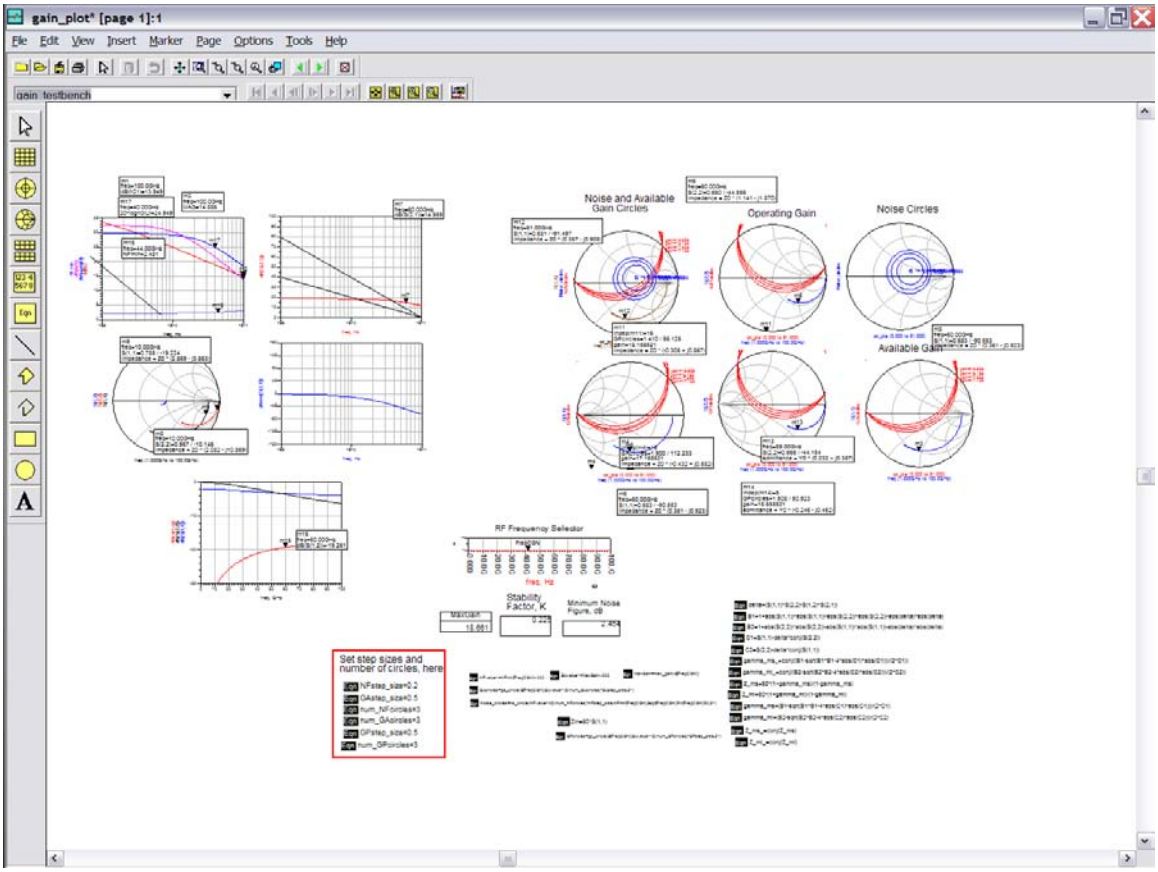
The device has a maximum operating current of ~ 3mA/micron * Le, and a maximum power dissipation of about 4 mW/micron*Le

If you pop into the transistor:



The transistor is defined as a SPICE model (DHBT model), with external parasitic resistances and capacitances which we have defined through equations from the transistor geometry. We do this at UCSB because it allows us to make fairly accurate predictions of performance of future transistors which we may not yet have built.

Now pop back up to the top level in the hierarchy and then "Simulate" , and (if it does not happen automatically), open up a plot of results (window→open data display→ "gain_testbench") to open up a set of plots I have pre-created:



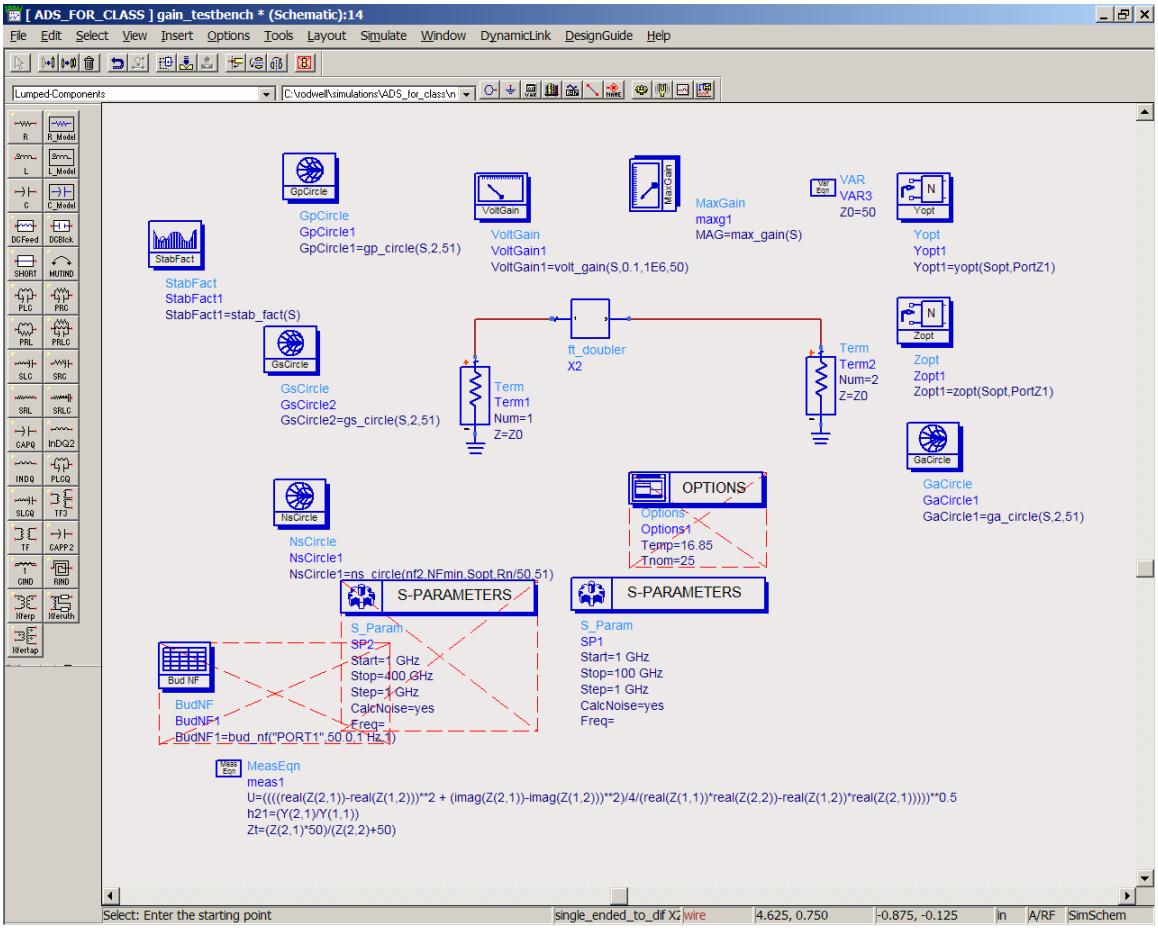
There are dB and polar plots of S-parameters, and plots of H21 and U (relevant for transistors, no relevant for the IC). There are also noise figure circles and gain circles, which are relevant to microwave tuned amplifier design.

If you change pages (push the green arrow buttons on the menu bar), there are also plots of Y and Z and S parameters, and plots of open-circuit voltage gains, etc.

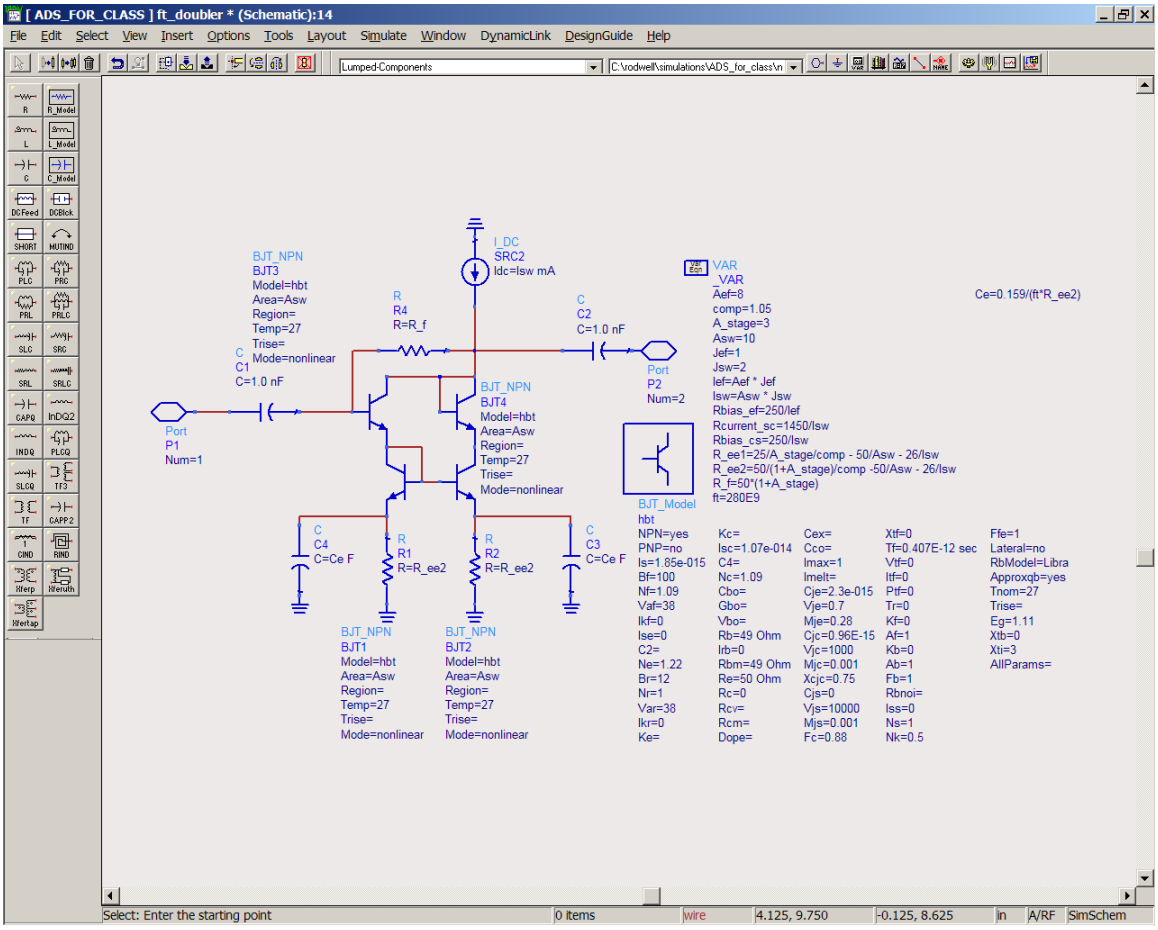
Click on the toolbar on the left-hand side, and you can add other plots.

S-parameter simulation single-ended

We can go back to the s_parameters circuit and edit the circuit under test to ft_doubler:

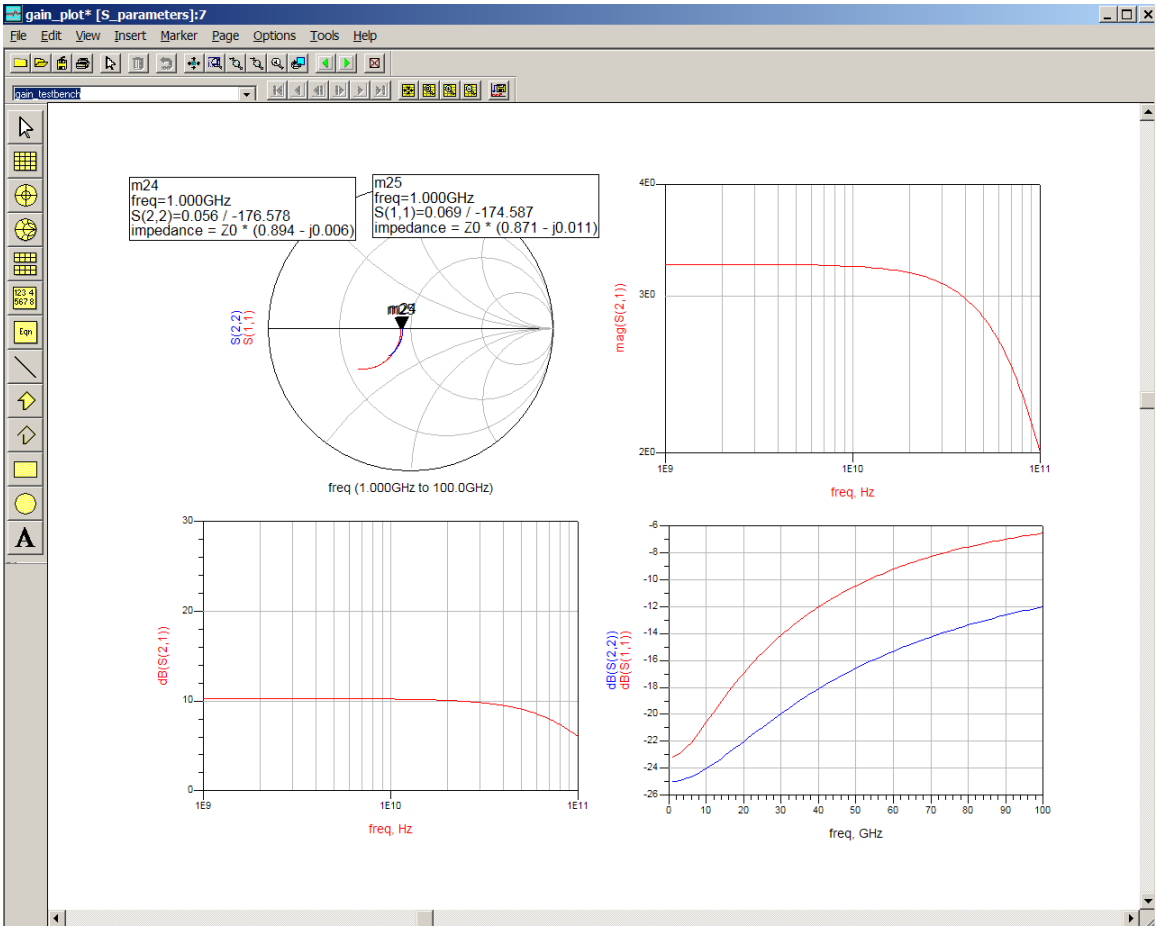


Pop into this (single_ended) circuit and see the amplifier:



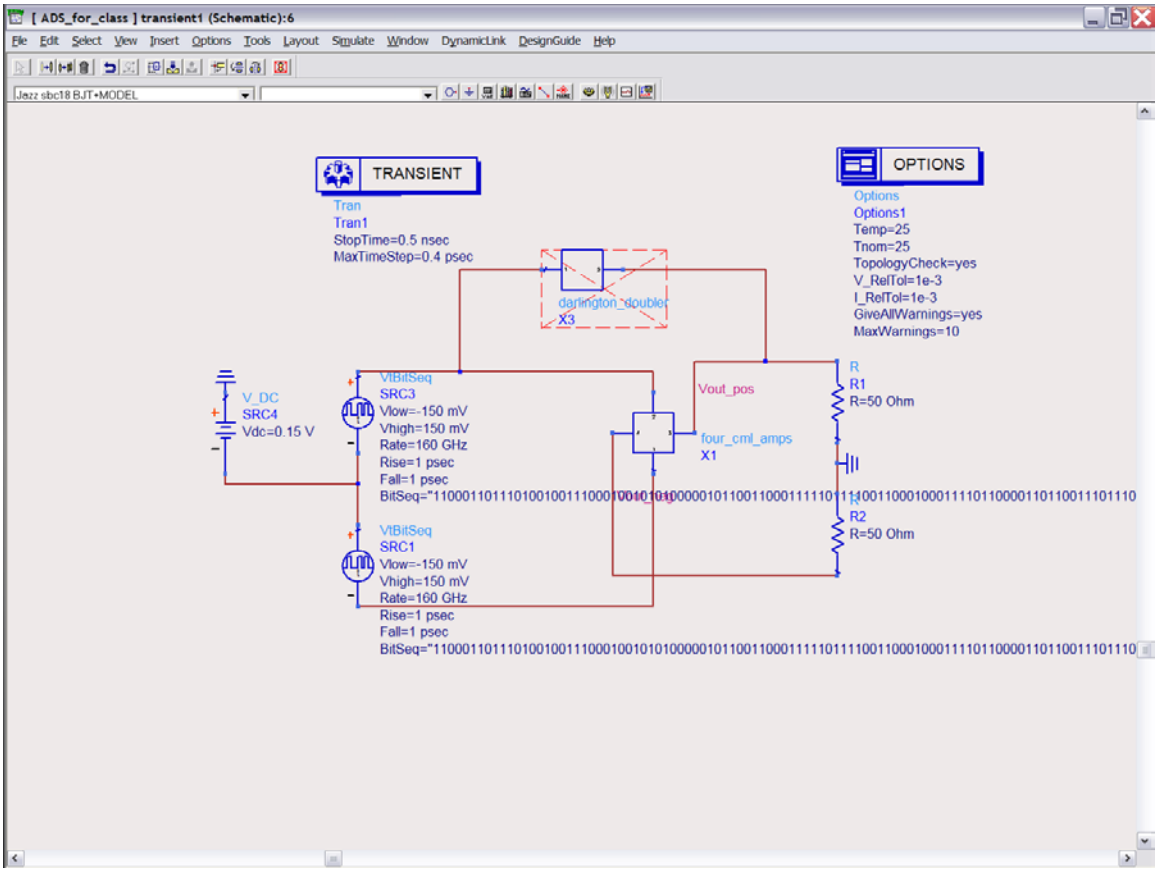
Note that a different transistor model is here used; there is no subcircuit with an associated set of equation-defined parasitics. This is the more common approach in IC design.

Pop back out, and hit simulate:



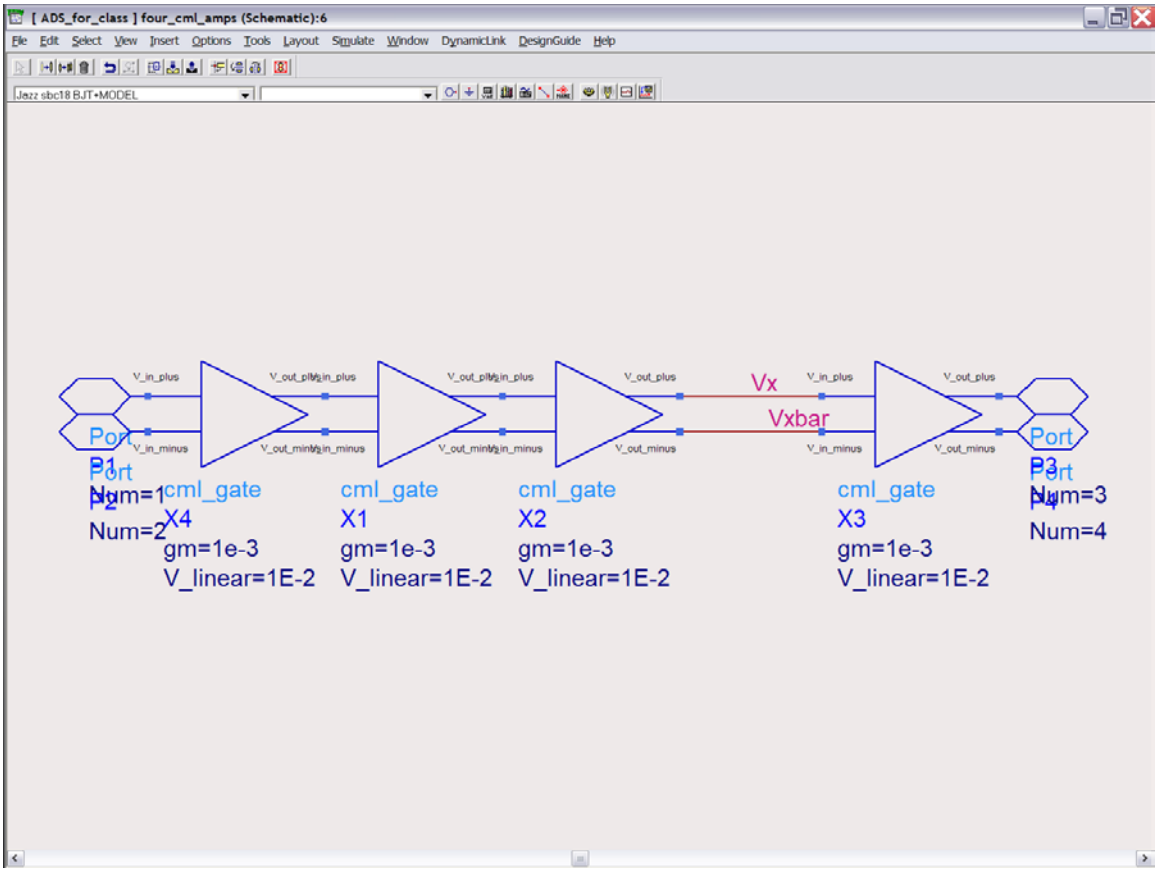
Transient Simulation and Eye Patterns--differential.

Now open up the test bench transient 1 (and zoom in a bit !)

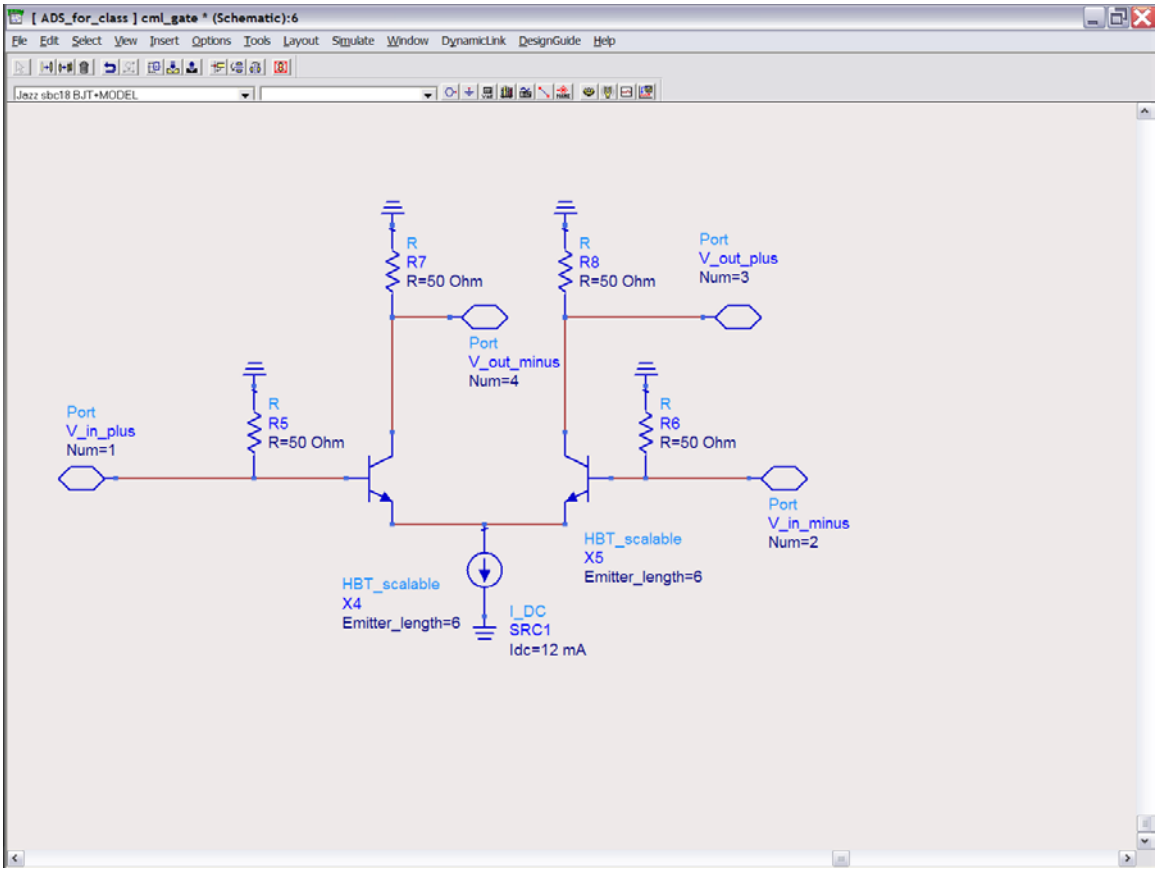


You will see a couple of circuits. Darlington_doubler is a single-ended circuit, and is currently "inactivated". There is a differential circuit, "four_cml_amps", which is active. There is a pair of (random) data string generators, which are logical complements of each other. For square-wave testing you could replace these with e.g. a square wave generator.

Pop into " four_cml_amps ". It is a set of four cascaded amplifiers, each of which is a simple CML logic gate, which we are operating here as simple differential limiting amplifier.

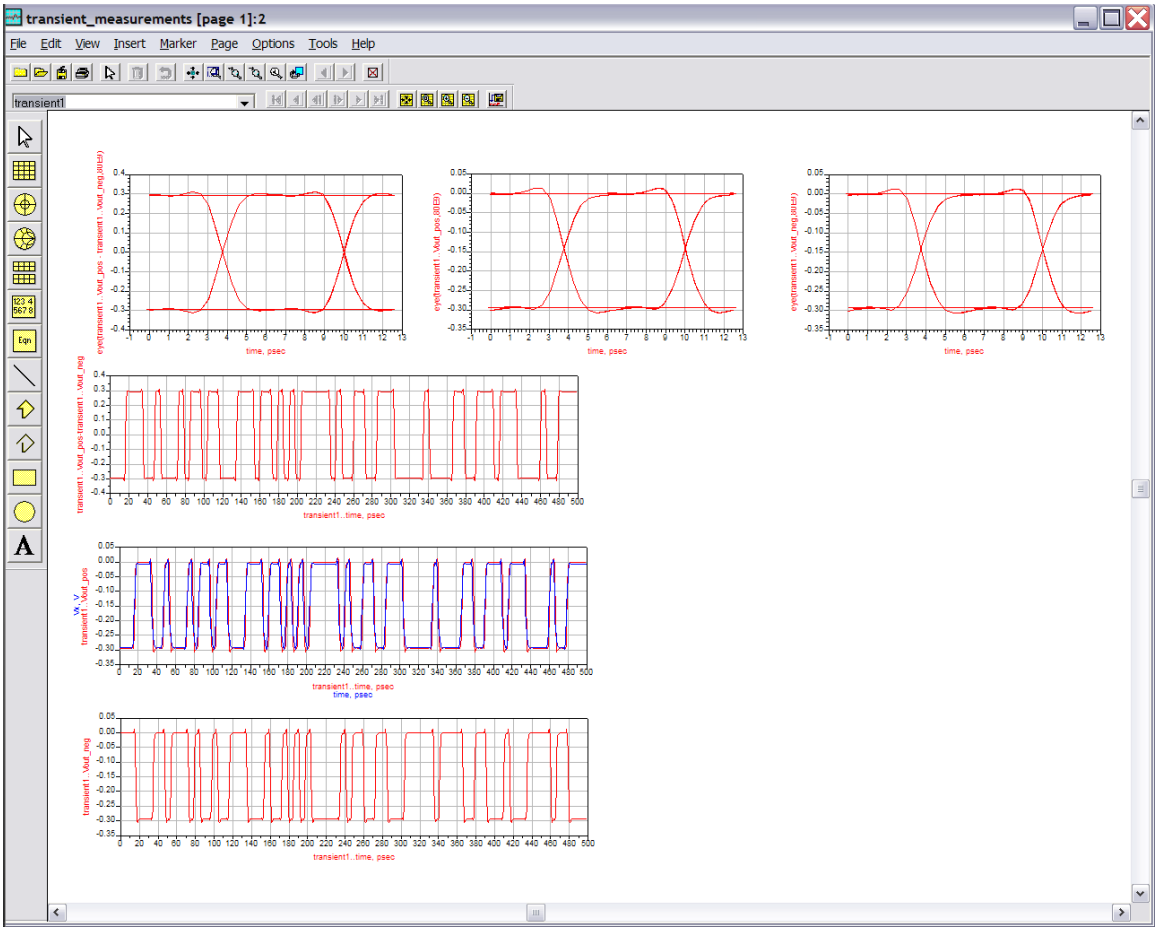


You can pop in again to see the circuit:

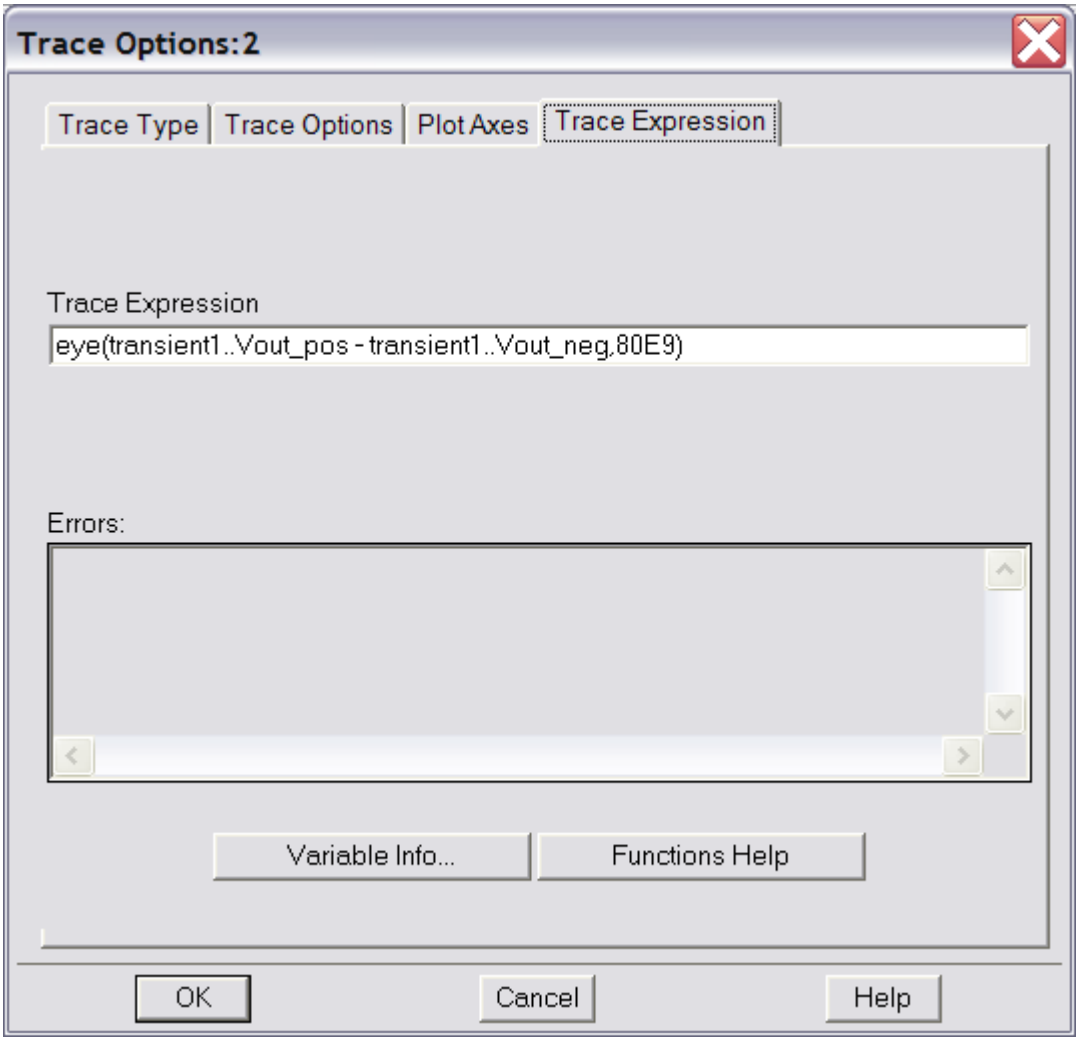


Pop back out to the highest level, and simulate. Then open up the graph window "transient measurements 1"

…and you can see the eye pattern and pulse response of Vout of the circuit:



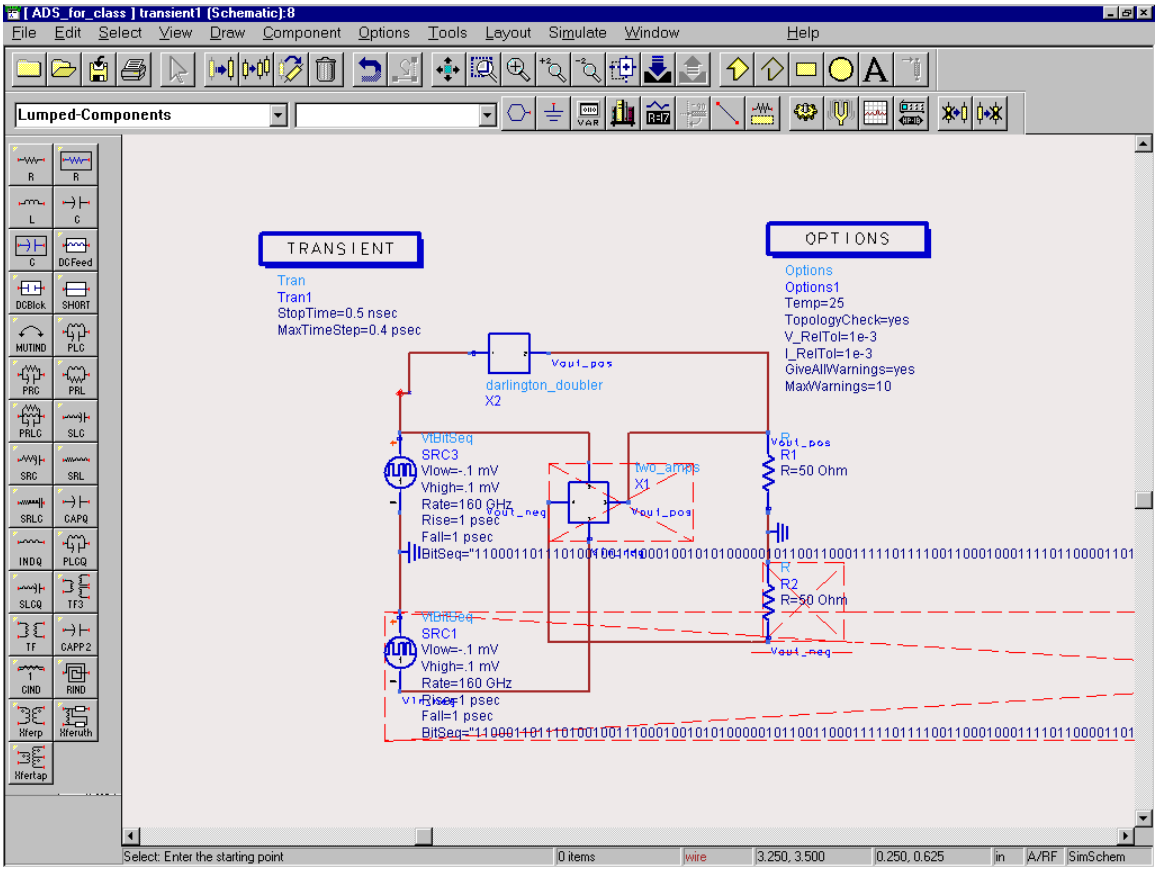
Clicking on the vertical axis of the eye pattern display reveals the syntax necessary to plot an eye pattern:



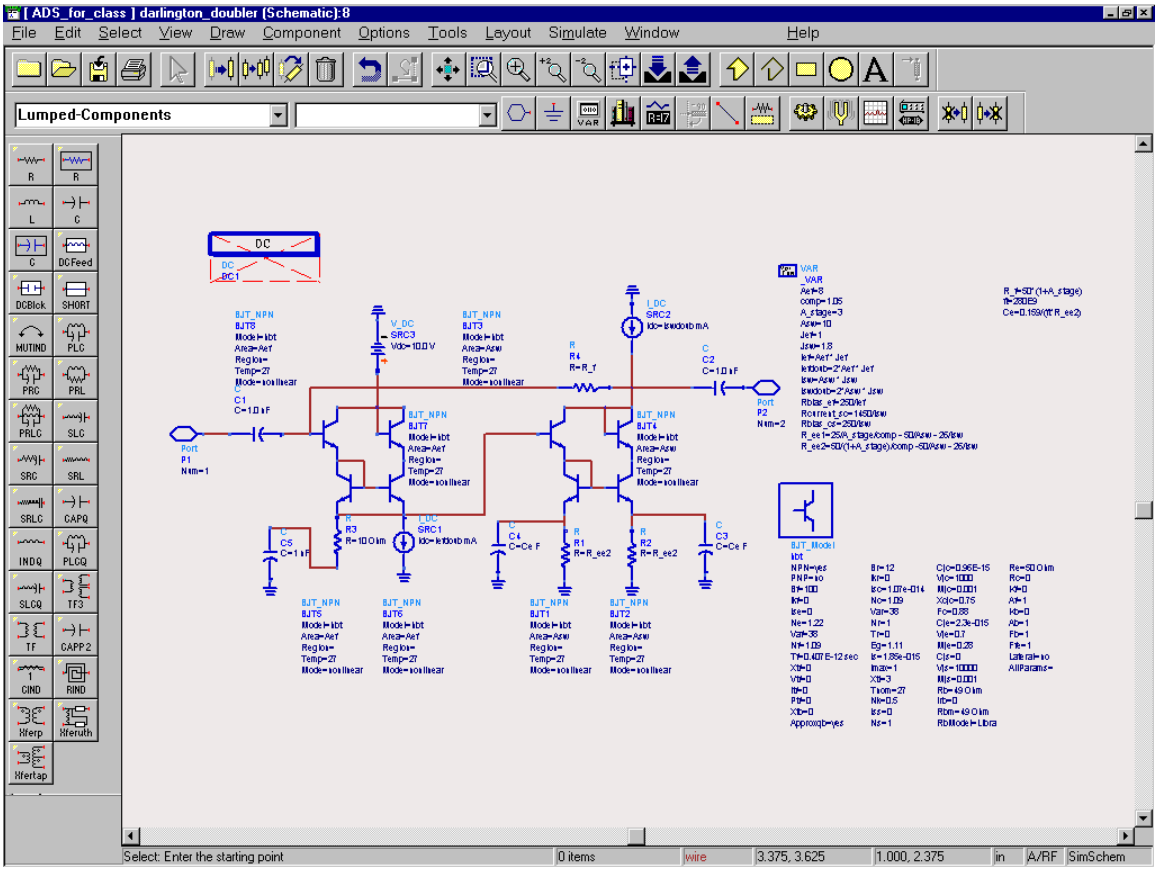
Note that the last number 80E9, is set at **1/2 the bit rate** of the simulation.

Transient Simulation and Eye Patterns--single-ended

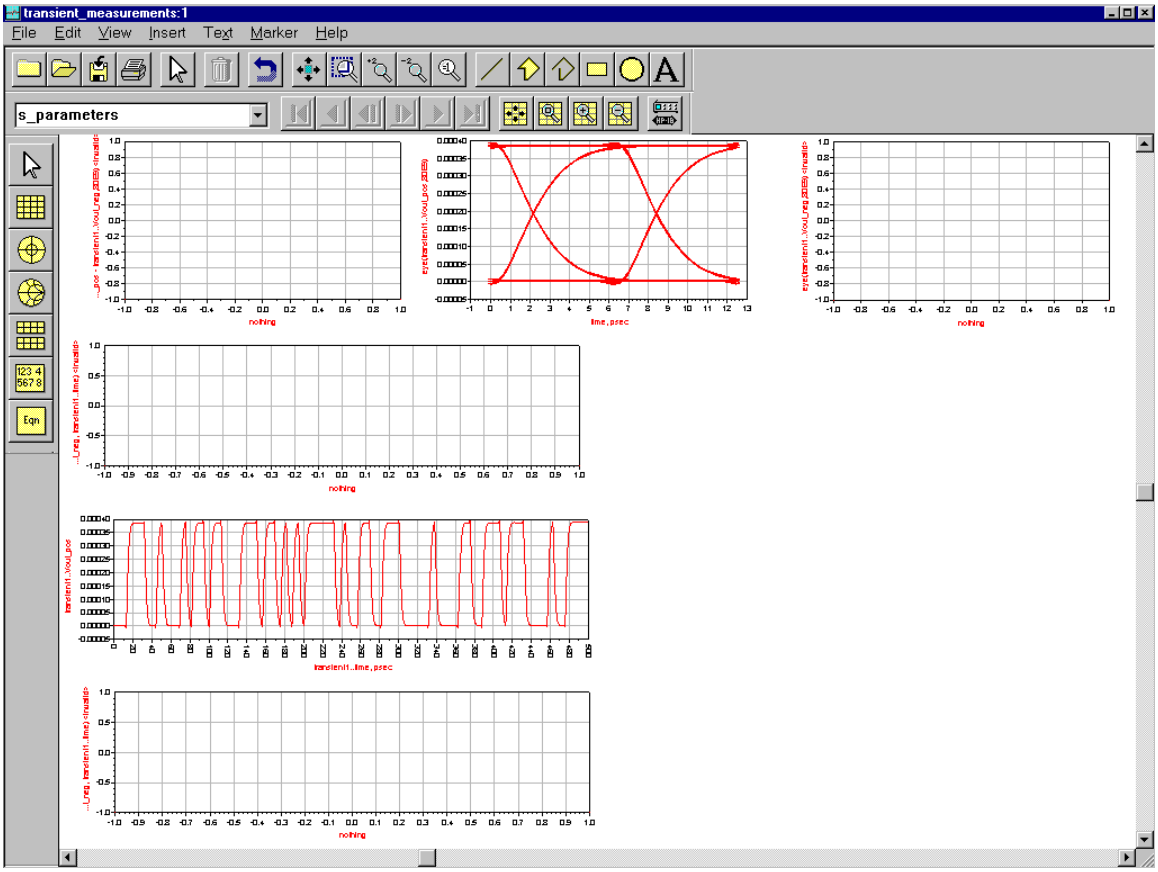
To simulate a single-ended circuit, we activate and deactivate a few components:



Pop into Darlington_doubler to take a look (a fairly exotic circuit block)



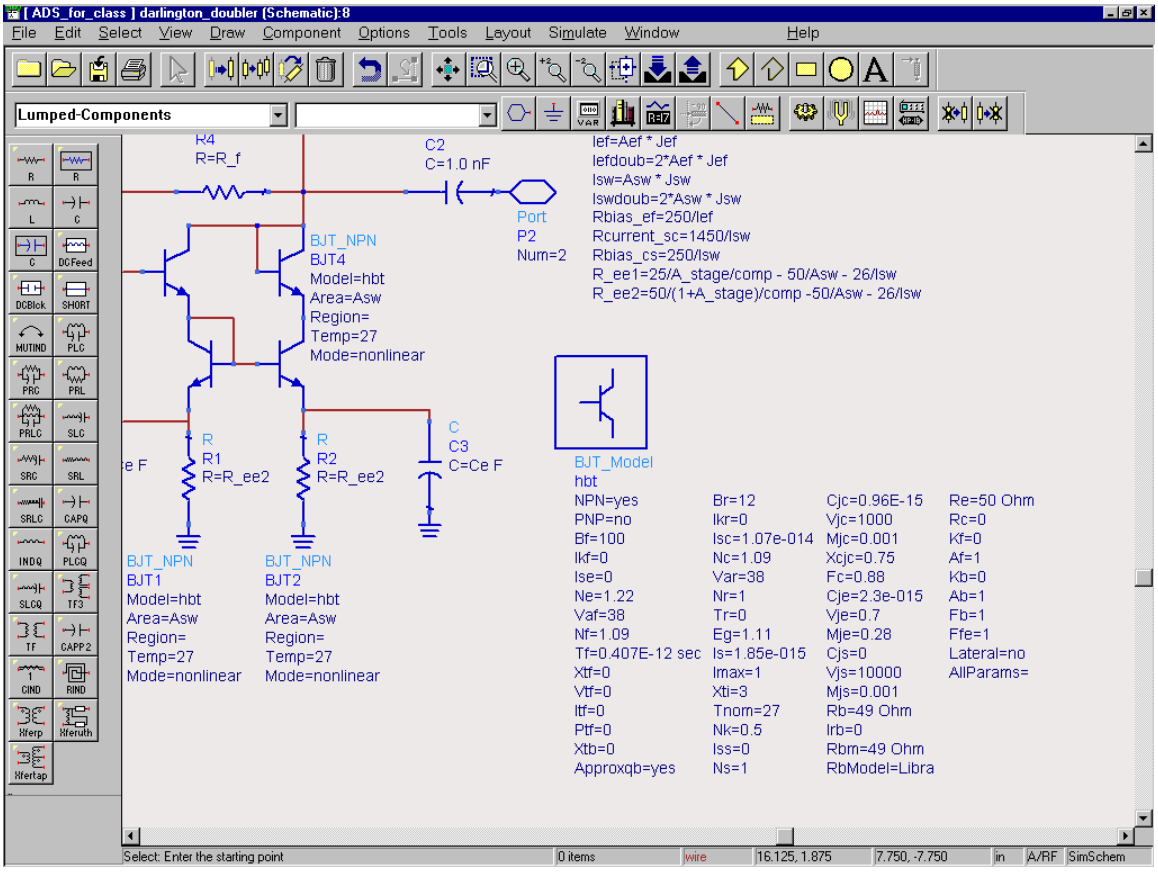
And pop back up and re simulate:


transient_measurements:1
File Edit View Insert Text Marker Help
s_parameters
Eqn
nothing
time, psec

Modeling Bipolar Transistors:

We can do this several ways:

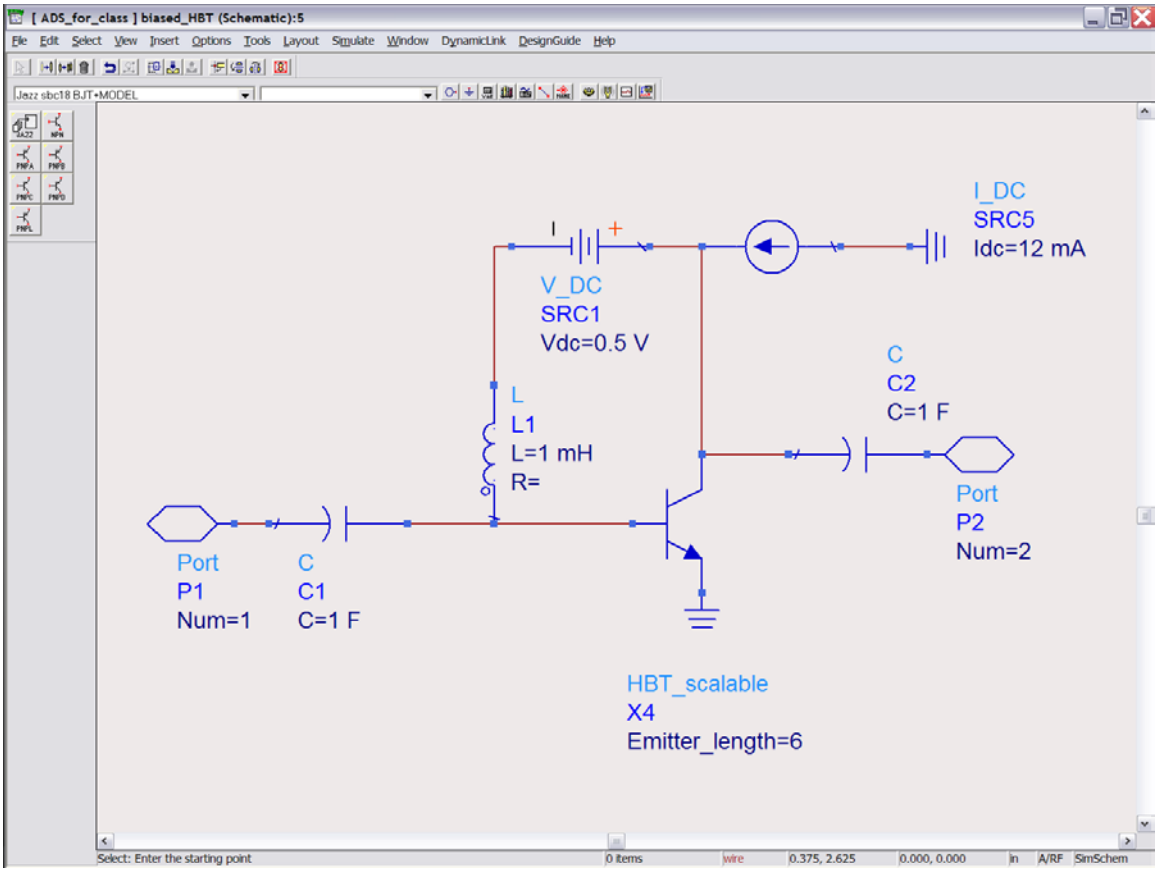
Spice MODEL



We have a model file, BJT_MODEL, which we then invoke with an area statement. This is a large signal model which defaults to a small signal model (with bias-dependent parameters) when we do a S-parameter simulation.

Subcircuit model with SPICE MODEL within

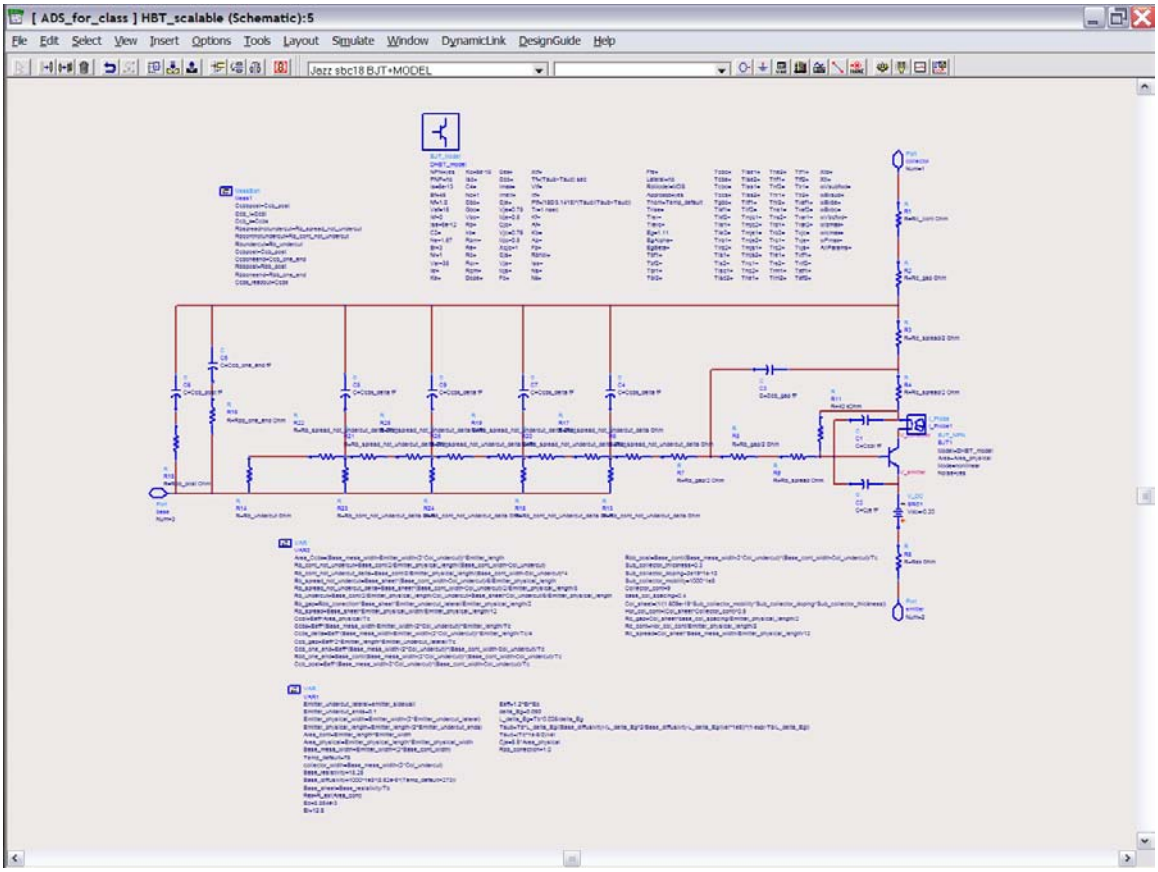
Here is a circuit (HBT under test)



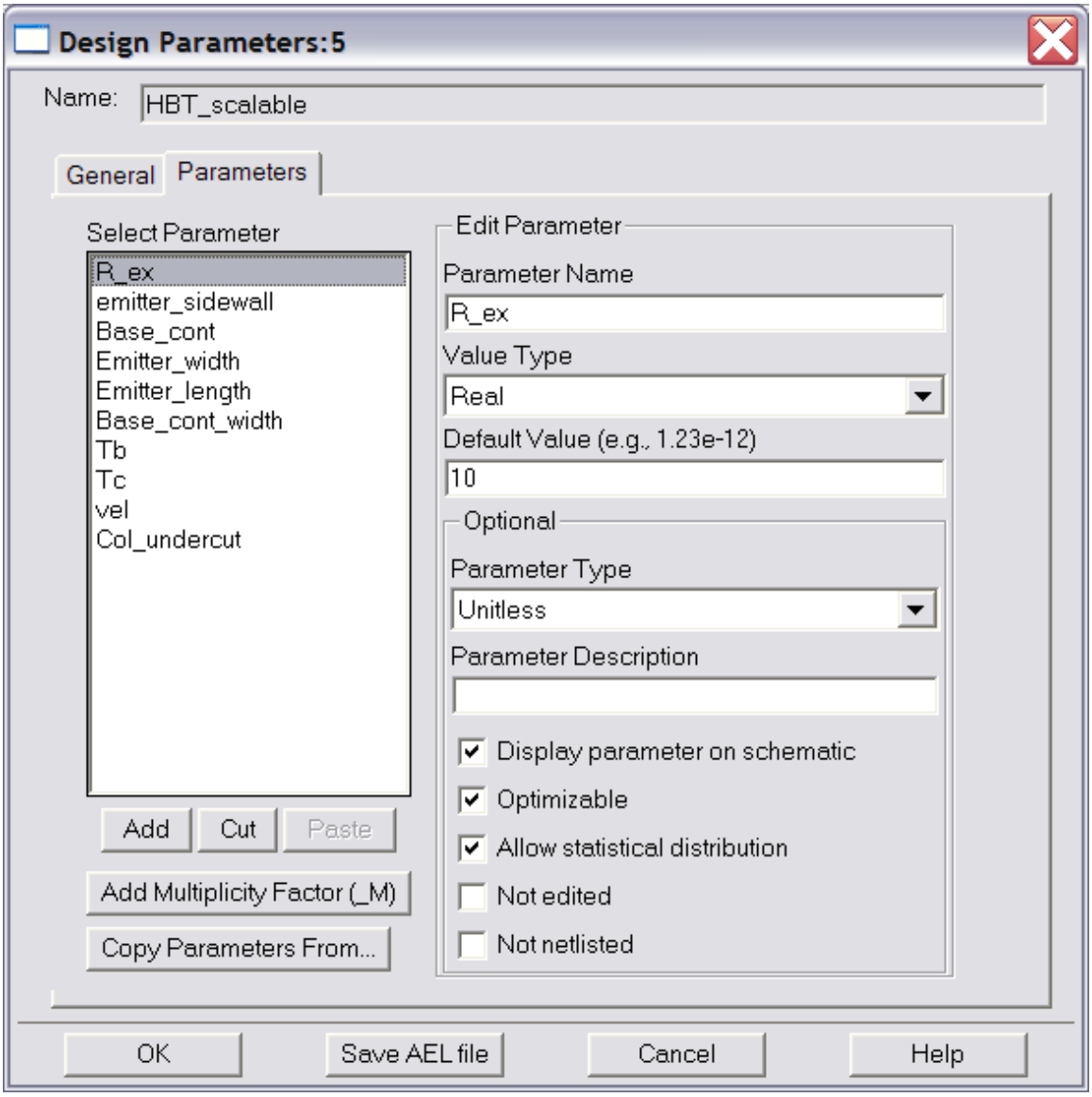
HBT_scalable is a subcircuit with PASSED parameters

Rex is the emitter resistance per unit area,
base_cont is the base contact resistivity in ohm-micron^2,
emitter width and emitter length are the length and width of the emitter in microns,
base_cont_width is the base contact width in microns
Tb is the base thickness in microns
Tc is the collector thickness in microns.
vel is the collector electron velocity
col_undercut is the undercut under the base contacts.

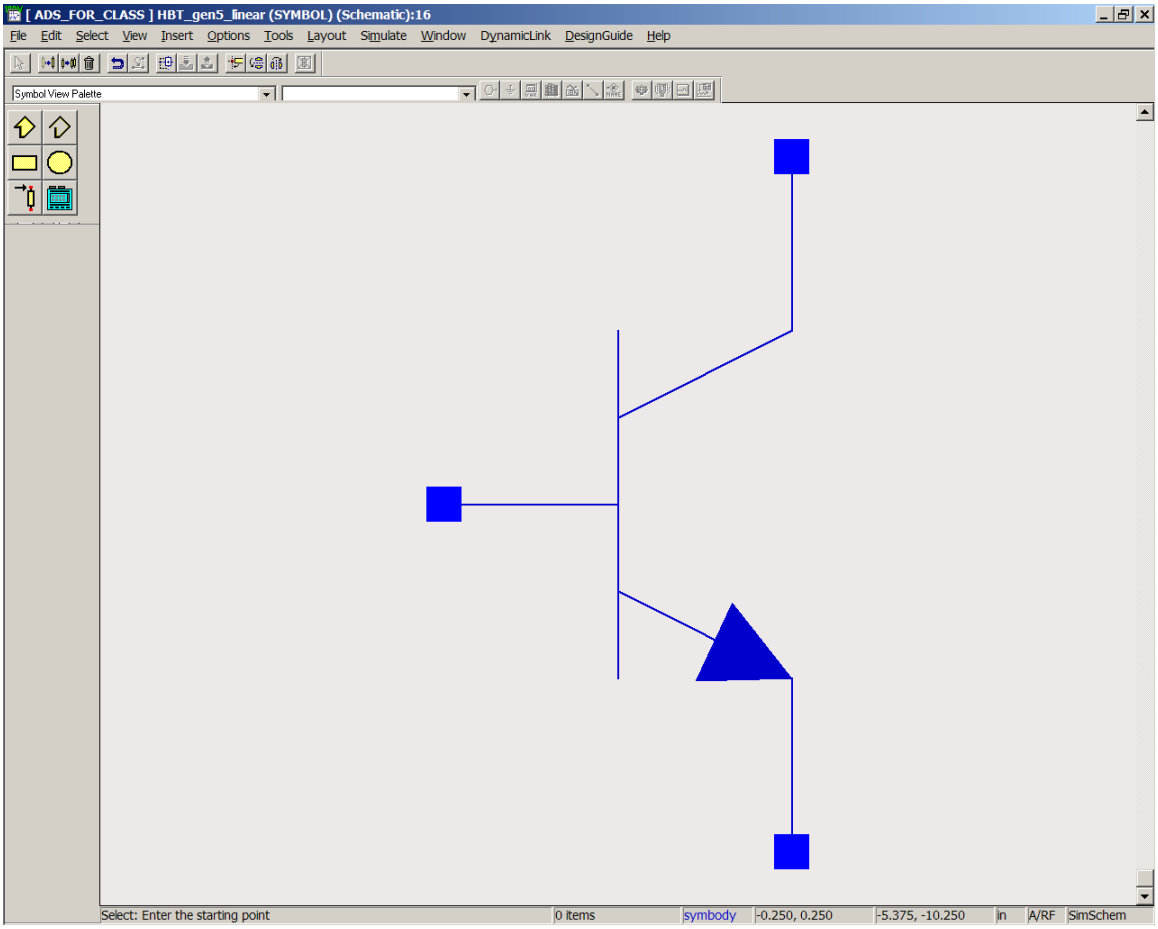
Popping into this model, as we did earlier, and you see the underlying definition based upon a SPICE model, with some additional parasitics and most of the device R's C's, and tau's defined in terms of the device physical dimensions:



Go to file→design parameters, and you will see how the passed parameters are defined (you should not need to play with this)

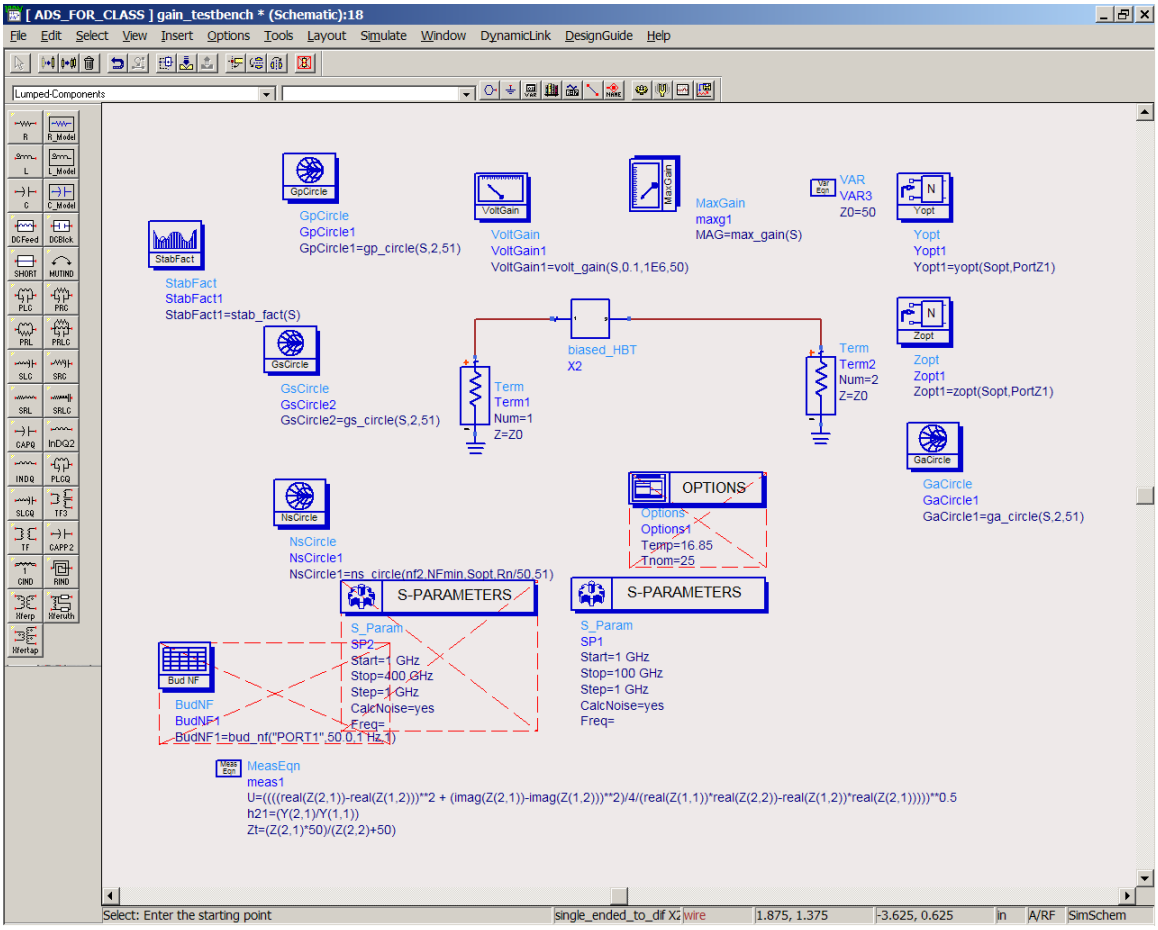


Go to view→ create/edit Schematic symbol and you will see how the subcircuit is given a pretty symbol:

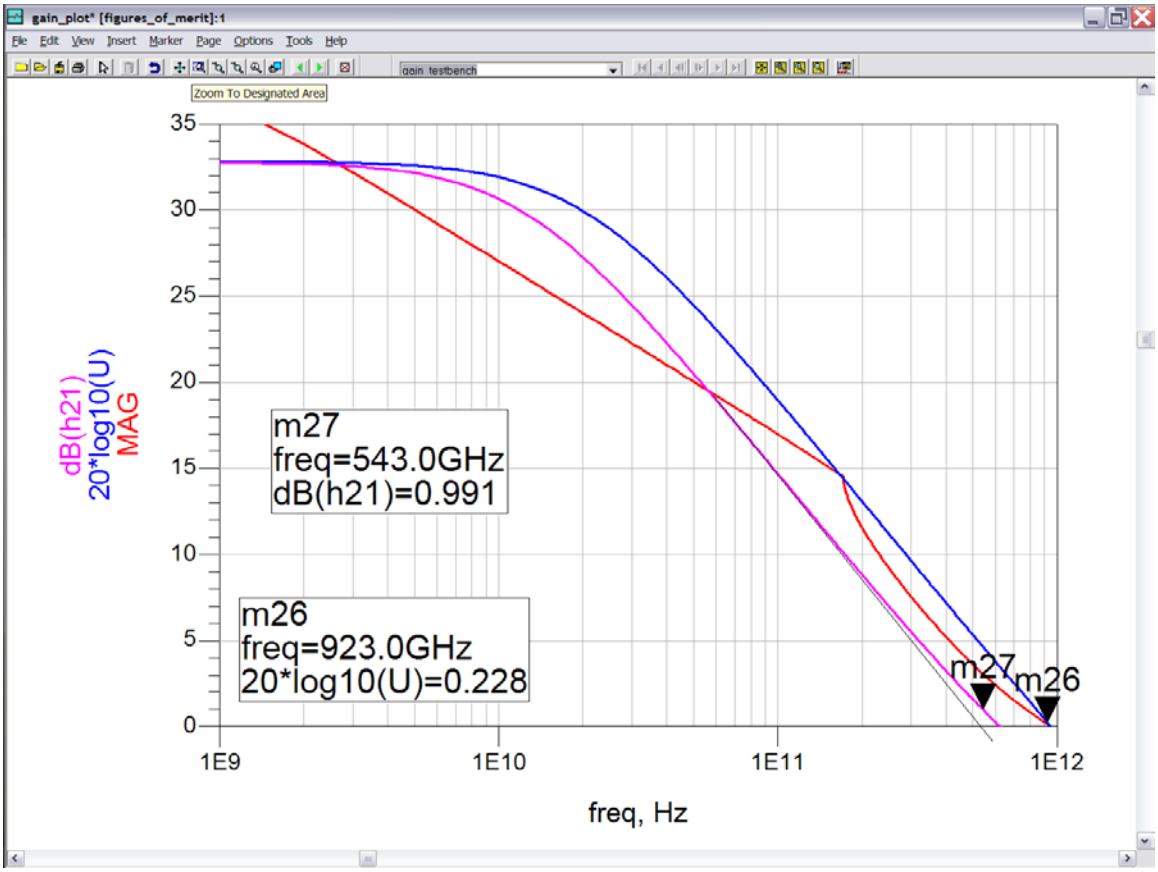


You can edit such symbols, but do so for a backed-up file first, as the editor is somewhat hard to use.

Pop back out to biased_HBT, then open the circuit gain_testbench, and change the DUT to biased_HBT:



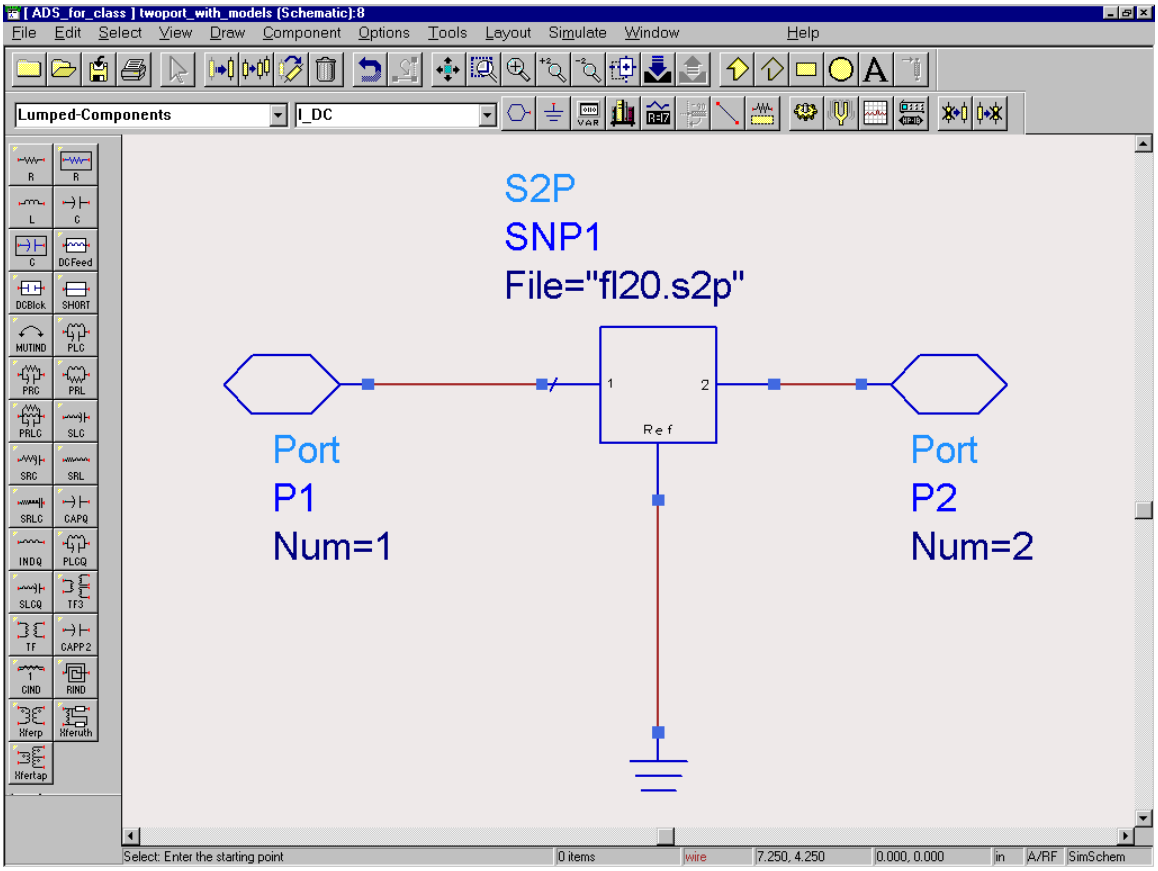
I have increased the frequency range to 1-1000 GHz, and on the figures_of_merit page of the gain plot we can now see the transistor high-frequency gains:



The device has an ft around 540 GHz and an fmax around 900 GHz.

S-parameter Data File

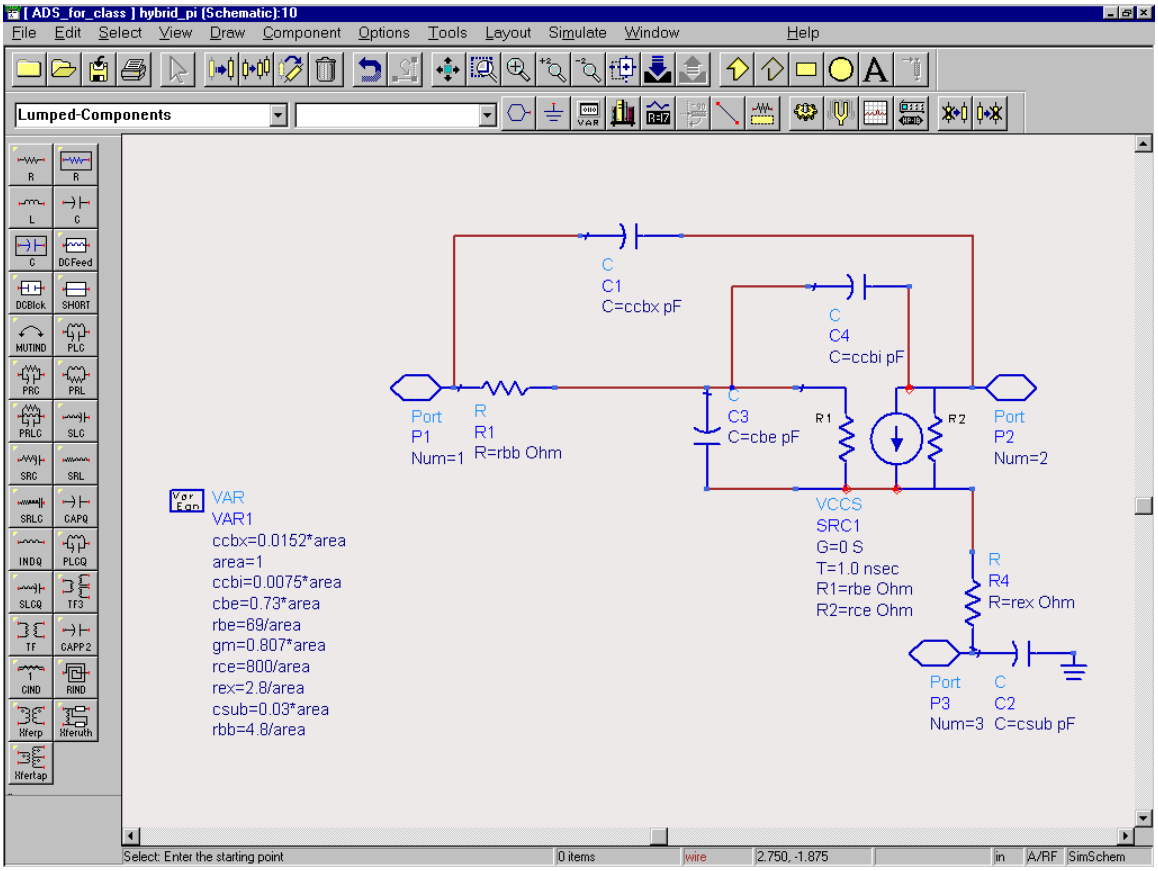
If you have measured data in tabular S parameter form, you can simulate from that



this is FYI only: I have not provided any such data set for you to use.

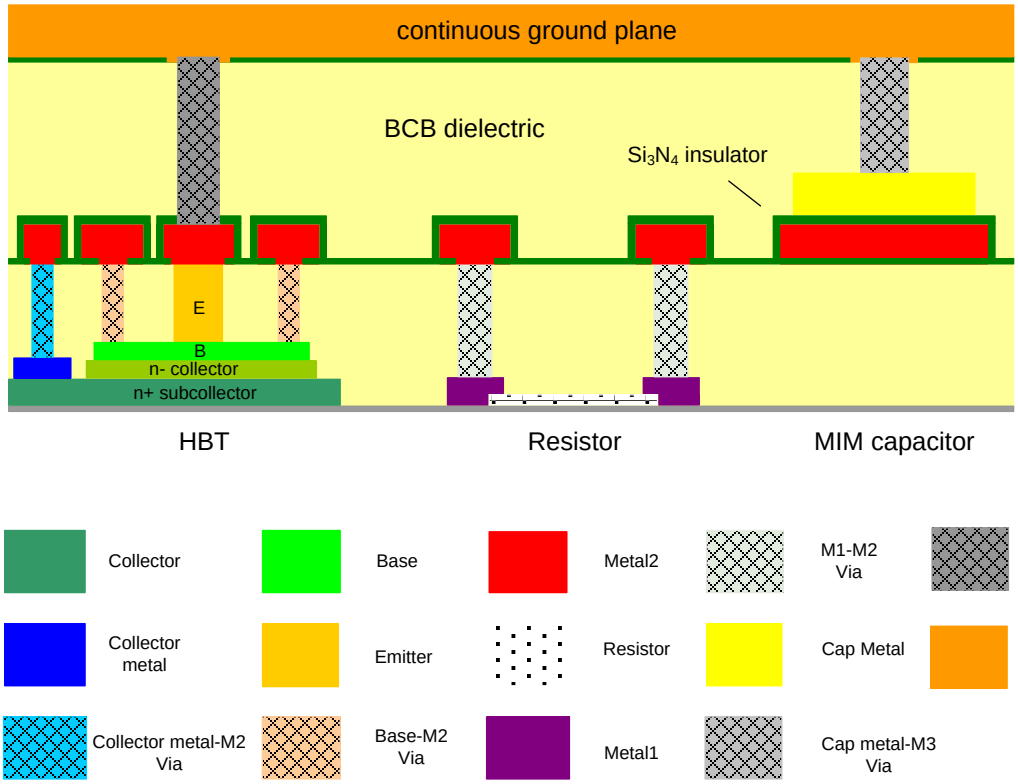
Small-Signal Models

You can directly create hybrid pi models by method. Of course, there is no DC information being carried in this simulation model.


[ADS_for_class] hybrid_pi [Schematic]:10
File Edit Select View Draw Component Options Tools Layout Simulate Window Help
Lumped-Components
C
C1
C=ccbx pF
C
C4
C=ccbi pF
Port
P1
Num=1
R
R1
R=rbb Ohm
C
C3
C=cbe pF
R1
R2
Port
P2
Num=2
VAR
VAR1
ccbx=0.0152*area
area=1
ccbi=0.0075*area
cbe=0.73*area
rbe=69/area
gm=0.807*area
rce=800/area
rex=2.8/area
csub=0.03*area
rbb=4.8/area
VCCS
SRC1
G=0 S
T=1.0 nsec
R1=rbe Ohm
R2=rce Ohm
R
R4
R=rex Ohm
Port
P3
Num=3
C
C2
C=csub pF
Select: Enter the starting point
0 items
wire
2.750, -1.875
in
A/RF
SimSchem

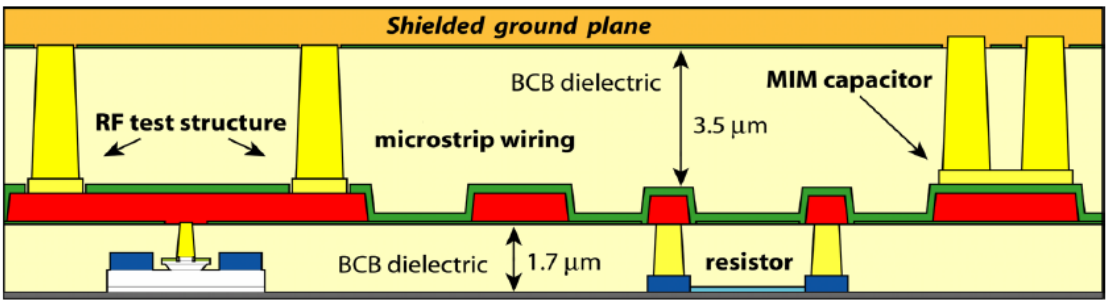
Interconnects

Process cross-section



The process provides three metal layers, M1, M2, and M3. Normally, M3 will be used as a ground plane and M1 or M2 as signal lines.

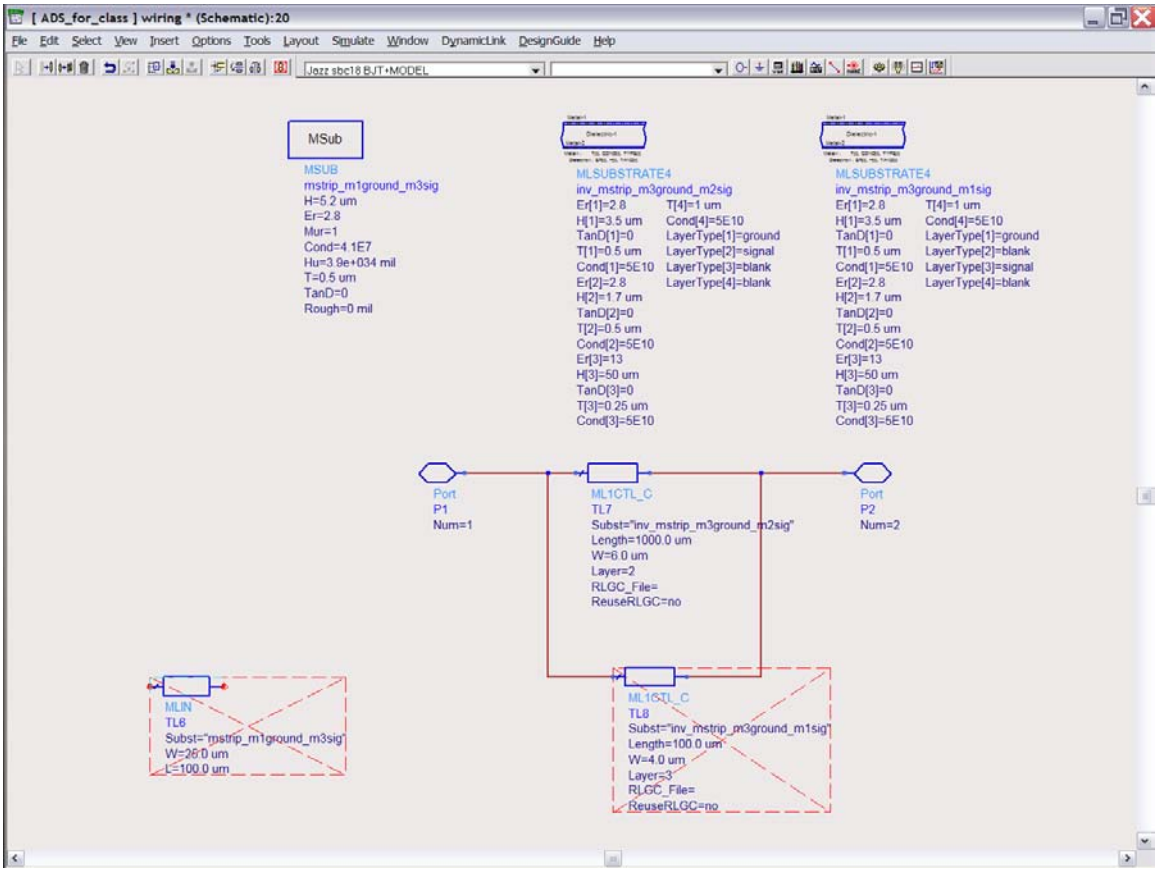
The cross-section below may also be helpful:



Note the thicknesses of the two BCB layers, and note that BCB has a dielectric constant of 3.8. Below the plane of the HBT subcollector, and below the resistors and metal 1, lies the semi-insulating InP substrate, having a dielectric constant of 13. We will approximate that M1 and M2 are each 1/2 micron thick.

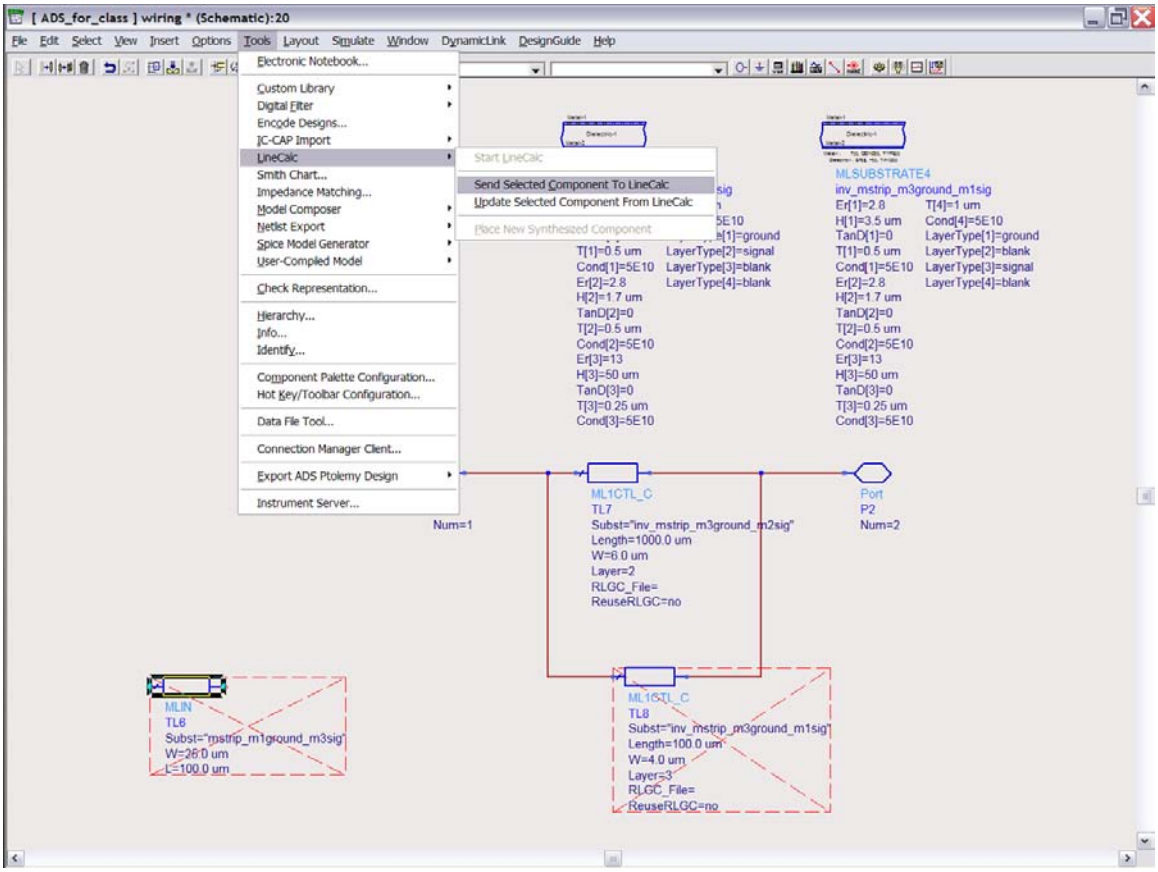
Interconnect models (roughly)

These interconnects can be ***roughly*** modeled as below:

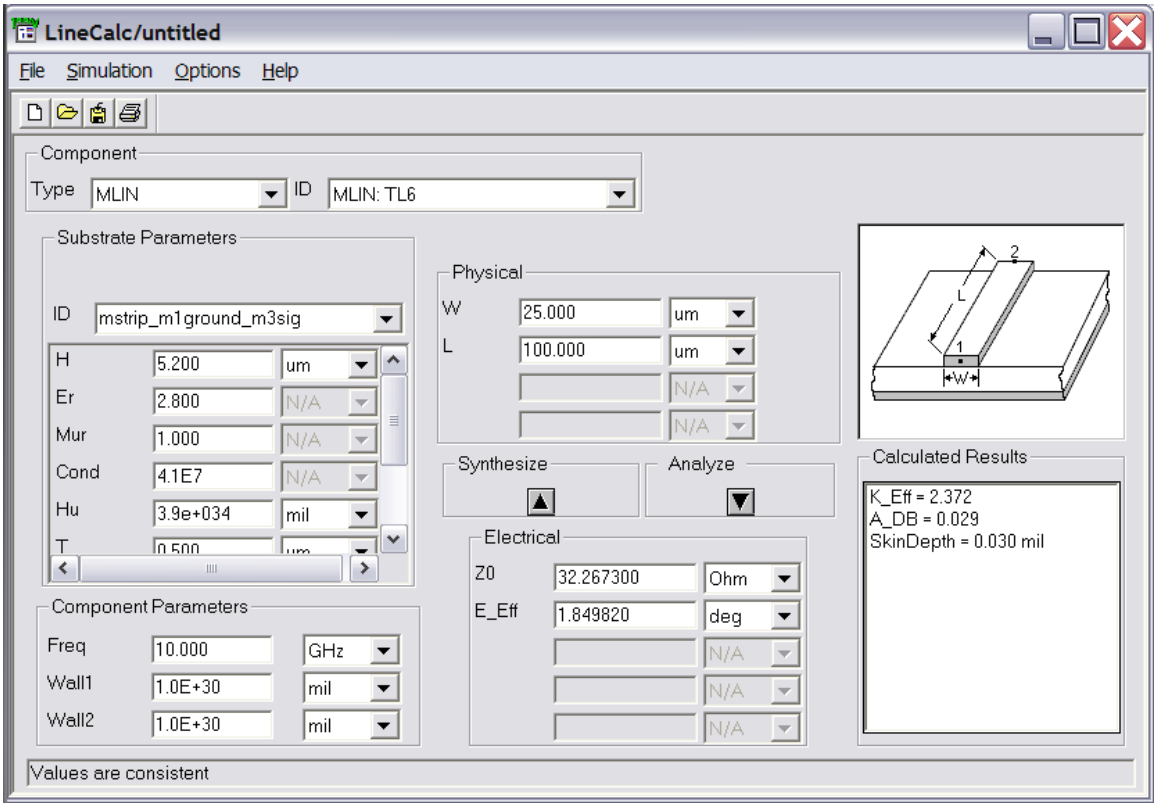


The model "inv_mstrip_m3ground_m2sig" is used for lines with an M3 ground plane and M2 signal line. The model "inv_mstrip_m3ground_m1sig" is used for lines with an M3 ground plane and M1 signal line. There are limitations for this modeling strategy:
a) I can't seem to make these models calculate metal line resistance correctly, so I have set the resistance to zero. This will be a serious error for long wires.
b) the associated substrate models are not supported by LINECALC. More on linecalc below.

The model "mstrip_m1ground_m3_sig" is a good and fairly accurate model for lines using metal 1 as the ground plane and m3 as the signal line. There are problems in using such lines; see the class notes. For lines of this type, we can select it, and then send its parameters to LINECALC:



...which allows very quick determination of line Zo, delay, and attenuation.



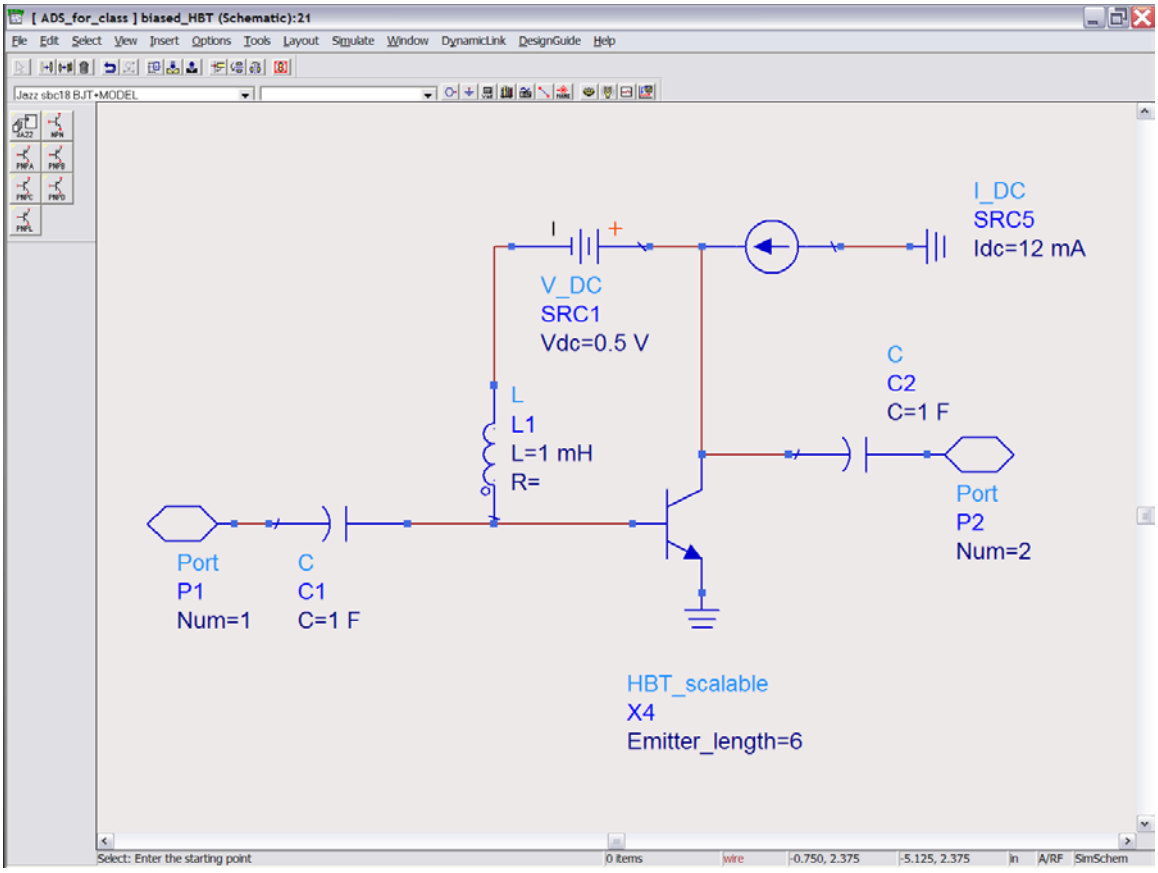
Comment: ADS MOMENTUM

In reality, we use ADS MOMENTUM to model a short section of line, and from this determine its inductance, capacitance, and resistance per unit length. From this we usually can fit lines of a specified geometry to a ADS TLINP model. MOMENTUM is a fast, accurate, and powerful tool, but there is not time to learn it in this class.

Exercises

1) first, please run all the above simulations. Make sure you can make them all work.

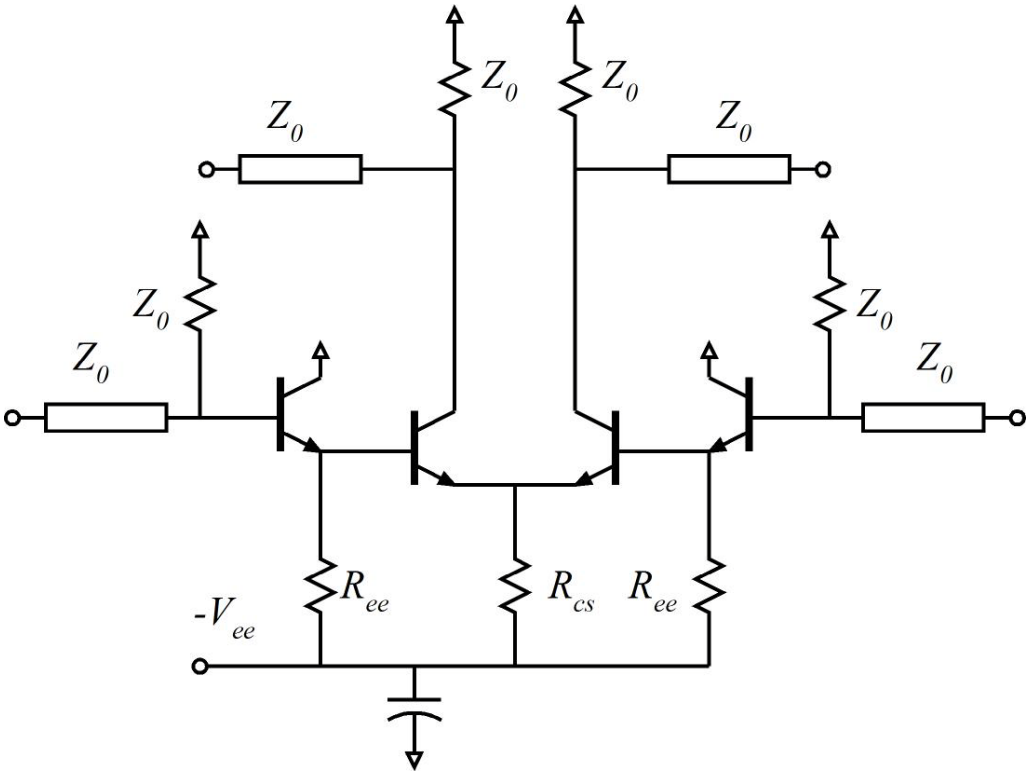
2) transistor simulation



Using the above model and the biased_HBT test bench, set up a transistor with the above parameters, but adjust the bias to a current density of 1 mA *per linear micron of emitter length* and Vcb=0.5 Volts. Simulate and plot H21 and U. Determine ft and fmax.

3) S-parameter Circuit simulation

Create a Darlington differential pair amplifier with 2 DHBTs, each having parameters as above, except adjust the emitter length so that all transistors are biased at 1.2 mA *per linear micron of emitter length* when biased at 10 mA each. The basis circuit is as below, but you must think hard about the DC biasing. The resistors labeled Zo are 50 Ohms. The external load is 50 Ohms



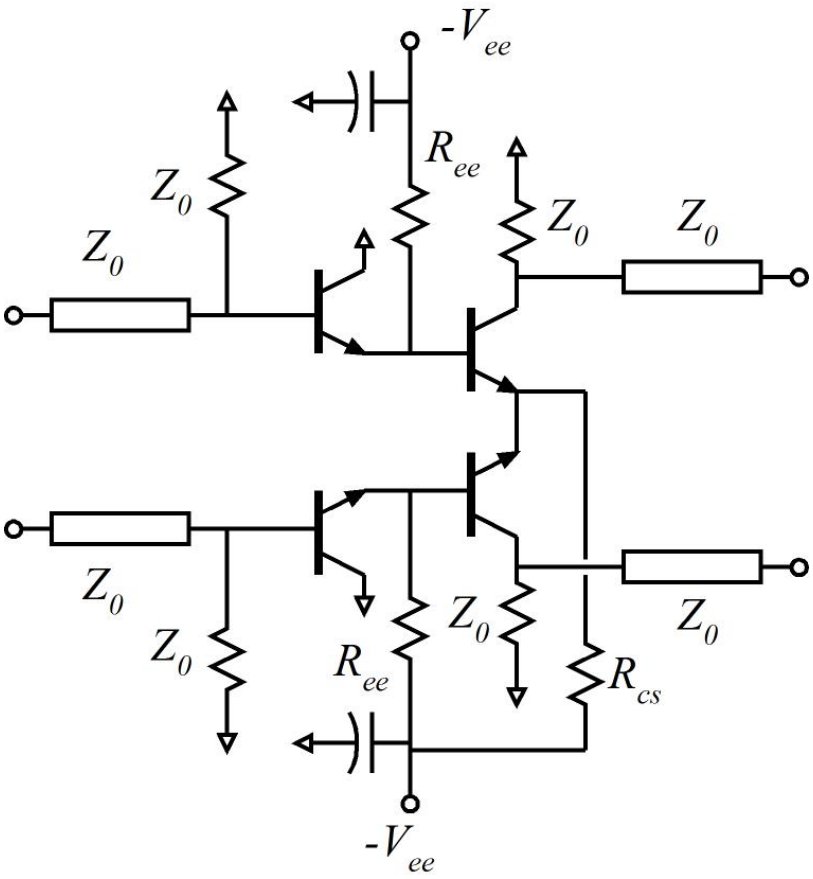
Simulate this to determine the 4 S-parameters. Determine the gain and bandwidth of S21. Recall that S-parameters are equivalent to gains with 50 Ohm generator and load impedances.

4) Transient Circuit simulation.

Simulate the transient response of this circuit to a PRBS pattern with 1 mV pp input. Adjust the data rate to that which provides reasonable performance. Recall the need for external 50 Ohm generator and loads.

5) Mask layout

Using the available CML differential amplifier as a guide, generate an ADS mask layout for the amplifier. Hint: a floorplan like that below will facilitate layout.



6) Re-simulation:
Measure the lengths of wires in your mask layout. Add models for these wires to the circuit file and resimulate.