
Corner Simulations

This document illustrates the approach to perform corner simulations for the OPAMP/System. Before starting with these steps, you are expected to have met the specification under nominal conditions.

Corners are used to verify the circuit performance under the extreme case conditions. If the performance is met in there, the confidence is high that the circuit will meet the requirements under all possible circumstances. A commercial used design is expected to be successful in all corners. Within the scope of this lab, your design must meet the specification for the nominal case and the majority of corners. It is not mandatory to reach the specifications in all corners, although a high number of successful corners (ideally all) will be beneficial for the evaluation of your work. In general, the following corners need to be considered for the extreme cases:

1. Process Corners:
 - MOS (ss, snfp, tt, fnsp, ff)
 - Capacitor (ss, typ, ff)
 - Resistor (ss, typ, ff)
2. Voltage Corners: 1.62 V, 1.8 V, 1.98 V ($\pm 10\%$ variation)
3. Temperature Corners: -25°C , 25°C , 85°C

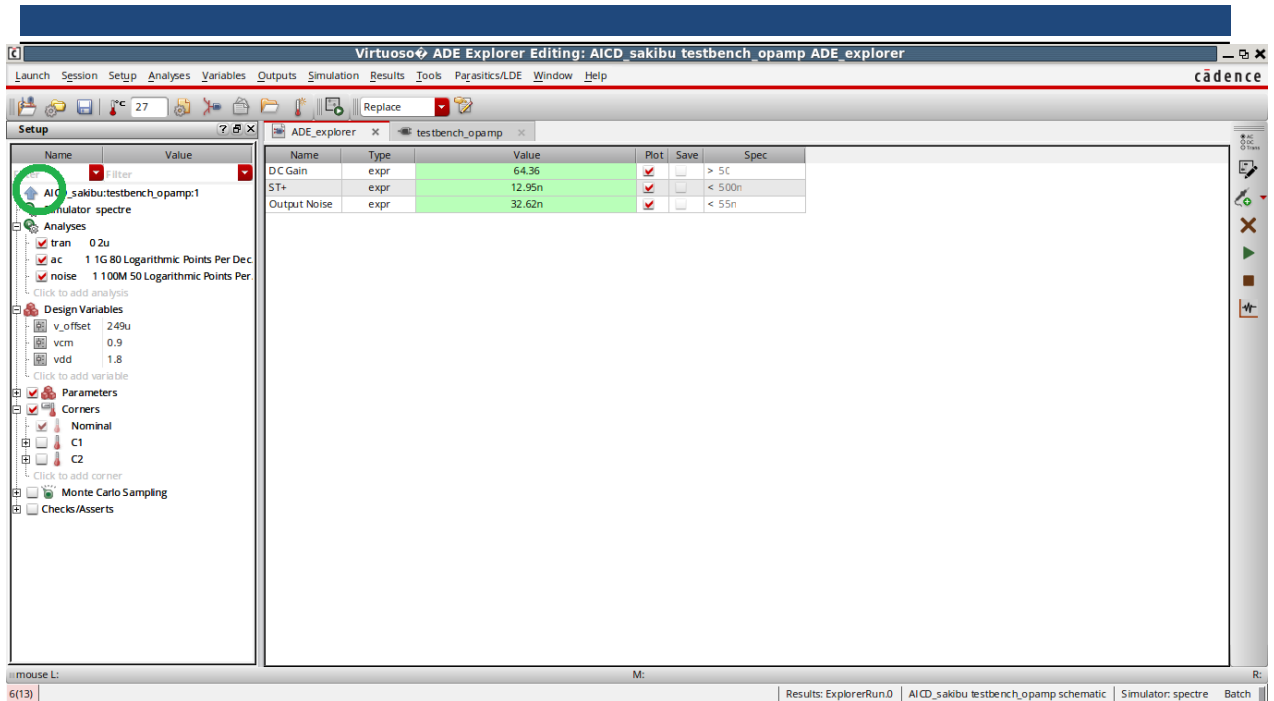
Overall this leads to $5*3*3*3*3 = 405$ possibilities. To reduce the simulation time for this lab, we will **only use the following extreme corners**: MOS (ss, snfp, fnsp, ff); Cap (min, max); Resistor (min, max); Voltage (1.62, 1.98); Temperature (-25°C , 85°C). The following Outputs are to be calculated for the Corner Simulation runs:

1. DC Gain
2. Gain Bandwidth
3. Phase Margin
4. Slew Rate +ve
5. Slew Rate -ve
6. Settling Time (high)
7. Settling Time (low)
8. Offset

Step 1

From library manager of cadence, open your ADE Explorer file that has the tests on which you want to run the corner simulation.

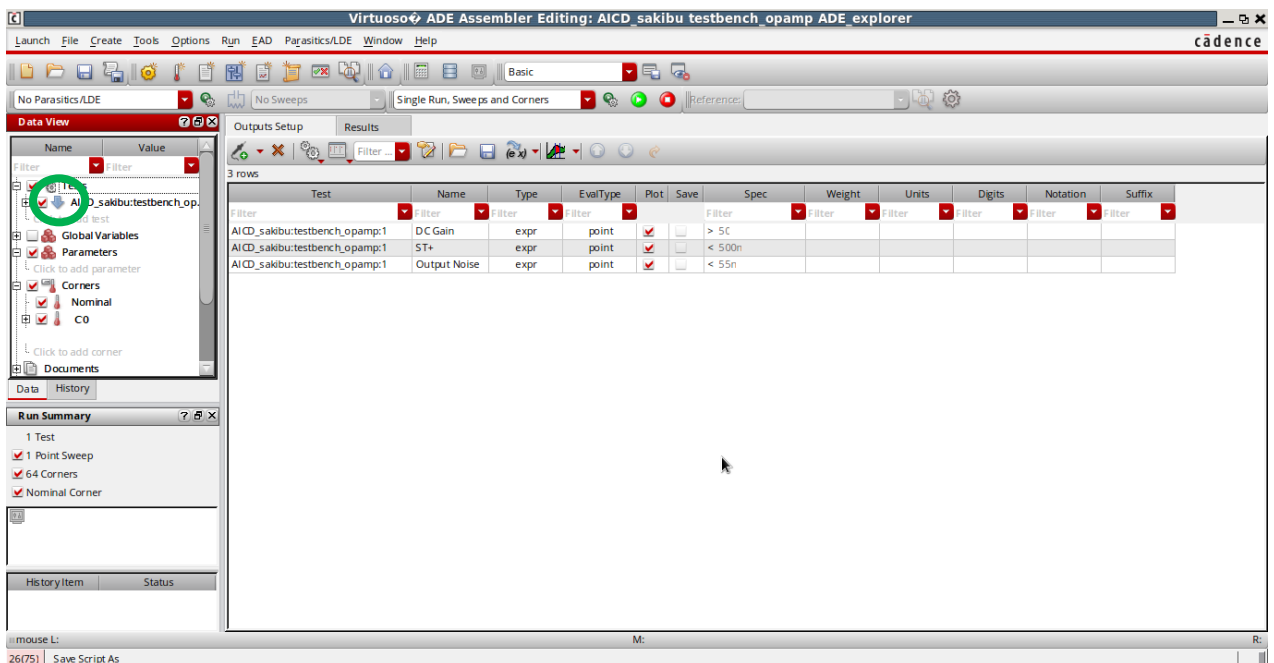
The ADE Explorer will look similar to the example below (In this example we have ac analysis: DC Gain, transient analysis: Settling time low to high and noise analysis for measuring Output noise):



Now click the blue arrow (marked in green circle).

Step 2

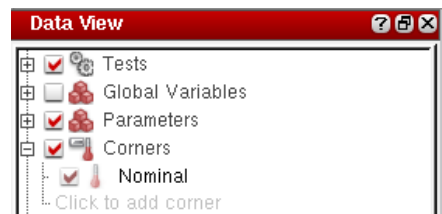
It should now bring the **ADE Assembler** Editing window similar to the one below.



Note that, you can always **go back to ADE Explorer**. If you want to go back, just from the menu, click on the blue arrow besides the test (marked in green circle). You can also access the explorer via the **Launch -> ADE Explorer** menu.

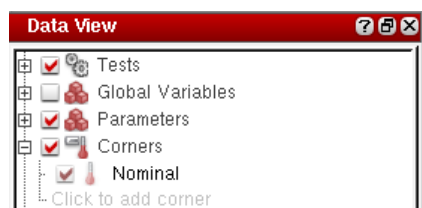
Step 3

Now in the **Data View** tab uncheck the **Global Variables** section.

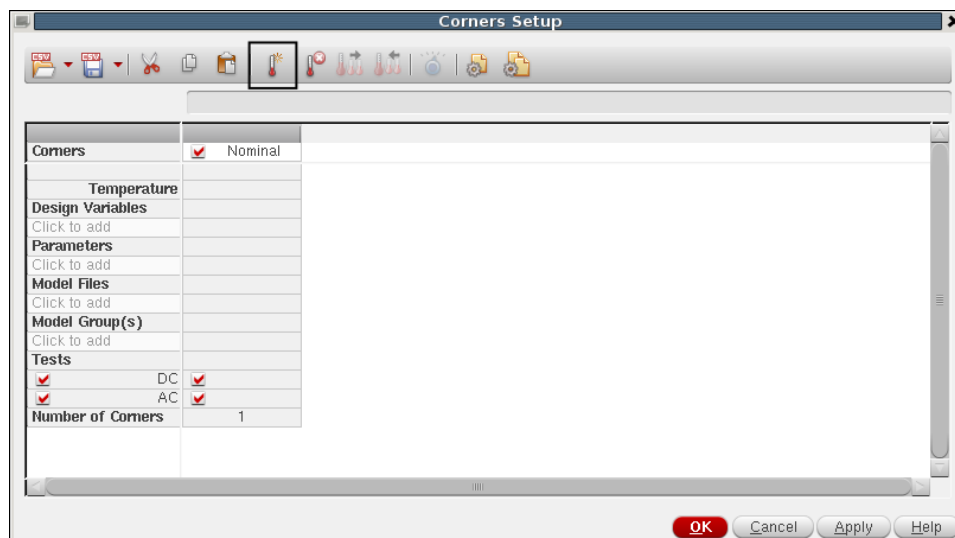


Now, click on the '+' option available before the **Corners** section to setup the required corners for the corner simulation simulations, in the following way:

+ -> **Corners** -> **Click to add corner**

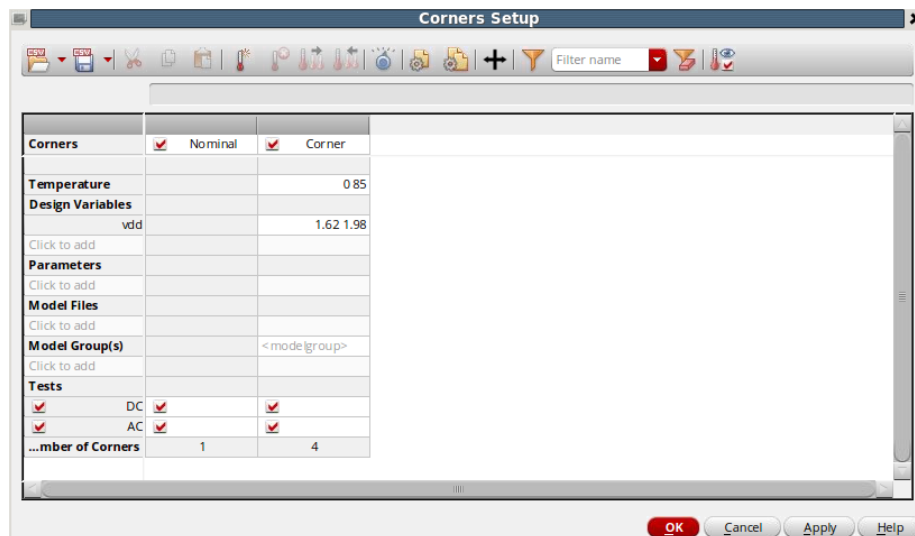


Once you click on the **Click to add corner** option the **Corners Setup** window opens up:



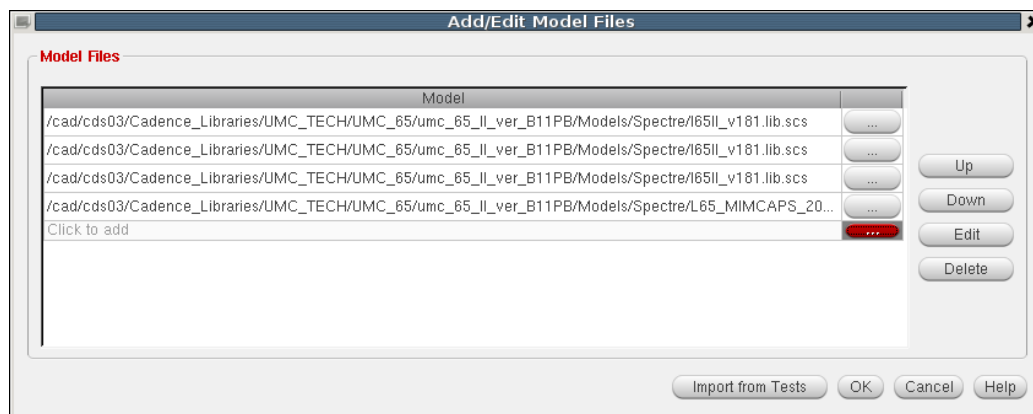
Click the highlighted option to start adding corners. Next, click the **click to add** option under **Design Variables** section and in the drop down menu select **All** to add all the variables at once.

Once all the required variables are added, setup the window as follows for the temperature and voltage corners:



Step 4

Next, to setup the process corners we will first have to add the model library files available from UMC. For this select the **click to add option** available under the **Model Files** which opens up the following window.



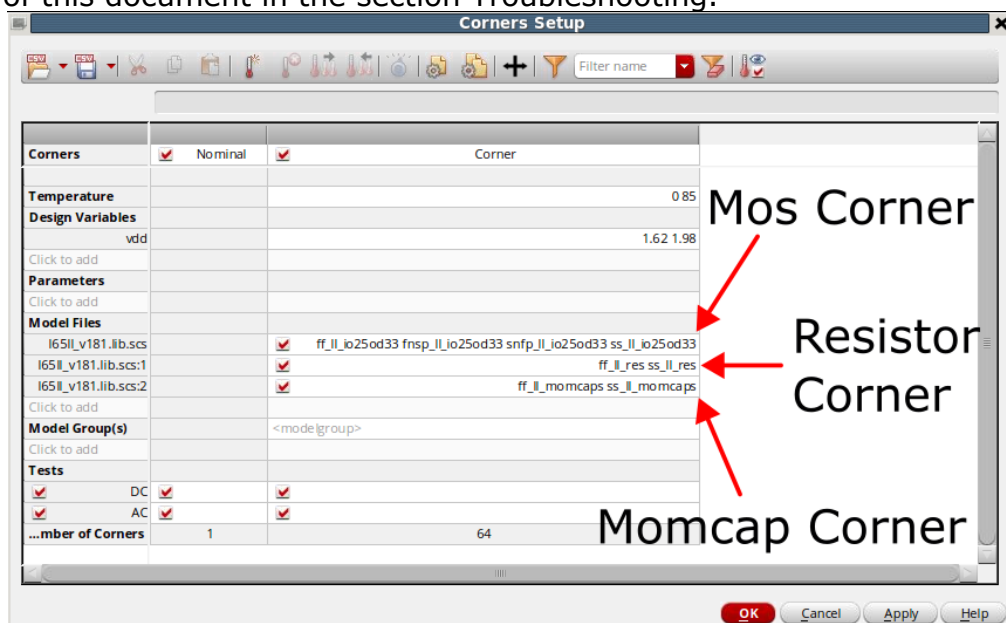
Click on the '...' option to browse and add model files. Both following paths are relevant:

1. MOS/RES/MOMCAP:
/cadstud/cds03/Cadence_Libraries/UMC_TECH/UMC_65/umc_65_II_ver_B11PB/Models/Spectre/l65II_v181.lib.scs
2. MIMCAP:
/cadstud/cds03/Cadence_Libraries/UMC_TECH/UMC_65/umc_65_II_ver_B11PB/Models/Spectre/L65_MIMCAPS_20F_KF_V101_RF.lib.scs

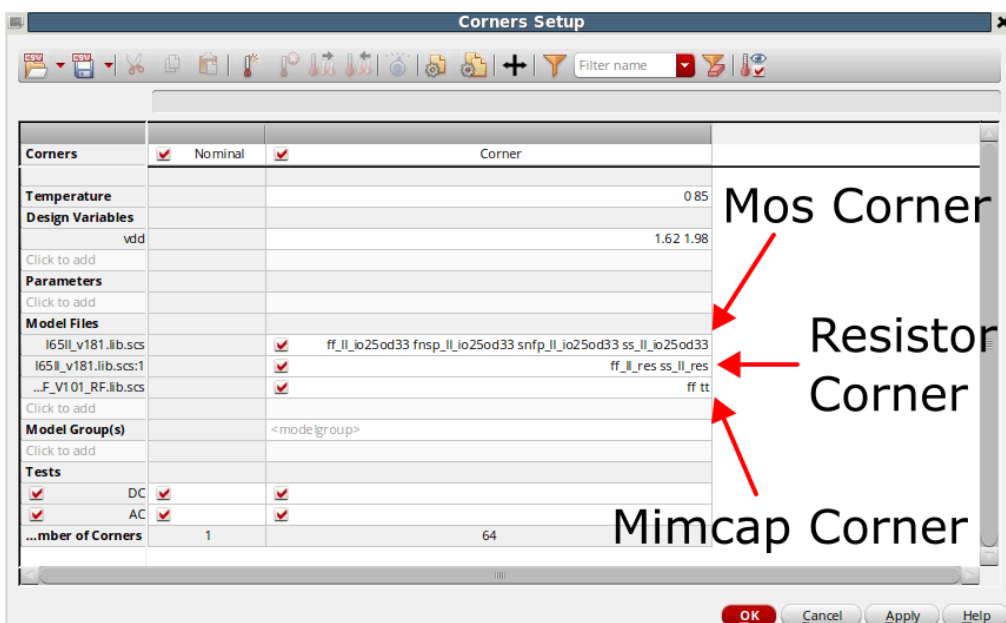
Here you have to choose the files **according to your own design**. All elements used in your design must be specified. So you will need to add the first path multiple times, for the MOS, RES and CAP each. For the cap, just use the type you have chosen (either MOMCAP or MIMCAP). If your design does not use a Resistor, it can be omitted too.

The added model files contain the internal model parameters for different devices and corners. So the device and corner which should be used by the simulator must be specified explicitly. For the MOSFETs add **"ss_II_io25od33**

ff_II_io25od33 fnsfp_II_io25od33 snfp_II_io25od33" in the row of the "l65ll_v181.lib.scs model". For the resistor add **"ff_II_res ss_II_res"** to the next "l65ll_v181.lib.scs:1" line (the index :1 at the end indicates, that this is a duplicate entry, which is ok here, DO NOT ADD THIS AT THE MODEL PATH. Depending on your design, you can use either MOMCAPS or MIMCAPS. For the MOMCAPS add next to the third line of the "l65ll_v181.lib.scs:2" add **"ff_II_momcaps ss_II_momcaps"**. For the MIMCAPS use the "L65_MIMCAPS_20F_KF_V101_RF.lib.scs" file instead and **"ff ss"**. A compact overview of the relevant files and corners definition is provided at the end of this document in the section Troubleshooting.



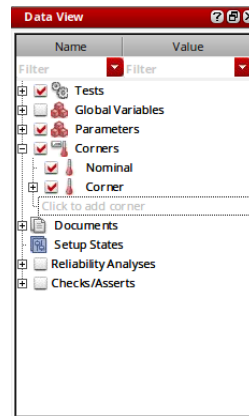
Example for MOMCAPS



Example for MIMCAPS

These are just examples, depending on your individual design, the configuration might vary. For individual setup, please refer to the Troubleshooting section at the end.

Next, click ok and the setup corner shows up as follows:



Step 5

To make the interpretation of results easier, click on the **Outputs Setup** tab and add the required specifications for the parameters being simulated.

Test	Name	Type	Details	EvalType	Plot	Save	Spec	Weight	Units	Digits	Notation	Suffix
dc_sweep_full	dc_out_follow	expr	VS("vout_dc_follow")	point								
dc_sweep_full	dc_in_follow	expr	VS("v_dc_sweep")	point								
dc_sweep_full	dc_follow_error	expr	(dc_out_follow - dc_in_follow)	point								
dc_sweep_full	lower_range [10 mV]	expr	cross(abs(dc_follow_error) (v...	point			< 0.5					
dc_sweep_full	upper_range [10 mV]	expr	cross(abs(dc_follow_error) (v...	point			> 1.3					
dc_sweep_full	power	expr	((value(S("i0/vdd") VAR("v...	point			< 600u					
dc_sweep_full	dc_in	expr	VS("v_dc_sweep")	point								
dc_sweep_full	dc_out	expr	VS("vout_dc")	point								
dc_sweep_small	sweep_out	expr	VS("vout_dc")	point								
dc_sweep_small	gain	expr	deriv(VS("vout_dc"))	point								
dc_sweep_small	max_gain	expr	ymax(deriv(VS("vout_dc")))	point								
dc_sweep_small	max_gain_db	expr	db20(max_gain)	point								
dc_sweep_small	offset	expr	(cross(sweep_out VAR("vcm...	point			range -5m 5m					
ac	Loop Gain Phase	expr	phaseDegUnwrapped(getDat...	point								
ac	Loop Gain dB20	expr	db(mag(getData("loopgain") ?...	point								
ac	Phase Margin	expr	getData("phaseMargin") ?res...	point			> 55					
ac	Gainbandwidth	expr	getData("phaseMarginFreq")...	point			> 20M					
ac	DC gain	expr	value(db(mag(getData("loop...	point			> 50					
tran	vin_sin	expr	VT("vin_tran_sin")	point								
tran	vout_sin	expr	VT("vout_tran_sin")	point								
tran	vin_rect	expr	VT("vin_tran_rect")	point								
tran	vout_rect	expr	VT("vout_tran_rect")	point								
tran	slew rate neg	expr	abs(slewRate(clip(VT("vout...	point			> 15M					
tran	settling time pos	expr	(settlingTime(clip(VT("vout...	point			< 500n					
tran	settling time neg	expr	(settlingTime(clip(VT("vout...	point			< 500n					
tran	slew rate pos	expr	slewRate(clip(VT("vout_tran...	point			> 15M					
noise	input noise: V / sqrt(Hz)	expr	getData("in" ?result "noise")	point								
noise	max input noise: V / sqrt(Hz)	expr	value(getData("in" ?result "n...	point			< 20n					
dc_op	o	expr	VDC("vout_dc")	point								
ac_cmrr	vout_cm	expr	db20(VF("vout_ac_cmrr"))	point								
ac_cmrr	vout_cm_dc	expr	value(db20(VF("vout_ac_c...	point								
ac_cmrr	vout_dm_dc	expr	value(db(mag(getData("loop...	point								
ac_cmrr	cmrr	expr	(vout_dm_dc - vout_cm_dc)	point			> 60					

The addition of these specifications under the **Spec** tab would allow you to easily distinguish between passed (green), failed (red) and near (yellow) corners.

With this the setup for corner simulations is done and the analysis can now be run using,

Run -> Single Run, Sweeps and Corners ...

Sample corner simulation results:

The color highlighting provides a quick overview on the critical corners and design targets.

Parameter				C0_0		C0_1	C0_2	C0_3	C0_4	C0_5	C0_6	C0_7	C0_8	C0_9	C0_10	C0_11	C0_12	C0_13	C0_14	C0_15	C0_16	C0_17	C0_18
I65II_v181.lib.scs				ff_II_m...		ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...	ff_II_m...
I65II_v181.lib.scs				ff_II_res		ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res	ff_II_res
I65II_v181.lib.scs				ff_II_io...		ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...	ff_II_io...
temperature				-25		85	-25	85	-25	85	-25	85	-25	85	-25	85	-25	85	-25	85	-25	85	-25
vdd				1.62		1.62	1.98	1.98	1.62	1.62	1.98	1.98	1.62	1.62	1.98	1.98	1.62	1.62	1.98	1.98	1.62	1.62	1.98

Test	Output	Spec	Weight	Pass/Fail	Min	Max	C0_0	C0_1	C0_2	C0_3	C0_4	C0_5	C0_6	C0_7	C0_8	C0_9	C0_10	C0_11	C0_12	C0_13	C0_14	C0_15	C0_16	C0_17	C0_18
dc_sweep_full	lower_range [10 mV]	< 0.5		near	221.3m	549.7m	341.1m	249.5m	317m	221.3m	405m	320.4m	378.2m	289.6m	486.6m	398.7m	459.5m	369.7m	549.7m	467.3m	516.5m	432.3m	341.1m	249.5m	317
dc_sweep_full	upper_range [10 mV]	> 1.3		pass	1.36	1.893	1.478	1.437	1.859	1.619	1.405	1.36	1.799	1.755	1.518	1.478	1.893	1.855	1.466	1.418	1.855	1.809	1.478	1.437	1.8
dc_sweep_full	power	< 600u		pass	248.3u	366.7u	284.8u	290.7u	362.9u	366.7u	278.3u	284.1u	356.4u	361.3u	265.6u	277.9u	355.3u	360.1u	248.3u	267.2u	347.2u	354.3u	284.8u	290.7u	362
dc_sweep_full	/AVdd																								
dc_sweep_small	offset	range -5m 5m		pass	538.7u	1.461m	749.8u	800.3u	538.7u	573.3u	974.3u	1.049m	683.2u	743.1u	1.045m	1.032m	608.4u	646u	1.461m	1.4m	772.4u	826.5u	749.8u	800.3u	538
ac	Phase Margin	> 55		near	50.42	61.42	59.02	61.42	58.61	61.16	57.93	60.14	57.1	59.6	59.97	61.25	57.98	60.13	60.01	60.45	56.55	58.58	59.02	61.42	58.6
ac	Gainbandwidth	> 20M		pass	20.28M	42.39M	28.99M	21.53M	29.0M	22.01M	20.28M	21.17M	29.48M	21.86M	27.33M	21.09M	29.69M	22.27M	25.51M	20.28M	29.14M	21.96M	28.99M	21.53M	29.0
ac	DC gain	> 50		pass	58.64	62.64	61.01	59.3	62.33	60.96	60.59	58.64	62.05	60.46	61.51	59.87	62.84	61.5	61.02	59.1	62.49	60.84	61.01	59.3	62
tran	vout_rect																								
tran	slew rate neg	> 15M		fail	7.891M	22.55M	13.81M	14.39M	15.23M	15.17M	12.32M	13.92M	15.04M	15.07M	8.826M	11.9M	14.41M	14.68M	7.891M	9.777M	13.82M	14.42M	13.81M	14.39M	15.2
tran	settling time pos	< 500n		pass	36.9n	138.4n	63.09n	59.18n	53.95n	55.22n	81.44n	62.38n	55.03n	55.8n	129.3n	82.2n	58.25n	57.43n	138.4n	111.5n	62.72n	59.06n	63.09n	59.18n	53.9
tran	settling time neg	< 500n		pass	33.07n	53.52n	50.53n	51.67n	49.67n	51.17n	51.2n	52.57n	50.09n	51.54n	50.03n	52.71n	50.3n	51.56n	48.14n	53.52n	50.84n	51.98n	50.53n	51.67n	49.6
tran	slew rate pos	> 15M		pass	15.64M	24.29M	16.04M	15.94M	16.25M	16.15M	15.94M	15.63M	16.16M	16.07M	15.81M	15.76M	16.06M	16M	15.87M	15.64M	15.97M	15.92M	16.04M	15.94M	16.2
noise	max input noise: V / sqrt(Hz)	< 20n		pass	6.936n	15.97n	9.998n	15.08n	9.721n	14.73n	10.32n	15.51n	9.925n	15.04n	10.42n	15.24n	9.729n	14.53n	11.1n	15.97n	9.933n	14.93n	9.998n	15.08n	9.72
dc_op	none																								
ac_cmrr	cmrr	> 60		fail	43.47	70.59	59.55	61.15	70.59	69.89	55.29	57.87	67.84	68.08	46.43	51.33	62.64	63.77	43.47	47.07	59.09	60.38	59.55	61.15	70.5

Troubleshooting:

Since you are manually specifying the Model Files for the simulator, the default settings are ignored. So you have to make sure, that **all** elements used by your design are specified within your Model File list. If the setup with the model libraries is incomplete, you might encounter a similar error message:

ERROR (SFE-23): "input.scs" 54: The instance `R0' is referencing an undefined model or subcircuit, `RNHR_LL'. Either include the file containing the definition of `RNHR_LL', or define `RNHR_LL' before running the simulation. Error found by spectre during circuit read-in.

In this example an element **R0** of the type **RNHR_LL** is specified in the schematic, but no definition for the resistor type **RNHR_LL** is found by the simulator. To solve this issue, you have to specify at least one case for the resistor (**tt_II_res**, **ff_II_res** or **ss_II_res**). For other possible devices, please refer to the following table.

Element	Model File	Options
RNHR_LL	[...]/I65II_v181.lib.scs	tt_II_res ff_II_res ss_II_res
mimcaps_20f_mm	[...]/L65_MIMCAPS_20F_KF_V101_RF.lib.scs	tt ff ss
momcaps_sy_mmkf or momcaps_as_mmkf	[...]/I65II_v181.lib.scs	tt_II_momcaps ff_II_momcaps ss_II_momcaps
n_25_II or p_25_II	[...]/I65II_v181.lib.scs	tt_II_io25od33 ss_II_io25od33 ff_II_io25od33 fnsp_II_io25od33

		snfp_ll_io25od33
The following elements should not be used by your design, but will be present in the given SAR logic		
n_12_llhvt or p_12_llhvt	[...]/l65ll_v181.lib.scs	tt_ll_hvt12
n_12_llrvt or p_12_llrvt	[...]/l65ll_v181.lib.scs	tt_ll_rvt12
n_12_lllvt or p_12_lllvt	[...]/l65ll_v181.lib.scs	tt_ll_lvt12
n_12_llnvt or p_12_llnvt	[...]/l65ll_v181.lib.scs	tt_ll_nvt12
If you experience other missing elements, you will likely use the wrong devices in your schematic		

Replace [...] with
/cadstud/cds03/Cadence_Libraries/UMC_TECH/UMC_65/umc_65_ll_ver_B11PB/Models/Spectre