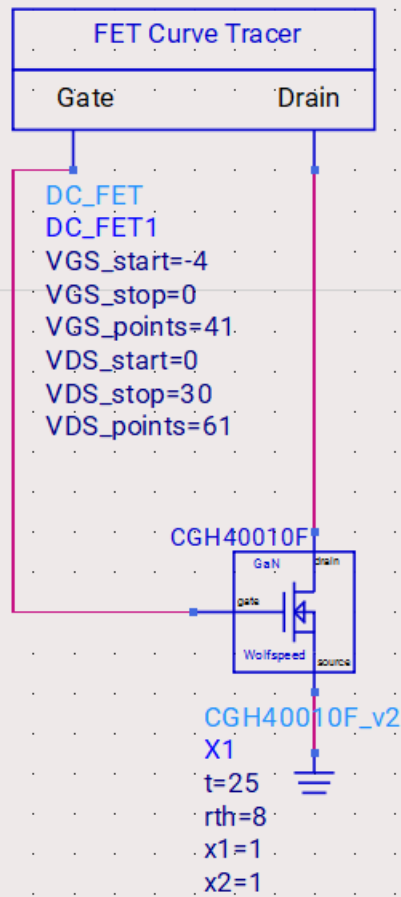
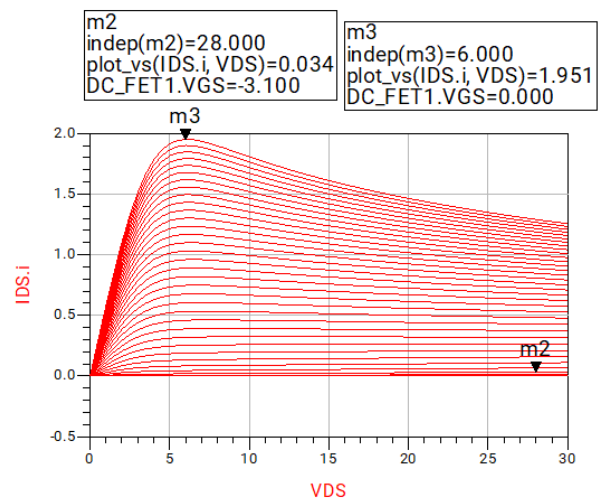
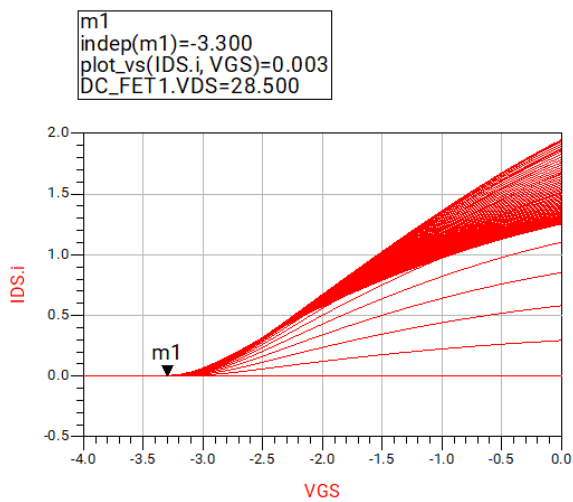


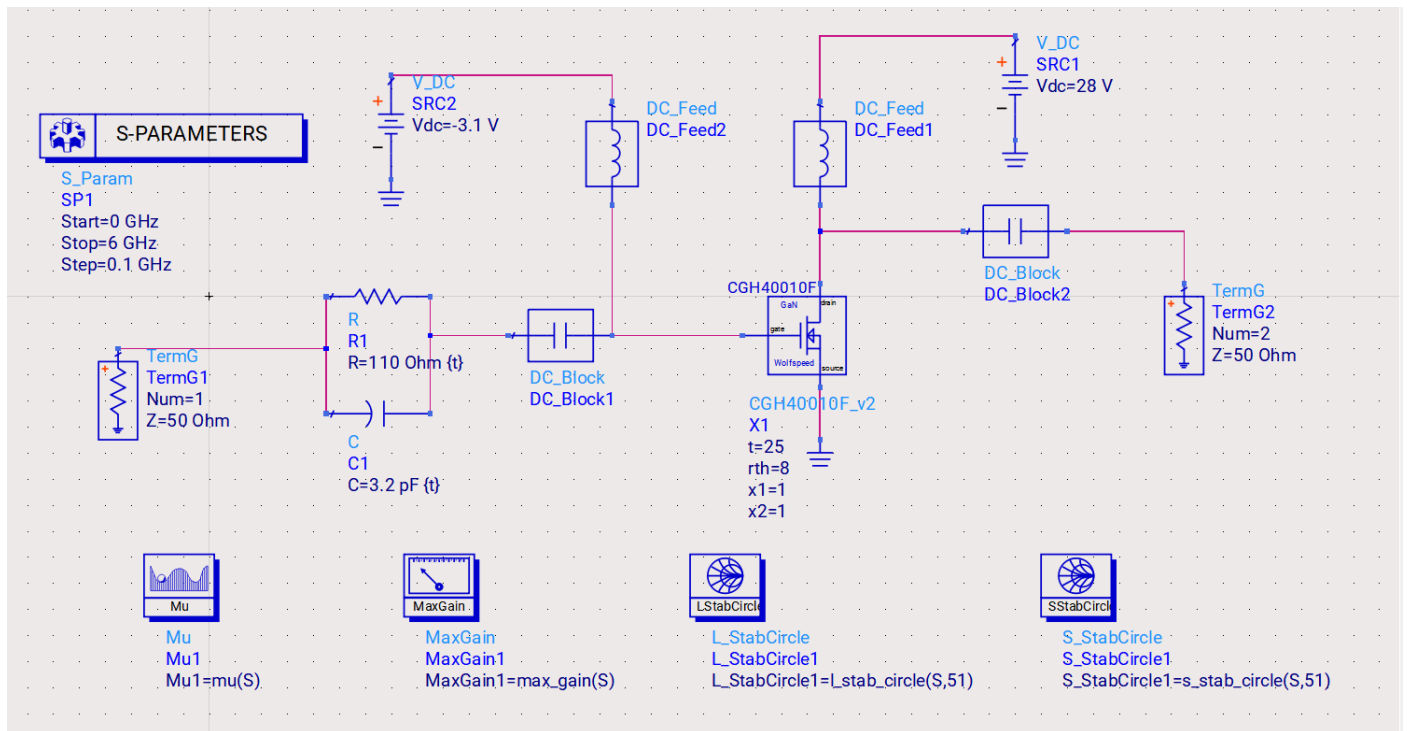
DC_I-V characteristics Schematic



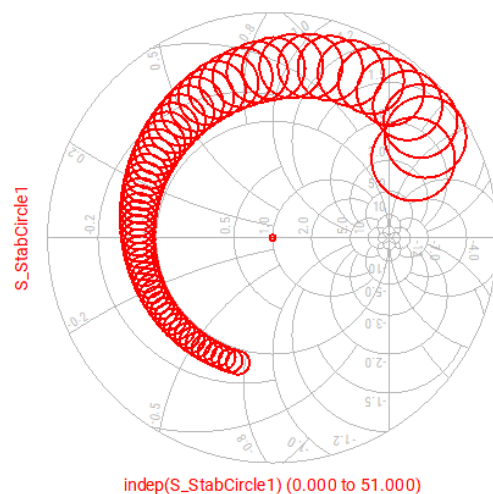
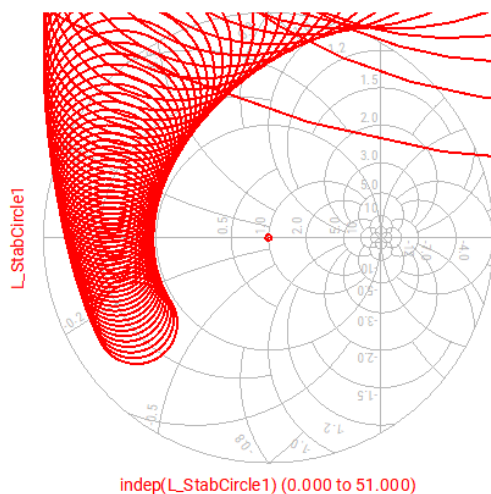
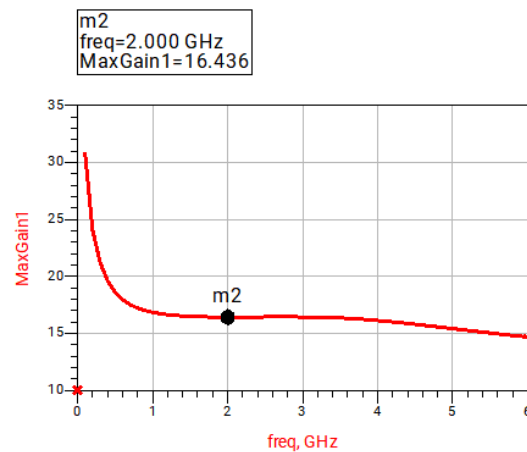
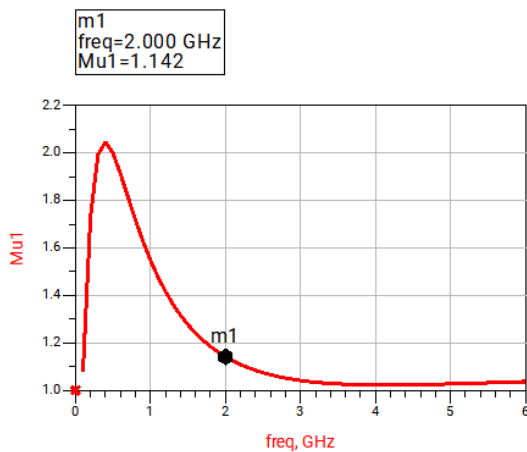
DC_I-V characteristics Results



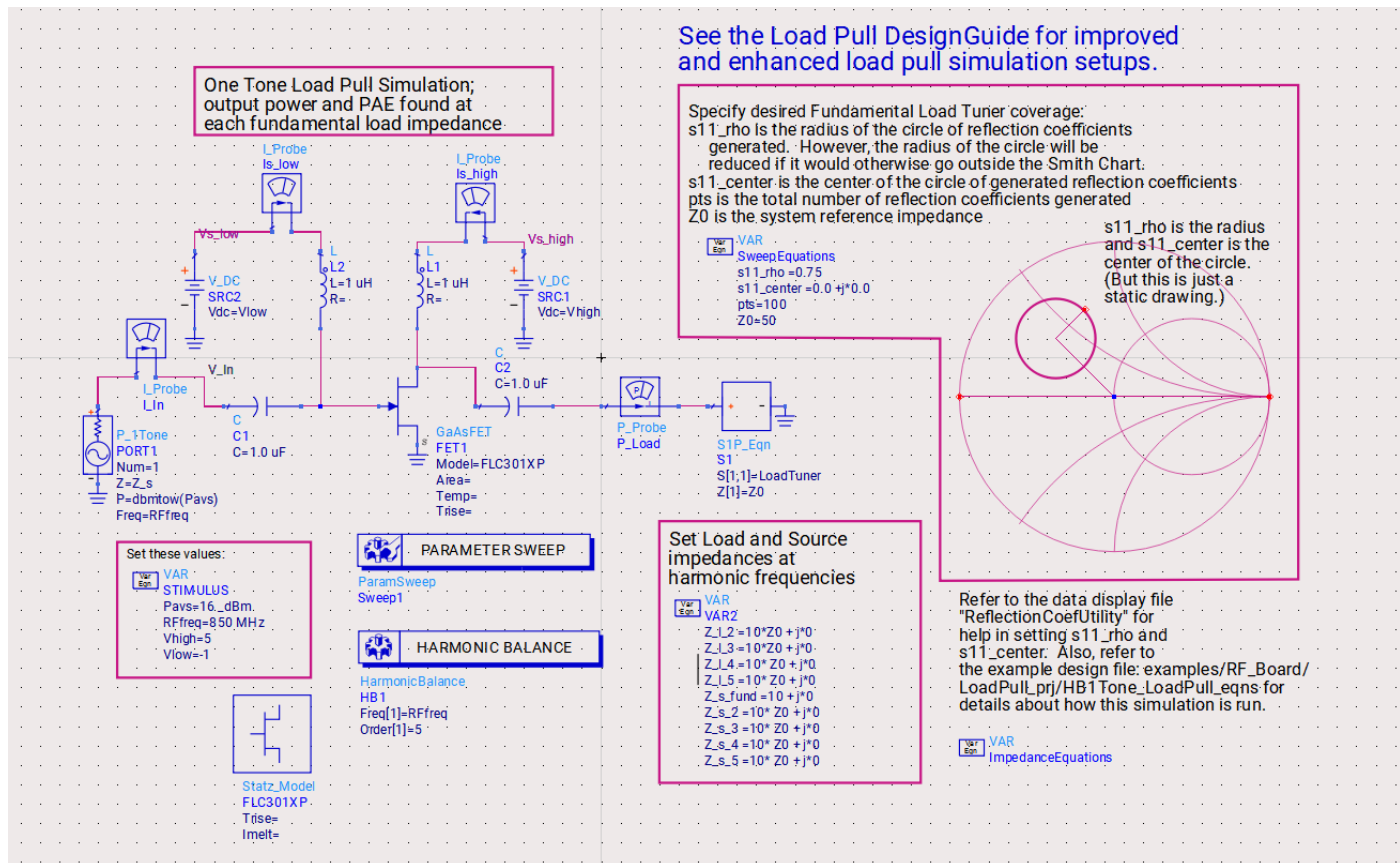
Stability Schematic:



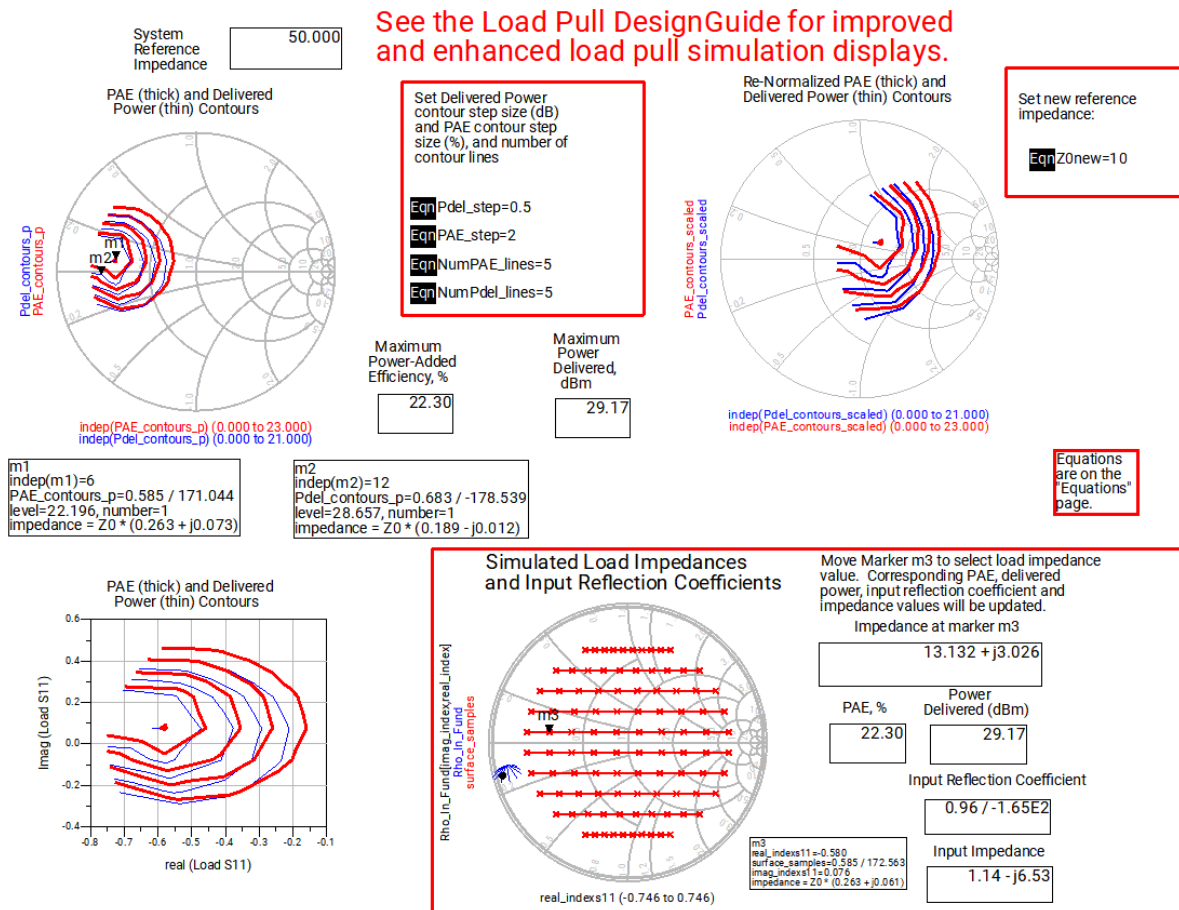
Stability Results:



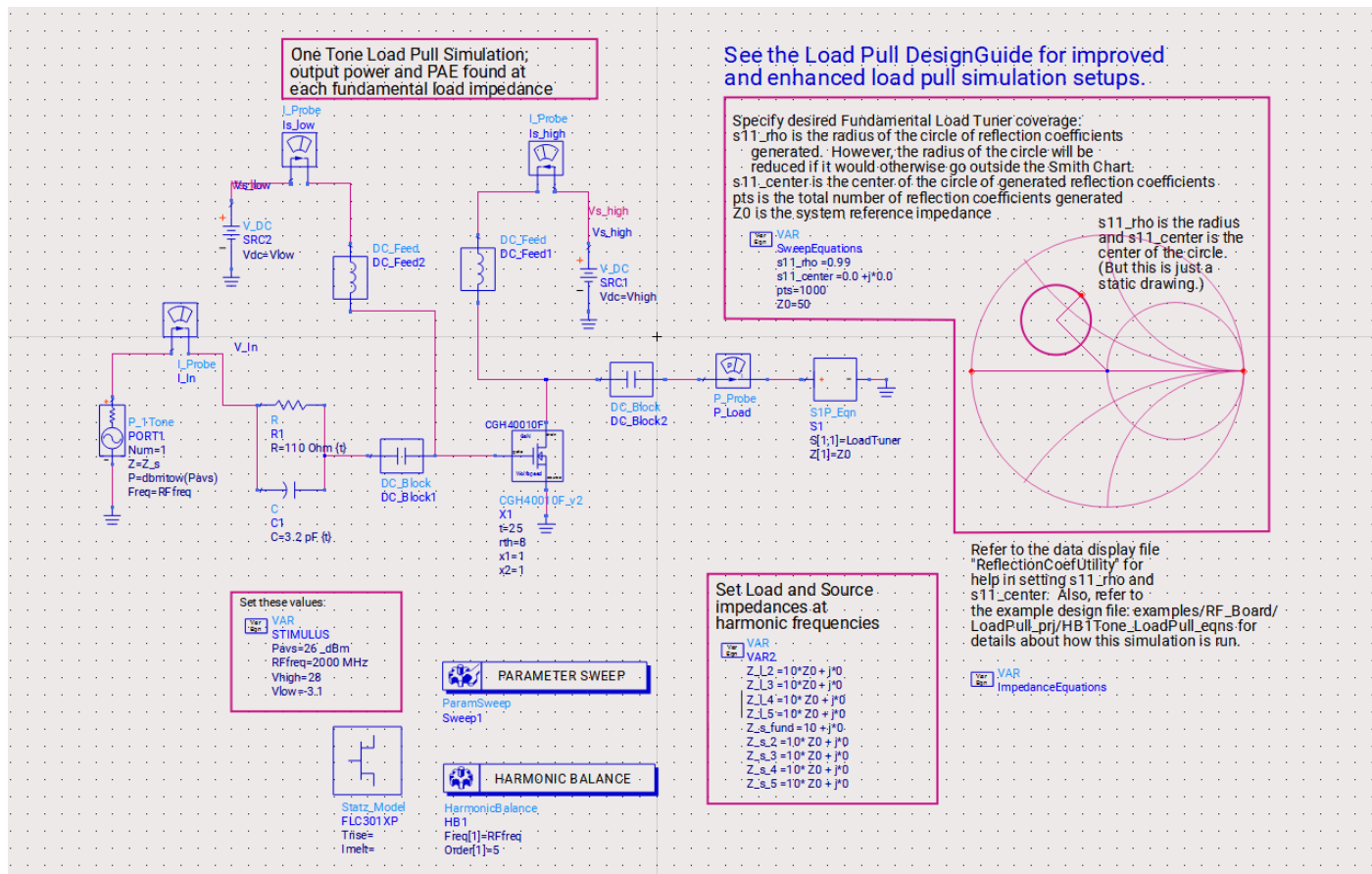
Load Pull Design Guide Schematic Design:



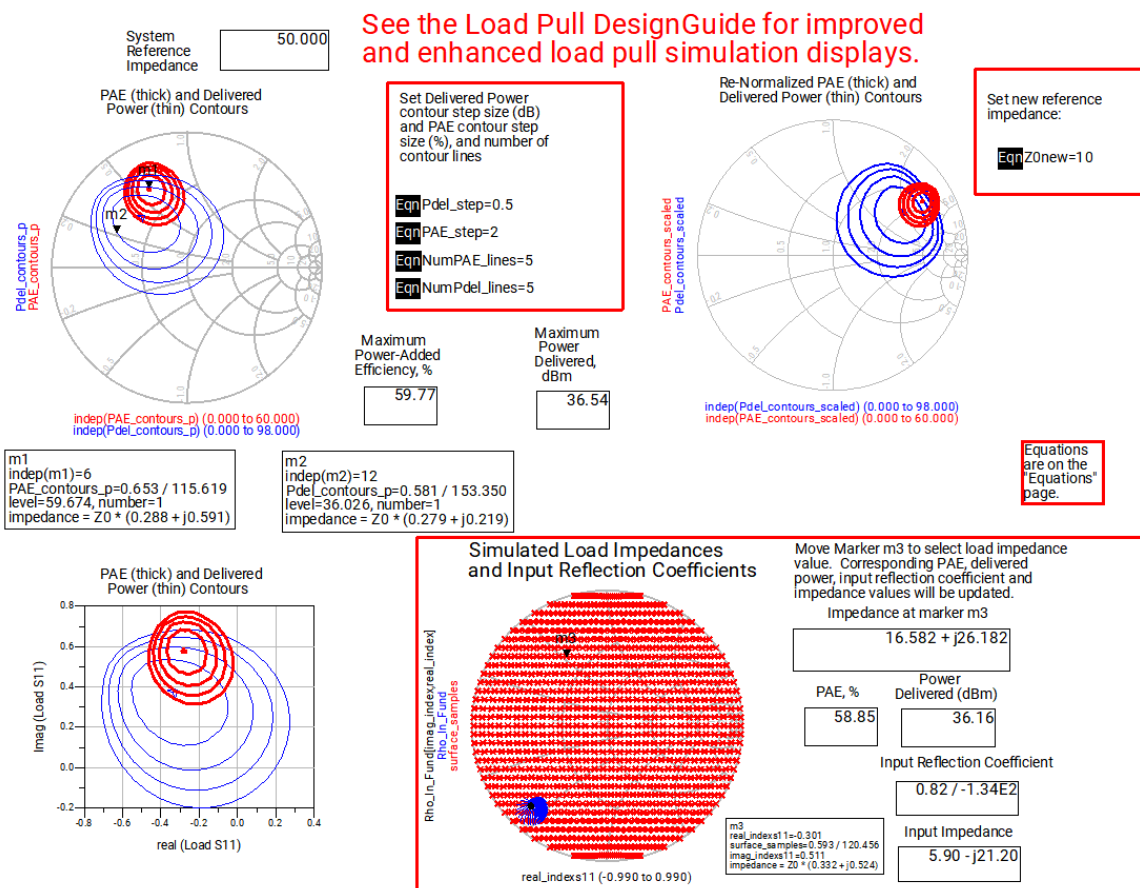
Load Pull Design Guide Schematic Design Results:



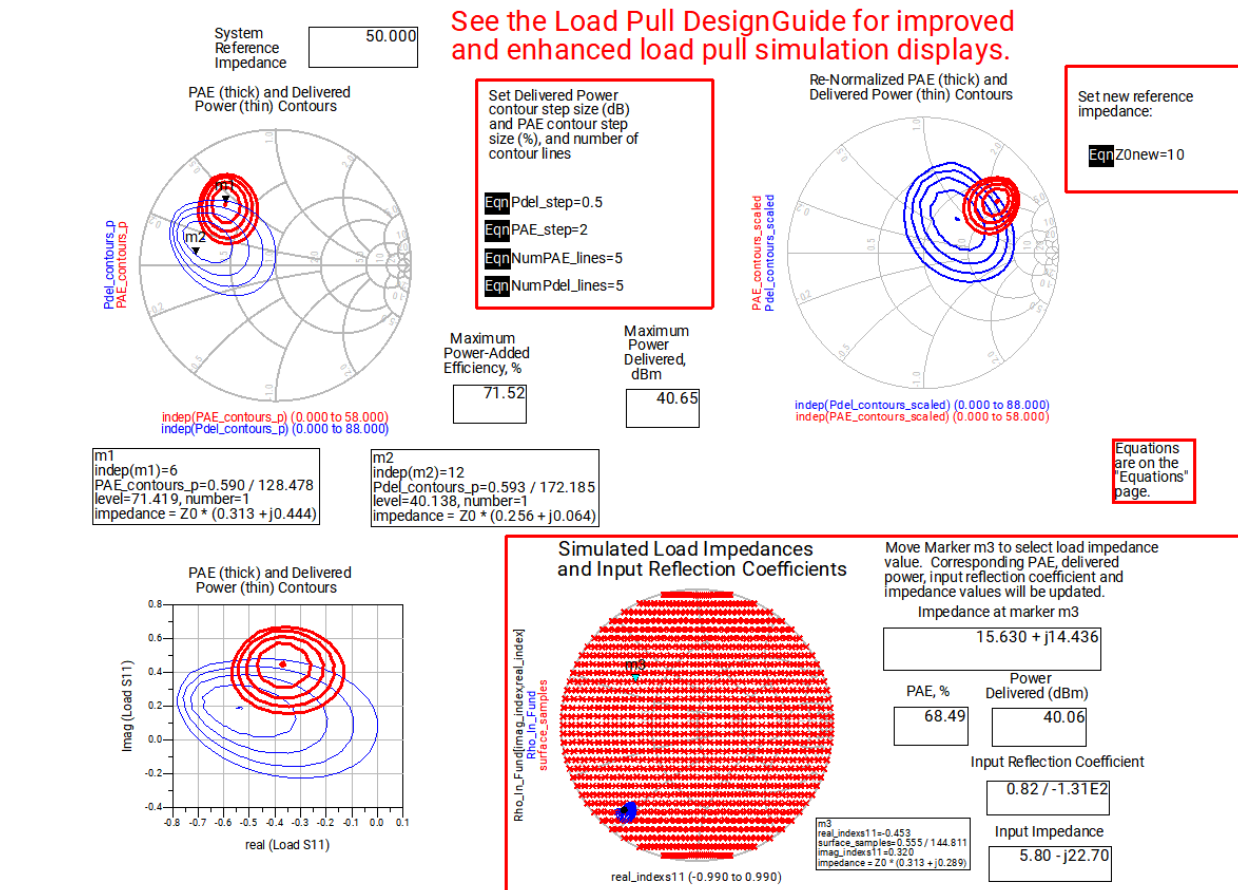
Load Pull Design Guide Replace with Stability Components Schematic:



Load Pull Design Guide Replace with Stability Components Results:



Load Pull Design Guide Replace with Stability Components and Matched With PAE % and Power Delivered (dBm):



DC Power Calculations

The exists() function checks to be sure the corresponding piece of data is in the dataset. If it is not, then the function returns 0.

Eqn Vs_l=exists("real(Vs_low[0])")

Eqn Vs_h=exists("real(Vs_high[0])")

Eqn Is_h=exists("real(Is_high.i[0])")

Eqn Is_l=exists("real(Is_low.i[0])")

Eqn Pdc=Is_h*Vs_h+Is_l*Vs_l+1e-20

Power Delivered and Power-Added Efficiency Calculations

Eqn Pdel_Watts=P_Load.p[1]

Pavs is the available source power, set on the schematic, and passed into the dataset using the Harmonic Balance controller.

Eqn Pavs_Watts=10**((Pavs[0,0]-30)/10)

Eqn PAE=100*(Pdel_Watts-Pavs_Watts)/Pdc

Eqn P_In_Watts=0.5*real(V_In[1]*conj(I_In.i[1]))

This power-added efficiency equation that uses the power absorbed at the input is considered to be a better, more accurate figure of merit than the one that uses the power available from the source.

Eqn PAE=100*(Pdel_Watts-P_In_Watts)/Pdc

Eqn Pdel_dBm = Pdel_Watts>0? 10*log10(Pdel_Watts)+30 : -200

Power-Added Efficiency contour calculations

Eqn PAEmax=max(max(PAE))

Eqn PAE_contours=contour(PAE,PAEmax-0.1-[0::(NumPAE_lines-1)]*PAE_step)

Conversion of the contours to polar coordinates for plotting on Smith Chart

This -0.1 term is included so the contour at the maximum is a small circle, rather than a single point.

Eqn PAE_contours_p=[indep(PAE_contours)+j*PAE_contours]

Conversion of the contours (which are actually curves of complex reflection coefficients that produce a constant PAE) to impedances

Eqn ZL_PAE=zin(PAE_contours_p,Z0[0,0])

Z0 is the reference impedance, set on the schematic, and passed into the dataset using the Harmonic Balance controller.

Re-scaling the contours for plotting on a Smith Chart with a different reference impedance

Eqn PAE_contours_scaled=(ZL_PAE-conj(Z0new))/(ZL_PAE+Z0new)

surface_samples are the actually-simulated reflection coefficients from which the contours are generated. real_indexs11 and imag_indexs11 are the swept variables in the simulation.

Eqn surface_samples=real_indexs11+j*expand(imag_indexs11)

Eqn Z_In_Fund=V_In[1]/I_In.i[1]

Eqn Rho_In_Fund=(Z_In_Fund-conj(Z0))/(Z_In_Fund+Z0)

Power Delivered contour calculations

Eqn Pdelmax = max(max(Pdel_dBm))

Eqn Pdel_contours=contour(Pdel_dBm,Pdelmax-0.01-[0::(NumPdel_lines-1)]*Pdel_step)

Conversion of the contours to polar coordinates for plotting on Smith Chart

Eqn Pdel_contours_p=[indep(Pdel_contours) +j*Pdel_contours]

Conversion of the contours (which are actually curves of complex reflection coefficients that produce a constant delivered power) to impedances

Eqn ZL_Pdel=zin(Pdel_contours_p,Z0[0,0])

Re-scaling the contours for plotting on a Smith Chart with a different reference impedance

Eqn Pdel_contours_scaled=(ZL_Pdel-conj(Z0new))/(ZL_Pdel+Z0new)

imag_index and real_index are used to calculate the PAE and power delivered that correspond to the reflection coefficient at the m3 marker location.

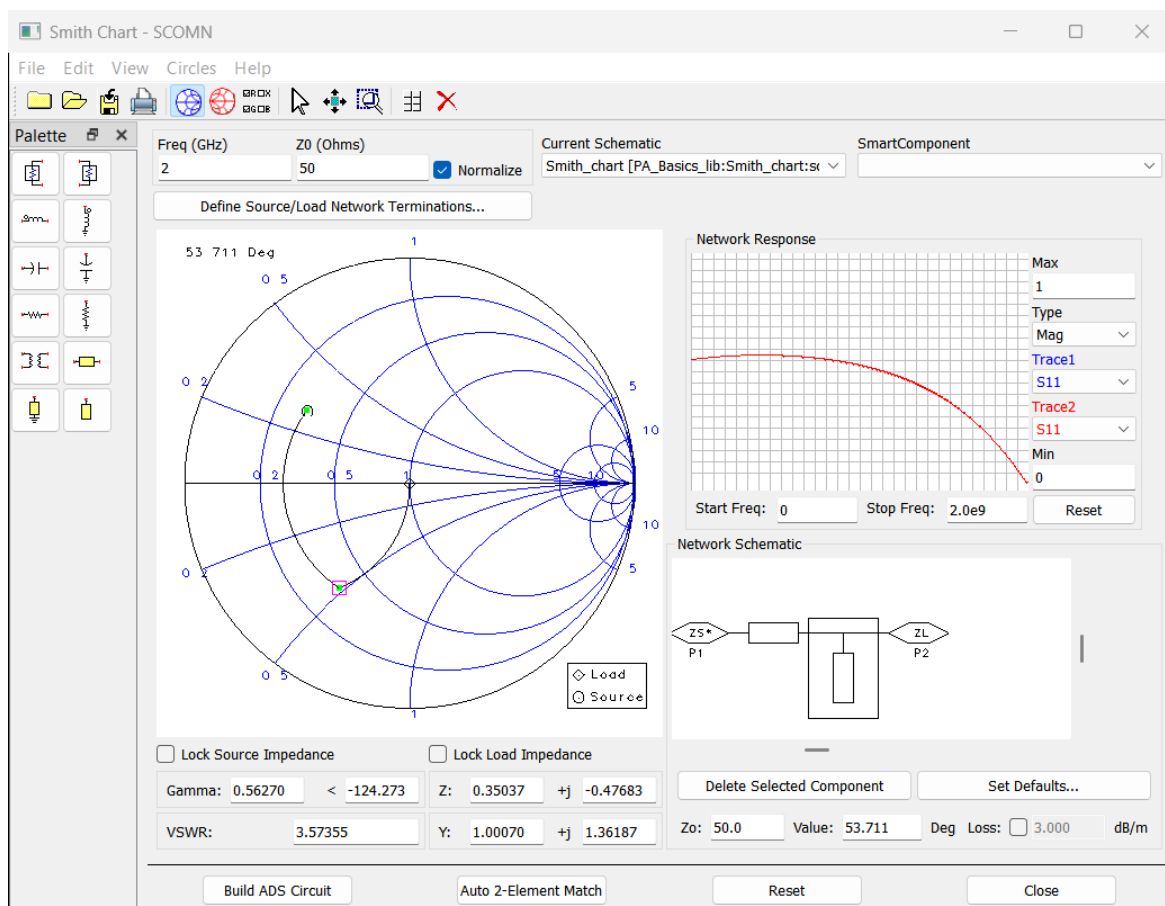
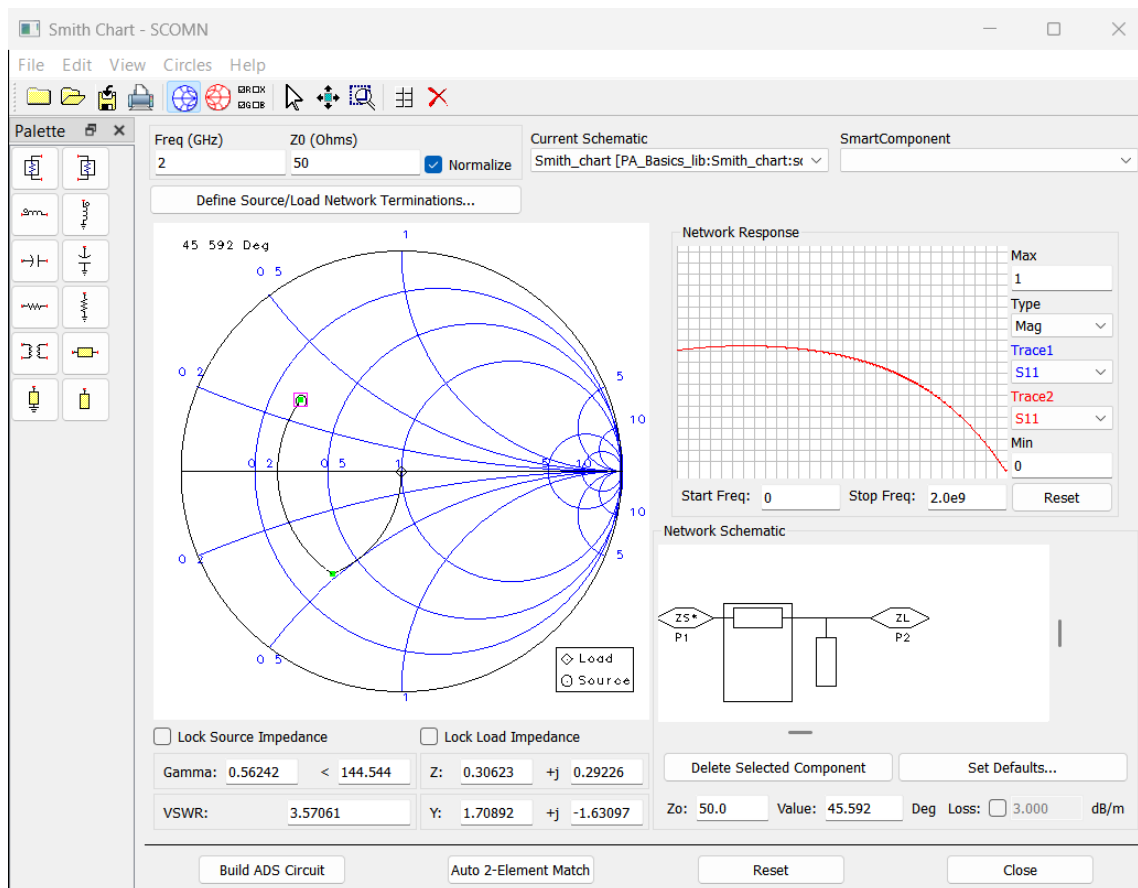
Eqn imag_index=find_index(imag_indexs11,imag(m3))

Eqn real_index=find_index(real_indexs11[imag_index,:],real(m3))

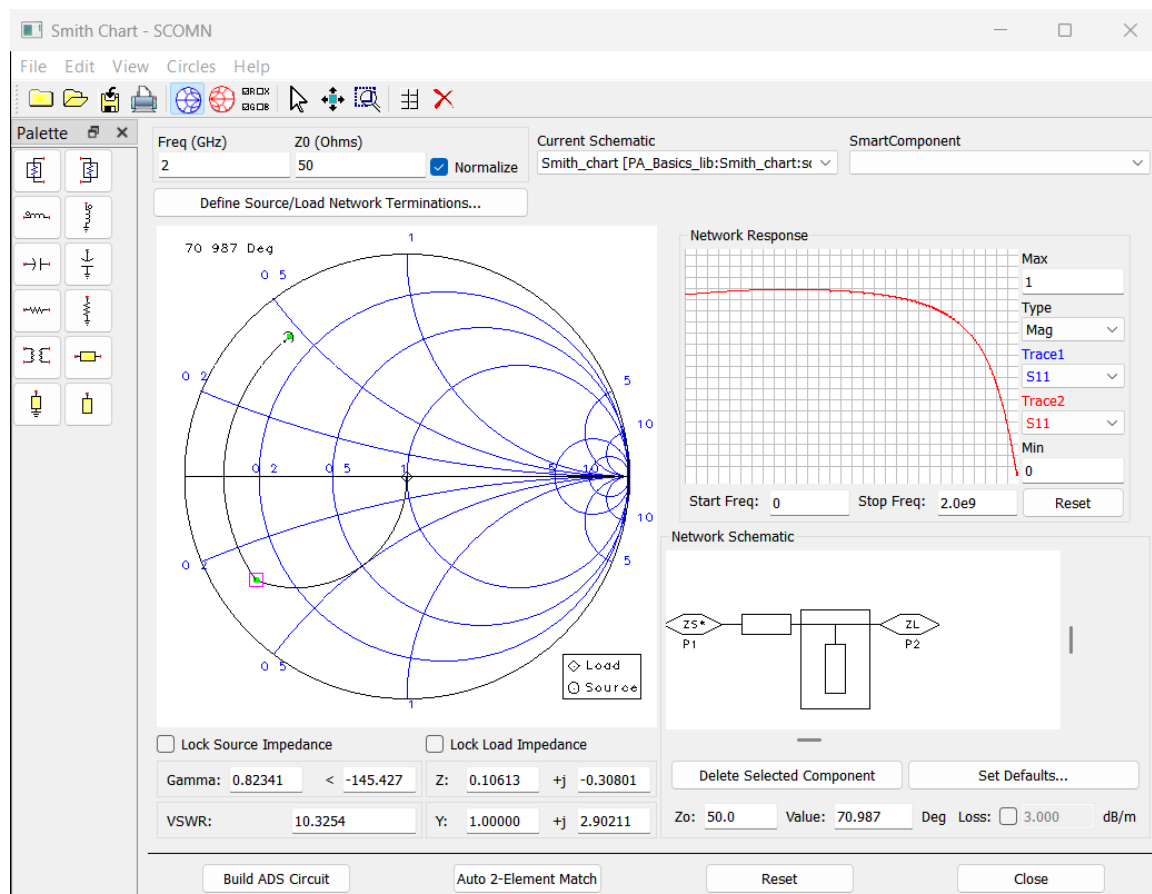
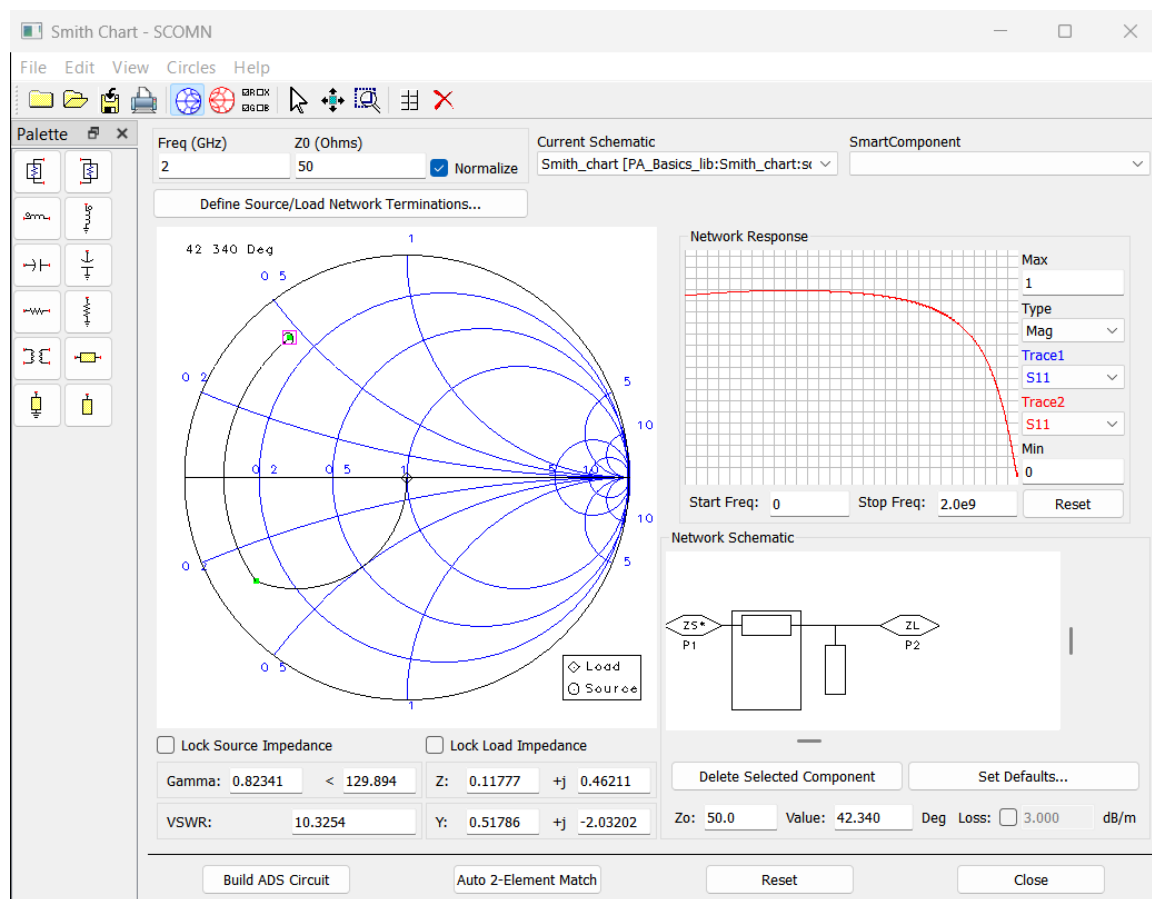
This calculates the impedance at marker m3.

Eqn Z_at_m3=(conj(Z0[0,0])+Z0[0,0]*m3)/(1-m3)

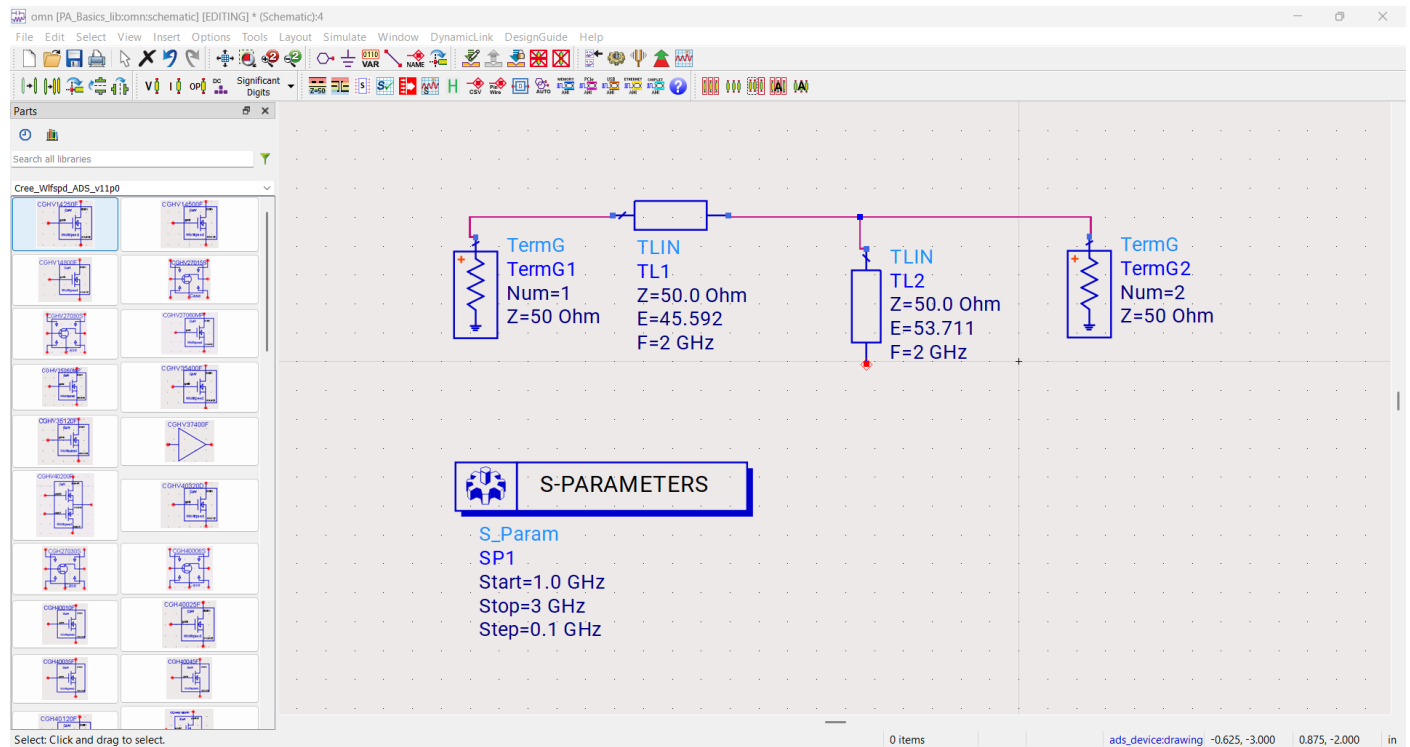
OUTPUT MATCHING NETWORK AT 2GHz:



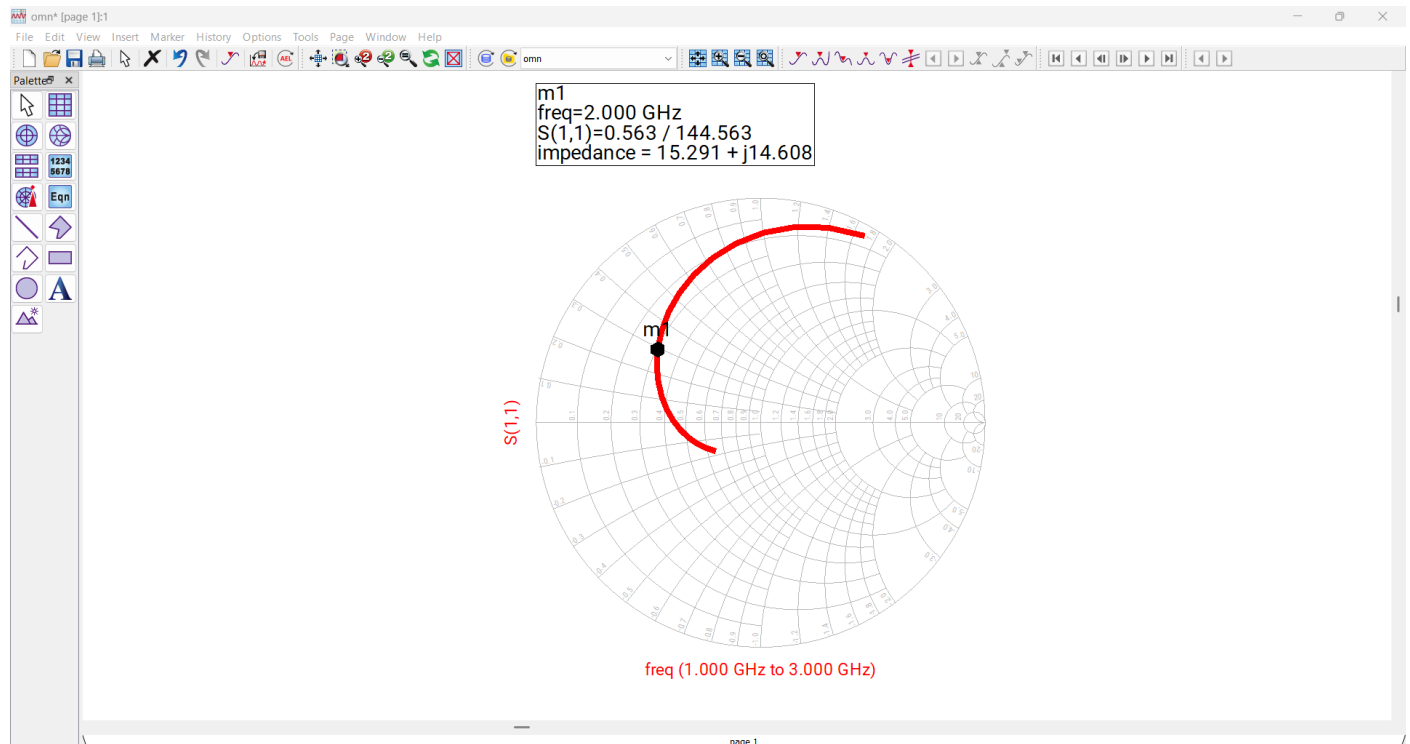
INPUT MATCHING NETWORK AT 2GHz:



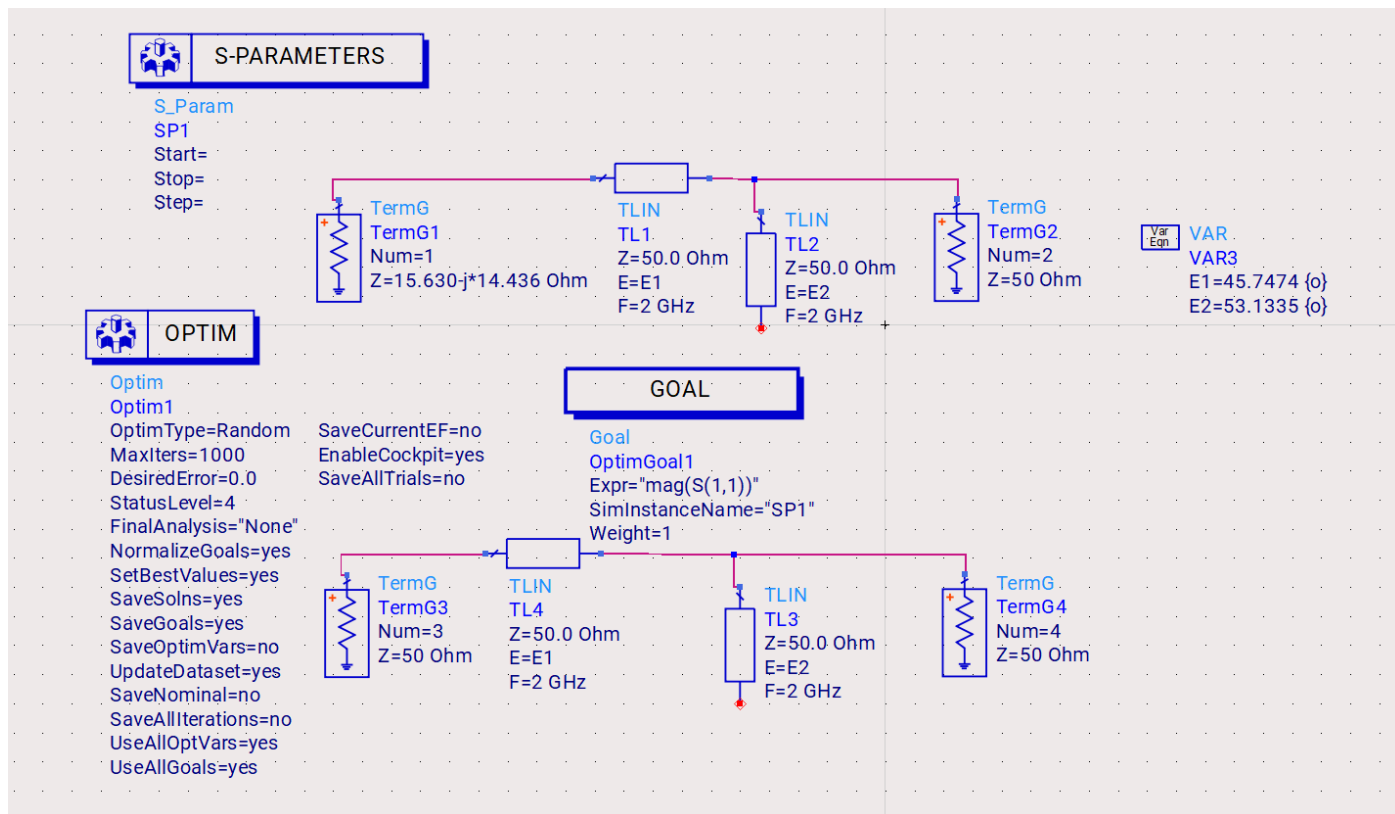
OUTPUT MATCHING NETWORK (WITHOUT OPTIMIZE) SCHEMATICS At 2GHz:



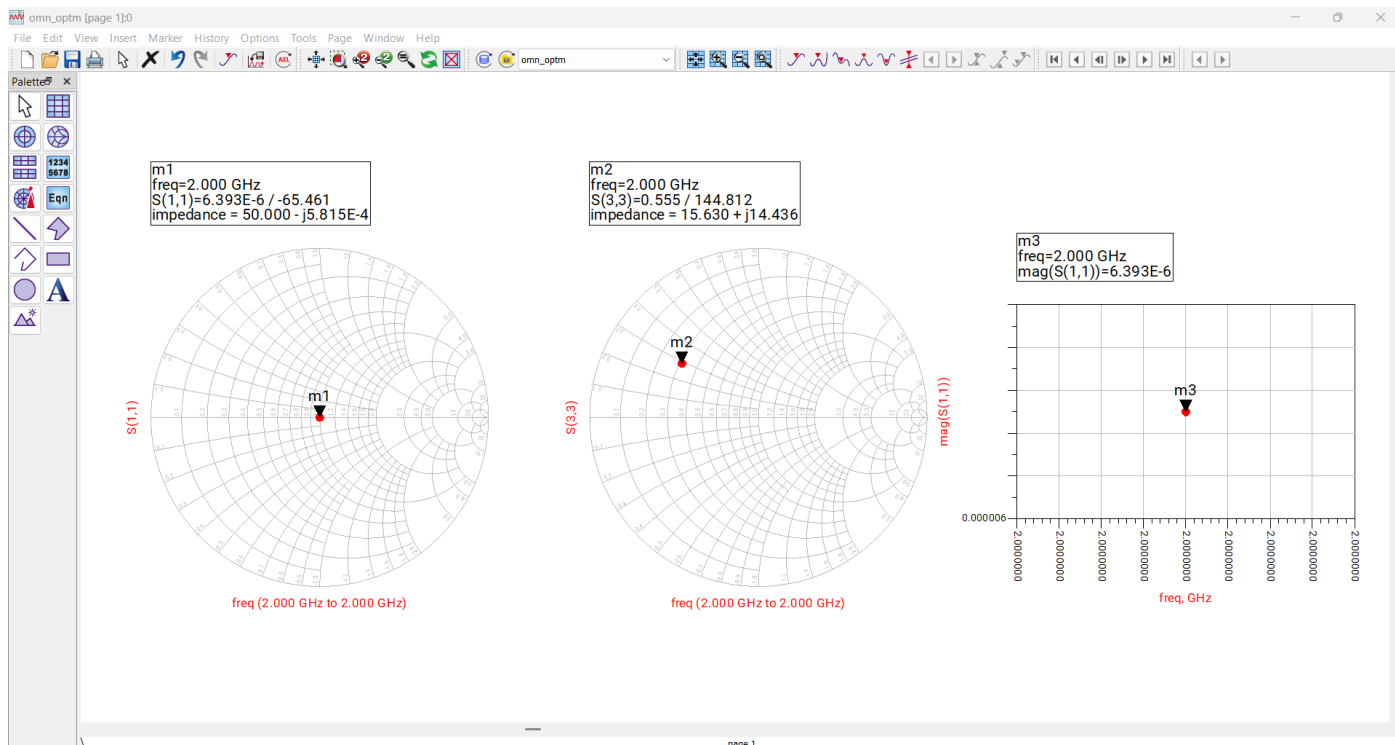
OUTPUT MATCHING NETWORK (WITHOUT OPTIMIZE) RESULTS At 2GHz:



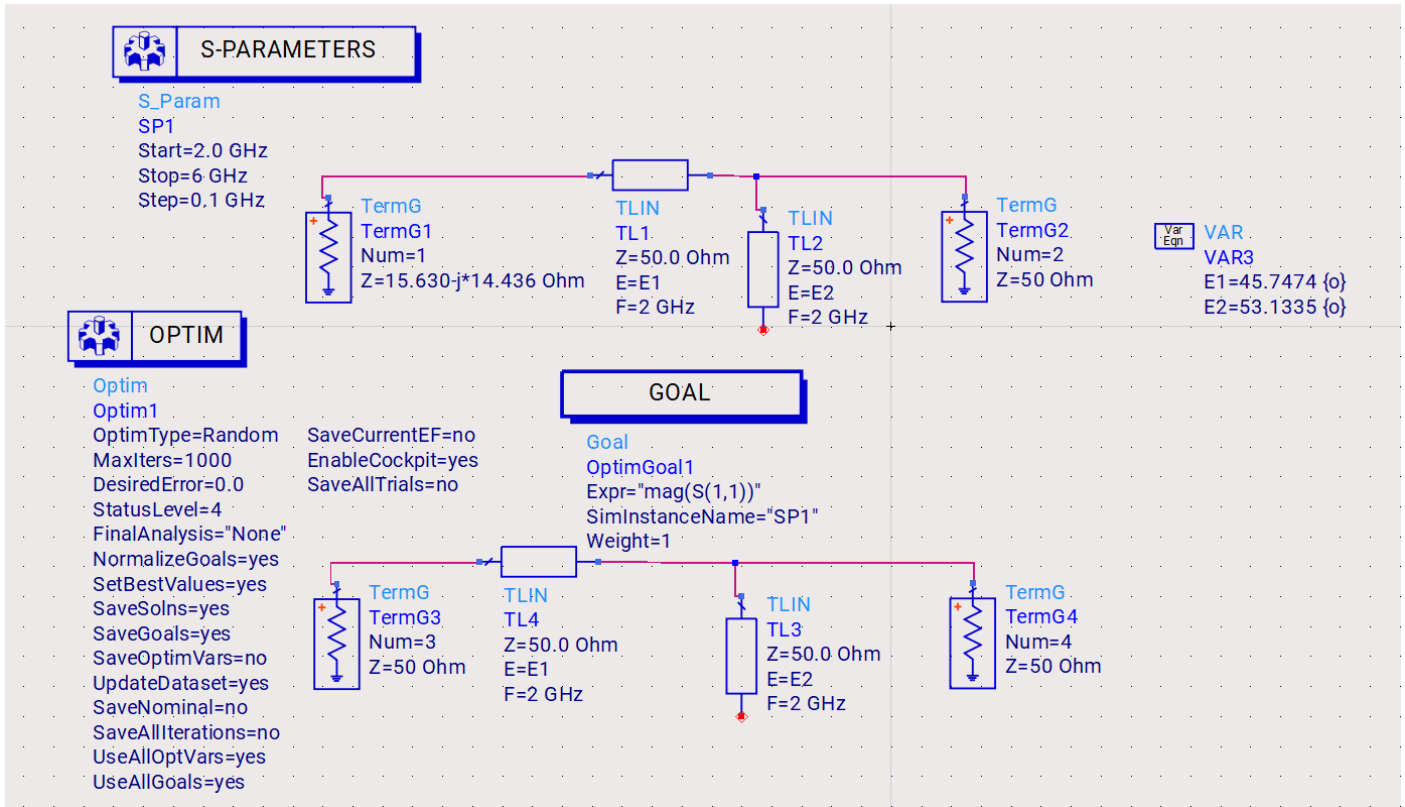
OUTPUT MATCHING NETWORK (WITH OPTIMIZE) SCHEMATICS AT SINGLE FREQUENCY 2GHz:



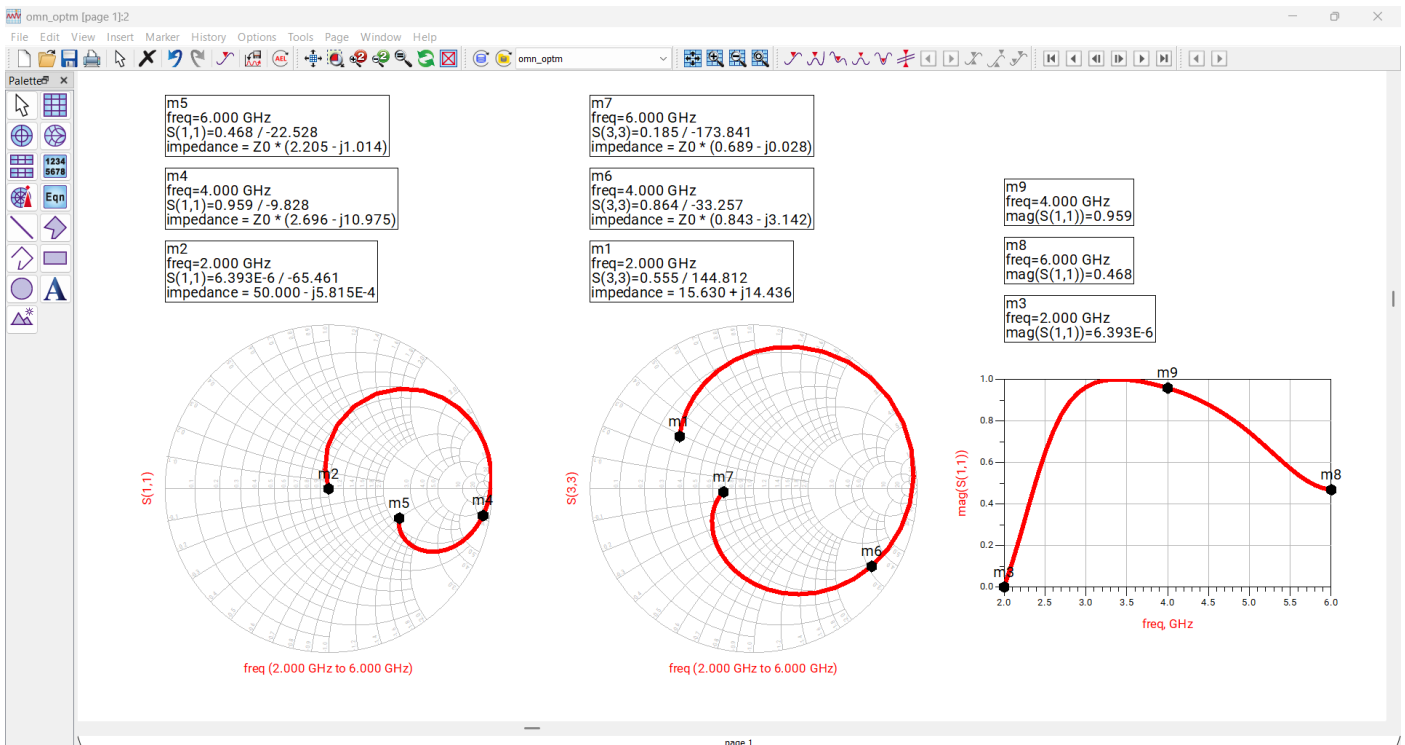
OUTPUT MATCHING NETWORK (WITH OPTIMIZE) RESULTS AT SINGLE FREQUENCY 2GHz:



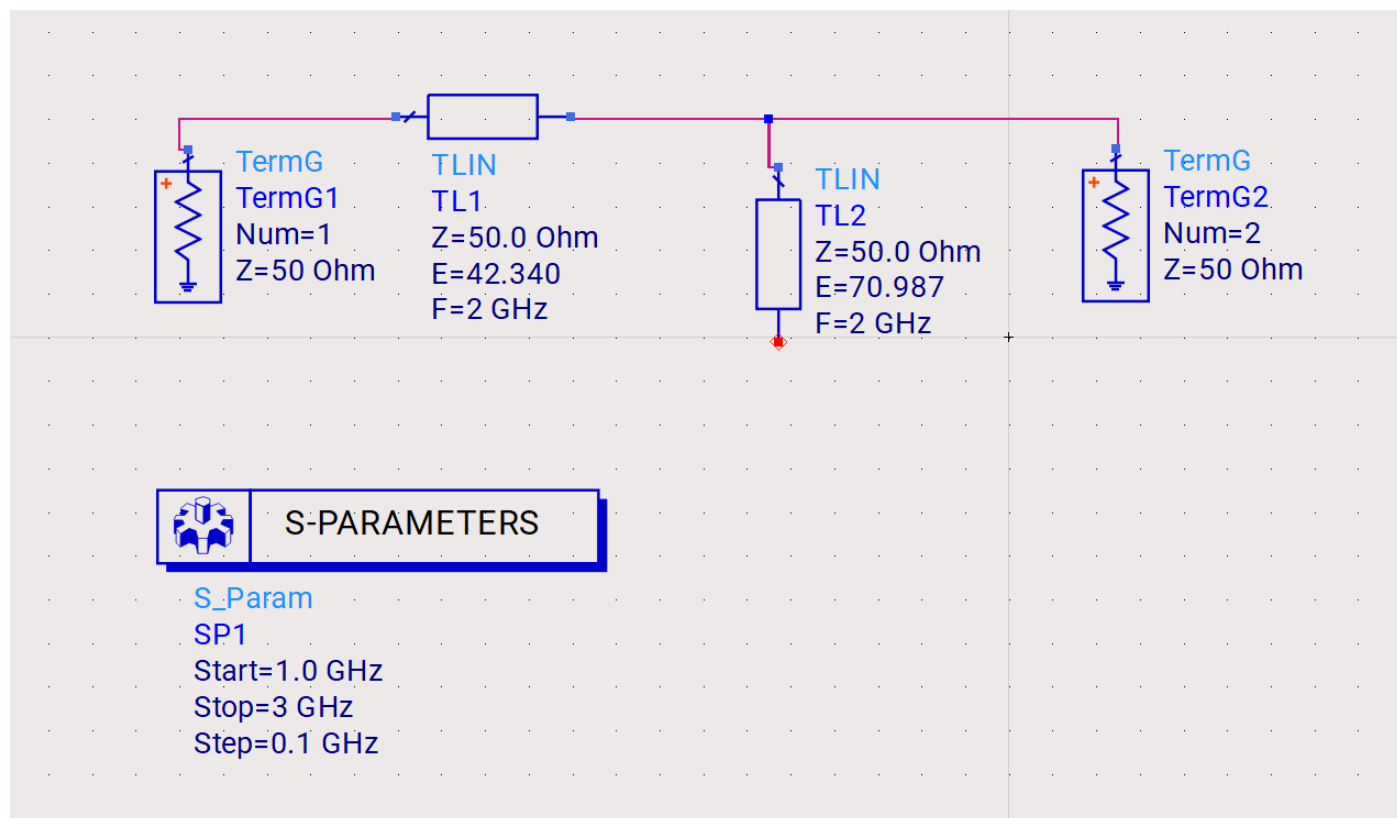
OUTPUT MATCHING NETWORK (WITH OPTIMIZE) SCHEMATICS AT FREQUENCY 2GHz,4GHz and 6GHz:



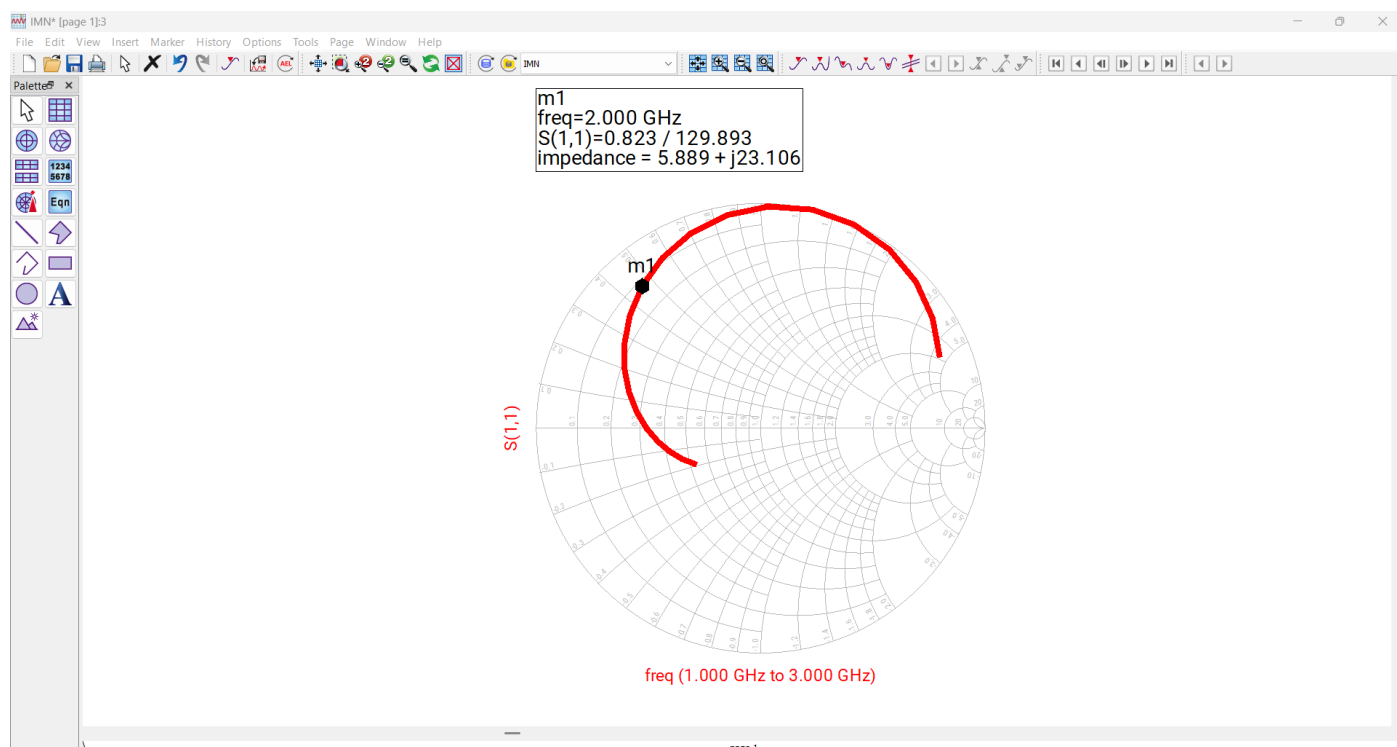
OUTPUT MATCHING NETWORK (WITH OPTIMIZE) RESULTS AT FREQUENCY 2GHz,4GHz and 6GHz:



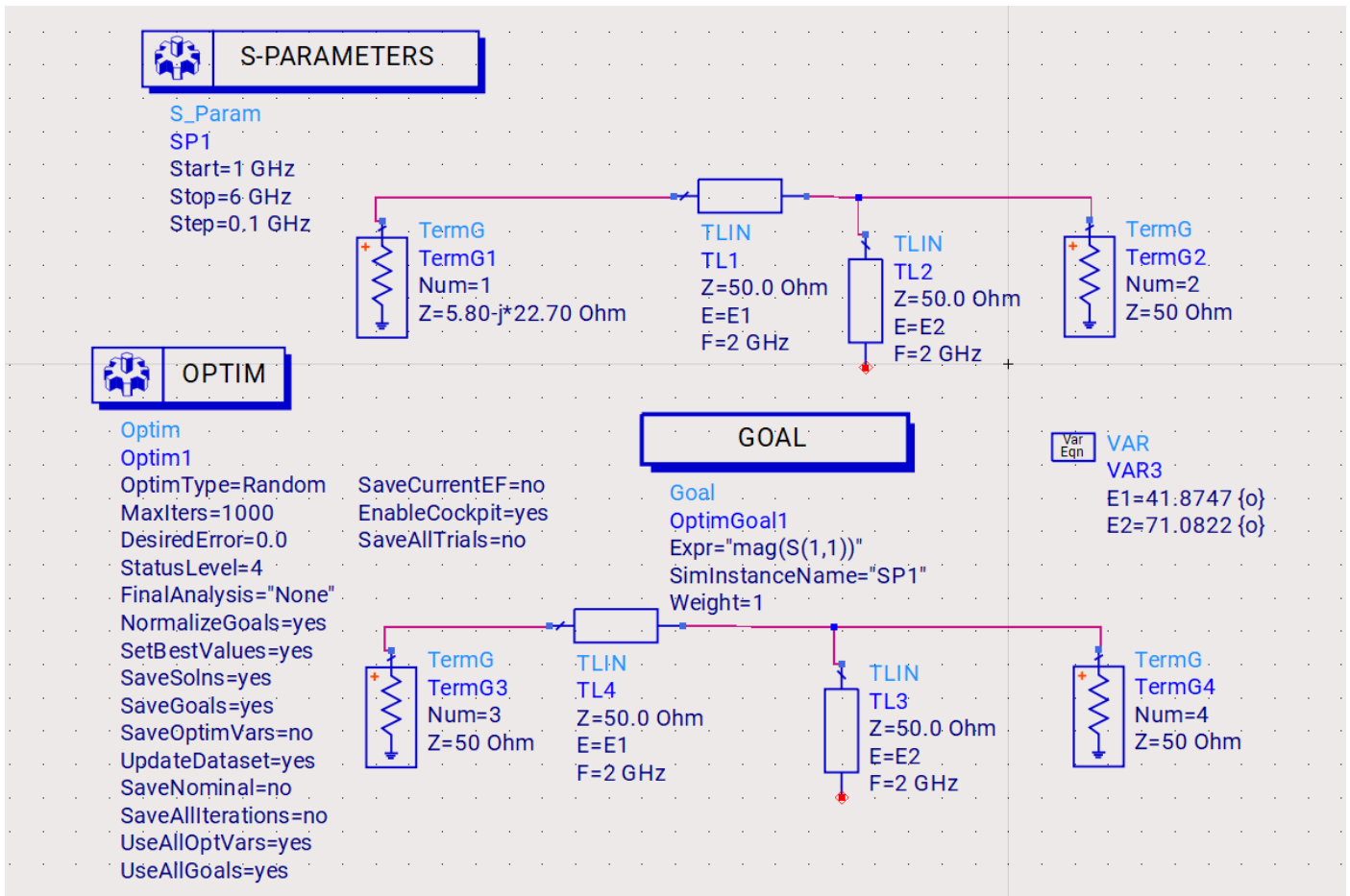
INPUT MATCHING NETWORK (WITHOUT OPTIMIZE) SCHEMATICS At 2GHz:



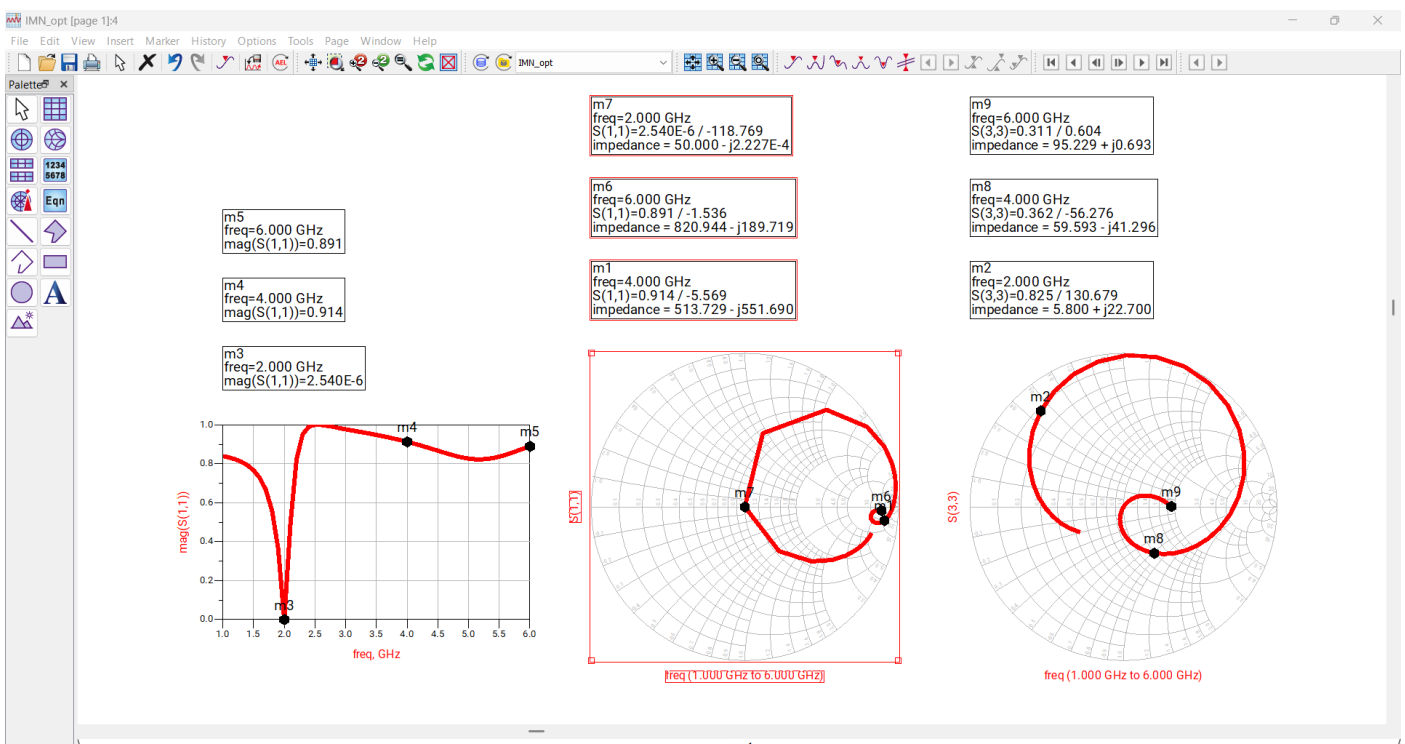
INPUT MATCHING NETWORK (WITHOUT OPTIMIZE) RESULTS At 2GHz:



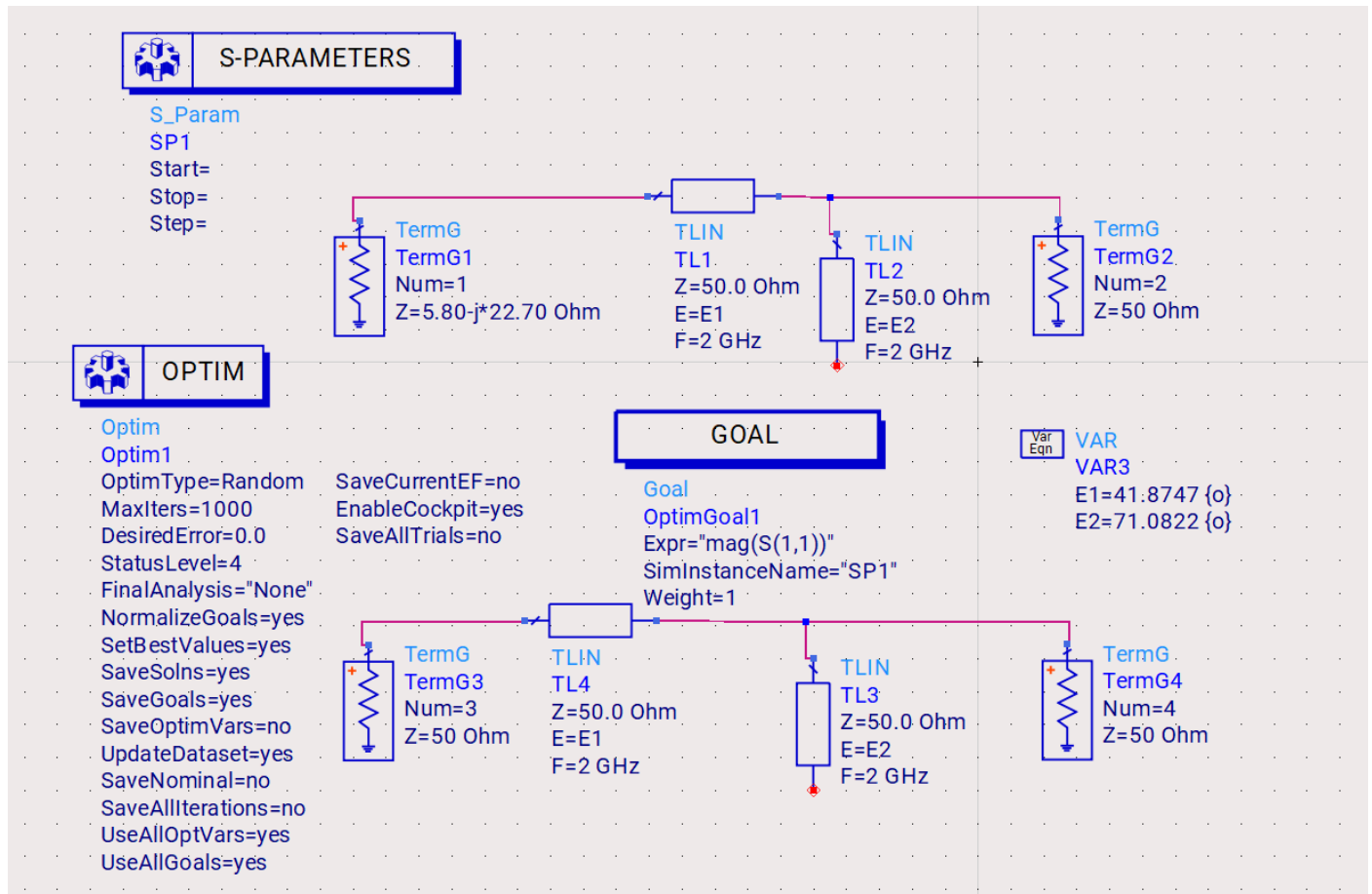
INPUT MATCHING NETWORK (WITH OPTIMIZE) SCHEMATICS AT FREQUENCY 2GHz,4GHz and 6GHz:



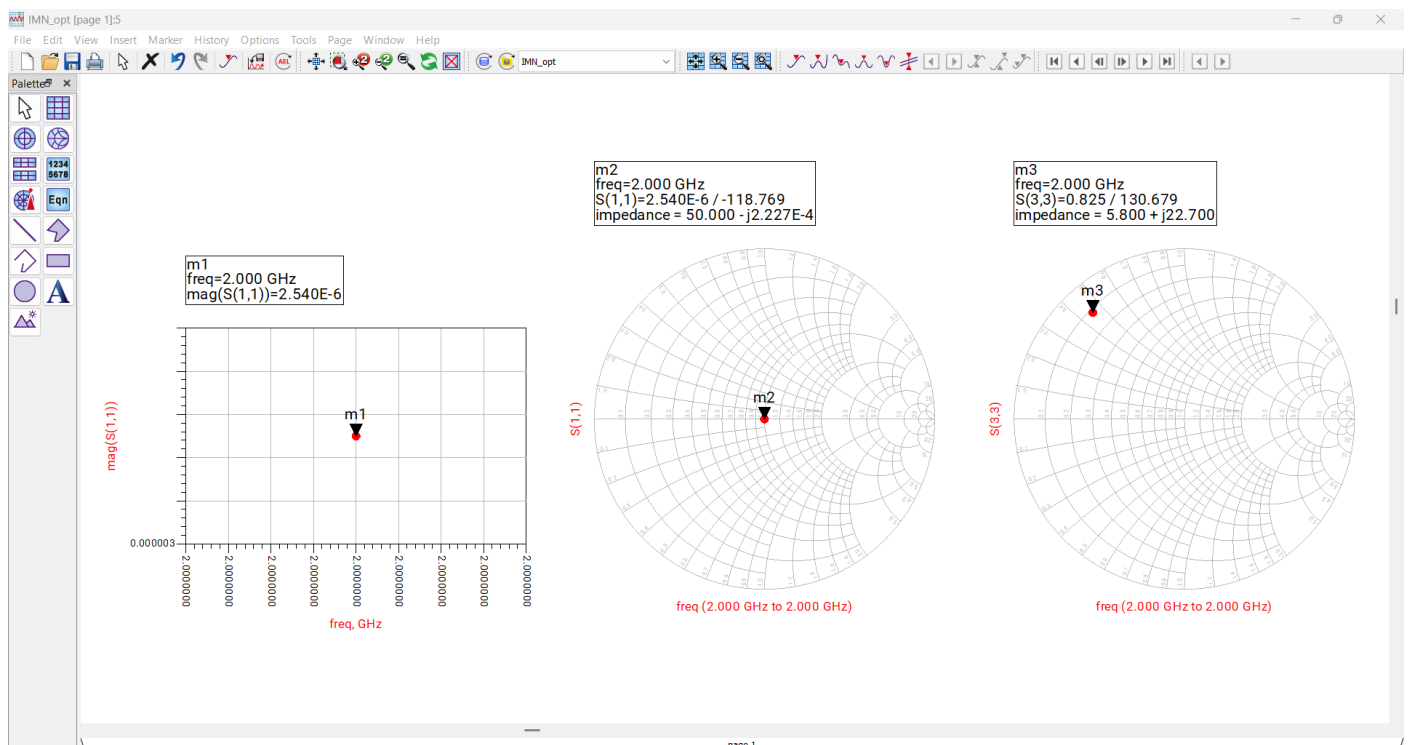
INPUT MATCHING NETWORK (WITH OPTIMIZE) RESULTS AT FREQUENCY 2GHz,4GHz and 6GHz:



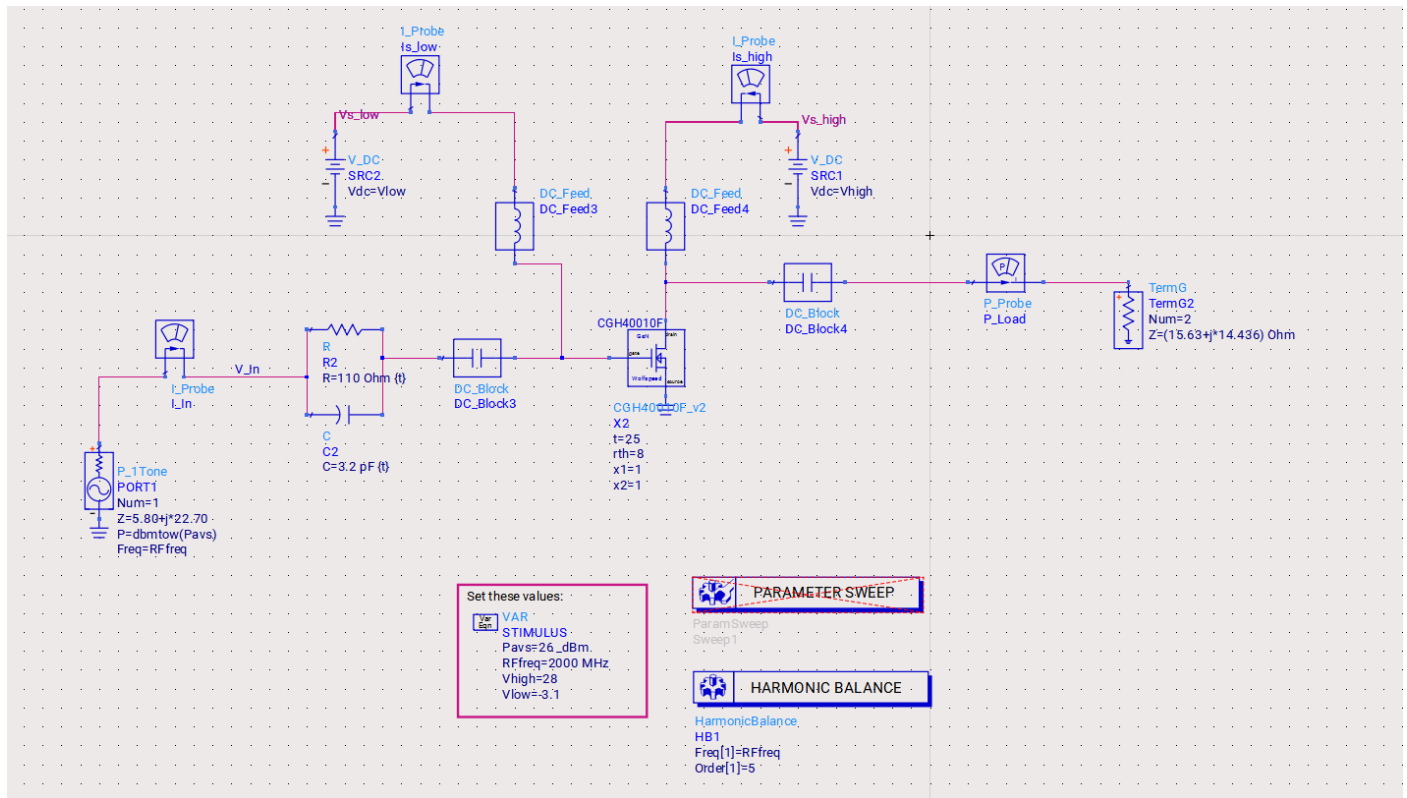
INPUT MATCHING NETWORK (WITH OPTIMIZE) SCHEMATICS AT SINGLE FREQUENCY 2GHz:



INPUT MATCHING NETWORK (WITH OPTIMIZE) RESULTS AT SINGLE FREQUENCY 2GHz:

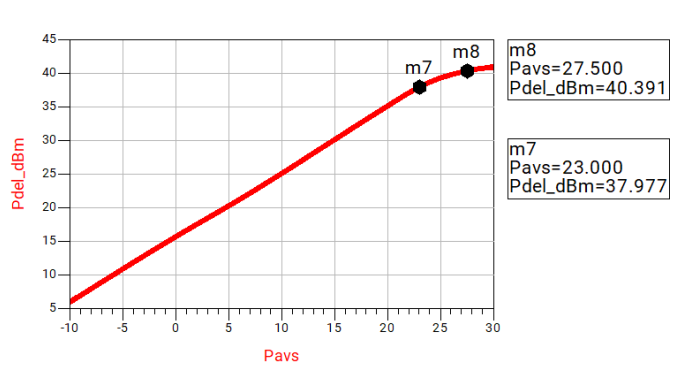
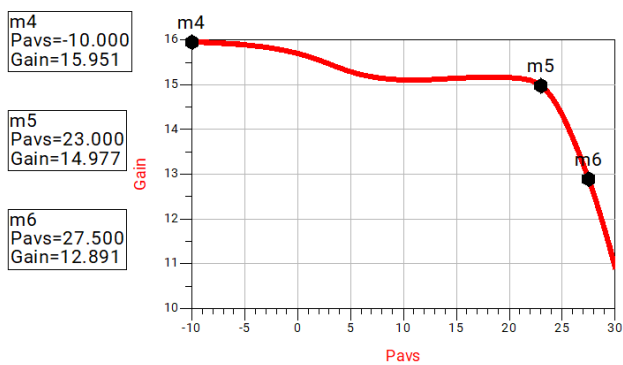
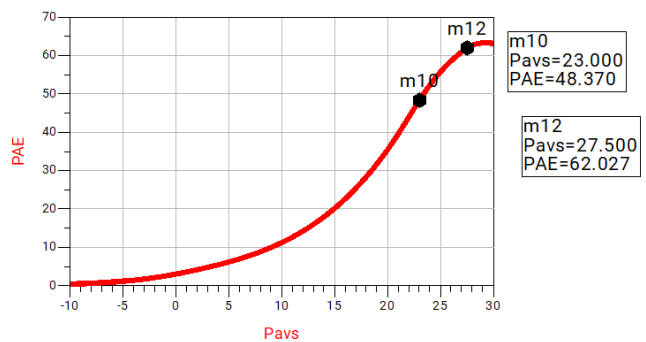
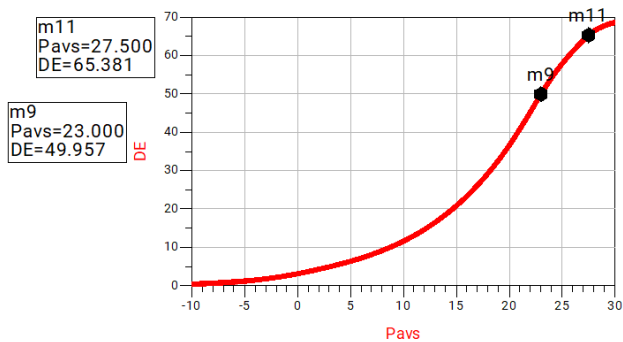


Load Pull Testing with Harmonic Balance Schematic:



Load Pull Testing with Harmonic Balance Drain Efficiency, Power-Added

Efficiency, Gain and Power Delivered (dBm) Results:



DC Power Calculations

The exists() function checks to be sure the corresponding piece of data is in the dataset. If it is not, then the function returns 0.

Eqn Vs_l=exists("real(Vs_low[0])")

Eqn Vs_h=exists("real(Vs_high[0])")

Eqn Is_h=exists("real(Is_high.i[0])")

Eqn Is_l=exists("real(Is_low.i[0])")

Eqn Pdc=Is_h*Vs_h+Is_l*Vs_l+1e-20

Power Delivered and Power-Added Efficiency Calculations

Eqn Pdel_Watts=P_Load.p[1]

Pavs is the available source power, set on the schematic, and passed into the dataset using the Harmonic Balance controller.

Eqn Pavs_Watts=10*((Pavs[0,0]-30)/10)

Eqn PAE=100*(Pdel_Watts-Pavs_Watts)/Pdc

Eqn P_In_Watts=0.5*real(V_In[1]*conj(I_In.i[1]))

This power-added efficiency equation that uses the power absorbed at the input is considered to be a better, more accurate figure of merit than the one that uses the power available from the source.

Eqn PAE=100*(Pdel_Watts-P_In_Watts)/Pdc

Eqn Pdel_dBm = Pdel_Watts>0? 10*log10(Pdel_Watts)+30 : -200

Power-Added Efficiency contour calculations

Eqn PAEmax=max(max(PAE))

Eqn PAE_contours=contour(PAE,PAEmax-0.1-[0:(NumPAE_lines-1)]*PAE_step)

Conversion of the contours to polar coordinates for plotting on Smith Chart

This -0.1 term is included so the contour at the maximum is a small circle, rather than a single point.

Eqn PAE_contours_p=[indep(PAE_contours)+j*PAE_contours]

Conversion of the contours (which are actually curves of complex reflection coefficients that produce a constant PAE) to impedances

Eqn ZL_PAE=zin(PAE_contours_p,Z0[0,0])

Z0 is the reference impedance, set on the schematic, and passed into the dataset using the Harmonic Balance controller.

Re-scaling the contours for plotting on a Smith Chart with a different reference impedance

Eqn PAE_contours_scaled=(ZL_PAE-conj(Z0new))/(ZL_PAE+Z0new)

surface_samples are the actually-simulated reflection coefficients from which the contours are generated. real_indexs11 and imag_indexs11 are the swept variables in the simulation.

Eqn surface_samples=real_indexs11+j*expand(imag_indexs11)

Eqn Z_In_Fund=V_In[1]/I_In.i[1]

Eqn Rho_In_Fund=(Z_In_Fund-conj(Z0))/(Z_In_Fund+Z0)

Power Delivered contour calculations

Eqn Pdelmax = max(max(Pdel_dBm))

Eqn Pdel_contours=contour(Pdel_dBm,Pdelmax-.01-[0:(NumPdel_lines-1)]*Pdel_step)

Conversion of the contours to polar coordinates for plotting on Smith Chart

Eqn Pdel_contours_p=[indep(Pdel_contours) +j*Pdel_contours]

Conversion of the contours (which are actually curves of complex reflection coefficients that produce a constant delivered power) to impedances

Eqn ZL_Pdel=zin(Pdel_contours_p,Z0[0,0])

Re-scaling the contours for plotting on a Smith Chart with a different reference impedance

Eqn Pdel_contours_scaled=(ZL_Pdel-conj(Z0new))/(ZL_Pdel+Z0new)

Eqn DE=100*(Pdel_Watts)/Pdc

Eqn Gain=Pdel_dBm-Pavs

imag_index and real_index are used to calculate the PAE and power delivered that correspond to the reflection coefficient at the m3 marker location.

Eqn imag_index=find_index(imag_indexs11,imag(m3))

Eqn real_index=find_index(real_indexs11[imag_index,:],real(m3))

This calculates the impedance at marker m3.

Eqn Z_at_m3=(conj(Z0[0,0])+Z0[0,0]*m3)/(1-m3)

HARMONIC BALANCE

HarmonicBalance
HB1
Freq[1]=RFfreq
Order[1]=5

PARAMETER SWEEP

ParamSweep
Sweep1

Set these values:

VAR	STIMULUS
Pavs=26_dBm	
RFfreq=2000 MHz	
Vhigh=28	
Vlow=-3.1	

VAR
VAR3
E1=+41.8747 (o)
E2=+71.0822 (o)

VAR
VAR1
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR2
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR4
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR5
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR6
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR7
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR8
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR9
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR10
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR11
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR12
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR13
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR14
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR15
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR16
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR17
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR18
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR19
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR20
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR21
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR22
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR23
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR24
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR25
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR26
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR27
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR28
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR29
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR30
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR31
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR32
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR33
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR34
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR35
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR36
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR37
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR38
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR39
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR40
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR41
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR42
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR43
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR44
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR45
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR46
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR47
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR48
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR49
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR50
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR51
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR52
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR53
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR54
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR55
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR56
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR57
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR58
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR59
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR60
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR61
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR62
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR63
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR64
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR65
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR66
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR67
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR68
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR69
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR70
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR71
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR72
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR73
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR74
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR75
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR76
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR77
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR78
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR79
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR80
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR81
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR82
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR83
E1=+45.7474 (o)
E2=+53.1335 (o)

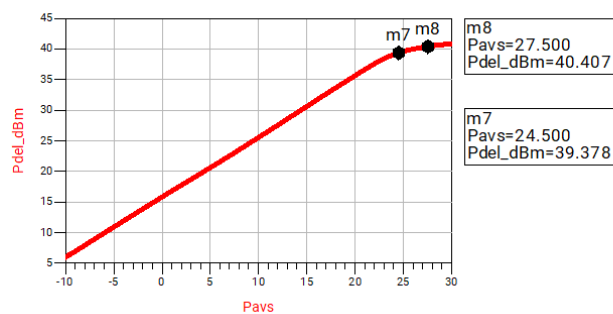
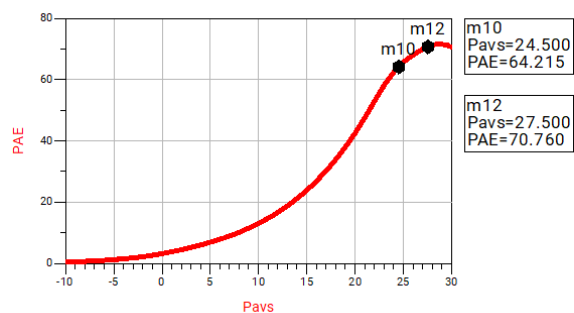
VAR
VAR84
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR85
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR86
E1=+45.7474 (o)
E2=+53.1335 (o)

VAR
VAR87
E1=+45.747

Efficiency, Gain and Power Delivered (dBm) Results:



DC Power Calculations

The exists() function checks to be sure the corresponding piece of data is in the dataset. If it is not, then the function returns 0.

Eqn Vs_l=exists("real(Vs_low{0})")

Eqn Vs_h=exists("real(Vs_high{0})")

Eqn Is_h=exists("real(Is_high.i{0})")

Eqn Is_l=exists("real(Is_low.i{0})")

Eqn Pdc=Is_h*Vs_h+Is_l*Vs_l+1e-20

Power Delivered and Power-Added Efficiency Calculations

Eqn Pdel_Watts=P_Load.p[1]

Pavs is the available source power, set on the schematic, and passed into the dataset using the Harmonic Balance controller.

Eqn Pavs_Watts=10**((Pavs[0,0]-30)/10)

Eqn PAE=100*(Pdel_Watts-Pavs_Watts)/Pdc

Eqn P_In_Watts=0.5*real(V_In[1]*conj(L_In.i[1]))

This power-added efficiency equation that uses the power absorbed at the input is considered to be a better, more accurate figure of merit than the one that uses the power available from the source.

Eqn PAE=100*(Pdel_Watts-P_In_Watts)/Pdc

Eqn Pdel_dBm = Pdel_Watts>0? 10*log10(Pdel_Watts)+30 : -200

Power-Added Efficiency contour calculations

Eqn PAEmax=max(max(PAE))

Eqn PAE_contours=contour(PAE,PAEmax-0.1-[0:::(NumPAE_lines-1)]*PAE_step)

Conversion of the contours to polar coordinates for plotting on Smith Chart

This -0.1 term is included so the contour at the maximum is a small circle, rather than a single point.

Eqn PAE_contours_p=[indep(PAE_contours)+j*PAE_contours]

Conversion of the contours (which are actually curves of complex reflection coefficients that produce a constant PAE) to impedances

Eqn ZL_PAE=zin(PAE_contours_p,Z0[0,0])

Z0 is the reference impedance, set on the schematic, and passed into the dataset using the Harmonic Balance controller.

Re-scaling the contours for plotting on a Smith Chart with a different reference impedance

Eqn PAE_contours_scaled=(ZL_PAE-conj(Z0new))/(ZL_PAE+Z0new)

surface_samples are the actually-simulated reflection coefficients from which the contours are generated. real_indexs11 and imag_indexs11 are the swept variables in the simulation.

Eqn surface_samples=real_indexs11+j*expand(imag_indexs11)

Eqn Z_In_Fund=V_In[1]/I_In.i[1]

Eqn Rho_In_Fund=(Z_In_Fund-conj(Z0))/(Z_In_Fund+Z0)

Power Delivered contour calculations

Eqn Pdelmax = max(max(Pdel_dBm))

Eqn Pdel_contours=contour(Pdel_dBm,Pdelmax-.01-[0:::(NumPdel_lines-1)]*Pdel_step)

Conversion of the contours to polar coordinates for plotting on Smith Chart

Eqn Pdel_contours_p=[indep(Pdel_contours) +j*Pdel_contours]

Conversion of the contours (which are actually curves of complex reflection coefficients that produce a constant delivered power) to impedances

Eqn ZL_Pdel=zin(Pdel_contours_p,Z0[0,0])

Re-scaling the contours for plotting on a Smith Chart with a different reference impedance

Eqn Pdel_contours_scaled=(ZL_Pdel-conj(Z0new))/(ZL_Pdel+Z0new)

Eqn DE=100*(Pdel_Watts)/Pdc

Eqn Gain=Pdel_dBm-Pavs

imag_index and real_index are used to calculate the PAE and power delivered that correspond to the reflection coefficient at the m3 marker location.

Eqn imag_index=find_index(imag_indexs11,imag(m3))

Eqn real_index=find_index(real_indexs11[imag_index,:],real(m3))

This calculates the impedance at marker m3.

Eqn Z_at_m3=(conj(Z0[0,0])+Z0[0,0]*m3)/(1-m3)