

RF PCB Simulation Cookbook

Thomas Almholt

ABSTRACT

This application report describes important PCB design considerations that can be used to maximize the RF performance of any specific hardware design.

Contents

1	Introduction	2
2	Electromagnetic Simulations Capabilities	2
3	Five-Step Electromagnetic Simulation Process.....	3
4	Balun Performance Summary	14
5	RF Filter Performance Summary	16
6	Optional PCB Antenna Performance Using 3D FEM Tool.....	16
7	Matlab Program Developed to Read s-parameter Files and Analyze Performance.....	19

List of Figures

1	How the RF Ports are Added to the Simulation (the length of each port has to be kept approximately equal). .	4
2	Ground Plane Only	4
3	Summary of Ground Impedances	5
4	Poor Return Current Path Caused by Ground Plane Cuts.....	5
5	Summary of Ground Impedances	6
6	Port Generation Dialog Box Inside Ansys SiWave	7
7	PCB Simulation is Ready to Go (the blue line represents the decoupling networks evaluated by the simulation)	7
8	Results of the Decoupling Analysis	8
9	List of 20 Resonances	9
10	First Example Heat Map Showing the Location of an Insignificant Peak Resonance Related to a Connector .	9
11	Second Example Heat Map Showing the Location of a Significant Resonance Related to a Edge to Edge Structure.....	10
12	Third Example Heat Map Showing the Location of a Significant Resonance Related to a Near Edge Structure at 881 MHz	10
13	Full Cross Coupling Matrix of a 51 x 51 Node Analysis (red, yellow and green indicate areas of varying degree of concern)	12
14	Cross Coupling Analysis Focused on the 32 KHz XTAL Only	12
15	Focused View of Just the 32 KHz XTAL and RF Section to Identify the Nets in the Cross Coupling Matrix ..	13
16	Single Ended and Differential Return Loss is Plotted for the Discrete Balun Designed on the CC1312R LaunchPAD	14
17	Insertion Loss, Phase and Amplitude Balance is Plotted for the Discrete Balun Integrated on the CC1312R LaunchPAD.....	14
18	Phase and Amplitude Imbalance is Plotted for the Discrete Balun Integrated on the CC1312R LaunchPAD.	14
19	Theoretical CMRR of a Balun	15
20	Attenuation of the Filter Portion of a Filter Balun at the 2nd and 3rd Harmonic	16
21	Overall Performance of the Antenna Design	17
22	Antenna Pattern for a Typical Monopole Antenna Should be Circular Shaped.....	17

23	Antenna Return Loss	18
24	Screen Capture of Simulation Setup	18

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The main reasons for performing simulations on PCB’s would be to increase the likelihood of a first pass success for the final product.

This document describes a cookbook type approach to executing 2D and 3D electromagnetic simulations to verify that the PCB design has the desired performance. This cookbook is based on executing a five step process and evaluating the results.

The five steps are:

- Quality of the main ground plane
- Round trip impedance of power and ground
- Power and ground resonance
- Isolation between critical nets
- Frequency response of RF traces

There is a sixth item, but it is optional. Using a full 3D solver, it is possible to simulate the antenna performance, radiation pattern, frequency range of a final product. Furthermore, if a 3D model of the plastic enclosure is available it is possible to predicting the influence of plastic enclosures to the final products antenna performance.

The rest of the application report describes the process and how to interpret the results.

2 Electromagnetic Simulations Capabilities

There are a number of 2½ and 3D simulations tools available. They are all quite powerful, but all different. The main ones are the Agilent Suite and the Ansys Suite.

The selection of tools is based on the following factors:

- Capable to perform the needed simulations (2½ and 3D)
- Capable to import PCB artwork directly from CAD packages
 - Including importing net names and component placement and values
- Execution speed of the simulations

2.1 Capable of Performing 2½ and 3D Simulations

For the most generic simulation capability, Full 3D solvers were used, like Ansys HFSS or Agilent ADS-FEM. These tools are capable of simulating any 3D structure and solving for complex coupling including antenna radiation patterns. Both of these tools require extended execution times (many hours) and required computers equipped with 64GB of RAM or more.

For PCB related simulation most of the features are planar in nature and the tools can be optimized for such structures thereby greatly reducing the execution time and memory resources required. These planar focused simulators are generically called 2½D solvers. When running simulations on the CC1352R LaunchPADs these 2½D solvers can finish the simulation in less than 10 minutes and requires less than 2GB of RAM.

2.2 Capable of Importing PCB Artwork Directly From CAD Packages

When doing simulations of PCB's, the ease of importing a database and getting the first simulations to run is very important. Each of the tools provides in excess of 10 types of import options and it is a major focus for the tools developers to make sure the import capabilities are comprehensive.

These import capabilities includes filters to directly import databases from Cadence, Mentor and Altium and AutoCad. To further enhance the import flexibility they include filters capable of importing ODB++ databases. ODB++ databases are quickly becoming the norm for sending production files to manufacturing in preference to Gerber files and therefore almost all tools can provide such an export capability.

NOTE: The author of this application note is mainly familiar with the Ansys products.

2.3 Execution Speed of the Simulations

The main advantage of the planar solvers are execution times for the simulations, this advantage is in most cases an increase in speed of 10 times or more. Additionally the execution times does not exponentially increase with problem size as it does for 3D solvers.

The main drawback of the 2½D solvers is that they are approximating the coupling between next when signals are traversing the layers using via's. This approximation is something to be aware off and sometimes to gain confidence in the results from the 2½D solvers it is good practice to run a simulation using a full 3D solver as a reference.

3 Five-Step Electromagnetic Simulation Process

The most important part of running a simulation is trying to figure out what to analyze and afterwards how to determine if a certain results is good or bad. A five step process was developed that has proven to be very good at catching most layout related issues.

- Quality of the main ground plane
- Roundtrip impedance of power and ground
- Power and ground resonance
- Isolation between critical nets
- Frequency response of RF traces

The following sections provide more details about what each analysis is designed to catch and how to execute the particular simulation.

3.1 Quality of the Main Ground Plane

On the surface the statement "Quality of the ground plane" sounds pretty subjective. Often times during the layout process of a multilayer PCB, there are signals that are being routed through or on the ground plane causing the ground plane to no longer be a single continues sheet of copper. In many cases, it is unavoidable to route these signals through the ground plane, but it is a good practice to evaluate the final ground plane structure and estimate the performance of it.

To execute this analysis. the engineer has to place a number of ports on the ground plane directly to force the tool to perform a ground to ground impedance analysis. These ports have to be placed manually and, therefore, some care has to be taken to get repeatable results. It is recommended to use no less than eight ports from the middle of the package in each of the major directions of the package. Most of the RF packages in use today are QFN style packages, which have a large "heatsink" on the bottom of them. This large piece of metal is a typically used as the common point for all the simulations in this analysis.

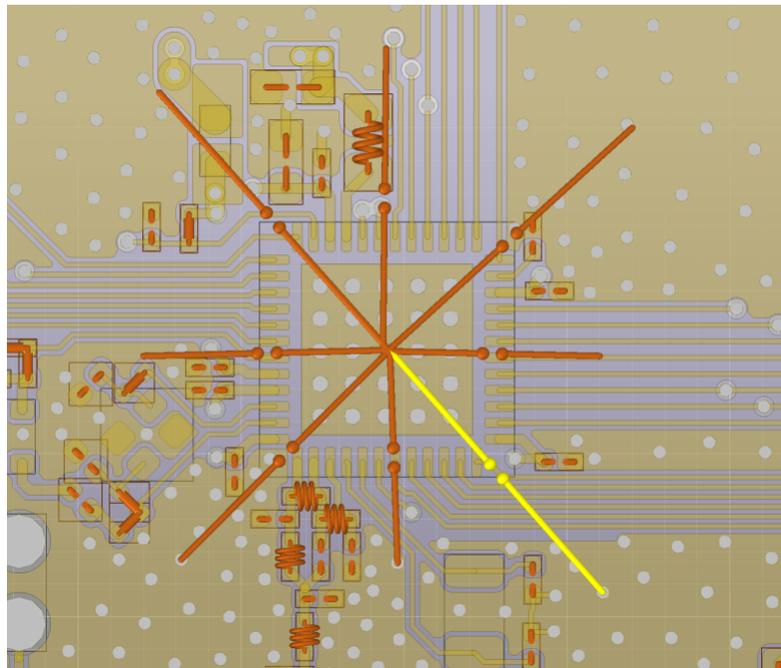


Figure 1. How the RF Ports are Added to the Simulation (the length of each port has to be kept approximately equal)

Figure 2 is of the ground plane only. Here, the small holes represent ground vias that are being drilled thru all layer and the bigger holes are signal vias that carry signal. The smaller ground vias do not impact the performance, because they are filled with a copper during the manufacturing process and the simulation tools takes this into account.

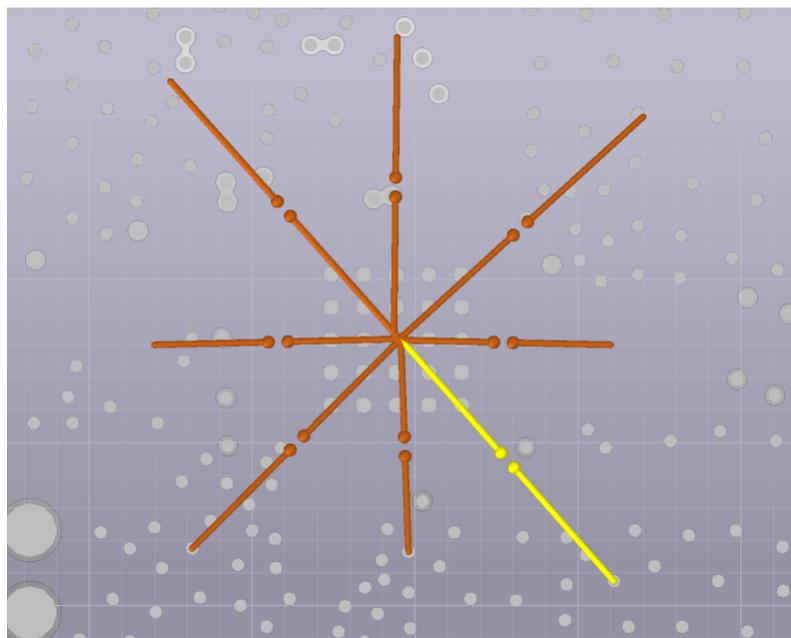


Figure 2. Ground Plane Only

The results from the simulation are typically s-parameters, but they can be translated into Z-parameters and here tabulated for three frequencies only. Typically, only 100 MHz, 1 GHz and 2 GHz are evaluated.

To evaluate whether the results indicates good performance or bad performance, a target has to be set. The target is based on being 10 times better than the impedance of the entire decoupling network simulated in the next step. So, the target is set here to 2.5 Ω at 1 GHz versus later the target will be 2.5 Ω at 100 MHz.

Figure 3 shows a summary of ground impedances are found by the simulation tool, color coded for quick review. Target for this simulation is better than 2.5 Ω at 1 GHz.

Input impedances LAUNCHXL-CC1312R1a.s8p				
	1.05E+08	1E+09	2E+09	[Hz]
{GND_0}	0.022	0.207	0.411	[Ohms]
{GND_1}	0.015	0.136	0.27	[Ohms]
{GND_2}	0.019	0.179	0.356	[Ohms]
{GND_3}	0.014	0.131	0.261	[Ohms]
{GND_4}	0.014	0.129	0.256	[Ohms]
{GND_5}	0.01	0.089	0.178	[Ohms]
{GND_6}	0.017	0.161	0.32	[Ohms]
{GND_7}	0.014	0.128	0.254	[Ohms]
Target frequency is 1GHz (second column)				
	0	Target		
	0.5	Ok		
	2.5	Need improvements		

Figure 3. Summary of Ground Impedances

For completeness, perform an analysis to see what happens if a ground plane has a signal routed inside causing a large cutout to be present and also the ground plane has been moved to the layer at the bottom of the design causing all the via's to become longer. Figure 4 shows an example of a signal being routed on the ground plane and the blue line represents the detour the current must take to return to the origin.

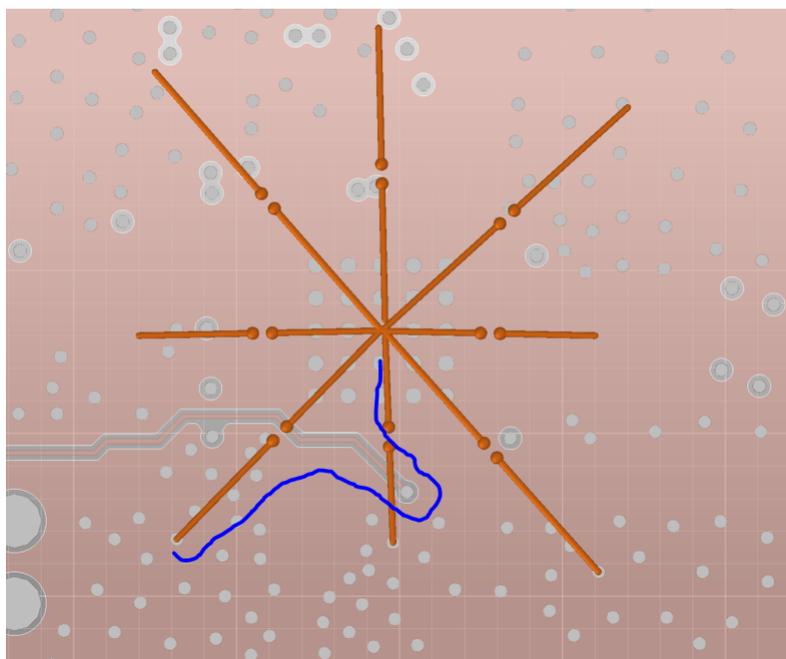


Figure 4. Poor Return Current Path Caused by Ground Plane Cuts

Figure 5 shows a summary of ground impedances are found by the simulation tool, color coded for quick review.

Input impedances LAUNCHXL-CC1312R1a.s8p			
	1.05E+08	1E+09	2E+09
{GND_0}	0.146	1.371	2.737
{GND_1}	0.092	0.864	1.731
{GND_2}	0.071	0.675	1.348
{GND_3}	0.059	0.552	1.1
{GND_4}	0.117	1.107	2.209
{GND_5}	0.075	0.711	1.419
{GND_6}	0.099	0.932	1.863
{GND_7}	0.174	1.646	3.295
Target frequency is 1GHz (second column)			
	0	Target	
	0.5	Ok	
	2.5	Need improvements	

Figure 5. Summary of Ground Impedances

By comparing the two tables, it can be found that most of the nets saw an increase in impedance of 3-4 times, however, two of the nets got 10 times worse. In this particular case, the performance is marginal; it becomes a judgment call on what to do.

Now that a good understanding of how the ground plane is performing has been presented, you can move to the next step in the process and start looking at the decoupling networks.

3.2 Roundtrip Impedance of Power and Ground

The next simulation to execute is a simulation looking into the performance of the power decoupling networks. This analysis is based selecting a common point, typically the middle of the RF package and then running simulations to estimate the impedances of entire power network. This simulation takes into account the component values chosen, the routing to each decoupling capacitor and the routing returning from the decoupling capacitors.

To run this type of simulation a port is required on all the power related nets of the package, in the newer tools this is an automated process. Inside the Ansys SiWAVE tool, this process is enabled by a feature called “Circuit Element Generation”.

Inside the “Circuit Element Generation” feature menu, you can find the part-name tab and navigate to the main device or devices in your design, the example here uses the CC13121F3. The tool will then populate all the port names for both “positive terminal” and “reference terminal”. Select all the active nodes on the positive terminal page, leave out the “GND” net and then select just one net (GND) for the reference terminal.

Press “Create” to generate all 48 ports at the same time.

Figure 6 illustrates the port generation dialog box inside Ansys SiWave, used to automatically generate RF ports between various nets in the design.

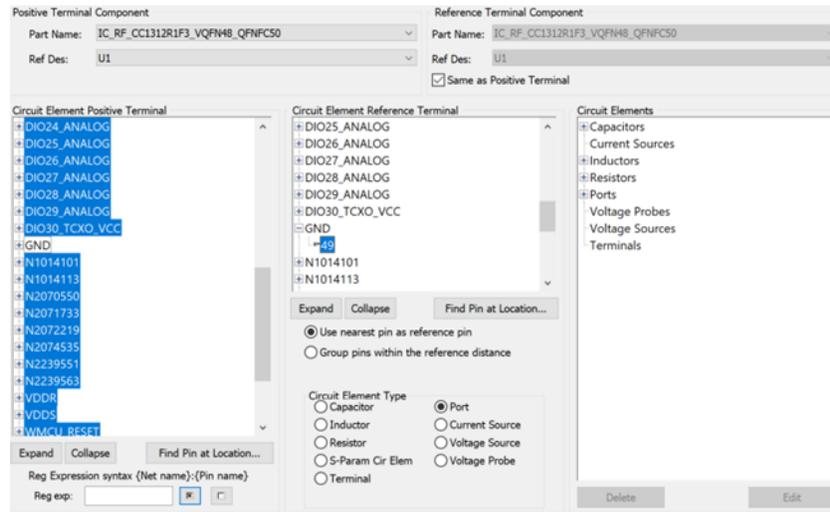


Figure 6. Port Generation Dialog Box Inside Ansys SiWave

The PCB simulation is ready to go. The blue lines in Figure 7 illustrate the return path the current takes when a decoupling capacitor is placed. This is what the tool will be solving. When the simulation project is ready to go, it will look something like the image shown in Figure 7.

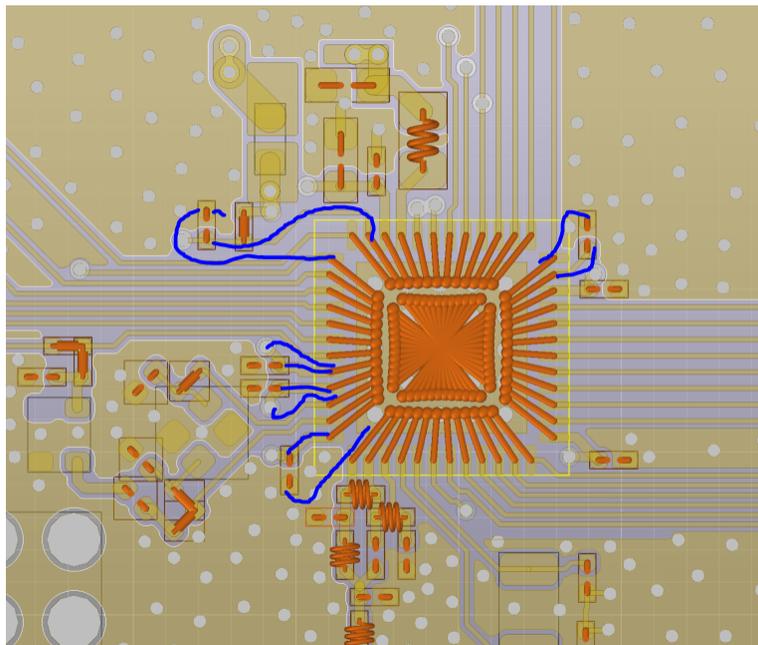


Figure 7. PCB Simulation is Ready to Go (the blue line represents the decoupling networks evaluated by the simulation)

The simulation should be run to include frequencies up to 2x the highest operational frequency of the design. The default setup is 5 MHz to 5 GHz in 401 steps; this data can then be used for the rest of the analysis.

The decoupling analysis particularly focused on the 100 MHz frequency range; 100 MHz is below where on-chip decoupling starts to work and at the upper range of where the large bulk decoupling capacitor works.

The results of the decoupling analysis provides insight into performance of the decoupling networks of the design. In [Figure 8](#), the results in the 100 MHz column are most important, though it is good practice to look for any outliers in the higher ranges. For this particular layout, the power nets are all labeled VDD*.

Input impedances MCU049B-LAUNCHXLCC1352P-2.s49p			
	1.05E+08	1E+09	2E+09
{N3101171_U1_23}	0.283	2.663	5.348
{N3101179_U1_46}	1.52	14.505	31.421
{N3101223_U1_47}	1.419	13.501	28.983
{RX_TX_U1_7}	9.158	10.514	33.485
{VDDR_U1_45}	0.297	2.904	5.966
{VDDR_U1_48}	0.479	4.62	9.715
{VDDS_U1_13}	0.25	2.463	5.122
{VDDS_U1_22}	0.424	4.098	9.709
{VDDS_U1_34}	0.263	2.524	5.116
{VDDS_U1_44}	0.355	3.453	7.152
{WMCU_RESET_U1_35}	1.482	14.395	30.15
Target frequency is 100MHz (first column)			
	0	Target	
	0.5	Ok	
	2.5	Need improvements	

Figure 8. Results of the Decoupling Analysis

Results for this simulation indicate that a very good layout has been done and the highest decoupling impedance found is under 0.5 Ω at 100 MHz, which is a good result.

3.3 Power and Ground Resonance

On larger EVM’s, the particular power and ground structures could be resonating within the frequency of operation. To execute this simulation, it is recommended to start at 100 MHz and then stop after 20 resonances. The tool will find many resonances that are not important, like the metallization around a USB connector that might resonate. Some good engineering discipline is required to distinguish between important and not important resonances, however, with practice this becomes very fast.

To illustrate this in a couple of screen captures of what is considered a potentially important finding and something that is not important at all are provided.

When evaluating a particular design, the tool has found 20 potential resonances between 100 MHz and 923 MHz. These results will be plotted sequentially and it makes sense to start from the top and work down into the stack-up. For a 4 layer stack-up, it makes sense to start with the top-layer versus layer-2, next plot layer-2 versus layer-3, and so on. Even though it is possible to plot all combinations of layers, it does not make sense.

Figure 9 shows a list of 20 resonances and their respective significance measured by evaluating the Q factor (quality factor of the resonance).

M...	Re. Freq (...)	Im. Freq (G...	k	Wavel...	Q
1	0.104100208	0.000493130	2.18177902	2.8798...	105.55...
2	0.319390357	0.006297515	6.69392689	0.9386...	25.363...
3	0.367491672	0.006210840	7.70205591	0.8157...	29.588...
4	0.383564469	0.006720451	8.03891683	0.7815...	28.541...
5	0.434399103	0.005670446	9.10433198	0.6901...	38.307...
6	0.539366858	0.002206887	11.30429343	0.5558...	122.20...
7	0.667704793	0.011217038	13.99405767	0.4489...	29.767...
8	0.692691061	0.013678494	14.51773113	0.4327...	25.325...
9	0.693985099	0.005510299	14.54485215	0.4319...	62.973...
10	0.709073055	0.014239319	14.86107232	0.4227...	24.903...
11	0.725845312	0.013823860	15.21259283	0.4130...	26.258...
12	0.823620399	0.015228690	17.26180713	0.3639...	27.046...
13	0.830852159	0.014327810	17.41337362	0.3608...	28.998...
14	0.848057074	0.017257891	17.77396197	0.3535...	24.575...
15	0.873819384	0.004203345	18.31390007	0.3430...	103.94...
16	0.881149806	0.015042567	18.46753435	0.3402...	29.292...
17	0.894367239	0.014992574	18.74455126	0.3352...	29.831...
18	0.916410280	0.017815740	19.20653923	0.3271...	25.723...
19	0.919917205	0.016283210	19.28003895	0.3258...	28.251...
20	0.923041637	0.004139032	19.34552220	0.3247...	111.50...

Figure 9. List of 20 Resonances

The resonance analysis tool has highlighted a resonance at 383 MHz, but when plotted it looks like the image in Figure 10

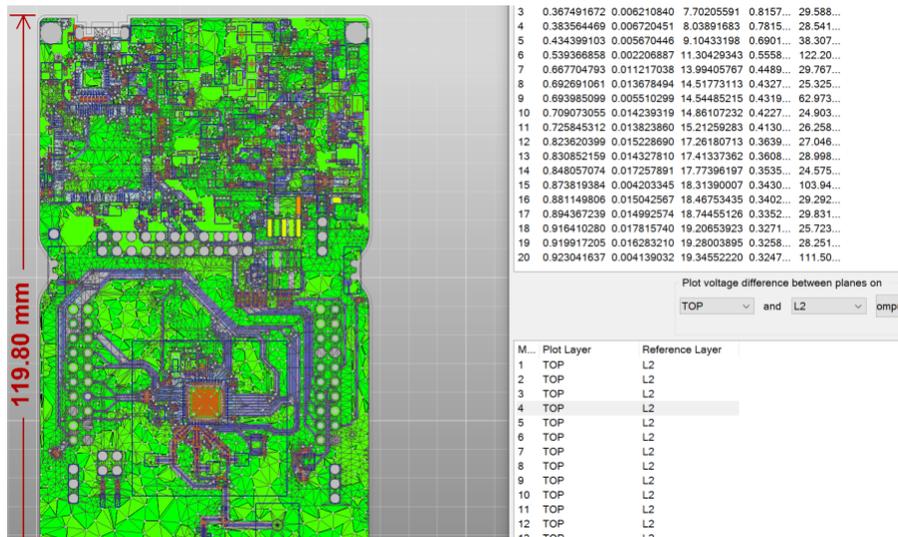


Figure 10. First Example Heat Map Showing the Location of an Insignificant Peak Resonance Related to a Connector

It is almost perfectly green, except for the pads of the JTAG connector. This is not a concern and this finding can be safely disregarded.

However, a layer-2 to layer-3 resonance at 693 MHz reveals this power plane that is not adequately decoupled at each end and, therefore, resonating. This particular frequency is 693 MHz and below the operation of the board, however, this kind of resonance is capable of carrying a signal across the board with very low insertion loss and the design runs a risk of an un-intended cross coupling.

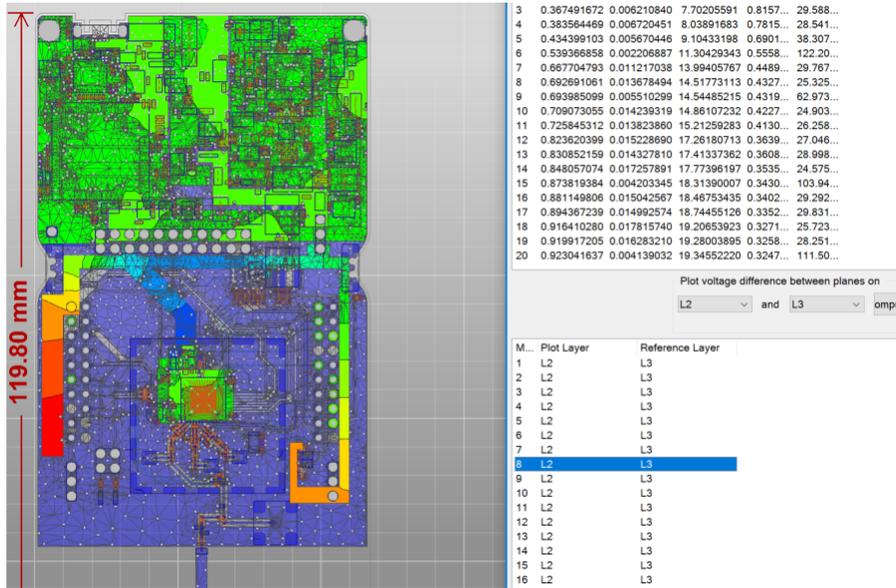


Figure 11. Second Example Heat Map Showing the Location of a Significant Resonance Related to a Edge to Edge Structure

A little higher up in the frequency range, in resonance number 16 at 881 MHz, there is an additional resonance in this design that could cause issues, highlighted in Figure 12 in yellow. This resonance is a power plane in the top right hand corner of the design.

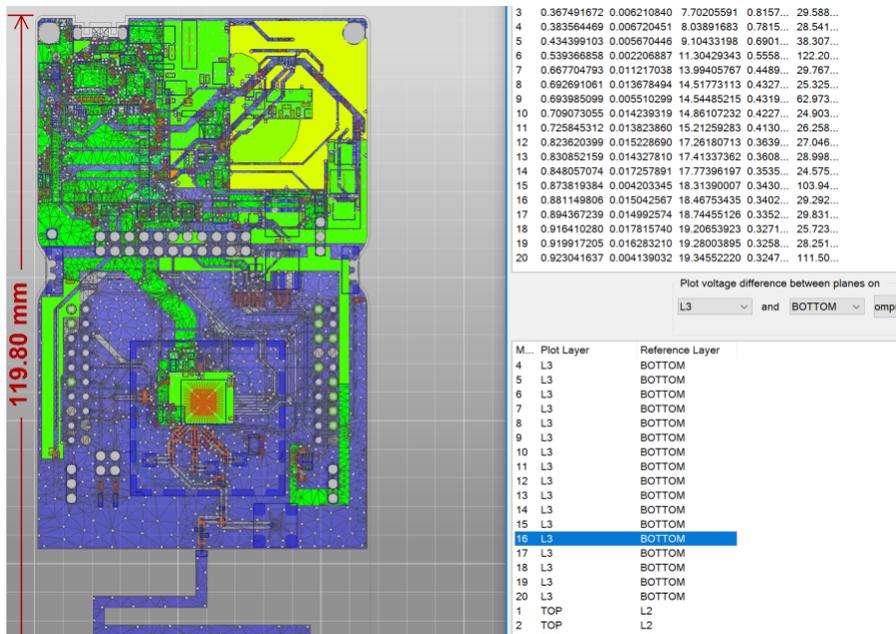


Figure 12. Third Example Heat Map Showing the Location of a Significant Resonance Related to a Near Edge Structure at 881 MHz

This particular resonance has access to the edge of the PCB and, therefore, the designer runs the risk of having a signal from deep inside the design be carried to the edge of the board and being radiated, effectively becoming an antenna. This particular resonance is in the low frequency analog portion of the design, should not create trouble but a risk is taken here.

3.4 Isolation Between Critical Nets

Critical nets are either nets that have been identified as victim or aggressors in a particular design. To determine what nets are critical, a detailed analysis of what type of signals are carries on each of the pins is required.

Here are some guidelines:

- Aggressors (only)
 - Digital clock signals, like SPI, I2S, and JTAG clocks.
 - These signals are typically rail-to-rail signals being driven by digital CMOS drivers capable of driving 4 mA or 8 mA loads. A good approximation of a 4mA port is a 50 Ω source impedance with a 10 dBm signal strength. An 8 mA port is lower impedance of 25 Ω and stronger signal, approximately 13 dBm.
 - These clocks will have harmonics of their fundamental frequency and there will be very little attenuation as the frequencies goes up.
 - Digital data signals can also be aggressors, they will have similar signal strength, but the energy will be randomized by the data pattern that is being carried. However, for simplicity in analysis, it is best to assume the worst and treat the data signals as clocks also.
- Aggressors and Victims
 - The high frequency XTAL and their paths are special as they can be both aggressor and victim.
 - The XTAL signal is smaller, typically only 0dBm, however, they are at the same time the reference for the entire system so any signal that happens to land on the XTAL paths can produce unintentional signals on the RF outputs.
 - The RF output path can also be both a victim and an aggressor as cross coupling to other RF nets could cause RF leakage issues and, therefore, compliance issues.
- Victims only
 - If the design is using an RX only path, then this path will contain no possible way to become an aggressor but still a victim for RF interference.
 - The 32 KHz XTAL clock and its associated paths in a victim only, the 32 KHz XTAL contains a very small signal that can quickly be disturbed.

It is difficult to generalize on what isolation is required at any given frequency for all designs; however for many designs if an isolation of greater than 60-70dB has been achieved at 1 GHz to 2 GHz, then the likelihood of any issues is greatly reduced. But in some cases as with a digital clock to an RF input, the target should be higher, closer to 100dB.

In this design a 48-pins QFN package is used and 3 additional RF ports on the PCB have been added, therefore this becomes a 51x51 cross coupling matrix. To make it simpler to make determinations on the results it is recommended to focus in on one item at a time and identify which columns and rows are responsible for that particular function. As an example, this document focuses on the 32 KHz XTAL nets. Results of the cross coupling analysis provides insight into what potential nets might have enough cross coupling that will cause issues.

In Figure 13, green is better than 100dB, yellow is 70dB and red is 40dB or worse.

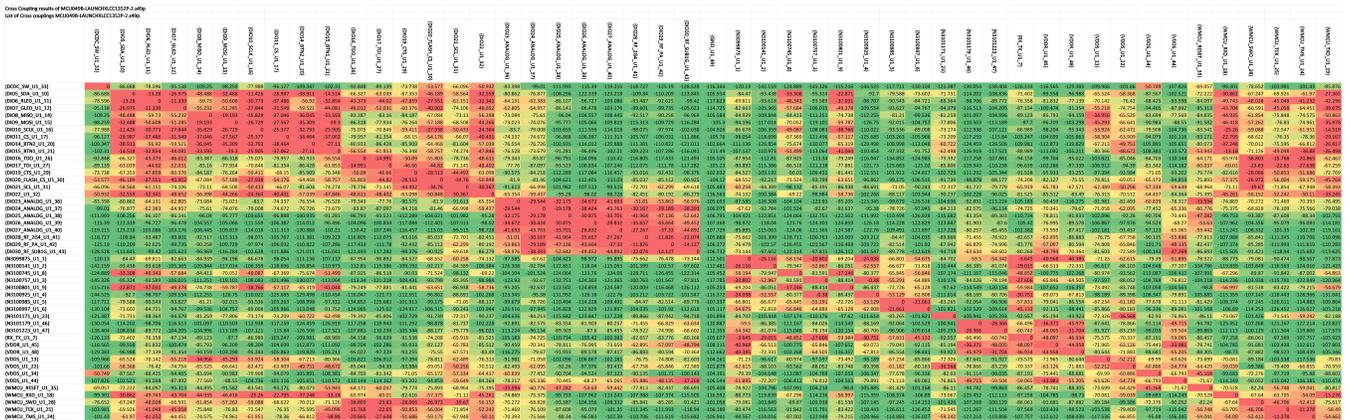


Figure 13. Full Cross Coupling Matrix of a 51 x 51 Node Analysis (red, yellow and green indicate areas of varying degree of concern)

By hiding all the nets not relevant for the 32 KHz XTAL, a much reduced matrix is found.

Cross Coupling results of LAUNCHXL-CC1312R1.s48p	
List of Cross couplings LAUNCHXL-CC1312R1.s48p	
{DIO21_U1_31}	-87.463 -74.042
{DIO22_U1_32}	-94.376 -85.358
{N1014101_U1_4}	0 -23.132
{N1014113_U1_5}	-23.132 0
{N2070550_U1_23}	-124.364 -121.209
{N2071733_U1_1}	-69.262 -77.158
{N2072219_U1_2}	-64.873 -73.677
{N2074535_U1_3}	-40.015 -49.712
{N2239551_U1_47}	-104.217 -101.838
{N2239563_U1_46}	-98.638 -99.588

Figure 14. Cross Coupling Analysis Focused on the 32 KHz XTAL Only

From the cross coupling analysis, it is found that only 2 nets have a cross coupling in the 40dB range: DIO1 and the RF ports. This makes sense as they are right next to each other as part of the design. If the layout had been more complex, it could have found a path that was unintentional.

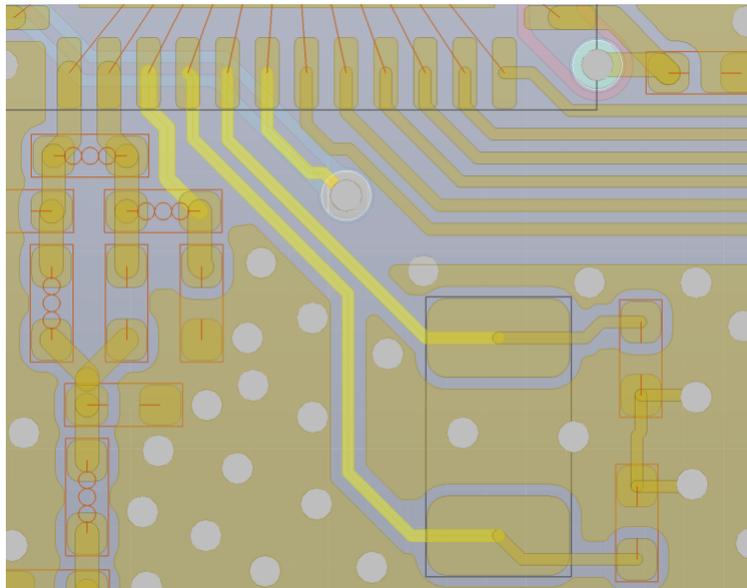


Figure 15. Focused View of Just the 32 KHz XTAL and RF Section to Identify the Nets in the Cross Coupling Matrix

This method must then be repeated for all possible combinations of victims and aggressors.

NOTE: For this specific analysis, the 2½D solver could in some circumstances provide an incorrect results and it is sometimes required to perform this type of analysis using full 3D solvers like HFSS. From experience with SiWAVE, 2½D solver has shown very good results for planar PCB type structures. The only issue that was found is if a product enclosure has a significant impact on the performance, this type of tool will fall short and the full 3D solvers become required.

3.5 Frequency Response of RF Traces

The final part of the five step process is to validate that the RF output path has been done correctly and it functions as it should. With increased frequency, the layout of the passive components used in the balun, RF matching and filtering section becomes increasingly important. It is a good idea to perform such a simulation to identify issues before producing the first PCB's.

In this analysis, focus is on the balun and the filtering section. For the balun the critical parameters are the frequency range, phase and amplitude error. For the filter section, the key parameter is the loss estimate in the pass band and the attenuation and frequency of the stop band.

4 Balun Performance Summary

The main task of a balun is to transform the RF signal from differential to single-ended or in reverse from single-ended to differential. With a total of five measurements, it is possible to summarize the most important the performance specifications of any balun. The five measurements are:

- Insertion loss
- Single-ended return loss
- Differential return loss
- Amplitude balance of the differential port
- Phase balance of the differential port

For this analysis, assume that the single ended port is called P1 and the two differential ports are called P2 and P3. The return losses for each of the ports is going to be called S11, S22, and so forth. To plot the performance of a balun, some arithmetic is required on the S-parameters to extract the wanted information.

- Insertion loss

$$IL = 20\log_{10} \left| S_{12}^2 + S_{13}^2 \right| \tag{1}$$

- Single-ended return loss: S₁₁
- Differential return loss

$$SD_{22} = 20\log_{10} \left| \frac{1}{2} (S_{22}^2 + S_{33}^2) - \frac{1}{2} (S_{23}^2 + S_{32}^2) \right| \tag{2}$$

- Amplitude balance of the differential port

$$AB = 20\log_{10} \left| \frac{S_{31}}{S_{21}} \right| \tag{3}$$

- Phase balance of the differential port

$$PB = \text{ang} \left| \frac{S_{31}}{S_{21}} \right| \tag{4}$$

When Equation 1 through Equation 4 are used to plot the information extracted from the simulation tool it is possible to estimate the performance of the balun.

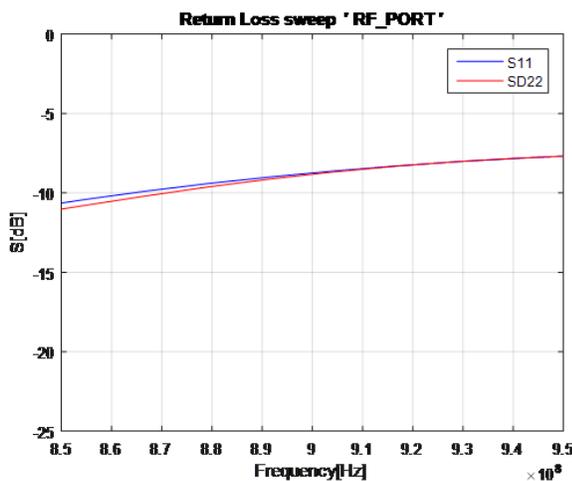


Figure 16. Single Ended and Differential Return Loss is Plotted for the Discrete Balun Designed on the CC1312R LaunchPAD

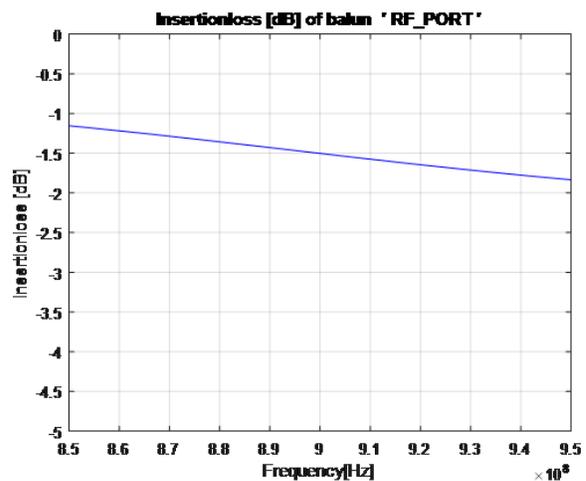


Figure 17. Insertion Loss, Phase and Amplitude Balance is Plotted for the Discrete Balun Integrated on the CC1312R LaunchPAD

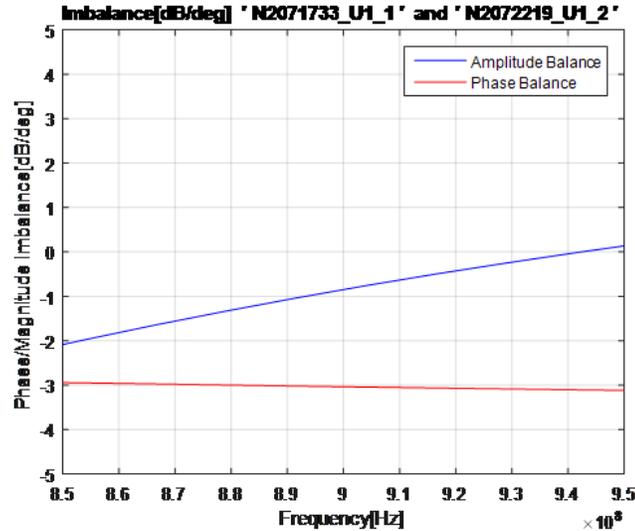


Figure 18. Phase and Amplitude Imbalance is Plotted for the Discrete Balun Integrated on the CC1312R LaunchPAD

From the results in Figure 11, it is found that the balun has a amplitude balance mismatch of 0.4dB and a phase balance error of -4.1° along with an in band insertion loss of 1.5dB.

Generally, the performance of a baluns phase and amplitude balun is combined into a single value called “Common Mode Rejection Ratio” or CMRR.

$$CMRR = \frac{S_{12} + S_{13}}{S_{12} - S_{13}} \tag{5}$$

This is plotted for reasonable values in a graph like Figure 19.

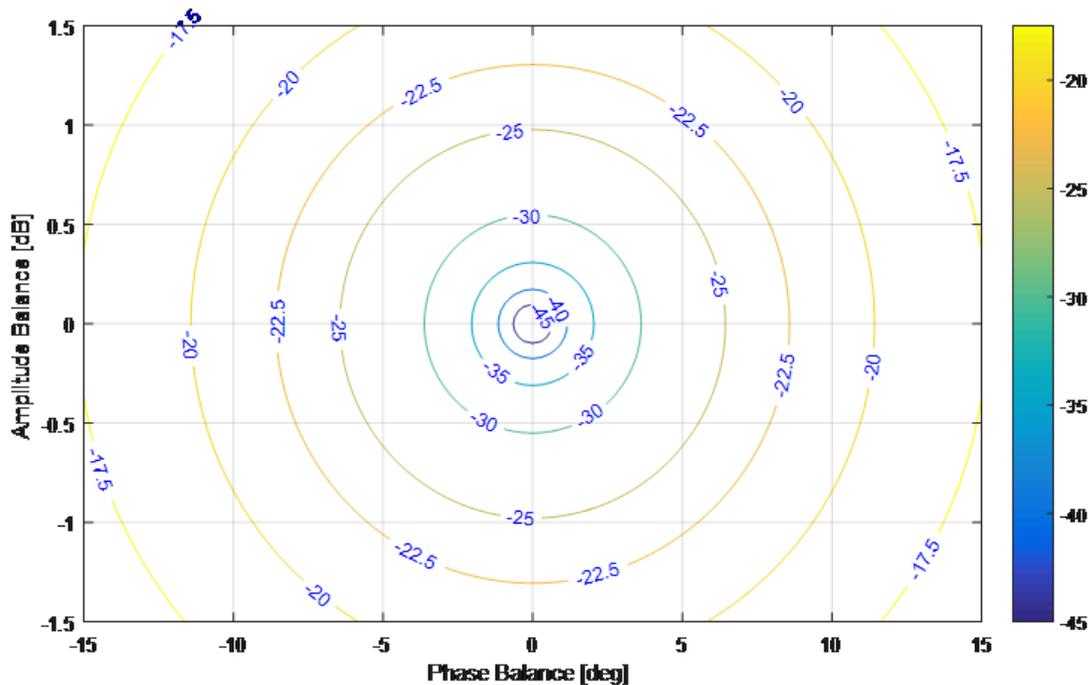


Figure 19. Theoretical CMRR of a Balun

If a balun has a CMRR of better than -25dB, then it is generally considered a good balun for low power wireless type devices. This places a requirement on the design to exhibit an amplitude imbalance of less than $\pm 1.0\text{dB}$ and phase imbalance of less than $\pm 8.0^\circ$.

5 RF Filter Performance Summary

To evaluate the performance of the discrete filter that has been deployed for the CC1312R1, extend the view to include second and third order frequencies.

Therefore, the same equation can be used as used for the balun insertion loss.

- Insertion loss or filter attenuation

$$IL = 20 \log_{10} |S_{12}^2 + S_{13}^2| \tag{6}$$

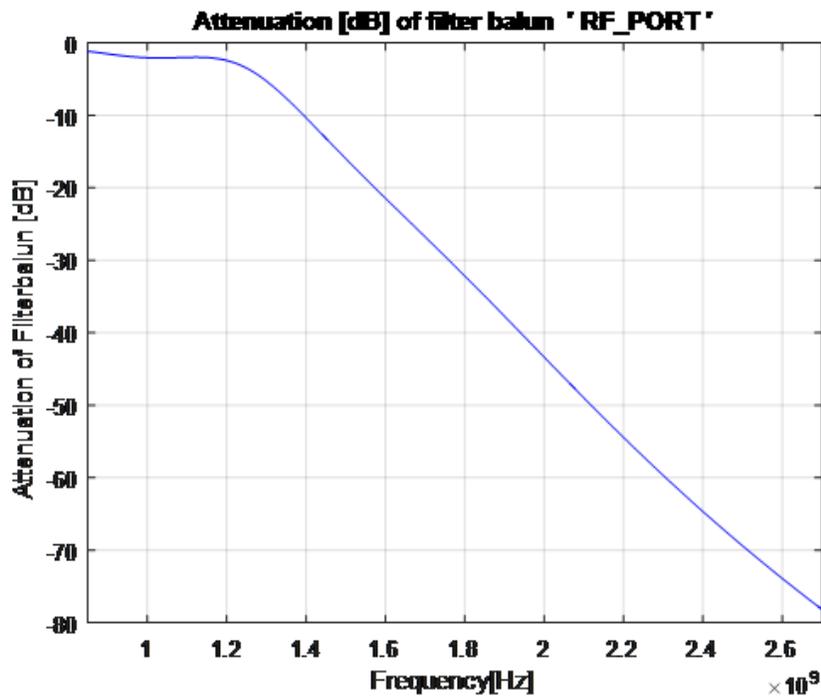


Figure 20. Attenuation of the Filter Portion of a Filter Balun at the 2nd and 3rd Harmonic

From the analysis, it is found that the filtering function of the RF network perform very well, it has about 32dB of attenuation at the second harmonic and 78dB at the 3rd harmonic.

6 Optional PCB Antenna Performance Using 3D FEM Tool

Using a full 3D solver, it is possible to simulate the antenna performance, radiation pattern, frequency range of a final product. Furthermore, if a 3D model of the plastic enclosure is available, it is possible to predicting the influence of plastic enclosures to the final products antenna performance.

The reason this requires a full 3D solver is that antenna radiation pattern are no planar in nature and, in many cases, the enclosures have a significant effect on the antenna performance and these enclosures have no planar features.

For antenna performance, the most important performance parameters to investigate are the antenna match, bandwidth. Then, the next parameters are radiation efficiency and radiation pattern.

The antenna match is simply to have the simulation tool plot the return loss over the band of interest plus some margin. The radiation efficiency is simply a measure of how much power comes into the antenna versus how much power successfully gets radiated.

Finally, it is a good idea to estimate the enclosure and run a simulation of the antenna inside the final enclosure, sometimes this has large impact that might cause a slight re-tuning of the antenna to a new frequency. Overall performance of the antenna design, including the ground plane is estimated best by evaluating the radiation efficiency (found to be 94.4% in [Figure 21](#)).

Antenna Parameters:

Quantity	Freq	Value
Max U	0.85GHz	100.82 mW/sr
Peak Directivity		1.4057
Peak Gain		1.327
Peak Realized Gain		1.267
Peak System Gain		1.267
Radiated Power		901.31 mW
Accepted Power		954.77 mW
Incident Power		1 W
System Power		1 W
Radiation Efficiency		0.944

Figure 21. Overall Performance of the Antenna Design

The antenna pattern for a typical monopole antenna should be circular shaped, the less circular the greater the risk of having blind spots in your end system, as shown in [Figure 22](#).

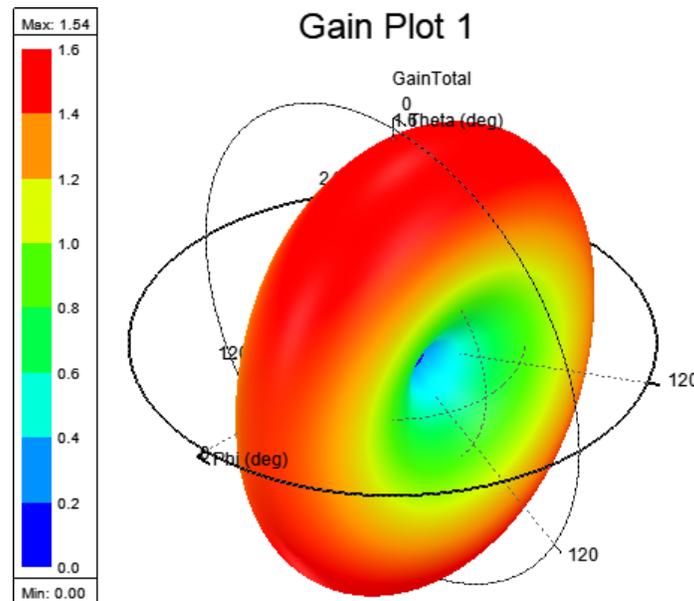


Figure 22. Antenna Pattern for a Typical Monopole Antenna Should be Circular Shaped

The antenna return loss is simply a measurement of how well the antenna is matched and over what frequency range it remains well matched (it is better than 10dB over 100 MHz of bandwidth in [Figure 23](#)).

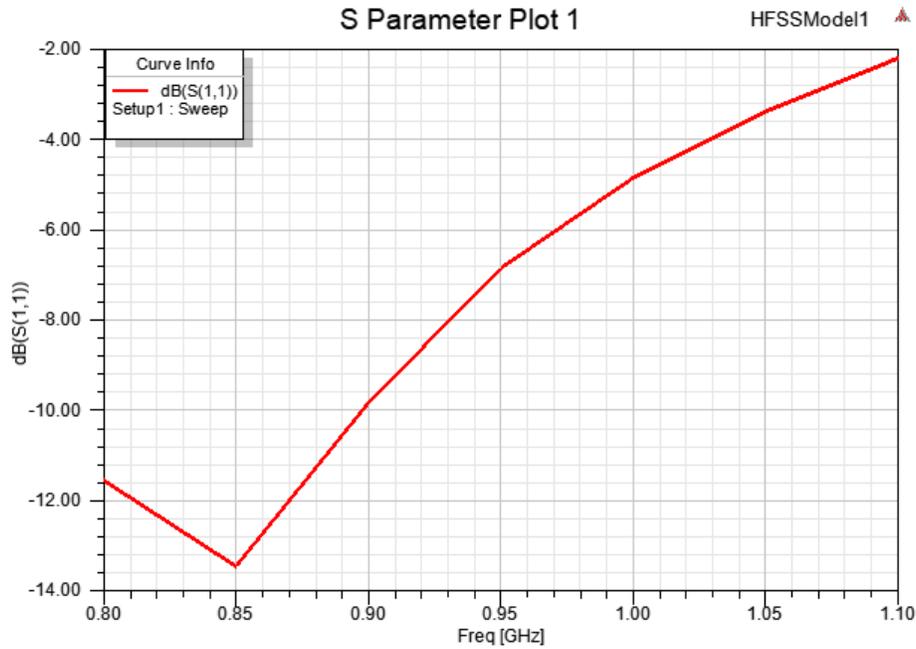


Figure 23. Antenna Return Loss

The ground plane had to be significant simplified for the full 3D solver to have enough RAM to execute, as shown in [Figure 24](#).

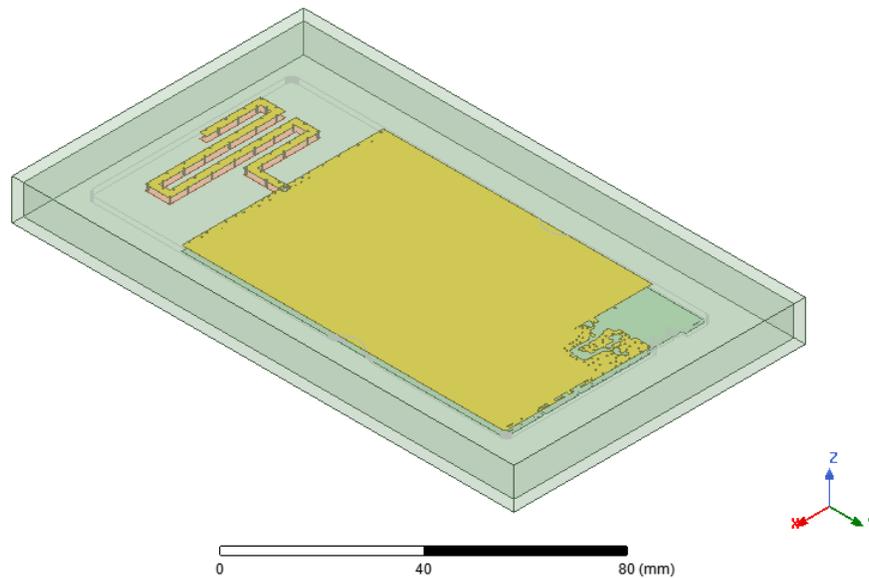


Figure 24. Screen Capture of Simulation Setup

This is a guide on how start running the simulation on RF enabled PCB's. RF is mentioned, but these same techniques apply also for many mixed signal applications.

7 Matlab Program Developed to Read s-parameter Files and Analyze Performance

This Matlab script is capable of reading the Touchstone formatted s-parameter files that Ansys products produce and, then, create return losses, impedance and cross coupling plots.

As a secondary output, the script also produces a comma separated (.csv) file that can be used in Excel to generate the cross coupling matrixes shown in [Figure 13](#) and [Figure 14](#).

```
% This little matlab script takes s-parameters as input and plot all input
% impedance and cross coupling plot and matrix output for the PCB analysis.
% The script is based on getting port information in the "information
% field" of the s-paramter file.
%
% From the matlab prompt run the script and a pop-up will ask you to select
% the s-paramter file for analysis.
%
% OPTIONS:
% PrintFigures = 1; (0: display only, 1: save each graph)
% PrintFigureOption = '-dtiff'; (format to save the graphs in)
% Impedance_analysis = 1; (1: execute analysis, 0: skip analysis)
% CrossCoupling_analysis = 0; (1: execute analysis, 0: skip analysis)
%
% HISTORY
% Updated 12/04/2007 with case insensitive search of frequency multiplier
% in the file
% Updated 11/14/2018 to treat "_" correctly in graphing functions.
% Updated 11/26/2018 added balun performance plotting functionality

clear all;
close all;

% Print options
PrintFigures = 0;
PrintFigureOption = '-dtiff';

% Analysis options (0 = do not run, 1 = run analysis)
Impedance_analysis = 1;           % default enabled, disable to save processing time
CrossCoupling_analysis = 1;      % default enabled, disable to save processing time
Balun_analysis = 1;             % default enabled, disable to save processing time
Balun_freq_list = [850e6, 900e6, 950e6]; % frequency list for balun analysis
Balun_port_list = [37, 32, 33]; % list of port used balun analysis

% Get filename of S-parameter file to analyse
[file_name,file_path] = uigetfile( {'*.s*', 'All snp files (*.s*)';...
    '*.*', 'All Files (*.*)'}, ...
    'Please Select a S-parameter file ',...
    'MultiSelect', 'off' );
file_ptr = fopen(strcat(file_path,file_name),'r');
cd(file_path);

% Pre-define the maximum array size
P_Names = cell(200,1);
f_list = cell(1601,1);
SxP = cell(200,100,1601);
Port_Count = 0;

% Initialize variable with default settings, these will be overwritten
data_format = 1;
system_impedance = 50;
freq_multiplier = 1;
f_point = 1;
end_of_file = 0;
progress_indicator = 1;

fprintf('\nReading. ');
% Run in a loop until the end of the file has been reached.
while ~end_of_file
```

```

t_line = fgets(file_ptr);
% check for end of file instance
if(t_line ~= -1)
    read_switch = sscanf(t_line,'%s', 1);
    switch read_switch
        case ''
            %% Skip these lines
        case '! '
            %% Search for port information from Ansoft tools
            [PC,COUNT,ERRMSG,Next_Index] = sscanf(t_line,' ! Port[%d] =',1);
            if(COUNT == 1)
                [PortName,COUNT,ERRMSG,Next_Index1] =
                sscanf(t_line(Next_Index:length(t_line)),'%s',1);
                Port_Count = Port_Count + 1;
                P_Names(Port_Count) = {PortName};
            end
        case '# '
            % decode the data format line using the following typical
            % format # Hz S MA R 50.000000
            [s11,COUNT,ERRMSG,Next_Index] = sscanf(t_line,' # %s',1);
            [s2,COUNT,ERRMSG,Next_Index1] = sscanf(t_line(Next_Index:length(t_line)),' S
            %s',1);

            [f1,COUNT,ERRMSG,Next_Index] =
            sscanf(t_line((Next_Index+Next_Index1):length(t_line)),' R %f',1);
            % lookup frequency array
            frq_array = {'hz', 'khz', 'mhz', 'ghz', 'thz'};
            % perform a case in-sensitive search
            frq_cmp_rst = strcmp(lower(frq_array),lower(s11));
            for ii=1:length(frq_array)
                freq_table(ii) = 1000^(ii-1)*frq_cmp_rst(ii);
            end
            freq_multiplier = sum(freq_table);
            % lookup frequency array
            format_array = {'RI', 'MA', 'DB'};
            for ii=1:length(format_array)
                if (strcmp(format_array(ii), s2) == 1 )
                    data_format = ii;
                end
            end
            % lookup frequency array
            system_impedance = f1;
        otherwise
            % Read the actual S-parameters from the data file.
            % Touchstone format is not always well formed, so this code
            % is trying to be flexible enough to handle different
            % topologies of file. It handles Ansoft files well, but we
            % have not proven any other sources yet (like equipment)
            not_end_of_dataset = 1;
            data_pointer = 0;
            f_point = 1;
            while not_end_of_dataset
                SxP_data = sscanf(t_line,'%f');
                if(data_pointer==0 && length(SxP_data)>1)
                    %% then it is frequency data %%
                    f_list(f_point) = {SxP_data(1)};
                    data_pointer = data_pointer + 1;
                    SxP_data = SxP_data(2:length(SxP_data));
                end
                for ii=1:2:length(SxP_data)
                    %% S-Parameter data %
                    P_Current = data_pointer - 1;
                    P_X = mod(P_Current,Port_Count)+1;
                    P_Y = floor(P_Current/Port_Count)+1;
                    complex(SxP_data(ii),SxP_data(ii+1));
                    % Touchstone supports 3 different formats
                    % MA is Magnitude and Angle format
                end
            end
        end
    end
end

```

```

        % RI is REAL and IMAG data format
        % DB is DB and angle format
        switch(data_format)
            case 1,
                R = SxP_data(ii);
                I = SxP_data(ii + 1);
                C_SP = complex(R,I);
            case 2,
                C_SP = mag2cpx(SxP_data(ii), SxP_data(ii+1));
            case 3,
                C_SP = db2cpx(SxP_data(ii),SxP_data(ii+1));
        end
        SxP(P_X,P_Y,f_point) = {C_SP};
        data_pointer = data_pointer + 1;
    end
    t_line = fgets(file_ptr);
    if t_line == -1
        not_end_of_dataset = 0;
    end
    if (data_pointer == (Port_Count * Port_Count)+1)
        data_pointer = 0;
        f_point = f_point + 1;
    end
end
% switch read_switch %
progress_indicator = progress_indicator + 1;
if(progress_indicator == 25)
    fprintf('.');
    progress_indicator = 1;
end
end
else
    end_of_file = 1;
end
%if(t_line ~= -1)%
end
fprintf('Done.\n');
fprintf('Data set contains %d ports and %d datapoints\n', Port_Count, f_point);
% Reallocate the arrays to the final size.
f_point = f_point - 1;
P_Names = P_Names(1:Port_Count);
P_Names_2 = P_Names(1:Port_Count);
f_list = f_list(1:f_point);
SxP = SxP(1:Port_Count,1:Port_Count,1:f_point);

% Relable the netnames that have an underscore '_' as this is a matlab keyword
% they will be changed to '\_' such they print out correctly with an
% underscore.
for ii = 1:length(P_Names)
    [P_Name_b P_Name_s] = cell2str(P_Names(ii));
    [f_value, f_index] = find(P_Name_s == '_');
    for jj = 1:length(f_index)
        ll = 1;
        P_Name_fixed = ' ';
        for kk = 1:length(P_Name_s)
            if(find(kk == f_index))
                P_Name_fixed(ll) = '\_';
                P_Name_fixed(ll+1) = P_Name_s(kk);
                ll = ll + 2;
            else
                P_Name_fixed(ll) = P_Name_s(kk);
                ll = ll + 1;
            end
        end
    end
end
P_Name_fixed(1) = ' ';

```

```

P_Name_fixed(length(P_Name_fixed)) = ' ';
if(length(f_index) > 0)
    P_Names_2(ii) = {P_Name_fixed};
else
    P_Names_2(ii) = P_Names(ii);
end
end

%% Input impedance plotter
f_find = [1e8,1e9,2e9];
if (Impedance_analysis == 1)
    fm_list = cell2mat(f_list)*freq_multiplier;
    ss11 = zeros(1,f_point);
    for pp = 1:Port_Count
        for ii = 1:f_point
            s11 = cell2mat(SxP(pp,pp,ii));
            ss11(ii) = s11;
            zz1(ii) = 50 * (((1-s11)^-1) * (s11+1));
        end
        fg_hnd = figure(2*pp-1);
        plot(fm_list,20*log10(abs(ss11)));
        xlabel('Frequency[Hz]');
        ylabel('S[dB]');
        ss = sprintf('Return Loss sweep %s', cell2str(P_Names_2(pp)));
        title(ss);
        %axis([start_fq stop_fq -105 -55]);
        grid on;
        fg_name = sprintf('Return Loss sweep %s',cell2str(P_Names(pp)));
        set(fg_hnd,'numbertitle','off','name',fg_name, 'units','normalized','resize','on');

        fg_hnd = figure(2*pp);
        plot(fm_list,abs(zz1));
        xlabel('Frequency[Hz]');
        ylabel('Z[magnitude]');
        ss = sprintf('Input impedance of %s', cell2str(P_Names_2(pp)));
        title(ss);
        %axis([start_fq stop_fq -105 -55]);
        grid on;
        fg_name = sprintf('Input impedance of %s',cell2str(P_Names(pp)));
        set(fg_hnd,'numbertitle','off','name',fg_name, 'units','normalized','resize','on');
        for ii = 1:length(f_find)
            found = find(fm_list > f_find(ii));
            if(found>0)
                Z_found(pp,ii) = abs(zz1(found(1)));
                Z_found_fm_list(pp,ii) = fm_list(found(1));
            else
                Z_found(pp,ii) = abs(zz1(length(zz1)));
                Z_found_fm_list(pp,ii) = fm_list(length(zz1));
            end
        end
    end
    if(PrintFigures == 1)
        orient(fg_hnd, 'portrait');
        print(fg_hnd, PrintFigureOption, fg_name);
    end
end

end
Z_found
Z_found_fm_list
%% Save information to text file
out_file_name = strcat(strtok(file_name, '.'), '_Impedance.txt');
out_file_ptr = fopen(strcat(file_path,out_file_name), 'w');
fprintf(out_file_ptr, 'Input impedances %s\n', file_name);
fprintf(out_file_ptr, '\t');
for pp_o = 1:length(f_find)
    fprintf(out_file_ptr, '%2.3f\t', Z_found_fm_list(1,pp_o));
end
end

```

```

    fprintf(out_file_ptr, '\n');
    for pp_i = 1:Port_Count
        fprintf(out_file_ptr, '%s\t', cell2str(P_Names(pp_i)));
        for pp_o = 1:length(f_find)
            fprintf(out_file_ptr, '%2.3f\t', Z_found(pp_i, pp_o));
        end
        fprintf(out_file_ptr, '\n');
    end
    fclose(out_file_ptr)
end

%% Cross coupling plotter
f_find = [9e8, 1e9];
if (CrossCoupling_analysis == 1)
    fm_list = cell2mat(f_list)*freq_multiplier;
    s11 = zeros(1, f_point);
    for pp_i = 1:Port_Count
        for pp_o = 1:Port_Count
            for ii = 1:f_point
                s11 = cell2mat(SxP(pp_i, pp_o, ii));
                s12(ii) = s11;
            end
            if(pp_i ~= pp_o)
                if(Impedance_analysis==1)
                    fg_hnd = figure(pp_i+2*Port_Count);
                else
                    fg_hnd = figure(pp_i);
                end
                hold on;
                plot(fm_list, 20*log10(abs(s12)));
                xlabel('Frequency[Hz]');
                ylabel('S[dB]');
                ss = sprintf('Isolation between nodes %s', cell2str(P_Names_2(pp_i)));
                title(ss);
                grid on;
                fg_name = sprintf('Isolation between nodes %s', cell2str(P_Names(pp_i)));
                set(fg_hnd, 'numbertitle', 'off', 'name', fg_name,
                    'units', 'normalized', 'resize', 'on');
                start_search = find(fm_list > f_find(1));
                if(start_search>0)
                    start_search = start_search(1);
                else
                    start_search = 1;
                end
                stop_search = find(fm_list > f_find(2));
                if(stop_search>0)
                    stop_search = stop_search(1);
                else
                    stop_search = f_point;
                end
                [M, I] = max(abs(s12(start_search:stop_search)));
                XC(pp_o, pp_i) = 20*log10(M);
                XC_index(pp_o, pp_i) = fm_list(start_search + I - 1);
            end
        end
        if(PrintFigures == 1)
            axis([0 5e9 -5 0]);
            orient(fg_hnd, 'portrait');
            print(fg_hnd, PrintFigureOption, fg_name);
        end
    end
    %% Save information to text file
    out_file_name = strcat(strtok(file_name, '.'), '_XC.txt');
    out_file_ptr = fopen(strcat(file_path, out_file_name), 'w');
    fprintf(out_file_ptr, 'Cross Coupling results of %s\n', file_name);
    fprintf(out_file_ptr, 'List of Cross couplings %s\n', file_name);

```

```

fprintf(out_file_ptr,'\t');
for pp_o = 1:Port_Count
    fprintf(out_file_ptr,'%s\t',cell2str(P_Names(pp_o)));
end
fprintf(out_file_ptr,'\n');
for pp_i = 1:Port_Count
    fprintf(out_file_ptr,'%s\t',cell2str(P_Names(pp_i)));
    for pp_o = 1:Port_Count
        fprintf(out_file_ptr,'%2.3f\t',XC(pp_i,pp_o));
    end
    fprintf(out_file_ptr,'\n');
end

fprintf(out_file_ptr,'\n\nList of Worst Case Frequencies %s\n',file_name);
fprintf(out_file_ptr,'\t');
for pp_o = 1:Port_Count
    fprintf(out_file_ptr,'%s\t',cell2str(P_Names(pp_o)));
end
fprintf(out_file_ptr,'\n');
for pp_i = 1:Port_Count
    fprintf(out_file_ptr,'%s\t',cell2str(P_Names(pp_i)));
    for pp_o = 1:Port_Count
        fprintf(out_file_ptr,'%2.3f\t',XC_index(pp_i,pp_o));
    end
    fprintf(out_file_ptr,'\n');
end

fclose(out_file_ptr);
end
fclose(file_ptr);

%% Input balun performance plotter
f_find = Balun_freq_list;
if (Balun_analysis == 1)
    fm_list = cell2mat(f_list)*freq_multiplier;
    ss11 = zeros(1, f_point);
    ss21 = zeros(1, f_point);
    ss31 = zeros(1, f_point);

    ss22 = zeros(1, f_point);
    ss33 = zeros(1, f_point);
    ss23 = zeros(1, f_point);
    ss32 = zeros(1, f_point);

    for ii = 1:f_point
        %% single ended returnloss and imbalance data
        s11 = cell2mat(SxP(Balun_port_list(1),Balun_port_list(1),ii));
        ss11(ii) = s11;
        s21 = cell2mat(SxP(Balun_port_list(2),Balun_port_list(1),ii));
        ss21(ii) = s21;
        s31 = cell2mat(SxP(Balun_port_list(3),Balun_port_list(1),ii));
        ss31(ii) = s31;

        %% differential returnloss only, rarely required
        s22 = cell2mat(SxP(Balun_port_list(2),Balun_port_list(2),ii));
        ss22(ii) = s22;
        s33 = cell2mat(SxP(Balun_port_list(3),Balun_port_list(3),ii));
        ss33(ii) = s33;
        s23 = cell2mat(SxP(Balun_port_list(2),Balun_port_list(3),ii));
        ss23(ii) = s23;
        s32 = cell2mat(SxP(Balun_port_list(3),Balun_port_list(2),ii));
        ss32(ii) = s32;
    end

    end
    fg_hnd = figure;

```

```

plot(fm_list, 20*log10(abs(ss11)), 'b');
hold on;
SD22 = 0.5*(ss22.*ss22 + ss33.*ss33) - 0.5*(ss23.*ss23 + ss32.*ss32);
plot(fm_list, 20*log10(abs(SD22)), 'r');
xlabel('Frequency[Hz]');
ylabel('S[dB]');
ss = sprintf('Return Loss sweep %s', cell2str(P_Names_2(Balun_port_list(1))));
title(ss);
axis([f_find(1) f_find(3) -25 0]);
grid on;
fg_name = sprintf('Return Loss sweep %s', cell2str(P_Names(Balun_port_list(1))));
set(fg_hnd, 'numbertitle', 'off', 'name', fg_name, 'units', 'normalized', 'resize', 'on');
legend('S11', 'SD22');
if(PrintFigures == 1)
    orient(fg_hnd, 'portrait');
    print(fg_hnd, PrintFigureOption, fg_name);
end

fg_hnd = figure;
plot(fm_list, 20*log10(abs(ss31./ss21)), 'b');
hold on;
plot(fm_list, unwrap(angle(ss31./ss21)), 'r');
xlabel('Frequency[Hz]');
ylabel('Phase/Magnitude Imbalance[dB/deg]');
ss = sprintf('Imbalance[dB/deg] %s and %s', cell2str(P_Names_2(Balun_port_list(2))),
cell2str(P_Names_2(Balun_port_list(3))));
title(ss);
axis([f_find(1) f_find(3) -5 5]);
grid on;
legend('Amplitude Balance', 'Phase Balance');
fg_name = sprintf('Imbalance[dB/deg] %s', cell2str(P_Names(Balun_port_list(1))));
set(fg_hnd, 'numbertitle', 'off', 'name', fg_name, 'units', 'normalized', 'resize', 'on');

if(PrintFigures == 1)
    orient(fg_hnd, 'portrait');
    print(fg_hnd, PrintFigureOption, fg_name);
end

fg_hnd = figure;
plot(fm_list, 20*log10(abs(ss31.*ss31 + ss21.*ss21)), 'b');
xlabel('Frequency[Hz]');
ylabel('Insertionloss [dB]');
ss = sprintf('Insertionloss [dB] of balun %s', cell2str(P_Names_2(Balun_port_list(1))));
title(ss);
axis([f_find(1) f_find(3) -5 0]);
grid on;
fg_name = sprintf('Insertion loss [dB] of balun %s', cell2str(P_Names(Balun_port_list(1))));
set(fg_hnd, 'numbertitle', 'off', 'name', fg_name, 'units', 'normalized', 'resize', 'on');

if(PrintFigures == 1)
    orient(fg_hnd, 'portrait');
    print(fg_hnd, PrintFigureOption, fg_name);
end

fg_hnd = figure;
plot(fm_list, 20*log10(abs(ss31.*ss31 + ss21.*ss21)), 'b');
xlabel('Frequency[Hz]');
ylabel('Attenuation of Filterbalun [dB]');
ss = sprintf('Attenuation [dB] of filter balun %s', cell2str(P_Names_2(Balun_port_list(1))));
title(ss);
axis([f_find(1) 3*f_find(2) -80 0]);
grid on;
fg_name = sprintf('Attenuation [dB] of filter balun %s',
cell2str(P_Names(Balun_port_list(1))));
set(fg_hnd, 'numbertitle', 'off', 'name', fg_name, 'units', 'normalized', 'resize', 'on');

```

```
if(PrintFigures == 1)
    orient(fig_hnd, 'portrait');
    print(fig_hnd, PrintFigureOption, fg_name);
end

end
```

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated