Dr. Mühlhaus Consulting & Software GmbH

www.muehlhaus.com
info@muehlhaus.com

Tutorial: Getting Started with RFIC Inductor Toolkit

Table of contents:

Document revised: 24. April 2020
Document revision: 1.5

Introduction

Mühlhaus RFIC InductorToolkit assists you to design "optimum" inductors in RFIC/MMIC technologies. It very quickly makes you an inductor design expert, using a very efficient synthesis approach with EM-accurate results. You only need to know the electrical properties that you want, and Inductor Toolkit will find out the best possible layout for you. As you are getting more experienced in this, you can later customize the tool and settings to your preferences.

This document will provide a quick start with several examples, starting from the most basic geometries.

Installation

Before you can use Inductor Toolkit, you have to install it in ADS as a designkit.
The procedure for that is similar to other ADS PDKs: in the ADS main windows, go to *DesignKits > Unzip Design Kit ...* and install it.

You must install Inductor Toolkit with <u>write access</u> for the users at file level, because users can customize settings and schematics. For example, if you store your ADS workspaces to $HOME/ads2017, you could create a new directory $HOME/designkits and unzip Inductor Toolkit to that location.

Mühlhaus RFIC InductorToolkit can be used stand-alone with no additional foundry PDK. Technology settings like layers are then defined in a library that is part of Inductor Toolkit.
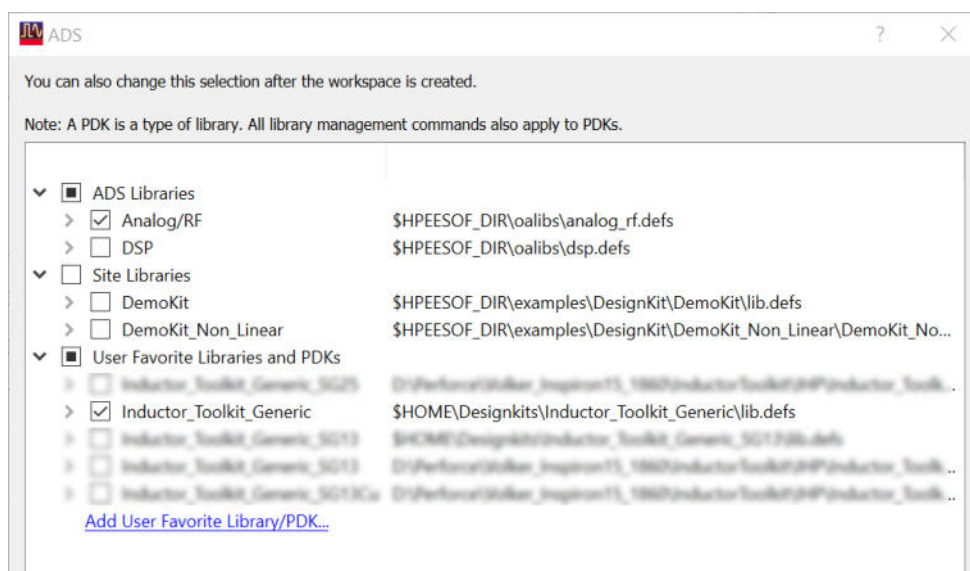
Create a new example workspace

We want to use a new workspace for this tutorial. Start ADS and from the menu, do
File > New Workspace.

Follow the wizard, and set the workspace name to Synthesis_Tutorial_wrk.
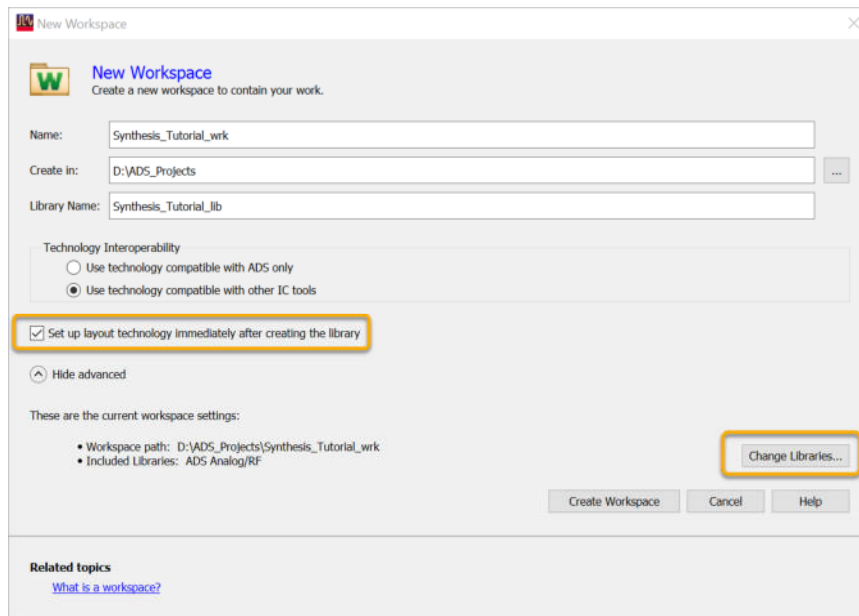
Check the "Setup layout immediately"setting, then click on "Change Libraries..."
Select Inductor Toolkit for your desired technology, the click OK.

**Next, ADS wants you to choose a technology. If this is missing, you forgot to enable
the "Setup layout immediately"setting.**

Select the correct technology for layout. The default Inductor Toolkit technology is usually
listed of top of the list. It is important to select the correct technology here.
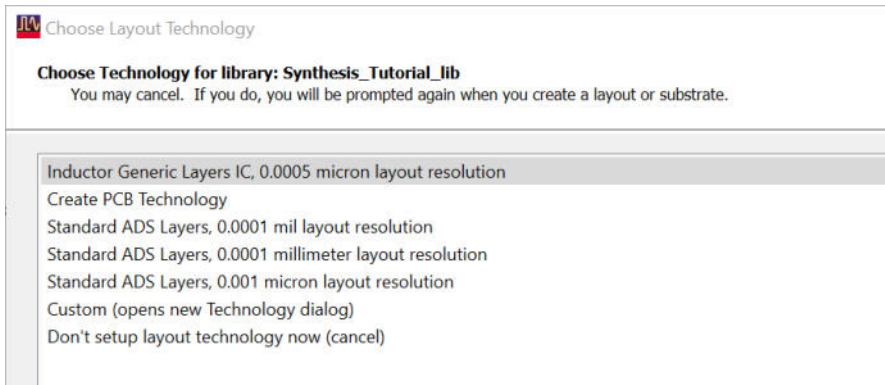
Press the Finish button to complete the workspace setup.
Your workspace is now ready for use, and you should see "Inductor Toolkit" in the ADS main window's menu.

For a better understanding of the library and PDK configuration, we recommend to set the ADS main window to "Library View". Just click on the corresponding tab at the top of the project tree display.
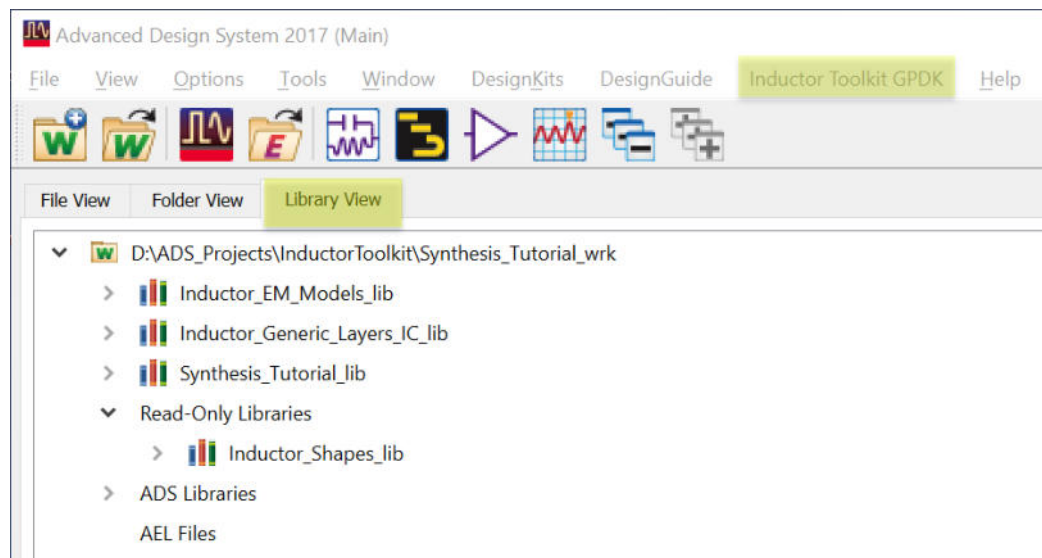
You can now see the structure of the libraries in your workspace:

"Synthesis_Tutorial_lib" was created with the new workspace, and is empty at this point. This is where we will store new inductors synthesized in this tutorial.

"Inductor_EM_Models_lib" is part of Inductor Toolkit, and contains the pre-defined inductor models and schematics for inductor synthesis. These will be used automatically "behind the scenes" when you synthesize new inductors.

"Inductor_Generic_Layers_IC_lib" is the technology library for Inductor Toolkit Generic – you will have a different library name when using an Inductor Toolkit version for IHP, for example.

In addition, there is a read-only library "Inductor_Shapes_lib" that holds to inductor layout definitions.

Design your first inductor

The examples below use the GPDK180 technology. With other technologies, the resulting Q factors and optimum geometries will be different, but don't worry: that is what Inductor Toolkit is about, to find the optimum values for your specs and your technology.

Let's start and design your first inductor: 2.5nH for target frequency 4 GHz.

Synthesis steps are numbered. No surprise, we start with step **1: Set layout option**. In the menu, click on "**Octagon diff.**"
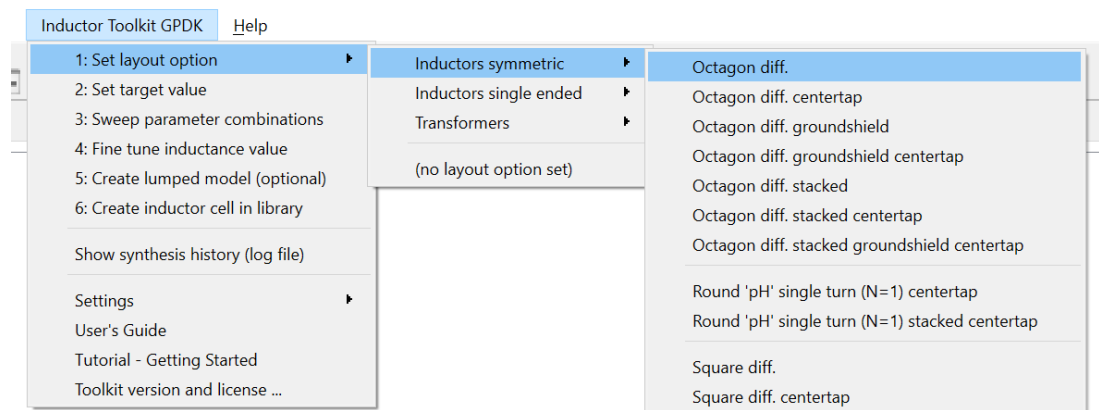

That differential (symmetric) octagon inductor is the standard shape for symmetric inductors when no center tap is needed. The octagon shape offers higher Q factor than the square shape, which is also available (Square diff.)
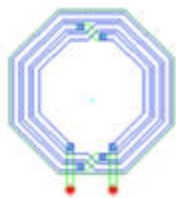
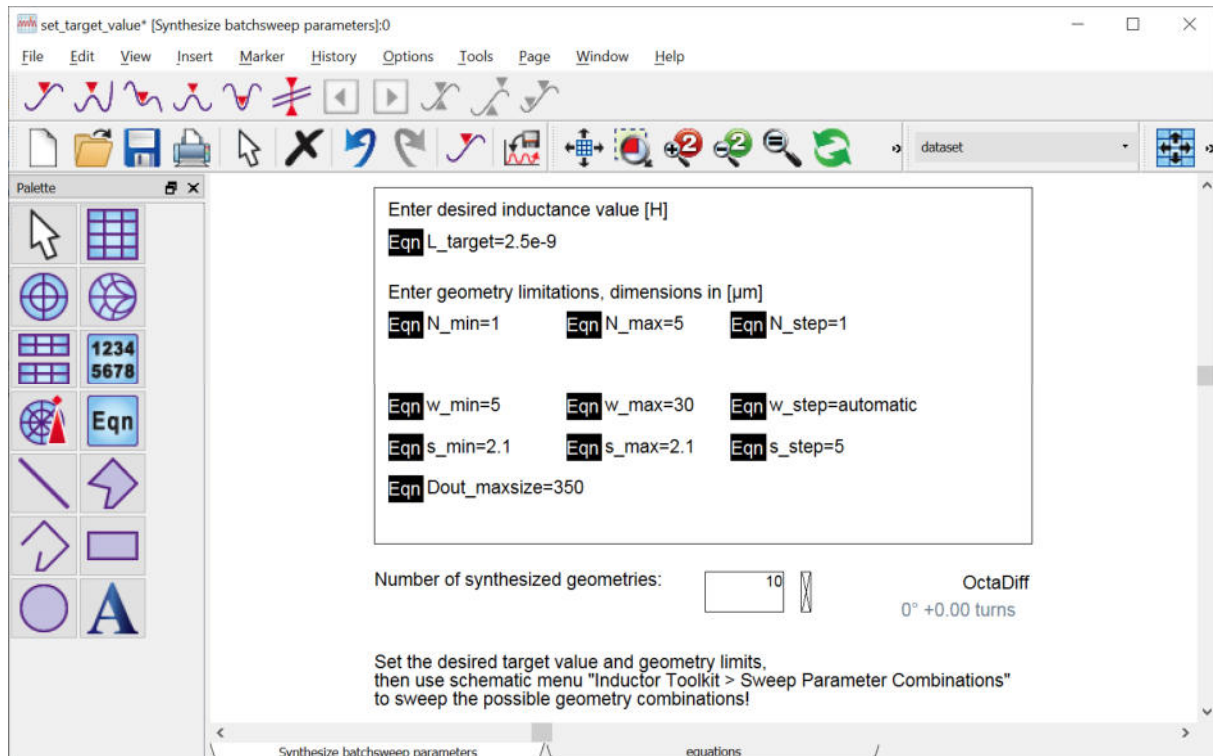The corresponding basic shapes with center tap are also available, for octagon and square geometry.

Besides these basic shapes, there are derived shapes with better performance for special cases: stacked metal versions are useful when the inductor performance is limited by ohmic conductor losses. They have increased metal cross section using multiple layers connected in parallel. This leads to smaller resistance at low frequencies (where skin effect is not yet pushing all the current to the edges) but also causes higher capacitance to the substrate, so it lowers the Q factor at higher frequency. For inductors that are limited by substrate coupling at high frequencies, there are octagons with patterned ground shield. That shield doesn't change the magnetic field, but shields the electric field from the substrate.

In this tutorial, we just go ahead with the "Octagon diff."
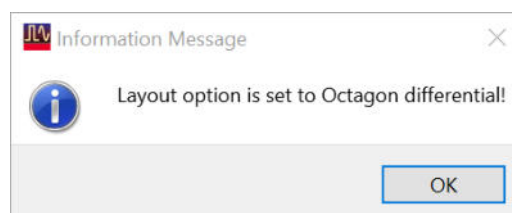
Now that we have set the layout option, go to step **2: Set target value.** This opens a data display form where the target value and geometry limitations are specified.

Change the target value to 2.5e-9 and set the geometry limits as shown below:

This defines the geometry space where Inductor Toolkit will search for the optimum inductor: number of turns from 1 to 5 turns in full turns, conductor width from 5 to 30 µm with automatic step size, spacing between turns fixed to 2.1µm, and inductor outer diameter no larger than 350µm.

The "Number of synthesized geometeries" at the bottom shows how many different layouts are possible with these geometry limitations, for the specified target value. That value is calculated in real time, based on your input.

You might wonder why we don't sweep the spacing between the turns. The reason is that Q factor is not very sensitive to that parameter, and it would be inefficient to always sweep it during our inductor optimization. We usually keep it at some small value, or possibly at the minimum possible value (see foundry layout rules).

In this example, we find 10 different layout candidates for the desired 2.5nH inductor. Other parameter combinations in the sweep range can't be physically built for this target inductance value.

Save the data display, so that we can proceed to the next step.

Inductor Toolkit has calculated the geometry parameters for 9 different inductors that match our specification. We are now ready to EM simulate all of them, to find the inductor with the best possible Q factor. In the Inductor Toolkit menu, click on **3: Sweep parameter combinations**.
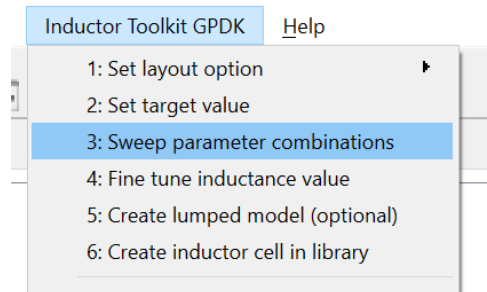

This will bring up a schematic window (batchsim_2port) with Momentum emModels, set up for batch simulation of the previously calculated geometries. The simulation of that schematic is started automatically
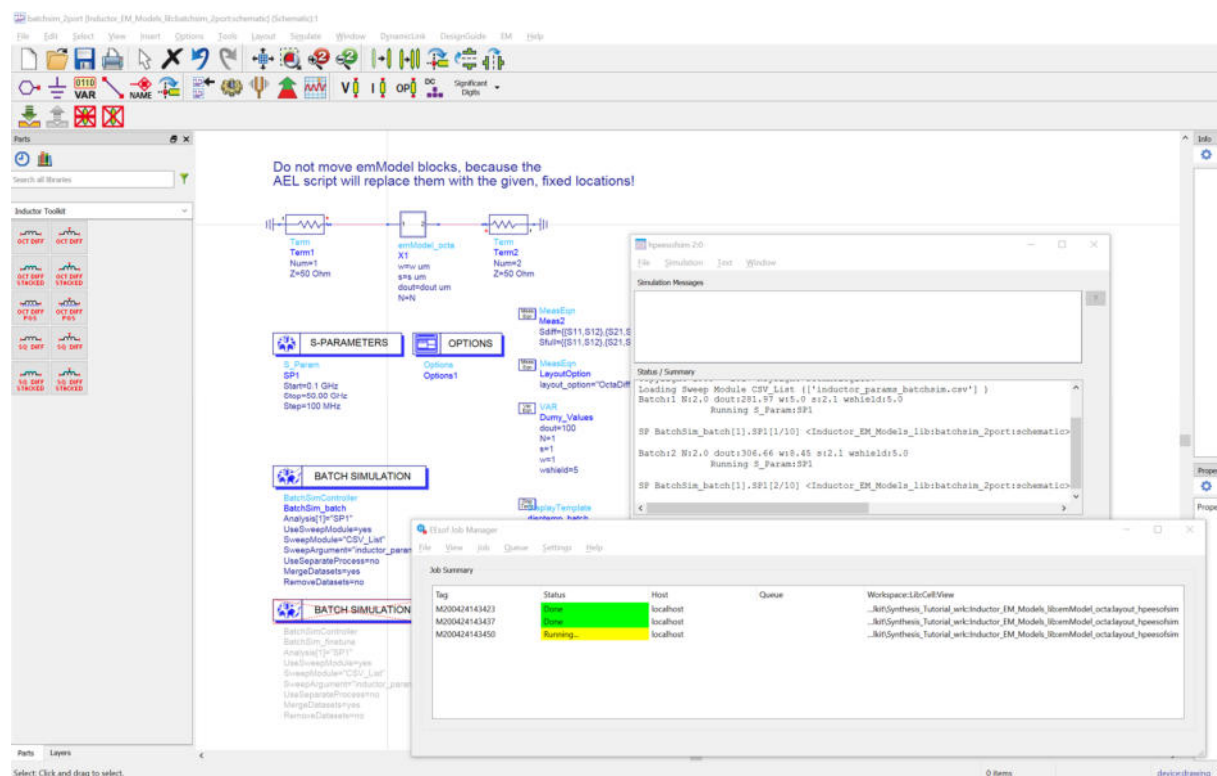

For the very first simulation, there might be a delay because Momentum calculates the substrate functions, before starting the actual inductor simulations.

It takes a few minutes to finish this simulation, because all the layouts are now passed to Momentum for EM simulation. Then, you will see a data display where results are shown and compared in terms in inductor parameters L and Q.

Here is the overview of the result page, which is automatically shown after simulation. The details are discussed and explained below.
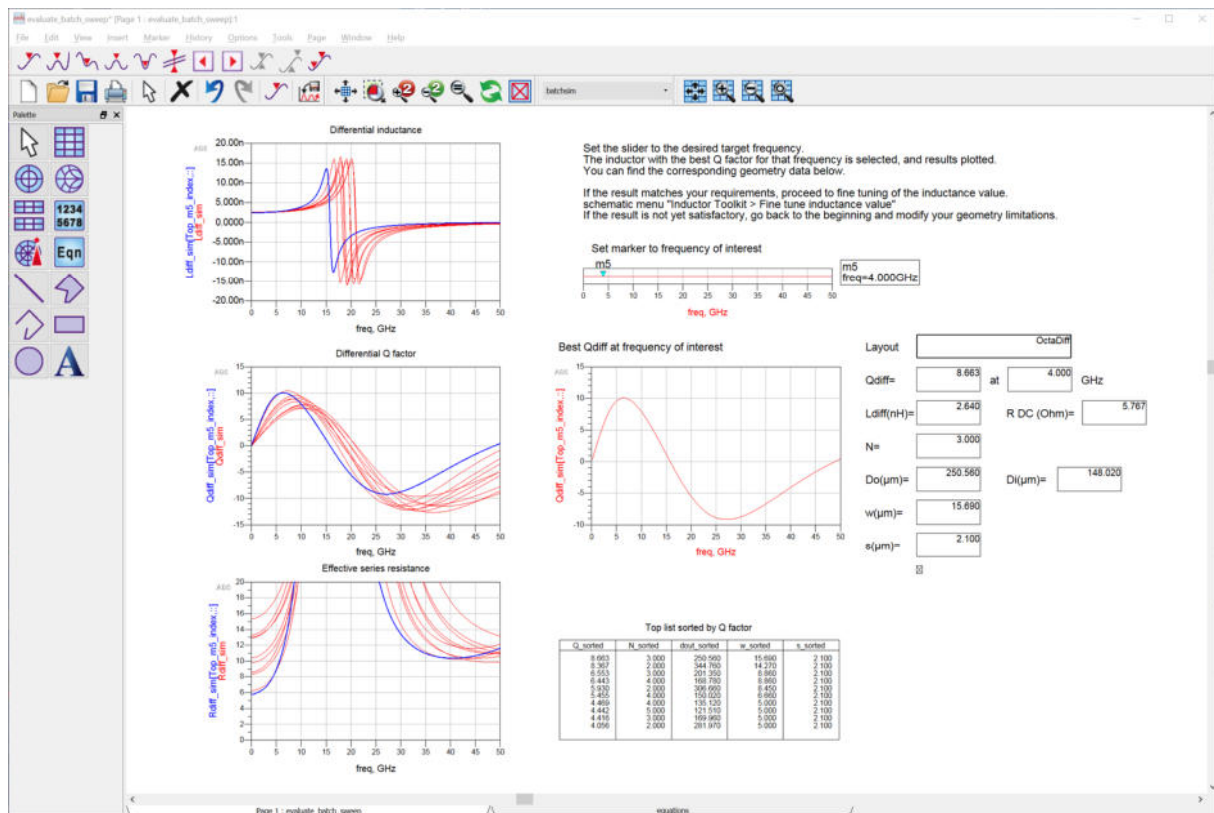
On the top left side, the effective inductance is plotted for the different layouts. This is the differential inductance, as measured differentially between the inductor pins.
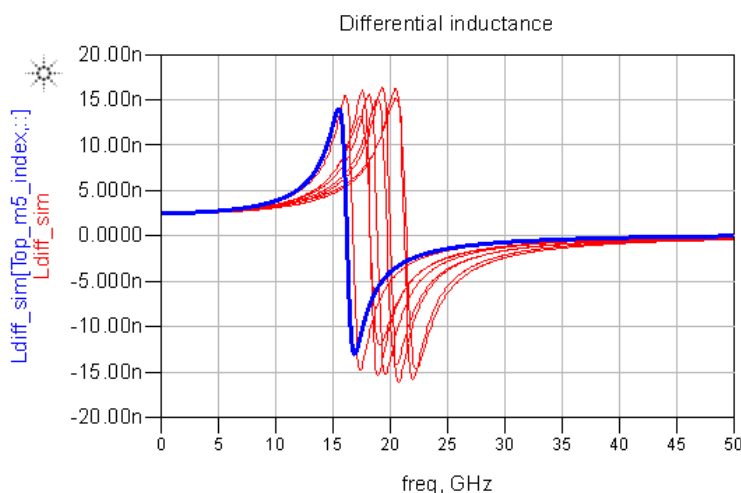

All the inductors have a low frequency inductance that is close to the specified 2.5nH target value. This confirms that the synthesis of geometries in step 2 worked well.

At higher frequencies, the effective inductance changes due to parasitic capacitance that leads to different self resonance frequency (SRF).

Let's look at the Q factor, because that is what we want to optimize.

The maximum Q factor is very different between the layouts, as expected.

One of the Q factor curves is highlighted in blue color. That is the result with the highest Q factor at the frequency of interest, set with the slider on the top right side.
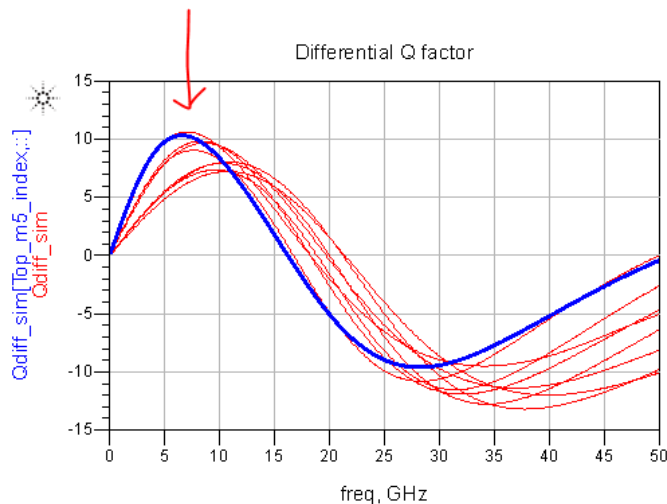
You can change the frequency of interest by moving the marker (hint: to select the marker, click on the readout box right from the slider), or by typing the value into the readout box. Notice how the values for the best inductor change as you change the target frequency.
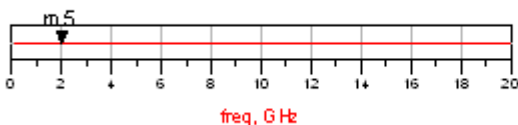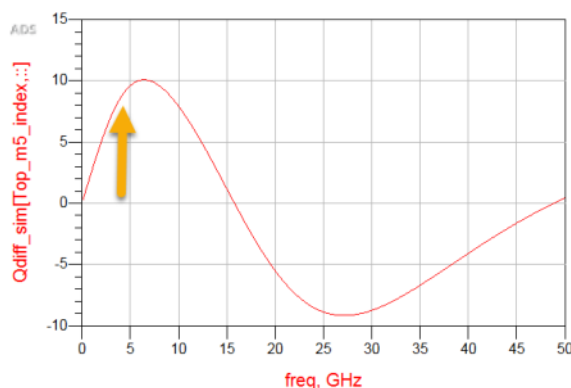

When the frequency is changed to our goal of 4GHz, the highlighted "best" inductor result is still the same. This is because the same layout is best at 2GHz and 4GHz.
The best inductor's parameters are shown on the top right side:

This is the best inductor within this batch simulation, and will be used to derive the best possible inductor layout:

- N=3 turns (range was set to N=1...5 turns)
- Outer diameter Do = 250µm (range was set to 350µm max.)
- Line width w= 15.7µm (range was set to 5µm ... 30µm in steps of 5µm)
- Spacing s= 2.1µm fixed

If we are satisfied with this inductor's performance, we can now proceed to the next step. Otherwise, we have to modify the geometry space, and allow inductors that are physically larger. We will look at that later, in the next tutorial.

So now, we have finished the batch simulation step, and selected the best performing inductor from the list, for our frequency of interest. The next step is **4: Fine tune inductance value**.
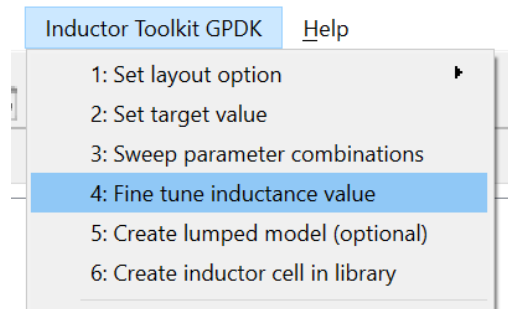
The initial synthesis was based on DC inductance. The workflow of Inductor Toolkit is designed to compensate for these error sources with an additional step, where the inductance is fine tuned by fine tuning the inductor diameter. Based on the best inductors from the batch sweep, this step fine tunes the in ductance by adjusting the inductor diameter. Also, the fine tuning step interpolates the optimum conductor width.

When you click **4: Fine tune inductance value**, this will bring up a simulation schematic that runs one more EM model, with the fine tuned inductor diameter. The dimensions for that tuned layout had been calculated automatically, based on the batch sweep results.

The result of the fine tuning step is the final, optimized inductor. Geometry parameters and electrical parameters are shown in the data display that comes up after simulation:
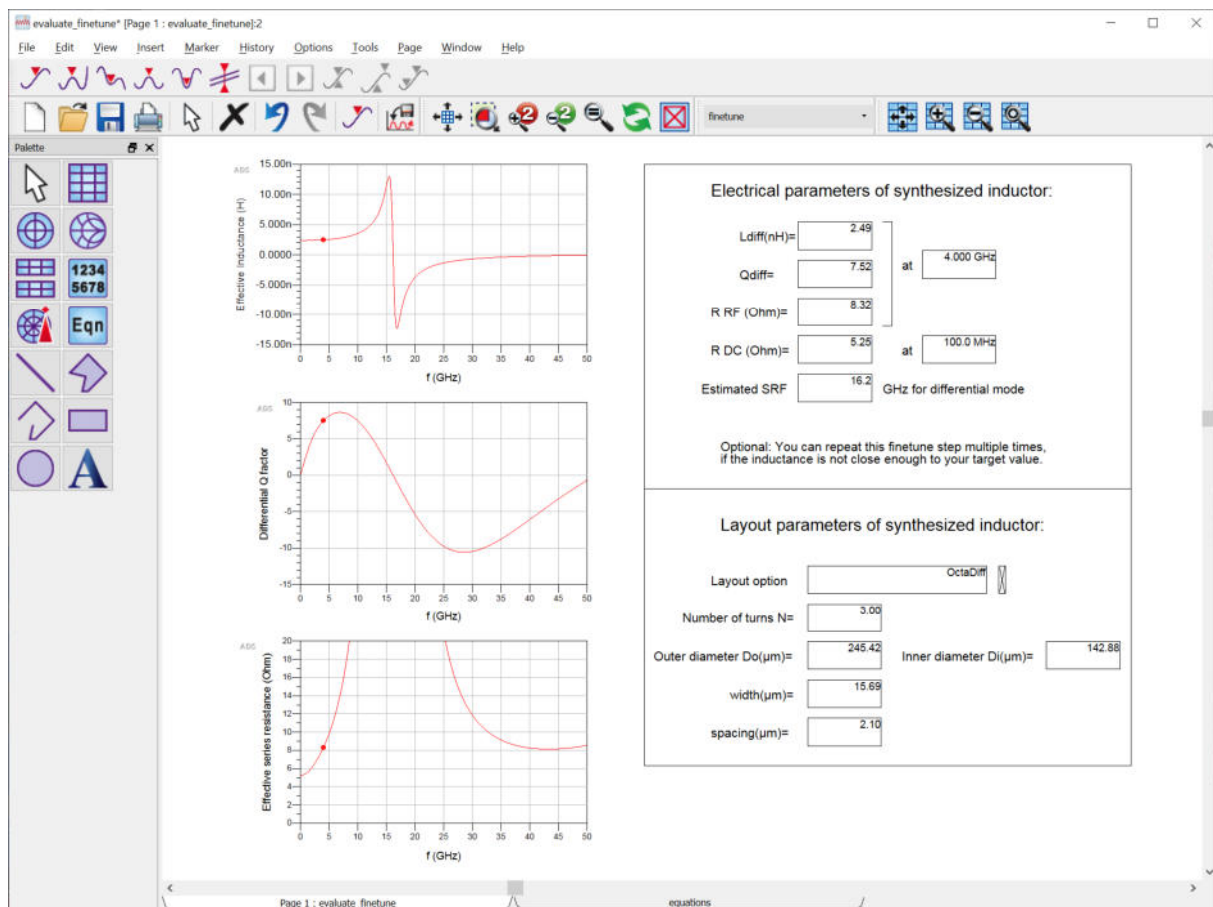
Comparing to the batch sweep simulation, you can see that the conductor width is unchanged at 15.7µm and the outer diameter was tuned to 245µm, to reach the target inductance at the target frequency.

Note on conductor width: The fine tune step can also tweak the conductor width, by interpolation or extrapolation from simulated conductor width values. This is done automatically, if the resulting value is within the geometry limits. In some cases, this might result in inductor diameters that are slightly larger than the specified maximum size.[1]

Iterative fine tune: If the inductance at the taget freqency is not close enough to your specification, you can now repeat the fine tune step, to iterate towards the final layout. In most cases, running the fine tune step once is enough, and there is no need to repeat it.

When the fine tune step in finished, we have two possible ways to proceed:

- Create an **equivalent circuit model**, for use in ADS or other simulators, with output as Spectre *.scs netlist file, or
- Directly proceed to the final step, where an **ADS cell** with layout view and schematic view is created.
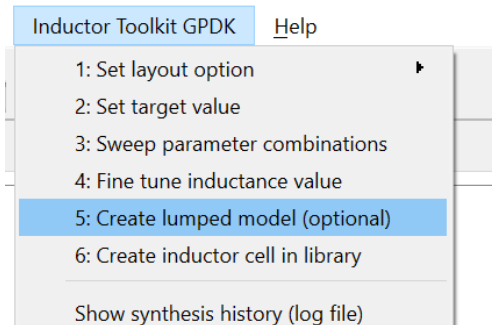
If we skip the equivalent circuit model, the final cell will only have the S-parameters available for schematic level simulation. If we do the equivalent circuit model fit first, the

---

[1] By default, an oversize diameter of up to 105% is tolerated. This can be configured on the equations page of the evaluate_batch_sweep data display.
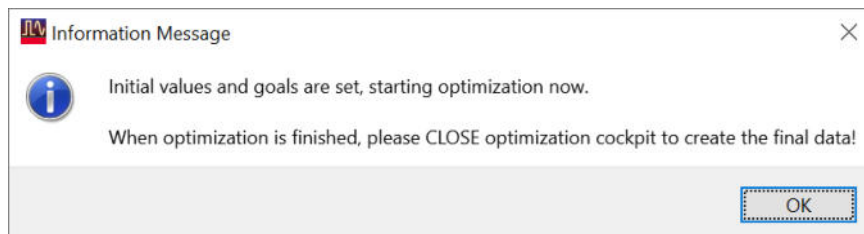
final cell will have two schematic views: one based on the S-parameters, and the other based on the equivalent circuit model.

In this tutorial, we proceed with the optional equivalent circuit model step.
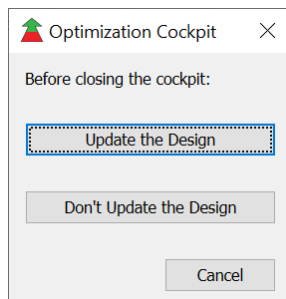
The equivalent circuit model extraction is embedded into the synthesis workflow, and can be done after the fine tune step is finished. The required data is automatically provided by the file tune step.

Inductor Toolkit GPDK	Help
1: Set layout option ▸	
2: Set target value	
3: Sweep parameter combinations	
4: Fine tune inductance value	
5: Create lumped model (optional)	
6: Create inductor cell in library	
Show synthesis history (log file)	

When you click on **5: Create lumped model (optional)**, this will bring up a schematic and automatically start optimization. The schematic is pre-configured with parameters and goals to optimize an equivalent circuit model to your inductor's EM simulation results.

Information Message ✕

Initial values and goals are set, starting optimization now.

When optimization is finished, please CLOSE optimization cockpit to create the final data!

OK

Actually, there are two different equivalent circuit models: with and without center tap. Inductor Toolkit knows what you are working on, and will pick the right one automatically.

Optimization Cockpit ✕

Before closing the cockpit:

Update the Design

Don't Update the Design

Cancel

The optimization is configured to use the ADS optimization cockpit, so that you can watch the progress of the model fit. When the optimization is finished, **close the optimization cockpit** and confirm that you want to **update the design** with the optimized parameters.

If you stop the optimization manually, before reaching the maximum number of iterations, you have to **manually** start a re-simulation of the optimized schematic. Otherwise, the circuit model fit curves will be empty and no model is created.

red = data from input file
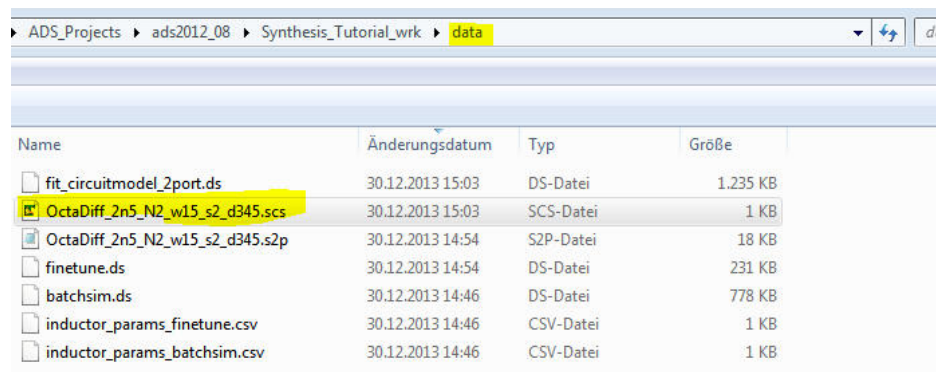blue with index m = model = fitted data
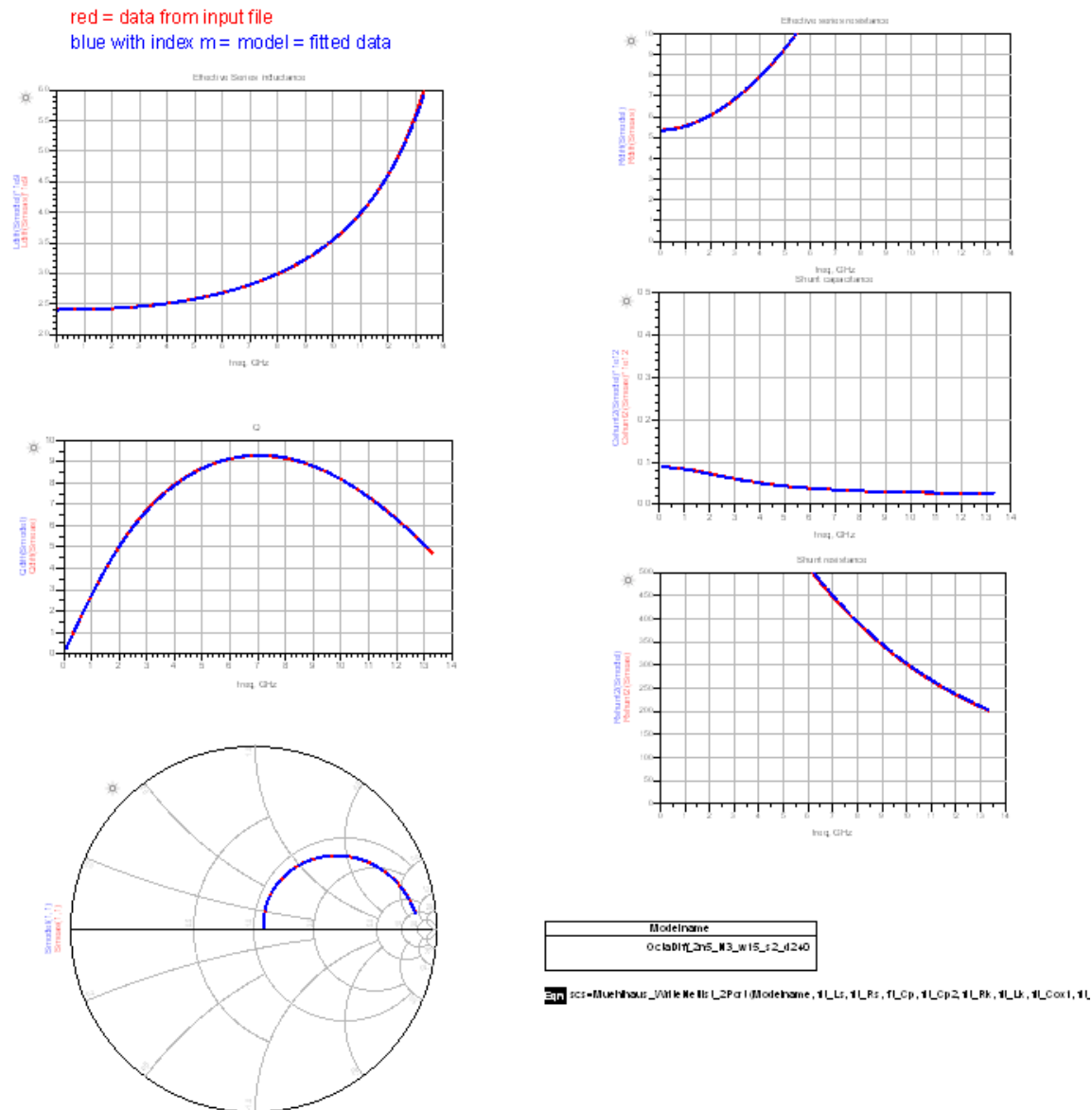
In most cases, you will see excellent agreement between the EM data (red curve) and the fitted circuit model (blue curve) for inductance, Q factor, S-parameters as well as series resistance and shunt path parameters.
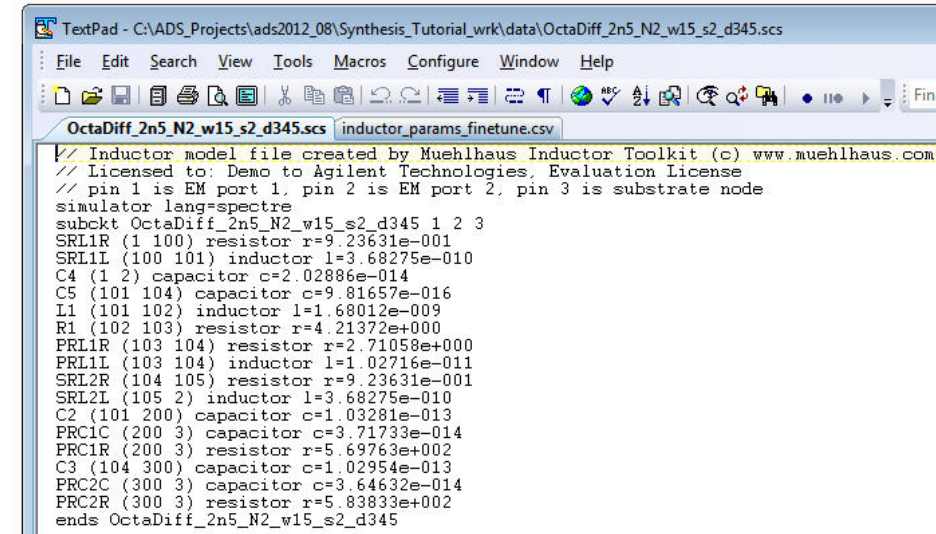
But if needed, in special cases, you can stop the optimizer and manually tweak the parameter range. You could also tweak the optimizer goals and frequency range, but that's hardly ever needed, because it is all set automatically based on your target frequency and the inductor's electrical parameters.

On the bottom right side, you see a list of equivalent circuit model parameters, and a function call that writes the equivalent circuit model to disk, in Spectre *.scs netlist format.

That file is created in the workspace's data directory. Looking at the file, you will see the netlist representation of the ADS equivalent circuit model.
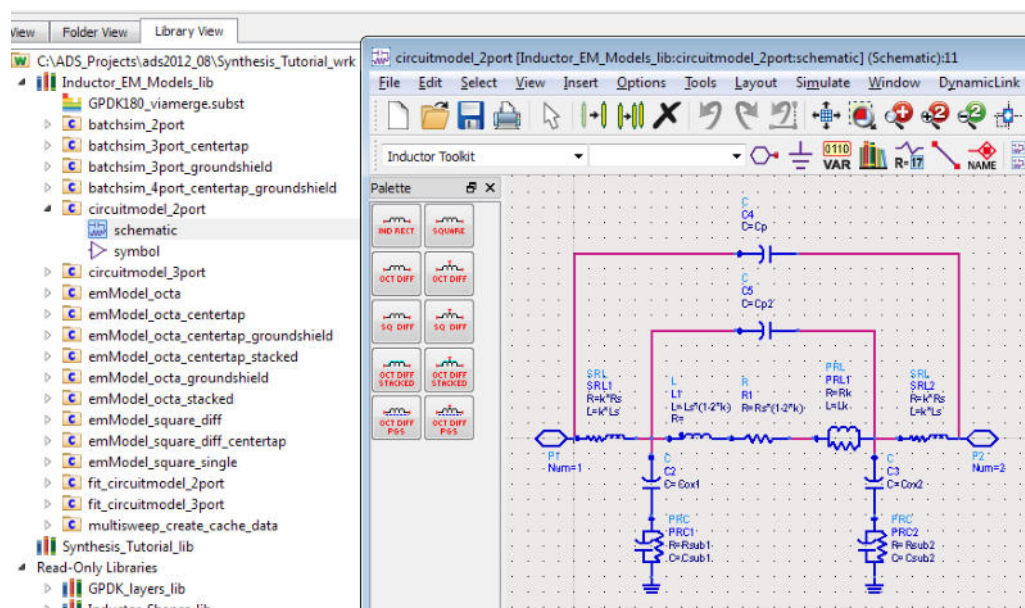

You can use that netlist in Cadence/Spectre. The topology is fixed, defined by Inductor Toolkit. The circuit model topology can be seen from the circuitmodel_2port.

Now, the final step is **6: Create inductor cell in library**.
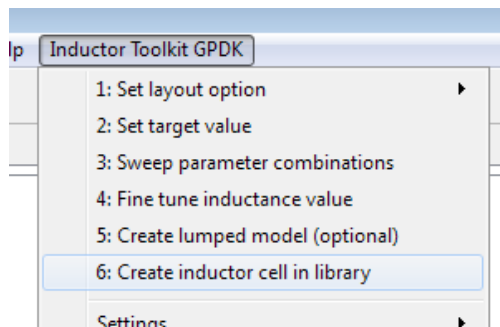
This will create layout and schematic view(s) of the optimum inductor that we have synthesized, and also create the final S-parameter file in the data directory. The inductor parameters are read from the fine tune synthesis step. The (optional) equivalent circuit model is the optimization result from step 5, Create lumped model.

The cell is created in the first writable workspace library that is not an internal Inductor Toolkit library. In our example, the new cell is created in the Synthesis_Tutorial_lib library.

The name of the cell is derived from the layout option and parameter values. For the GPDK example, the optimized inductor cell is OctaDiff_2n5_N3_w15_s2_d240 which tells us that this is a differential octagon inductor (without center tap) with 2.5nH target inductance, N=3 turns, w=15µm conductor width, s=2µm conductor spacing (actually 2.1), and 240µm outer diameter.
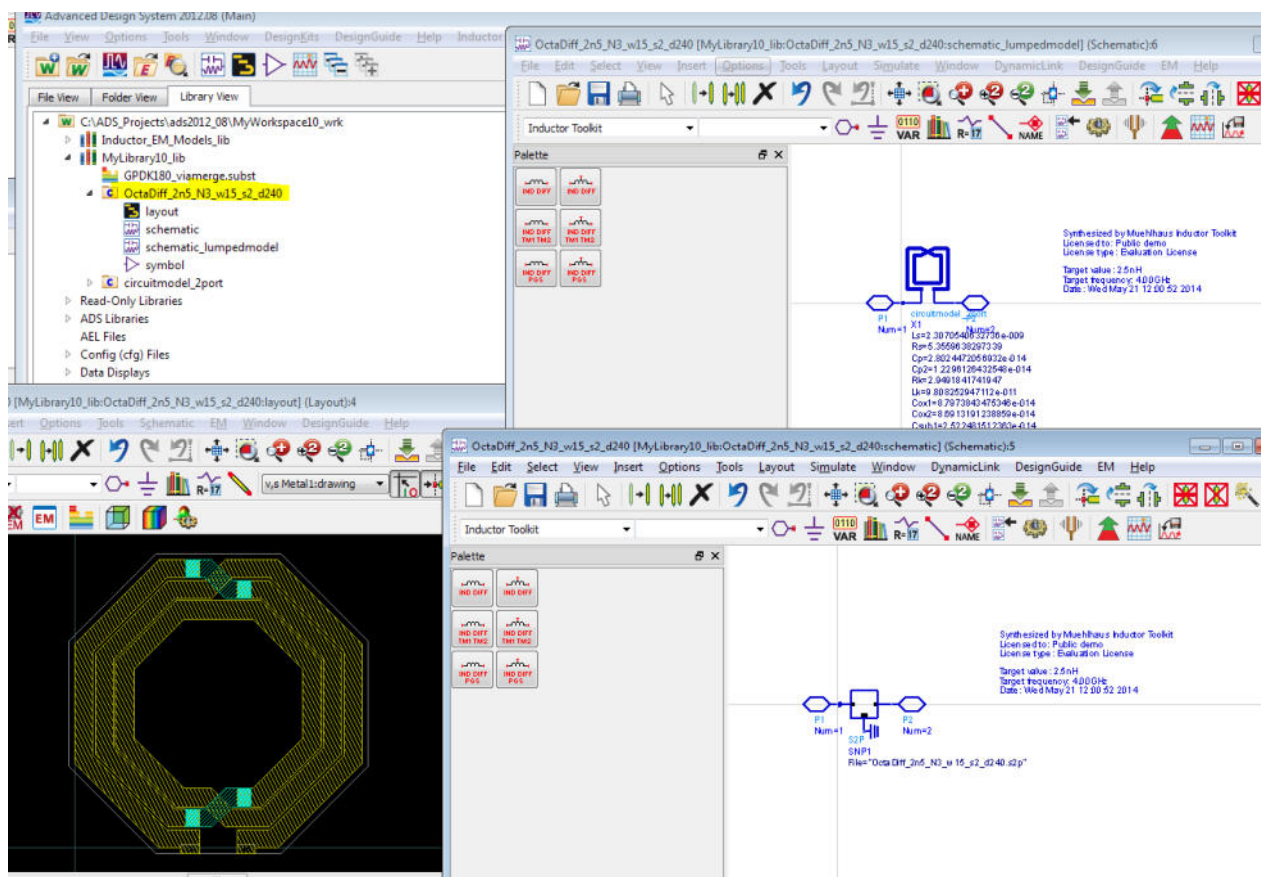
In the layout view, we have an instance of the differential octagon inductor pcell, with these geometry parameters.

The S-parameter and *.scs Spectre model file have been created in the workspace's data directory, with base name OctaDiff_2n5_N3_w15_s2_d240.

15

Second example: adjust the geometry limits

In this chapter, we want to go back to some more synthesis details.

In the initial example, you might have noticed that we specified 4GHz target frequency, but the best inductor had the Q factor peaking at 7GHz. Why wasn't that 4 GHz?

To understand that, we need just a tiny bit of theory.
At low frequency, the Q factor is determined by the series resistance.
At high frequency, the Q factor is determined by the shunt path, i.e. substrate coupling.

Finding the best possible inductor is about finding the best tradeoff between these two. We can reduce the (DC) series resistance by making the conductors wider, but the side effect is that the shunt capacitance to the substrate increases. This means that self resonance frequency (SRF) and frequency of maximum Q will go down.

In our first example, the frequency of maximum Q at 7GHz was higher than the 4 GHz target frequency, so that we could afford some wider conductor that causes f(Qmax) to drop. Why didn't the algorithm select that inductor with wider lines?

The answer is simple. It means that within our geometry space, there was no other inductor performing better at the 4GHz target frequency.
At the bottom of the data display, we can check the "top list" with layouts sorted by Q factor at the target frequency. This is what it looks like:

Top list sorted by Q factor

Q_sorted	N_sorted	dout_sorted	w_sorted	s_sorted
8.773	3.000	245.840	15.000	2.100
8.653	2.000	349.320	15.000	2.100
7.037	3.000	210.030	10.000	2.100
6.883	4.000	178.130	10.000	2.100
6.707	2.000	317.220	10.000	2.100
4.607	4.000	135.120	5.000	2.100
4.507	5.000	121.510	5.000	2.100
4.464	3.000	169.960	5.000	2.100
4.027	2.000	281.970	5.000	2.100

The best inductor has a conductor width of 15µm. This is **not** the maximum possible width of 30µm. Why is that, if 30µm width could possibly increase the Q factor?

Again, the answer is simple. You can see that for 15µm conductor width, the inductor's outer diameter is already close to the 350µm maximum outer diameter that we had specified. With 30µm conductor width, the inductor would be physically larger than our limit.

Why do we have that limit? The simple reason is that we can often build good inductors with a single turn that are physically large. However, these are just too large, because chip area is expensive. We want to be able to limit the size, and find the best inductor for a given, restricted inductor size.

Now, for our second example, we can adjust the maximum diameter and see what we get. Before you start, close all open Windows from the previous example. This is done with File > Close All from the ADS Main Window menu.

In the Inductor Toolkit menu, select **2: Set target value.** This opens the data display form where you can modify the geometry limitations.

Change Dout_maxsize from 350µm to 600µm and save your changes. Notice that the number of synthesized geometries has increased from 9 to 12.

Enter desired inductance value [H]

`Eqn` L_target=2.5e-9

Enter geometry limitations, dimensions in [µm]

`Eqn` N_min=1   `Eqn` N_max=5   `Eqn` N_step=1

`Eqn` w_min=5   `Eqn` w_max=30   `Eqn` w_step=5

`Eqn` s_min=2.1   `Eqn` s_max=2.1   `Eqn` s_step=5

`Eqn` Dout_maxsize=600

Ground shield inductors only: shield finger width in [µm]

`Eqn` w_shield = 5

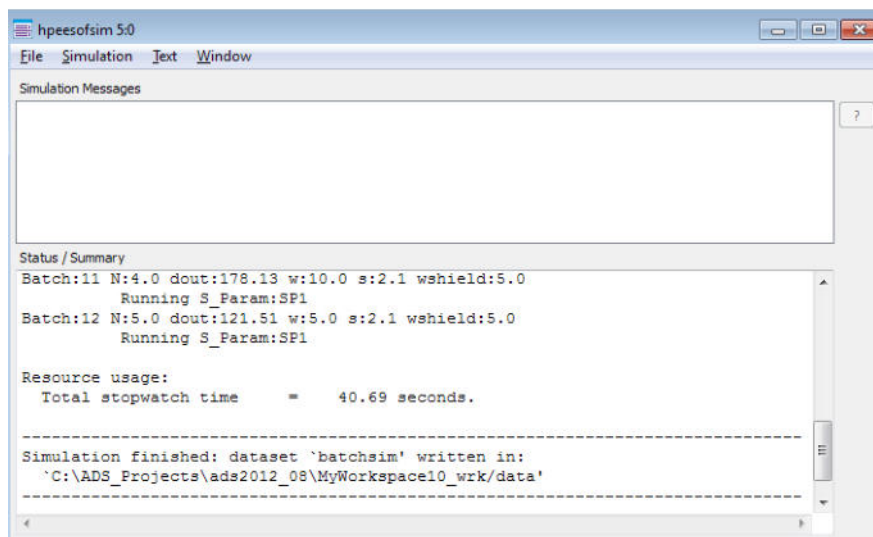Number of synthesized geometries: 12   OctaDiff

Set the desired target value and geometry limits,
then use schematic menu "Inductor Toolkit > Sweep Parameter Combinations"
to sweep the possible geometry combinations!

We have 3 additional inductor layouts that match the requirements. Let's see how they perform. In the Inductor Toolkit menu, click on **3: Sweep parameter combinations**. This will start the batch simulation of all the 12 different layouts.

You will notice that most of the simulations are very fast now, and only two layouts need some time for simulation. That is because the previously calculated results had been stored to disk, for later re-use. Only the two additional layouts must be EM simulated now, all the others results can be found in the existing emModel database.

```
hpeesofsim 5:0
File   Simulation   Text   Window
Simulation Messages                                                    ?




Status / Summary
Batch:11 N:4.0 dout:178.13 w:10.0 s:2.1 wshield:5.0
        Running S_Param:SP1
Batch:12 N:5.0 dout:121.51 w:5.0 s:2.1 wshield:5.0
        Running S_Param:SP1

Resource usage:
   Total stopwatch time    =    40.69 seconds.

----------------------------------------------------------
Simulation finished: dataset `batchsim' written in:
   `C:\ADS_Projects\ads2012_08\MyWorkspace10_wrk/data'
----------------------------------------------------------
```
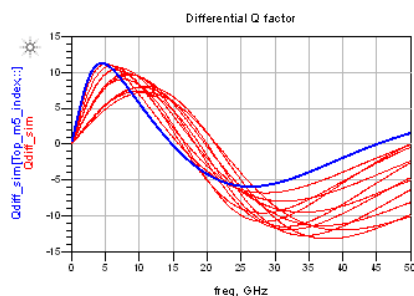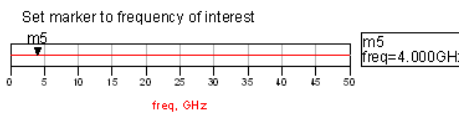
So what about results? If you had saved the data display in tutorial example 1, it will remember the target frequency. Otherwise, please set it to 4GHz again.
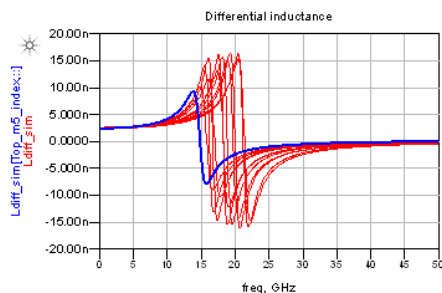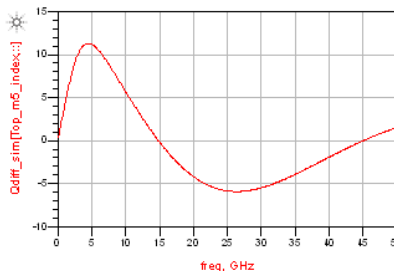
Set the slider to the desired target frequency.
The inductor with the best Q factor for that frequency is selected, and results plotted.
You can find the corresponding geometry data below.

If the result matches your requirements, proceed to fine tuning of the inductance value.
schematic menu "Inductor Toolkit > Fine tune inductance value"
If the result is not yet satisfactory, go back to the beginning and modify your geometry limitations.

Set marker to frequency of interest

m5
freq=4.000GHz

Layout			OctaDiff
Qdiff=	11.183	at	4.000 GHz
Ldiff(nH)=	2.645	R DC (Ohm)=	
N=	2.000		
Do(µm)=	434.440	Di(µm)=	310.
w(µm)=	30.000		
s(µm)=	2.100		

Top list sorted by Q factor

Q_sorted	N_sorted	dout_sorted	w_sorted	s_sorted
11.183	2.000	434.440	30.000	2.100
10.751	2.000	407.480	25.000	2.100
9.879	2.000	379.230	20.000	2.100
8.773	3.000	245.840	15.000	2.100
8.653	2.000	349.320	15.000	2.100
7.037	3.000	210.030	10.000	2.100
6.883	4.000	178.130	10.000	2.100
6.707	2.000	317.220	10.000	2.100
4.607	4.000	135.120	5.000	2.100
4.507	5.000	121.510	5.000	2.100
4.464	3.000	169.960	5.000	2.100

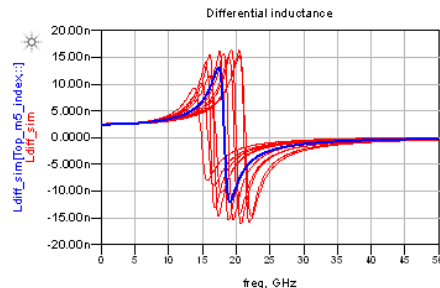Now, a different "best" inductor is found, with higher Q than before.

As expected, that inductor now uses the widest possible metal (30µm) which brings the frequency of maximum Q down to 4.5GHz. The outer diameter is 434µm now, which was outside the valid geometry range in our first example. Now, it is an engineering decision if the better Q factor is worth the extra chip area, or if you want to go with the smaller inductor and compromise on Q factor.

Top list sorted by Q factor

Q_sorted	N_sorted	dout_sorted	w_sorted	s_sorted
11.183	2.000	434.440	30.000	2.100
10.751	2.000	407.480	25.000	2.100
9.879	2.000	379.230	20.000	2.100
8.773	3.000	245.840	15.000	2.100
8.653	2.000	349.320	15.000	2.100
7.037	3.000	210.030	10.000	2.100
6.883	4.000	178.130	10.000	2.100
6.707	2.000	317.220	10.000	2.100
4.607	4.000	135.120	5.000	2.100
4.507	5.000	121.510	5.000	2.100
4.464	3.000	169.960	5.000	2.100

Of course, going for the widest metal isn't always benefical. You can easily see that by moving the target frequency to other values. Try it, move the slider and watch how different target frequencies lead to different optimum geometries.
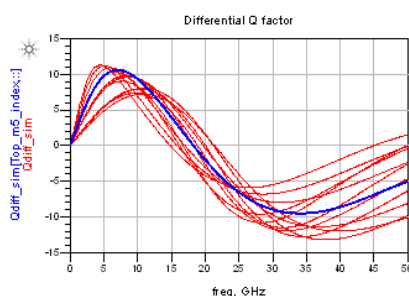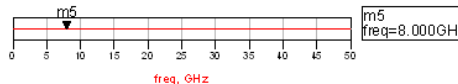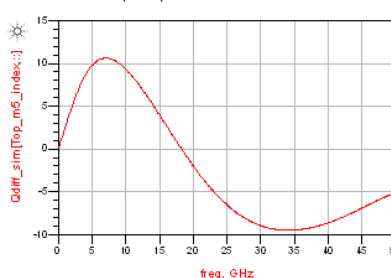
Let's change the target frequency to 8GHz, and design the best possible inductor for that case. You can see that the conductor width is 15µm now, and we still have 2 turns as the best choice.


The inductance for that layout is 3nH at 8GHz, instead of our 2.5nH target inductance. One reason is that the synthesis equation isn't perfect, and the other reason is the increase in effective inductance towards the self resonance frequency.
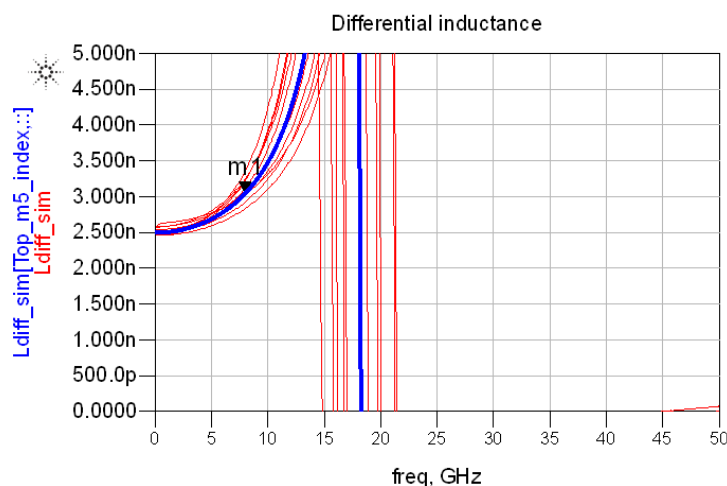
Save the data display, and proceed to **4: Fine tune inductance value**, to adjust the inductance for the new target frequency.

The fine tuned result is a slightly smaller inductor, with the inductance value decreased to 2.51nH. That is very close to our target value already. In those cases where it is not close enough, we could also repeat the fine tune step **again**, to iterate towards the specified value.
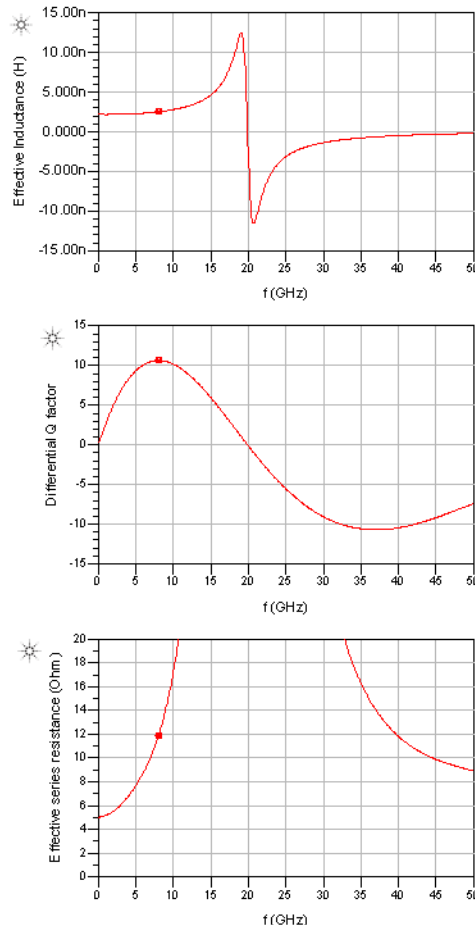
**Let's have a closer look, because it is important to understand what Inductor Toolkit fine tuning can do, and what it can't do.** The fine tune step adjusts the inductance by tweaking the inductor diameter, and it also takes that frequency dependence into account.

However, the frequency dependence of the tuned inductor is slightly different, because the parasitics also change when the diameter changes. Inductor Toolkit tries its best to account for all these effects, but that becomes more difficult as the target frequency is on that steep slope towards self resonance. If the inductance after the initial fine tune was not accurate enough, just repeat it to iterate towards the desired value.
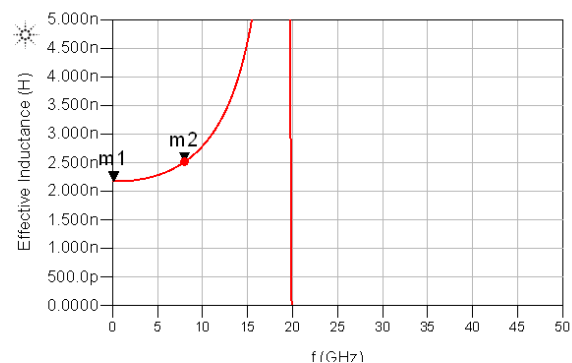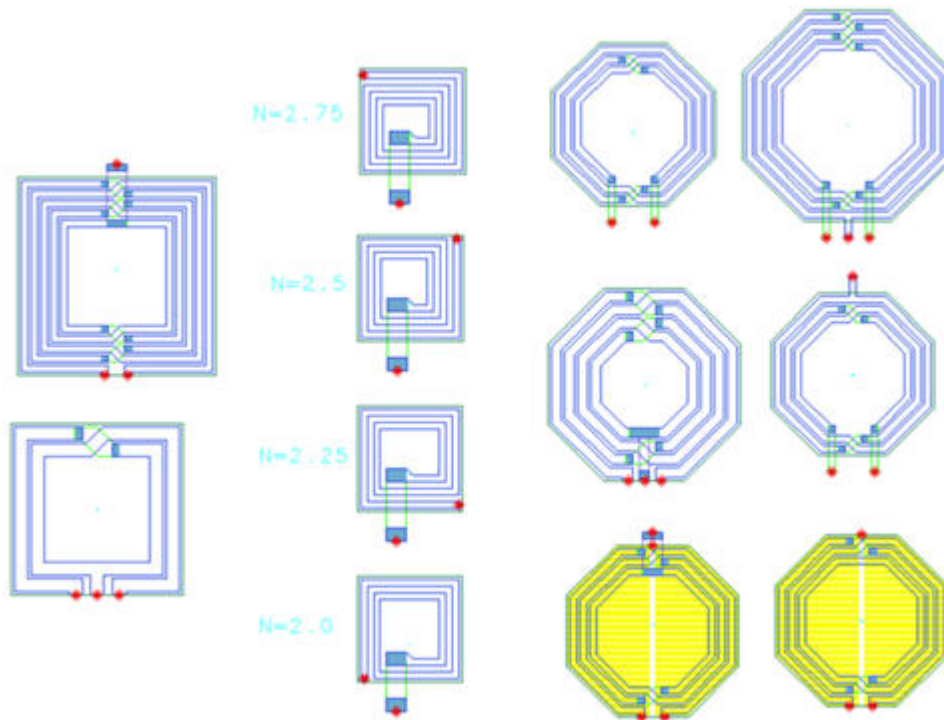
Third example: other layout options

So far, we have synthesized octagon inductors, and investigated the effect of geometry parameters. Now, it's time to have a look at other layout options.

The full, detailed list of layout options is included in the User's Guide.
For our tutorial, we keep it short and look at this overview picture:

There are two main categories: symmetric (differential) inductors and asymmetric inductors.

The symmetric inductors are implemented with square or octagon shape, and some special options for those are available as well. All these symmetric inductors must have "full" number of turns, e.g. 2 or 3.

For symmetric inductors with center tap, the center tap location and orientation is different between even and odd number of turns. For even number of turns, the center tap (pin 3) is located between inductor pins 1 and 2. For odd number of turns, it is located on the opposite side. To restrict the location of the centertap, you can specify the number of turns to change in steps of 2, so that only layouts with even (or odd) turns are synthesized.

For asymmetric inductors, the only layout option is square shape. Here, the number of turns can be set in steps of 0.25 and that determines the pin location/direction. See picture above for an example.

Now, in this third example, we want to synthesize an asymmetric 12nH inductor with feedlines on opposite sides.

Before we start, close all open windows. Then, in the ADS main window, go to the Inductor Toolkit menu and in **1: Set layout option**, click on "**Square single ended**"
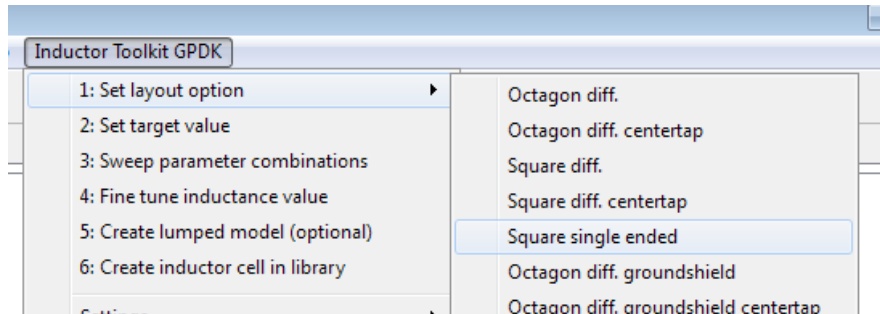
Now, go to step **2: Set target value.** This will open the input form where we can set the exit direction by specifying the appropriate fractional number of turns (N=x.5 turns).
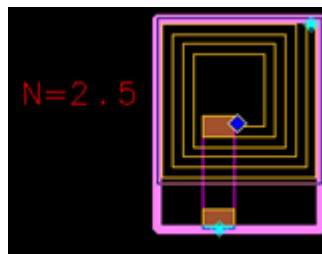
For the target, we set 12nH. Minimum and maximum turns are changed to 1.5 turns and 5.5 turns, respectively, for feedlines at 180° direction. The N_step setting is 1 again, so that the algorithm uses only those 180° inductor layouts (and not 0°, 90°, -90°).

For the spacing, we can set the value to the minimum possible value 2μm (see design rules). This is because such a larger inductance value will require more turns, and we want to keep the physical size small. For the minimum outer diameter, we set 500μm.

Enter desired inductance value [H]

Eqn L_target =12e-9

Enter geometry limitations, dimensions in [μm]

| **Eqn** N_min=1.5 | **Eqn** N_max=5.5 | **Eqn** N_step=1 |
| **Eqn** w_min=5 | **Eqn** w_max=30 | **Eqn** w_step=5 |
| **Eqn** s_min=2 | **Eqn** s_max=2 | **Eqn** s_step=2 |

Eqn Dout_maxsize=500

Ground shield inductors only: shield finger width in [μm]

Eqn w_shield = 5

Number of synthesized geometries: 11   SquareSingle

Set the desired target value and geometry limits,
then use schematic menu "Inductor Toolkit > Sweep Parameter Combinations"
to sweep the possible geometry combinations!

Now with these settings, we can start the batch simulation of possible layouts. In the Inductor Toolkit menu, click on **3: Sweep parameter combinations**.
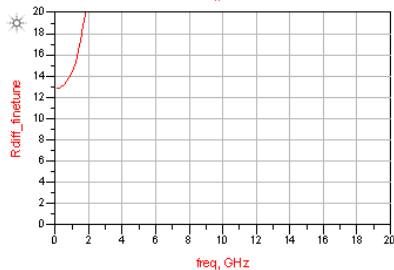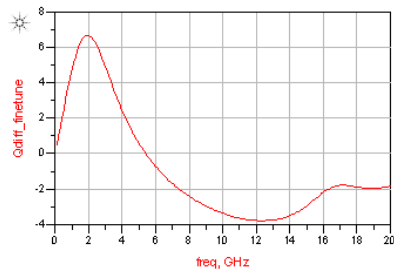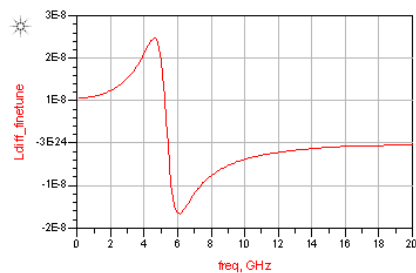
After simulation, the data display comes up. It still has the 8GHz target frequency setting from our previous example, and Inductor Toolkit issues a warning that this can't be applied to the current data.
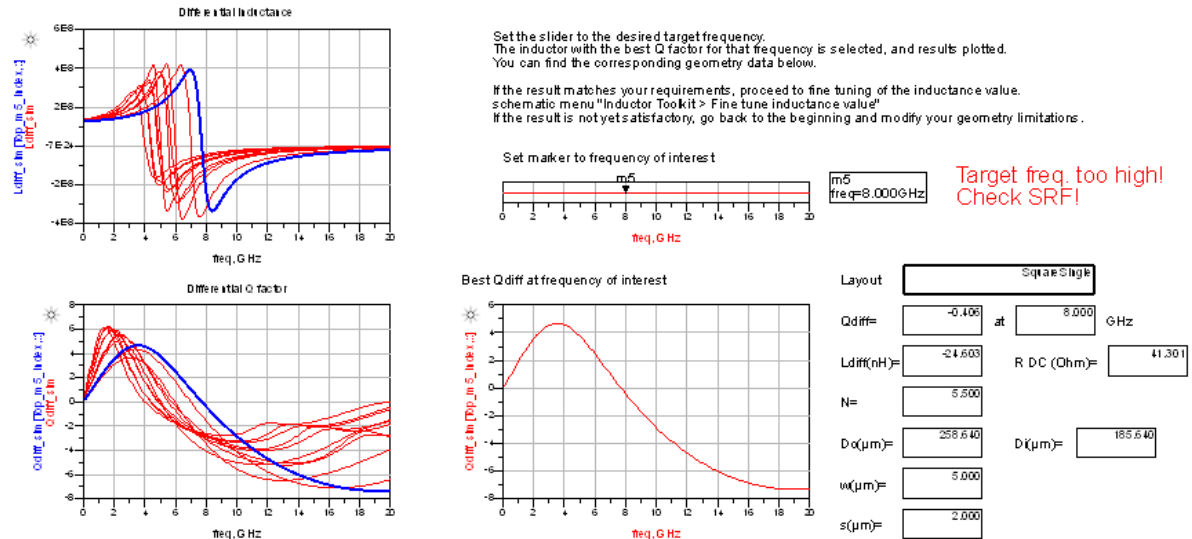
Change the target frequency to 1.8 GHz and proceed to **4: Fine tune inductance value**.
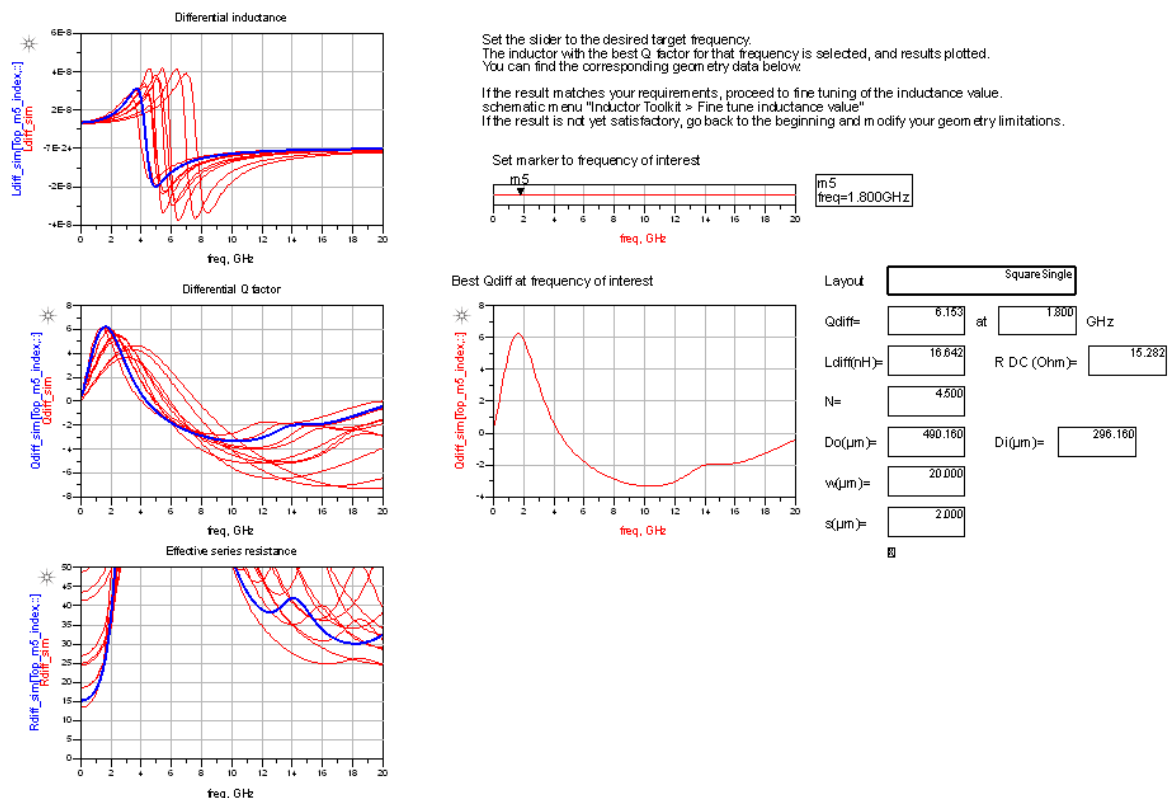
The fine tune step results in 12.01 nH inductance at 1.8GHz, with a Q factor of 6.6.

| Layout	SquareSingle		
Qdiff=	6.656	at 1.800 GHz	GHz
Ldiff(nH)=	12.010		
R DC (Ohm)=	12.837		
N=	4.500		
Do(µm)=	425.230	Di(µm)=	231.230
w(µm)=	20.000		
s(µm)=	2.000		

Touchstone S-parameters are written to workspace data directory
Eqn snp=write_snp(Modelname, Sfull[0,::])

Let's assume we don't need an equivalent circuit model, so we directly proceed to step **6: Create inductor cell in library**. This creates a new cell with layout view and schematic (S-parameter) view: SquareSingle_12.0nH_N4.50_w20_s2_d425
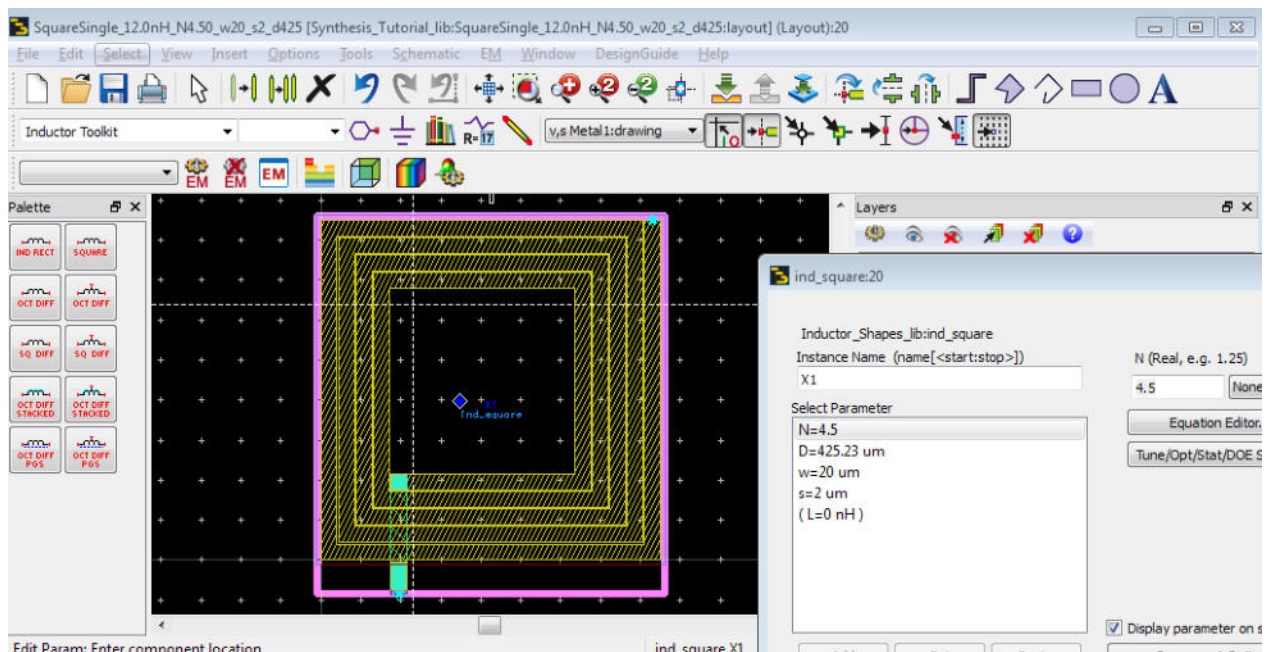
This ends our tutorial tour. You are now well prepared to explore other layout options, and enjoy efficient and fast inductor synthesis with Inductor Toolkit!

Changes

Document version 1.5 of 24. April 2020:
Updated screenshots and instructions for newer ADS versions

Document version 1.4 of 21. May 2014:
Updated examples. Mentioned possible iterative finetune step, if first iteration isn't close enough to target value.

Document version 1.2 of 29. January 2014:
Updated installation instructions.

Document version 1.1 of 20. December 2013:
Updated for new interpolation of width in fine tuning step, changed first example maximum outer and spacing.