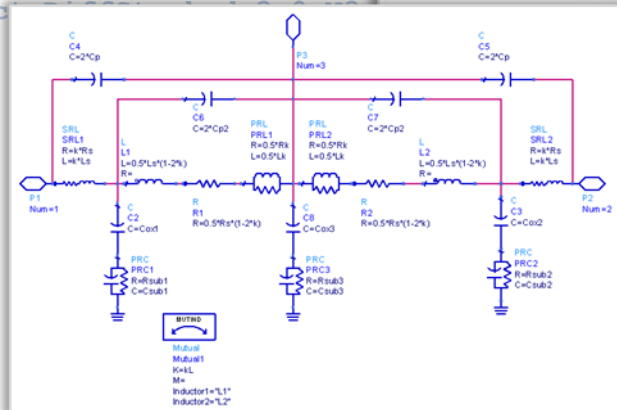
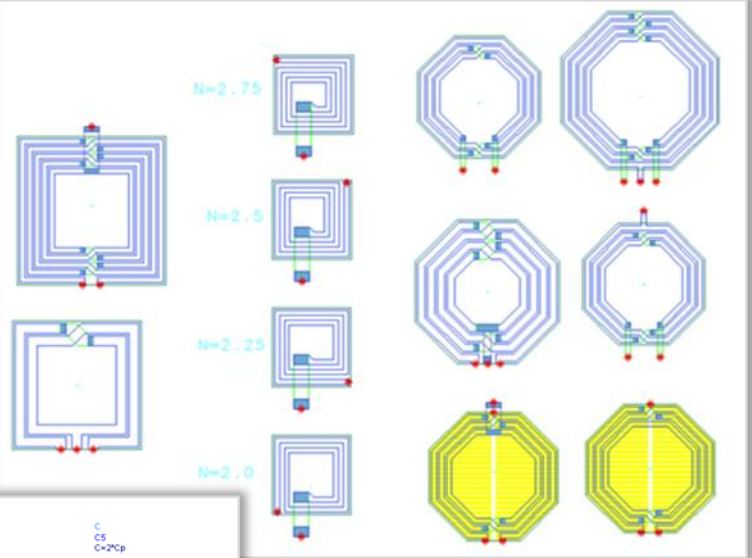


# Muehlhaus RFIC Inductor Toolkit Generic Version

## Inductor Synthesis for Keysight ADS

```
// Inductor model file www.muehlhaus.com
simulator lang=spectre
subckt OctaDiffStacked_2n0_N2_wl8_s2_d332 1 2 3
SRL1R (1 100) resistor r=9.81372e-002
SRL1L (100 101) inductor l=2.00001e-010
C4 (1 2) capacitor c=3.38452e-010
C5 (101 104) capacitor c=8.9e-010
L1 (101 102) inductor l=1.41e-010
R1 (102 103) resistor r=6.09e-002
PRL1R (103 104) resistor r=1.2e-002
PRL1L (103 104) inductor l=8.9e-010
SRL2R (104 105) resistor r=9.81372e-002
SRL2L (105 2) inductor l=2.00001e-010
C2 (101 200) capacitor c=1.41e-010
PRC1C (200 3) capacitor c=4.41e-010
PRC1R (200 3) resistor r=1.2e-002
C3 (104 300) capacitor c=1.41e-010
PRC2C (300 3) capacitor c=4.41e-010
PRC2R (300 3) resistor r=1.2e-002
ends OctaDiffStacked_2n0_N2_wl8_s2_d332
```



## Table of contents:

Table of contents:	2
Introduction	3
Requirements	4
Installation	4
Installation path	4
Adding Inductor Toolkit to an ADS workspace	5
ADS2016	5
ADS2017	7
License File	8
License Installation, License Upgrade	8
How it works	9
Getting started	10
Step 1: Set layout option	11
Step 2: Set target value	12
Step 3: Sweep parameter combinations	13
Step 4: Fine tune inductance value	15
Step 5: Create lumped model	18
Lumped model topologies	21
Step 6: Create inductor cell in library	22
Using the new inductor cell in your ADS designs	23
Using the new inductor in Cadence	24
Import inductor layout to Cadence	24
Schematic view for simulation in Cadence	24
Step 1: Symbol view	24
Step 2: Spectre view	24
Step 3: Add *.scs file to ADE library path	26
Layout Options - Inductor Pcells	27
Octagon differential (ind_diff_2port)	27
Octagon differential centertap (ind_diff_3port)	27
Octagon differential stacked (ind_diff_stacked_2port)	28
Octagon differential stacked centertap (ind_diff_stacked_3port)	28
Octagon differential groundshield (ind_diff_pgs_2port)	29
Octagon differential groundshield centertap (ind_diff_pgs_3port)	29
Square differential (ind_square_diff_2port)	30
Square differential centertap (ind_square_diff_3port)	31
Square single ended (ind_square)	31
Single turn (N=1) centertap (ind_pH_singleturn_3port)	32
Single turn (N=1) stacked centertap (ind_pH_singleturn_stacked_3port)	32
Data Display Templates and Expressions	33
Single ended and differential operation	33
Custom inductor expressions for ADS data display	34
Configuration and Customization	35
Frequency range	35
Ground ring around inductor	35
Technology settings	40
Edit Momentum substrate file vs. complete technology configuration	40
Create Inductor Toolkit version for your own technology from scratch	41
Editing the Inductor Toolkit technology file	45
Auto-update the emModels after technology changes	48
Preference file	49
Changes	50

## Introduction

Mühlhaus RFIC Inductor Toolkit assists ADS users with finding the “optimum” inductor layout in Silicon-based RFIC/MMIC technologies.

In SiGe and CMOS, the lossy silicon substrate causes significant high frequency losses in the substrate, limiting the maximum available Q factor. Designing an inductor for these technologies requires a trade-off between low frequency conductor losses (series losses) and high frequency substrate losses (shunt losses), where the optimum geometry depends on frequency.

Thus, instead of using a library with a limited set of inductors, many design groups today want to synthesize the best possible inductor for their specific needs. This is where Inductor Toolkit can assist ADS users, by automating the inductor design process, based on the user's requirement.

Inductor Toolkit provides:

- Layout Pcells for popular inductor geometries
- emModels for these inductors
- Preconfigured example substrates for the Momentum EM simulator
- Automation to synthesize inductor dimensions based on inductance
- Automation to find the best performing inductor with best possible Q factor, within given geometry limitations
- Automation to generate the layout for that “optimum” inductor
- Automation to create an ADS cell with layout view, S-parameter simulation view and equivalent circuit model view
- Automation to fit a lumped inductor model, and output that as an \*.scs netlist file

**Inductor Toolkit GENERIC is configured for a sample technology, and can be configured for your actual process technology.** This configuration is done by the end user, as described in this document, or available as an optional service from Mühlhaus Consulting & Software GmbH.

Factory configured versions of Inductor Toolkit are available for these technologies:

IHP SG25H3/H4, IHP SGB25V, IHP SG13S, IHP SG13G2  
Infineon B7HF200, B11HFC

## Requirements

Inductor Toolkit requires for **ADS 2016 or later** (license not included).

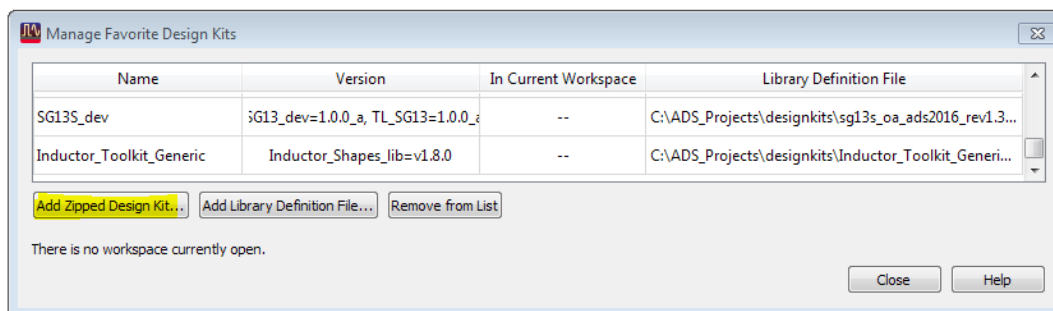
For full features, a **purchased license** for Inductor Toolkit Generic is required. The free demo version only supports a limited set of inductor layouts.

The pre-configured example technology uses the ADS “IC interoperability” feature. If you prefer to use Inductor Toolkit with workspaces in the “ADS only” mode, some initial changes are required, as described later in this manual.

## Installation

Mühlhaus RFIC Inductor Toolkit can be used “stand alone”, or together with an ADS RFIC PDK for your technology. In this document, the “stand alone” use is documented, based on example technology/layer data included in the Inductor Toolkit PDK.

Installing the InductorToolkit PDK is similar to other ADS PDKs: in the ADS main windows, go to *DesignKits > Unzip Design Kit ...* and install the PDK from the \*.zip file.



### Installation path

Unlike other PDKs, **the inductor toolkit libraries must be writable**, because emModels and schematics from Inductor\_EM\_Models\_lib are updated during synthesis.

**You must install Inductor Toolkit with write access for the users at file level, because users can customize settings and schematics.** For example, if you store your ADS workspaces to \$HOME/ads2016.01, you could create a new directory \$HOME/designkits and unzip Inductor Toolkit to that location.

**Do not install Inductor Toolkit to a shared directory where users have read access only!**

## Adding Inductor Toolkit to an ADS workspace

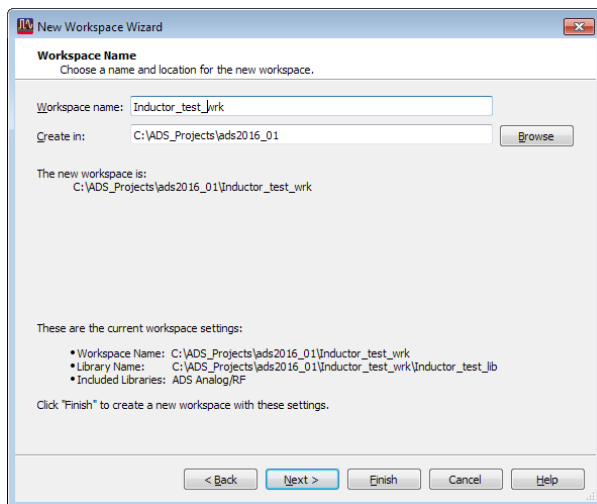
Inductor Toolkit requires some technology settings, e.g. layer and substrate definitions. These can come from an existing Foundry PDK, or from the default technology library that is included with Inductor Toolkit.

By default, Inductor Toolkit is configured to use its internal default technology. In this case, your ADS workspace only needs to load the Inductor Toolkit PDK.

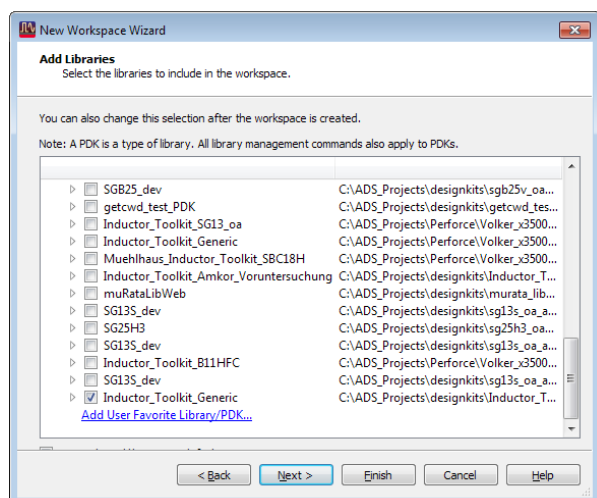
The use of design kits in ADS 2017 has changed, so both ADS2016 and ADS2017 use of Inductor Toolkit is shown below.

### ADS2016

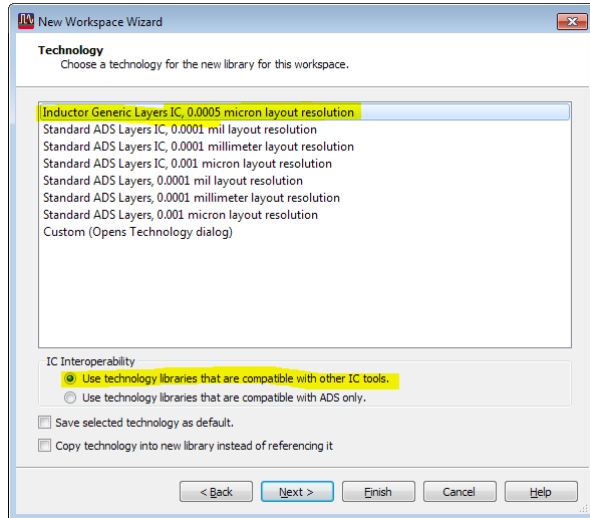
To create a new workspace that uses Inductor Toolkit, use File > New > Workspace ...



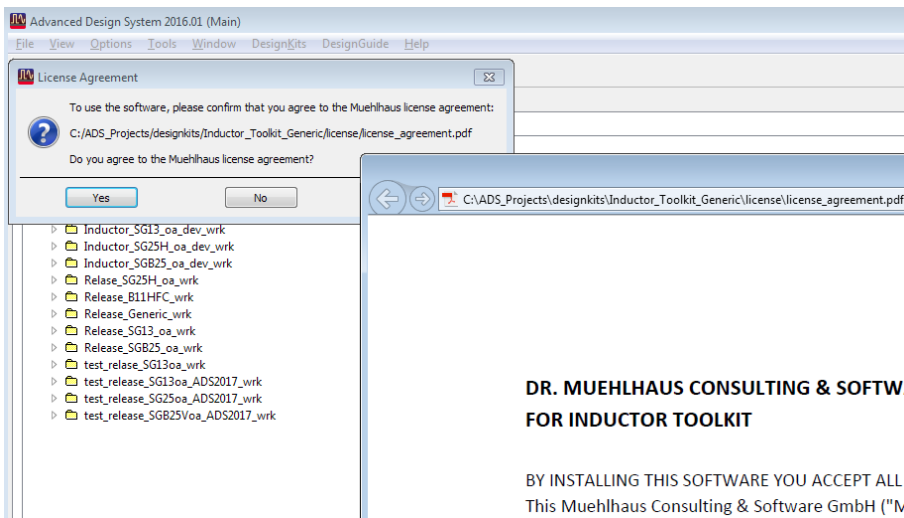
... enable the Inductor\_Toolkit\_Generic design kit, assign a name to the workspace library and then ...



... then select the Inductor Toolkit default technology. This default library uses ADS IC interoperability mode, so you need to select the “Use technology libraries that are compatible with other IC tools” option.



You can now press Finish, and the Inductor Toolkit license agreement PDF document will be shown. After you have agreed to that, you can start using Inductor Toolkit.



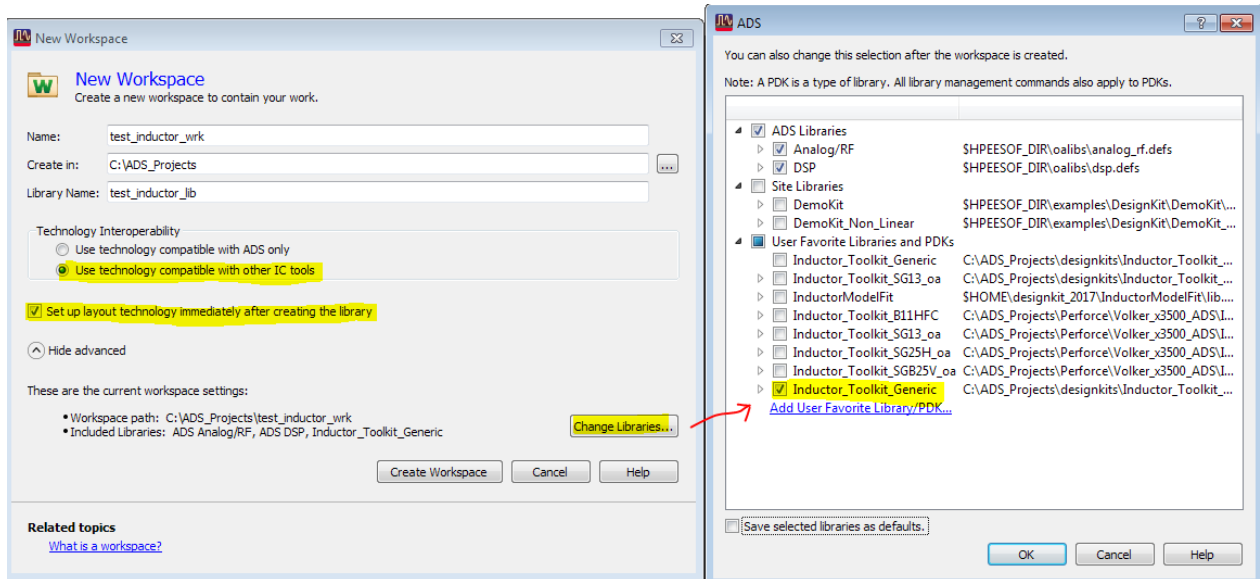
**DR. MUEHLHAUS CONSULTING & SOFTWARE  
FOR INDUCTOR TOOLKIT**

BY INSTALLING THIS SOFTWARE YOU ACCEPT ALL T  
This Muehlhaus Consulting & Software GmbH ("M")

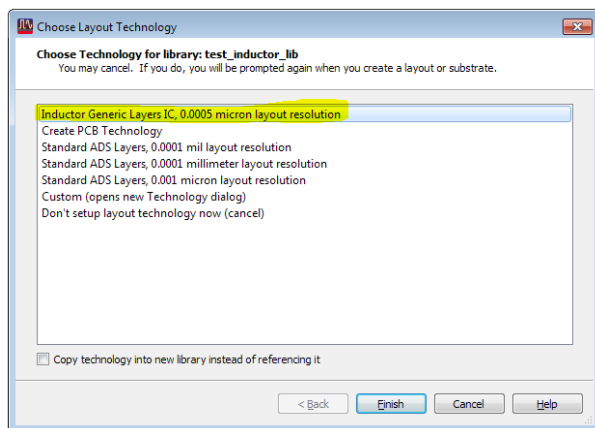
## ADS2017

To create a new workspace that uses Inductor Toolkit, use File > New > Workspace

Click on the “Show advanced” button, **enable** the “Use technology compatible with other IC tools” option and **enable** “Set up layout technology immediately after creating the library”. Then, click the “Change Libraries...” button to enable the Inductor\_Toolkit\_Generic PDK.



Click the „Create Workspace“ button. The Choose Layout Technology dialog will be displayed, and you need to select the Inductor Toolkit default technology. This default library uses ADS IC interoperability mode.



When you press finish, the Inductor Toolkit license agreement PDF document will be shown. After you have agreed to that, you can start using Inductor Toolkit.

## License File

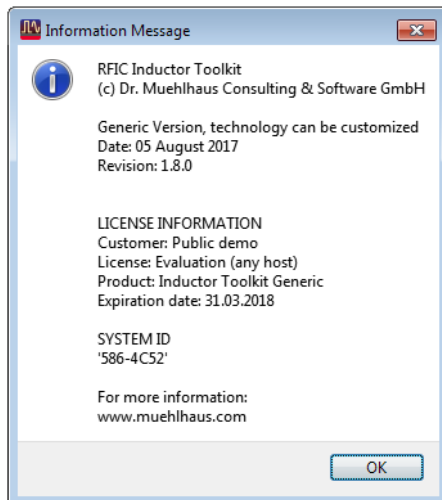
Inductor Toolkit comes pre-configured with an evaluation license. The license file is located in the license directory in the PDK, e.g.

*C:\designkits\Inductor\_Toolkit\_GPDK\license\license.lic*

You can check the license information from the Inductor Toolkit menu item

*"Toolkit version and license information..."*

which brings up a dialog window with all license information. The license expiration date is shown in format DD/MM/YYYY.



### **License Installation, License Upgrade**

If you have purchased an Inductor Toolkit license, you need to **manually replace the license file**. **Copy** the new "Inductor Toolkit Pro" license file into the license directory in the Inductor Toolkit PDK path, e.g.

*C:\designkits\Inductor\_Toolkit\_GPDK\license\license.lic*

and overwrite the existing license file. Delete or rename any other \*.lic file in that directory.

**You must not have multiple \*.lic files in the license directory**, because that will cause license detection problems!



## How it works

Here is a brief overview of the method, so that you can use Inductor Toolkit efficiently.

To synthesize an inductor, you first specify the required layout option and your target value for inductance, along with the geometry limitations. With this data, possible parameter combinations are calculated, which all result in this inductance value. Some parameter combinations are dropped, because they are not physically possible with an inductor diameter that is too small to draw a valid layout, or too large for the given size limitation. Only those parameter combinations are used that enable a valid layout for the given target inductance.

Next, a scalable layout cell is EM simulated with these valid parameter combinations. EM simulation is done with Momentum RF mode, with a default mesh setting that is fast and reasonably accurate. The EM simulations use different parameters, but the same parameterized emModel, which is predefined in the PDK for different inductor topologies (octagon or square, with/without center tap, normal or stacked metal, with/without ground shield). The advantage of using an emModel is that existing results are stored in a database, and can be reused for the same set of parameters, or reused for interpolating values that are close.

Now that different results over a range of possible implementations are available, the data display template (evaluate\_batchsweep) will plot inductor parameters for all these results, and pick the geometry that has performed best. Best performance is defined as the best Q factor at the frequency of interest. This frequency of interest is set by the user, with a slider on the data display.

Now the best implementation is selected, but the inductance might not be exactly the desired value at the frequency of interest. To adjust the value, the difference between target inductance and simulated inductance is calculated, resulting in the new, final inductor size. Also, the conductor width is fine tuned, based on the performance of different conductor width in the batch simulation step. With the final dimensions, another EM simulation is started. After that simulation is finished, the final electrical parameters and geometry parameters are shown (evaluate\_finetune). The fine tune step can be repeated iteratively, if the calculated inductance is not close enough to the target value. If you wish, you can define separate EM simulation settings for the fine tune step, to make this more accurate than the initial batch simulation.

Up to this point, all simulations had been done on a parametric layout, with parameters supplied as data tables. Now that we have determined the best dimensions, a “static” layout cell with that exact geometry can be created.

Also, for geometries without ground shield, an equivalent circuit model can be fitted to the EM simulated data. This uses the ADS optimizer, but it is highly automated, so that you don’t need to set up anything yourself. You only need to start the optimizer and save back the optimized results. Results are shown in the data display (inductor\_parameters\_2port) and also written to a Spectre netlist file<sup>1</sup> (\*.scs) in the workspace data directory.

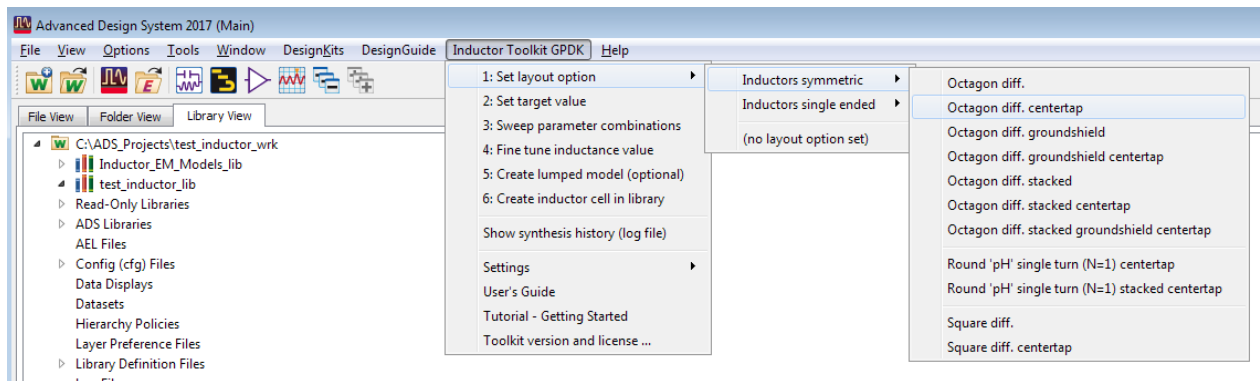
---

<sup>1</sup> This netlist output is programmed specifically for this data display and this equivalent circuit model in Inductor Toolkit, so it cannot be used with other input data.

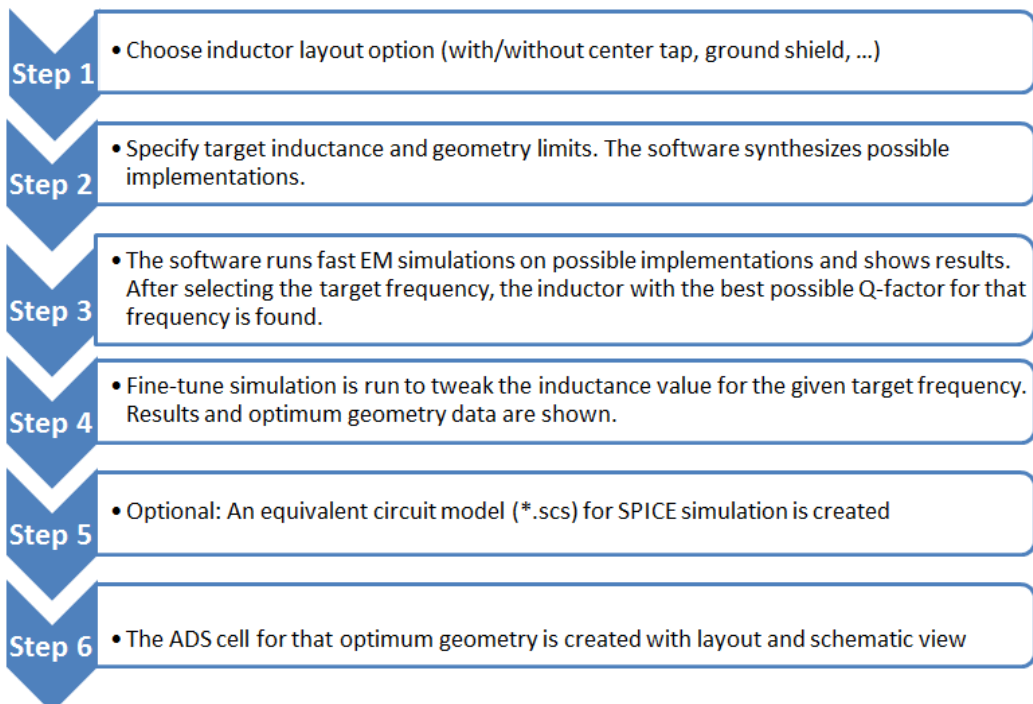
## Getting started

This chapter describes the synthesis process, without looking “behind the scenes” of the underlying technology. If you are interested, those details can be found later in this document.

Inductor synthesis is done from the “Inductor Toolkit” menu that you can find in the ADS main window.

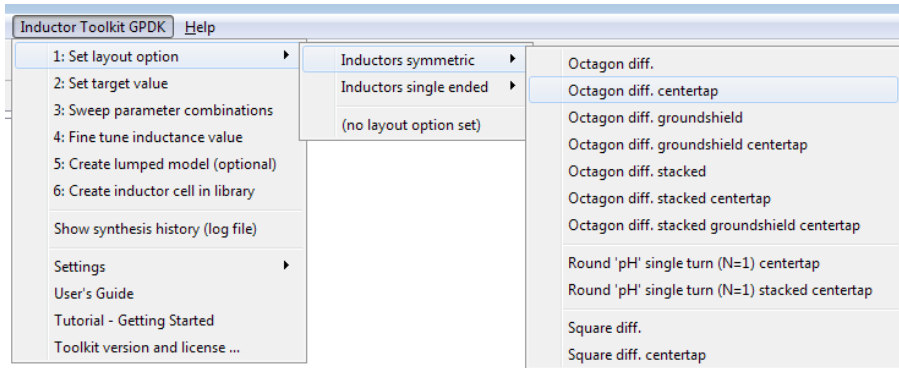


The menu structure is based on the synthesis workflow steps. These must be followed exactly in the given order, without skipping a step in the middle. The only exception is the optional step 5, equivalent circuit model fit.



## Step 1: Set layout option

Synthesis starts with the choice of an inductor geometry. Depending on your licensed option, you can choose from different octagon geometries. Layout choices might also include some foundry-specific options, that match the foundry's inductor Pcells.



Here is a brief description of the layout choices. More details and pictures can be found in chapter "Layout Options - Inductor Pcells".

**Octagon diff.** is an 8-sided inductor on a single metal layer, with two pins. The layer for that inductor is usually the topmost thick metal layer, this is configured in the technology file. Synthesis is done for full turns, where both feedlines are located on the same side.

**Octagon diff. centertap** is similar to Octagon diff., but includes a center tap (port 3).

**Octagon diff. groundshield** is similar to Octagon diff., but with a patterned ground shield on the lowest metal level (usually Metal1). This ground shield attracts the electric field that would otherwise couple to the lossy substrate. Connection to that ground shield is made through an additional pin. When grounded, this increases the shunt capacitance, but short circuits the lossy R||C shunt path in the substrate.<sup>2</sup> Note that there are some patents around related to inductors with patterned ground shields.

**Octagon diff. groundshield centertap** is similar to Octagon diff. groundshield, but with a center tap.

**Octagon diff. stacked** is similar to Octagon diff., but with stacked metal on two metal layers. Usually, these layers are the two topmost layers with thick metal. Compared to the regular octagon, this inductor has less resistance at low frequencies, but an increased capacitance to the substrate, so that the frequency of maximum Q is lower. Use this geometry when the Q factor is limited by ohmic conductor loss.

**Octagon diff. centertap stacked** is similar to Octagon Diff. stacked, but with a center tap.

**Square diff.** is similar to Octagon diff., but for square inductor shape.

**Square diff. centertap** is similar to Square diff, but includes a center tap (port 3).

<sup>2</sup> Yue, C.P.; Wong, S.S., "On-chip spiral inductors with patterned ground shields for Si-based RF ICs," *Solid-State Circuits, IEEE Journal of*, vol.33, no.5, pp.743,752, May 1998, doi: 10.1109/4.668989

**Square single ended** is a regular square inductor (not symmetrical) with four different orientations of the underpass. This can be set by specifying fractional number of turns in steps of 0.25. See chapter “Layout Options - Inductor Pcells” for more details.

## Step 2: Set target value

Next, we need to set the target value. This brings up a window where the required inductance value and geometry limits are specified. Note that the target frequency is not yet specified here – that will be done later. In this first step, all inductors are synthesized based on their DC values, and the best inductor for a given frequency will then be selected in step 3.

Enter desired inductance value [H]  
Eqn L\_target=2n

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=automatic

Eqn s\_min=5

Eqn s\_max=5

Eqn s\_step=5

Eqn Dout\_maxsize=500

Number of synthesized geometries:

OctaDiffTap  
0° +0.00 turns

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

The target inductance is the inductance that you want to obtain at your frequency of interest. Although the initial geometries will be synthesized for inductance at DC, there will be an automatic fine tuning later that adjusts the inductance for your design frequency.

The geometry limitations specify what layout you want to build, or what the process related limits<sup>3</sup> are. You can specify a range for the number of turns, for the conductor width and the gap size.

For the width, the step size can also be defined as “automatic”. This will generate four to five log-spaced width values between the specified minimum value and the maximum possible width for this parameter combination. Automatic step size is only available for width, all other parameters require a numeric input.

Of course, you also need to specify the maximum inductor diameter that is acceptable in your design. Large single turn inductors usually perform better than small multi-turn inductors, but they are expensive in terms of die area, so you will need to set your design’s limit here.

<sup>3</sup> Values are checked against the limits in the inductor toolkit’s technology.ael file.

Finally, for inductors with patterned ground shield, there is a hidden setting “w\_shield” for shield width. This can be found on the “equations” tab of the input form. Don’t use too small value, because Q factor is surprisingly insensitive to the width of the ground shield strips, and inductors with wider shield strips simulate much faster.

**At the bottom of the input form, you see the field “Number of synthesized geometries” which gives feedback how many inductors conform with your requirements.** A good number is around 5-10 inductors, which covers a reasonable range of geometries. It is usually not necessary to set values which lead to 20 or more inductors for simulation, which can take a long to simulation all the geometries.

There is no need to have very fine steps in the width or spacing. For conductor width, the fine tuning step will automatically find the optimum value by interpolation from the batch simulation results. For spacing, you can usually go with a small value, near the minimum design rule value. If the design rule requires min spacing 2.0µm, you can specify 2.1µm to ensure DRC clean layouts.

When you have made your settings, you can close the window. The synthesis data is stored in the background automatically, but your input data is not. If you want to come back later and refine your target value or geometry limitations, you can to save the \*.dds form with your settings. You can save to the standard form (set\_target\_value.dds) which is located in the inductor synthesis PDK directory, or to some other filename in your workspace. Whenever you re-open that \*.dds file, it will automatically store new synthesis parameters, and overwrite previous parameters.

### ***Step 3: Sweep parameter combinations***

In this step, different geometry implementations are EM simulated. They all conform to the specifications made in step 1, and based on the EM simulations, the best geometry is selected. In this case, “best” is selected based on the Q factor at the target frequency.

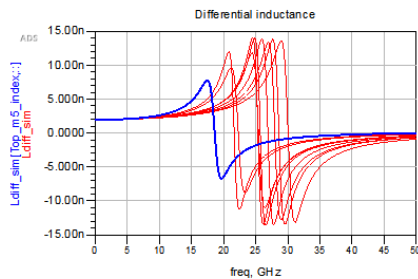
**If you run this for the first time, Momentum might<sup>4</sup> need to calculate the substrate functions, which takes some time. This is only needed once, and the pre-computed substrate file is reused my Momentum for future simulations.**

When the EM simulation is finished, a data display template is opened that shows inductor parameters of the simulation results. The **synthesis target frequency** (design frequency, frequency of interest) is set with the slider at the top of the page. Notice that the “best” inductor will change, depending on the frequency that you have selected. The plots show the differential<sup>5</sup> inductance, differential Q factor and differential resistance for all simulated geometries, with the “best” inductor results highlighted. The geometry value for the best inductor can be found on the bottom right, along with the differential L,Q,R values at the frequency of interest.

---

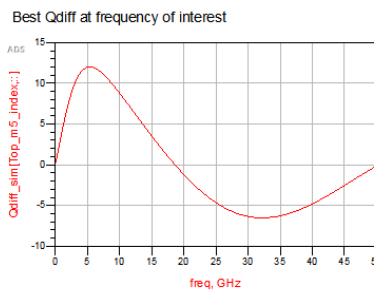
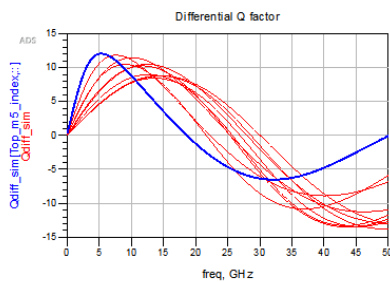
<sup>4</sup> Pre-calculated substrate functions are included in the PDK, and will be copied to the workspace if there is no substrate directory in the workspace. This check is done when the Inductor Toolkit PDK is loaded, and when a workspace is opened that uses Inductor Toolkit. However, if the substrate directory already exists in the workspace, that directory is not changed and no files are copied.

<sup>5</sup> Differential values are obtained for differential excitation. This is the typical use case. These numbers are different from the single ended results (measured into one port, with the other port grounded).



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.



Layout

Qdiff=  at  GHz

Ldiff(nH)=  R DC (Ohm)=

N=

Do(μm)=  Di(μm)=

w(μm)=

s(μm)=

☐

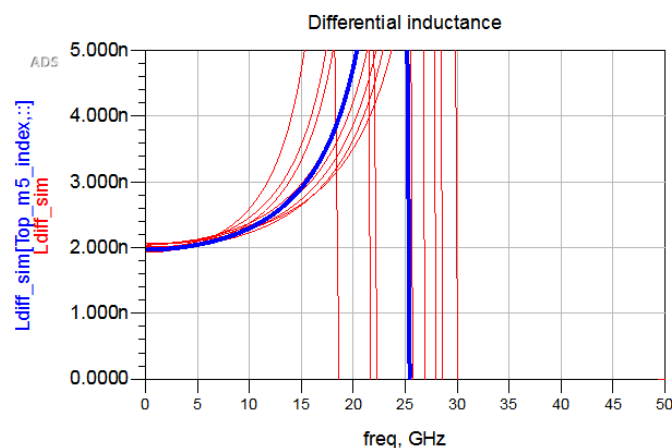
Top list sorted by Q factor

Q_sorted	N_sorted	dout_sorted	w_sorted	s_sorted
8.561	2.000	386.960	30.000	5.000
6.566	2.000	330.960	19.170	5.000
4.761	3.000	218.960	12.970	5.000
4.650	3.000	261.970	12.250	5.000
3.514	3.000	154.120	5.050	5.000
3.343	2.000	263.960	7.530	5.000
2.305	4.000	149.550	6.000	5.000
2.400	4.000	138.450	5.000	5.000
2.363	3.000	162.210	5.000	5.000
2.315	4.000	244.960	5.000	5.000

We can see that for 10 GHz, the inductor with a medium metal width performs best, giving a compromise between conductor losses and substrate losses.

Note that all inductor values are close to the 2.0 nH target at DC, but then value increases with frequency, depending on the actual geometry and resulting self resonance. At the target frequency 10GHz, the top inductor with highest Q factor has 2.36 nH.

In the next step, the inductance for this geometry is fine tuned to have 2.0 nH at the target frequency (instead of DC).



Notice also the "top list" of inductors at the bottom of the page, listing the geometry parameters of the inductors in order of Q factor. This might be useful to identify implementations that are almost as good as the best candidate, but somehow preferred for



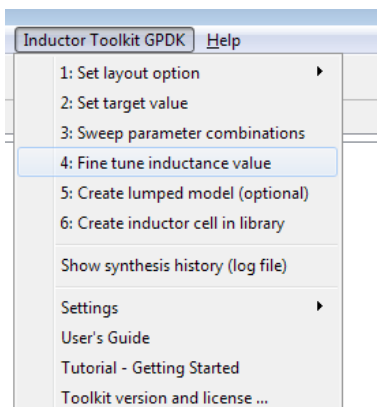
smaller diameter or other reasons. It also gives a good idea for trends if you want to refine the geometry sweep parameters.

Top list sorted by Q factor

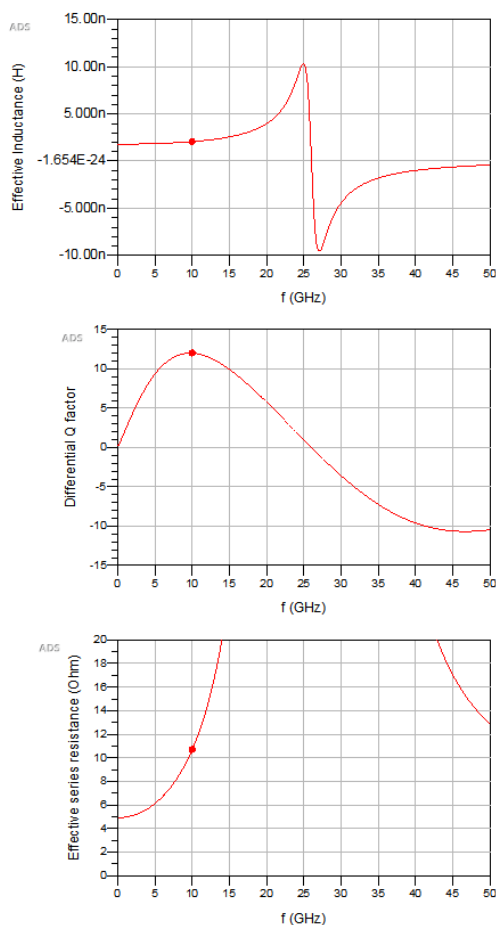
Q_sorted	N_sorted	dout_sorted	w_sorted	s_sorted
11.463	2.000	291.370	12.250	5.000
11.121	2.000	330.860	19.170	5.000
10.362	3.000	218.090	12.970	5.000
10.039	2.000	263.860	7.830	5.000
10.006	3.000	184.720	8.050	5.000
8.805	2.000	386.960	30.000	5.000
8.538	4.000	146.550	6.000	5.000
7.868	3.000	162.210	5.000	5.000
7.756	2.000	244.960	5.000	5.000
7.738	4.000	138.450	5.000	5.000

#### Step 4: Fine tune inductance value

After sweeping the geometries, the best performing inductor was found, but it still needs some tweaking to reach the exact inductance value at target frequency. This is now done in step 4.



The data display from step 3 has automatically evaluated results for different widths, and calculated dimensions for that fine tune layout. When we choose "Fine tune inductance value" from the menu, that tweaked layout is simulated and results are shown in a new data display window.



#### Electrical parameters of synthesized inductor:

$L_{diff}(nH) = 2.05$   
 $Q_{diff} = 12.03$   
 $R_{RF} (Ohm) = 10.71$   
 $R_{DC} (Ohm) = 4.90$   
 Estimated SRF  $26.1$  GHz for differential mode

at 10.00 GHz

at 100.0 MHz

Optional: You can repeat this finetune step multiple times, if the inductance is not close enough to your target value.

#### Layout parameters of synthesized inductor:

Layout option

Number of turns N=

Outer diameter  $D_o(\mu m) = 289.10$  Inner diameter  $D_i(\mu m) = 219.26$

width( $\mu m$ )=

spacing( $\mu m$ )=

The inductance was automatically adjusted by changing the diameter from the initial 291 $\mu m$  to 289 $\mu m$ , with a an optimized width of 15 $\mu m$  instead of 12.2 $\mu m$ . With that change, the inductance is now very close to the target value, at the target frequency.

For frequencies that are too close to the self resonance frequency, where the inductance changes rapidly with frequency, the fine tune step might have difficulties to exactly match the target inductance, but it should be reasonably close.

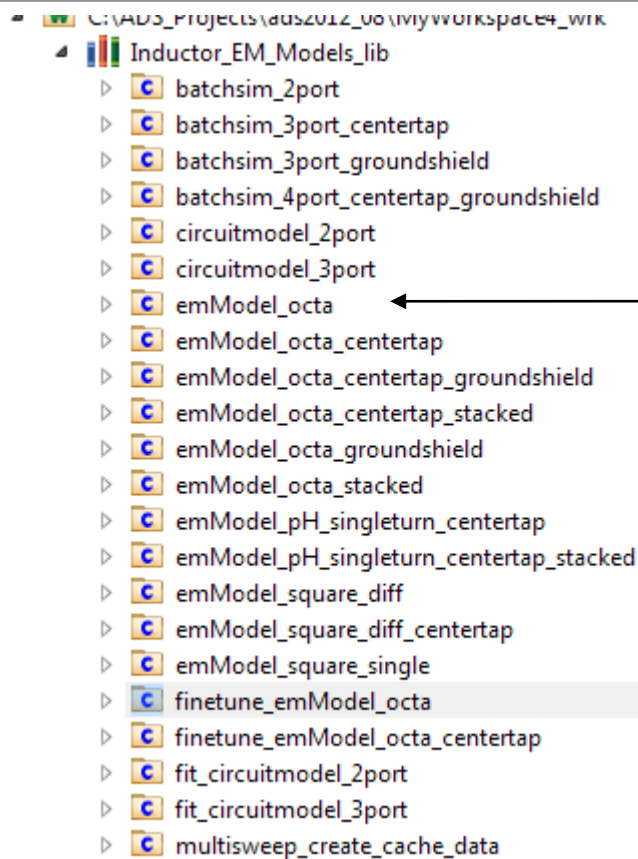
If needed, the fine tune step can be repeated, to iterate closer to the target inductance.

If you wish, you can define separate EM simulation settings for the fine tune step, to make this more accurate than the initial batch simulation. This is optional, and is done by defining a cell with a specific name prefix. By default<sup>6</sup>, that name prefix is "finetune".

Example: if the cell with the regular emModel is named "emModel\_octa", then the name of the optional finetune cell is "finetune\_emModel\_octa". Inductor Toolkit will check for that cell name, and use the cell if it exists. Otherwise, it will use the same cell that was used for batch sweep.

<sup>6</sup> The prefix that identifies optional separate cells for the fine tune step can be configured in the preference file: <Inductor Toolkit PDK directory>/ de/acl/ preferences.acl





Please note that by default, the optional finetune cell is only defined for a few geometries, to show you the method. If you wish to use these optional finetune cells, please create them by copying and renaming the regular emModel cells, and then change the EM mesh settings according to your preferences. Some guidelines can be found in chapter "[emModel/emSetup settings that determine the EM accuracy](#)" at the end of this document.

Up to now, all simulations were done with parameterized emModels. To create a new "static" layout cell for this optimized inductor, we can proceed to step 6: Create inductor cell in library.

Only if we need an equivalent circuit model (ADS schematic and Spectre netlist), we need to do the optional step 5 "Create lumped model (optional)"

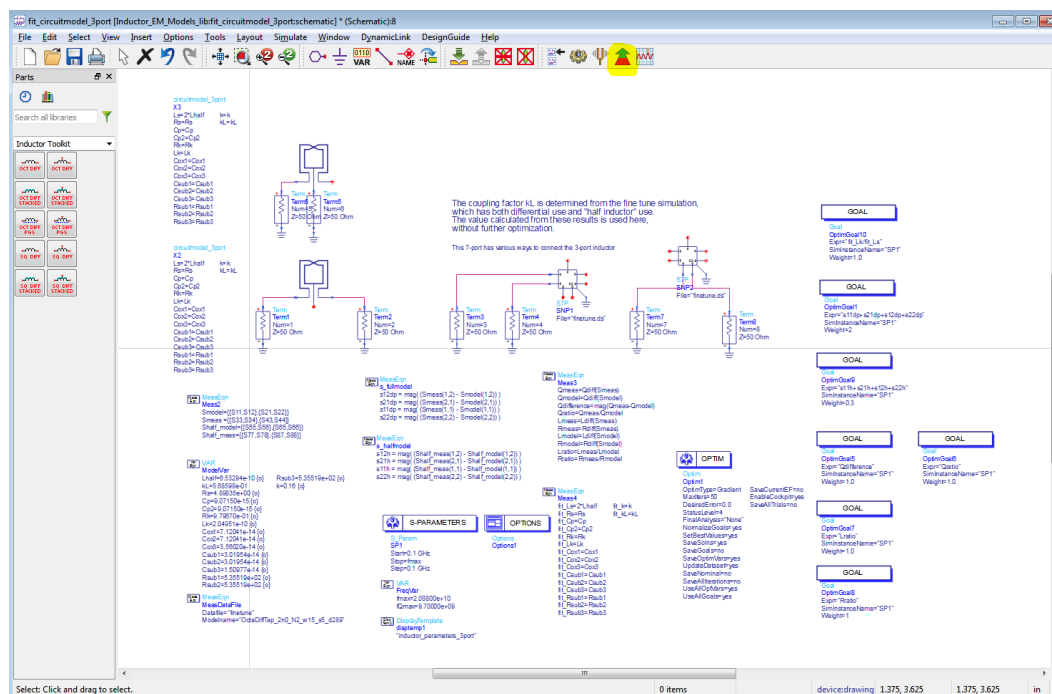
## Step 5: Create lumped model

After calculating the “best” dimensions, we might want to have an equivalent circuit model for use in external simulators, including a Spectre netlist (\*.scs file).

This optional step is described here. Note that lumped model creation is not yet supported for inductors with patterned ground shield.

When you click on “Create lumped model”, a schematic for optimization will open. The schematic is set up for ADS optimization, to fit an equivalent circuit model to the EM simulated data. You don’t need to make any changes, the various fit parameters and goals are already configured for your inductor.

**If optimization does not start automatically**, please start it manually by clicking on the optimizer button or choosing “Simulate > Optimize” from the menu.<sup>7</sup>

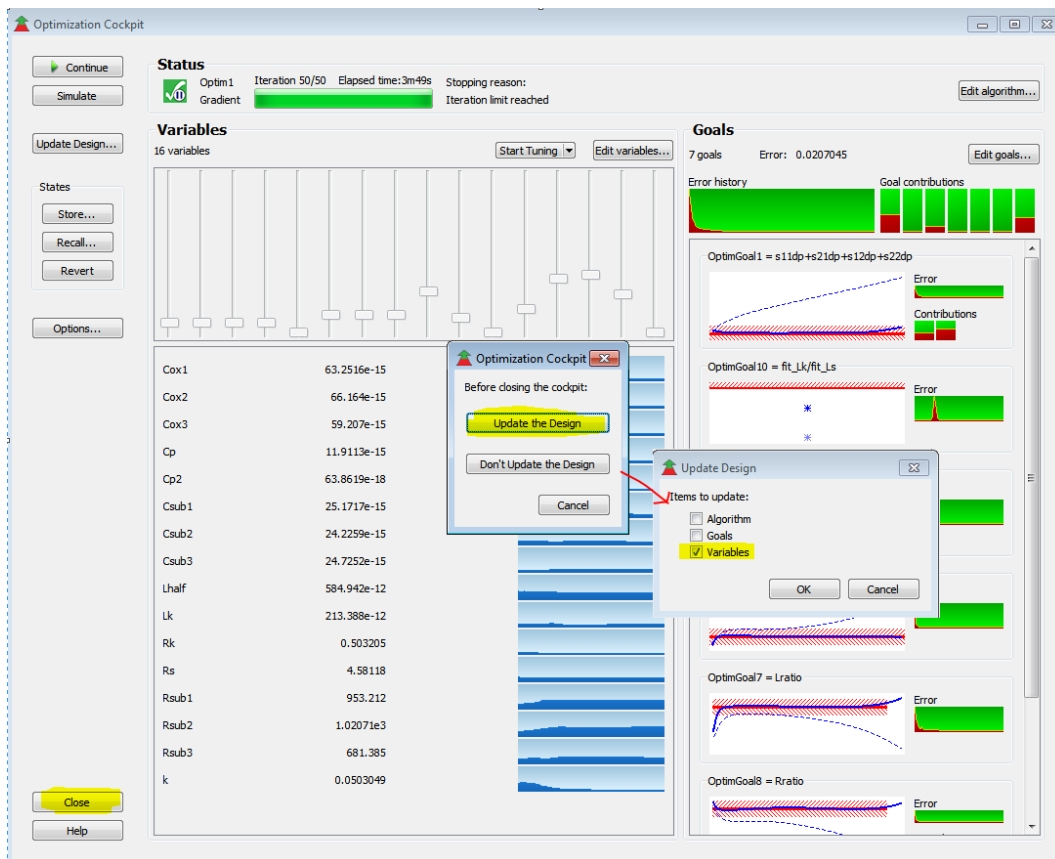


**When the optimization has finished, close the optimizer cockpit and update the schematic with the optimized values<sup>8</sup>.**

After a few seconds of postprocessing time, ADS will bring up a data display that shows two curves for each parameter: the EM simulated data and the result of the equivalent circuit model fit. If you are satisfied with the agreement, no further action is needed. Otherwise, go back to the optimized and try to find a better fit.

<sup>7</sup> Optimization will start automatically if *lumpedmodel\_start\_optimizer* is set to *TRUE* in the inductor toolkit *preferences.ael* file. This file is located at <Inductor Toolkit PDK directory>/ *de/ael/preferences.ael*

<sup>8</sup> 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.



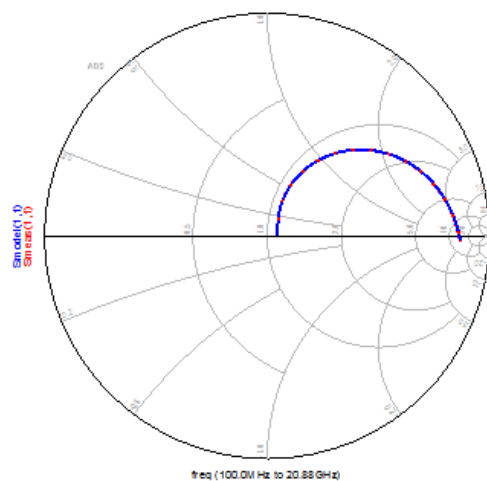
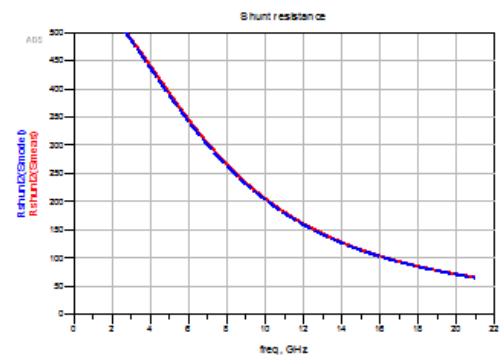
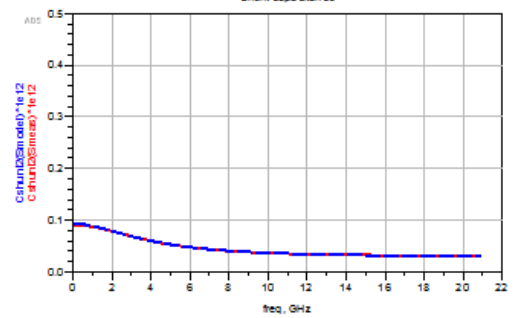
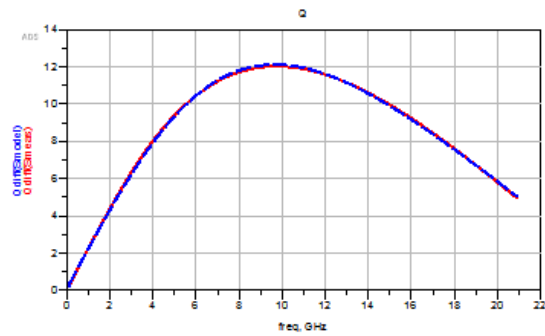
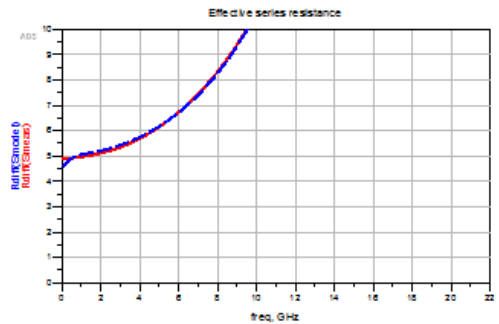
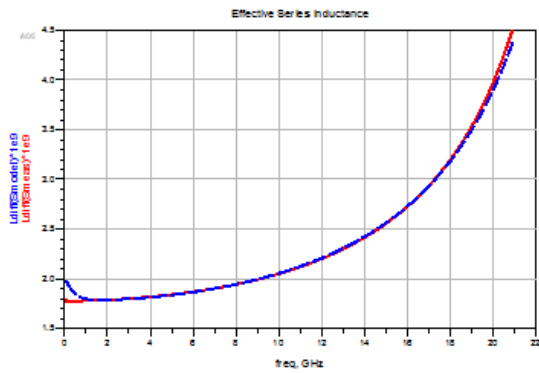
Plots show the comparison for differential L,Q,R and shunt capacitance and shunt esistance and single ended S11. The equivalent circuit model parameters are found in the bottom right table.

The data display is configured to **automatically write an \*.scs netlist for Spectre to the workspaces "data" directory**, using these values.

The values are also used to create an **additional** ADS schematic view in step 6, Create inductor cell in library.

**Important: When you start "Create lumped model" from the menu, this will reset all parameters to the starting values, and overwrite previous optimization results.** If you want to re-open the schematic without resetting the values, open it from the ADS file browser and not from the Inductor Toolkit menu!

red = data from input file  
blue with index m = model = fitted data



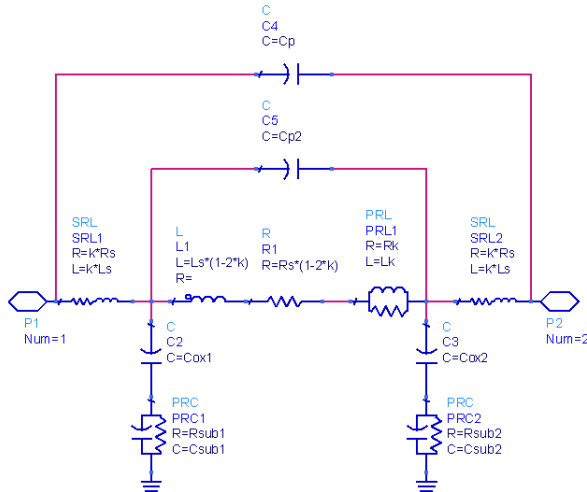
Modelname
OctaDiffTap_2n0_N2_w15_s5_d289

Eqn scs=Muehlhaus\_WriteNetlist\_3Port(Modelname, fit\_Ls, fit\_kL, fit\_Rs, fit\_Cp, fit\_Cp2, fit\_Rk, fit\_Lk, fit\_Coxl

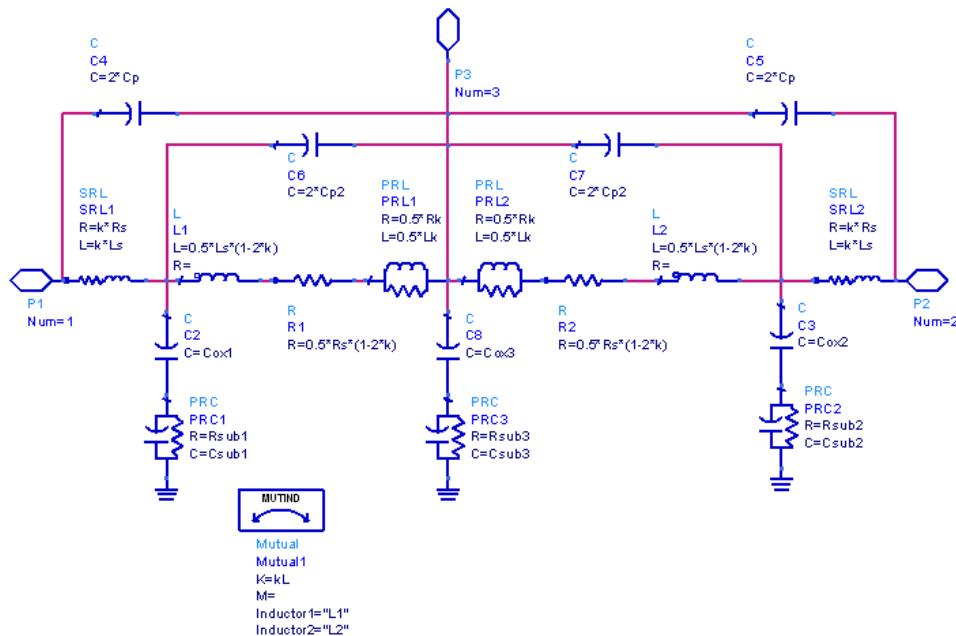
## Lumped model topologies

There are two equivalent circuit model types, with and without center tap. In this release, equivalent circuit models are not yet available for patterned ground shield inductors.

Inductor without center tap (2 ports):



Inductor with center tap (3 ports):

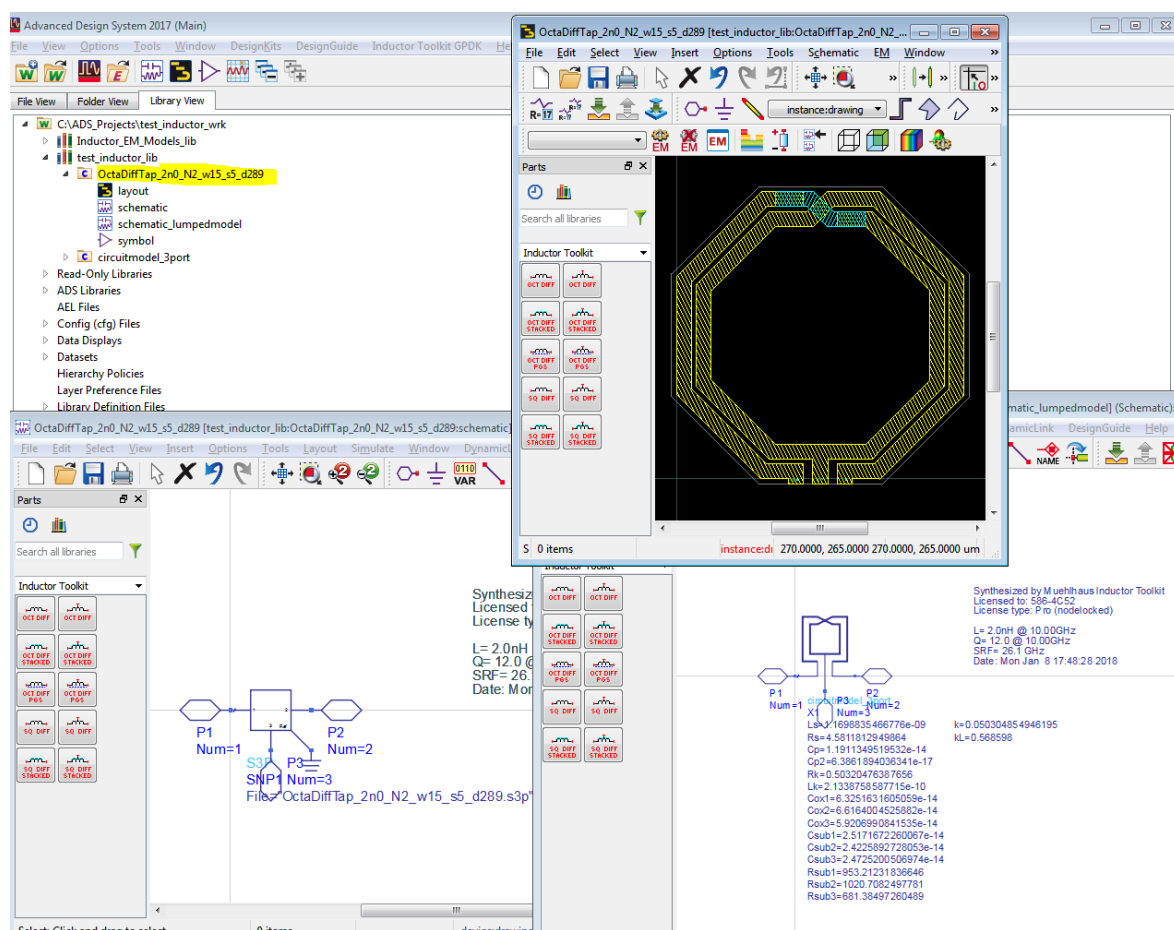


## Step 6: Create inductor cell in library

After the optimized geometry parameters are calculated, we can create a library cell with these inductor parameters. From the menu, choose “Create layout cell”. This will identify the inductor layout that was specified in step 1, the dimensions calculated in step 4 and (optional) the equivalent circuit model parameters, and create a new cell.

The cell has **layout view** (instance of layout Pcell with the calculated “best” parameters), **schematic view** (pointing to the S-parameter results) and **symbol view**.

If the (optional) [equivalent circuit model](#) was fitted in step 5, the cell also includes a second schematic view (schematic\_lumpedmodel) with an instance of the equivalent circuit model and the fitted parameters. This can be used for ADS feature Dynamic Model Selection<sup>9</sup>.



This cell will be created in the **first writable library of the workspace** (excluding the inductor toolkit libraries). The name of the cell tells about the geometry parameters. If you look at the Pcell properties, you will notice that an inductance value is reported there. That approximate value is calculated inside the Pcell with textbook equations, and can be ignored because we have already EM-simulated the accurate value.

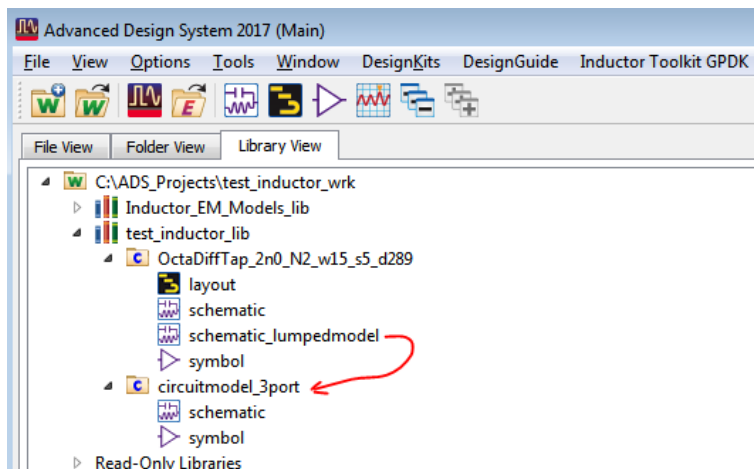
<sup>9</sup> If you place the symbol for this cell on a schematic, the simulation will default to the schematic view, which in our case holds the S-parameters. You can switch the simulation to use the schematic\_lumpedmodel view by right clicking on an instance on the schematic, and select *Component > Choose View for Simulation...* from the right mouse button menu. See ADS help topic “Dynamic Model Selection”.

## Using the new inductor cell in your ADS designs

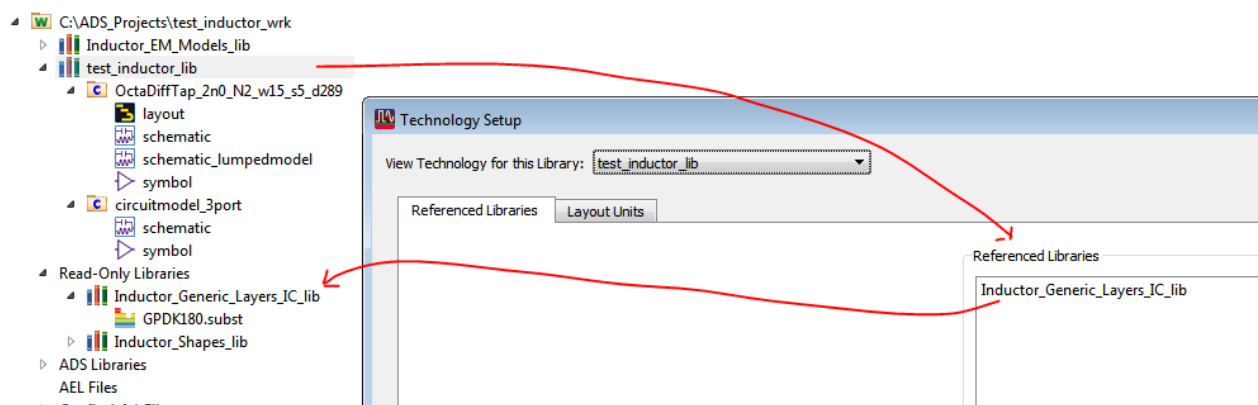
In step 6, an inductor cell was created in your workspace library.

After creating the inductor cell, do we need the Inductor Toolkit library to use that cell?  
The answer is: NO.

The layout is flattened to polygons, so that it no longer references the Inductor Toolkit layout libraries. If you have created an equivalent circuit model for the inductor, the required circuit model schematic (circuitmodel\_2port or circuitmodel\_3port) is copied to your library. So for all this, no reference to the Inductor Toolkit libraries is needed.



If you want to run your own EM simulations on this layout, you can easily do this if the Inductor Toolkit technology library (or if you used another tech library during synthesis, that tech library) is referenced in your workspace library.

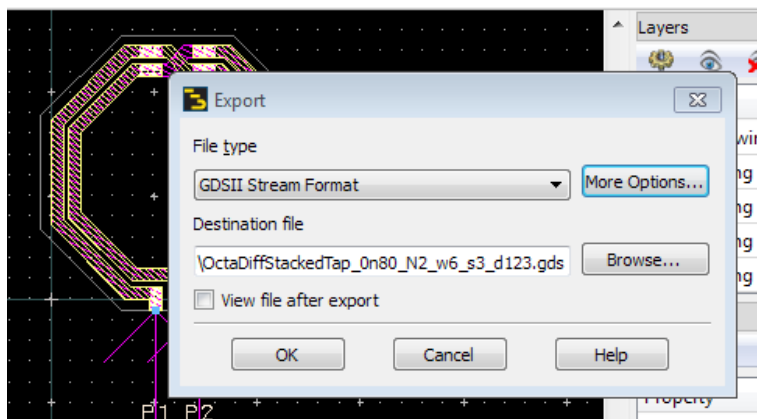


## Using the new inductor in Cadence

To use the synthesized inductor in Cadence, you need to take the layout and the Spectre model to Cadence. This is described below.

### **Import inductor layout to Cadence**

Open the layout view of inductor cell that you created in step6. In the ADS layout window, select *File > Export* and create the GDSII (\*.gds) layout file. By default, this is stored to the “mfg” directory in your workspace.



Move that GDSII file to a location where you can access it from Cadence, and import it into a new layout cell. This is your layout view in Cadence.

### **Schematic view for simulation in Cadence**

Besides layout, we also need the equivalent circuit model \*.scs for the inductor. Inductor Toolkit creates this \*.scs file in the “data” directory of your workspace, with the same base name as the layout. Copy this to a location where you can access it from Cadence.

To access the \*.scs model file from your schematic, you need a schematic symbol that references the \*.scs model. This is described below.

#### **Step 1: Symbol view**

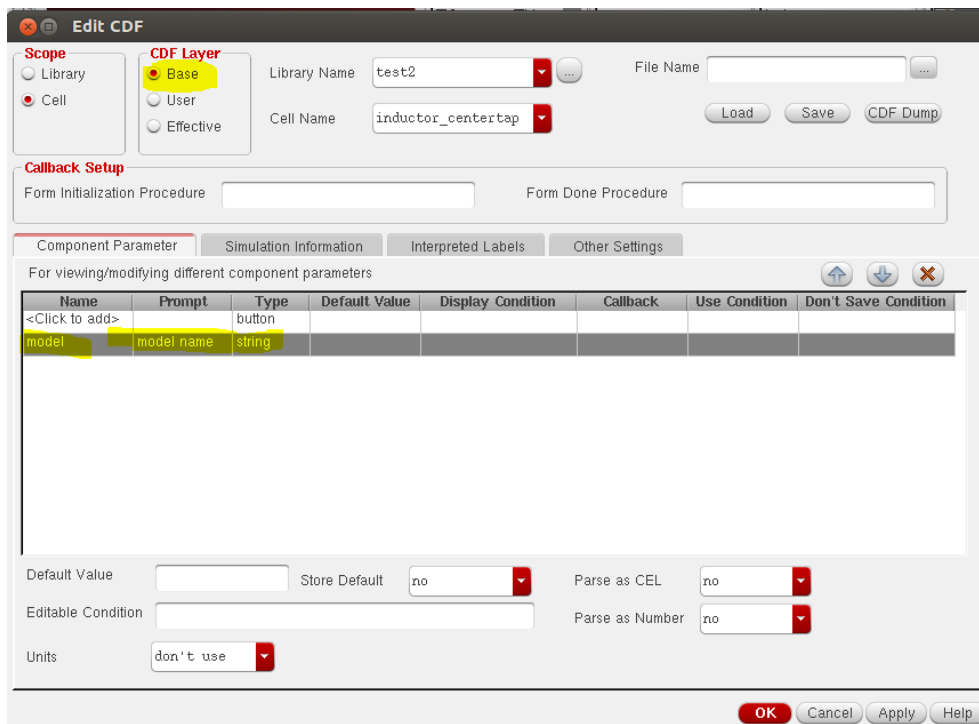
Create a symbol view in the same cell where you have imported the \*.GDSII layout, and draw the symbol graphics with the required number of pins. For the \*.scs models from Muehlhaus Inductor Toolkit, the symbol needs 3 pins (2 inductor terminals + ground reference) for the regular inductors and 4 pins (2 inductor terminals + centertap + ground reference) for the inductor with center tap.

#### **Step 2: Spectre view**

Make a copy of the symbol that you have created, and call the new view “spectre” instead of “symbol”.

Start the CDF editor from the Cadence main window (CIW) and open the cell that you have created. Open the base CDF and add a component parameter **model** as a string parameter, with the settings shown below:





**Edit CDF**

**Scope**  
☐ Library  
☒ Cell

**CDF Layer**  
☒ Base  
☐ User  
☐ Effective

Library Name: test2  
 Cell Name: inductor\_centertap

File Name:   
 Load Save CDF Dump

**Callback Setup**  
 Form Initialization Procedure:   
 Form Done Procedure:

**Component Parameter** | Simulation Information | Interpreted Labels | Other Settings

For viewing/modifying different component parameters

Name	Prompt	Type	Default Value	Display Condition	Callback	Use Condition	Don't Save Condition
<Click to add>		button					
model	model name	string					

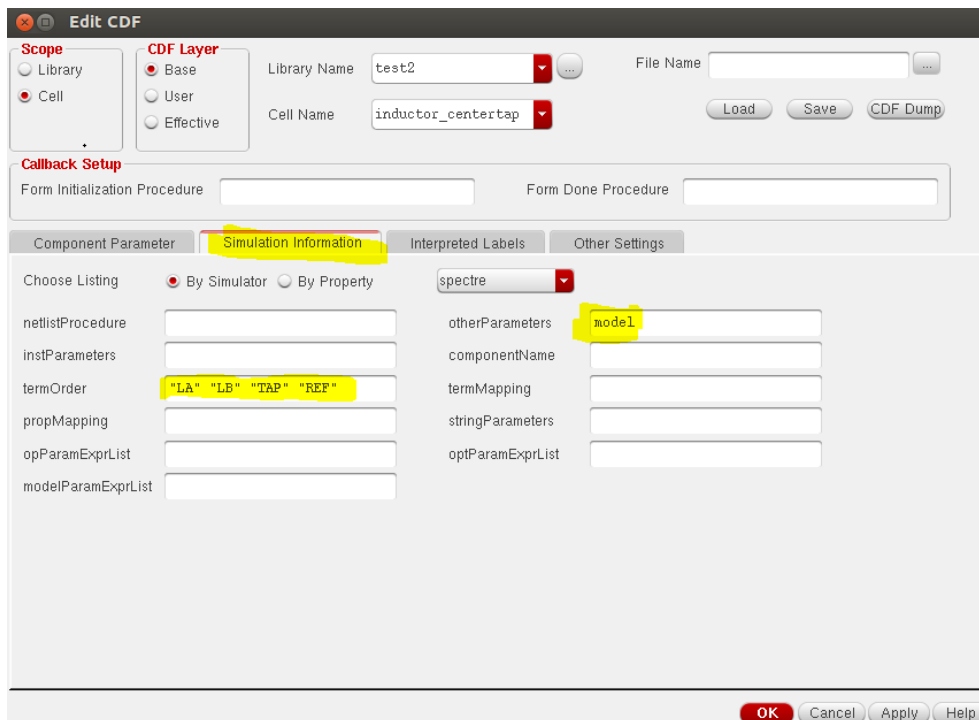
Default Value:   
 Store Default: no  
 Parse as CEL: no  
 Editable Condition:   
 Parse as Number: no  
 Units: don't use

OK Cancel Apply Help

Click "Apply" and change to the "Simulation Information" tab of the dialog.

In the **termOrder** field, enter the pin names (as defined in the symbol). The pin names must be enclosed in "quotes".

Set the **otherParameters** field to: model



**Edit CDF**

**Scope**  
☐ Library  
☒ Cell

**CDF Layer**  
☒ Base  
☐ User  
☐ Effective

Library Name: test2  
 Cell Name: inductor\_centertap

File Name:   
 Load Save CDF Dump

**Callback Setup**  
 Form Initialization Procedure:   
 Form Done Procedure:

**Component Parameter** | **Simulation Information** | Interpreted Labels | Other Settings

Choose Listing: ☒ By Simulator ☐ By Property  
 spectre

netlistProcedure:   
 instParameters:   
 termOrder: "LA" "LB" "TAP" "REF"  
 propMapping:   
 opParamExprList:   
 modelParamExprList:

otherParameters: model  
 componentName:   
 termMapping:   
 stringParameters:   
 optParamExprList:

OK Cancel Apply Help

Click "OK" to save your settings.

You have now created a symbol with one parameter ("model"). When you place this symbol on the schematic, set the value of parameter "model" to the name of the actual name of the model subcircuit from your \*.scs file.

**This is the name of the model subcircuit, not the filename.** The difference is that the filename has suffix ".scs" but the subcircuit model name has not.

This will netlist your new symbol with the model name as set in the symbol parameter "model".

To simplify your work, we can also provide a filesset with these two cells, which you can copy into your library. Please contact [support@muehlhaus.com](mailto:support@muehlhaus.com).

### **Step 3: Add \*.scs file to ADE library path**

Finally, for simulation with Spectre you need to add the path to the \*.scs netlist file in ADE, using the ADE menu command Setup > Model Libraries. That's all, you are ready to simulate now.

## Layout Options - Inductor Pcells

The Inductor Toolkit includes a set of layout cells in library Inductor\_Shapes\_lib, which are based on AEL code. By default, these layout cells are used for inductor synthesis, but with some customization, other layout cells could be used as well. Drawing layers are determined by the technology setup in Inductor Toolkit's PDK directory

<ToolkitDir>/configuration/technology.ael

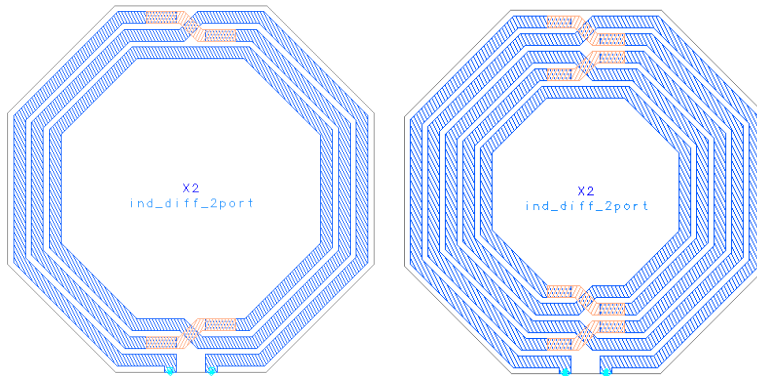
### **Octagon differential (ind\_diff\_2port)**

This is a symmetric octagon inductor with 2 pins/ports, without center tap. The number of turns can be set in full turns.

Parameters: N=number of turns, D=outer diameter, w=width, s=spacing.

Metal layers used: spiral\_layer, crossover\_layer

Via layers used: via\_layer\_spiral\_crossover



### **Octagon differential centertap (ind\_diff\_3port)**

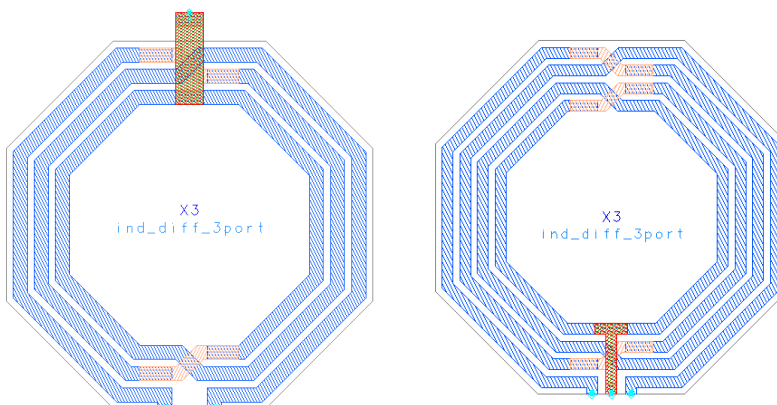
This is a symmetric octagon inductor with 3 pins/ports, where port 3 is the center tap. The number of turns can be set in full turns.

For even number of turns, the center tap is on the same side as the inductor terminals. For odd number of turns, it is on the opposite side.

Parameters: N=number of turns, D=outer diameter, w=width, s=spacing.

Metal layers used: spiral\_layer, crossover\_layer, underpass\_layer, underpass2\_layer

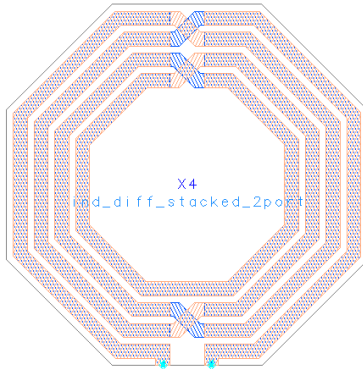
Via layers used: via\_layer\_spiral\_crossover, via\_layer\_crossover\_underpass, via\_layer\_underpass\_underpass2



### ***Octagon differential stacked (ind\_diff\_stacked\_2port)***

This is a symmetric octagon inductor, similar to ind\_diff\_2port, but with stacked metal on two layers.

Parameters: N=number of turns, D=outer diameter, w=width, s=spacing.  
Metal layers used: spiral\_layer, crossover\_layer  
Via layers used: via\_layer\_spiral\_crossover

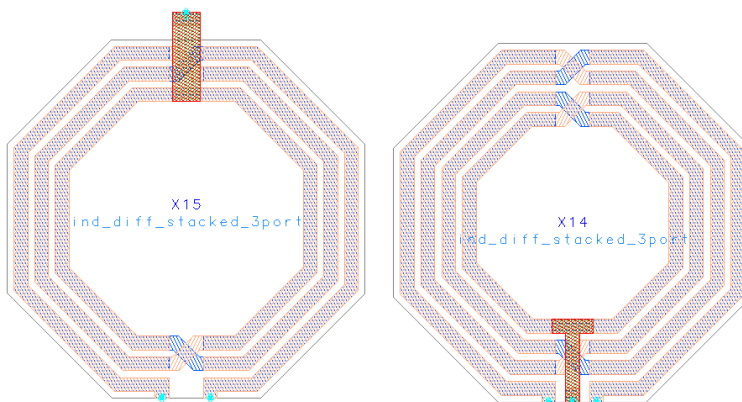


### ***Octagon differential stacked centertap (ind\_diff\_stacked\_3port)***

This is a symmetric octagon inductor, similar to ind\_diff\_3port, but with stacked metal on two layers.

For even number of turns, the center tap is on the same side as the inductor terminals. For odd number of turns, it is on the opposite side.

Parameters: N=number of turns, D=outer diameter, w=width, s=spacing.  
Metal layers used: spiral\_layer, crossover\_layer, underpass\_layer, underpass2\_layer  
Via layers used: via\_layer\_spiral\_crossover, via\_layer\_crossover\_underpass, via\_layer\_underpass\_underpass2

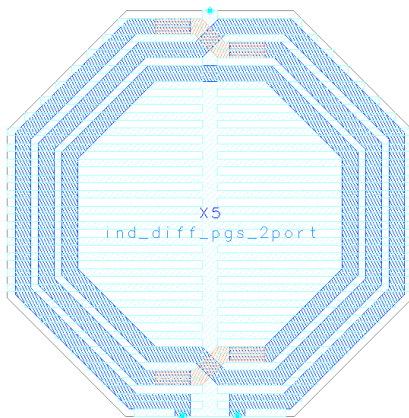


### ***Octagon differential groundshield (ind\_diff\_pgs\_2port)***

This is a symmetric octagon inductor with 2 pins/ports, without center tap. The number of turns can be set in full turns.

There is a patterned ground shield under the inductor, connected through an additional pin. The width of the ground shield lines/gaps can be set as a parameter. The inductor performance is not very sensitive to that width, so it is recommend to use larger values that simulate much faster ground shields with many narrow thin lines and narrow slots.

For synthesis, the ground shield width is fixed, and not part of the optimization.



Parameters: N=number of turns, D=outer diameter, w=width, s=spacing, wshield  
Metal layers used: spiral\_layer, crossover\_layer  
Shield: pgs\_layer  
Via layers used: via\_layer\_spiral\_crossover

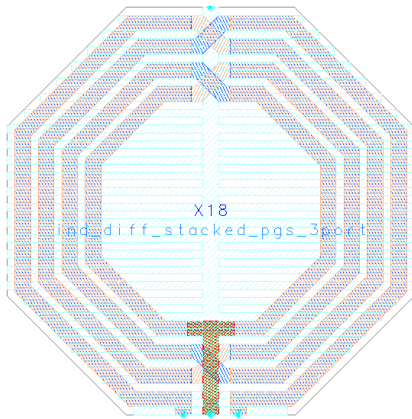
The metal and shield layers are defined in the Inductor Toolkit technology setup.

### ***Octagon differential groundshield centertap (ind\_diff\_pgs\_3port)***

This is a symmetric octagon inductor between pin 1 and 2, with center tap at pin 3. The number of turns can be set in full turns.

There is a patterned ground shield under the inductor, connected through pin 4. The width of the ground shield lines/gaps can be set as a parameter. The inductor performance is not very sensitive to that width, so it is recommend to use larger values that simulate much faster ground shields with many narrow thin lines and narrow slots.

For synthesis, the ground shield width is fixed, and not part of the optimization.



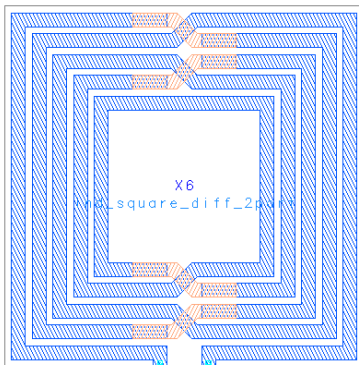
Parameters: N=number of turns, D=outer diameter, w=width, s=spacing, wshield  
Metal layers used: spiral\_layer, crossover\_layer  
Shield: pgs\_layer  
Via layers used: via\_layer\_spiral\_crossover

The metal and shield layers are defined in the Inductor Toolkit technology setup.

### ***Square differential (ind\_square\_diff\_2port)***

This is a symmetric square inductor with 2 pins/ports, without center tap. The number of turns can be set in full turns.

Parameters: N=number of turns, D=outer diameter, w=width, s=spacing.  
Metal layers used: spiral\_layer, crossover\_layer  
Via layers used: via\_layer\_spiral\_crossover

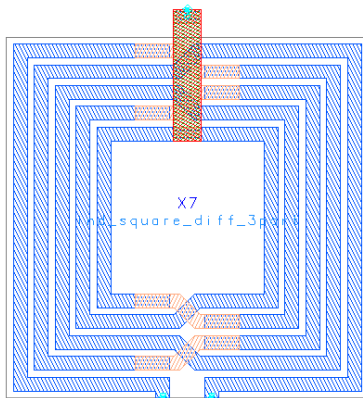


### ***Square differential centertap (ind\_square\_diff\_3port)***

This is a symmetric square inductor with 3 pins/ports, where port 3 is the center tap. The number of turns can be set in full turns. For even number of turns, the center tap is on the same side as the inductor terminals. For odd number of turns, it is on the opposite side.

For even number of turns, the center tap is on the same side as the inductor terminals. For odd number of turns, it is on the opposite side.

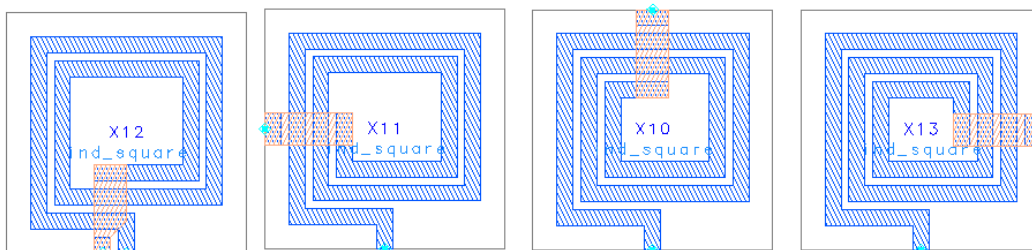
Parameters: N=number of turns, D=outer diameter, w=width, s=spacing.  
Metal layers used: spiral\_layer, crossover\_layer, underpass\_layer, underpass2\_layer  
Via layers used: via\_layer\_spiral\_crossover, via\_layer\_crossover\_underpass, via\_layer\_underpass\_underpass2



### ***Square single ended (ind\_square)***

This is a simple (non-symmetric) square inductor with 2 pins/ports, without center tap. The number of turns can be set in quarter turns, which sets the exit direction.

Parameters: N=number of turns, D=outer diameter, w=width, s=spacing.  
Metal layers used: spiral\_layer, crossover\_layer  
Via layers used: via\_layer\_spiral\_crossover





### ***Single turn (N=1) centertap (ind\_pH\_singleturn\_3port)***

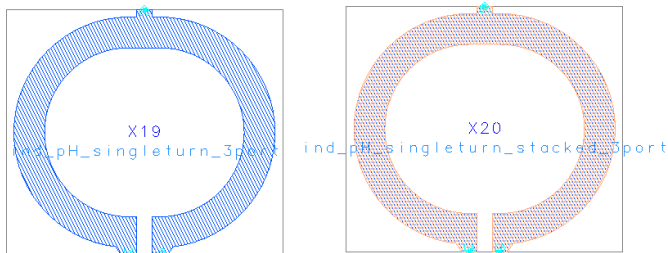
This is a single turn (N=1) symmetric octagon inductor between pin 1 and 2, with center tap at pin 3. It does only support single turn layouts, and parameter N is ignored.

This inductor is designed for small values (<500pH) where single turn inductors perform best, and the other layout shapes might be limited in minimum diameter.

Parameters: N=1 fixed, D=outer diameter, w=width, s=feedline spacing.  
Metal layers used: spiral\_layer, crossover\_layer  
Via layers used: via\_layer\_spiral\_crossover

For this layout, the feedline width is different from the inductor trace width w, so that small inductor value and small inner diameters are possible. The inductor trace width can be set through the cell parameter w, as usual. The feed width is pre-defined in the Inductor Toolkit preference file (see chapter "Preference File") and defaults to 10 µm.

```
decl pH_single_feedline_width = 10; // feedline width for single turn picoHenry inductor
```



### ***Single turn (N=1) stacked centertap (ind\_pH\_singleturn\_stacked\_3port)***

This layout is similar to Single turn (N=1) centertap, but with stacked metal on two metal layers. Usually, these layers are the two topmost layers with thick metal. Compared to the regular octagon, this inductor has less resistance at low frequencies, but an increased capacitance to the substrate, so that the frequency of maximum Q is lower. Use this geometry when the Q factor is limited by ohmic conductor loss, and you want to maximize the conductor cross section.



## Data Display Templates and Expressions

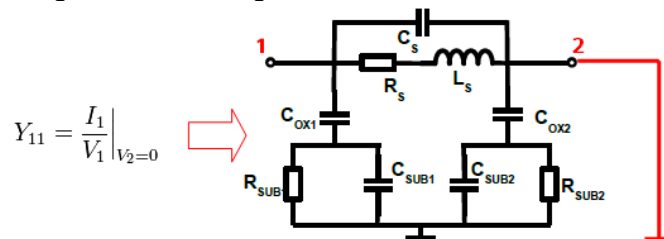
Inductor toolkit includes several custom expressions for plotting inductor parameters. These are described below.

### Single ended and differential operation

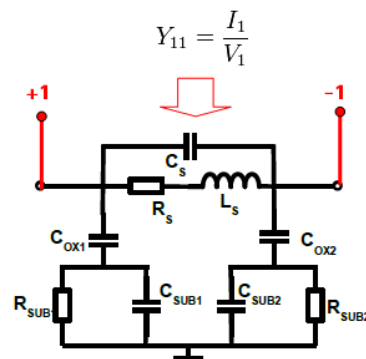
Inductors are simulated with ports on each terminal, so that we have 2-port results for a regular inductor without center tap. From this, we can calculate the results for single ended operation (measured into port 1, with port 2 grounded) or for differential operation (measured differentially between port 1 and 2).

Results depend on this choice, which represents two different ways of using the inductor. The Q factor for single ended configuration is lower, with a lower self resonance frequency (SRF).

Single ended configuration:



Differential configuration (here shown with one differential port).



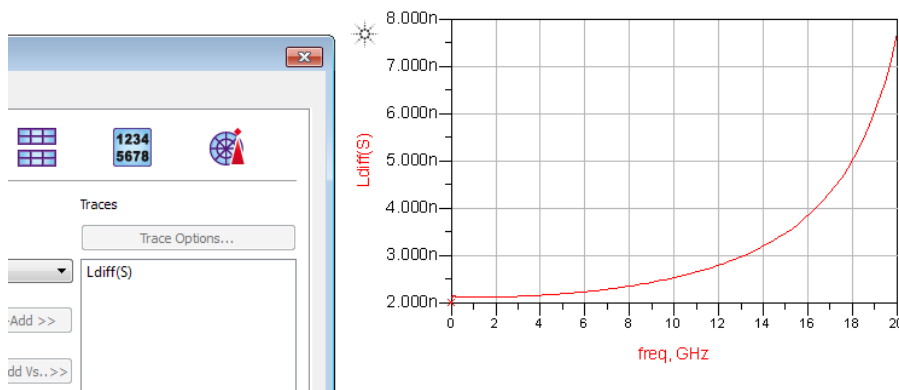
Muehlhaus InductorToolkit comes with equations to calculate both single ended and differential results from the 2-port configuration shown above. You do not need to assign differential ports in the ADS simulations. The differential results are obtained by calculation in the data display.

## Custom inductor expressions for ADS data display

The following custom expressions are included with InductorToolkit:

Expression	Description
<b>Ldiff(S)</b>	Series inductance for differential mode between port 1 and 2, with all other ports open
<b>Qdiff(S)</b>	Q factor for differential mode between port 1 and 2, with all other ports open
<b>Rdiff(S)</b>	Effective series resistance for differential mode between port 1 and 2, with all other ports open
<b>L1port(S)</b>	Series inductance for a 1-port
<b>Q1port(S)</b>	Q factor for a 1-port
<b>Lser(S)</b>	Effective series inductance between port 1 and 2, <u>excluding</u> shunt effects $L_{ser} = \text{imag}((Z(1,1) \cdot Z(2,2) - Z(1,2) \cdot Z(2,1)) / Z(2,1)) / \omega$
<b>Rser(S)</b>	Effective series resistance between port 1 and 2, excluding shunt effects $R_{ser} = \text{real}((Z(1,1) \cdot Z(2,2) - Z(1,2) \cdot Z(2,1)) / Z(2,1))$
<b>Qser(S)</b>	Single ended Q factor from Y11 (with all other ports shorted)
<b>Cshunt2(S)</b>	Shunt capacitance to ground for 2-port data (one side)
<b>Rshunt2(S)</b>	Shunt resistance to ground for 2-port data (one side)

You can use these on your data display forms to plot inductor parameters. Parameter for all equations is the S-parameter dataset that you want to evaluate. To use the default dataset, use "S".



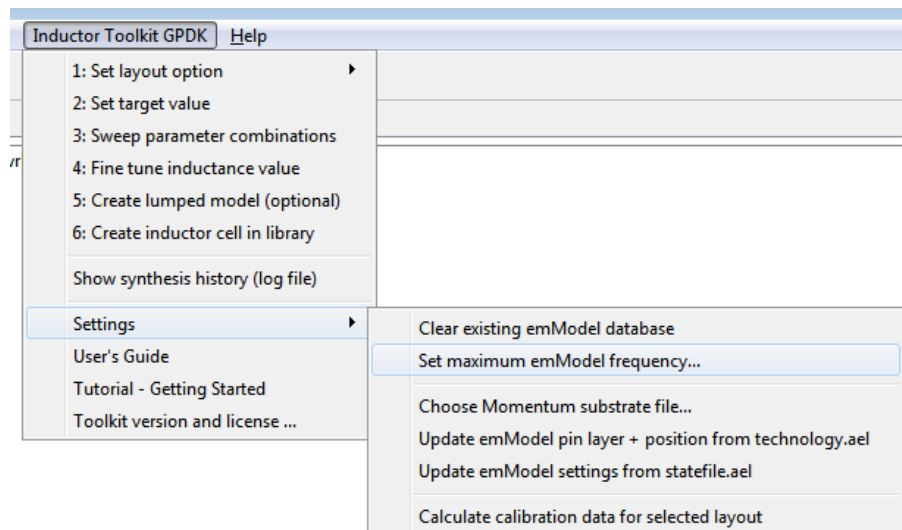
## Configuration and Customization

In this chapter, configuration and customization options are discussed.

### Frequency range

By default, EM simulations in Inductor Toolkit cover the frequency range up to 20 GHz. If you need a wider frequency range, changes must be made to the emModels and to the schematics that use these emModels.

To make that change, use “Settings > Set maximum emModel frequency” from the Inductor Toolkit menu. Note that using that function clears the existing emModel database, so that previous EM simulations are no longer available for reuse. Also note that the *Inductor\_EM\_Models\_lib* library must be writable to make changes.



### Ground ring around inductor

If you want to include a ground ring around the inductor, to imitate some ground ring in your chip layout, you need to change a preference file settings. By default, there is no ground ring around the inductors. To enable it, edit the preference file:

```
<Inductor Toolkit directory>/configuration/preferences.ael
```

and set `draw_groundframe` to `TRUE`. The details of the ground frame can be configured there as well.

```
// ===== LAYOUT OPTIONS =====
```

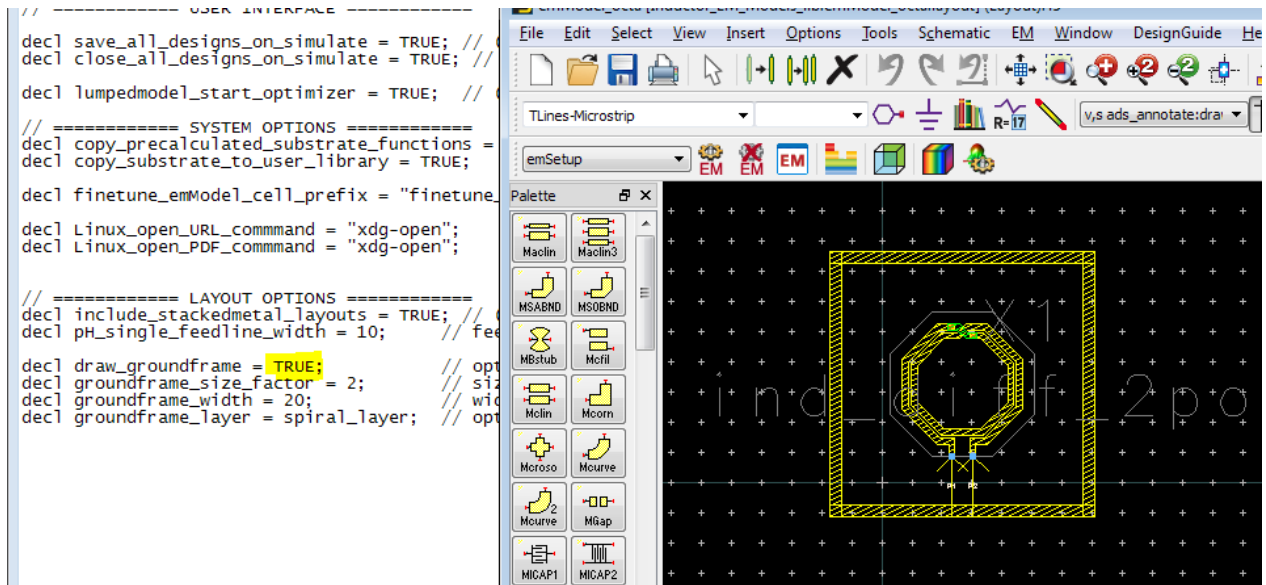
```
decl draw_groundframe = FALSE;           // optional groundframe around inductor
decl groundframe_size_factor = 2;        // size factor of ground frame
decl groundframe_width = 20;             // width of ground frame
decl groundframe_layer = spiral_layer;    // optional ground frame around inductor
```

To see the effect, **reload** the workspace (!) and look at the layouts shown in the emModel cells. Also, you need to manually **clean existing emModel results** after enabling/disabling

the ground ring. This is done in the Inductor Toolkit menu: Settings > Clear existing emModel database.

Note that the ground ring setting is a global setting that affect **all** inductor layouts. Also note that the ground ring will have an impact on L and Q, and the initial pre-EM synthesis math does not include the ground ring.

You will find that the EM-simulated inductance might be slightly less than the initial synthesis value. This offset will be compensated automatically during the fine-tune step, which adjusts inductance based on the actual EM results (with ground ring included).



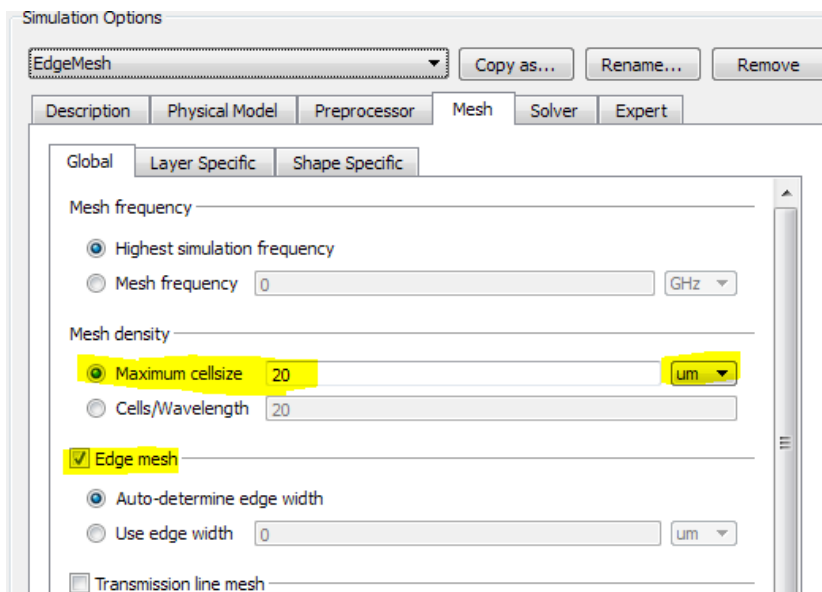
For the `groundframe_layer` parameter, you can use alias names from the technology definition, e.g. `spiral_layer`, or you can specify the ADS layer number where you want the ground frame to be drawn. If you modify the setting, make sure that your layer is included (mapped) in the EM substrate setup.

At the present time, ports can not be placed on the ground ring. If you need that capability, please contact us: [support@muehlhaus.com](mailto:support@muehlhaus.com)

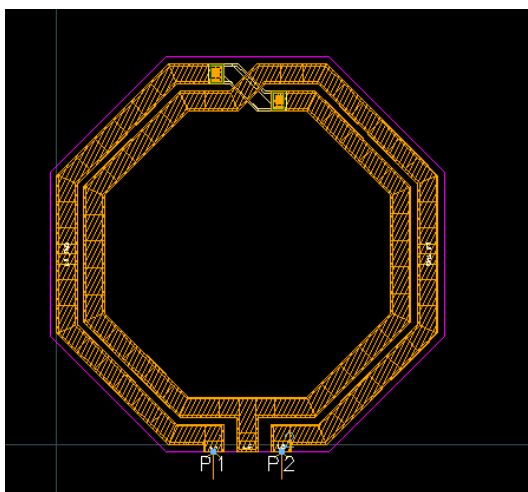
## ***emModel/emSetup settings that determine the EM accuracy***

By default, Inductor Toolkit uses emModels that are optimized for fast simulations. These use the “default” Momentum settings, with no edge mesh and large mesh cells only limited by the 20 cells/wavelength setting. For many cases, these defaults provide reasonable accuracy with short simulation times.

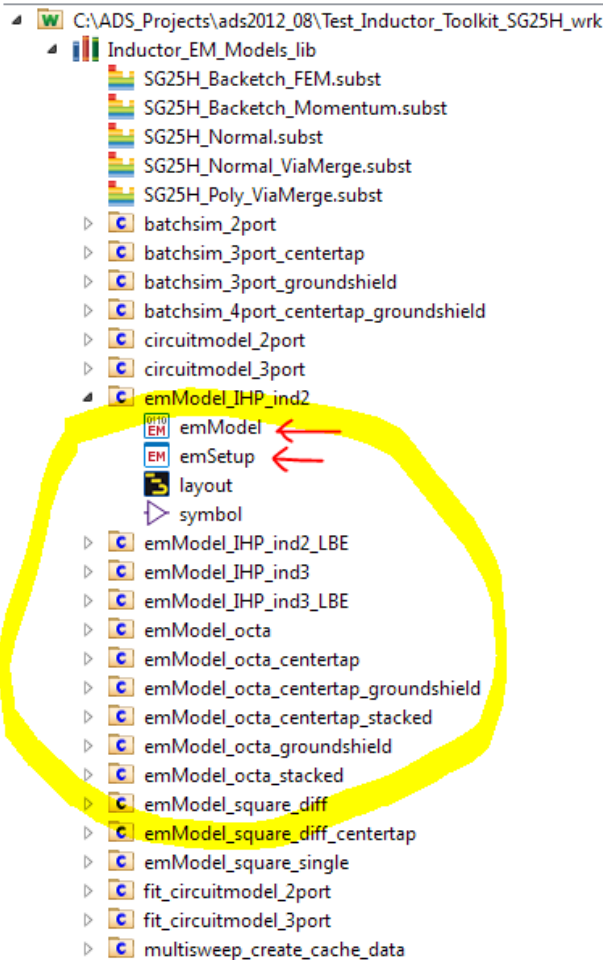
For highest accuracy, you can change the emModel settings to use a finer mesh. From experience, we recommend to **not** just switch to edge mesh, because this might create less accurate results than the default settings. Good results are obtained by **combining** edge mesh with a limit of the maximum mesh size, so that inductor segments are divided into multiple pieces.



For limiting the mesh size, don't use the cells/wavelength setting, and instead specify an absolute max. dimension value like 20µm or 30µm. From experience, this gives very good results for measured-simulated comparisons.



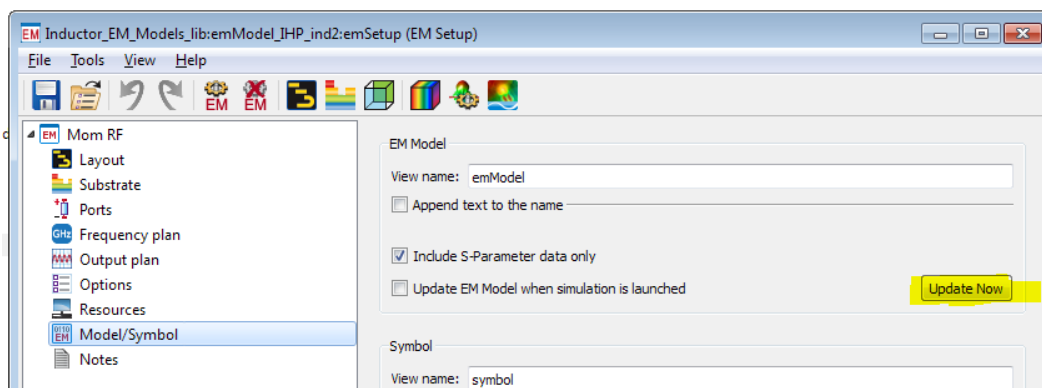
For each inductor layout option, we have a separate cell in the “Inductor\_EM\_Models\_lib” library. When you make these changes, you need to be aware of the **difference between emSetup and emModel**. Inductor Toolkit uses the emModel blocks, which have been created from emSetups.



#### NEW:

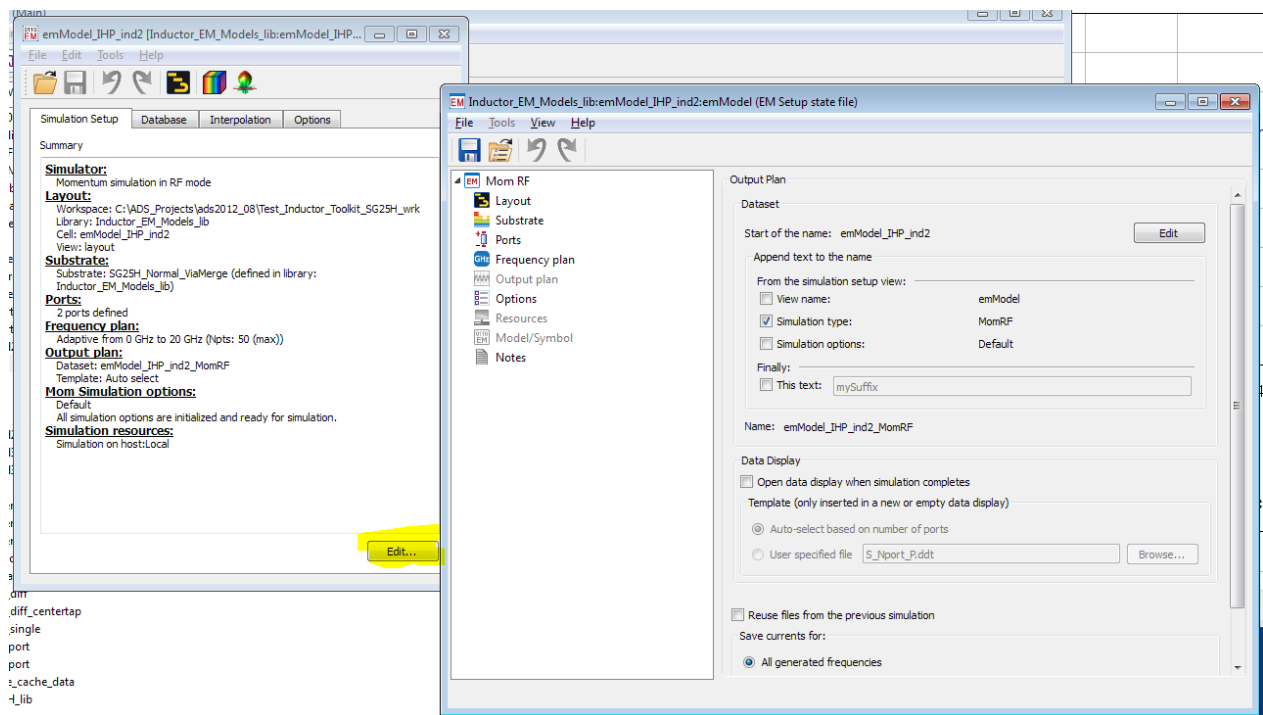
Please note the new feature for using separate EM settings for the fine tune step, which is described in chapter [“Step 4: Fine tune inductance value”](#).

If you make changes to the **emSetup**, the **emModel** must be updated, so that the changes are actually used by Inductor Toolkit. Update manually with the “*Update Now*” button on the Models/Symbol tab.



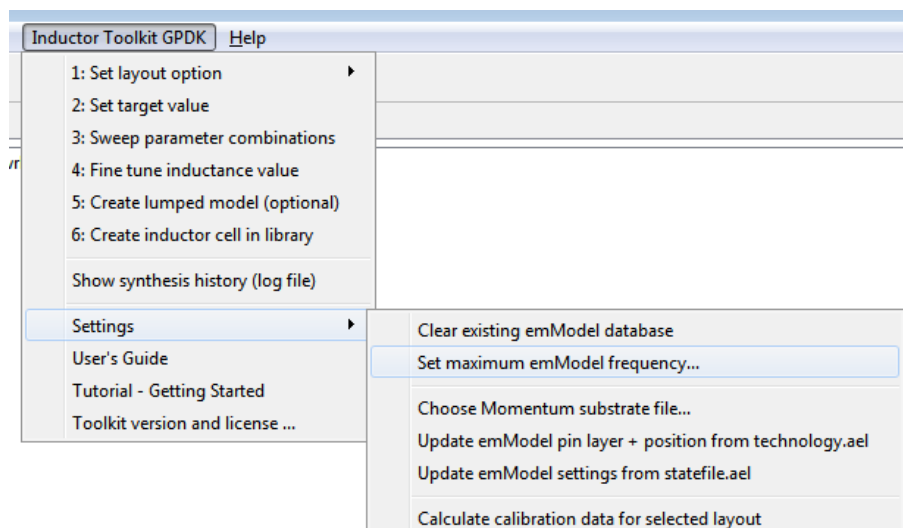
You can make changes directly to the **emModel**, without updating the **emSetup**. That might be slightly confusing if **emSetup** and **emModel** have different settings, but it is possible.

To update the **emModel** directly, open the **emModel** and click on the “*Edit*” button.



Both methods – changing the emSetup and creating an updated emModel from there, or opening the existing emModel and change it directly - are working fine. This is a matter of personal preferences which you prefer.

The Inductor Toolkit “Settings” menu item for changing the frequency range works directly on the emModel settings, and does not go through emSetup. Note that changing the frequency range will delete the emModel data cache, so that previous simulation results are no longer available.



## Technology settings

Inductor Toolkit is factory configured for a technology, including substrate definitions and layer mappings. For Inductor Toolkit GENERIC, the user can modify all technology settings for a different technology

For multiple technologies, you need to create multiple copies of the Inductor Toolkit PDK. Each PDK supports one technology then. Use meaningful directory names to identify the different versions.

### ***Edit Momentum substrate file vs. complete technology configuration***

In the past, we noticed that some users took the free Inductor Toolkit GPDK demo, and modified only the Momentum substrate with the actual metal and dielectric parameters from their technology. This is not what the tool was designed for!

The disadvantage of this simple solution is that layer numbers, via dimensions and DRC settings are not properly configured for the new technology.

The recommended, correct method to prepare Inductor Toolkit for a new technology is this:

- If you have an existing technology library from a PDK, you can use that. Otherwise, create a new library in the Inductor Toolkit PDK path (!) with the required layers for metal and vias, using the correct layer numbers for your technology.  
If you are not sure what to do, you can also use copy & paste from `Inductor_Generic_Layers_IC_lib` to a new library and make the necessary changes.
- If it does not exist, add a Momentum substrate file to this library.
- It is no longer necessary to define derived layers for via merging, because ADS2016 and later can now merge standard via arrays (emSetup Preprocessor setting)
- Use a text editor to open the technology file  
`<Inductor Toolkit directory>/configuration/technology.ael`  
and modify it for your technology, as described in chapter "Editing the Inductor Toolkit technology file". Don't forget to set a meaningful name in the `PDK_Menu_Name` at the beginning of the file, so that you can later identify what technology this Inductor Toolkit is configured for.
- Changes to `technology.ael` take effect when the workspace with Inductor Toolkit PDK is re-loaded.
- Inductor Toolkit must be configured to use this new/modified technology, as described in the next chapter.

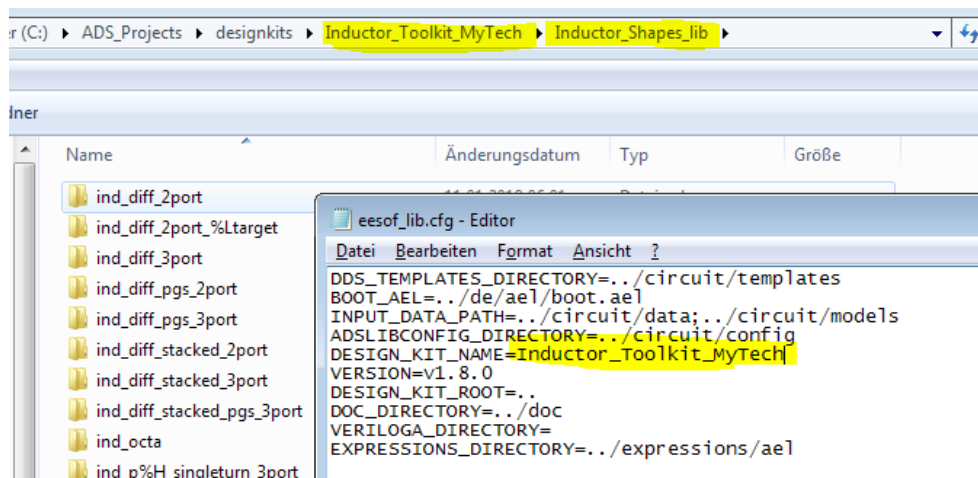


## Create Inductor Toolkit version for your own technology from scratch

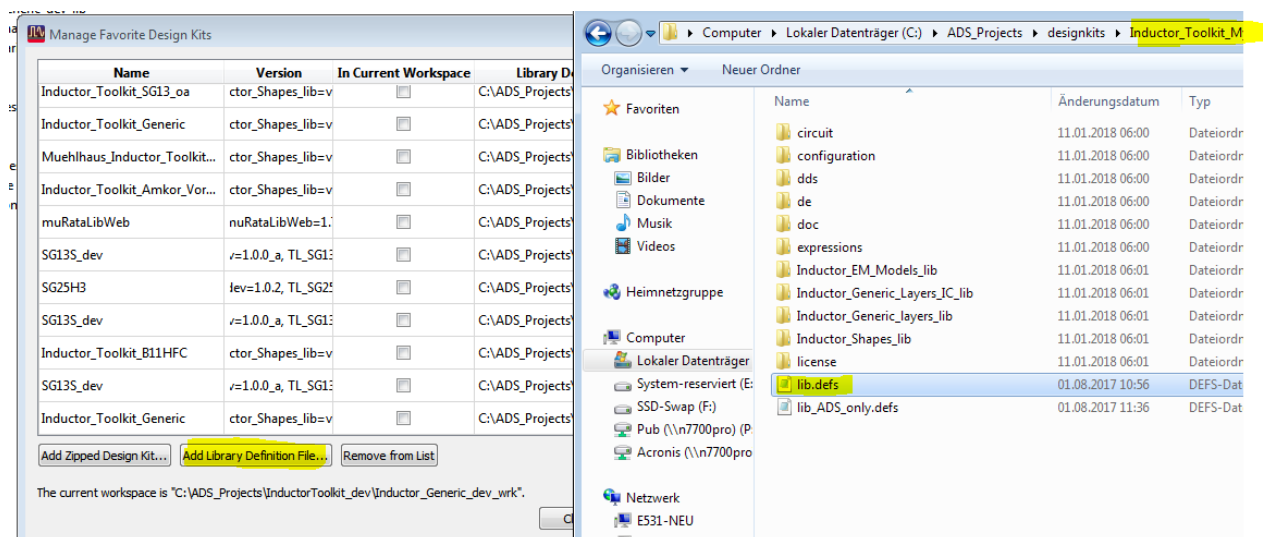
In this example, we assume that no ADS technology library exists for your technology.

First, copy the Inductor Toolkit directory to a new directory, and rename that new directory so that it reflects the name of the new technology. In our example, we will use technology MyTech and directory name Inductor\_Toolkit\_MyTech.

In the new directory, go to directory Inductor\_Shapes\_lib and open eesof\_lib.cfg in a text editor. Change the entry for DESIGN\_KIT\_NAME to Inductor\_Toolkit\_MyTech and save the file.

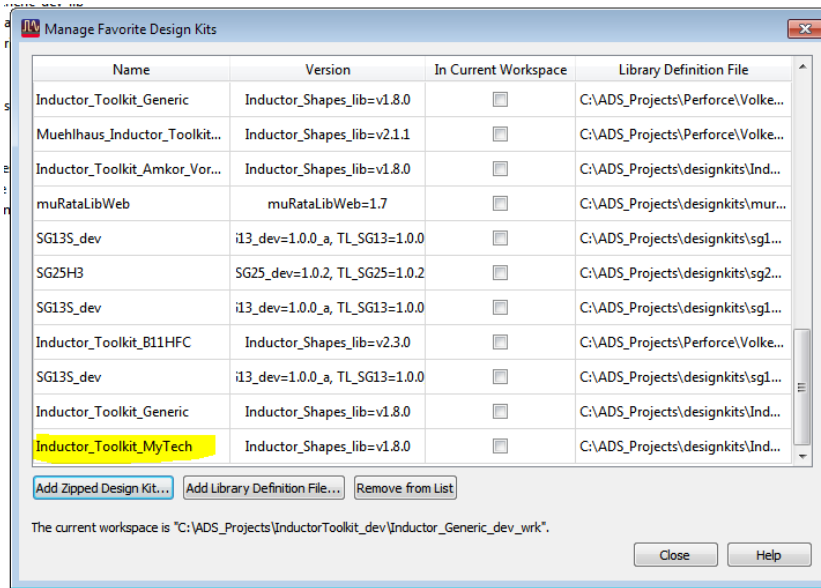


Start ADS, but do not open a workspace yet. Go to “Designkits > Mange Favorite Designkits ...” and click “Add Library Definition File...”. Browse to the new Inductor Toolkit directory, select the libs.def file and press OK.



If you have a workspace open, you will now be asked if you want to add this designkit to the workspace: NO. If you followed the instructions above, and there is no workspace open, there will be no such message.

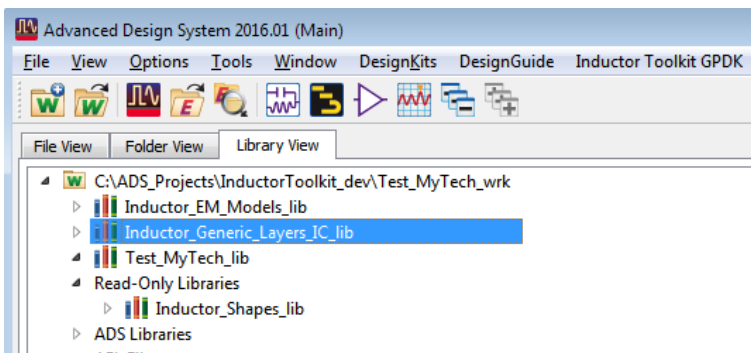
Now, the Inductor Toolkit PDK is registered to ADS with the new name, and we can start to modify the technology.



You could now create your own technology library completely from scratch<sup>10</sup>, but it is easier to modify the existing Inductor\_Generic\_Layers library. This is described below.

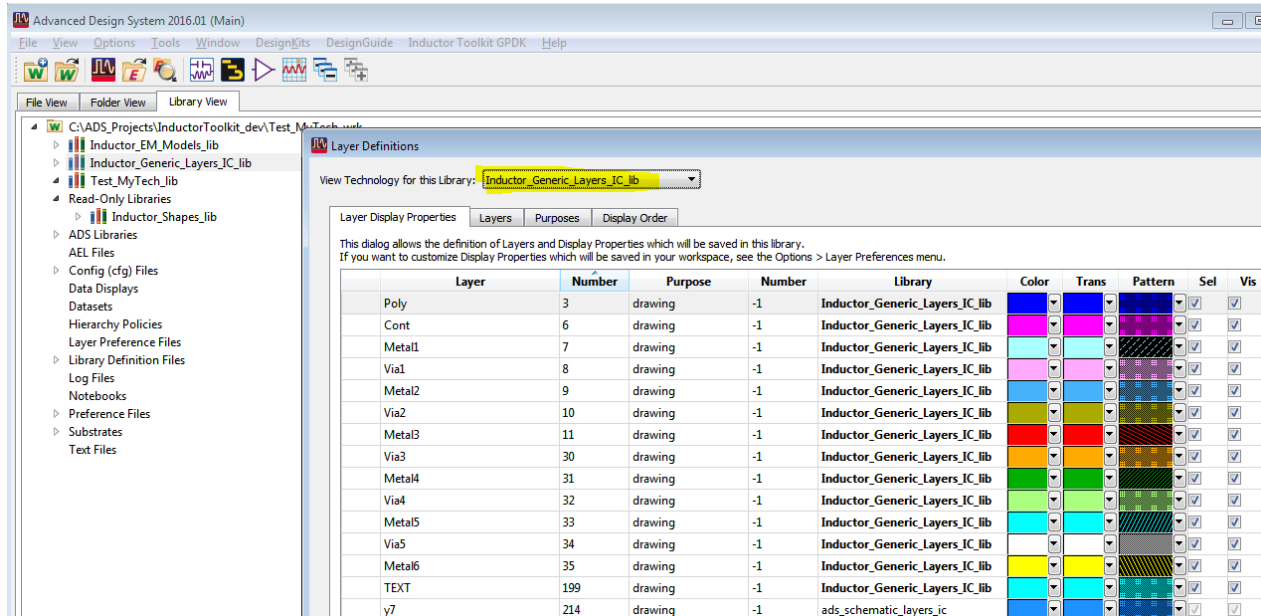
Open a new workspace and add our new Inductor Toolkit MyTech to your workspace, as described in chapter “Adding Inductor Toolkit to an ADS workspace” at the beginning of this User’s Guide. Select the Inductor\_Generic\_Layers technology (not the ADS default layout library!)

In ADS, change to Library View mode (tab above the project tree), and have a look at the libraries. You will notice that Inductor\_Generic\_Layers library is writable, so we can apply our changes.



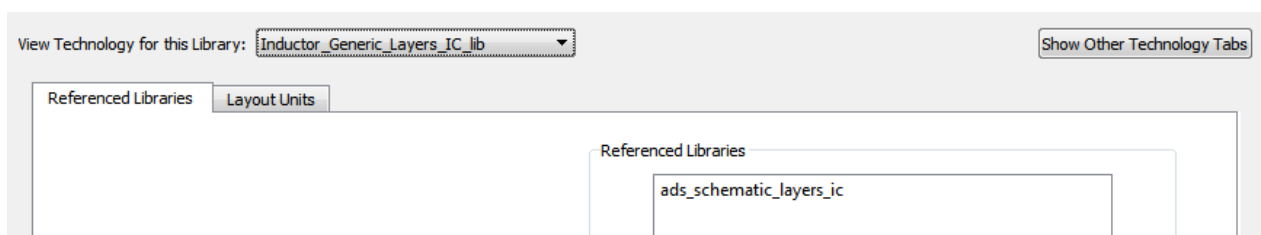
<sup>10</sup> If you already have an ADS PDK for your technology, you can skip the steps described here, and use that directly as a referenced library, instead of creating your own technology. See [https://youtu.be/7CD4\\_NomKls](https://youtu.be/7CD4_NomKls)

Now, we are ready to apply technology changes. Go to “Options > Technology > Layer Definitions” where you can modify the layer names and layer numbers (ADS layer number = GDSII layer number).



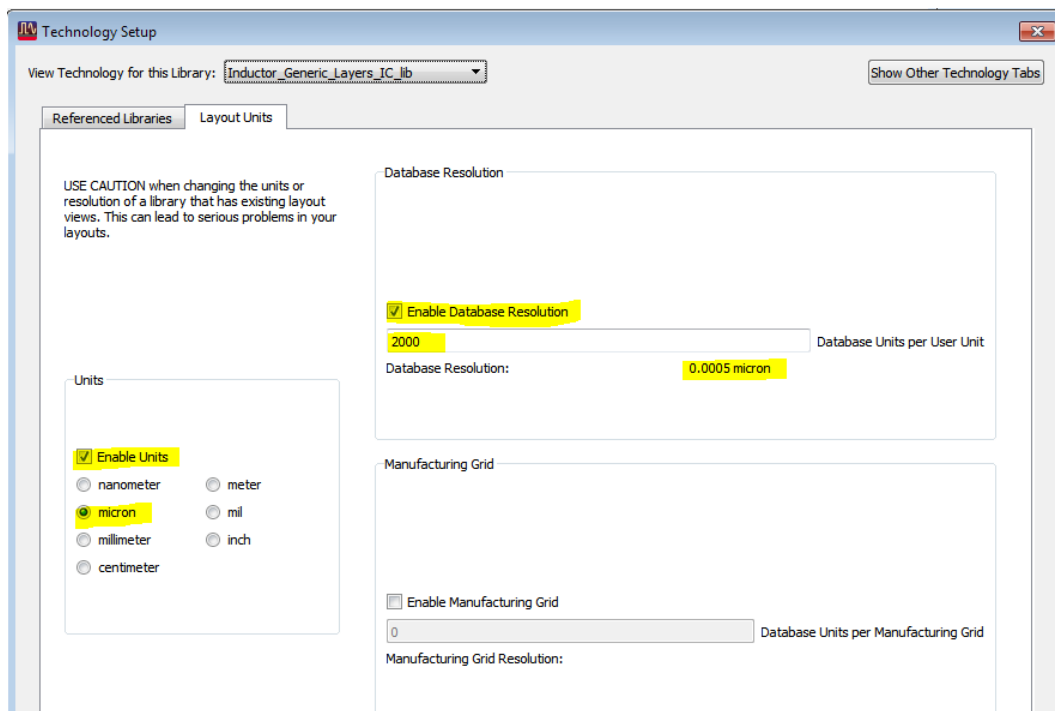
Save your changes, and go to Options > Technology > Technology Setup.

Our tech library Inductor\_Generic\_Layers has its own built-in configuration for layout. For schematics, it references the ads\_schematic\_layers\_ic. No changes required here, go to the Layout Units tab.

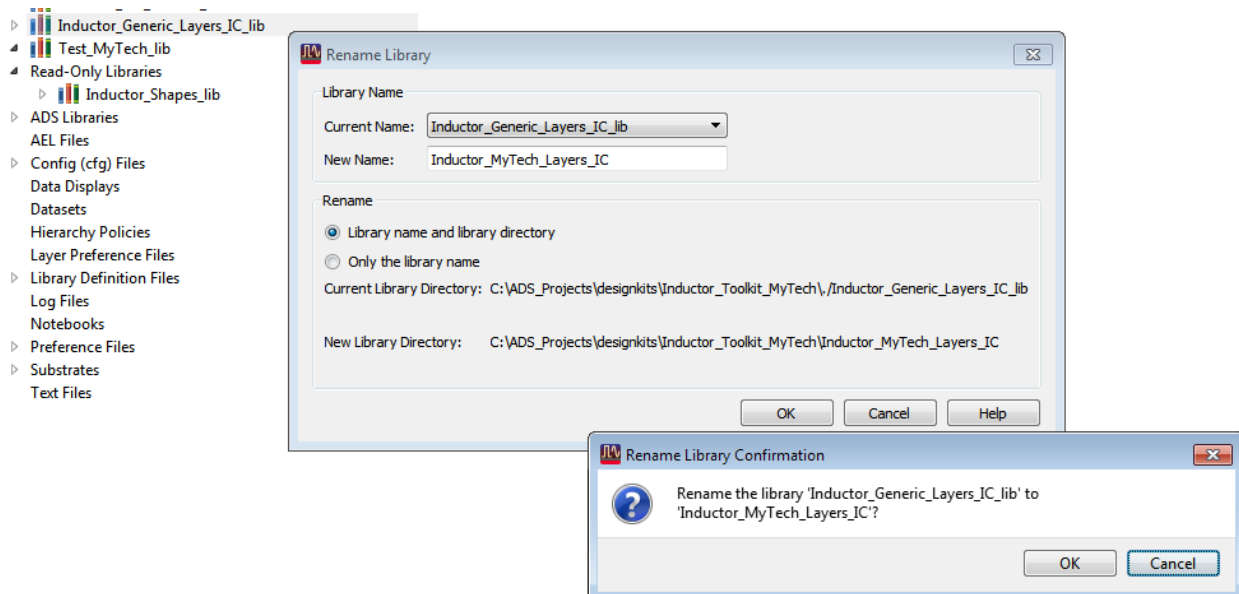


In the layout units tab, you will notice that the default tech library uses a database resolution of 0.5nm, which results from 2000 database units per user unit (micron). If your technology requires a different database unit for smooth GDSII data transfer, you can change that setting. When you are done, press OK.

Note: We will run an automated cleanup later that adjusts Inductor Toolkit layouts and pin positions if the database unit is changed.



If you wish, you can also rename the tech library from Inductor\_Generic\_Layers to Inductor\_MyTech\_Layers, to reflect your changes. Right click on the library name in the ADS project tree and select “Rename Library...”



What is still missing now:

- We need to configure Inductor Toolkit to use your actual layout layers for drawing, and configure technology details like via array dimensions. This is described in chapter “in chapter “Auto-update the emModels after technology changes”
- Editing the Inductor Toolkit technology file”.

- We need a valid Momentum substrate file for the technology. This is usually stored in the Inductor Toolkit technology library.
- After finishing the above steps, an automated update of the Inductor Toolkit libraries and emModels is needed, as described in chapter “Auto-update the emModels after technology changes”

## ***Editing the Inductor Toolkit technology file***

The layers used in the inductor artwork (for drawing the Pcell layout) are mapped to ADS layer numbers. For convenience, instead of working with layer numbers, you can first assign variables for the layer names, and then map those variable names.

```
// ===== start technology settings =====

decl PDK_Menu_Name = "Inductor Toolkit GPKD"; // Name of PDK in ADS menu

// Optional, to make things more readable, we assign variable names for
// the technology layers first, and use these variable names below.

decl Metal6 = 35;
decl Metal5 = 33;
decl Metal4 = 31;
decl Metal3 = 11;
decl Metal2 = 9;
decl Metal1 = 7;

decl Via5 = 34;
decl Via4 = 32;
decl Via3 = 30;
decl Via2 = 10;
decl Via1 = 8;
decl Cont = 6;
```

But of course, you can also map the layer numbers directly. The actual mapping of functional layers to ADS layer numbers (here using variables) is in the next section:

```
// ===== Layer definitions =====

// Now, we assign the ADS layer numbers for layout artwork.
// If the above section is properly defined, use those variable names. Otherwise, use the ADS layer numbers.

// All inductors
decl spiral_layer = Metal6; // spiral turns
decl crossover_layer = Metal5; // underpass for octagon crossover, also used as second metal for stac
decl underpass_layer = Metal4; // underpass for center tap for SG13
decl underpass2_layer = Metal3; // second underpass

decl via_layer_spiral_crossover = Via5; // via from spiral to crossover level
decl via_layer_crossover_underpass = Via4; // via from crossover level to underpass
decl via_layer_underpass_underpass2 = Via3; // via from underpass level to second underpass

// Octagon patterned ground shield
decl pgs_layer = Metal1; // patterned ground shield
```

The next section are design rules for the inductor layer (to check your input parameters for width and spacing) and underpass dimension rules that help to minimize the resistance of an underpass that is often located on a lower, more resistive layer, so that we want to make it wider than the top layer spiral.

```
// ===== Metal line dimension rules =====

decl min_linewidth = 0.5; // layout rule
decl max_linewidth = 30; // layout rule
decl min_spacing = 0.5; // layout rule

// ===== Underpass dimension rules =====

// underpass width is w*underpass_width_factor, but no more than max_underpass_width
decl underpass_width_factor = 2; // underpass same width as metal line
// maximum width used in underpass
decl max_underpass_width = 20; // underpass no wider than 20 micron
```

Next, we have a section that defines the via array details. It is recommended to define these settings, for an accurate and DRC-clean via array layout. However, if you don't have the required data available, you can also set `define_via_array = FALSE;` to just draw one big bounding box instead of the via array. In that case, all the other via definition entries are ignored.

The via array setting blocks define:

- The size of each individual via
- The distance between two vias in the array
- The distance between the outer via edges and the edge of the surrounding metal

```
// ===== Via dimension rules =====
//
// This section defines the size of vias used to create via arrays.

decl define_via_array= TRUE; // TRUE means vias are drawn as an array. FALSE

// If define_via_array is TRUE, all entries below MUST be defined for all ma

// size of single via
defun mi_get_via_size (layernum) {
  if (layernum == Cont) return 0.2;
  if (layernum == Via1) return 0.2;
  if (layernum == Via2) return 0.2;
  if (layernum == Via3) return 0.2;
  if (layernum == Via4) return 0.2;
  if (layernum == Via5) return 0.2;
}

// distance between two single vias in array
defun mi_get_via_gap (layernum) {
  if (layernum == Cont) return 0.2;
  if (layernum == Via1) return 0.3;
  if (layernum == Via2) return 0.3;
  if (layernum == Via3) return 0.3;
  if (layernum == Via4) return 0.3;
  if (layernum == Via5) return 0.3;
}

// distance from single via edge to surrounding metal edge
defun mi_get_via_margin (layernum) {
  if (layernum == Cont) return 0.1;
  if (layernum == Via1) return 0.1;
  if (layernum == Via2) return 0.1;
  if (layernum == Via3) return 0.1;
  if (layernum == Via4) return 0.1;
  if (layernum == Via5) return 0.1;
}
```

The next two entries are optional. The `get_pin_shape_layer` definition sets the purpose for drawing the pin shape polygon at the inductor terminals. The default setting is -1 for purpose drawing. If you have no special requirements, leave this default.

```
// ===== PIN layer/purpose handling, these functions are optional =====

defun get_pin_shape_layer (layer_number)
{
  // input parameter is the layer number (=GDSII stream number)
  // return is the ADS layerid for the corresponding pin layer-purpose-pair
  // example code to draw area pins on same layer, but purpose 2 for pin:
  // return db_layerid (layer_number, 2);
  // Area pins can be skipped/surpressed if this function is removed

  return db_layerid (layer_number, -1); // same layer, use purpose -1
}

defun get_pin_label_layer ()
{
  // return is the ADS layerid for the layer-purpose-pair for pin text (labels)
  // Pin text labels can be skipped/surpressed if this function is removed

  return db_layerid (199);
}
```

Function `get_pin_label_layer` sets the ADS layer number used for drawing the pin labels. If this function is removed from the file, no pin labels are drawn.

The next section defines extra layers that are drawn “under” the inductor. A typical use for this is blocking PWell or blocking filler, or adding a polygon to mark the area as an inductor for LVS recognition.

The syntax is a bit special, because each layer entry requires a layer number and a purpose. Examples: `list("102:-1", "10:2")` means: draw on extra layers 102 purpose -1 and layer 10 purpose 2. An empty `list()` entry means: don't draw on any extra layers.

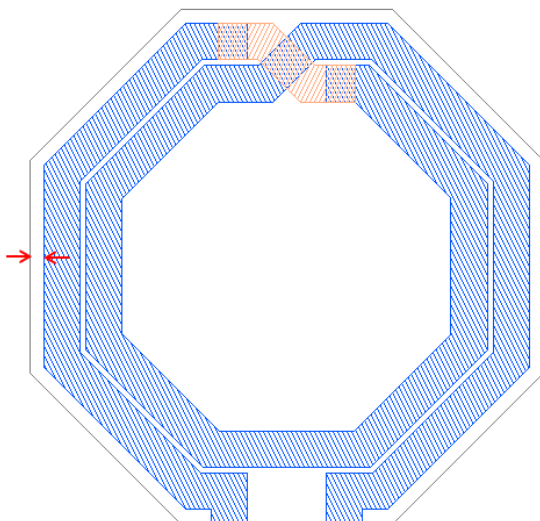
```
// ===== Extra layers, dimension of extra layer boundary =====

// Extra layers that must appear under inductors, like PWellBlock, NoFiller, ...
// list("102:-1", "10:2") means: draw on extra layers 102 purpose -1 and layer 10 purpose 2
// list() means: don't draw on any extra layers

decl extra_inductor_layers_list = list("102:-1");

// Distance between outline on extra layers (e.g. PWellBlock, NoFiller) and the inductor
decl extra_layer_extension = 5;
```

The last parameter, `extra_layer_extension`, defines how much the extra layers extend outside the outer inductor diameter. Unit is micron.





Documentation is also included in the file as comments.

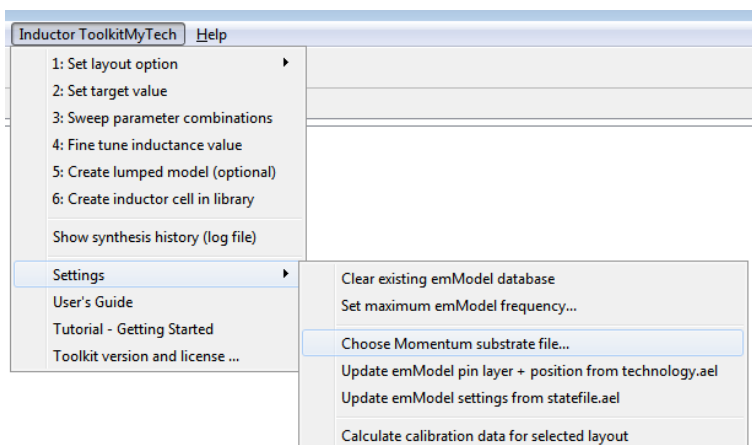
The technology.ael file is written in ADS AEL language. If you get an error on loading the PDK after you made changes to this file, check for these syntax issues:

- Code lines must end with a ;
- Variable declarations start with keyword decl, function definitions with keyword defun
- Comments start with //
- Strings items are included in “ “

### ***Auto-update the emModels after technology changes***

After the technology library was created and the technology.ael file was updated with the corresponding data, and a valid Momentum stackup was added to the technology library, we can run some automated steps to update the Inductor Toolkit models.

First, use “Inductor Toolkit > Settings > Choose Momentum substrate file...” and select the Momentum substrate that you want to use. This must be located the Inductor Toolkit technology library!



Next, run “Update emModel pin layer + position from technology.ael”. This will use the layer information from technology.ael to move emModel pins to the correct layers, and will also move pins to the correct location after changing the technology database unit.

Do not use “Update emModel settings from statefile.ael”, unless instructed by support to do so. This is an expert setting that allows to automatically apply global changes to all emModel files.

Do not use “Calculate calibration data for selected layout”, unless instructed by support to do so. This menu item is used to re-calculate the synthesis parameters after (large) changes to the metal thickness or inter-metal dielectrics.

## Preference file

Some settings can be customized by the user in the Inductor Toolkit preference file. This file is part of the ADS AEL code that runs when Inductor Toolkit is loaded, so be careful with any changes that you make.

The preference file is located in the license directory in the PDK, e.g.

*C:\designkits\Inductor\_Toolkit\_GPDK\configuration\preferences.ael*

*Below is an example of some settings.*

```
// Preference file for inductor toolkit

// ===== USER INTERFACE =====
decl save_all_designs_on_simulate = TRUE; // (TRUE|FALSE) save all open designs before synthesis
decl close_all_designs_on_simulate = TRUE; // (TRUE|FALSE) close all open designs before synthesis
decl lumpedmodel_start_optimizer = TRUE; // (TRUE|FALSE) automatically start optimizer

// ===== SYSTEM OPTIONS =====
decl finetune_emModel_cell_prefix = "finetune_"; // use this emModelsuffix for finetune
decl Linux_open_URL_commmmand = "xdg-open"; // command to open PDF and URLs.
decl Linux_open_PDF_commmmand = "xdg-open"; // command to open PDF and URLs.

// ===== LAYOUT OPTIONS =====
decl include_stackedmetal_layouts = TRUE; // (TRUE|FALSE) include stacked metal layouts in menu
decl pH_single_feedline_width = 10; // feedline width (µm) for single turn picoHenry inductor
```

## Changes

Inductor Toolkit Generic 2.1: Updated for ADS 2017. Various automation to configure emModels after making technology changes. Demo mode now restricted to square inductor shapes. Re-written documentation for technology customization.