# SpInd Manual

Vito Minerva

June 20, 2004

# Contents

# Chapter 1

# General information

SpInd is a program for the analysis and design of integrated inductors. It was (and, for the time being, still is) developed in the Matlab environment[1].
SpInd is the acronym for SPiral INDuctor and computes inductor's electrical parameters using analytical formulas[2]. This way, it is much faster than electromagnetic software, but still accurate enough for you to rely upon it to design your integrated inductors[3].
The literature on integrated inductors is very huge, but the fully-reproducible structures are very few, because essential process parameters are usually omitted from papers for confidential reasons. Nonetheless, from the ones available[4], I have come to the following conclusions:

- the low-frequency inductance is generally very accurate;
- the quality factor is close to reality (sometimes it's a bit higher, sometimes a bit lower), when the inductor is excited single-endedly. On the other hand, when differentially driven, the quality factor peak is usually 30 % higher than it should;

[1] SpInd works with Matlab 6.5; if you have an older version of Matlab, it's still possible to use SpInd after some minor changes in the code; see Par. 1.2. SpInd also needs the Matlab's Optimization Toolbox to run the optimization (see Chapter 6); but for the other features SpInd provides, you can do without this toolbox.

[2] For more details about this, see V. Minerva, *Analysis and Design of Integrated Inductor for RF Applications*, Ph.D. Thesis, Politecnico di Milano, Spring 2004.

[3] However, needless to say, this program comes with NO WARRANTY whatsoever; please don't unleash your lawyers against me, if this software doesn't work as expected. After all, it's completely free and comes with the source code, so that you can improve it by yourself.

[4] In particular, I'd like to cite:

- J.N. Burghartz, M. Soyuer and K.A. Jenkins, *"Microwave Inductors and Capacitors in Standard Multilevel Interconnect Silicon Technology,"* IEEE Trans. on Microwave Theory and Techniques, Vol. 44, No. 1, pp. 100-104, January 1996.
- J.N. Burghartz, D.C. Edelstein, M. Soyuer, H.A. Ainspan and K.A. Jenkins, *"RF Circuit Design Aspects of Spiral Inductors on Silicon,"* IEEE J. Solid-State Circuits, Vol. 33, No. 12, pp. 2028-2034, December 1998.
- M. Danesh and J.R. Long, *"Differentially Driven Symmetric Microstrip Inductors,"* IEEE Trans. on Microwave Theory and Techniques, Vol. 50, No. 1, pp. 332-341, January 2002.

Those three papers pretty much exhaust all the available literature on fully reproducible inductors.

- the self-resonance frequency $f_{SR}$ is generally accurate, both for single-ended and for differential excitation, especially for single layer structures. For multi-layer structures, it overestimates the $f_{SR}$ a bit.

In all, if you condone the differential quality factor, it works quite well. Nonetheless, since nowadays ground shields or deep trenches are usually present underneath integrated inductors, a word of advice becomes necessary.
Ground shields increase up to 30 % the maximum quality factor, depending on technology, while can decrease the $f_{SR}$ by an equal amount. As for deep trenches, they increase both $f_{SR}$ and $Q$, but their effect is less dramatic. Keep that in mind when you use SpInd , since it doesn't model neither shields nor trenches.

## 1.1 Installation and execution

The installation steps are very straightforward: create a new folder for SpInd , unzip the file in it and start Matlab. From the Matlab prompt, change the working directory and go to the SpInd directory. For example, if you saved the zipped file in

```
c:\spind
```

then type at the Matlab prompt:

```
cd c:\spind
```

and then

```
main_win
```

The SpInd graphical user interface opens. It contains nine menus that are briefly described in the following chapters.

## 1.2 Execution with older versions of Matlab

If you have a version of Matlab older than 6.5, you can still use SpInd . Basically, you need to modify the command path. Open the `main_win.m` file. As you see, it contains the following line of code:

```
allsubdir = genpath(progpath);
```

Some previous versions of Matlab don't have the `genpath` command. In this case, manually include all SpInd subdirectories, like this:

```
path = path(path,'c:\spind\asitic');
```

In the above example, the subdirectory `asitic` has been included in the path. There are currently 18 subdirectories, so that this operation is rather tedious. Remember to include the SpInd directory

```
c:\spind
```

in the path too.
Unfortunately, other fixes are necessary. In some older versions, the `try` and `catch` commands work only if they are followed by a comma, like this:

```
try,
  [...]
catch,
  [...]
end
```

It's very likely that I forgot some commas after the `try` and `catch` commands. If Matlab gives you an error, just add the comma.
Finally, in some older versions of Matlab, the `ismember` command doesn't return, quoting from the Matlab help, "an index array LOC containing the highest absolute index in S for each element in A which is a member of S and 0 if there is no such index."
I have created a file, named `ismemebr2`, that can do that. So you have two possibilities:

1. either you rename all the occurrences of `ismember ismember2`;
2. or you rename the `ismember2` file, so that Matlab uses it instead of its built-in file.

The second option is obviously handier. This way, though, you can not use the Matlab's `ismember` command when working with SpInd . The replacing command works just as well, but I thought it right to tell you.

## 1.3 How to report a bug

This program was developed as a side project of my Ph.D. activity, so that I worked on it in short bursts over a long period, when the workload was lesser. As a result, the code is a bit messy. To make things worse, I usually don't comment my code. All this is just to say that bugs in SpInd are very likely, even though they should be minor bugs, i.e, bugs not affecting the results but just causing annoyance.
When you find a bug, you have two ways to go:

1. either you correct them by yourself; in this case you're a genius, because, as I said, I don't comment my code (to be fair, the code is quite straightforward, so that any Matlab buff can tinker with it);
2. or you send me an email (my email address is `kubik@tin.it`), providing as many details about the bug as possible (when it occurs, what it causes, what the Matlab workspace reports about it, and so on).

Even if you fix the bug by yourself, I'd like you to send me an email about it, so that I can fix my version of SpInd and make available to others the best code at that moment, with the right credit due (I know, I'm claiming SpInd will have more than one user).

# Chapter 2

# File

The `File` menu contains the following commands:

- `Import Touchstone File`,
- `Load Workspace`,
- `Save Workspace`,
- `Save As Touchstone File`,
- `Import Inductor From DXF File` and
- `Exit`.

When you open SpInd , the third and fourth commands are not enabled, while the others are. This is obvious, since you have nothing to save yet.
`Import Touchstone File` is a self-explaining command; after you have imported a Touchstone file, you can see its scatter parameters or find its wide-band model (see Chapter 5). If you have also performed a frequency sweep analysis, you can also compare SpInd 's results with those of the imported file (again, see Chapter 5).
`Load Workspace` loads a Matlab workspace saved previously.
`Save Workspace` and `Save As Touchstone File` become active only after that a simulation (Chapter 5) or an optimization (Chapter 6) has been performed.
`Import Inductor From DXF File` allows you to import an inductor's layout from another program[1]. You can then simulate the imported inductor with SpInd .
`Exit` teminates SpInd and returns to the Matlab workspace.

[1]To be honest, I have imported only DXF files previously saved by SpInd . It may well be that the importing doesn't work with DXF files from other programs. I still have to work on that.

# Chapter 3

# Inductor topologies

Inductor topologies that SpInd can handle are listed in the `Inductor Topology` menu. Currently, they are:

- Square Spiral Inductor;
- Rectangular Spiral Inductor;
- Inter-wound Spiral Inductor;
- Twin-spiral Inductor;
- Solenoid Inductor;
- Inner Rectangular Spiral Inductor;
- Multi-metal Series-connected Spiral Inductor;
- Multi-metal Shunt-connected Spiral Inductor;
- Polygonal Spiral Inductor;
- Tapered Spiral Inductor;
- Differential Square Spiral Inductor;
- Differential Polygonal Spiral Inductor;
- Wire Inductor.

Each inductor is illustrated in the following sections. When you choose one of them, a dialog box opens asking for the inductor's geometrical parameters. They slightly vary from structure to structure, but generally they are the outer dimension $D_{out}$ (or the height and the base, if the structure is rectangular), the metal width $w$, the spacing $s$, the number of turns $N$ and the metal level(s) the structure lies on. If the values you enter infringe some geometrical rules, a warning box opens, saying some data (but not *which* data) are incorrect. When your inductor is doable, click the `Confirm button` and go to the simulation menu (Chapter 5).

How does SpInd know which metal levels are available by the process? Every time it starts, SpInd loads the default technology file. To see how to define the default technology file, go to Chapter 4.
We point out that, before starting a simulation, an optimization or any other job with SpInd , you need to define the structure you are going to work on. That's why this menu will certainly be the first you'll access when working with SpInd (unless you import data from other sources, see Chapter 2).

## 3.1 Square Spiral Inductor

The square spiral inductor is illustrated in Fig. 3.1. It's the topology usually employed for integrated inductors.

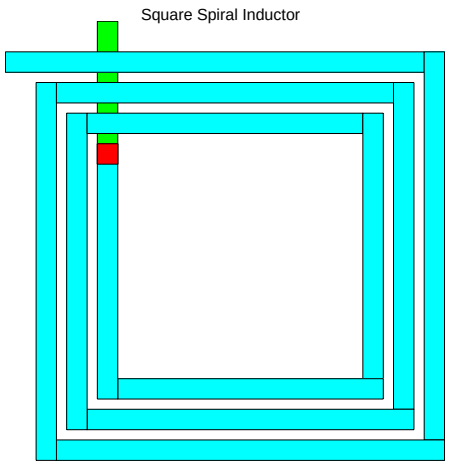


Figure 3.1: Square Spiral Inductor.

## 3.2 Rectangular Spiral Inductor

It's just like the spiral in Fig. 3.1, but it's rectangular.

## 3.3 Inter-wound Spiral Inductor

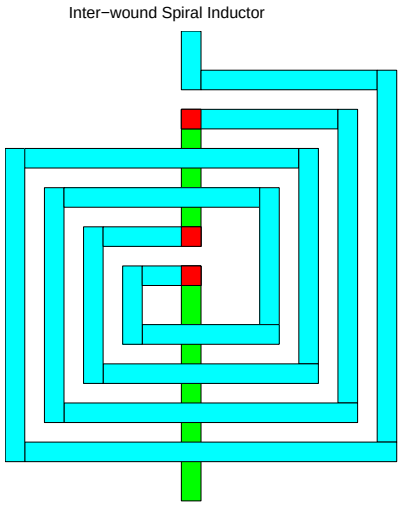


Figure 3.2: Inter-wound Spiral Inductor.

## 3.4 Twin-spiral Inductor

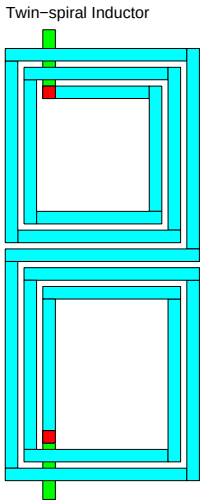


Figure 3.3: Twin-spiral Inductor.

## 3.5 Solenoid Inductor

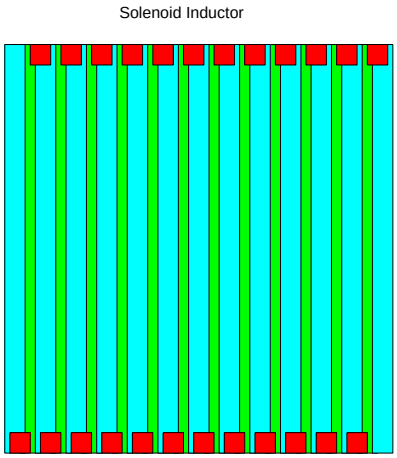


Figure 3.4: Solenoid Inductor.

## 3.6 Multi-metal Series-connected Spiral Inductor

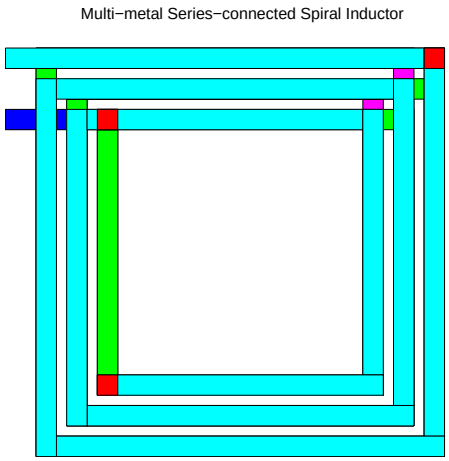


Figure 3.5: Multi-metal Series-connected Spiral Inductor.

## 3.7 Multi-metal Shunt-connected (Stacked) Spiral Inductor

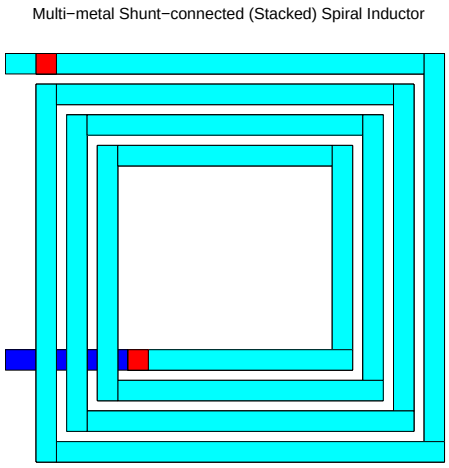


Figure 3.6: Multi-metal Shunt-connected (Stacked) Spiral Inductor.

## 3.8 Polygonal Spiral Inductor

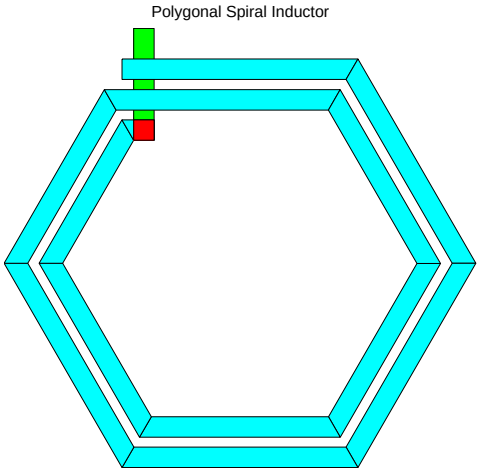


Figure 3.7: Polygonal Spiral Inductor.

## 3.9 Tapered Spiral Inductor

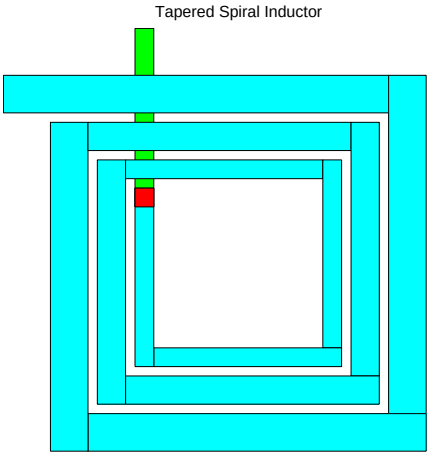


Figure 3.8: Tapered Spiral Inductor.

## 3.10 Differential Square Spiral Inductor

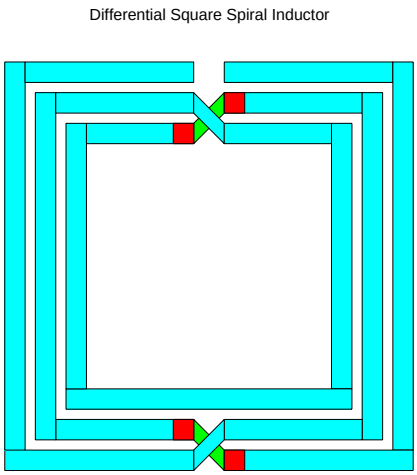


Figure 3.9: Differential Square Spiral Inductor.

## 3.11 Differential Polygonal Spiral Inductor

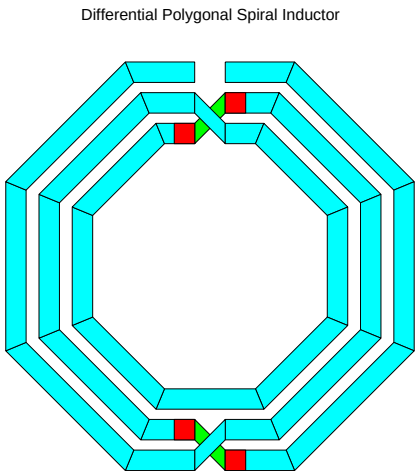


Figure 3.10: Differential Polygonal Spiral Inductor.

## 3.12 Wire Inductor

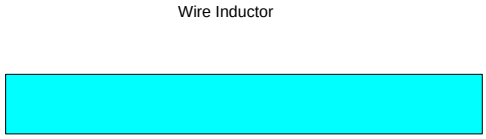


Figure 3.11: Wire Inductor.

# Chapter 4

# Technology

This menu lets you load, create, modify and see the technology files for your simulations.
The commands available in the `Technology` menu are:

- `Load Data From File`,
- `Insert Data Manually`,
- `Change Metal Parameters`,
- `Change Layer Parameters`,
- `Save Manually Inserted Data`,
- `View Current Technology Parameters In Use` and
- `Set Default Technology File`.

After you launch the program at the Matlab prompt, only the first, second and last command are enabled. By clicking on `Load Data From File`, a menu opens, from which you can choose the technology file for your simulations. Notice that it's not necessary that you choose a technology file every time you start a simulation with SpInd . In fact, the default technology file is loaded and used every time the program is invoked. `Set Default Technology File`, as the name suggests, lets you set the default technology file. This way, you have to bother about technology parameters only when you switch technology. As long as you work with a given technology, you won't need to access this menu any more.
How do you create technology files? There are basically two ways to do that. One is to open a technology file that comes with the version of SpInd you are working on. A technology file looks like this:

```
techf = 'sige5am';

% Oxide
layers(3) = 1;
roxi = 10^12;       % oxide resistivity
```

```
epsox = 4.10;       % oxide relative dielectric permittivity
toxi = 17.91e-6;    % oxide thickness

% Substrate (bulk)
layers(1) = 1;
rsub = 0.16;        % substrate resistivity
epssub = 11.9;      % substrate relative dielectric permittivity
tsub = 300e-6;      % substrate thickness

% Metal

% metal level 1 (Bottom Metal)
mlayer(1).str = 'm1';
rrsh(1) = 0.076;
tt(1) = 0.63e-6;
dd(1) = 1.90e-6;
% metal level 2
mlayer(2).str = 'm2';
rrsh(2) = 0.045;
tt(2) = 0.85e-6;
dd(2) = 3.73e-6;
% metal level MT
mlayer(3).str = 'mt';
rrsh(3) = 0.045;
tt(3) = 0.83e-6;
dd(3) = 5.78e-6;
% metal level AM (Top Metal)
mlayer(4).str = 'am';
rrsh(4) = 0.00725;
tt(4) = 4e-6;
dd(4) = 9.61e-6;
```

There is a $1 \times 3$ vector called `layers` that gives information about the technology dieletric layers. In the example, `layers` = [101], that is, the central layer (usually an epi layer) is not present and only the substrate and the oxide are available. If you have a technology with more than three dielectric layers, you must leave one off, since SpInd allows only three layers. However, this is usually enough for most technologies, unless you have two or more thin epi-like layers between the substrate and the oxide. In that case, you can define an equivalent central layer whose parameters are the series of the middle layers.
Now, a quick guide to the technology parameters. In the example, the oxide thickness `toxi` is set to 17.91 $\mu$m, but it can be set to any value, as long as it is bigger than $d_4$ (see Fig. 4.1). So why do we use this very specific distance of 17.91 $\mu$m? Because, if we want to export technology parameters for an electromagnetic simulation, we must specify this value accurately. Otherwise, it suffices that $t_{oxi} \geq d_{MAX}$, the distance between the dielectric layer under the oxide and the top metal layer.
As for the substrate thickness `tsub`, this value is usually not provided by technology manuals, but it is in the hundreds of microns. Set `tsub` = 200 or 300 $\mu$m

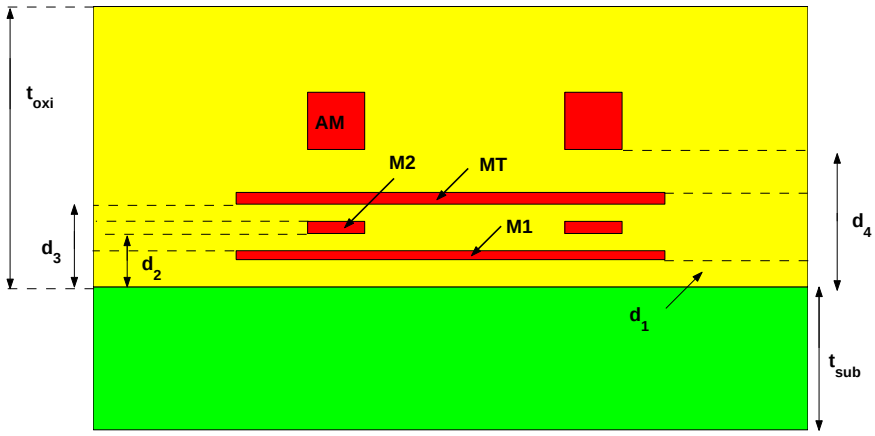


Figure 4.1: Cross section of a typical bipolar process.

if you don't have this information.
Leave the other oxide parameters unchanged, because they are typical of oxide (well, sometimes manuals report `epsox` = 4 instead of 4.1, but you shouldn't be too picky). On the contrary, the substrate resistivity is important and depends on the technology you are using. This value is normally provided by technology manuals in Ω-cm, while SpInd uses Ω-m. When you write the SpInd tehcnology file, remember to divide the substrate resistivity you found in your manual by 100!
For a bipolar process, `rsub` = 10 − 20 Ω-m, while in a CMOS process `rsub` is usually 3 or 4 order of magnitude smaller[1].
Once you have provided the information about dielectric layers, you must enter the data about the metals. Metals are numbered from the bottom to the top. For each metals, four data are necessary, each one of them stored in a vector (the first entry contains data about the bottom metal layer):

- the metal layer's name (stored in `mlayer.str`);
- the metal layer's sheet resistance (stored in `rrsh`);
- the metal layer's thickness (stored in `tt`), and
- the metal layer's distance from the dielectric layer under the oxide (stored in `dd`).

Of course, you can use the fancy names you want to identify the metals, but you should use the same ones as your process. This also comes in handy when you export the inductor's layout (see Chapter 8). You can easily find the other data in your manual as well. One caveat, though; manuals give sheet resistances in

[1]That is why ground shields are an unavoidable choice in CMOS processes. They are commonly used in bipolar processes too, but their effect there is less dramatic. Since SpInd doesn't model ground shields (well, at least not yet), it returns very low quality factors for CMOS processes, while the actual values can be much higher thanks to the shield. You could then use an "equivalent" substrate resistivity, i.e. increase `rsub`, to take the shield into account. For bipolar processes, this operation is unnecessary.

m$\Omega/\square$, while SpInd wants $\Omega/\square$. If quality factors provided by SpInd are too low, you have probably forgotten to divide the sheet resistance by 1000.
The distances $d_i$, ($i = 1, \ldots, 4$ in this example) are calculated from the dielectric layer below (in this case the substrate, see Fig. 4.1).
Finally, don't forget to write the technology file's name in the variable `techf` and to save the file you have just created.
The other way to create a technology file is to use the `Insert Data Manually` command. A window opens and you must enter the number of metals and whether an epi layer is present or not (choose from options `with Epi | without Epi`). When you have finished, click the button `Confirm` to go to the next window. You must now enter the metal parameters: thickness, distance and sheet resistance. Confirm and go to the following window, when you must enter the metals' names. Confirm again and the last windows opens, where you must enter the data about the dielectric layers.
You can use the process you have created this way during the open session, but when you exit SpInd , these data are lost. You must save them in a file, if you wanna use them in a following session. To do that, choose the `Save Manually Inserted Data` command. A dialog box opens for you to save the process you have just created. Notice that, after you have completed the creation of the technology file, the commands `Change Metal Parameters` and `Change Layer Parameters` are enabled. You can use them to modify metal and dielectric properties, respectively. (You could also do that by selecting again the `Insert Data Manually` command, but, this way, you must repeat the whole process of insertion).
The `Set Default Technology File` command lets you define the default technology file, that is, the process properties that SpInd loads and uses to simulate inductors, every time it starts. It can be any technology file, even one created by you (but you must save it first with `Save Manually Inserted Data` command).
The `View Current Technology Parameters In Use` becomes active only after that `Load Data From File` command has been executed or an inductor topology has been chosen (see Chapter 3). It lets you see the properties of the process currently used by SpInd .

# Chapter 5

# Simulate

This menu contains
The commands available in the `Simulate` menu are:

- `Start Simulation`,
- `Simulation Frequency Range`,
- `Start Simulation Over Frequency Range`,
- `Compare Simulation with S2p File`,
- `See Scatter Parameters`,
- `Of The Equivalent Circuit Plot`,
- `Find Wide-Band Model`,
- `Find Wide-Band Model of S2p File` and
- `Save Wide-Band Model`.

At the start of a SpInd session, none of those commands is enabled. In fact, you must first select an inductor (see Chapter 3) or import it from a DXF file (see Chapter 2). This enables the first two commands. You can also import a Touchstone file, which, for the uninitiated, contains the scatter parameters of a device. In that case, the `See Scatter Parameters` and the `Find Wide-Band Model of S2p File` commands activate[1].
The `Start Simulation` command performs a one-frequency-point simulation (SpInd asks for the frequency of simulation when you choose the inductor topology or import it from a DXF file). If you want to change the frequency of simulation later on, just go back to the inductor topology or re-import the DXF file and change the frequency. After the simulation, results are shown in the Matlab workspace. They look like the following:

[1] You can import a Touchstone file at any time; only then will the two above-mentioned options become active.

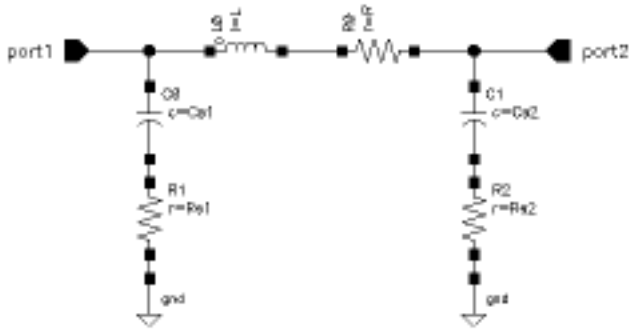


Figure 5.1: The $\pi$-model equivalent circuit at the frequency of simulation.

```
Total Inductance: 2.79 nH
 Quality factor at 1.0 GHz: 6.86 (Port 1) -- 6.87 (Port 2) -- 6.92 (Diff)

 Pi Model at 1.0 Ghz: Ls = 2.771 nH & Rs = 2.495 Ohm
                      Cs1 = 78.346 fF & Rs1 = 298.788 Ohm
                      Cs2 = 77.471 fF & Rs2 = 259.951 Ohm

 Estimated Self-Resonance Frequency: 14.50 GHz

>>
```

The total inductance is the inductance at Port 1 (with Port 2 grounded), while the three quality factors represent respectively $Q$ at Port 1 (with Port 2 grounded), $Q$ at Port 2 (with Port 1 grounded) and the quality factor when the structure is differentially driven. A $\pi$-model equivalent circuit at the frequency of simulation is also provided (see Fig. 5.1), together with an estimation of the self-resonance frequency $f_{SR}$. In addition, a Matlab figure with the layout of the simulated inductor opens.
If you want to simulate the inductor in a certain frequency range, then select the `Simulation Frequency Range` command. A dialog box opens and you are asked for the minimum and maximum frequency and for the number of points to be sweeped within the frequency range.
At this point the `Start Simulation Over Frequency Range` command activates; select it and the simulation starts. At each iteration, results are shown in the Matlab workspace just like in the single frequency's case. However, iterations are so fast you can only watch the results passing by. Don't worry: when the simulation finishes, four boxes open: the first one plots the three inductances (at Port 1, at Port 2 and differential); the second one plot the three quality factors (again at Port 1, at Port 2 and differential). The third figure plots the $S_{11}$ and $S_{12}$ scatter parameters on a Smith chart, while the fourth figure illustrates the inductor's layout.
On each figure, five buttons are present: `Zoom`, `Save Figure`, `Export in EPS`, `Print` and `Close`. As you have probably guessed, they do what they say (no surprises here).

At this point, the options `Of The Equivalent Circuit Plot` and `Find Wide-Band Model` are active. Each of them is a menu that contains in turn a list of commands. They are:

- `Series Inductance`,
- `Series Resistance`,
- `Substrate Capacitance` and
- `Substrate Resistance`,

for the `Of The Equivalent Circuit Plot` command and

- `Model 5`,
- `Model 1`,
- `Model 6` and
- `Model 7`,

for the `Find Wide-Band Model` command (the `Find Wide-Band Model of S2p File` menu contains exactly the same list of commands).
Now, from the `Of The Equivalent Circuit Plot` command, you can select for example one of the $\pi$-model lumped elements depicted in Fig. 5.1 and plot it versus the frequency range.
Since a point-by-point equivalent circuit is of little use for, say, a time-domain analys, it is also possible to find a wide-band model for the simulated inductor. Five possible models are available (the above-mentioned `Model 1, Model 5, Model 6` and `Model 7`). Some work better than others, depending on the inductor's structure. Try all of them (all of them take a very short time to execute) and then choose the one that best fits the data[2]. After the fitting process is finished, three figures open. One contains the comparison between the simulated inductance at Port 1 and the same inductance provided by the wide-band model. The second figures compares the quality factor at Port 1 by the simulation and by the wide-band model. Finally, a thrid figure lists the wide-band model's parameters.
After executing one of the four wide-band models, the `Save Wide-Band Model` command is enabled. It allows you to save the last wide-band model you have executed in SCS format, so that you can import it in your favourite circuit simulator[3].
The `Find Wide-Band Model of S2p File` command is identical to the `Find Wide-Band Model` command, except that it works on imported Touchstone files (it becomes active after that one of such files has been imported). The `See Scatter Parameters` command is enabled after importing a Touchstone file, too. It plots the magnitude (in dB) and phase (in rad/s) of the $S_{11}, S_{12}$ and

[2]If none of the four models fits your data accurately (luckily, it doesn't happen very often), try then to increase (or to decrease) the frequency range where data must be fitted.

[3]I have imported it in ADS and it works; I think it should also work with other circuit simulators, such as Cadence. Notice that by opening the SCS file in a SPICE-like environment, you can also see how the four wide-band models look like. For the time being, there is no such a visualization option in SpInd .

$S_{22}$ parameters. If a frequency sweep has been performed by means of the `Start Simulation Over Frequency Range` command, scatter parameters of the SpInd -simulated inductor are also available and are plotted together with the imported ones.
Of course, this possibility makes sense only if the imported scatter parameters and the ones produced by SpInd refer to the same structure. In fact, this option has been introduced to compare results by electromagnetic programs with results by SpInd .
If you're not familiar with scatter parameters, you can also compare results imported from Touchstone files with results provided by SpInd , through the `Compare Simulation with S2p File` command.
It becomes active after that both `Start Simulation Over Frequency Range` and `Import Touchstone File` have been executed. This command plots exactly the same figures as `Start Simulation Over Frequency Range`; this time, though, the correspondent curves from the imported file are also present. Again, this comparison makes sense if the results by SpInd and the imported Touchstone file refer to the same structure.

# Chapter 6

# Optimization

The `Optimization` menu allows you to find the best quality factor for a given inductance or a given area occupation, within the constraints you have set.
It is made of just two commands:

- `Insert Constraint` and
- `Start Optimization`.

The first command becomes available after you have selected an inductor's structure (see Chapter 3), while the second command is enabled after you have finished the first.
When you select `Insert Constraint`, a dialog box with a pop-up menu opens. Three options are available: `Choose below`[1], `Inductance` and `Outer Dim.` If you choose the first option, nothing happens, because you must choose one of the last two options.
`Inductance` allows you to find the optimum inductor for a given inductance (see Fig. 6.1a). It asks for the inductance you are looking for and for some constraints: the minimum and maximum number of turns, the minimum and maximum metal width and metal spacing and for the minimum and maximum outer dimension. You must also specify at which frequency the optimization has to be performed and which quality factor you want to maximize (the one at Port 1, the one at Port 2 or the differential one).
If you choose the option `Outer Dim.`, a similar dialog box open (Fig. 6.1b). This time, you want to get the optimum quality factor for a given area occupation, regardless of the inductance.
In both cases, after clicking on `Confirm`, the `Start Optimization` command is enabled. When you select it, the optimization starts and can take several minutes to complete (the looser the constraints the longer it takes to finish).
At the end of the optimization, the optimum inductor's characteristics are summed up in the Matlab workspace, like this:

```
Optimized Outer Base: 104.05 um
Optimized Metal Width: 5.00 um
Optimized Metal Spacing: 5.00 um
Optimum Number of Turns: 2.75
```

[1] That is, choose either `Inductance` or `Outer Dim.`.

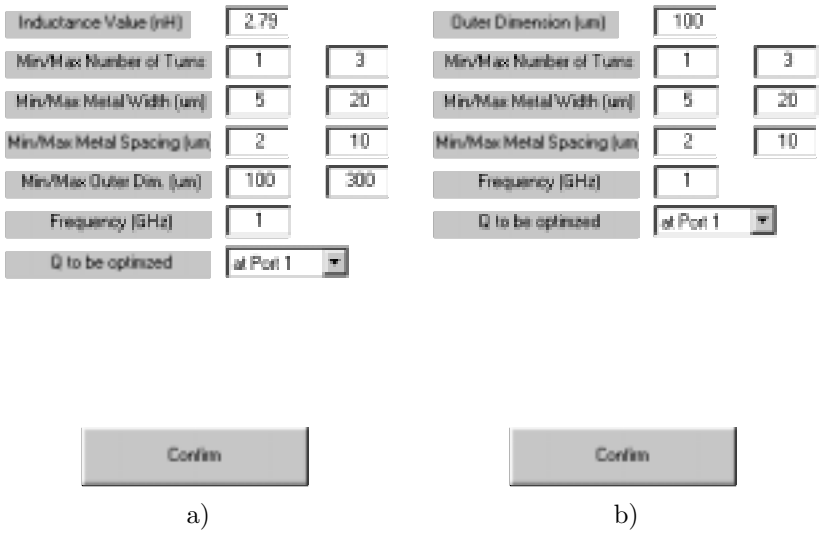


Figure 6.1: The dialog box asking for the optimization constraints: a) the target is the inductance; b) the target is the area occupation.

```
Total Inductance: 1.00 nH
Quality factor at 1.0 GHz: 2.84 (Port 1) -- 2.85 (Port 2) -- 2.85 (Diff)

Pi Model at 1.0 Ghz: Ls = 0.997 nH & Rs = 2.199 Ohm
                     Cs1 = 26.086 fF & Rs1 = 906.256 Ohm
                     Cs2 = 24.726 fF & Rs2 = 514.842 Ohm

Estimated Self-Resonance Frequency is over 40.00 GHz

>>
```

A figure depicting the optimized inductor layout also opens, just like for the single-point simulation (see Chapter 5). It has five buttons (`Zoom`, `Save`, `Export`, `Print` and `Close` to let you perform the main operations on the figure.
Sometimes the optimization can't reach the target because the target is out of the constraints. In that case, loosen the constraints and simulate the structures at the extremities of the range you have set up to see how far your target is from the constraints' range.
A word of caution: the higher the inductance, the lower the self-resonance frequency. So, if you choose a too high frequency, you must be aware that the inductor could be close to the resonance. At that point, the curve start to rise steepily and the inductance at low frequency could be very different from the one you get from the optimization. Make sure then, that the optimized inductor at the frequency of interest is far enough from resonance.

# Chapter 7

# HFSS

Drawing an integrated inductor, or for that matter any integrated device, can be very time-consuming. So, once you have found the optimum inductor you are looking for, it is much better to have a way to export it, without having to redraw it in the new environment. SpInd can do that. It exports the inductor's layout in three widely available formats (see next chapter). However, if you happen to use Ansoft HFSS for your electromagnetic analysis, then the `HFSS` menu is the best choice. This menu contains two commands:

- `HFSS 8.5` and
- `HFSS 9`.

The former command creates macros for HFSS version 8.5, while the latter produces scripts for HFSS version 9[1]. Each of them in turn has two options, which become active after that a simulation (see Chapter 3) or an optimization (see Chapter 6) has been performed. For `HFSS 8.5` the two sub-commands are:

- `Save Layout As A HFSS8.5 Macro` and
- `Save Only The Inductor As A HFSS8.5 Macro`.

(`HFSS 9` has identical sub-commands, provided that you replace "`HFSS8.5 Macro`" with "`HFSS9 Script`").
If you choose to save the whole layout, both with HFSS 8.5 and HFSS 9, a complete structure ready for simulation will be generated. That is, this structure will include, apart from the inductor, the substrate and oxide layers, the air layer above the chip, the boundary box and the excitation ports. If the layout is saved in HFSS 9, even the boundary conditions, the materials and the simulation parameters will be set. In that case, you'll only have to start the simulation, without any additional steps to take in HFSS. SpInd takes the burden of the all the simulation set-up.
However, this feature comes with some strings attached. In fact, I had to make some (debatable) choices in order to give you a product ready to use. For instance, since electromagnetic simulations need a current return path, I have placed a perfect conductor (i.e. the ground) on the bottom metal layer.

[1] HFSS 9 doesn't support the macro language of HFSS 8.5

Moreover, I have made a hole in this ground, underneath the inductor, which is one and a half time the inductor diameter wide[2].
If you only want to export the layout in HFSS, you can select the `Save Only The Inductor As ...` option. This way, the script will create only the inductor's layout and you'll have to complete the setup by yourself. Otherwise, you can also export the layout in DXF or GDS format and import it in HFSS (see also next chapter).
Once created, how do you execute the scripts in HFSS? For version 8.5, go to `File` ⇒ `Macro` ⇒ `Execute` and save when the macro finishes. Regardless of the option you have chosen, you must set up the materials and the boundary conditions by yourself in the HFSS 8.5 environment.
As for HFSS 9, select `Tools` ⇒ `Run Script`. When the script finishes, just save the file and click the `Analyze` button: you have nothing else to do, if you have chosen the `Save Layout As ...` option.

[2]For more details about the HFSS setup, see V. Minerva, *Analysis and Design of Integrated Inductor for RF Applications*, Ph.D. Thesis, Politecnico di Milano, Spring 2004.

# Chapter 8

# Export

Now, let's suppose that thanks to SpInd you have found the right inductor for your design and that you want to import it in Cadence or in an electromagnetic simulator like Ansoft HFSS or ADS Momentum, to verify if the results match well.
Drawing it by hand is very time-consuming and also prone to errors. That's why the `Export` menu has been included in SpInd .
The commands available in the `Export` menu are:

- `As CIF File`,
- `As DXF File` and
- `As GDS File`.

They are enabled after that a simulation (see Chapter 5) or an optimization (see Chapter 6) is performed. Let's suppose that you want to save the inductor's layout in CIF format. Then choose the `As CIF File` command. As you see, it contains two sub-commands: `CIF Options` and `Save Inductor in CIF Format`. The latter is enabled only after you have executed the former.
The former lets you choose whether you wanna save the layout's coordinates in hundredths or thousandths of microns, and in which geometry (`Polygon`, `Box` or `Wire`). The default options should work fine, in case you don't know which options to choose. With `Save Inductor in CIF Format`, a dialog box opens to let you save the layout in a CIF file.
The `As DXF File` command contains two sub-commands too: `Hierarchical DXF` and `Flattened DXF`. I recognize that the names are a bit misleading, since the real difference between the two is not about hierarchy, but about syntax. In other words, while the Autocad file created by `Hierarchical DXF` can be read by any program, because it complies with the DXF syntax, the one created by `Flattened DXF` can be read only by ADS Momentum and few other programs, because its syntax is a bit "looser". I don't wanna dissertate about DXF syntax, so it suffices to say that.
But why have I included the `Flattened DXF` command in SpInd ? Because if I save a layout with the `Hierarchical DXF` command, ADS Momentum can read all the layers, but then it imports only the top one, while with the other command it imports the whole layout, even though "flattened" on a single layer

(that's where the command's name comes from). Since an inductor is usually made of three layers (spiral, via and underpass), it takes a wink to "de-flatten" the inductor and analyze it with Momentum.
The last command, `As GDS File`, is composed of two sub-commands, as the others. They are `Save Technology File` and `Save Inductor in GDS Format`. The latter is enabled only after you have executed the former.
By selecting `Save Technology File`, a window opens, which contains two columns: `METAL NAME` and `LAYER NUMBER`.
SpInd takes the metal names from the technology file (see Chapter 4), while it names the vias simply as "Via_topmetalname_bottommetalname". In Cadence, names of vias' layers aren't usually assigned like that. Moreover, metal names could also be different from those in Cadence, if you have assigned them on your whim. Finally, SpInd has no clue to how layers are mapped on Cadence and simply numbers them in ascending order, starting from 1. If you look in your technology file, this information is available. So fill in the dialog box accordingly. When you click `Confirm`, another dialog box opens, which asks you to save this information in a TECH file. You can use this file when importing the GDS in Ansoft HFSS, for instance. This way, HFSS knows about the layers' thicknesses and can automatically create a tridimensional structure, without you doing that. After you have saved the TECH file, the `Save Inductor in GDS Format` command is enabled. Now, just clik on it and save the layout in GDS format[1].

[1] In SpInd , vias are simply equal to the overlapping area between the top and the bottom metal level; this naturally infringes the layout design rules, so that you'll have to redraw the vias in Cadence by yourself.

# Chapter 9

# Asitic

Asitic is a free software for the simulation of integrated passive devices, mainly inductors, courtesy of Ali Niknejad, a Ph.D from the University of California at Berkeley[1]. Its name stands for *Analysis and Simulation of Spiral Inductors and Transformers for Silicon ICs*. Because it's free and also rather accurate, at least if the "electromagnetic" mode is activated (see the site in the footnote for details), it's been used quite extensively for the design of integrated inductors. However, Asitic can be resource-consuming in its electromagnetic mode, while its results are a bit unsatisfactory in its "faster" mode, i.e. with analytical formulas. (Needless to say, that's where SpInd steps in).
This menu lets the user prepare the simulations for Asitic in the SpInd environment and takes back the results from Asitic for processing in SpInd . Of course, you can do all that within Asitic, but SpInd has the advantage of a graphical interface and let you generate Asitic scripts very quickly.
The commands available in the `Asitic` menu are:

- `Insert Asitic Parameters`,
- `Create Asitic Script`,
- `Load Asitic Results`,
- `Compare Asitic with SpInd` ,
- `Save Asitic Technology File` and
- `Set Asitic Directories`.

At the beginning, only the first, the third and the sixth commands are available. Let's discuss the last first. `Set Asitic Directories` lets you specify where Asitic is located on your computer.
By clicking on `Set Asitic Directories`, a window with three fields opens: `My Asitic Directory`, `Directory for Simulation Files` and `Directory for Optimization Files`.

[1] You can download Asitic at *http://rfic.eecs.berkeley.edu/~niknejad/asitic.html* after registering free.

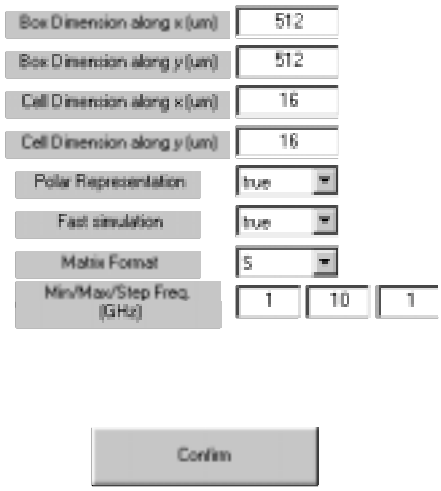


Figure 9.1: The dialog box asking for the Asitic parameters.

By default, `My Asitic Directory` contains a Windows-like path (on my computer, which runs on Windows, Asitic is nested inside Cygwin, a Linux-like environment for Windows; in fact, Asitic was originally developed on a Linux platform). I think that there should be no problem even if you have a Linux/Unix-like operating system; just enter the directory where Asitic is located.
The `Directory for Simulation Files` and the `Directory for Optimization Files` are instead in Unix form, so that no problem due to the operating system should arise. The former directory will store the Asitic scripts created by SpInd , while the latter was intended for optimization scripts to be run under Asitic. This feature has not been implemented yet, so just forget about it.
Once you have set up your Asitic directories, select `Insert Asitic Parameters` (see Fig. 9.1). The first four are the parameters Asitic needs to run simulations; they are about the box and cell dimensions on the XY plane. See the Asitic documentation on the website in the footnote for more details. Otherwise, just accept the default values.
We avail ourselves of this opportunity to remind you reader that, when you run Asitic, you must load a technology file with the same box and cell dimensions. By the way, the scripts you are going to create are unrelated to the technology file, so that you can run them with different processes as long as the constraint on dimensions are met.
The other fields in Fig. 9.1 allows you to choose the maximum and minimum frequency of sweep and the number of points to sweep and the output matrix format (`Y,Z` or `S`; `PI` returns the frequency-by-frequency equivalent circuit of Fig. 5.1). You can also choose if the output must be in polar representation or in real and imaginary part. The option `fast` is for the type of simulation to be run when launching the script; if set to true, analytical simulations are performed (fast but inaccurate); if set to false, electromagnetic simulations are performed (accurate but memory-hungry).
After clicking on `Confirm`, you can pass to the following command, `Create Asitic Script`, which is now enabled. As in Chapter 10 for the design maps,

you must enter the range of parameters to sweep and the number of points. A pop-up menu lets you choose the structure to analyze. Currently available are `sq`, the standard square spiral inductor (see Par. 3.1 for more details on this structure), `mmsq`, the multi-metal series-connected spiral inductor (Par. 3.5), `symsq`, the symmetrical square spiral inductor (Par. 3.9, `poly`, the polygonal spiral inductor (Par. 3.7) and `shsq`, the multi-metal shunt-connected spiral inductor (Par. 3.7).
When you have filled the fields in, click `Confirm`. Another window opens. It asks you whether you want to include the underpass in the Asitic analysis. (To be fair, this option is available only for certain structures, but it appears always on the monitor regardless of the inductor). Choose whether to include the underpass and then click the `Confirm` button and finally save your script in a LOG file.
At this point, you have created the scripts for Asitic; now, open Asitic and run them. Asitic stores the results in the simulation directory specified by the `Set Asitic Directories` command. Each simulated inductor is saved in a "`name_matrix_`nN`_w`N`_s`N`_d`N.txt" file, where `name` can be `sq`, `mmsq`, `symsq`, `poly` or `shsq`, depending on the structure you chose. The `N` capital letter must be replaced by the numeric value of $n$ (the number of turns), $w$ (the metal width), $s$ (the spacing) and $D_{out}$ (the outer dimension). As for `matrix`, it can be `s`, `y`, `z` or `pi`, depending on which output format you chose.
Select `Load Asitic Results`, which is always enabled, and load one of those text files. Three windows open: one plots the inductance, the other the quality factor (in both cases at Port 1, Port 2 and differential) and the third gives a summary of the geometric characteristics of the inductor you loaded.
The `Compare Asitic with SpInd` and the `Save Asitic Technology File` commands are now enabled. The former command analyze the loaded inductor with SpInd ; when the analysis completes, fourth figures appear on, as described in Chapter 5. The only difference is that now the results by Asitic are also plotted, so that you can compare the two programs. (My evaluation is partisan, but believe me when I say that SpInd performs much better than Asitic in the fast-simulation mode).
Finally, by selecting `Save Asitic Technology File`, you can use the Asitic technology file with SpInd in a later session.

# Chapter 10

# Design Maps

Under this menu are present several options to plot inductor's main parameters ($L$, $Q$ and $f_{SR}$) on the XY plane. Namely, on the abscissa is reported the outer dimension $D_{out}$, while on the ordinate is reported the metal width $w$. This way, the designer can see how electrical parameters change as functions of geometrical variables $D_{out}$and $w$. The other geometrical parameters ($N$ and $s$) are kept constant for each map. In fact, electrical parameters are weakly dependent on $s$ and, what's more, it is generally true that the smallest the spacing the better the inductor's performance. As for $N$, inductance and quality factor are heavily dependent on it, but designers usually need a very specific inductance, so that the number of turns they can use to get it is limited and therefore a large sweep of $N$ unnecessary.
It's more important, we think, to plot electrical parameters as function of $D_{out}$, since the area occupation is one of the major concerns for designers. As for $w$, it affects both $L$ and $Q$ significantly.
The commands available in the `Design Maps` menu are:

- `Set Design Maps Parameters`,
- `Plot Maps`,
- `Save Maps` and
- `Load Maps`.

When SpInd is launched from the Matlab prompt, only the `Load Maps` command is enabled, since no inductor has been selected yet. To select an inductor, see Chapter 3.
After selecting an inductor, the `Set Design Maps Parameters` command activates. By clicking on it, the `Design Maps Parameters` window opens and we can specify the variables to create the maps. Note that $s$ and $N$ are fixed, while for $D_{out}$and $w$ we need to set the minimum, the maximum and the step of the range to sweep. We must also insert the inductance we are looking for and the frequency at which the sweep is performed, since the electrical parameters are function of frequency and different results are obtained for different frequencies. After clicking on the `Confirm` button, the `Plot Maps` is enabled.
By selecting this command, a waitbar appears and simulations are performed within the range defined in the previous step. This operation can take several

minutes. When simulations are over, three maps appear automatically: an inductance map, a quality factor map and a self-resonance frequency map.
The contour plots of $L$, $Q$ and $f_{SR}$give precious information on how these parameters change according to the values of $w$ and $D_{out}$. The last two maps also contain the contour plot of the inductance we set in the previous step, so that we can see, for a fixed $L$, how $Q$ and $f_{SR}$vary. This way and based on the constraints he must deal with, the designer can choose the best trade-off on the inductance countour line between area occupation, quality factor and resonance.
Maps can be saved with the `Save Maps` command as *.mat or *.fig files. The file name must be entered only once; strings "_1", "_2" and "_3" are then attached to the file name for the $L$, $Q$ and $f_{SR}$map, respectively.
You can later retrieve the maps you saved with the `Load Maps` command (that's why is always enabled when SpInd is started). After the maps are loaded, plots don't pop up as before at the end of the sweep, but the `Plot Maps` command must be explicitly selected. If you load a MAT file, the sweep is made anew, while that is not necessary with the FIG files. So, it makes more sense to save maps in fig format.
Remember that the `Plot Maps` command executes the last option you entered. It can be either a MAT file or a FIG file or a simulation sweep resulting from the `Set Design Maps Parameters` command. So, if you wanna plot again the maps you plotted before the current ones, just select the appropriate command (`Set Design Maps Parameters` or `Load Maps`).