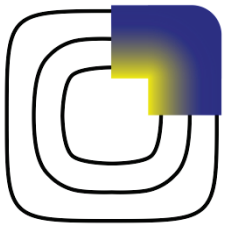
InductEx®



User's Manual

Version 4.30 - 2015

Coenrad J. Fourie

Stellenbosch University

Available at: www.inductex.info

This document dated 14 March 2015

Contents

1. Credits

InductEx (the code base, file translators, pre-processor and post-processing algorithms) is the product of the combined research and development efforts of:

- Coenrad Fourie
- Mark Volkmann
- Rebecca Roberts
- Thomas Weighill
- Pierre Lötter

InductEx also builds on the work done by Matton Kamon and Steve Whiteley (*FastHenry*), Paul Bunyk and Sergei Rylov (*Lmeter*), Angus Johnson and Bala Vatti (the polygon clipper unit) and Sergey Bochkanov (the ALGLIB project and SVD code).

Process-specific support with the assistance of Thomas Ortlepp, Olaf Wetzstein and Jürgen Kunert (FLUXONICS), Oleg Mukhanov and Alex Kirichenko (Hypres), Nobuyuki Yoshikawa, Yuki Yamanashi and Kohei Ehara (AIST STP and ADP) and Robin Cantor (Star Cryoelectronics).

Integration into LayoutEditor was done by Jürgen Thies (Juspertor).

Integration into Cadence design environment with the help of Vasili Semenov (Stony Brook University) and Masamitsu Tanaka (Nagoya University).

Quality assurance (by the patient Beta testers and bug reporters): Timur Filippov (Hypres), Pascal Febvre and Romain Collot (IMEP-LAHC), Felix Jaeckel (University of New Mexico), Xizhu Peng and Naoki Takeuchi (Yokohama National University), Damian Steiger (ETH Zurich), Oliver Brandel (IPHT Jena) and Steve Kaplan, as well as Mark Volkmann, Rodwell Bakolo and Hein van Heerden (in-house).

InductEx is based upon work supported financially by the South African National Research Foundation, grant numbers 69006, 78789 and 93586.

2. Introduction

Before you read any further, or do any experiments, take note:

Numerical inductance calculations are only as reliable as the models used to obtain them. No solution is absolutely correct – some are just better than others. You are strongly advised not to put too much trust in a calculation result until you are familiar with modelling and interpreting results, and have verified one of your own extractions against a known measurement result for a similar structure.

If you have limited experience in modelling layouts for inductance extraction, please study the examples described in this manual carefully before you attempt to do your own extractions. A simple modelling mistake (such as inverting the polarity of a port on a layout, or making a connection mistake in a circuit netlist) can lead to wildly inaccurate solutions.

InductEx [1]-[5] was developed to enable VLSI circuit designers to extract the inductance of complex 3D structures in superconducting integrated circuits, but also supports normal conductivity and the calculation of inductance in simpler structures such as SQUID loops or wires. Current distribution in, and magnetic fields in and around conductors can also be computed. Although early versions were focused on inductive substructures, *InductEx* is now aimed at full-gate extraction [5]. *InductEx* functions well as a stand-alone command line programme that accepts GDSII or CIF input files created by any layout software, but was carefully designed to be integrated with CAD software in the same way that *Lmeter* [6] is done through its own IXI interface.

InductEx 4.30 is a front and back-end to the inductance extraction software *FastHenry* [7], of which the adapted superconducting version is freely available via the Whiteley Research website (http://www.srware.com). *InductEx* uses *FastHenry* as the field solver, which it can call automatically to do extraction. However, the multi-terminal extraction capabilities of *InductEx* require a capability-upgraded *FastHenry* (available in both 32-bit and 64-bit versions as versions 3.1wr+su32 and 3.1wr+su64). We have improved the computation of the sparse matrix preconditioner in *FastHenry* significantly, so that *InductEx 4.30* is up to 5 times faster than earlier versions of *InductEx* for calculations with 10,000 to 15,000 segments.

If you are familiar with *Lmeter*, then *InductEx* will be easier to get used to. If you are only interested in calculating the inductances of components in standard cells, *Lmeter* is a fast and well-proven alternative. Use *InductEx* if you have difficulties with *Lmeter*, require inductance calculations where holes in a ground plane, the absence of ground planes, or very narrow lines complicate calculations with *Lmeter*, or where the complexity of your layout warrants the use of a more powerful solver.

Your use of *InductEx* is subject to the licence terms made available on the *InductEx* website.

InductEx was developed for and tested to interface well with LayoutEditor and LASI. It also interfaces easily with XIC, and with the correct script files it works directly from Cadence Virtuoso. It also works within LayoutEditor.

Finally, *InductEx* was developed to allow inductance extraction of complete circuit cells. The intention is to allow users to work directly on "ready-for-fabrication" cells without the need to alter layouts during modelling. To this end, *InductEx* supports auxiliary layers and operators, and cells that pass extraction may be sent for fabrication without the need to strip out modelling text or objects. The resistance of inductors with normal metal sections can be calculated, but not specified separately in netlist files.

3. Limitations

InductEx v4.30 is free to use, although such use is subject to the license terms. Although updates will be made available to fix serious bugs if any are reported, no active user support is offered for this version of *InductEx*. This version is also limited to calculation models with less than 15,000 segments, 6 ports and 12 inductors.

Versions 5.0 and later are more powerful, even faster, support multi-threading, and have no limitations on model or circuit size. Consider purchasing such a version via the *InductEx* website if you need to solve models that exceed the capabilities of *InductEx v4.30.*

4. Installing InductEx

Download the zipped binary distribution file for your system from the *InductEx* website (the direct link is http://www0.sun.ac.za/ix/?q=ixfree), and unpack the binary files to your working directory, or any directory that you prefer.

Then add the directory to the path. If, for example, your install directory is c:\usr\local\bin, you can (permanently) set the path in MS Windows from the command prompt with:

```
Setx path "%path%;c:\usr\local\bin"
```

The install directory can also be added to the path by selecting *Advanced System Settings* from the *Control Panel*, opening the *Advanced* tab and clicking on *Environment Variables*.

5. Layer definition file

The layer definition file is the most important aspect of any calculation setup, because it describes to *InductEx* how a layout drawing (which is essentially a set of disconnected two-dimensional objects) should be modelled in three dimensions with the correct layer sequence and connections. As such, it describes the fabrication process, with the important caveat that the **description is focused on creating numerical models that have the same geometry as fabricated circuit structures**, but may differ (substantially) from the actual fabrication sequence.

The layer definition file controls or configures *InductEx* by defining both the modelling and process technology parameters used to build extraction models. **Modelling parameters** are not derived from the actual fabrication process, and control segment size, ground plane modelling, optional segment blanking, etc. **Process technology parameters** describe the actual layers in terms of model construction order (which is not necessarily the same as the fabrication order), dimensions, mask-to-wafer bias and physical parameters such as penetration depth.

There is no limit on the number of configuration files, as the user controls which file is used when *InductEx* is called.

InductEx is distributed with standard layer definition files for the IPHT RSFQ1F, Hypres 4.5 kA/cm^2 and AIST's STP 2.5 kA/cm^2 (standard) processes. Until you are an experienced user, it is strongly recommended that you do not build a process file from scratch, but rather adapt one of these files to suit your needs.

Table 1 shows the parameters available in the layer definition file, and what they control. The parameters are defined inside a control block starting with

```
$Parameters
```

and ending with

```
$End
```

Parameters are defined as:

```
Parametername = value
```

Every layer has to be defined as well. Layers are defined in control blocks starting with

```
$Layers
```

and ending with

```
$End
```

Layer parameters are defined similarly to global parameters, and are listed with functionality in Table 2.

Table 1: Layer definition file global parameters.

Parameter name	Default value	Function
`Units`	`1e-6`	The unit size in meters in the layer definition file. The default is 1 μm. **(Only change this if you are an advanced user).**
`CIFUnitsPerMicron`	`100`	This sets the number of units per micrometre in CIF files. The default is 100. However, XIC uses 1000 unless forced to revert through the "strip for export" function.
`Frequency`	`10e9`	The frequency of the port excitation voltages. For purely superconductive circuits, this does not influence the results. With resistive components, the frequency dependent skin depth affects current distribution in the normal conductors.
`GapMax`	`2.51`	Sets the maximum *x* and *y* dimensions of any segment in Units. Larger segments are subdivided. For IPHT's RSFQ1F process, 2.5 is a good value. Use 2.0 for Hypres's 4.5 kA/cm^2 process, and 1.0 for AIST's STP and ADP processes. Larger values speed up calculations, but cause higher inductance results. Always compare calculations against a known result after you adjust GapMax.
`AbsMin`	`0.01`	Sets the test sphere around a current density node within which it is mapped to a FastHenry geometry node (should NOT be zero). **(Don't change this value without consulting the InductEx team first).**
`GPOverHang`	`2.5`	Sets the distance in Units between the edge of any object and the edge of the ground plane that is generated to cover the smallest necessary area beneath structures. A smaller value results in artificially higher calculation results. 2.5 is a good value for all processes, but 7.5 is recommended when lines crossing holes in the ground plane are investigated (the ground plane automatically wraps around holes).
`ProcessHasGroundPlane`	`TRUE`	Boolean value to indicate ground plane in process. Usually only FALSE for monolayer HTS circuits.
`CropGP`	`TRUE`	Boolean value to indicate that ground planes should be cropped to within GPOverHang of the union of all non-ground plane structures (this reduces unnecessary segments).

ZSegsToEC	FALSE	Boolean value to force InductEx to replace z-directed segments with electrical connections. This removes segment size matching between layers, and generally leads to reduced memory requirements and improved speed, especially in large cells. **Use with care.**
LastDieLayerOrder	-	Indicates the order of the last layer fabricated on the wafer.
GPLayer	-	Indicates the GDS number of the ground plane.
TermLayer	-	Indicates the GDS number of the layer on which terminals are drawn.
TextLayer	-	Sets the text layer.
BlankAllLayer	-	Sets the global blanking layer number. No objects or parts of objects falling within blanking layer objects are segmented, which allows user-defined model trimming.
BlankXLayer	-	Sets the *x*-direction blanking layer number. Any objects or parts of objects falling with *x*-blanking layer objects are only segmented in the *y* direction.
BlankYLayer	-	Sets the *y*-direction blanking layer number. Any objects or parts of objects falling with *y*-blanking layer objects are only segmented in the *x* direction.
Lambda	0.09	Default (global) value for London penetration depth. Can be overrided per layer.
Sigma	10	Bulk conductivity of resistive layers (global). Calculated as the inverse of resistance per square times layer thickness for thin films, and expressed as Ω^{-1}Units^{-1}. For example, if unit size is in μm, and layer thickness is 80 nm, a conductivity of 10 $\Omega^{-1}\mu m^{-1}$ equals a sheet resistance of 1.25 Ω/□.
HFilaments	1	The global value for the number of height filaments into which segments are divided. Can be overrided per layer.
Colour	1	Global default for AutoCAD DXF layer colour.
TerminalInRange	1.0	Sets the distance (in Units) from the rectangular boundary around any terminal object within which a text label will be accepted as "linked" to that object.
ResZero	1e-3	Sets the smallest calculated resistance value printed to the output – below this the resistance is considered zero and an element is considered to be purely inductive.
JoinShortSegments	FALSE	Activates node optimisation (stitching segments together length-wise when orthogonal segments are blanked out).
BlankAllCutsGP	FALSE	If TRUE, BlankAllLayer also blanks the ground plane.
DataTypeNotZero	FALSE	If TRUE, GDSII objects with DataType <> 0 are also accepted.

Table 2: Layer definition file parameters for layers.

Layer parameter name	Default value	Function
`Number`	–	This parameter is required, and sets the layer number associated with this layer (which coincides with the GDSII layer number if GDSII input is used). Since version 4.26, this number can range from 0 – 255.
`Name`	–	This parameter is required, and sets the name of the layer in DXC and CIF input files (to be supported), as well as in terminal label definitions. The name is also applied to output files for visualization.
`Bias`	`0`	This parameter is optional, and defines the mask-wafer offset for the layer. The value is added to the boundaries of any object on the layer (positive values "grow" the object, while negative values "shrink" the object). Non-zero values are only necessary for Hypres's processes. For other processes it is a calibration tool.
`Thickness`	–	Required parameter that defines layer thickness in Units.
`Lambda`	`Global lambda`	Optional parameter that defines the London penetration depth of the layer. If undefined, the global value is used.
`Sigma`	`Global sigma`	Optional parameter that defines the bulk conductivity of a layer (see the description under global parameters in Table 1 for more detail). If undefined, the global value is used.
`Order`	–	Required parameter that sets the construction order of the wafer. **The lowest layer must start at 0, and order numbering must be sequential.** If one layer is higher than another in the wafer, it must have a higher order. Different layers cannot share the same order. **If you add layers to a process file, always verify the order numbers, and set the global value** `LastDieLayerOrder` **to the last physical layer's order**.
`Mask`	`0`	Sets the mask polarity of the layer. The options are: `1` Positive layer (deposited where objects are defined). Normally all superconductive and resistive layers. `0` Layer does not contribute to the vertical height of the wafer – typically used to define auxiliary or operational layers. `-1` Negative layer (etched away where objects are defined). Normally all isolation and anodization layers. `-2 | -3` Are reserved for the blanking layers `-4` Is reserved for the terminal layer
`Filmtype`	–	Sets the film type of the layer. The options are: `S` Superconducting layer `N` Normal (resistive) layer that is not segmented (used for instance for superconductive circuits where only inductance is extracted, and resistive components should be ignored during calculation). `R` Resistive layer that is segmented `I` Isolation layer

		A Auxiliary layer C Cut layer (everything underneath ablated away)
IsGP	FALSE	Used for auxiliary ground planes in high layer count processes such as ADP. When set to true, this layer is cropped similarly to the main ground plane if the global parameter CropGP is true.
IDensity	-	Optional parameter that sets the current density of a superconductive Josephson junction layer. This is multiplied by the area of objects in the layer if it exists between the terminals of a port to give junction critical current. This parameter can be used in other processes to yield any linear relationship to the area of a layout object of interest.
PlanarModel	0	Sets the planarization model applied to this layer. The options are: 0 No planarization. 1 Full planarization up to last layer before this one. If set to 1 for an isolation layer directly below a metal layer, it creates depressions at vias through this isolation layer. This implements the complemented caldera planarization method used by AIST for the ADP process [7]. If set to 1 for a metal layer, vias to lower layers will be studded, as envisioned for the MIT Lincoln Lab 10-layer process. 2 (and higher) Reserved for future models.
HFilaments	1	Sets the number of height filaments into which segments on this layer will be divided. This is only applied to superconductive layers. More filaments result in exponentially longer calculation times, but it is suggested that this value is set to 2 if a layer is thicker than 1.5 times the penetration depth, and to 3 if a layer is thicker than 2.5 times the penetration depth. For the standard processes, it is of no benefit to exceed 3.
GapMax	Global GapMax	Optional parameter that sets the maximum *x* and *y* dimensions of any segment in Units (see the description under global parameters in Table 1 for more detail). If undefined, the global value is used.
ViaBypass	FALSE	When set to true for a conductive layer of mask type -1 or +1, via etching through an isolation layer directly above this layer will continue to the isolation layer directly below this layer if no conductive object is in the way. This is used for example to propagate vias through I1B past R2 in the Hypres process when R2 is included for resistance modelling.
Colour	1	Sets the DXF colour of the layer for visualization purposes. The range is 0 to 255 (indexed colours). Use different values for each superconductive layer to tell them apart during visualization.
LayerAdd	-	GDS/layer number of any lower order layer to be added to this layer. Useful for layer operations, for example adding layer BC to layer RC in the AIST STP process to allow via continuity. Up to 10 layers may be added to any layer.
LayerSub	-	GDS/layer number of any lower order layer to be subtracted from this layer. Useful for layer operations, for example subtracting resistive layer R2 from via layer I1B in the Hypres 4.5 kA/cm^2 process (this prevents vias from upper wiring layer M2 to resistors

	from shorting to lower wiring layer M1 when InductEx removes normal metal layers during segmentation). Up to 10 layers may be subtracted from any layer.

Layer addition and subtraction precedence is: 1 – `LayerAdd`; 2 – `LayerSub`.

Apart from parameter and layer definitions, *InductEx* allows the user to define specific operators that may be added to layouts to control modelling. These operators, if used effectively, can reduce segment count and solution time, but need to be used with care. Operator parameters are listed in Table 3.

Every operator has to be defined in the layer definition file. Operator control blocks start with

```
$Operator
```

and end with

```
$End
```

Table 3: Layer definition file operator parameters.

Operator parameter name	Default value	Function
`Name`	–	This parameter is required, and sets the name of the operator as it is used in text labels on layouts.
`Type`	`UD`	Sets the operator type. The options are: `UD` Undefined – does nothing `MR` Make rectangle – Changes all polygons in the layers defined by `LayersRem` and within which the operator text label falls to rectangles. `EC` Electrical connection – Connects nodes on the layers defined in `LayersConnect` electrically (without segments), and removes objects in layers defined by `LayersRemove` within which operator text labels fall. Electrical connections are only made between nodes falling inside the polygon with smallest area in any of the layers defined by `LayersRemove`. `OD` Object delete – Removes objects in layers defined by `LayersRemove` within which operator text labels fall. Useful to get rid of excess structures, such as multiple ground planes when these do not influence extraction results. `LD` Layer delete – Removes all objects in layers defined by `LayersRemove`. Useful to get rid of excess structures, such as multiple ground planes when these do not influence extraction results.
`LayersTransform`	–	Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on which objects should be transformed by the operator.
`LayersRemove`	–	Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on which objects should be removed

		by the operator.
`LayersConnect`	-	Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on which nodes should be electrically connected by the operator.

6. Layout input file requirements

A light-weight text-based input file format (IXI) was developed for InductEx, and a converter from Cadence Virtuoso to IXI has been implemented. The IXI input format also makes it easy to generate simple inductance experiments by hand.

However, *InductEx* was primarily developed to processes structures in the GDSII stream file format. This is an industry standard binary format, and most layout tools can convert layouts to GDSII. *InductEx* also supports input processing from ASCII-format CIF files. Autocad DXF compatibility will be added in the future.

In order to provide port/terminal information to *InductEx*, a TERM layer is needed. You can use any layer, as long as it is called TERM, and the layer number or index corresponds to the parameter `TermLayer` in the layer definition file.

Terminals are drawn as polygons, boxes or paths in the terminal layer. It is not necessary to draw terminals over vias or Josephson junctions, as *InductEx* will automatically use the via or junction boundaries to determine the terminal dimensions. Therefore, it is mostly only necessary to draw terminals on the edges of input/output lines or on bias lines (we shall call them "line terminals". Collapsed boxes (zero width) can be used to draw line terminals, but these do not conform to proper GDSII protocol and might not be exported correctly by most programmes. When using GDSII files, it is better to draw line terminals as paths or 2-point polygons on the TERM layer. The path width is immaterial, as *InductEx* only uses the centre line as the actual terminal (thereby "forcing" a zero width path). Path width is therefore only used as a visual aid during layout.

6.1. IXI

The IXI input file format is text based and human readable, and uses a flat hierarchy. It was developed to allow an easy interface from Cadence Virtuoso while maintaining readability.

Every layout object is represented by a polygon with an unlimited set of (x, y) coordinate pairs and a layer name (rather than a layer number). Boxes and paths are also represented as polygons. Coordinate values are in drawing units, as specified in the layer technology file. Optionally, unit sizes can be specified to override the layer technology file. Object identifiers are preceded by the `$` character, and object declarations are terminated with an end command. The file is ended with `$EOF`.

Drawing objects are defined as:

```
$POLY
layername
(x1, y1)
(x2, y2)
...
(xN, yN)
(x1, y1)
```

```
$EOP
```

Terminal layer polygons may have the terminal number and text directly beneath the layer name:

```
$POLY
TERM
number
textstring
(x1, y1)
(x2, y2)
...
(xN, yN)
(x1, y1)
$EOP
```

All text labels (to identify ports/terminals or operators) are represented with an (*x*, *y*) coordinate pair and a text string, currently limited to 40 characters:

```
$TEXT
textstring
(x, y)
$EOT
```

The (optional) unit size in metres is defined as:

```
$UNITSIZE
Real value;
```

or

```
$UNITSIZE
Real value
$EOR
```

The (optional) number of drawing units in the unit size is defined as:

```
$DUPERUNIT
Integer value;
```

or

```
$DUPERUNIT
Integer value
$EOI
```

6.2. GDSII

For GDSII input files, *InductEx* currently requires all object DATATYPE and text label TEXTTYPE values to be equal to zero. This is easily defined in *LASI* (by layer), *XIC* and *Cadence*. If, for any reason, DATATYPE or TEXTTYPE values other than zero should be supported, you may set the `DataTypeNotZero` parameter in the layer definition file to `TRUE`. If you need more specific support, you may contact us with a request.

InductEx also requires GDSII file structures to be sorted according to ascending rank.

6.3. CIF

CIF file read-in was developed for interface with LASI.

CIF allows easy manual changes to object coordinates because it is ASCII-based. It also allows you to define a simple layout in text format by hand (without the need for a circuit layout programme). However, CIF is inferior to GDSII. If your layout editor supports both, you are advised to always use GDSII rather than CIF.

When *InductEx* processes CIF input files, it only uses the CIF layer names to obtain the layer numbers (for compatibility with GDSII inputs). CIF layer names therefore do not have to correspond to the layer names in port/terminal labels, as those are defined in *InductEx's* layer definition files (.ldf). When exporting to CIF format from *XIC*, select index-based structure references (which uses the layer number instead of layer name).

6.4. DXC

DXC file read-in was developed for interface with Cadence.

Support for the *Lmeter*-specific DXC input file format was added to *InductEx* to enable integration into the Cadence design environment as used by superconductive electronic circuit designers. In its current implementation, *InductEx* acts as a one-to-one replacement for *Lmeter* in the Cadence environment. This allows the reuse of all *Lmeter* scripts (except when executing the solver), but limits port modelling to that supported by *Lmeter*.

Therefore, when using *Lmeter* scripts from the Cadence environment, ports cannot be defined as with positive and negative terminals linked to different terminal objects (see Section 4.4). If such modelling is required, layouts should be exported to GDSII format and *InductEx* executed separately.

6.5. Port / Terminal declarations

Ports or terminals are declared in the text layer, and their geometries defined in the TERM layer. The (*x*,*y*) coordinate of a text label is used to match the port text to a structure in the TERM layer. Syntax is:

```
Pname[s {+ / | \ -}] b [c]
```

with the first letter always "`P`". The port name may be up to 20 characters long. If the optional suffix `s` is in {`+ -`}, it indicates that this is only one terminal of the port and specifies the polarity (positive or negative). If `s` is in {`\ | /`}, the port is a virtual cut. `b` is the name of the layer on which the terminal (defined by the polarity) is located, or the positive terminal if the polarity is not specified. If polarity is not specified, `c` is required and is the name of the layer on which the negative terminal is located. For a virtual cut port, `c` defines the axis along which the port is connected to a conductor (*x* or *y*, which should coincide with the predominant axis of current flow) and the direction of the positive terminal.

Examples of valid labels are:

`P1 M2 M0`	(Port 1, with the positive terminal in layer M2 and the negative in M0)
`PTWO M1 M0`	(Port TWO, with positive terminal in layer M1 and the negative in M0)
`P3+ M2`	(Positive terminal of Port 3, in layer M2)
`P3- M0`	(Negative terminal of Port 3, in layer M0)

`P4 YBCO +y` (Port 4, a virtual cut through a conductor in layer YBCO, with current flow and port connection along the *y* axis. The positive terminal is in the positive direction along the *y* axis.)

`P5 M3 -x` (Port 5, a virtual cut through a conductor in layer M3, with current flow and port connection along the *x* axis. The positive terminal is in the negative direction along the *x* axis.)

If the port text label coordinates fall inside or on the boundary of an object in the TERM layer, this is used as the port definition. Note that only Manhattan-type objects are supported as terminals.

When no TERM layer object can be linked to the port label coordinates, and if the positive and negative terminals where defined in the same label, InductEx searches for a via object (with MASK = -1) that encloses the label coordinates and is ordered between the superconducting layers on which the terminals are declared. If more than one via object satisfies these conditions, the one with the smallest area is used. When such a via is used as a terminal object, it is moved to the TERM layer (and should show up as such when a GDSII output file is generated with the –d option). For convenience, it is therefore not necessary to define a terminal layer object over a Josephson junction when it is to be used as a port. **However, all vias changed to terminals by InductEx are converted to rectangular objects** to accommodate octagonal or circular junctions. Furthermore, all polygons on isolation/anodization layers (with MASK = -1) between the upper and lower terminal layers, and within which the terminal layer text coordinates fall, are also converted to rectangles encompassing their furthest coordinates. This is done to reduce the segmentation requirements of junctions, specifically octagonal and circular junctions.

Terminal geometries are restricted to Manhattan structures, but may be defined anywhere in on the edge or inside of a superconductive layer object.

Currently, *InductEx* also accepts text labels on layers other than the text layer when reading in terminal declarations. However, terminal text must be placed in the highest ranking cell of a layout.

For instances where text labels are not supported by a layout tool (such as *WaveMaker*), or where it is required for convenience to alter port labels without the use of a layout tool, port terminals can be defined in a text file. This is discussed in Section 4.5.

6.6. Port label input file

When a port label input file (in ASCII text format) is specified with the `-p` parameter, port labels (both text and coordinates) are read in from the file. This overrides any port labels that may have been read in from the geometry input file. The file may have any suffix, although `.txt` is preferable.

Port declarations follow the rules set out in Section 4.4, and the file syntax is:

```
Plabel; (x, y)
```

where `Plabel` is the port label that contains the port name and positive and/or negative terminal layers as described in Section 4.4. The label coordinates within the layout are at `x` and `y`, which is in `Units` as defined in the layer definition file.

Examples of valid port label definitions are:

`P1 M2 M0; (0, 25.5)` (Port 1 between layers M2 and M0, at x=0; y=25.5)

```
PIN+ M2; (10, 20)
```
(Port IN, with positive terminal on M2, at x=10; y=20)

6.7. Operator declarations

Operators are declared with text labels (on any layer). Operators act at the coordinates of the text label. Syntax is:

```
@Name
```

with the first character always "@". The name of the operator must match (case-insensitive) that of an operator declared in the layer definition file. If no match is found, the operator is ignored.

7. Inductance calculation with InductEx

Examples shown in this chapter, as well as more detailed examples not discussed here, are available on the *InductEx* website for download.

7.1. Circuit netlist file

In order to run an inductance calculation with *InductEx*, a circuit netlist file is needed. This netlist file is a subset of the Spice electrical netlist. The netlist should contain all the inductors and mutual inductors in the circuit model, and the values assigned to each will be used as the "design value" during *InductEx* post-processing.

The netlist file must end with the port definitions for the extraction setup.

Valid netlist elements are:

```
L1      node+  node-   designvalue
Lname   node+  node-   designvalue
Kname   Lname1 Lname2  designvalue
P1      node+  node-
Ptwo    node+  node-
PAny_name  node+  node-
```

For compatibility with JSIM, keep node names numerical. Design values are in pH. Mutual inductors are specified with `K` (coupling) for compatibility with Spice, although *InductEx* writes the output as a mutual inductance in pH.

A note on resistivity: *InductEx* can calculate the resistive component of an inductive structure that contains normal metal segments. However, resistors cannot yet be specified separately in the circuit netlist file. Even if the impedance of a resistor is of interest, it should be defined in the netlist as an inductor. *InductEx* is primarily an inductance calculation tool, and resistance is seen only as a parasitic element. In a practical superconductive circuit, the inductance of shunt resistors is of interest, while the resistance can be determined more accurately with analytical methods.

7.2. Minimum setup requirements

To execute *InductEx*, you need an input geometry file (GDS or CIF, since DXC and DXF files are not yet supported), a layer definition file, a circuit netlist and a version of *FastHenry* that supports superconductivity.

InductEx can be executed from the command prompt or terminal, and simple examples are listed in the next section.

The minimum command line statement to execute *InductEx* is:

```
inductex structurename.gds –l layerdefinitionfile.ldf –i name.inp -fh
```

The first parameter is the geometry input file. The -l switch identifies the layer definition file. The -i switch precedes the name of the FastHenry model file, while -fh forces a call to FastHenry.

If the -fh switch is omitted, *InductEx* terminates after dumping the .inp model file. The user can then run *FastHenry* manually.

When used as above, *InductEx* searches for the circuit netlist file by using the name of the geometry input file (thus: `structurename.cir` or `structurename.js`). However, any other circuit netlist file can be specified with the `–n circuitfilename.cir` switch.

If it is necessary to see the adjusted geometry (after mask-to-wafer bias adjustments were made), *InductEx* can be forced to dump the result to a GDSII file with the -d switch, followed by the name of the output file.

7.3. Single inductor example

This example is available on the *InductEx* website as *ex1*.

Consider the calculation of the inductance of a single, straight inductor above a ground plane (a problem best solved analytically [8], but used here due to its simplicity). A cross-sectional view of such a microstrip line above a ground plane is shown in Figure 7.1.

For our example, we use a microstrip line of length 100 μm and width (W) 10 μm. Line thickness (t_1) is 0.25 μm, ground plane thickness (t_2) is 0.2 μm and dielectric isolation is thickness (h) is 0.15 μm. The analytical solution, assuming an infinite ground plane, is 3.897 pH.

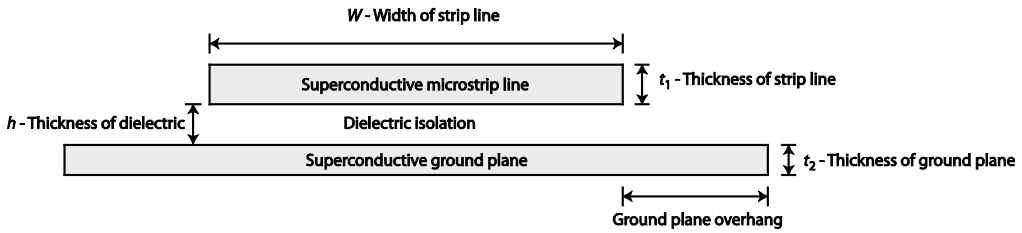


Figure 7.1: Cross-sectional view of superconductive microstrip line and ground plane

Calculating the same inductance with *InductEx* requires us to draw a 100 × 10 μm line in a CAD tool. For the example discussed here, and shown in Figure 7.2, LASI is used. Four layers are defined: M0 as the ground plane, I0 as the dielectric isolation layer, M1 as the conductor, and TERM as the terminal layer. The microstrip line is drawn as a box in M1, and the terminals are added as paths of width 1 μm at the furthest edges of the microstrip line. Text labels are placed with their coordinates exactly on the edges of the microstrip line, inside the terminal paths, to identify the ports. The ground plane is not drawn, as *InductEx* uses a negative ground plane representation – it exists everywhere except where it is drawn. The next step is to export the layout to GDS or CIF file formats (from LASI; DXF and DXC formats may be used from other CAD tools as soon as these are supported). For this example, we use `lman1.gds`.

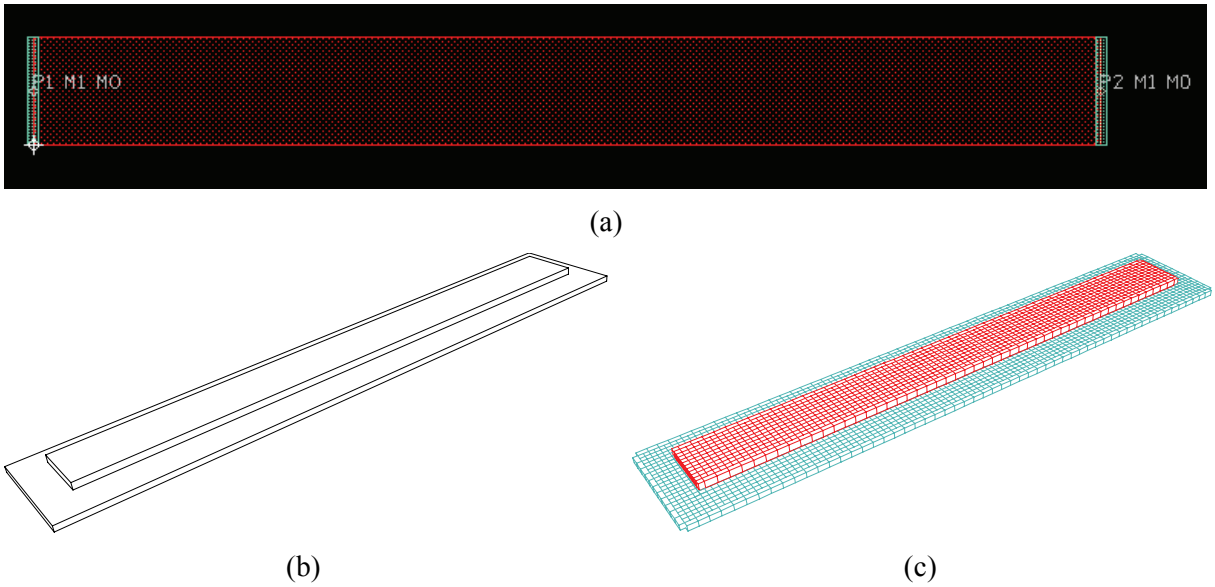


Figure 7.2: (a) LASI screenshot of microstrip line layout with two ports, (b) three-dimensional drawing showing finite ground plane, and (c) InductEx segmented model. Vertical dimensions in (b) and (c) enlarged 5 times for clarity.

A circuit netlist file is required to link ports to inductors. For simplicity, we use the same project name, with the extension `.cir` (although *InductEx* will also search for `.js` files if it cannot find a `.cir` file). The netlist file `lman1.cir` for a single inductor with two ports (one at each node) is:

```
* spice netlist for InductEx user manual example 1
* Inductors
L1      1    2    10
* Ports
P1      1    0
P2      2    0
.end
```

Finally, we need a layer definition file that defines layer dimensions and simulation variables. *InductEx* will accept any file extension, but we use `.ldf` by default. The layer definition file `ixman1.ldf` is:

```
*Layer Definitions for InductEx
* manual - ex1
*
$Parameters
Units              =  1e-6
CIFUnitsPerMicron  =  100
GapMax             =  2.0
AbsMin             =  0.025
GPOverhang         =  5.0
ProcessHasGroundPlane = TRUE
LastDieLayerOrder  =  2
GPLayer            =  17
TermLayer          =  19
TextLayer          =  18
HFilaments         =  1
TerminalInRange    =  1.0
$End
*
```

```
* I0
$Layer
Number      =      2
Name        =      I0
Thickness   =      0.15
Order       =      1
Mask        =     -1
Filmtype    =      I
$End
*
* M1
$Layer
Number      =      5
Name        =      M1
Thickness   =      0.25
Lambda      =      0.09
Order       =      2
Mask        =      1
```

```
* LAYERS
*
* M0
$Layer
Number      =      17
Name        =      M0
Thickness   =      0.2
Lambda      =      0.09
Order       =      0
Mask        =     -1
Filmtype    =      S
HFilaments  =      2
Colour      =      135
$End
*
```

```
Filmtype    =      S
HFilaments  =      3
Colour      =      10
$End
*
* TERM
$Layer
Number      =      19
Name        =      TERM
Order       =      3
Mask        =     -4
$End
```

The parameters are discussed in Section 0. Note that the ground plane uses a negative mask, and that the TERM layer uses a mask value of -4. Since we cannot model an infinite ground plane, the ground plane overhang (`GPOverhang`) is specified as 5 μm (given the dimensions of the problem, this is sufficient).

InductEx is executed from the command prompt with:

```
inductex lman1.gds –l ixman1.ldf –i lman1.inp -fh
```

The screen output under Windows is shown below. Under Linux and OS X, FastHenry is executed in the same terminal and the output is more verbose. A summary of the *InductEx* output is written to the file sol.txt.

```
c:\usr\local\bin>inductex lman1.gds –l ixman1.ldf –i lman1.inp -fh
InductEx v4.26 (21 March 2014). Copyright 2003-2014 Coenrad Fourie
lman1.gds -l ixman1.ldf -i lman1.inp -fh
Spice netlist lman1.cir read. Totals: L = 1, k = 0, P = 2.
GDS file lman1.gds read: db units in 1E-0009 m, 0.001 units per user unit.
1 structures read. Reduced 3 objects to 2 polygons and 2 terminals.
Techfile ixman1.ldf read: Units in 1E-0006 m. AbsMin=0.025 GapMax=2
DIAG preconditioner.
FastHenry version 3.0wr+su64 found.
Total unique loops identified in netlist = 2
Terminal blocks = 2; Labels = 2; Extracted Ports = 2

Port                  Positive terminal    Negative terminal
P1                    M1,   line along y;  M0,   same as "+" terminal.
P2                    M1,   line along y;  M0,   same as "+" terminal.
Minimum filaments in FastHenry = 4323

Impedance   Inductance [pH]     Resistance [Ohm]  AbsDiff   PercDiff
Name      Design    Extracted Design    Extracted (L only)  (L only)
L1        3.89700   3.89751   --        --        +0.0005   +0.01%
Deallocating memory.
Cycles found in 0.000 seconds.
SVD solution in 0.000 seconds.
Job finished in 5.492 seconds.
```

The calculated inductance compares very well to the analytical result, but is a strong function of segmentation size, filament numbers and ground plane overlap. For example, if the overhang is reduced to 2.5 μm, L1 is calculated as 3.911 pH (in 3.96 seconds). If height filaments for M0 and M1 are then reduced to 1 each, L1 is calculated as 4.2253 pH (in 1.36 seconds), and if the segment size is increased to 2.5 μm (`GapMax = 2.5`), L1 becomes 4.234 pH (in 0.41 seconds).

7.4. Single inductor example with non-Manhattan geometry and off-edge ports

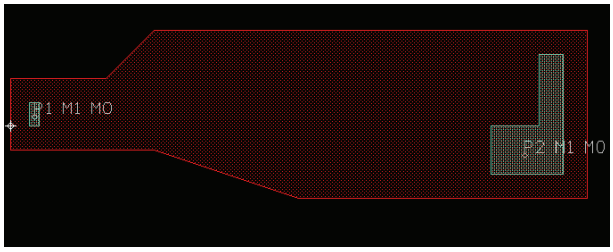
This example is available on the *InductEx* website as *ex2*.

Staying with a single inductor calculation, we can define the conductor in layer M1 as a polygon, and add ports away from the boundary of the conductor. Port 1 is defined as a path with non-zero width, and port 2 as a Manhattan-type polygon, as shown in Figure 7.3(a). The GDSII geometry file is `lman2.gds`.

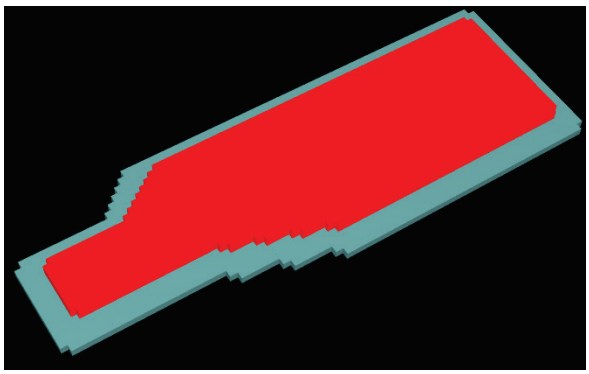
For this example, we set `GapMax = 2.0`, `GPOverhang = 2.5` and all height filaments equal to 1, in a layer technology file named `ixman2.ldf`. Since a netlist file for a single inductor already exists, it can be reused. The netlist file name differs from that of our current example, so that *InductEx* will not assign it automatically. We have to specify the netlist file with the `-n` parameter. *InductEx* is executed with:

```
inductex lman2.gds -l ixman2.ldf -i lman2.inp -fh -n lman1.cir
```

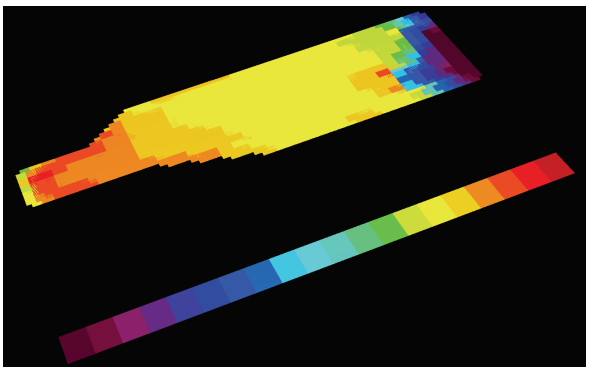
InductEx calculates L1 = 1.7447 pH. The segmented model and current density distribution, rendered in DXF Sharp Viewer from DXF output files, are shown in Figure 7.3(b) and (c).



(a)



(b)



(c)

Figure 7.3: (a) LASI screenshot of polygon inductor layout with port 1 a non-zero width path located inside the conductor and port 2 a Manhattan-type polygon, (b) three-dimensional rendering of InductEx model, and (c) calculated current density distribution in the conductor showing the effects of ports located inside the conductor boundaries. Vertical dimensions in (b) enlarged 5 times for clarity

7.5. Three-inductor multi-layer example

As a third example (available on the *InductEx* website as *ex3*), consider a three-inductor network as shown in Figure 7.4(a). We use two conductive layers above a ground plane to demonstrate multi-layer calculations. Furthermore, we use two isolation layers between the first conductive layer (M1) and the second conductive layer (M2). These mimic typical

anodization and isolation layers. The layout is shown in Figure 7.4(b), with port 2 connected to an inductor in M2. A via between M2 and M1 is created with holes defined in the negative-mask anodization and isolation layers I1A and I1B.

The circuit netlist file `lman3.cir` is shown below:

```
* spice netlist for InductEx user manual example 3
* Inductors
L1      1   2   1
L2      2   3   1
L3      2   4   1
* Ports
P1      1   0
P2      3   0
P3      4   0
.end
```

The layer definition file needs to be expanded to account for the anodization and isolation layers (I1A and I1B) and the conductive layer M2. Note that the `LastDieLayerOrder` is updated. Layers I1A and I1B have `Mask = -1` (negative mask, thus removed where structures are drawn) and `Filmtype = I`. The layer definition file `ixman3.ldf` is listed below:

```
*Layer Definition File for
InductEx manual - ex3
*
$Parameters
Units             =  1e-6
CIFUnitsPerMicron =  100
GapMax            =  2.0
AbsMin            =  0.025
GPOverhang        =  2.5
ProcessHasGroundPlane = TRUE
LastDieLayerOrder =  5
GPLayer           =  17
TermLayer         =  19
TextLayer         =  18
HFilaments        =  1
TerminalInRange   =  1.0
$End
*
* LAYERS
*
* M0
$Layer
Number      =     17
Name        =     M0
Thickness   =     0.2
Lambda      =     0.09
Order       =     0
Mask        =    -1
Filmtype    =     S
HFilaments  =     1
Colour      =     135
$End
*
* I0
$Layer
Number      =     2
Lambda      =     0.09
Order       =     2
Mask        =     1
Filmtype    =     S
HFilaments  =     1
Colour      =     10
$End
*
* I1A
$Layer
Number      =     7
Name        =     I1A
Thickness   =     0.1
Order       =     3
Mask        =     -1
Filmtype    =     I
$End
*
* I1B
$Layer
Number      =     11
Name        =     I1B
Thickness   =     0.2
Order       =     4
Mask        =    -1
Filmtype    =     I
$End
*
* M2
$Layer
Number      =     12
Name        =     M2
Thickness   =     0.35
Lambda      =     0.09
Order       =     5
Mask        =     1
```

```
Name        =     I0
Thickness   =     0.15
Order       =     1
Mask        =    -1
Filmtype    =     I
$End
*
* M1
$Layer
Number      =     5
Name        =     M1
Thickness   =     0.25
Filmtype    =     S
Colour      =     182
$End
*
* TERM
$Layer
Number      =     19
Name        =     TERM
Order       =     6
Mask        =    -4
$End
```

InductEx is executed from the command prompt with:

```
inductex lman3.gds -l ixman3.ldf -i lman3.inp -fh
```

An excerpt from the solution file is shown below:

```
Port                  Positive terminal    Negative terminal
P1                    M1,   line along y;  M0,   same as "+" terminal.
P2                    M2,   line along x;  M0,   same as "+" terminal.
P3                    M1,   line along y;  M0,   same as "+" terminal.
Minimum filaments in FastHenry = 1643

Impedance   Inductance [pH]     Resistance [Ohm]  AbsDiff   PercDiff
Name      Design    Extracted Design    Extracted (L only)  (L only)
L1        1.00000   1.65865   --        --        +0.6586   +65.87%
L2        1.00000   3.21384   --        --        +2.2138   +221.38%
L3        1.00000   1.27783   --        --        +0.2778   +27.78%
Deallocating memory.
Cycles found in 0.016 seconds.
SVD solution in 0.016 seconds.
Job finished in 2.402 seconds.
```

The *InductEx* model, with segments, is shown in Figure 7.4(c), and a solid rendering in Figure 7.4(d). Metal flow through the via is visible, as well as vertical offset at layer crossings. For this process, planarization is not modelled.

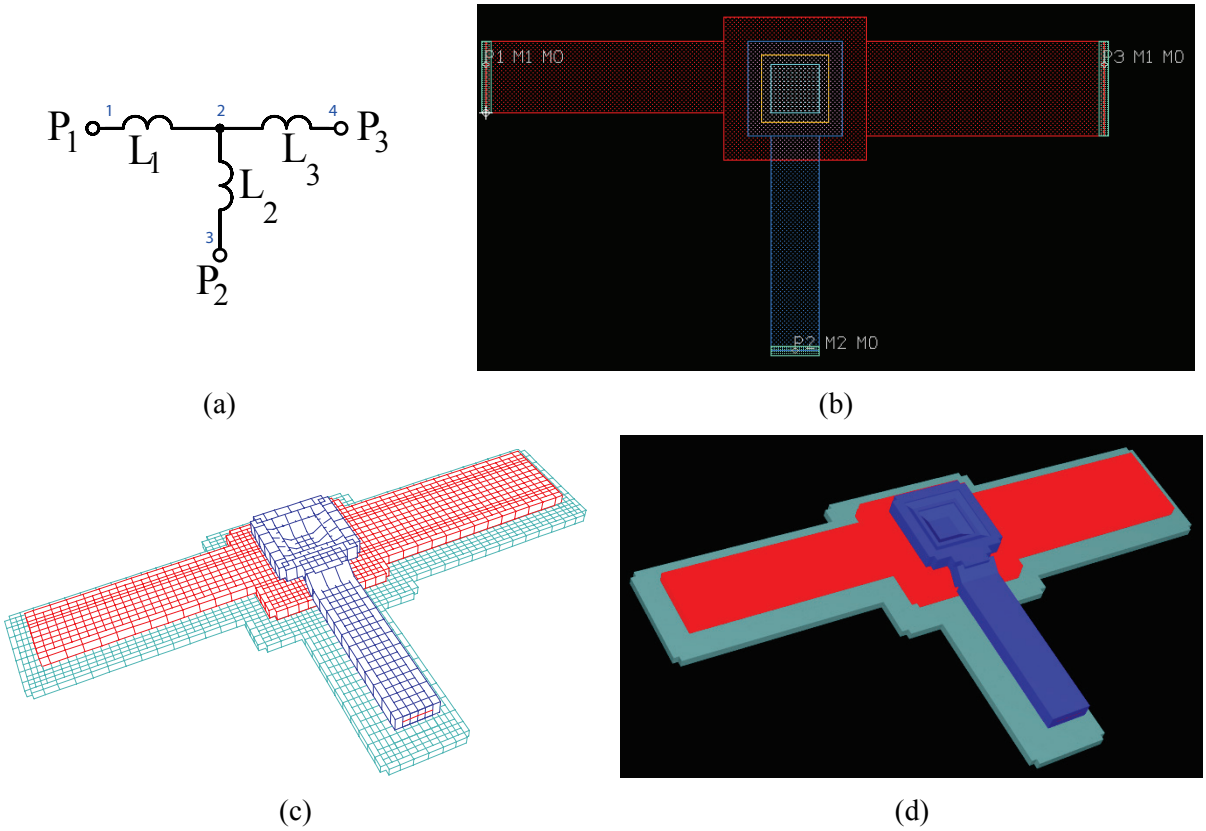


Figure 7.4: (a) Circuit netlist for three-inductor example showing node numbers, inductors and ports, (b) LASI screenshot of layout, (c) three-dimensional rendering of InductEx model showing segments, and (d) solid three-dimensional rendering of InductEx model. Vertical dimensions in (c) and (d) enlarged 5 times for clarity.

7.6. Mutual inductance example with ground plane hole

This example (available on the *InductEx* website as *ex4*) demonstrates mutual coupling and a ground plane hole. Two conductors, one in M1 and one in M2, are laid out as shown in Figure 7.5(a). A ground plane hole underneath the coupling inductors is created by drawing a box in the M0 layer. The layer definition file is the same as `ixman3.ldf` shown in Section 7.5, except that `GPOverhang = 7.5`. As can be seen in this example, the ground plane is wrapped around holes that intersect structures in any layer other that the designated ground plane. We use the larger value for ground plane overhang to prevent a too-narrow ground plane ribbon around the hole (you should experiment with this overhang parameter, and observe the increase in calculated inductance when the ribbon around the ground plane hole becomes too narrow).

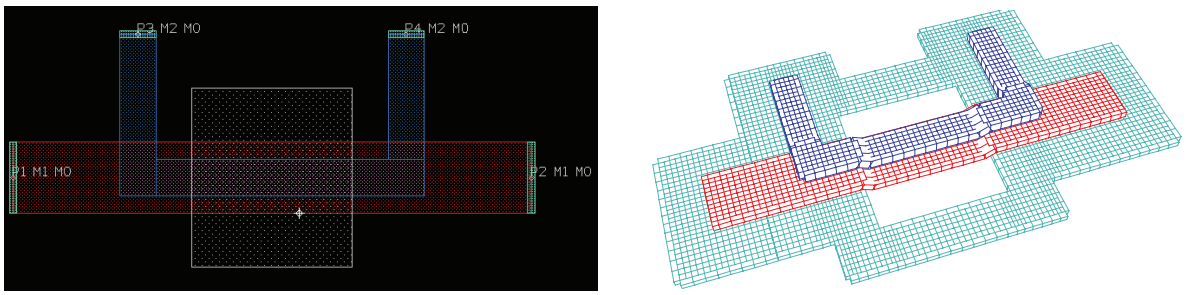
The spice netlist is shown below. Coupling is defined by the K element for compatibility with Spice. The value in the netlist is the coupling factor.

```
* spice netlist for InductEx user manual example 4
* Inductors
L1      1    2    17
L2      3    4    22
K1      L1   L2   0.75
* Ports
P1      1    0
P2      2    0
P3      3    0
P4      4    0
```

```
.end
```

An excerpt from the solution file is shown below. Note that the coupling is now listed as a mutual inductance (M1), and the design and extracted values are listed in picohenry. The design value for mutual inductance is calculated from the inductance values and coupling factor in the netlist.

```
Inductor    Design      Extracted    AbsDiff      PercDiff
L1          17.00000    16.92200     -0.07796     -0.46%
L2          22.00000    22.35300     +0.35273     +1.60%
M1          14.50400    14.63800     +0.13405     +0.92%
```



(a) (b)

Figure 7.5: (a) LASI screenshot of coupled inductors over ground plane hole and (b) three-dimensional rendering of InductEx model showing segments. Vertical dimensions enlarged 5 times for clarity.

8. Output files

In order to aid CAD integration, *InductEx* writes several output files.

8.1. sol.txt

The file sol.txt is a recreation of the screen output generated by *InductEx*, and lists progamme version number, information about the circuit netlist and layout geometry input files, the units used, the *FastHenry* version and pre-processor used, the number of loops (cycles) identified in the circuit netlist, and information on the ports.

The minimum number of filaments used by *FastHenry* is also listed (the actual number is higher if *FastHenry* subdivides narrow elements). This is a convenient debug tool: if the number exceeds roughly 100 000, demands on memory and CPU time will be severe, and the user might consider early termination and recalculation with a larger discretization size.

The extraction results are listed in pH and Ω, as shown below:

```
Impedance    Inductance [pH]      Resistance [Ohm]   AbsDiff    PercDiff
Name       Design     Extracted  Design     Extracted  (L only)   (L only)
L1         2.00000    2.09480    0.000      0.000      +0.0948    +4.74%
L2         0.60000    0.48300    0.000      2.430      -0.1170    -19.50%
L3         0.00000    6.86310    0.000      8.080      +6.8631    --%
```

```
Mutual Inductance [pH]                                  Coupling factor
Name      Design    Extracted AbsDiff   PercDiff            k
M1        +0.2000   -0.0293   -0.2293   +114.68%            -0.0141
M2        +0.2000   +0.0185   -0.1814   +90.74%             +0.0090

Ports     Design    Extracted
Pj1       250.000   250.200
```

Some observations are highlighted:

Inductor L1 is a typical superconductive microstrip line. The absolute and percentage differences between the extracted value and the design value (listed in the netlist file) are shown.

Inductor L2 includes a bias resistor, but absolute and percentage differences are only shown for the inductive component. The design value for a resistance is always zero, because the circuit netlist input file currently only supports inductances (but we reserve the position for future use).

Inductor L3 is a "don't care" inductance, specified in the circuit netlist with a zero value. In this case, no percentage difference is calculated.

Mutual inductance M1 is listed in pH. The negative extracted value indicates that the coupling is in the opposite direction. The coupling factor is calculated from the extracted mutual inductance as well as the *extracted inductance* of each of the coupled inductors linked by M1. This coupling factor should be substituted into the circuit netlist if a simulation on a layout-extracted circuit is required.

For Josephson junctions, the exact junction area (with wafer-to-mask bias accounted for) is used to calculate the critical current (the unit is left dimensionless). If specified in the circuit netlist file, the design value is also printed.

8.2. i_imag.out

If the IXI input format is used (developed for Cadence Virtuoso), *InductEx* writes the calculated port currents to the text file `i_imag.out`.

The format is:

Excited port (no.) *Measured port (no.)* *Scaled current value*

The scaled current is the **imaginary component only** of the port current at the measured port when the excited port is driven with 1 Volt. It is normalised for frequency (1 Hz) and to dimensionless PSCAN units. This allows the legacy tool *lm2sch* to calculate inductances directly.

Warning: the port numbers correspond to the order in which ports were defined in the circuit netlist, **not** to the order in which they appear in the layout input file. This could lead to ambiguity. Finally, this file should only be used if all structures in a layout are superconductive (omission of the real current components leads to errors otherwise).

8.3. extract.out

If the IXI input format is used (developed for Cadence Virtuoso), *InductEx* writes the calculated inductance values to the text file `i_imag.out`.

The format is:

Inductance name *Dimensionless extracted value*

The extracted value is in dimensionless PSCAN units, which can be converted to pH by multiplying each by 2.63. Although it only lists the inductance, these are still correct even if the extraction model contains resistive components.

9. Interpreting solutions and controlling accuracy

Numerical solutions are never as well-defined as analytical solutions. The biggest contributor to accuracy is correct modelling. If you calculate a negative inductance, reassess your model, starting with the circuit netlist. The same applies if some calculated inductance values fluctuate more than expected when small dimensional changes are made to a layout. You can improve model stability by remodelling all nodes with four or more inductive branches as more than one node interconnected by small inductances so that no node has more than three inductive branches.

The other main contributor to inaccuracies is discretization. Simply put, large segments means choppy current distribution and inaccurate solutions. Finer discretization leads to more accurate answers, but always with a hugely increased burden on computing resources.

10. Techniques for speeding up the extraction time

You can make extractions with *InductEx* run faster by applying some basic techniques to your circuit models. Firstly, any structures not forming part of the inductances you are trying to analyse, or not coupled to these, can be omitted. This cuts down on the number of segments, and cause an exponential improvement in solution time.

The quickest way to simplify a model is to remove or trim unnecessary objects. One way to do so requires you to save a duplicate of your layout, and trim this duplicate. For instance, the furthest parts of a damping resistor's geometry (connecting the resistor and the lower wiring layer) can be deleted without affecting calculation results. Bias lines can also be trimmed down to within one or two squares of the modelled inductors.

However, it is never convenient to work on a duplicate (if you for instance adjust the geometry to alter inductance, and then have to migrate the same adjustments to a master layout). *InductEx* therefore allows you to declare blanking layers, and use geometries on these layers to remove segments in the *x* or *y* directions, or to remove all segments within the blanking objects.

11. Bundled visualization tools

InductEx is distributed with visualization tools included:

Inp2dxf: Creates a 3D DXF file from the *FastHenry* input model file. This is best viewed with AutoCAD, DXF Sharp Viewer (v2) or similar programmes.

Execution: `inp2dxf inpfile.inp dxffile.dxf [-h n]`

The optional switch `-h` followed by a real number *n* sets the height scaling factor to aid visualization of vertical dimensions.

Idensity: Creates a 3D DXF file from the FastHenry input model and current density output files with 20 colour values corresponding to the size of current density, normalised so that the top of the colour scale equals the maximum current density found in the output file(s).

For *Idensity* to execute, current density output files are needed. When *InductEx* is executed with the option `-k` switch, current density output files are created. These files, one for each port in the layout, are named `j_P`*xxx*`.mat`, where *xxx* is the corresponding port name.

Execution: `idensity inpfile.inp j_Pport.mat dxffile.dxf [-h` *n*`] [-s` *n*`] [-a` *n*`]`

The optional switch `-h` followed by a real number *n* sets the height scaling factor to aid visualization of vertical dimensions.

The optional switch `-s` followed by a real number *n* sets the current density colour scale to *n* dB. The default value is 6 dB. For comparison to legacy plots generated with *Idensity* (before November 2013), use `-s 3`.

The optional switch `-a` followed by a real number *n* adjusts the maximum of the colour scale by *n* dB down from the maximum current density found in the FastHenry current density output file(s). This is useful to shift the scale down when the low current density structures (such as ground or shields plains) are of interest.

The current distribution over the entire model when the selected port is excited while all others are shorted, is calculated and rendered. The combined current distribution of multiple ports can be plotted by substituting the second parameter with a list of file names without the .mat extension, no spaces in the list, and list items separated by commas, such as:

```
idensity inpfile.inp [j_P1,j_P2,j_Pportname2] dxffile.dxf
```

12. Integration into existing CAD tools

InductEx is **not** a CAD tool – it is merely an inductance calculation utility that can be used as a loose-standing tool or as part of a larger CAD suite. Some examples of CAD suites that can use *InductEx* are given below.

12.1. Cadence Virtuoso

InductEx can be integrated into *Cadence Virtuoso* with the appropriate SKILL interface. Such an implementation is currently under development by Vasili Semenov from Stony Brook University. For this interface, Cadence outputs a .ixi file that is read by *InductEx*. This ascii text file is easy to read and debug, allows direct identification of port/terminal objects from within Cadence, and bypasses the need to convert layouts to GDSII format before extraction.

12.2. LayoutEditor

LayoutEditor, by Juspertor, has direct support for *InductEx*. To learn more, visit: http://www.layouteditor.net/wiki/InductEx.

13. Command line parameters / switches

InductEx uses command line parameters and switches to enable CAD tool integration.

The first command line parameter must be the layout input file, which can be any of the following formats (identified by the file extension): Calma GDSII, CIF or DXC.

The other parameters or switches can be organized at random, and many are optional.

`-b` – Disables SVD solution, so that the user must use `matrix_I.txt` and `matrix_v.txt` to solve inductance.

`-c n` – specifies *n* as the GMRES iteration limit for *FastHenry*. The *FastHenry* default is 200 iterations, but *InductEx* lifts this default to 400 iterations when no preconditioner is selected.

`-d [outfilename.gds]` – the -d switch instructs *InductEx* to write the geometry input to a GDS file after flattening all cell hierarchies and applying mask-wafer bias adjustments. If the optional output file name is omitted, *InductEx* will add the prefix `inductex_` to name of the geometry input file and use this as the output file name. This switch is useful to verify what *InductEx* thinks it read, and to view mask-wafer bias effects. *InductEx* also writes a file with slicing and node number information for debugging purposes.

`-i filename.inp` – specifies the name of the meshed *FastHenry* input file generated by *InductEx*. If this switch is omitted, *InductEx* will use the value assigned to the environment variable `IXINP`. Failing that, the default name `ix.inp` will be used.

`-fh` – the -fh switch enables *FastHenry* execution. If the switch is omitted, only the `.inp` mesh file is created, and the output from the last execution of *FastHenry* is used to solve the impedance network. If a change to the circuit netlist is made without any change to the layout, leaving out the –fh switch lets *InductEx* solve the network without recalculating current distribution with *FastHenry*.

`-k` – Disables file cleanup after *InductEx* execution. This leaves the current density files (j_P*xxx*.mat) for processing/visualization with *IDensity2* and any DXF viewer.

`-l filename.ldf` – specifies the layer definition file from which process parameters are read. If no filename is specified (the switch omitted), *InductEx* will use the default layer definition filename assigned to the environment variable `IXLDF`. If the environment variable is also not set, *InductEx* will attempt to read the layer definitions from the default file `ix.ldf`.

`-m` – Temporary switch to allow multicore execution of a special build of *FastHenry*. This is still experimental and not fully supported.

`-n {filename.cir | filename.js}` – specifies the name of the Spice netlist file used by *InductEx* to find the inductance network. If this switch is omitted, *InductEx* will try to read the netlist from the file with the same name as the geometry input file (in GDSII or other format). For example, if the input file is `jtl.gds`, then *InductEx* will search for `jtl.cir` and `jtl.js` (in that order) to read the netlist from.

`-o` – sets *FastHenry* parameter override. This requires the override parameter string to be defined in the environment variable `IXFHP`. *InductEx* will only pass the mesh file (filename.inp) to *FastHenry*, while the value of `IXFHP` will replace all other parameters. This is useful for playing with the accuracy and preconditioner options in *FastHenry* to optimize solution speed of a specific problem.

`-p filename.txt` – specifies the name of the port label text file used by *InductEx* to override port labels in the layout. The ASCII text file may have any extension, but `.txt` is preferable.

`-s n` – specifies *n* as the speed-up option. If the parameter is omitted, *InductEx* uses the (on average) fastest option available. In version 4.30, the default is the DIAG pre-processor in FastHenry. The options are: 0 – CUBE pre-processor, 1 – DIAG pre-processor, 2 – no pre-processor with a default limit of 400 GMRES iterations. The limit can be altered with the `-c` parameter.

`-t filename` – specifies the name of the text-based terminal definition file that declares ports/terminals in *Lmeter*-compatible format. If the filename or the switch is omitted, *InductEx* defaults to `lmeter.term` for compatibility with Cadence-based *Lmeter* scripts.

`-v` – Verbose mode on. With this switch set, *InductEx* prints more information to the screen and the output file `sol.txt`.

`-w name` – writes all native output files with `name_` as prefix (yielding `name_sol.txt`, `name_ix.cur`, `name_fastout.out` and `name_a.txt`). This allows *InductEx* to execute without overwriting previous output files, and enables multiple instances of the programme to execute in the same working directory.

`-x enginename` – specifies the name of an alternate numerical solver. Currently this should be a modified instance of *FastHenry*, such as *pFastHenry*, *FastHenry64*, etc.

`-y filename.txt` – specifies the name of an ASCII text file that contains extra *FastHenry* parameters on the first line. This allows users to override precision, maximum cores, etc.

14. Environment variables

InductEx reads default values from the following environment variables:

- `IXINP` – default name for meshed FastHenry input file
- `IXLDF` – default layer definition file
- `IXFHP` – FastHenry parameter string passed from InductEx if override switch `-o` is set.

15. Specifications

Table 3: Tested specifications of InductEx

Specification	Verified for
OS	Windows 7 32-bit & 64-bit, Windows 8.1 32-bit Mac OS 10.9.5 (Mavericks) Linux Ubuntu 14.04 LTS (Trusty Tahr)
GDSII file read-in	LASI 6 XIC 3.2.25 LayoutEditor
CIF file read-in	LASI 6

InductEx supports GDSII stream version 3 read-in, and writes out files in version 3 format.

For GDSII input files, all boundaries, boxes and paths must be of DATATYPE 0, and all text of TEXTTYPE 0, unless the `DataTypeNotZero` parameter is `TRUE`. This is implicit in LASI and XIC, but not in Cadence. Make sure to specify this if your layout tool allows multiple data- and text types.

InductEx places no limit on the number of vertices per path or polygon in GDSII or CIF input files.

No text in *InductEx* input files is case sensitive. All text is converted to lowercase before processing.

InductEx discards resistive layers automatically, even though these should be defined to allow the correct vertical offset of upper layers, and to allow operational layers to process vias that terminate on resistive layers (such as I1B for Hypres).

From version 4.25 onwards, *InductEx* allows positive or negative mask ground planes (where earlier versions only supported negative mask ground planes).

The maximum supported GDSII structure reference name length is 127 characters.

Apply text labels for port declarations in the topmost level only for GDSII files. All text in subcells is discarded. Text labels may be in any layer, and have to start with the letter "P".

Text labels for port / terminal declarations are matched to structures defined in the TERM layer. For polygon structures, text labels must be placed *within* the boundaries of the polygon. For paths (lines) or crushed boxes, the text labels must be placed on the edge of the terminal structure. If a terminal object cannot be identified, *InductEx* will process vias to try and link one to the port / terminal text layer.

Maximum text label length (for terminal declaration): 40 characters.

The layer names in a text label must correspond to the layer names defined in the technology definition (.ldf) file, and do not have to be the same as those in your layout editor or CIF file. *InductEx* uses only layer numbers to identify layers from layout files (for compatibility with GDSII).

Ports can be defined in a single text label, which assigns the same terminal block or boundary to the positive and negative terminals of the port. Ports can also defined one terminal at a time, which can then be linked to different boundaries. Ports must have one positive and either one or two negative terminals.

Port names may be up to 20 characters long, and may contain any alpha-numeric characters except SPACE, + and -. (SPACE is a separator in the label, while + and – indicate terminal polarity (which does not form part of the port name).

Since ports on different layers may overlap in the *xy* plane, port polygons are not merged during processing. Thus, if non-rectangular ports/terminals are required, these should be defined as a single polygon in layout, rather than abutting blocks.

Port structures must have all edges aligned with the *x* or *y* axes.

Paths are converted to polygons by *InductEx*. Non-Manhattan layouts are accepted, but the bend angle must not exceed 90 degrees.

Objects are processed by layer order, and layer operations (add, subtract, etc.) are therefore only effective for layers of lower order. If multiple isolation layers need to be combined into one, for instance, include the `layerADD` operation in the highest order isolation layer.

16. Exit codes

Exit codes are also printed in the exit message to the standard output. A comprehensive list, with possible solutions, is shown below.

1 : Error while trying to open GDS file. The file does not exist, or cannot be read.

2 : GDS, CIF, IXI or DXC input filename does not exist.

3 : No layer definition file (`filename.ldf`) was supplied, either as a command line parameter or an environment variable.

4 : Error while trying to open layer definition file.

5 : Error while trying to open Spice netlist file, either as ".cir" or as ".js".

6 : Error creating FastHenry .inp deck file.

7 : Could not execute FastHenry.exe.

8 : Could not execute shell application.

9 : Error while trying to open port label file.

10 : Layer definition file does not specify ground plane layer. Add the parameter `gplayer=n` to the .LDF file, where *n* is the ground plane layer's GDS number.

11 : Duplicate layer number in .LDF file. This will overwrite the first instance of the layer, and could cause errors. Look for duplication in the .LDF file by searching the layer number.

12 : Duplicate layer order in .LDF file. This will overwrite the layer order sequence, so that the process will most likely not be modelled correctly. Look for duplication in the .LDF file by searching the layer order number.

13 : No parameter block defined in .LDF file. Check if parameter block starts with `$Parameters`.

20 : Cannot resolve all inductor (branch) currents. This could mean that there are not enough ports.

21 : Singular value decomposition failed.

40 : Inductor name duplicated in circuit netlist file. Remove ambiguity by renaming one of the inductors (case – capitals or not – is ignored).

42 : Port with given name not completely defined. Verify that the corresponding port labels in the layout are correct.

43 : Number of ports in geometry input file do not match number of ports in circuit netlist.

44 : Number of ports extracted from text labels does not match number found in terminal layer plus converted vias.

45 : Cannot link positive or negative terminal of a port to a layout figure. This aborts InductEx execution.

46 : Port positive or negative terminal layer definition invalid, or does not support terminals (should be FilmType "S" or "R").

47: There are more ports in the current output file (ix.cur) than in the netlist/layout. This will cause junk solutions.

48: There are fewer ports in the current output file (ix.cur) than in the netlist/layout. This will cause junk solutions.

50 : GDSII file header corrupt or invalid. Check GDSII conversion or verify with external viewer.

51 : GDSII structure name exceeds 127 characters. Check that all cell names are shorter than 128 characters.

52 : GDSII SREF has magnification unequal to 1. Check that no cells are magnified.

53 : GDSII SREF has angle not equal to 0, 90, 180 or 270 degrees.

55 : Path read with type = 1 (thus extends beyond the endpoints of the path in semicircles). This is not supported. Check paths in all cells.

56 : Path with bend angle greater than 90 degrees read, causing ambiguity at corner.

57 : Polygon with wrong orientation causes ground plane delimit failure. Try to make give all polygons the same orientation in your layout.

58 : Duplicate terminal label names found at different coordinates in layout file. Ambiguity causes program termination.

60 : Duplicate layer name in CIF file – ambiguity causes *InductEx* to terminate. Check the top of the CIF file (especially LASI output) for text such as (LAYER 1 CIF="M0");. If another layer number has the same name, delete the incorrect definition and correct the layer table in your layout tool.

80 : Error opening FastHenry port current output file after field solving. Check your FastHenry version.

81 : Error opening Solver output file `solver_out.txt` after SVD. File might not exist – check that Solver executes correctly or is in the path.

90 : Exception – positive or negative terminal of a port has no nodes. This is an *InductEx* node-building error and should be reported if seen.

100 : Bad cycle found from netlist. This could be due to series inductors in a branch, or a branch with a port but no inductance.

110 : Could not read FastHenry version number. This might indicate an incompatible FastHenry.

111 : FastHenry version is incompatible with *InductEx*. Use the FastHenry binaries distributed with this version of *InductEx*.

17. Common mistakes new users make

InductEx, despite its small size and console-only implementation, is a complex tool. It takes some practice getting used to, and some user mistakes crop up regularly. If you are a new user, browse through the list if your extraction does not want to work.

- **Text labels not placed on ports (InductEx cannot link text to ports/terminals)**
  In all demonstrations of *InductEx*, the lower left part of a port text label is placed on or in a port object because that is where the coordinate cursor is. In LASI, the coordinates of a text label are displayed prominently, and by default is at the left bottom of the text. The position of this coordinate cursor is used by *InductEx* to link text to a port or terminal. However, some layout utilities centre text around the coordinates (mid centre), and if a user then tries to put the lower left part of text on a port object, the actual coordinates might be far to the right and top, causing a port mismatch. The first time you draw your own extraction model, make sure that you know where text is displayed in relation to the text coordinates, and place the coordinates over port objects. You may also try to set the text label properties to bottom-left.

- **Series inductors**

If your circuit netlist contains inductors in series, the individual inductor values are mathematically intractable.

18. Frequently asked questions

1. Why are some inductances negative?

Negative inductance results are incorrect. This is mostly the result of ports defined the wrong way round. If you find a negative inductance in your results, scrutinize your layout model and make sure that every port's terminals (positive first and negative second) correspond to the nodes defined in the circuit netlist. For instance, if a port is defined as

```
P1 1 0
```

in the circuit netlist, the positive terminal in the layout should **not** be on a layer connected immediately to ground. The problem is easily solved by swopping around the terminals of offending ports.

If all port polarities are correctly defined, the cause of negative inductance is often a disregarded mutual inductance. Scrutinize your layout for inductors that are collocated or that might be coupled (strongly or weakly). Add coupling factors (K elements) to the netlist file, and solve again.

2. Why do I lose connectivity through vias for SDP and ADP, and how do I solve it?

For the SDP and ADP layer definitions, there are two isolation negative mask layers that specify connection from the upper wiring layer (COU) to a lower structure, either the lower wiring layer (BAS) or the trilayer (JJ). These mask layers, BC and JC respectively, have different GDS numbers and must therefore have different orders in the layer definition file for *InductEx*. When a connection from the COU layer to a lower layer is checked, the layer with negative mask layer with the highest order will be investigated first. If an object is found, the next highest order layer will be interrogated for an overlapping object. If one is not found, *InductEx* calculates that no connection exists. Since the SDP and ADP processes assume both BC and JC to be sufficient for connection, and these are therefore not drawn as overlapping, an indication is needed for *InductEx* that connectivity is not broken if one mask layer is absent. This is easily done with the `LayerADD` parameter in the layer definition file. In the example SDP.LDF, the layer BC has a `LayerADD` parameter equal to the GDS layer number of layer JC. This lets *InductEx* add all structures from the JC layer to BC before checking connectivity and segmenting models.

19. Known bugs

- Using "ZSegsToEC = TRUE" can cause vertical connections to be missed if a via dimension is smaller than GapMax (do to conductor nodes inadvertently falling just outside the via.) If this happens, reduce GapMax to the smallest via dimension or use ZSegsToEC = FALSE.
- (RC; 2013-3) Using GDSII files of which structures are not defined according to ascending rank, with the top level structure at the end, causes InductEx to discard all but the final structure during read-in.
- (TF; 2014-10) If a layout has no drawn terminals on the TERM layer, e.g. only Josephson junctions terminals identified with text lables, InductEx aborts when the number of extracted ports is verified. To bypass this, place a "dummy" terminal block with no label anywhere in the layout if all ports are junctions or vias.

- (TF; 2014-10) Visualization: the –d option in InductEx writes the geometry read in and processed by InductEx to a GDS file. Holes in polygons are drawn as separate polygons to this GDS file, rather than subtracted from surrounding polygons. Although the geometry is correct inside InductEx, the GDS output may cause user confusion.
- (CF; 2014-10) If more than 1 height filament is used for superconducting layers above resistive layers, filaments in vias between the superconducting and resistive layers sometimes overlap in FastHenry, causing infinite mutual inductance and a subsequent calculation failure. If this happens, set "hfil = 1" for superconducting layers in your LDF file.

20. Binaries and source files distributed with InductEx

- *Idensity* – Visualization tool. Processes a *FastHenry* .inp and associated .mat current density output file to produce an AutoCAD DXF file with a 3D colour or grayscale representation of current distribution (binary only).
- *Inp2dxf* – Visualization tool. Processes a *FastHenry* .inp file to produce an AutoCAD DXF file with a 3D representation of the elements in the numerical model. Layer numbers and colours are determined from the layer definition file for ease-of-view (binary only).
- *FastHenry* – Magnetoquasistic MoM field solver. Stellenbosch University made a few alterations that allow version verification, current density output and 64-bit memory addressing. The source code is open and available on the *InductEx* website.

21. References

[1] C. J. Fourie, O. Wetzstein, T. Ortlepp and J. Kunert, "Three-dimensional multi-terminal superconductive integrated circuit inductance extraction," *Supercond. Sci. Technol.*, vol. 24, 125015, December 2011.
[2] C. J. Fourie, O. Wetzstein, J. Kunert and H.-G. Meyer, "SFQ circuits with ground plane hole-assisted inductive coupling designed with InductEx," *IEEE Trans. Appl. Supercond.*, vol. 23, no. 3, pp. 1300705, June 2013.
[3] C. J. Fourie, O. Wetzstein, J. Kunert, H. Toepfer and H.-G. Meyer, "Experimentally verified inductance extraction and parameter study for superconductive integrated circuit wires crossing ground plane holes," *Supercond. Sci. Tech.*, vol. 26, 015016, January 2013.
[4] C. J. Fourie and W. J. Perold, "Simulated inductance variations in RSFQ circuit structures," *IEEE Trans. Appl. Supercond.*, vol. 15, no. 2, pp. 300-303, June 2005.
[5] C. J. Fourie, "Full-gate verification of superconductive integrated circuit layouts with InductEx," *IEEE Trans. Appl. Supercond.*, vol. 25, no. 1, 1300209, February 2015.
[6] P. I. Bunyk and S. V. Rylov, "Automated calculation of mutual inductance matrices of multilayer superconductor integrated circuits," *Ext. Abs. ISEC*, p. 62, 1993.
[7] M. Kamon, M. J. Tsuk and J. K. White, "Fasthenry: a multipole-accelerated 3-d inductance extraction program," *IEEE Trans. Microw. Theory Tech.*, vol. 42, pp. 1750-1758, 1994.
[8] S. Nagasawa *et al.*, "New Nb multi-layer fabrication process for large-scale SFQ circuits," *Physica C*, vol. 469, pp. 1578-1584, 2009.
[9] W. H. Chang, "The inductance of a superconducting strip transmission line," *J. Appl. Phys.*, vol. 50, pp. 8129-8134, 1979.

22. Appendix A – LDF file example

```
*Layer info Hypres 4k5 A.cm-2 process
*
$Parameters
* Global parameters
Units             =  1e-6
CIFUnitsPerMicron =  100
```

```
AbsMin            =  0.001
GapMax            =  2.5
GPOverhang        =  2.5
ProcessHasGroundPlane = TRUE
LastDieLayerOrder =  10
GPLayer           =  30
BlankAllLayer     =  60
BlankXLayer       =  61
BlankYLayer       =  62
TermLayer         =  63
TextLayer         =  64
Lambda            =  0.09
Sigma             =  10
HFilaments        =  1
Colour            =  1
TerminalInRange   =  1.0
$End
*
* LAYERS
* M0
$Layer
Number     =     30
Name       =     M0
Bias       =     0.2
Thickness  =     0.1
Lambda     =     0.09
Order      =     0
Mask       =    -1
Filmtype   =     S
HFilaments =     1
Colour     =     130
$End
*
$Layer
Number     =     1
Name       =     M1
Bias       =     0
Thickness  =     0.135
Lambda     =     0.09
Order      =     2
Mask       =     1
Filmtype   =     S
HFilaments =     2
Colour     =     10
$End
*
$Layer
Number     =     6
Name       =     M2
Bias       =    -0.2
Thickness  =     0.3
Lambda     =     0.09
Order      =     7
Mask       =     1
Filmtype   =     S
HFilaments =     3
Colour     =     182
$End
*
$Layer
Number     =     10
Name       =     M3
Bias       =    -0.4
Thickness  =     0.6
Lambda     =     0.09
Order      =     9
Mask       =     1
Filmtype   =     S
HFilaments =     3
Colour     =     160
$End
*
* I0
$Layer
Number     =     31
Name       =     I0
Bias       =     0.2
Thickness  =     0.15
Order      =     1
Mask       =    -1
Filmtype   =     I
$End
*
* I1C
$Layer
Number     =     4
Name       =     I1C
Bias       =     0
Thickness  =     0.05
Order      =     3
Mask       =     0
Filmtype   =     A
$End
*
* I1B
$Layer
Number     =     3
Name       =     I1B
Bias       =     -0.1
Thickness  =     0.2
Order      =     6
Mask       =    -1
Filmtype   =     I
LayerSUB   =     9
* We subtract layer 9 (R2) from I1B to
* eradicate I1B vias to resistors that
* will short to M1 otherwise (all
* normal layers are discarded during
* segmentation, thus removing them
* from the z-directed stack for
* connectivity checking).
$End
*
* I2
$Layer
Number     =     8
Name       =     I2
Bias       =     0.2
Thickness  =     0.5
Order      =     8
Mask       =    -1
Filmtype   =     I
$End
*
* R2
$Layer
Number     =     9
Name       =     R2
Bias       =     0
Thickness  =     0.07
Sigma      =     10
Order      =     5
Mask       =     1
Filmtype   =     N
$End
*
```

```
* R3
$Layer
Number     =     11
Name       =     R3
Bias       =     0
Thickness  =     0.35
Sigma      =     10
Order      =     10
Mask       =     1
Filmtype   =     N
$End
*
* A1
$Layer
Number     =     5
Name       =     A1
Bias       =     0
Thickness  =     0.04
Order      =     4
Mask       =     0
Filmtype   =     A
$End
*
* TERM
$Layer
Number     =     63
Name       =     TERM
Bias       =     0
Thickness  =     0.1
Order      =     11
Mask       =    -4
$End
*
* OPERATORS
** Define operators here
*
$Operator
Name            =     GPM3M0
Type            =     EC
LayersRemove    =     1 6 31 4 3 8 9 5
LayersConnect   =     30 10
$End
*
$Operator
Name            =     SQJJ
Type            =     MR
LayersTransform =     3 4 5
$End
```