Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Inductance calculations

Inductance calculations for the design of superconducting qubits

By

Damian S. Steiger
ETH Zürich
dsteiger@stud.phys.ethz.ch

Supervised by

Dr. Abdufarrukh Abdumalikov
ETH Zürich
abdumalikov@phys.ethz.ch

January 7, 2013

Abstract

Quantum computers have the potential to revolutionize the field of computing because they can for example efficiently factorize an integer into its prime numbers. The experimental implementation of quantum bits is however everything else than straightforward. One possible solution might be superconducting quantum bits, like the transmon qubit, which can be controlled by a magnetic field using flux lines. In this project, I calculate the mutual inductance between flux lines and different qubits. Therefore the program InductEx is used and applied to different possible geometries of flux lines to find out which geometry has the best coupling to the closest qubit and the least effect on all other qubits on the same chip. The software InductEx is explained so that it can be used for future chip designs.

Contents

1 Introduction

1.1 General

Quantum computing is a very powerful new approach to the field of computing but also very interesting in context of fundamental physics. A quantum computer tries to exploit the physical laws of quantum mechanics in order to provide a much more powerful tool than a classical computer. Like computers, which perform mathematical operations on bits, a quantum computer uses quantum bits, also called qubits. The experimental implementation is very difficult, challenging and topic of current research.

The theoretical foundation of computing was invented by Alan Turing with his famous paper "On computable numbers, with an application to the Entscheidungsproblem" in 1936 [1]. The basic building blocks of computers are transistors on integrated circuits. In the last decades, large improvements of the transistor fabrication allowed to double the processor speed about every two years as predicted by Moore's law [2].
However, we are approaching the limit of how small a transistor can be physically built. In 2012 a single-atom transistor was realized in a lab [3]. This will be the smallest possible transistor size and therefore we will have to go new ways in future.

One possibility is to build computers based on the principle of quantum mechanics. In 1994 Peter Shor demonstrated that a quantum computer can efficiently factorize an integer into its prime numbers [4, 5], which leads to the possibility to break the commonly used RSA cryptosystem [6, p. 11]. This algorithm was one of the milestones in the quantum computation field, since it has shown the power of quantum computers and boosted therefore the research towards implementation of quantum computers. There are also other algorithms that would benefit a large improvement of speed thanks to quantum computers, e.g. Grover's algorithm for searching an unsorted database [7].

1.2 Qubits

What is so special about qubits? A classical bit is either 0 or 1 but a qubit has to be described in the mathematical framework of quantum mechanics. A qubit has two states denoted by $|0\rangle$ and $|1\rangle$, but it can be in any superposition $|\Psi\rangle$ of these states:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \quad , \quad \alpha, \beta \in \mathbb{C}\,.$$

The result of a measurement of a qubit will always be either state $|0\rangle$ or $|1\rangle$. For a qubit in a superposition state $|\Psi\rangle$, the result will be $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$. Therefore α and β must obey the following relationship:

$$|\alpha|^2 + |\beta|^2 = 1\,.$$

Note that even with this restriction, there is still an infinite number of choices for the amplitudes α and β and therefore an infinite number of superposition states for a single

qubit and not just two states 0 or 1 as for a classical bit.
A two bit system of a classical computer has 4 possible states: 00, 01, 10 and 11. A system of two quantum bits also has 4 basis states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ but as before it can be in a superposition $|\Psi\rangle$ of all these basis states:

$$|\Psi\rangle = \alpha_1 |00\rangle + \alpha_2 |01\rangle + \alpha_3 |10\rangle + \alpha_4 |11\rangle \ , \quad \alpha_i \in \mathbb{C} \, \text{and} \sum_{i=1}^{4} |\alpha_i|^2 = 1 \, .$$

Going further we see that a system of only 500 qubits has 2^{500} basis states and the superposition state of this system is therefore described by 2^{500} amplitudes α_i, which are more than the number of atoms in the universe [6]."*It is as if Nature were keeping* 2^{500} *hidden pieces of scratch paper on the side, on which it performs her calculations as the system evolves.*"[6, p. 17]

Qubits can be built by using superconducting circuits. The superconducting qubits, which are considered in this report are so called transmon qubits, they are used for example in [8]. These qubits can be controlled by a magnetic field. One would like to control each qubit separately. Hence, the inductance between the current carrying loop, called flux line, and the corresponding qubit should be large and at same time the inductance of this flux line to all other qubits should be as small as possible in order to reduce unwanted cross-coupling. The aim of this project is to find and test a program which is capable of calculating inductances between the flux lines and the different qubits to improve future chip designs.

2 Inductance

In this section, I briefly review some basics of inductance calculations. A special emphasis is given to the topic of partial inductances because it is important for the interpretation of the results. A program called InductEx [9] is used to calculate the inductances in this project, which is based on a program called FastHenry [10]. Both programs will be discussed here.

2.1 Mutual and self-inductance

An electric current I flowing in a conducting loop creates a magnetic flux Φ through the loop. The proportionality constant between the current I and the magnet flux Φ is called the self-inductance [11, p. 313]:

$$L \cdot I = \Phi. \tag{2.1}$$

For two loops the mutual inductance M_{ij} is defined by:

$$M_{ij} = \frac{\Phi_{ij}}{I_j} \, , \tag{2.2}$$

where Φ_{ij} is the magnetic flux from loop j within loop i, when there is a current I_j flowing in loop j [12, p. 216].

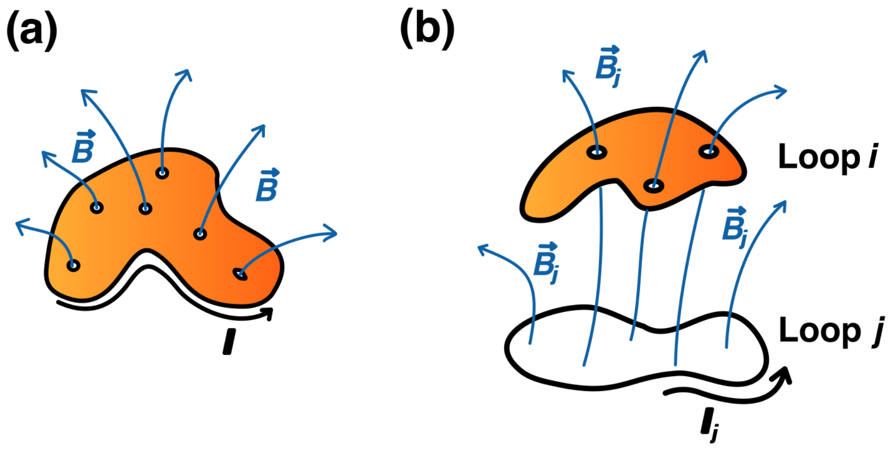


Figure 1: (a) Self-inductance: Current I produces magnetic flux through the surface of the loop, which is given by the component of the $\vec{B}$-field passing through that surface. (b) Mutual inductance: Some of the flux created by loop j penetrates loop i.

It can easily be shown that:

$$M_{ij} = M_{ji} \, ,$$

see for example [11, p. 311]. This fact is very useful for the calculation of mutual inductances between a larger and a smaller loop. The magnetic field at the center of the small loop $\vec{x}$ due to a current in the larger loop can be calculated using the Biot-Savart law:

$$d\vec{B} = \frac{\mu_0}{4\pi} \frac{\left(d\vec{l} \times \vec{x}\right)}{|\vec{x}|^3} \, ,$$

where $\vec{B}$ is the field created by the current in the larger loop and $I \cdot d\vec{l}$ is a current element of the larger loop [12, p. 175]. By taking the path integral around the larger loop, the magnetic field in the center of the small loop can be calculated and by assuming that the magnetic field inside the small loop is approximately constant, one directly gets the magnetic flux Φ by multiplying with the area of the small loop. Therefore the mutual inductance is given by equation (2.2).

2.2 Partial self-inductance and partial mutual inductance

So far just the inductances of closed loops were considered because otherwise there is no current flowing and therefore no magnetic flux. Weber said in his book [13] "*It is important to observe that inductance of a piece of wire not forming a closed loop has no meaning*". However, in 1972 Ruehli invented the notion of partial inductance [14].

A good review is given in appendix E of the book by H. Ott [15]. Ruehli defined the unique inductance for an isolated segment of a current-carrying loop. The surface area associated with a partial self-inductance is shown in Fig. 2. It is bounded by the conductor segment, two lines perpendicular to it and infinity. The magnetic flux through this surface generated by current I flowing through the conductor can be calculated. Then, equation (2.1) can be used to get the partial self-inductance.

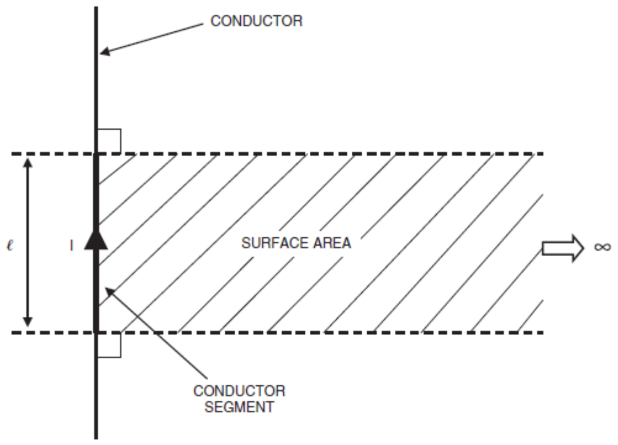


Figure 2: Surface area associated with the partial self-inductance of a current-carrying conductor segment. Copied from [15, p. 771].

The area associated with the mutual inductance of two current carrying conductor segments is shown in Fig. 3. The area is bounded "*at the ends by the conductor segments* k *and infinity, and on the sides by two straight lines which go through the points* b_k *and* c_k *and the normal to the line connecting the points* b_m *and* c_m" [14]. Then the partial mutual inductance between current carrying segments m and k is given by:

$$L_{p_{km}} = \int_a \vec{B}_{km} \cdot d\vec{a} \,, \tag{2.3}$$

where a is the shaded area of Fig. 3.

"*The net partial inductance* L_{np} *of a conductor segment is equal to the partial self-inductance of that segment, plus or minus the partial mutual inductance from all nearby current-carrying conductors.*" [15, p. 776]. It is plus if the currents in the two segments are in the same direction or minus if the currents are flowing in opposite directions or 0 if the conductors are orthogonal to each other.
The loop inductance is equal to the sum of all net partial inductances of each element. This shows that the idea of partial inductances is more fundamental than the concept of loop inductances.

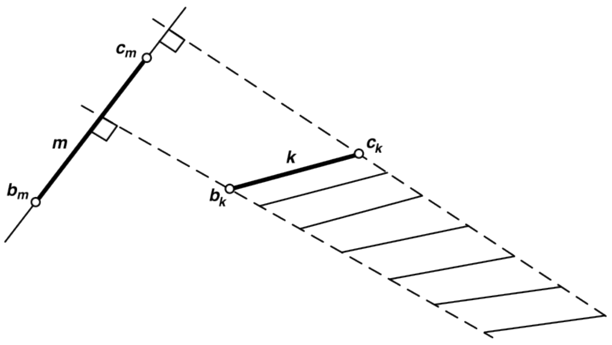


Figure 3: Surface area associated with the partial mutual inductance of two current-carrying conductor segments. Copied from [14].

2.3 FastHenry and InductEx

The main task of this project was to find a program that can directly import files with different geometries of qubits and then calculate the inductances.
In this project the program InductEx [9] is used. It is a pre-processor and post-processor for FastHenry. InductEx can import the geometry from a GDSII file and create a meshing. Then it automatically launches FastHenry with all necessary parameters and the created mesh.
FastHenry is a 3 dimensional inductance extraction program [10]. The original version doesn't support superconducting circuits, however, there exists a version from Whiteley Research Inc., which supports superconducting circuit through the inclusion of the two-fluid London equations [16]. InductEx comes with its own 64 bit version of FastHenry with support of superconducting circuits. Because it is a 64 bit version, it can allocate more RAM than the standard 32 bit version and hence it can handle larger simulations. InductEx also makes a post-processing of FastHenry to output the inductances in a table.
FastHenry was used for example to calculate the coupling of flux qubits in a recent work by Groszkowski et al. [17].

3 InductEx Manual

This section contains a detailed review of how to use InductEx. The latest version of InductEx can be downloaded from the official website [18]. I used the 64 bit windows version with release date 23. August 2012. While I was writing this report, a new version of InductEx came out the 26. Oct. 2012, which removed bugs that I had found and with new features I had suggested to add. All calculations for this project were done on a PC with 16 GB of RAM and a quad core CPU.

To calculate the inductance with InductEx the following files are needed:

- Layout file in GDSII format
- Layer definition file (.ldf)
- Port label input file (.txt)
- Circuit netlist file (.cir)

In the following subsections I will go through a complete example of a calculation. The results are shown in section 4.2.1. For more details about InductEx please see its manual [18].

3.1 Layout file

InductEx required a GDSII layout file, which defines the geometries and the ports for calculating the inductances.
All the geometries provided were only available as a DXF file. The conversion of a DXF file to a GDSII file was done by LinkCAD. The GDSII file which I consider in this section of the report is shown in Fig. 4. It has exactly 4 different layers (all using positive layer masks):

- qubits (layer "L1" , blue)
- superconducting resonator, flux and charge lines (layer "L24" , black)
- term layer with all ports (layer "TERM" , red)
- auxiliar layer to optimise the meshing of InductEx (layer "A0" , orange)

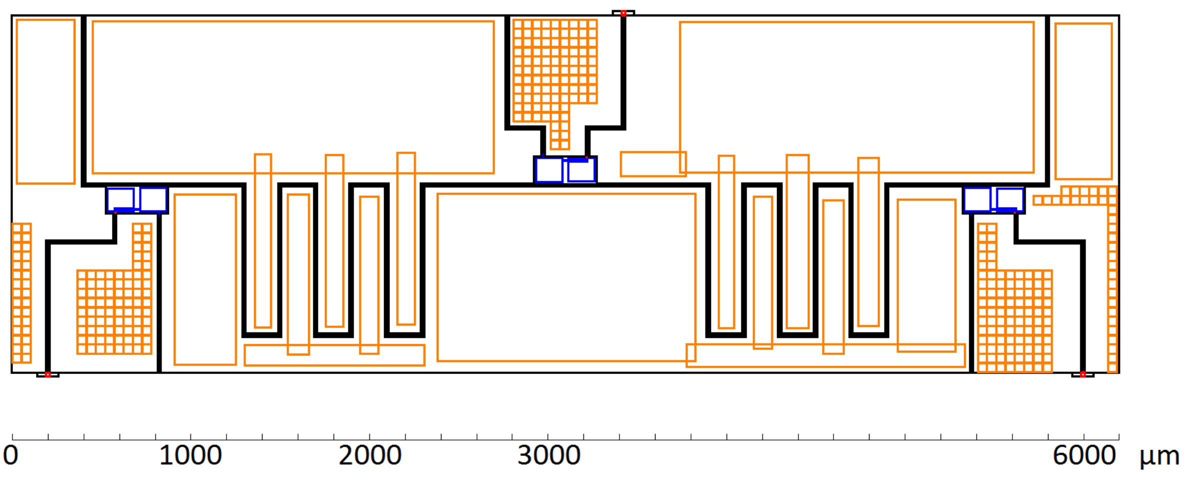


Figure 4: GDSII file of the first chip design.

Step 1

AutoCAD can be used to draw the chip layout and save it as a DXF file. The following points are important for the DXF file:

- Draw all the structures as a positive layer mask. InductEx would also support one negative layer mask and that is the ground plane. Because of a bug in InductEx, for some designs it assumes a much larger ground plane than specified in the layer definition file. This resulted in large number of unnecessary filaments and respectively long simulation time.

- If possible, the geometry should be reduced to rectangular structures. This results in less filaments and therefore will speed up calculations.

- The layout should be snapped to a certain value, e.g. 0.1 micron. Although this is not necessary, in some cases this could avoid another bug in the program. For example, if two points are too close to each other, there might be a bug activation and InductEx crashes.

- Additional layers can be defined to force InductEx to either ignore completely certain parts of the layout for the calculation or to force the meshing process to just make the mesh in either x or y direction. This will result in a smaller number of filaments and will give a good speed up. The orange layer in Fig. 4 is such an additional layer. The mesh created by InductEx is shown in Fig. 5 and indeed the orange areas have been ignored.

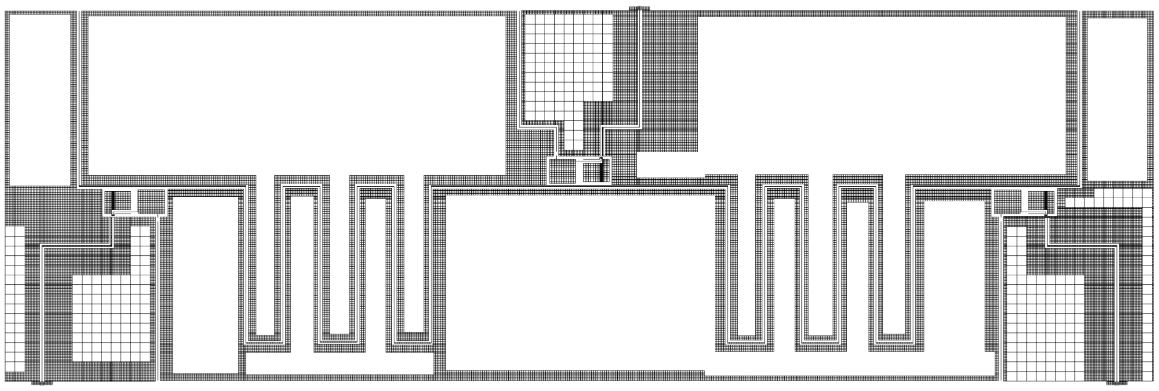


Figure 5: First chip design. There is no meshing of the inner part of the superconducting planes as specified by the orange layer in Fig. 4. But the edges of all planes still have a small meshing.

- The positive and negative port for each terminal should be drawn in one separate layer called "TERM". Small rectangles must be used for that purpose. InductEx will make all points inside these rectangles electrically equivalent. A bug in InductEx will appear if the ports are too large.

- The inductance of closed loops cannot be calculated, therefore the loops need to be opened by a very small gap. An example is shown later in Fig. 10.
- InductEx calculates the partial element equivalent circuit (PEEC) inductance, with current return path at infinity [19]. In reality this is never the case and the inductance always depends on the current return path. Therefore the negative and the positive ports of a terminal should be as close together as possible, so that they form an almost closed loop.
  For the flux line of the first chip design discussed in section 4.2.1, this is done by adding a little superconducting part to close the loop, see Fig. 10(d). For the second chip design discussed in section 4.2.2, the flux line was a closed loop, see Fig. 12(c). For comparison reasons one calculation was done with an open loop flux line, see Fig. 12(b).
- No text can be used in the DXF file. Because when converting to GDSII, LinkCAD will not use Texttype 0 and this will make InductEx crash. There is a feature in the newest version of InductEx to prevent that but it has not been tested yet.

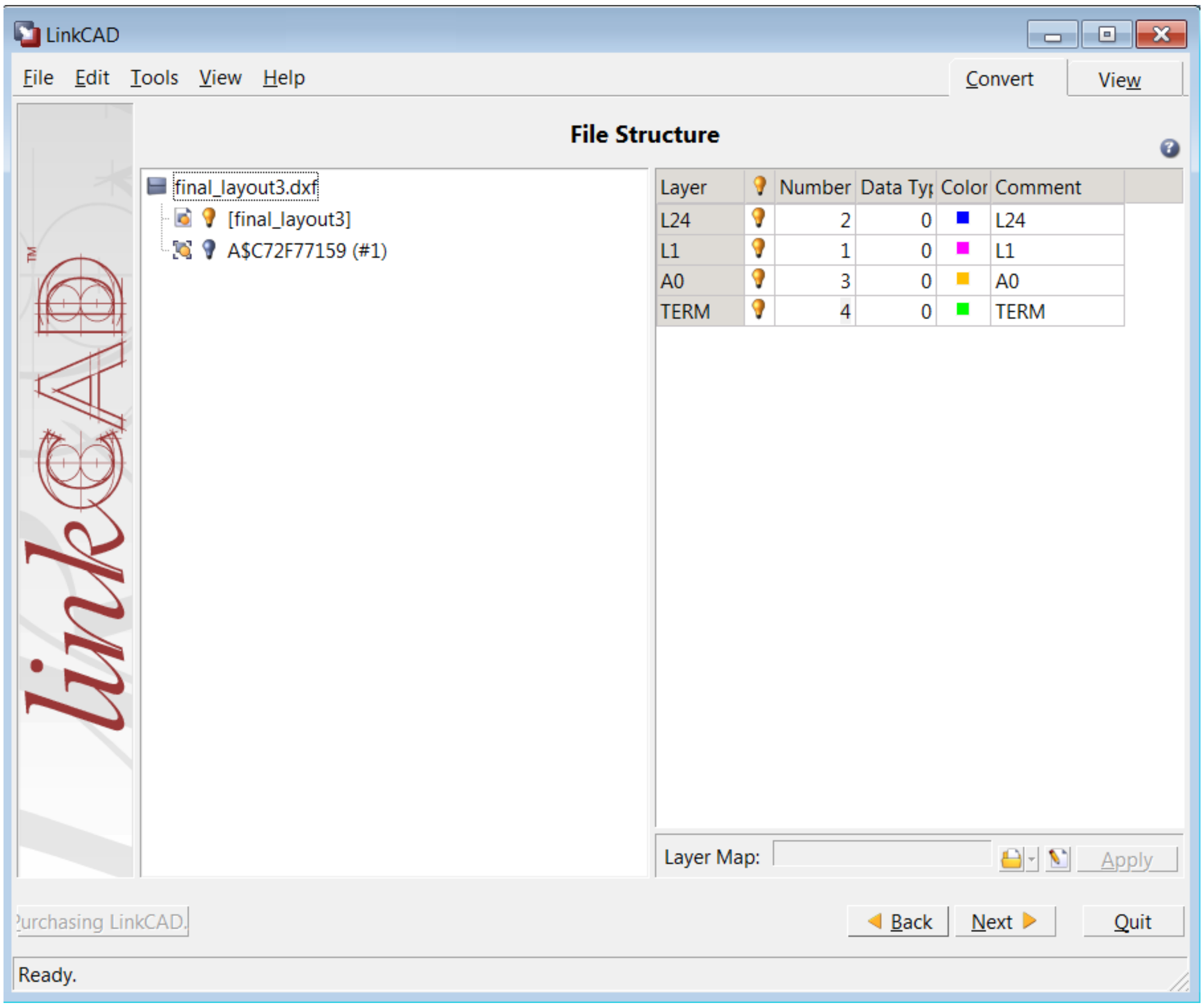
Step 2

Convert the DXF file created by AutoCAD to GDSII format using LinkCAD. The different layers need to be numbered. There are no strict rules but I suggest not to use number 0 and make the "TERM" layer the highest number, see Fig. 6.

3.2 Layer definition file

The layer definition file for this calculation is given in the appendix 9.1. The complete list of keywords can be looked up in the manual of InductEx [18]. The keywords, which are changed for different designs of this report, are discussed here:

- "GapMax" specifies the largest mesh size generated by InductEx (it is in microns in this example). It needs to be adjusted so that the total number of filaments, which InductEx produces, is not too large for a simulation (150'000 filaments take about 30 min to calculate).
- "Name" and "Number" must be identical as defined in the conversion from DXF to GDSII, see Fig. 6.
- "Order" must be sequential integers starting from 0 specifying the construction order of the chip. Physical layers come first. Last fabricated layer must be specified in "LastDieLayerOrder". The "Term" layer has always the highest order number. You must specify the "Order" of the term layer with the keyword "TermLayer".
- If an auxiliary layer is used for a more efficient meshing, e.g. layer "A0", its "Order" number has to be used with the keyword "BlankAllLayer".



Layer	💡	Number	Data Typ	Color	Comment
L24	💡	2	0	🟦	L24
L1	💡	1	0	🟪	L1
A0	💡	3	0	🟧	A0
TERM	💡	4	0	🟩	TERM

Figure 6: Numbering the layers in LinkCAD when converting from DXF to GDSII.

- If a negative layer mask should be used, then change "ProcessHasGroundPlane" to "True" and specify the "Order" number of that layer in "GPLayer" and change the "Mask" of this layer to "-1". In addition with "GPoverHang" you need to specify how far the ground plane should further extend at any edge of any object.

3.3 Port label file

A port label file is needed to assign names to the excitation ports which are in the "TERM" layer of the GDSII file. It would also be possible to specify the names of the ports directly in the GDSII file on a separated layer (which needs to be specified by "TextLayer" in the layout definition file). However, converting Text from DXF to GDSII with LinkCAD doesn't use Texttype 0 in the GDSII file and InductEx therefore doesn't accept it.

Here is the port label file used for this example:

ports.txt

```
P1+ L24; (300, 100.1)  *positive port of terminal 1 on L24 (x,y)
P1- L24; (300, 87)     *negative port of terminal 1 on L24 (x,y)
P2- L1; (680.5, 996.05)
P2+ L1; (680.5, 995.85)
P3- L24; (3525, 2099)
P3+ L24; (3525, 2112)
P4+ L1; (3319.5, 1303.45)
P4- L1; (3319.5, 1303.65)
P5+ L24; (6100, 101)
P5- L24; (6100, 87)
P6- L1; (5714.5, 996.05)
P6+ L1; (5714.5, 995.85)
```

Use numerical names for the different terminals (Pname) and specify the coordinates of the positive port and the negative port of a terminal. In addition the physical layer of the port has to be specified, i.e. L1 for ports on the qubit or L24 for ports of the flux line.

3.4 Circuit netlist file

A circuit netlist file is needed as well. It specifies the circuit through which currents flow. The inductors are specified in the following way:

```
Lname node+ node- designvalue
```

The design value is specified in pH and is not important for the calculation, any value is fine. The mutual inductances are specified in the following way:

```
Kname Lname1 Lname2 designvalue
```

In this case the design value is between 0 and 1 and can also be arbitrarily chosen.
Note that the circuit netlist file should have the same filename as the GDSII file. In that case you don't need to specify its path and name when using InductEx from the command line.

Here is the example of a netlist file for calculating the inductance of one flux line and one qubit and the mutual inductance between them:

example.cir

```
* spice netlist example
*Inductors
L1 1 2 10
L2 3 4 10
K12 L1 L2 0.5
*Ports
P1 1 2
P2 3 4
.end
```

In the appendix 9.2 there is the netlist file used for the example in this section.
A good summary of how to write the netlist file is given in the InductEx manual [18].

3.5 Running InductEx

InductEx is run by command line:

```
inductex.exe example.gds -l layout_definition_file.ldf
-i fasthenry.inp -p ports.txt -fh
```

This will run InductEx. The input file for FastHenry will be saved as "example.inp". The syntax of this file format is described in the manual of FastHenry. In the appendix 9.3 there is a short Mathematica code to display the mesh created by InductEx and saved into the .inp file. The mesh is shown in Fig. 5. For debugging purposes the following option should be included to the above ones:

```
-d exampleRead.gds
```

If you use this debugging option, then InductEx will save a file called exampleRead.gds which shows if InductEx could correctly read your GDSII file. This is very useful for checking if a negative layer is read correctly.
InductEx will output all mutual and self-inductances in pH. Note: A file with 150'000 filaments takes about 30 min to calculate. Try not to use more filaments.

3.6 Current densities

InductEx comes with a program called IDensity2. It allows the user to display the current densities at different parts of the chip. Information about how to run this program can be found in the user manual of InductEx [18]. The output will be a DXF file. An easy way to open this usually quite heavy DXF file is to use Adobe Illustrator. An example is shown in Fig. 11. Note that the color scale the program outputs is in steps of 3dB current. [19].

3.7 Debugging

In this section three possible sources of bugs are listed and also how to avoid them.

Inductance values are all 0

Check if the positive and the negative port of a terminal are connected by looking at the .inp file using the Mathematica code given at the end of this thesis. If they are not connected, then add more points there such that InductEx makes a smaller mesh on that specific locations.

GapMax value

A lot of bugs appear just for certain GapMax values. So try to change GapMax values and see if the bug disappears.

Distance between points is too close

If InductEx crashes it might be due to points which are too close to each other. Therefore always snap your layout, e.g. to 0.1 microns.

4 Results

4.1 Rings

As a first test, the inductances of superconducting rectangular rings were calculated. These calculations can give a first feeling about CPU time used for different problem sizes and how the inductance changes with smaller meshing. All rings had a hole width of $4\mu m$. One example is shown in Fig. 7. The results of calculations with different mesh sizes are shown in Fig. 8. This is a typical example which shows that with decreasing meshing size there is also a decrease in the calculated inductance. However this comes to the costs of CPU calculations time.

The results of calculations with different conductor widths is shown in Fig. 9 (blue curve). The meshing was adjusted such that all rings were meshed with the same amount of filaments. The results are compared to calculations found in [20] (red curve).

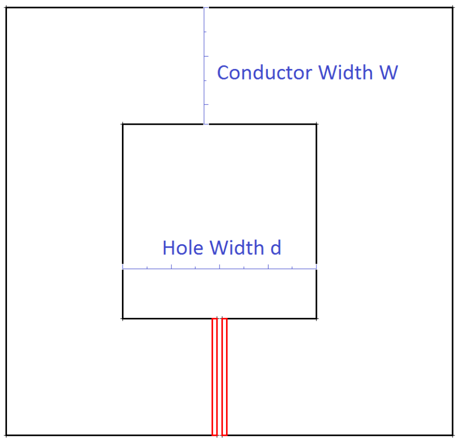


Figure 7: Superconducting rectangular ring with conductor width $2.4\mu m$ and hole width $4\mu m$. The two ports are marked in red.

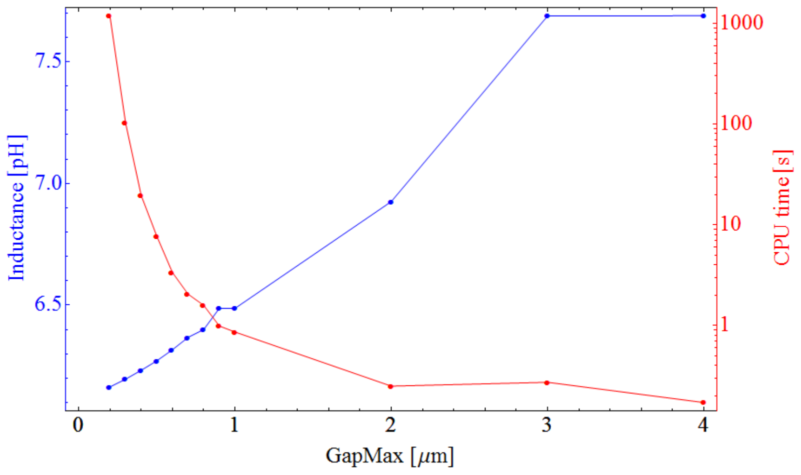


Figure 8: Blue curve: Inductance of a rectangular superconducting loop calculated by InductEx plotted as a function of the largest gap size of the meshing GapMax (blue curve). Hole width is 4 μm and conductor width is 8 μm. Red curve: Calculation time for different GapMax values.

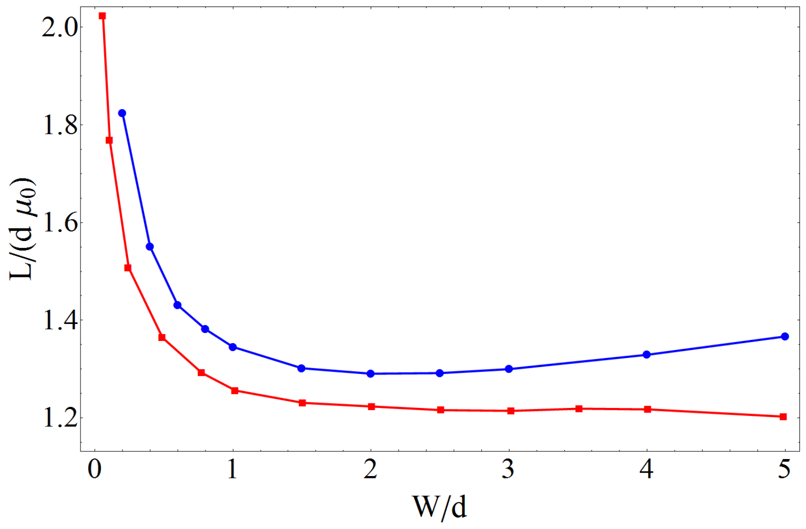


Figure 9: Inductance of a rectangular superconducting loop with hole width d and conductor width W. Blue curve is calculation with InductEx, red curve is a calculation from [20].

4.2 Qubits

The final goal of this project is to find and test a program so that the inductances for different chip designs can be calculated and improved before the chips are actually being produced. In this subsection the calculation results for 3 different qubit and flux line designs are shown.

4.2.1 First chip design

The first chip design is shown in Fig. 10. There are three qubits on the chip and for each qubit there is a corresponding flux line to control it. Anything inside the orange rectangles in Fig. 10 were neglected for the inductance calculation because they don't contribute much and so they would just make the calculation time too long.
In reality all edges of the chip are grounded and therefore the current producing the magnetic flux can flow in two different ways back to ground, see Fig. 10. InductEx cannot simulate the ground on the sides of the chip. However, one can connect the two different return paths by adding a small superconducting part close to the start of the flux line, see Fig. 10(d). The ports for the calculations are indicated in red. They are very close and so the calculated inductance is the actual loop inductance of the flux line. The results of the calculation can be found in table 1. They are calculated using a maximum mesh size of 10μm, which results in 91'346 filaments. The total calculation time was 41 min. The current densities created by flux line 1 is shown in Fig. 11. Note that the meshing corresponds to the actual meshing created by InductEx for the simulation. It automatically created a smaller meshing at some parts around the qubit. For comparison the same chip design was used but without ignoring large part of the

	Qubit 1	Qubit 2	Qubit 3	Flux Line 1	Flux Line 2	Flux Line 3
Qubit 1	10.96	0.00001	0.00001	0.08930	0.00021	0.00011
Qubit 2		10.97	0.00001	0.00017	0.09030	0.00019
Qubit 3			10.97	0.00009	0.00022	0.04731
Flux Line 1				567.53	0.00237	0.00152
Flux Line 2					442.45	0.00295
Flux Line 3						566.91

Table 1: Results of the first chip design. All values are in pH. Diagonal elements are the self-inductances, off-diagonal elements are the mutual inductances.

superconducting planes for the calculation. The results are in table 2. The meshing can be seen in Fig. 5. The maximum meshing size was also $10\mu m$ but this time it needed 212'066 filaments and 168 min CPU time.

	Qubit 1	Qubit 2	Qubit 3	Flux Line 1	Flux Line 2	Flux Line 3
Qubit 1	10.96	0.00002	0.00001	0.08872	0.00025	0.00013
Qubit 2		10.97	0.00001	0.00023	0.08336	0.00018
Qubit 3			10.97	0.00010	0.00022	0.02666
Flux Line 1				561.2	0.00304	0.00169
Flux Line 2					434.89	0.00429
Flux Line 3						560.16

Table 2: Results of the first chip design. All values are in pH. Diagonal elements are the self-inductances, off-diagonal elements are the mutual inductances. This time the superconducting planes are not removed totally for the calculation.

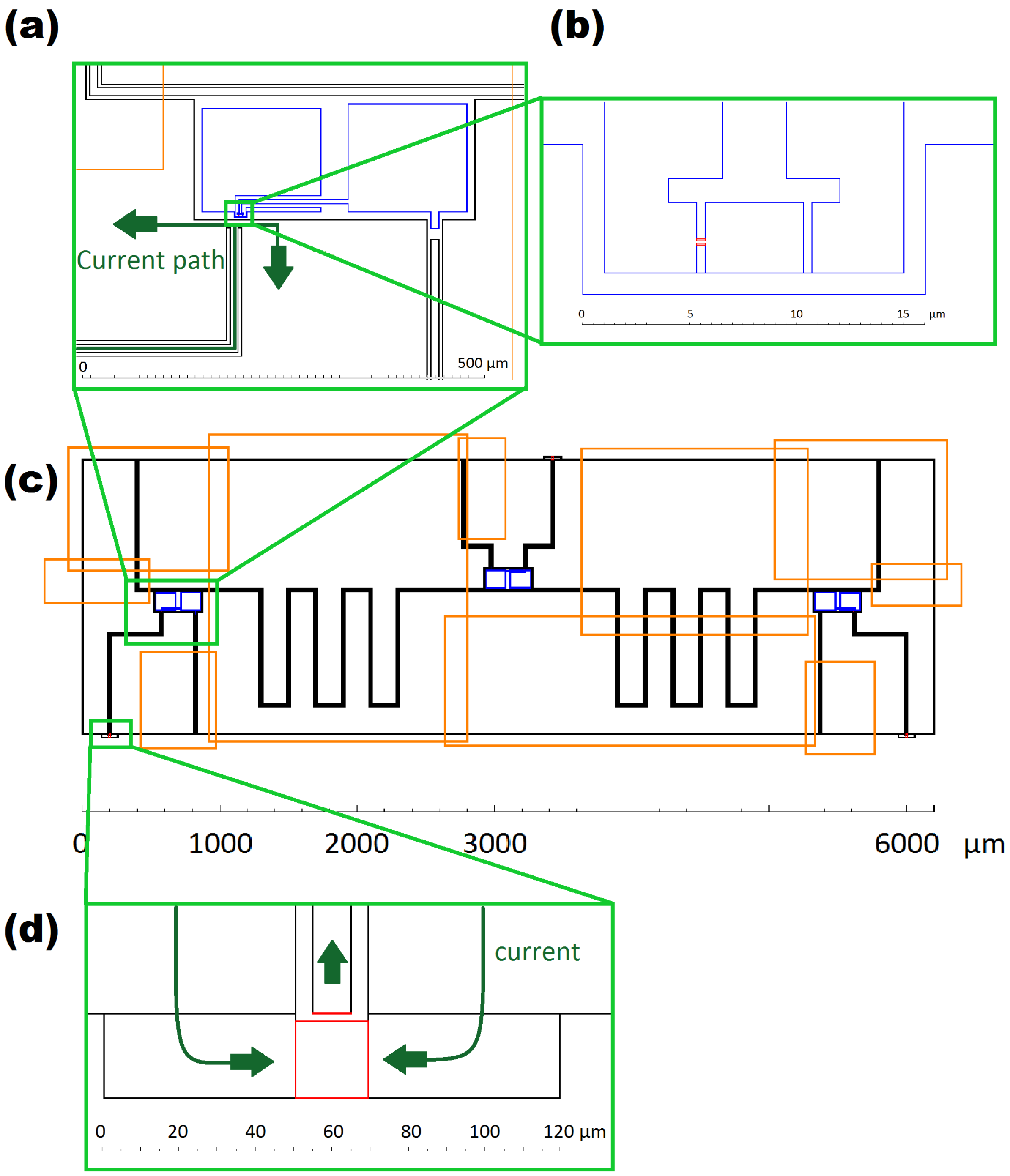


Figure 10: First chip design. The ports for the calculation are marked in red and the current paths of the flux line are indicated in dark green. The parts of the chip inside of the orange rectangles are ignored for the calculation.

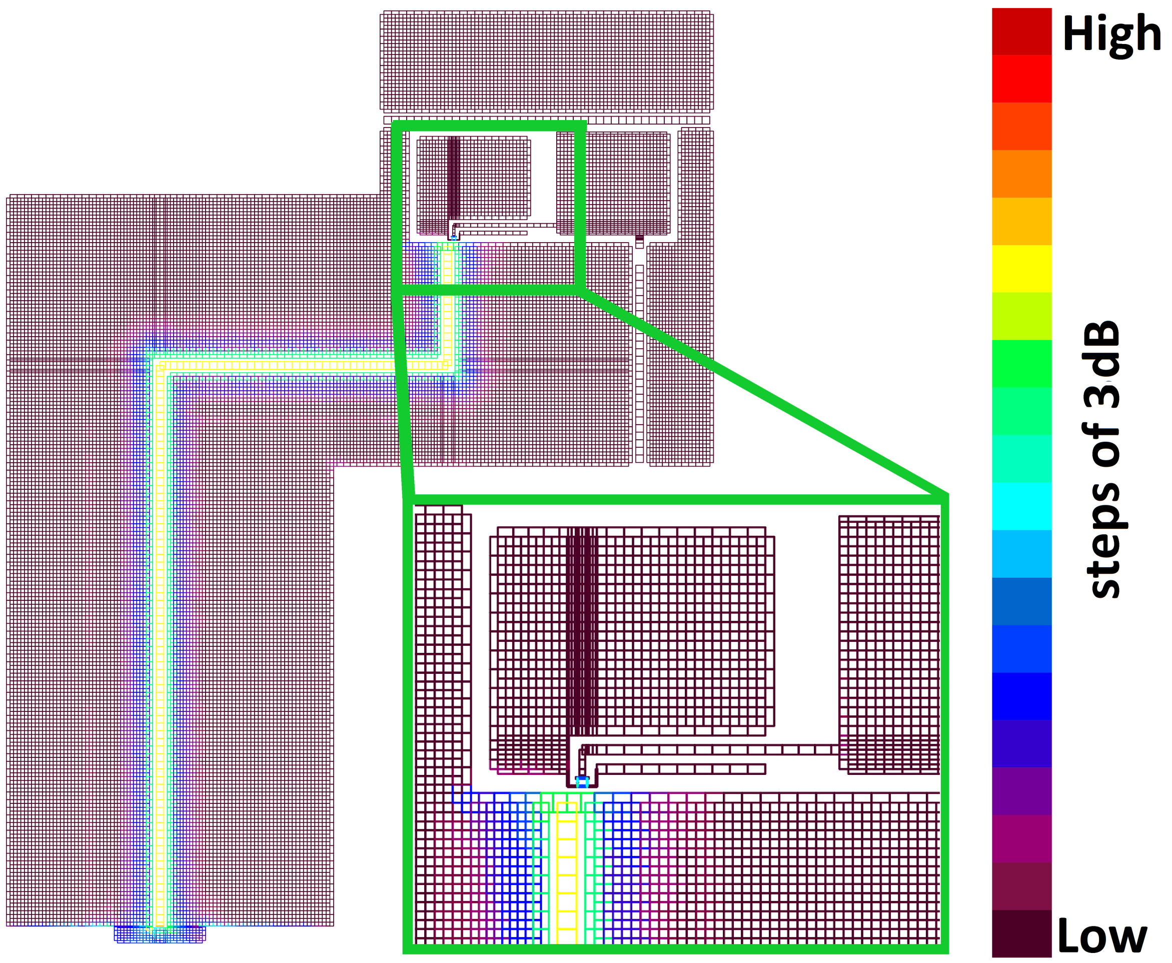


Figure 11: First chip design. Current density produced by flux line 1.

4.2.2 Second chip design

The second chip design, which I had a look at, is shown in Fig. 12 and 13. One problem was the meshing of the flux line because it doesn't have a rectangular structure. This could be solved by adding additional points and so InductEx created a smaller mesh, see Fig. 14. The flux line was almost closed for the calculation, see Fig. 12(c).

Originally the flux lines were further away from the qubit, as shown in Fig. 13(a). The results of this calculation are shown in table 3. For this calculation a maximum mesh size of 30μm was used.

	Qubit 1	Qubit 2	Qubit 3
Flux line 1	0.02223	0.00249	0.00088

Table 3: Results of the second chip design with closed flux line, see Fig. 12(c). The gap between the flux line and the qubit is large, see Fig. 13(a). All values are in pH. The self-inductance of each qubit is 10.96 pH and the self-inductance of the flux line 1 is 1'673 pH.

To have a better coupling exclusively to the close qubit, the distance between the flux line and the qubit was strongly reduced, see Fig. 13(b). The maximum meshing size was 25μm and resulted in 107'978 filaments, which took 11 min to calculate. The results are shown in table 4. Indeed the mutual inductance of the flux line 1 to qubit 1 increased by a factor of 10 compared to the previous calculation with a large gap. But more importantly the mutual inductance of the flux line 1 to the qubit 2 is now a factor of 100 less than the mutual inductance to qubit 1. In the previous calculation with a larger gap between the flux line and the qubit, the mutual inductance of flux line 1 to qubit 2 was just a factor of 10 lower than the mutual inductance to qubit 1.

	Qubit 1	Qubit 2	Qubit 3
Flux line 1	0.21302	0.00255	0.00092

Table 4: Results of second chip design with closed flux line, see Fig. 12(c). The gap between the flux line and the qubit is small, see Fig. 13(b). All values are in pH. The self-inductance of each qubit is 10.96 pH and the self-inductance of the flux line 1 is 1'764 pH.

Just as a little side mark, a calculation has been performed to show the effect of an opened flux line. The problem is that the result will then be the PEEC inductance instead of the loop inductance. The calculation was done with an open flux line, see Fig. 12(b), and a large gap between the flux line and qubit, see Fig. 13(a). The maximum mesh size of 20μm and the results are shown in table 5. The mutual inductances between the left flux line 1 and the qubit 1 is not even 50% larger than the mutual inductance between the flux line 1 and qubit 2. This is contradictory to experiments with this chip design. By comparing the results of table 5 and 3, one can see the importance of having a closed current loop to obtain meaningful results.

	Qubit 1	Qubit 2	Qubit 3
Flux line 1	0.02370	0.01677	0.00917

Table 5: Results of second chip design with an opened flux line, see Fig. 12(b). The gap between the flux line and the qubit is large, see Fig. 13(a). All values are in pH. The self-inductance of each qubit is 10.20 pH and the self-inductance of the flux line 1 is 2’133 pH.

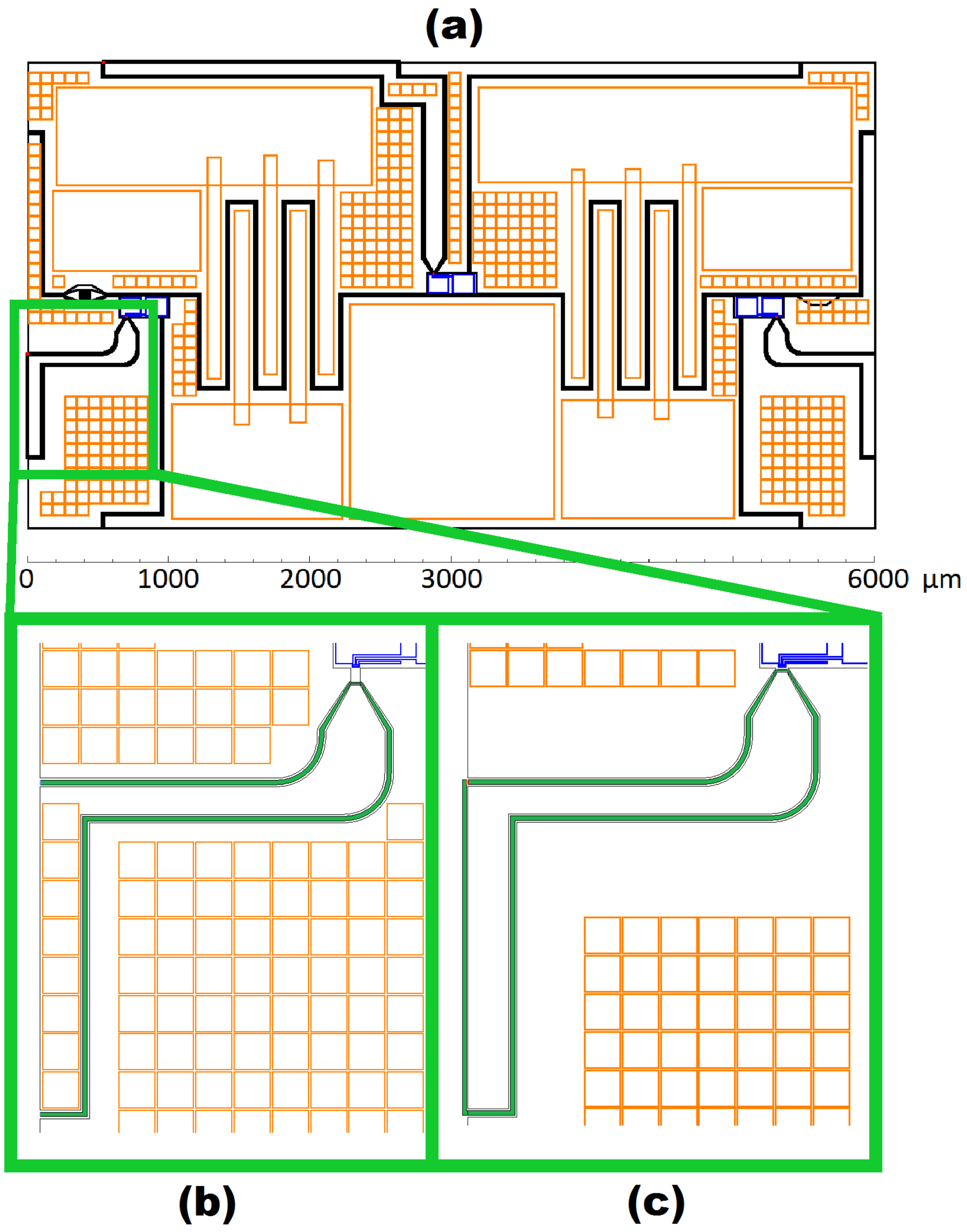


Figure 12: Second chip design. The current path of the flux line is indicated in dark green. In version (b) the flux line is opened and in version (c) the flux line forms an almost closed loop. Inside the orange rectangles InductEx will not make a meshing and so the numbers of filaments for the calculation is reduced.

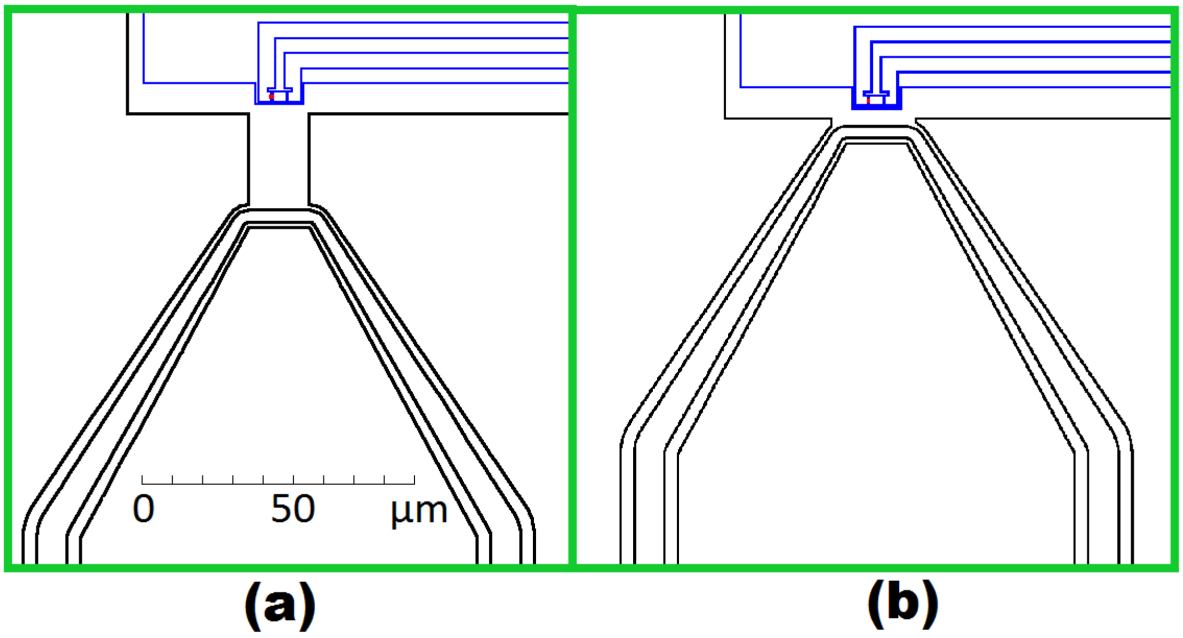


Figure 13: Zoomed view of the flux line 1 and qubit 1 of Fig. 12. In version (a) there is a large gap between the flux line and the qubit, whereas in version (b) the gap is small.

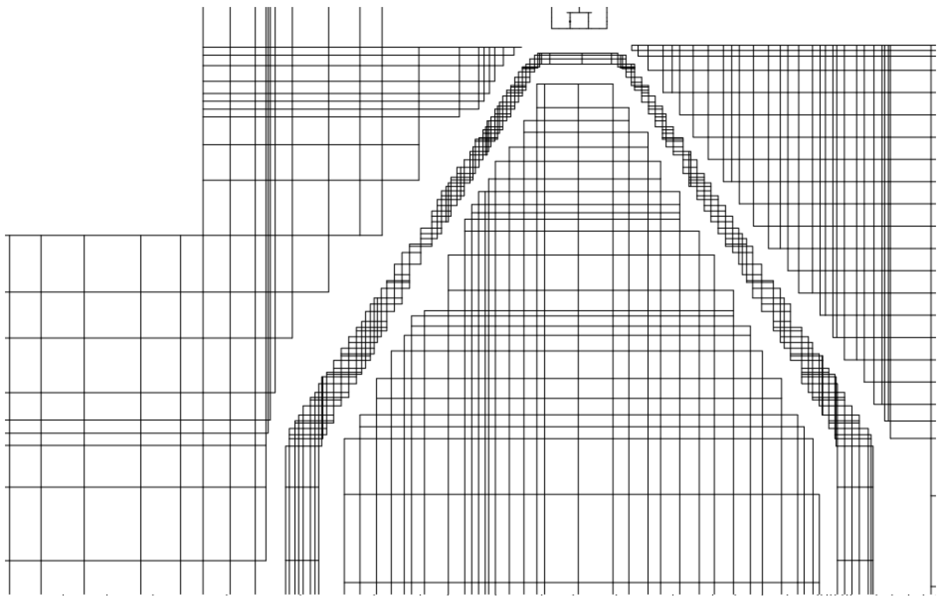


Figure 14: Second chip design. Mesh of the flux line created by InductEx.

4.2.3 Third chip design

A third chip design was tested. The design is shown in Fig. 15. A maximum mesh size of $20\mu m$ was used for the simulation. The results are shown in table 6. The mutual inductance is not high enough between the flux line and the qubits. This calculation reached the minimum precision of InductEx and therefore these results are not trustable.

	Qubit 1	Qubit 2	Qubit 3
Flux line 1	0.00006	0.00003	0.00002

Table 6: Results of the third chip design, see Fig. 15. All values are in pH. The self-inductance of each qubit is 10.96 pH and the self-inductance of the flux line 1 is 79.4 pH.

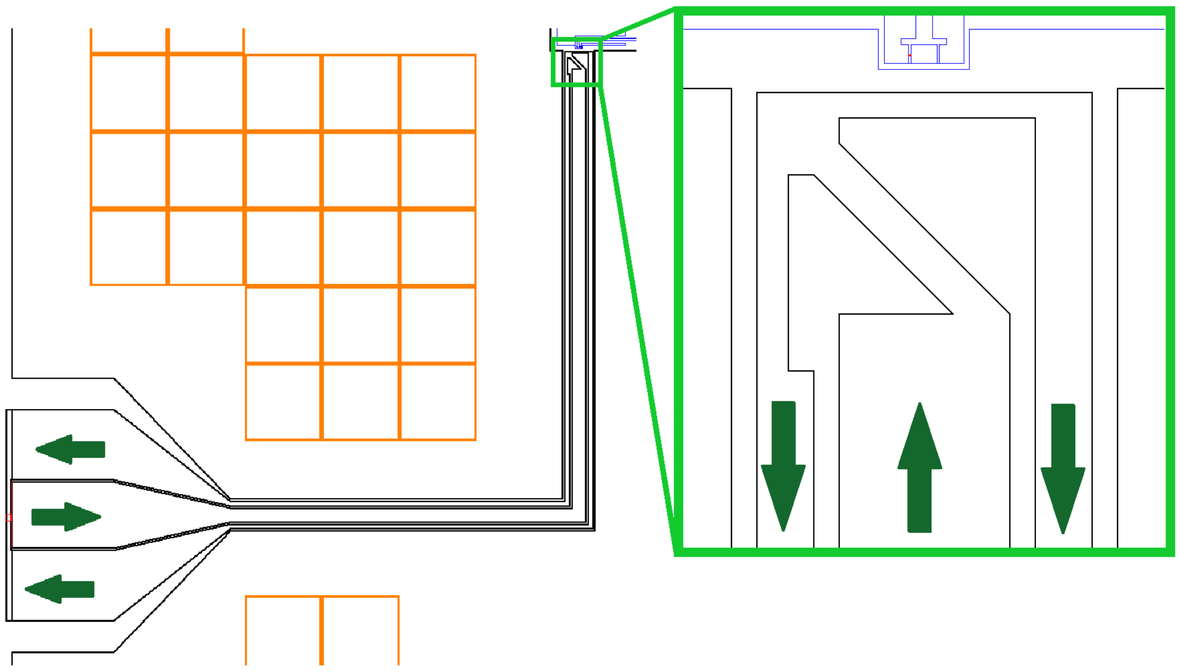


Figure 15: Third chip design. Current flow is indicated with dark green arrows.

5 Discussion

The aim of this project was to find and use a program to calculate the mutual and self-inductances of flux lines and qubits. Instead of the program FastHenry, which was initially proposed by my supervisor, InductEx was used because it includes its own 64 bit version of FastHenry with superconductor support. In addition, the pre- and post-processing engine of InductEx makes it ready to use with all kinds of chip designs.
Using InductEx had one drawback and that was the amount of bugs in the program and dealing with them resulted in a significant part of my work. Fortunately, the developer of InductEx could solve most of the bugs by now. And therefore I strongly recommend to use InductEx for future inductance calculations.
The results of the inductance calculations of simple superconducting rings show the same behaviour as stated in [20], see Fig. 9. By using a smaller meshing the calculated inductance would get a bit smaller and then fit even better.
The results for the mutual inductance of actual chip designs show how much potential improvements could be achieved by choosing good current return paths and by putting the flux lines very close to the qubits. For example the first chip design shows that the flux line has a more than 300 times smaller mutual inductance to the other qubits than to the closest qubit, which one would like to control. This is about 3 times better than the second qubit design. The current return path of the first qubit design was changed to enable its calculation. Given these results it might be useful to actually build a chip with exactly these flux line properties.

6 Outlook

Using InductEx for actual chip designs turned out to be very useful. For example it could be used to determine the optimal distance between the flux line and the qubit. In addition, it can be used to create better current return paths.
As a future improvement to this project, InductEx could be implemented into the Mathematica code used in this research group and make it therefore very efficient to perform inductance calculations. This wasn't possible so far because of the bugs, which should be fixed.

7 Acknowledgments

I would like to thank a lot my supervisor Dr. Abdufarrukh Abdumalikov for his excellent guidance, for his patience and for his very helpful discussions about upcoming problems during this project. My sincere thanks also goes to Prof. A. Wallraff for giving me the opportunity to make a semester thesis in his group. And last but not least, I would like to thank Coenrad Fourie, the developer of InductEx, for his support while using InductEx.

8 References

1. Turing, A. M. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* **s2-42,** 230–265 (1937).
2. Moore, G. Cramming More Components onto Integrated Circuits. *Electronics Magazine* **38** (1965).
3. Fuechsle, M. *et al.* A single-atom transistor. *Nature Nanotechnology* **7,** 242–246 (Feb. 2012).
4. Shor, P. W. in *Proceedings, 35th Annual Symposium on Foundations of Computer Science, Santa Fe* (IEEE Computer Society Press, 1994), 124. doi:10.1109/SFCS.1994.365700.
5. Shor, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review* **41,** 303–332 (1999).
6. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
7. Grover, L. K. in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (ACM, Philadelphia, Pennsylvania, United States, 1996), 212–219. <http://portal.acm.org/citation.cfm?id=237814.237866#>.
8. Fedorov, A., Steffen, L., Baur, M., da Silva, M. P. & Wallraff, A. Implementation of a Toffoli gate with superconducting circuits. *Nature* **481,** 170–172 (2012).
9. Fourie, C. J., O., W., Ortlepp, T. & J., K. Three-dimensional multi-terminal superconductive integrated circuit inductance extraction. *Supercond. Sci. Technol.* **24,** 125015 (2011).
10. Kamon, M, Tsuk, M. J. & White, J. K. FASTHENRY: A Multipole-Accelerated 3-D Inductance Extraction Program. *IEEE Trans. Microwave Theory Tech.* **42,** 1750 (1994).
11. Griffiths, D. J. *Introduction to electrodynamics* 3rd ed. (Prentice Hall, 1999).
12. Jackson, J. D. *Classical Electrodynamics* 3rd ed. (John Wiley & Son, Inc., 1998).
13. Weber, E. *Electromagnetic Theory* (Dover, 1965).
14. Ruehli, A. E. Inductance calculations in a complex integrated circuit environment. *IBM Journal of Research and Development* **16,** 470–481 (1972).
15. Ott, H. W. *Electromagnetic Compatibility Engineering* doi:10.1002/9780470508510 (John Wiley & Son, Inc., 2009).
16. Whiteley, S. R. *FastHenry 3.0wr* <http://www.wrcad.com/> (2001).
17. Groszkowski, P., Fowler, A. G., Motzoi, F. & Wilhelm, F. K. Tunable coupling between three qubits as a building block for a superconducting quantum computer. *Phys. Rev. B* **84,** 144516 (2011).

18. <http://www.sun.ac.za/inductex>.
19. Fourie, C. private communication.
20. Jaycox, J. M. & Ketchen, M. B. Planar coupling scheme for ultra low noise DC SQUIDs. *IEEE Trans. on Magnetics* **17,** 400 (1981).

9 Appendix

9.1 Layer definition file

layer_definition_file.ldf

```
*Layer Definitions for InductEx
*
$Parameters
Units                   =       1e-6     *set units to microns
CIFUnitsPerMicron       =       100
GapMax                  =       10       *maximum mesh size
AbsMin                  =       0.025
GPOverhang              =       5.0      *cutoff distance of GP
ProcessHasGroundPlane   =       False    *No ground plane (GP)
LastDieLayerOrder       =       1        *Last physical layer
GPLayer                 =       2        *Order of GP
TermLayer               =       4        *Order of terminal layer
TextLayer               =       18       *Order of text layer
BlankAllLayer           =       3        *Order of auxiliar layer
HFilaments              =       1        * # of height filaments
TerminalInRange         =       1.0
$End
*
* Layers
*
*
*L1 Qubits
$Layer
Number          =       1       *Corresponding to GDSII file
Name            =       L1      *Corresponding to GDSII file
Thickness       =       0.15    *Thickness of layer
Lambda          =       0.09    *London penetration depth
Order           =       1       *Construction order of wafer
                                *Must be sequential and
                                * starting from 0
Mask            =       1       *Positive mask
Filmtype        =       S       *Superconductor
HFilaments      =       2       *# of height filaments
Colour          =       135     *Color for visualization
$End
*
*L24 Superconductor
$Layer
Number          =       2
```

```
Name            =       L24
Thickness       =       0.15
Lambda          =       0.09
Order           =       0
Mask            =       1
Filmtype        =       S
HFilaments      =       2
Colour          =       135
$End
*
*Auxiliary layer
$Layer
Number          =       3
Name            =       A0
Order           =       2
Mask            =       -1        *Negative layer mask
* Term
$Layer
Number          =       4
Name            =       TERM
Order           =       3
Mask            =       -4
$End
```

9.2 Circuit netlist file

```
* spice netlist for first chip design
*Inductors
L1 1 2 10
L2 3 4 10
L3 5 6 10
L4 7 8 10
L5 9 10 10
L6 11 12 10
*Mutual inductances between all inductors
K12 L1 L2 0.5
K13 L1 L3 0.5
K14 L1 L4 0.5
K15 L1 L5 0.5
K16 L1 L6 0.5
K23 L2 L3 0.5
K24 L2 L4 0.5
K25 L2 L5 0.5
```

```
K26 L2 L6 0.5
K34 L3 L4 0.5
K35 L3 L5 0.5
K36 L3 L6 0.5
K45 L4 L5 0.5
K46 L4 L6 0.5
K56 L5 L6 0.5
*Ports
P1 1 2
P2 3 4
P3 5 6
P4 7 8
P5 9 10
P6 11 12
.end
```

9.3 Mathematica Code

```
list1 = Import[FileNameJoin[{NotebookDirectory[], example.inp}]];
length1 = Length[list1];

list2 = {};
For[i = 1, i < length1 + 1, i++, If[StringMatchQ[list1[[i]], N ~~ DigitCharacter ~~ __][[1]],
AppendTo[list2, list1[[i]]]]]
length2 = Length[list2];
list3 = {};
For[i = 1, i < length2 + 1, i++, AppendTo[list3, Flatten[StringSplit[list2[[i]]], 1]]]
list4 = {};
For[i = 1, i < length2 + 1, i++,
AppendTo[list4, {list3[[i, 1]], ToExpression[StringCases[list3[[i, 2]], NumberString][[1]]],
ToExpression[StringCases[list3[[i, 3]], NumberString][[1]]],
ToExpression[StringCases[list3[[i, 4]], NumberString][[1]]]}]]
```

```
list5 = {};
For[i = 1, i < length1 + 1, i++, If[StringMatchQ[list1[[i]], E ~~ DigitCharacter ~~ __][[1]],
AppendTo[list5, list1[[i]]]]];
length4 = Length[list5];
list6 = {};
For[i = 1, i < length4 + 1, i++, AppendTo[list6, Flatten[StringSplit[list5[[i]]], 1]]]
list7 = {};
For[i = 1, i < length4 + 1, i++, AppendTo[list7, {list6[[i, 2]], list6[[i, 3]]}]];
length3 = Length[list7]
list8 = {};
For[i = 1, i < length3 + 1, i++,
AppendTo[list8, {ToExpression[StringCases[list7[[i, 1]], NumberString][[1]]],
ToExpression[StringCases[list7[[i, 2]], NumberString][[1]]]}]]
list9 = {};
For[i = 1, i < length3 + 1, i++,
AppendTo[list9, {{list4[[list8[[i, 1]], 2]], list4[[list8[[i, 1]], 3]]},
{list4[[list8[[i, 2]], 2]], list4[[list8[[i, 2]], 3]]}}]];

ListPlot[Map[{#[[2]], #[[3]]}&, list4, {1}]]

Graphics[Line[list9]]
```