

MarginX マニュアル

Japanese ver.

2025/7/3

横浜国立大学大学院 理工学府

Graduate School of Engineering Science, Yokohama National University

松岡 昇

Sho Matsuoka

目次

第 1 章	はじめに	1
1.1	本マニュアルの説明	1
1.2	MarginX の概要	1
第 2 章	導入方法	2
2.1	ダウンロード及びコンパイル	2
2.2	初期設定	2
第 3 章	入力ファイル	4
3.1	ネットリスト	4
3.2	動作判定ファイル (judgement file)	5
3.3	ネットリスト内で使用可能な機能	6
第 4 章	実行とモード	10
4.1	実行方法	10
4.2	0. Remove intermediary files	11
4.3	1. Judge operation	11
4.4	2. Calculate margin (Accurate)	12
4.5	3. Calculate margin (binary search)	12
4.6	4. Calculate margin (binary search with synchro)	13
4.7	5. Optimization with Critical Margin Method	13
4.8	6. Optimization with Center of Gravity Method (CGM) の入出力ファイル	14
4.9	7. Sequential Optimization with CGM	14
4.10	8. JSIM modes	15
4.11	コマンドライン	15
	参考文献	17

第 1 章

はじめに

1.1 本マニュアルの説明

本マニュアルは超伝導回路のパラメータ最適化ツールである、MarginX の使い方を示すものである。本章でその概要、第 2 章で導入方法、第 3 章で入力ファイル、第 4 章で各メニューの説明を行う。

1.2 MarginX の概要

MarginX は SFQ 回路にとどまらない、Josephson 接合 (Josephson Junction: JJ) を含む超伝導回路全般に用いることができるパラメータ最適化ツールである。このツールは 2014 年に横浜国立大学の中石氏が作成したパラメータマージン抽出ツール [1] を改良して作成された。MarginX における回路シミュレーションは JoSIM [2] を用いている。MarginX は C++ で記述され、Linux 系の OS で使用することを前提として開発されているため、Windows 環境下では WSL、cygwin 等を用いないと使用できないことに注意していただきたい。MarginX に関する細かい説明は電子情報通信学会誌にて出版された論文 [3] を確認されたい。

第 2 章

導入方法

2.1 ダウンロード及びコンパイル

MarginX のソースは GitHub(<https://github.com/Yamanashi-laboratory/MarginX>) にて公開されている。以下、MarginX の導入方法について説明する。

以下のコマンドで、カレントディレクトリに MarginX をダウンロードする。

```
$ git clone https://github.com/Yamanashi-laboratory/MarginX
```

次に、ダウンロードしたフォルダに移動し、以下の手順で build フォルダを作成してそこに移動する。

```
$ cd MarginX
```

```
$ mkdir build
```

```
$ cd build
```

その後、以下の手順でコンパイルを行う。

```
$ cmake ..
```

```
$ cmake --build .
```

コンパイルが完了されると、MarginX/build 下に実行ファイル”MarginX”が生成される。生成された MarginX の実行ファイルを実行すると、図 2.1 のような出力が得られるはずである。

2.2 初期設定

初期設定は

```
$ Path_to_MarginX -s
```

で行う。実行すると図 2.2 の画面が表示されるため、ユーザーの環境で JoSIM もしくは JSIM を実行する際のコマンドを入力する。ここで入力した JoSIM や JSIM のコマンドは、"MarginX/include/function.hpp"に以下のように記述されている 7,8 行目を変更しているだけであるため、

```
#define JOSIM_COMMAND "josim"
#define JOSIM_COMMAND "jsim"
```

ここから直接 JoSIM や JSIM のコマンドを設定しても良い。

```
MM          MM          ii          'XX      XX'
MMM         MMM         XXX      XXX
MMMM        MMMM        XXX      XXX
MM MM      MM MM      .aAAAAa.  RRrr'   .gGGGGgg.  II  NN.nNNNNn.  XXxXX
MM MM      MM MM      AA      RR'   .gg      Gg  II  NN      NN      xXx
MM MM      MM MM      aAAAAAA  RR      gg      Gg  II  NN      NN      XXxXX
MM MM      MM MM      Aa      aAa  RR      gg      Gg  II  NN      NN      XXX  XXX
MM MM      MM MM      Aa      aAa  RR      gg      .Gg  II  NN      NN      XXX  XXX
MM      m      MM      aAAAAAa  RR      gGGGGggg  II  NN      NN      .xX      Xx.

          gg.  .gg
          'gGGGg'

~ Parallel Algorithm with JoSIM ~

MarginX is Margin Calculator and Circuit Optimizer from Yamanashi Lab, YNU
Please prepare some files to calculate margin or optimize your circuit
Before executing MarginX, please check settings to use "-s" command.
If you want more infomation, you can use "-h" command.
```

図 2.1 MarginX の実行結果

```
MM          MM          ii          'XX      XX'
MMM         MMM         XXX      XXX
MMMM        MMMM        XXX      XXX
MM MM      MM MM      .aAAAAa.  RRrr'   .gGGGGgg.  II  NN.nNNNNn.  XXxXX
MM MM      MM MM      AA      RR'   .gg      Gg  II  NN      NN      xXx
MM MM      MM MM      aAAAAAA  RR      gg      Gg  II  NN      NN      XXxXX
MM MM      MM MM      Aa      aAa  RR      gg      Gg  II  NN      NN      XXX  XXX
MM MM      MM MM      Aa      aAa  RR      gg      .Gg  II  NN      NN      XXX  XXX
MM      m      MM      aAAAAAa  RR      gGGGGggg  II  NN      NN      .xX      Xx.

          gg.  .gg
          'gGGGg'

~ Parallel Algorithm with JoSIM ~

Please select a setting you want to setup.

1. setting JoSIM command (default command is "josim")
2. setting JSIM command (default command is "jsim")
Selected Mode : █
```

図 2.2 セットアップモードの実行結果

第 3 章

入力ファイル

3.1 ネットリスト

マージンの測定及びパラメータ最適化を行いたい素子は以下のように素子名の頭文字を小文字にする。

```
l1    3    4    10pH
r1    3    0    10ohm
```

ネットリストを読み込むプログラムの関係上、各素子の単位は記述することを推奨する。これは電圧源など一部の素子において、正常に値を読み取ることができない場合があるからである。最適化およびマージン測定が可能な素子は、JJ、インダクタ、キャパシタ、抵抗、電流源、電圧源である。JJ のシャント抵抗がある場合は、JJ の次の行に必ず記述する。電圧源、電流源は直流のみ対応しているため、以下のような記述の仕方が必要である。

```
vb    5    0    PWL(0ps 0mV 0ps 2.5mV)
```

.tran (タイムステップ幅や実行時間の設定) の行におけるタイムステップ幅は任意であるが、JJ の位相を用いて判定している関係上、多少粗くても問題ないため、1ps を推奨する。.FILE (出力ファイルを設定) の行の内容は任意である (プログラムによって書き換えられる)。
.print(出力の設定) する行では、判定する JJ の位相を記述する。以下にジョセフソン伝送線路 (Josephson transmission line: JTL) の例を示す。

```
.model jmod jj ( rtype=1 ,vg=2.8mV,cap=0.07pF,r0=160,rN=16,icrit=0.1mA)
```

```
b01    3    7    jmod    area=2.16
RS1    3    7    5.23ohm
```

```

b02      6      8      jmod      area=2.16
RS2      3      7      5.23ohm
l01      4      3      2pH
l02      3      2      2.425pH
l03      2      6      2.425pH
l04      6      5      2.031pH
LP01     0      7      0.086pH
LP02     0      8      0.096pH
LPR01    2      1      0.278pH
LRB01    7      9      0.086pH
LRB02    8      10     0.086pH
rB01     9      3      5.23ohm
rB02    10     6      5.23ohm
rOUT     5      0      2ohm

iB01     0      1      pw1(0ps 0uA 5ps 280uA)
VIN      4      0      pw1(0ps 0uA 300ps 0ps 302.5ps 827.13uA 305ps 0uA)

.tran 1ps 1000ps 0ps
.FILE test.csv
.print phase B01
.print phase B02

.end

```

3.2 動作判定ファイル (judgement file)

動作判定ファイル (judgement file) は、対象となるネットリストの正常動作を定義する。judgement file の名前は、拡張子以外ネットリストと同様とし、拡張子は”.txt”とする。正常動作判定は JJ の位相の出力結果を用いて行わる。以下にある JJ の例を示す。

```

B1
0      100    1
100    200    3

```

はじめに判定する JJ の名前を記した後 (プログラ的には”B”が入っていればよい)、次の行に関して、左の数字 (0) は開始時間、中央の数字 (100) は終了時間、右の数字 (1) は π で規格化された位相を表している。開始時間から終了時間までのタイムウィンドウで、位相が指定した値 (上の例では π) を通過していれば、その行は正常に動作しているとみなされる。1つの素子に記述されている行全ての正常動作が確認されると、その素子は正常動作とみなされ、judgemnet file 内の全ての素子の正常動作が確認されると、そのネットリストは正常動作しているとみなされる。judgement file を記述する際、位相が大きい値から小さい値に通過していても正常動作とみなされるため、以下のような位相がマイナスになる場合でも動作判定が可能である。

```
B2
0      100   -1
100    200   -3
```

上の2つの例において、 π で規格化された位相を全て奇数にしている。これは、JJ の位相が 2π ごとに遷移することから、その中間をとって判定を容易にするためである。 π で規格化された位相の数字は奇数以外 (小数、負数、etc...) でも問題ないため、最適化する回路に応じて変更してほしい。以下に 3.1 で示した JTL の例に対応する judgement file の例を、

```
B01
300    400    1
600    700    3
```

```
B02
300    400    1
600    700    3
```

3.3 ネットリスト内で使用可能な機能

MarginX では、最適化を行う際の機能をいくつか設けている。以下で、それぞれの機能と記述の仕方を説明する。

3.3.1 シャント抵抗の計算

JJ のシャント抵抗は、JJ の値に応じて決定する。JJ の面積 (area) とシャント抵抗の積が一定になるようにシャント抵抗を調整するモードと JJ の β_c を一定にするようにシャン

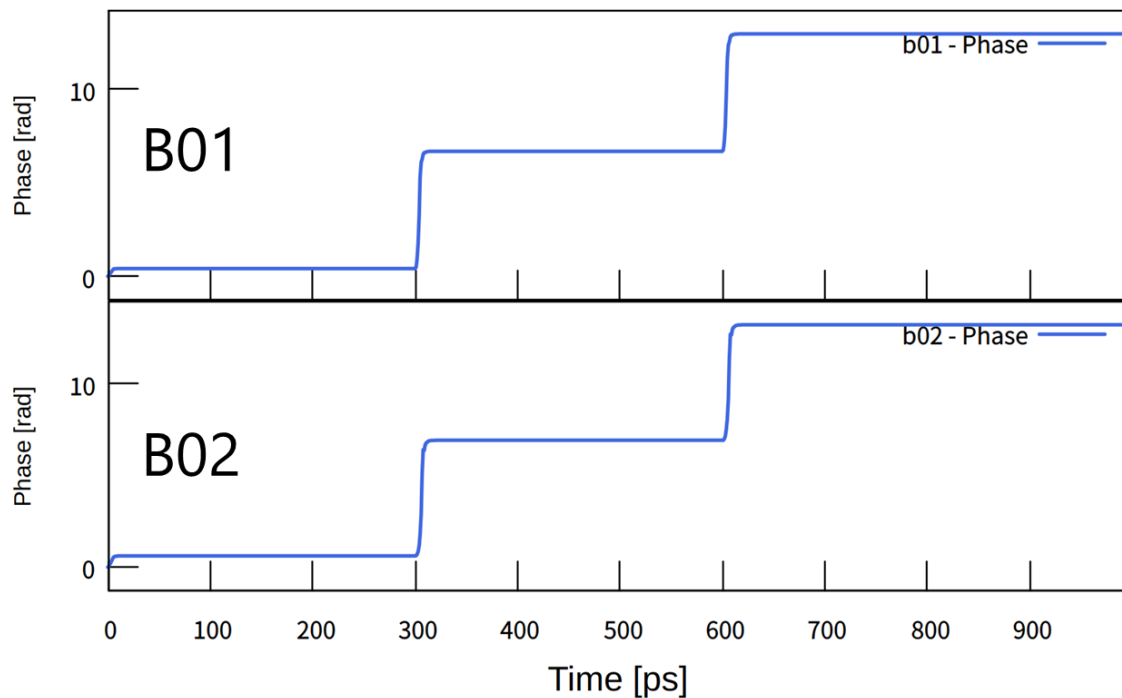


図 3.1 3.1 節で示した JTL の出力結果 (位相のみ)

ト抵抗を調整するモードが存在する。それぞれ以下のように記述する。

b01	3	7	jmod	area=2.16
RS1	3	7	5.23ohm	*SHUNT=11.30
b02	6	8	jmod	area=2.16
RS2	3	7	5.23ohm	*Bc=1

b01 の例が、JJ の面積とシャント抵抗の積が一定になるようにシャント抵抗を調整するモードであり、b02 の例が JJ の β_c を一定にするようにシャント抵抗を調整するモードである。何も記述されていない状態では、JJ の面積とシャント抵抗の積が 11.30 になるようにシャント抵抗が調整される。

3.3.2 パラメータの固定

最適化の際にパラメータを変動させたくない素子 (バイアス電圧源など) は素子の次の行に”*FIX”を付す。JJ の値を固定させたいときは、シャント抵抗のあとに記述する。以下に例を示す。

```
11  3  4  10pH
```

*FIX

vb 5 0 PWL(0ps 0mV 0ps 2.5mV)

*FIX

b01 3 7 jmod area=2.16

RS1 3 7 5.23ohm

*FIX

3.3.3 パラメータの同期

最適化の際に、特定の素子同士のパラメータを同期させることが可能である。この機能は、回路に対象な部分が存在する際に有効な手法である。同期させたい素子の次の行に”*SYN=(任意の数字)”と記述する。この際、数字が同じ素子同士のパラメータが同期する。数字が同じ素子がいくつあっても構わない。以下に例を示す。

11 3 4 10pH

*SYN=1

12 5 6 10pH

*SYN=1

b01 3 7 jmod area=2.16

RS1 3 7 5.23ohm

*SYN=2

b01 3 7 jmod area=2.16

RS1 3 7 5.23ohm

*SYN=2

この例の場合、L1、L2 と B01、B02 のパラメータはそれぞれ同期する。

3.3.4 パラメータの最大、最小値

素子のパラメータを乱数によってばらつかせる際に、その最大値と最小値をそれぞれ設定することができる。素子によってはどんどん非常に大きな (小さな) 値に変化していくことがあるので、最適化の様子をモニタリングしながら、必要に応じて使用することをおすすめする。最大、最小値を設定したい素子の次の行に*MAX=(最大値)、*MIN=(最小値)

というように記述する。どちらか片方でも、両方同時でも使用できる。以下に例を示す。

L1 3 4 10pH

*MIN=5

B01 3 7 jmod area=2.16

RS1 3 7 5.23ohm

*MAX=3

*MIN=1

この場合、L1 の最小値は 5pH に、B01 の最大値と最小値はそれぞれ 3、1 に設定される。

この機能は素子ごとに個別に設定できるだけでなく、同一の素子全体に適用できる。以下に一覧を示す。

*LMIN= インダクタンスの最小値

*LMAX= インダクタンスの最大値

*KMIN= 結合係数の最小値

*KMAX= 結合係数の最大値

*BMIN= の最小値 JJ

*BMAX= の最大値 JJ

*CMIN= キャパシタンスの最小値

*CMAX= キャパシタンスの最大値

*RMIN= 抵抗の最小値

*RMAX= 抵抗の最大値

*VMIN= 電圧源の最小値

*VMAX= 電圧源の最大値

*IMIN= 電流源の最小値

*IMAX= 電流源の最大値

第 4 章

実行とモード

4.1 実行方法

以下のように実行する。

path_to_MarginX ネットリスト名

この際、ネットリスト名は必ず judgemnet ファイル名と共通にし、拡張子は入力しない。
うまくいけば、ロゴと共に以下のように表示される。

```
Checking Judgement File ...
Number of Elements : 2
Total of Judgement : 4
Target File Name   : test_JTL.txt
-----PASS-----
```

```
Checking CircuitFile ...
Sum of Target      : 3
Target File Name   : test_JTL.cir
-----PASS-----
```

Please select an operation mode.

- 0. Remove Intermediary File
- 1. Judge operation

2. Calculate Margin (acurate)
3. Calculate Margin (binary search)
4. Calculate Margin (binary search with synchro)
5. Optimization with Critical Margin Method
6. Optimization with Center of Gravity Method (CGM)
7. Sequential Optimization with CGM
8. JSIM modes

それぞれのモードについては、以降の節で説明する。

4.2 0. Remove intermediary files

MarginX 実行時に生成される中間ファイルを全て削除する。MarginX では JoSIM の実行とその出力ファイルを用いるために中間ファイルを生成する。通常は実行完了時に削除されるが、強制終了した際はそれらが残ってしまうため、その際にこのモードを使用するとよい。

4.3 1. Judge operation

ネットリスト内の回路の出力が judgement file の内容と合っているかを検証する。入力ファイルが準備できたら、マージン測定やパラメータ最適化を行う前にこのモードで正常に動作するかを確かめることを推奨する。JJ にシャント抵抗が接続されている場合、3.3.1 節で述べた通り、シャント抵抗の値が自動的に調整されるため、もとのネットリストでは正常に動作していたが、この調整によって動いていないという判定になるため、注意されたい。

表 4.1 judge operation の入力と出力

入力ファイル	ネットリスト (.cir or .inp)、judgement file
出力ファイル	シャント調整後のネットリスト (ネットリスト名_out.cir)

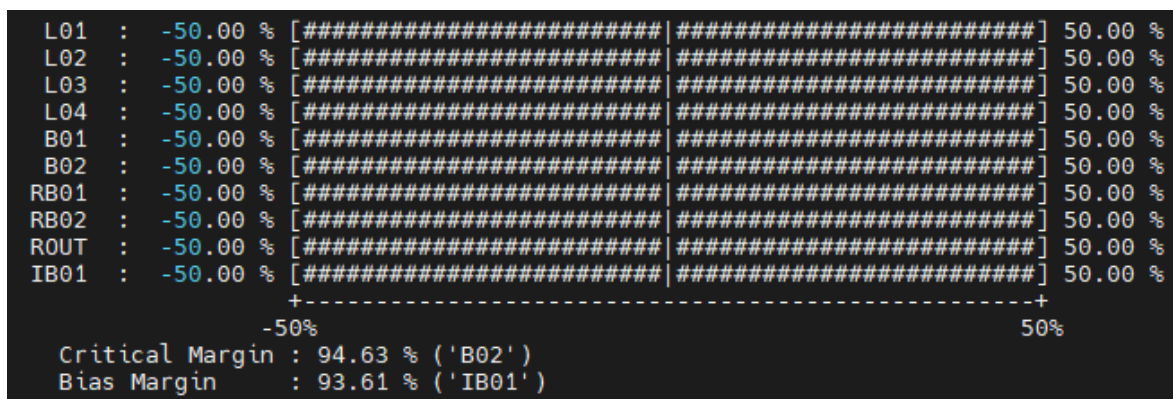


図 4.1 2. Calculate margin (Accurate) の出力結果

4.4 2. Calculate margin (Accurate)

3.1 節によって指定された素子全てのマージンを測定する。計算方法は、対称の素子のオーダーの数字を 1 ずつ上げて動作判定という操作を繰り返し、正常動作しなくなったら、1 つオーダーを下げて同様の作業を繰り返すというものである。これを 4 回行うため、4 桁までのマージンの上限、下限が精度高く計算される。計算が完了すると図 4.1 のような結果が表示されるとともに、表 4.2 に示すようにテキストファイルと csv ファイルが出力される。テキストファイルには詳細な結果が、csv ファイルにはマージン図作成用の表が格納されている。

4.5 3. Calculate margin (binary search)

3.1 節によって指定された素子全てのマージンを測定する。計算方法はいわゆる二分探索で、1 つの素子あたり、上限と下限を求める場合においてそれぞれ 10 回の探索を行う。Calculate margin (Accurate) よりも多少精度は落ちるが、実行時間は短い。出力されるファイルは同様である。

表 4.2 2. Calculate margin (Accurate) の入出力ファイル

入力ファイル	ネットリスト (.cir or .inp)、judgement file
出力ファイル	result_ネットリスト名.csv、result_ネットリスト名.txt

表 4.3 3. Calculate margin (binary search) の入出力ファイル

入力ファイル	ネットリスト (.cir or .inp)、judgement file
出力ファイル	result_ネットリスト名.csv、result_ネットリスト名.txt

表 4.4 4. Calculate margin (binary search with synchro) の入出力ファイル

入力ファイル	ネットリスト (.cir or .inp)、judgement file
出力ファイル	result_ネットリスト名.csv、result_ネットリスト名.txt

表 4.5 5. Optimization with Critical Margin Method の入出力ファイル

入力ファイル	ネットリスト (.cir or .inp)、judgement file
出力ファイル	result_ネットリスト名.csv、result_ネットリスト名.txt、ネットリスト名_out.cir

4.6 4. Calculate margin (binary search with synchro)

3.1 節によって指定された素子全てのマージンを測定する。計算方法と出力されるファイルは Calculate margin (binary search) と同様である。このモードでのみ、3.3.3 節で指定したパラメータの同期が適用される。

4.7 5. Optimization with Critical Margin Method

3.1 節によって指定された全ての素子に対してクリティカルマージン法 [4] を用いたパラメータ最適化を行う。同一の素子が連続してクリティカルマージンを持つ素子になった場合に加え、3.3.2 節で説明した*FIX によって固定されている素子がクリティカルマージンを持つ素子になった場合においても最適化が終了する。このモード中、3.3.3 節で指定したパラメータの同期は有効となる。クリティカルマージン法実行中のマージン測定は二分探索によって行われる。最適化が完了すると、表 4.5 に示すように、テキストファイルと csv ファイルとネットリストが出力される。ネットリストには最適化後のパラメータが反映されている。

表 4.6 6. Optimization with Center of Gravity Method (CGM) の入出力ファイル

入力ファイル	ネットリスト (.cir or .inp)、judgement file
出力ファイル	result_ネットリスト名.csv、result_ネットリスト名.txt、ネットリスト名_out.cir

4.8 6. Optimization with Center of Gravity Method (CGM) の入出力ファイル

3.1 節によって指定された全ての素子に対して、重心法を用いたパラメータ最適化を行う。このモードは並列アルゴリズムを用いて実行される。実行すると、以下のように表示される。

Select the Kind of Score

- 1: Only Critical Margin
- 2: Only Bias Margin
- 3: Only upper critical Margin
- 4: Only lower critical Margin
- 5: Sum of Critical Margin and Bias Margin
- 6: Sum of Critical Margin and Bias Margin * 2
- 7: Others (input yourself)

Selected Score :

これはパラメータセットに対するスコアのつけ方の一覧であり、最終的にスコアが最も高いパラメータセットが出力される。入出力ファイルは 4.6 に示す通り、5. Optimization with Critical Margin Method と同様である。

4.9 7. Sequential Optimization with CGM

3.1 節によって指定された全ての素子に対して、重心法を用いたパラメータ最適化を行う。このモードは 5. Optimization with Critical Margin Method を逐次的に処理するモードであり、実行するマシンの処理能力が強くない場合、こちらのモードを使用することを推奨する。表 4.7 に示す通り、5. Optimization with Critical Margin Method と同様である。

表 4.7 7. Sequential Optimization with CGM の入出力ファイル

入力ファイル	ネットリスト (.cir or .inp)、judgement file
出力ファイル	result_ネットリスト名.csv、result_ネットリスト名.txt、ネットリスト名_out.cir

4.10 8. JSIM modes

MarginX ではデフォルトで JoSIM が用いられるが、これを JSIM に変更することができる。8. JSIM modes を選択後、同様にモードを選択することで、JSIM を用いて MarginX を実行することが可能である。

4.11 コマンドライン

MarginX にはコマンドライン引数によるオプションが用意されている。以降でこれについて説明する。

4.11.1 詳細表示 "-d"

デフォルトでマージンの図のみがテキストで表示されるが、実行時のコマンドに"-d"を追加で記述することで図 4.2 に示すようなパラメータなどの詳細を表示させることができる。最適化進行時にパラメータの変動を確認する際に有効である。

4.11.2 マージン図表示 "-f"

実行時のコマンドに"-f"を追加で記述することで、python の matplotlib を用いて作成された図 4.3 に示すようなマージン図を作成することができる。

[#####] 100 %									
	-50%	+-----+-----+ 50%							
L01 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
L02 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
L03 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
L04 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
B01 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
B02 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
RB01 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
RB02 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
ROUT :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
IB01 :	-50.00 %	[#####]	#####	#####	#####	#####	#####	50.00 %	
	-50%	+-----+-----+ 50%							
Critical Margin : 94.63 % ('B02')									
Bias Margin : 93.61 % ('IB01')									
L01 :	2.000 pH	Lower: 0.001	Upper: 4.000	Median: 2.001	-99.95 %	~	100.00 %		
L02 :	2.425 pH	Lower: 0.002	Upper: 4.850	Median: 2.426	-99.92 %	~	100.00 %		
L03 :	2.425 pH	Lower: 0.002	Upper: 4.850	Median: 2.426	-99.92 %	~	100.00 %		
L04 :	2.031 pH	Lower: 0.001	Upper: 4.062	Median: 2.032	-99.95 %	~	100.00 %		
J01 :	2.160	Lower: 0.001	Upper: 4.320	Median: 2.160	-99.95 %	~	100.00 %		
J02 :	2.160	Lower: 0.001	Upper: 4.204	Median: 2.103	-99.95 %	~	94.63 %		
RB01 :	5.230 ohm	Lower: 0.015	Upper: 10.460	Median: 5.238	-99.71 %	~	100.00 %		
RB02 :	5.230 ohm	Lower: 0.036	Upper: 10.460	Median: 5.248	-99.31 %	~	100.00 %		
ROUT :	2.000 ohm	Lower: 0.024	Upper: 4.000	Median: 2.012	-98.80 %	~	100.00 %		
IB01 :	280.000 uA	Lower: 0.100	Upper: 542.100	Median: 271.100	-99.96 %	~	93.61 %		

図 4.2 ”-d”を使用した際のパラメータの詳細例

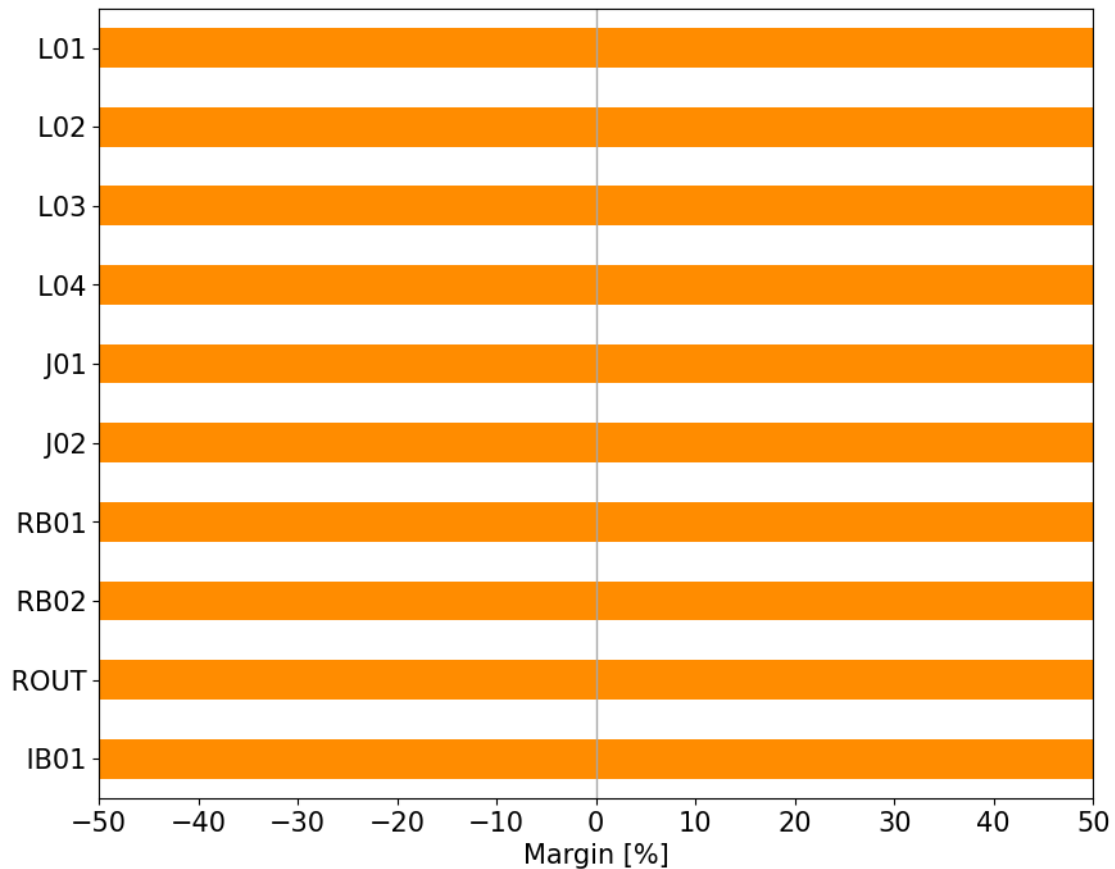


図 4.3 ”-f”を使用した際のマージン図例

参考文献

- [1] Y. Yamanashi, S. Nakaishi, A. Sugiyama, N. Takeuchi, and N. Yoshikawa, "Design methodology of single-flux-quantum flip-flops composed of both 0- and π -shifted Josephson junctions," *Supercond. Sci. Technol.*, vol. 31, no. 10, p. 105003, Oct. 2018.
- [2] J.A. Delport, K. Jackman, P.L. Roux, and C.J. Fourie, "JoSIM - Superconductor SPICE Simulator," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, p. 1300905, Aug. 2019.
- [3] S. Matsuoka, N. Yoshikawa, and Y. Yamanashi, "MarginX: Simple and Fast Circuit Parameter Optimization Tool for Superconductor Circuits," *IEICE Trans. Electron.*, vol. E108-C, no. 11, to be published. doi: 10.1587/10.1587/transele.2024ECP5100.
- [4] C. Hamilton and K. Gilbert, "Margins and yield in single fluxquantum logic," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 4, pp. 157-163, Dec. 1991.