

超伝導単一磁束量子 (SFQ) 回路設計のための タイミング制約を考慮した自動配置アルゴリズム

出島 貴史¹ 高木 一義¹ 高木 直史¹

概要: 本稿では, 超伝導単一磁束量子 (SFQ) 回路のレイアウト設計におけるセル配置アルゴリズムを提案する. 提案アルゴリズムでは近距離配線はジョセフソン伝送線路 (JTL), 長距離配線に受動線路 (PTL) を用いる混合配線を行い, 両者を適切に使い分けることで配線遅延を最小限に抑える. まず, 直接接続されたゲートの集合を 1 つのブロックとみなし, ブロック内部のレイアウトを決定する. その後, 全てのブロックを格子状に配置しブロック間を PTL で配線する. 提案手法を用いた場合と, PTL のみで配線した回路で回路性能の比較を行ったところ, 回路面積が 10% 以上削減でき, レイテンシが小さくなることを確認した.

A Placement Algorithm for Single-Flux-Quantum Circuits Considering Timing Constraints

Abstract: In this paper, we propose a cell placement algorithm in layout design of superconducting single flux quantum (SFQ) circuit. In the proposed algorithm, mixed wiring using Josephson transmission lines (JTLs) for short distance and passive transmission lines (PTLs) for long distance is performed. Wiring delay is minimized by using the both appropriately. First, a set of directly connected gates is regarded as one block, and the layout inside the block is determined. After that, all the blocks are arranged in a lattice, and the blocks are wired with PTLs. Comparing performance of a circuit with the proposed method and a circuit wired only with PTLs, we confirmed that the circuit area is reduced by more than 10% and the latency is reduced.

Keywords: SFQ 回路, レイアウト設計, 配置・配線, CAD

1. はじめに

従来の半導体回路より高速かつ低消費電力で動作する単一磁束量子 (SFQ; Single Flux Quantum) 回路は超伝導リング中の磁束量子をジョセフソン接合によって制御し, 磁束量子の有無によって論理値 1 及び 0 を表現する [1][2]. SFQ 回路では, 論理値の 0 とパルスの未到着を区別するために全てのゲートにクロックを供給する必要があり, CMOS 回路と比べ, タイミングの制約が厳しい. そのため, 大規模な SFQ 回路の正確かつ迅速な設計には計算機による設計支援 (CAD) が必要不可欠である. 一方で, SFQ 回路は, 動作や設計手法が CMOS 回路とは本質的に異なるため, CMOS 回路の CAD ツールを SFQ 回路にそのまま適用す

ることができず, レイアウト設計は主に人手で行われているというのが現状である.

これまでに SFQ 回路の設計手法として, JTL (Josephson Transmission Lines) などの能動素子を用いた配線手法や, PTL (Passive Transmission Lines) を用いた配線手法が提案されている [3][4][5][6]. JTL は安定して SFQ を伝送することが可能な半面, 長距離配線の際には, 複数段の JTL が必要であるため, 配線遅延が大きくなる. 一方で, PTL は超伝導マイクロストリップラインを用いることで磁束量子を電磁波に変換し, 高速に伝搬可能である. しかし, 配線の両端に遅延時間が大きいドライバ (DRV) とレシーバ (REC) を接続する必要があるため, 近距離配線には向かない [7]. そのため, 近距離配線には JTL を, 長距離配線には PTL を用いることで配線遅延が小さな回路を実現できる.

そこで本研究では, JTL と PTL の混合配線を前提とす

¹ 京都大学 大学院情報学研究所

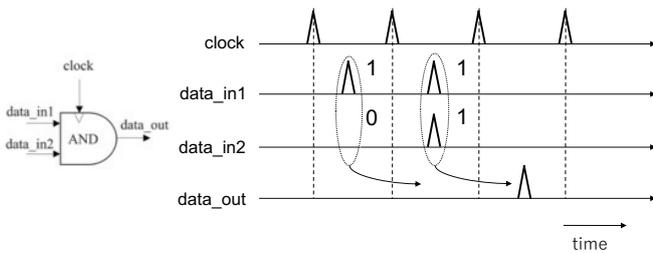


図 1 クロックゲートの動作例

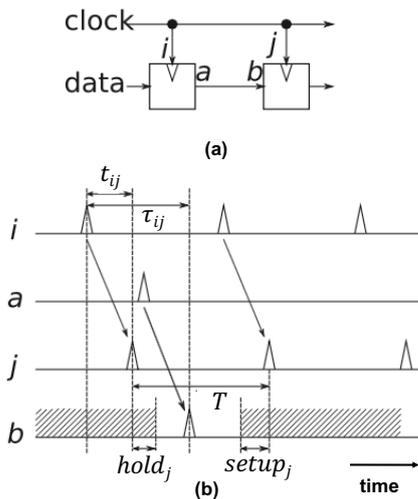


図 2 SFQ 論理回路のタイミング制約

る配置アルゴリズムを提案し、レイアウト設計の自動化と、設計に要する時間の削減を図る。また、提案アルゴリズムをサンプル回路に適用し、JTL と PTL の混合配線手法の有用性を明らかにする。

2. 準備

2.1 SFQ 論理ゲートの動作

図 1 に、SFQ 回路のクロックゲートの動作例を示す。ここでは、AND ゲートを例に説明する。SFQ 回路では、全てのゲートにクロックが供給され、パルスの有無で論理値の 1,0 を決定する。隣接するクロックパルスの中にデータパルスが存在すれば 1、存在しなければ 0 として振る舞う。データパルス到着後、クロックパルスが入力されると結果が出力される。

2.2 クロッキング方式

SFQ 回路は配線遅延の相対的な影響が大きいため、全てのクロックゲートに同時にクロックを入力することは難しい。そこで、SFQ 回路ではクロックスキューを許容したクロッキング方式を用いる。

図 2(a) において、ゲート j にクロックが到着する時刻を T_j 、ゲート j にデータが到着する時刻を T_b 、クロック周期を T とすると、以下の式が成り立つようにクロックを供給する。

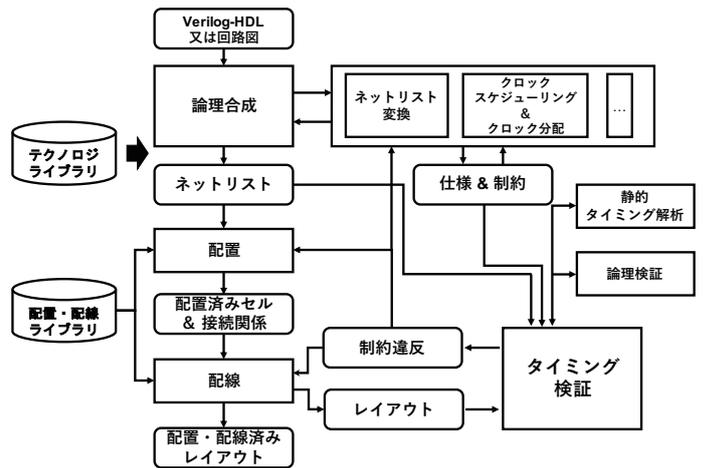


図 3 SFQ 回路の設計フロー

$$T_j < T_b < T_j + T \quad (1)$$

全てのゲートにおいて、クロック入力とデータ出力がこのような関係にあるようにクロックを供給する方式をコンカレントフロー・クロッキングという。

2.3 SFQ 回路のタイミング制約

図 2 (b) に、SFQ 論理ゲートへのデータおよびクロック入力のタイミングチャートを示す。 t_{ij} を 2 つのゲートのクロックピンでのクロックスキュー、 τ_{ij} をゲート i からゲート j の遅延、クロック周期を T とする。論理ゲート j のホールド時間 $hold_j$ 、セットアップ時間 $setup_j$ は各ゲート毎に定められており、全ての論理ゲートで以下の制約を満たす必要がある

$$\tau_{ij} + setup_j < t_{ij} + T \quad (2)$$

$$t_{ij} < \tau_{ij} - hold_j \quad (3)$$

2.4 SFQ 回路の設計フロー

図 3 に SFQ 回路の設計フローを示す。デザインエントリは Verilog HDL などのハードウェア記述言語、あるいはスキーマティックエディタで設計した回路図である。ハードウェア記述言語を入力とした場合は、半導体向けの論理合成ツールを用いて論理合成を行い、SFQ 回路に適したネットリストに変換する。ネットリストの情報を元にセルの配置・配線を行い、回路のレイアウトを決定する。最後に、設計した回路の論理検証を行うことで、回路の論理的な動作を確認する。

2.5 SFQ 回路のセルライブラリ

SFQ 回路は、論理ゲートおよび配線に対応するセルをタイル状に並べて設計する。セル境界にはピンが定義されており、ピンを隣接させることでセル間が接続される。利用可能なセルは、セルライブラリとして用意される。図 4 に SFQ 回路設計で使用する基本的なセルを示す。

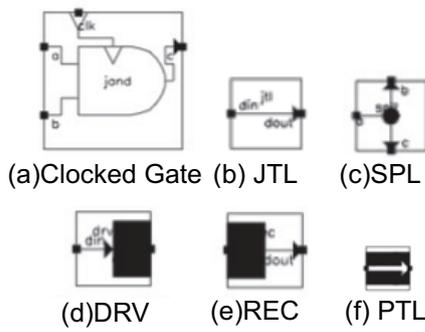


図 4 SFQ 回路のセルライブラリ

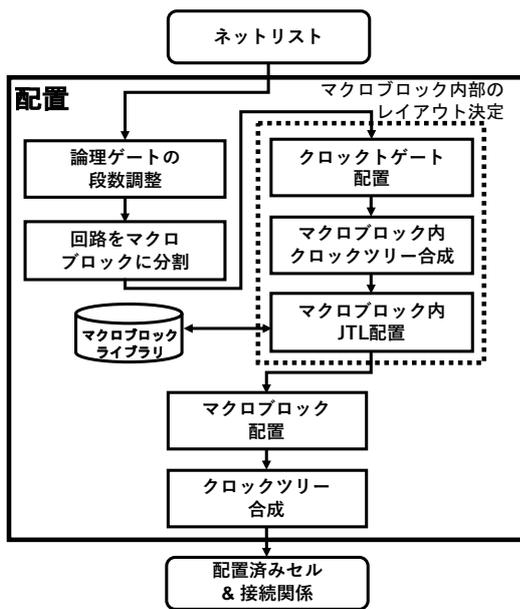


図 5 提案する SFQ 回路の配置フロー

クロックの供給が必要な論理ゲートと、ゲート間の配線要素である JTL と PTL が基本要素となる。PTL は超伝導マイクロストリップラインを用いるため、JTL に比べて高速に磁束量子の伝搬が可能である一方で、磁束量子と電磁波を変換するための DRV/REC セルを接続する必要がある。SFQ 回路では、2 個以上のゲートに対しパルスを伝播させる際、SFQ パルスを 2 つの経路へ分配し、ファンアウトを確保するためのスプリッタが必要である。

3. 提案手法

本章では、SFQ 回路の設計フローにおいて、セルの配置を決定する手法を説明する。図 5 に提案する配置フローを示す。

3.1 論理ゲートの段数調整

本手法では、動作周波数が高い回路を設計するために、コンカレントフロー・クロッキング方式を採用する。この方式では、回路全体がパイプライン動作するため、全てのデータパスに対して論理ゲートの段数を合わせる必要があ

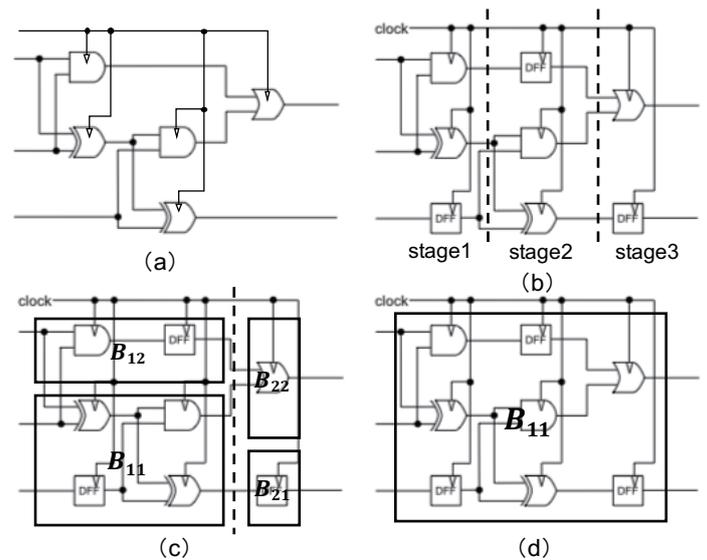


図 6 セルのマクロブロック化: (a) 元の SFQ 回路 (b) 段数調整後の回路 (c) $N_{max} = 4$ の場合 (d) $N_{max} = 8$ の場合

り、D-FF を挿入することで段数調整を行う。

3.2 マクロブロックへの分割

段数調整後の回路において、直接接続されたゲートの集合をブロックとし、回路全体をブロックに分割する。まず、回路をステージごとに分割する (図 6(b))。次に、入力ピンから出力ピンに向かってステージを合併し、互いに接続されたゲートの集合を 1 つのブロックとする (図 6(c)(d))。 N_{max} をブロック内の論理ゲート数の最大値とする。ブロック内の論理ゲート数が N_{max} 以下である間、合併を繰り返し、 N_{max} を超えた場合は最後のステージから改めて合併を始める。以上の手順により生成された各ブロックをマクロブロックと呼ぶ。以降のフローでは、マクロブロック内部の配置とマクロブロック間の配置に分けて説明する。

3.3 マクロブロック内部のレイアウト決定

回路全体をマクロブロックに分割した後、各ブロックごとにマクロブロック内部のレイアウトを決定する。クロックゲートの配置と、クロックネットワークのトポロジを生成した後に、ブロック内に配置する JTL を決定する。

3.3.1 クロックゲート配置

マクロブロックの内部は JTL を用いて配線を行う。マクロブロック内部において、クロックゲート間の配線距離の最大値を最小化することで、遅延時間が小さいマクロブロックを実現する。そのために、接続関係のあるクロックゲートを近距離に配置し、配線距離の最大値が最小となるようにマクロブロック内でセルの配置やピンの割り当てを入れ替える。マクロブロック内部のクロックゲートの集合が与えられ、それらを格子状に配置し、入力ピンか

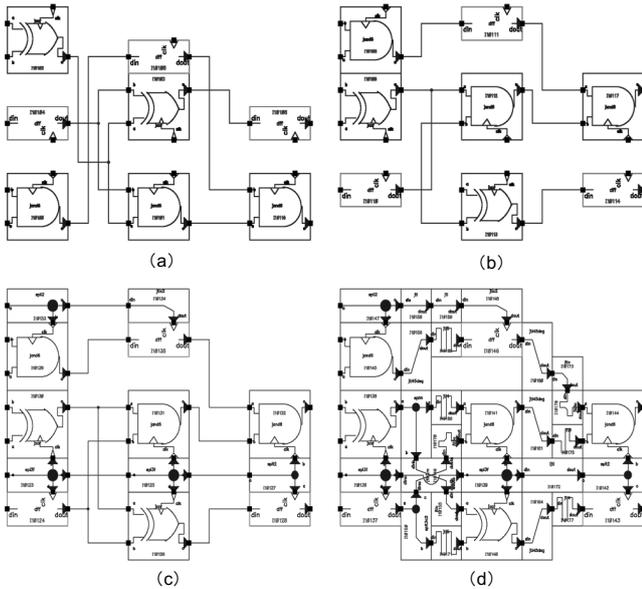


図7 マクロブロック内部のレイアウト例 ((a) 元のマクロブロック (b) 配置順序を入れ替えた後のマクロブロック (c) クロックツリー合成後のマクロブロック (d) JTL 配線後のマクロブロック)

らの段数が等しいクロックゲートは同じ列に配置する。(図7(a)). その際、マクロブロック内の総配線長が最小となるようにクロックゲートを配置する行を総当たりで探索する(図7(b)). これにより、マクロブロックの入力から出力までの遅延時間を最小にすることが可能である。

3.3.2 ブロック内クロックツリー合成

2分岐, 3分岐のスプリッタを用いて, マクロブロック内部のクロックネットワークのトポロジを生成する. マクロブロック内部で, 配置されたクロックゲートの行間に沿ってスプリッタを配置し, 同じ行に配置されるスプリッタ間を接続することで, クロックのトポロジを生成する(図7(c)).

3.3.3 マクロブロック内のJTL配置

マクロブロック内部で配置の最適化を行った後, マクロブロック内部で論理ゲートの配線をJTLで行う. セルライブラリには, 様々な形状と伝搬遅延時間のJTLが用意されており, これをつなぎ合わせることで配線を実現する. 配線の組み合わせは膨大であるが, 有用なパターンは限られるため, マクロブロック内部のレイアウトパターンを予めライブラリとして用意しておき, マクロブロック内の情報をキーにしてライブラリを検索することで, SFQ回路のタイミング制約を満足するレイアウトを決定する(図7(d)).

3.4 マクロブロック配置

マクロブロック内部の配置を決定した後, 全てのマクロブロックの配置を決定する. 全てのマクロブロックを格子状に配置し, 入力ピンからの段数が同じマクロブロックは

Algorithm 1 Macro Block Placer

Input: 未配置マクロブロック $B = \{b_1, b_2, \dots, b_n\}$

Output: マクロブロックの配置順序 $P = (p_1, p_2, \dots, p_n)$

```

1:  $\mathcal{B}_{all} \leftarrow 2^{|B|}$ 
2:  $l(i, j) \leftarrow b_i$  を  $j$  行目に配置する際の配線コスト ( $1 \leq i, j \leq n$ )
3: for  $i \leftarrow 1$  to  $|B|$  do
4:    $\mathcal{V}_i \leftarrow \mathcal{B}_{all}$  の要素数が  $i$  の集合
5:   for each:  $V \in \mathcal{V}_i$  do
6:      $c_V \leftarrow \infty$ 
7:     for each:  $b_j \in V$  do
8:        $W \leftarrow V \setminus \{b_j\}$ 
9:       if  $c_W + l(i, j) < c_V$  then
10:         $c_V \leftarrow c_W + l(i, j)$ 
11:         $p_j \leftarrow i$ 
12:      end if
13:    end for
14:  end for
15: end for

```

同じ列に配置する. その後, 各列で何行目にマクロブロックを配置するかを決定する. その際, マクロブロック間の総配線長が最小となるように配置する. 本節では, マクロブロックの配置アルゴリズムを説明する.

全ての列に対して, 配置するマクロブロックの行を総当たりで求めるのは計算量が膨大であるため現実的ではない. そのため, 入力ピンに近い列のマクロブロックから順に, 前段のマクロブロックとの間の配線長が短くなるように後段のマクロブロックの配置を決定する.

各段の配置順序は, 動的計画法に基づくアルゴリズムで決定する. マクロブロック数 n に対し, 計算時間は $O(2^n)$ である. ある列に配置するマクロブロックの集合 $B = \{b_1, b_2, \dots, b_n\}$ が入力として与えられ, マクロブロックの配置順序 $P = (p_1, p_2, \dots, p_n)$ を出力する. p_j は, ブロック b_j を配置する行番号である. 例えば, 入力が $B = \{b_1, b_2, b_3\}$ で, 出力が $P = (2, 3, 1)$ である場合, ある列の一行目に b_3 を, 二行目に b_1 を, 三行目に b_2 を配置する. 配線コスト c_V を, マクロブロック集合 V を配置した際の, 前段のマクロブロックとの間の総配線長とし, $l(i, j)$ をマクロブロック b_i を j 行目に配置する際の, 前段のマクロブロックとの間の総配線長とする. 以下にアルゴリズムを示す. マクロブロックを i 個配置する際に, マクロブロックを $i-1$ 個配置した際の配線コストの結果を用いる. その際, 配線コストが最も小さくなるようにコストの計算を行う. 例えば, $V = \{b_1, b_2, b_3\}$ を配置する際の配線コストは, 以下の3つの場合のうち, コストが最小となる方の配置順序を採用し, V の配線コスト c_V とする.

- $W = \{b_1, b_2\}$ を配置した時の配線コスト c_W と, b_3 を配置する際の配線コストの和
- $W = \{b_2, b_3\}$ を配置した時の配線コスト c_W と, b_1 を配置する際の配線コストの和
- $W = \{b_3, b_1\}$ を配置した時の配線コスト c_W と, b_2 を配置する際の配線コストの和

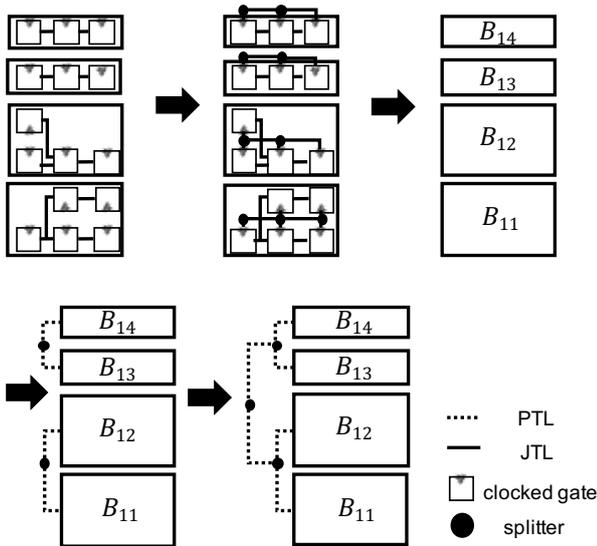


図 8 クロックツリー合成フロー

この手順を $|B|$ 回繰り返す。 $B = \{b_1, b_2, \dots, b_n\}$ の配線コストが最小となるように、配置順序を決定する。以上の手順を、入力ピンから出力ピンまで順番に、各列で行う。

3.5 クロックツリー合成

これまでにクロックツリー合成のアルゴリズムがいくつか提案されている [8][9][10]。 [8][9] では大規模な H ツリーネットワークのオーバーヘッドを削減するために、クロックネットワークに HL ツリーを用いている。しかし、L ツリーによるクロックスキューの影響が大きく、クロック周波数を大きくすることが困難である。一方 [10] では、同一段の論理ゲートごとにクロックツリーを生成しているため、クロック周波数を上げやすい一方で使用するジョセフソン接合数 (JJ 数) が多くなってしまふ。そこで、同一段のマクロブロックごとにクロックツリーを生成し、マクロブロックごとにクロックネットワークを構築することで、使用する JJ 数を削減する。図 8 にクロックツリー合成の流れを示す。

2 分岐、3 分岐のスプリッターを使用し、入力ピンから出力ピンに向かってマクロブロックのクロックネットワークのトポロジを決定する。その際、SPL/DRV/REC セルを用いて木を生成し、入力ピンからの段数が同じマクロブロックに対して、クロックパルスの到着時刻が等しくなるようにスプリッター間を接続する。各マクロブロックへのクロック配線は PTL を用いる。

4. 提案手法の実装と評価

4.1 評価環境

本研究では、Cadence Design Systems 社の Virtuoso 上で SFQ 回路の設計を行い、セルライブラリは、ADP プロセス向けのものを用いた [11]。論理合成ツールには、Synopsys 社の Design Compiler を用いた。提案手法の実装は

Java 言語で行い、SFQ 回路のネットリストを入力として、論理ゲートや JTL の座標が記述されたファイルを出力し、Virtuoso にロードする。ツールの実行環境は、Intel Core i7 1.7GHz、メモリ 8GB を搭載した MacBook Air を使用する。

4.2 評価結果

提案手法をツールとして実装し、JTL と PTL の混合配線を行った回路の回路性能の見積もりを行い、PTL のみで配線を行った回路との性能比較を行った。さらに、JTL と PTL の混合配線を行う回路において、マクロブロックのサイズを変化させた時の回路性能の見積もりを行った。

図 9 に提案手法を適用した場合の 8 ビット桁上げ伝搬加算器を示す。表 1 に評価結果を示す。

回路面積は、提案手法を用いた結果、PTL のみで配線を行ったレイアウトと比べ、10%以上削減することができた。原因としては、クロックツリーの深さが小さくなったことが考えられる。PTL のみで配線を行う回路は、ゲート 1 段ごとにクロックツリーを生成するため、ゲートの段数分の木が生成される。一方、提案手法では、同じ列のマクロブロックごとにクロックツリーを生成するため、PTL のみで配線する回路と比べてクロックツリーの深さは小さくなる。

本手法では、コンカレントフロー・クロッキングを用いており、回路全体がパイプライン動作する。そのため、回路の段数を n とすると、 $n+1$ サイクル後にデータが出力される。回路のレイテンシは、 N_{max} が大きいほど小さくなることを確認した。しかし、 N_{max} を大きくしすぎると、マクロブロック内のクロックゲートの配線距離が大きくなってしまふため、マクロブロックの入力から出力までの遅延時間が大きくなってしまふことが推測される。

実行時間は、Java プログラムの実行時間であり、 N_{max} が大きいほど実行時間が短いことがわかる。これは、 N_{max} を大きくするとマクロブロックの数が少なくなるからである。しかし、 N_{max} を大きくしすぎるとマクロブロック内部の配置の計算量が膨大となるため、パラメータの適切な値を設定する必要がある。

5. おわりに

本稿では、SFQ 回路向けのセル配置アルゴリズムを提案した。配線遅延の小さな PTL のみで配線を行うよりも、JTL と PTL の両方を使い分けて配線を行う方が回路のレイテンシが小さくなることを確認した。

今後は、マクロブロック内部の情報をどのように利用して、マクロブロックライブラリから適切なマクロブロックを選択するかを検討し、マクロブロックライブラリの詳細な実装を行う予定である。その後、マクロブロック間を PTL で配線するためのアルゴリズムを検討し、SFQ 回路

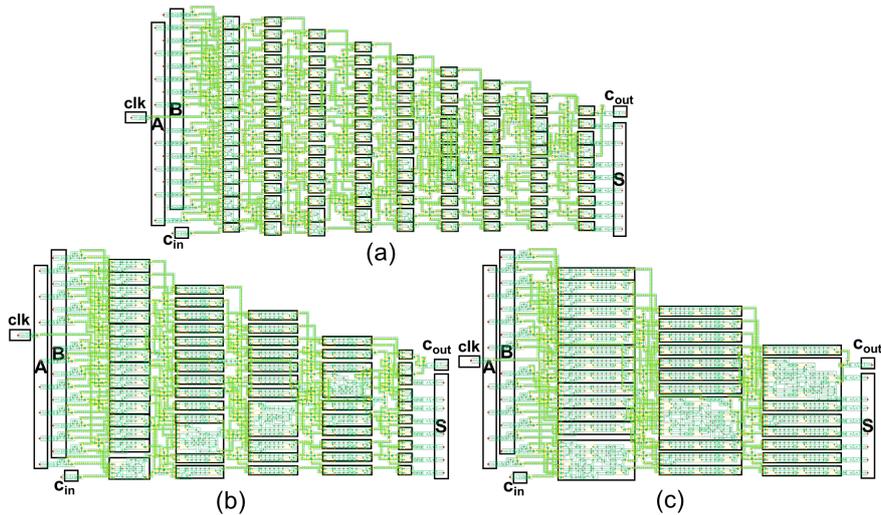


図 9 8bit Ripple Carry Adder のレイアウト ((a) $N_{max} = 1$ の回路 (b) $N_{max} = 4$ の回路 (c) $N_{max} = 8$ の回路)

表 1 パラメータを変化させた時の回路性能の比較

N_{max}	回路面積 [mm^2]	JJ 数	レイテンシ [ps]	動作周波数 [GHz]	実行時間 [s]
1	5.2483	5106	$330 + 20 * 10$	50	1865
4	4.5261	5104	$280 + 20 * 10$	50	246
8	4.4298	4968	$275 + 20 * 10$	50	26

の設計効率向上を目指す。

謝辞 本研究はJSPS 科研費 18H05211 および 18K11213 の助成を受けたものである。本研究は東京大学大規模集積システム設計教育研究センターを通し、日本ケイデンス株式会社、シノプシス株式会社の協力で行われたものである。

参考文献

- [1] K. K. Likharev and V. K. Semenov: “RSFQ logic/memory family: A new josephson-junction technology for sub-terahertz-clock-frequency digital system”, IEEE Transactions on Appl Supercond Vol. 1, No. 1, pp. 3–28, 1991.
- [2] K. K. Likharev and V. K. Semenov: “Advancement of superconductor digital electronics”, IEICE Electronics Express Vol. 9, No. 22, pp. 1720–1734, 2012.
- [3] Y. Hashimoto, S. Yorozu, Y. Kameda, A. Fujimaki, H. Terai, and N. Yoshikawa “Development of Passive Interconnection Technology for SFQ Circuits”, IEICE Trans. Electron vol. E88-C, No. 2, pp. 198–207, 2005.
- [4] Y. Kameda and S. Yorozu and Y. Hashimoto: “A new design methodology for single-flux-quantum (SFQ) logic circuits using passive-transmission-line (PTL) wiring”, IEEE Trans. Appl. Supercond. Vol. 17, No. 2, pp. 508–511, 2007.
- [5] M. Tanaka, K. Obata, Y. Ito, S. Takeshima, M. Sato, K. Takagi, N. Takagi, H. Akaike and A. Fujimaki: “Automated Passive-Transmission-Line Routing Tool for Single-Flux-Quantum Circuits Based on A* Algorithm”, IEICE Trans. Electron, Vol. E93-C, No. 4, pp. 435–439, 2010.
- [6] N. Kito and K. Takagi and N. Takagi: “Automatic Wire-Routing of SFQ Digital Circuits Considering Wire-Length Matching”, IEEE Transactions on Appl Supercond Vol. 26, No. 3, Article 1300305, 2016.
- [7] H. Suzuki, S. Nagasawa, K. Miyahara, and Y. Enomoto: “Characteristics of Driver and Receiver Circuits with a Passive Transmission Line in RSFQ Circuits”, IEEE Trans. Appl. Supercond. Vol. 10, No. 3, pp. 1637–1641, 2000.
- [8] S. Shahsavani and T. Lin and A. Shafaei and C. J. Fourie and M. Pedram: “An Integrated Row-Based Cell Placement and Interconnect Synthesis Tool for Large SFQ Logic Circuits”, IEEE Trans. Appl. Supercond. Vol. 27, No. 4, 2017.
- [9] S. Shahsavani and T. Lin and A. Shafaei and C. J. Fourie and M. Pedram: “A Placement Algorithm for Superconducting Logic Circuits Based on Cell Grouping and Super-Cell Placement”, Design, Automation And Test in Europe, 2018.
- [10] K. Takagi and Y. Ito and S. Takeshima and M. Tanaka and N. Takagi: “Layout-Driven Skewed Clock Tree Synthesis for Superconducting SFQ Circuits”, IEICE Trans. Electron Vol. E94C, No. 3, pp. 288–295, 2011.
- [11] S. Nagasawa et al., “Nb 9-layer fabrication process for superconducting large-scale SFQ circuits and its process evaluation”, IEICE Trans. Electron., vol. E97-C, no. 3, pp. 132–140, 2014.