

# 単一磁束量子回路によるビット幅可変加減算器の設計と評価

石川 伊織<sup>1</sup> 長岡 一起<sup>2</sup> 石田 浩貴<sup>1,†1</sup> 福光 孝介<sup>1</sup> 岡 慶太郎<sup>1</sup> 田中 雅光<sup>2</sup> 川上 哲志<sup>1</sup>  
谷本 輝夫<sup>1</sup> 小野 貴継<sup>1</sup> 藤巻 朗<sup>2</sup> 井上 弘士<sup>1</sup>

**概要:** 半導体の微細化の限界が見え始め、新奇デバイスへの期待が高まっている。そのひとつに単一磁束量子 (Single Flux Quantum: SFQ) 回路がある。本研究では SFQ 回路と親和性の高いベクトルアーキテクチャに着目し、SFQ ベクトルプロセッサを実装することを目標としている。しかしながら、現状では SFQ 回路向けのベクトルアーキテクチャは提案されていない。そこで、近年主流の柔軟性を持ったベクトルプロセッサに必要な構成要素である、ビット幅可変加減算器の設計を行い、動作検証を行った。シミュレーションによる検証の結果、50 GHz での正常動作が確認でき、SFQ 回路の高速性を維持しつつ、ビット幅可変加減算器が実現できることが確認できた。この回路はチップ試作を行っており、試作チップを用いた検証も行う予定である。

## 1. はじめに

半導体コンピュータは微細化に伴う高速化、低消費電力化によって発展してきた。しかしながら、半導体デバイスの微細化は限界に近づき、さらなる高速化、低消費電力化が難しくなっている。この問題を根本的に解決するために、半導体に替わる新奇デバイスが求められている。そのようなデバイスのひとつに単一磁束量子 (Single Flux Quantum: SFQ) 回路がある [1]。超伝導ジョセフソン接合を用いて構成されるこのデバイスは、超伝導体特有の高速性・低消費電力性を備えており、半導体回路の十倍以上の速度、千分の一以下の消費電力で動作する回路が実証されている [2]。

しかしながら、SFQ 回路は半導体とは異なる点が複数存在し、半導体に用いられてきた手法をそのまま適用するだけでは、その性能を発揮することはできない。たとえば、電圧の高低ではなくパルスの伝播でデジタル情報を表現する点、各論理ゲートがラッチ機能を備えている点、などがある。そのため、SFQ 回路を用いて高い性能を達成するためにはこれらを考慮したアーキテクチャが必要である。たとえば、各論理ゲートがパイプラインレジスタを兼ねる構造を採るゲートレベルパイプライン方式が適していることが示されている [3]。しかしながら、あらゆる回路に対して単純に適用できるわけではない。ここで、マイクロプロセッサへのゲートレベルパイプラインの適用を考える。そ

の場合、極めて深い命令パイプライン構造となる傾向にあり、各命令の実行レイテンシが増大する。また、SFQ 回路ではフィードバックループ構造の実装は苦手としており、動作速度が低下するため、従来のプロセッサで適用されているフォワーディング機構の導入は難しい。その結果、実行命令列にフロー依存が存在する場合にはパイプラインストールが頻発し、大幅な性能低下を招く可能性が高い。

この問題を解決すべく、現在我々は SFQ ベクトルプロセッサ・アーキテクチャに関する研究を行っている。ベクトルアーキテクチャではデータレベル並列性を利用して、処理を時間方向に展開する。SFQ 回路では追加のハードウェアなしにゲートレベルパイプラインが実装可能であるため、深いパイプライン構造を利用するベクトルアーキテクチャとの親和性が高いと考えられる。ベクトルプロセッサは演算器やデータパスなど、スカラプロセッサとは異なったコンポーネントが必要となるが、SFQ 回路の高速性を維持しつつ、ベクトルプロセッサ向けコンポーネントが実現可能であるかは不明である。よって、SFQ ベクトルプロセッサの設計のためにはコンポーネントレベルでの実証を重ねる必要がある。そこで、本研究ではベクトルプロセッサの主要構成要素となるベクトル演算器に着目し、SFQ 回路での実装可能性を評価する。具体的には、RISC-V Vector Extension [4, 5] で採用されている柔軟性を持ったベクトルプロセッサを念頭に置き、その実現に必要なビット幅可変加減算器の回路設計を行った。ポストレイアウト・シミュレーションの結果、50 GHz での高速動作が確認できた。

<sup>1</sup> 九州大学

<sup>2</sup> 名古屋大学

<sup>†1</sup> 現在、株式会社ソシオネクスト

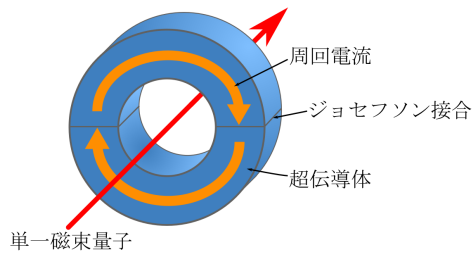


図 1: SFQ 回路

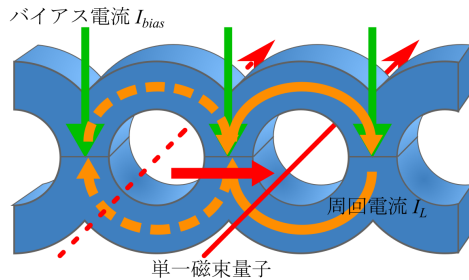


図 2: SFQ 回路による磁束の伝搬

本稿の構成は以下の通りである。第 2 章と第 3 章で本研究の基礎である SFQ 回路とベクトルアーキテクチャについてまとめる。次に、第 4 章で加減算器の仕様策定について、第 5 章で設計の詳細について述べる。その後、第 6 章で設計した加減算器のポストレイアウト・シミュレーションによる評価を行う。第 7 章では SFQ プロセッサについての関連研究についてまとめる。最後に、第 8 章で本稿をまとめる。

## 2. 単一磁束量子回路

### 2.1 動作原理

単一磁束量子 (Single Flux Quantum: SFQ) 回路はジョセフソン接合を含む超伝導ループから成る回路である [1]。SFQ 回路の模式を図 1 に示す。このループ内の磁束  $\Phi$  は離散的な値しかとることができず、 $\Phi = n\Phi_0$  ( $n$  は整数) で表される。 $\Phi_0 = \frac{h}{2e} \simeq 2.07 \times 10^{-15} \text{Wb}$  のことを、磁束量子と呼ぶ。SFQ 回路では  $n = 0$  または  $1$  となるように設計される。このループ内に磁束が存在する場合、ループ内に周回電流  $I_L$  が流れる。

また、図 2 のように接合にはバイアス電流  $I_{bias}$  を流しておく。図 1 の右の接合には  $I_L + I_{bias}$  の電流が、左の接合には  $-I_L + I_{bias}$  の電流が流れ、二つの接合のうち周回電流とバイアス電流の向きが同じ方の接合がスイッチする。 $I_L + I_{bias} < I_c$  の場合は接合がスイッチすることなく、磁束はループ内にとどまる。 $I_L + I_{bias} > I_c$  の場合は接合はスイッチし、磁束が次段のループへ移動し、ループ内に磁束は存在しなくなる。ここで、 $I_c$  はジョセフソン接合の臨界電流値である。このように、ジョセフソン接合の

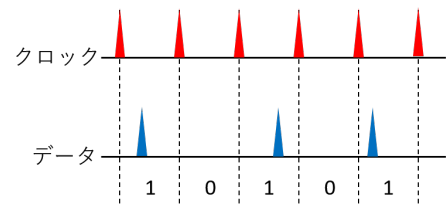


図 3: パルス論理のクロックとデータの関係

スイッチを介して磁束の保持・伝搬を制御でき、インダクタンスの調整やループの組み合わせによって接合に流れる周回電流の大きさを調整することで、様々な論理ゲートや伝送路を実現する。

### 2.2 パルス論理

SFQ 回路では接合がスイッチするときに発生する微弱なパルス (SFQ パルス) によって情報を伝達する。半導体回路が電圧レベルの高低により '1'/'0' を表現する「レベル論理」であるのに対し、SFQ 回路ではこの SFQ パルスの有無で '1'/'0' を表現する「パルス論理」である。

SFQ 回路におけるパルス論理では、「パルスなし」と「パルス未着」を区別するためにクロック信号を用いたある種のタイムウィンドウを定める。このクロックも SFQ パルスとして伝搬し、これをクロックパルスと呼ぶ。これに対して、データを表すパルスをデータパルスと呼ぶ。つまり、図 3 に示すように、クロックパルスの到着までにデータパルスが到着すれば「パルスあり」、到着しなければ「パルスなし」とする。

このように、SFQ ゲートはクロックパルスが到着するまで入力データパルスを待ち、それを論理ゲート内に留め、クロックパルスが到着すると演算を行い出力をする。これは、SFQ ゲートがラッチ機能を備えることによって実現できるものである。

### 2.3 ゲートレベルパイプライン

SFQ 回路の特徴を利用した構造に、ゲートレベルパイプラインがある。ゲートレベルパイプラインとは論理ゲート 1 段がパイプラインステージ 1 段を構成するパイプライン構造である。このように、細かくステージが区切られたパイプライン構造を細粒度パイプラインと呼ぶ。ゲートレベルパイプラインは最も細粒度なパイプライン構造であるといえる。

SFQ 論理ゲートはラッチ機能を持つため、パイプラインステージを構成するためにパイプラインレジスタを挿入する必要がなく、ゲートレベルパイプラインを容易に実装できる。また、細粒度パイプラインは、パイプラインレジスタ間のデータパスが短くなることによって、回路の動作周波数を高めることができ、SFQ 回路の高速性を活かすことができる。

また、SFQ回路では第2.2節で述べたとおり、データパルスの「なし」と「未着」を区別するために、すべての論理ゲートをクロックパルスで駆動する。そのため、通常のプロセッサで採用されている粗粒度パイプライン構造（各パイプラインステージにて複数段の論理ゲートが存在する構造）をSFQで実現する場合、パイプラインステージ内論理ゲートを駆動するためのクロック信号（ローカルクロック信号）と、パイプラインステージ間を制御するクロック信号（グローバルクロック信号）が必要となる [6]。そのため、制御が複雑化し、必要な配線量も増大する。一方で、ゲートレベルパイプラインではローカルクロックとグローバルクロックが共通化される。このことからSFQ回路においてゲートレベルパイプラインを採用することの利点は大きいと言える。実際にゲートレベルパイプラインを採用したプロセッサ [7] や ALU [8]、乗算器 [9] が実証されている。

### 3. ベクトルアーキテクチャ

#### 3.1 ベクトルアーキテクチャの概要

ベクトルアーキテクチャはプログラム中に存在するデータレベル並列性を利用してスループットを高めるものである [4, 10, 11]。ベクトルアーキテクチャを採用したコンピュータとしては Cray-1 [12] が有名である。データレベル並列性を持つデータ構造としては配列があり、科学技術計算やマルチメディア処理など、様々なプログラムで使われている。

ベクトルアーキテクチャではデータレベル並列性を利用して性能向上するが、並列度に応じた複数の演算器を用いてすべての要素を同時に演算するわけではない。ベクトルプロセッサでは、主記憶から読み出したデータをベクトルレジスタへ格納し、ベクトル演算器へ連続転送して実行する。すなわち、時間並列処理に基づく実行モデルであり、その動作周波数の向上が高性能化への重要な鍵となる。

命令レベル並列性を拠り所とする（スーパー）スカラアーキテクチャにおいては、パイプラインが細粒度になるとパイプライン中に多くの命令が滞在（インフライトな命令の数が増加）するため、データハザードの発生頻度の増加にともない性能が低下する。一方、ベクトルアーキテクチャではデータレベル並列性を活用するため、細粒度パイプラインにおいもフロー依存の発生頻度の低下を期待でき、その利点を十分に利用できる。

#### 3.2 SFQ回路との親和性

SFQ回路では第2.3節で述べたとおり、粗粒度パイプラインを採用するよりも、細粒度パイプラインであるゲートレベルパイプラインを採用する方が効率が良い。しかしながら、ゲートレベルパイプラインを採用するプロセッサが既存の半導体プロセッサの性能を大幅に上回るためには、

ハザードによるパイプラインのストールを99%以上隠蔽する必要があることがわかっている [3]。

パイプラインのストールを隠蔽する一般的な方法は分岐予測やフォワーディング、アウトオブオーダー実行などがある。しかしながら、SFQ回路はピコ秒オーダーのタイミング設計を必要とするため、複雑な論理回路を形成する必要のある分岐予測器やリオーダーバッファ、大きなフィードバックループを形成する必要があるフォワーディング機構の実装において50GHzなどの高速動作を維持することは難しい。

そこで、回路の複雑化を招くことなしにパイプラインストールを隠蔽する方法として、ベクトルアーキテクチャの採用が考えられる。先述のとおり、ベクトルアーキテクチャはデータレベル並列性を利用して、互いに依存関係のない演算でパイプラインを埋め尽くす。つまり、パイプライン段数に対して十分に大きなベクトル長を有する限り、データハザードに起因するストールを回避できる。よって、SFQ回路とベクトルアーキテクチャの親和性は高いと考えられる。

### 4. 加減算器の仕様策定

#### 4.1 要求仕様

近年ではRISC-V Vector Extension [4, 5] など、データ型やベクトル長などの演算パラメータをプログラム実行中に変更可能な柔軟な命令セットが開発されている。本研究ではRISC-V Vector Extensionの採用を基本としており、本稿ではSFQベクトルプロセッサの重要構成要素となる演算器に焦点を当てる。

当命令セットでは、ビット数（演算ビット幅）が異なる複数のデータ型でベクトル演算が可能である。このような複数の演算ビット幅をサポートする場合、その実装選択肢として以下が考えられる。ここで、サポートすべき演算ビット幅をそれぞれ  $BW_8 = 8$ ,  $BW_{16} = 16$ ,  $BW_{32} = 32$  とする。

**固定演算ビット幅逐次処理方式** ベクトル演算器が対応可能なビット幅は固定とし、 $BW_8$ ,  $BW_{16}$ ,  $BW_{32}$  それぞれのベクトル演算器を実装する。各演算器の構成は単純化できる一方、とあるビット幅のデータに対する演算実行中は残り2つの演算器を利用できず、ハードウェア資源利用率が低下するといった問題が生じる。

**可変演算ビット幅逐次処理方式**  $BW_8$ ,  $BW_{16}$ ,  $BW_{32}$  の全てをサポートする単一のベクトル演算器を実装する。実装すべきベクトル演算器は単一であり、固定演算ビット幅方式と比較してハードウェア資源利用率を高めることができる。しかしながら、 $BW_8$  ならびに  $BW_{16}$  での演算を実行する場合には出力の上位ビットを切り捨てることとなり、依然としてベクトル演算器内でのハードウェア資源利用率の低下が発生する。

**可変演算ビット幅並列処理方式** 可変演算ビット幅逐次処

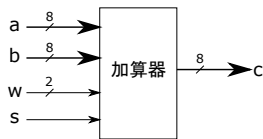


図 4: 演算器の入出力図

表 1: ビット幅指定信号  $w$  と演算ビット幅の関係

$w$	ビット幅
00	2 ビット
01	4 ビット
10	8 ビット
11	-

理方式と同様、 $BW_8$ ,  $BW_{16}$ ,  $BW_{32}$  の全てをサポートする単一のベクトル演算器を実装する。そして、 $BW_8$  の場合は 4 要素、 $BW_{16}$  の場合は 2 要素を並列に処理する ( $BW_{32}$  の場合は逐次処理)。これにより、全てのデータ型においてハードウェア資源利用率を高めることができる。その結果、 $BW_8$  ならびに  $BW_{16}$  の実行においては、空間並列性の活用により実効性能が向上する。ただし、データ型に応じて演算ビット幅と並列度を設定可能とする必要があり、回路の複雑化が予想される。

本研究においては、ハードウェア資源を無駄なく利用可能な可変演算ビット幅並列処理方式を採用する。

## 4.2 機能仕様

本研究の目的はビット幅可変の演算器の実現可能性を確認することであるため、実際のベクトルアーキテクチャが要求するビット幅の演算器ではなく、最大 8 ビット幅の加減算器を設計する。たとえば、RISC-V が要求する演算器は最低 32 ビット必要であり、チップ試作を鑑みた場合には面積制約を満たすことができない。そこで、過去に ALU [8, 13] で実証されている 8 ビットを最大幅として設定した。この加減算器は 2 ビット 4 要素、4 ビット 2 要素、8 ビット 1 要素の 3 つのビット幅に対応する。

設計する加減算器の入出力を図 4 に示す。この加減算器は加算と減算をする回路で、入力として 8 ビットのオペランド  $a$ ,  $b$ , 2 ビットのビット幅指定信号  $w$ , 1 ビットの減算モード指定信号  $s$  を持ち、出力として 8 ビットの演算結果  $c$  を持つ。入力  $a$  と  $b$  に対し、 $w$  で指定されたビット幅による加算 ( $s$  が 0) または減算 ( $s$  が 1) を実行する。 $w$  と演算のビット幅の対応は表 1 のとおりである。

## 5. 加減算器の設計

### 5.1 論理回路の設計とシミュレーション方法

#### 5.1.1 論理回路の設計

本研究では、論理ゲートや配線などの基本的な回路をあ

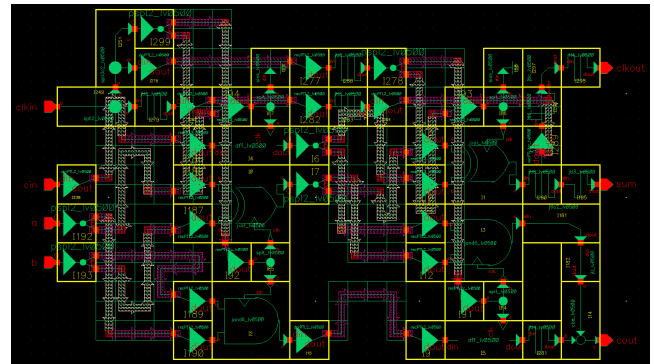


図 5: スケマティックの例

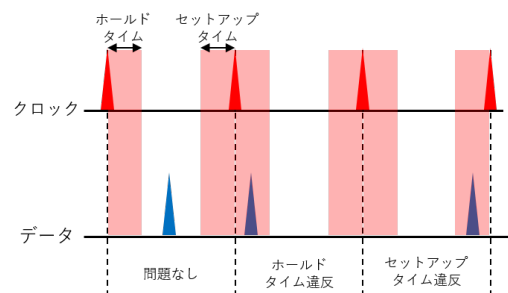


図 6: タイミング制約

らかじめセルライブラリとして準備し、セルを組み合わせで大規模回路を構成するセルベース設計を行う。この手法は階層設計手法であり、セル同士の接続やバイアス電流の供給などの問題はセルの設計時に解決されており、論理回路の設計では回路の動作や信号のタイミングのみに注意すれば良い。設計は Cadence 社の Virtuoso 上で行う。このツール上では大きさや入出力端子などセルの物理的な形を表したシンボルをタイル状に並べて設計を行う (図 5)。このシンボルを並べたものをスケマティックと呼ぶ。SFQ 回路では配線の遅延を考慮したタイミング設計が必要で、回路の論理設計と配置配線を同時に行うため、このように物理的なレイアウトを反映したスケマティックを用いて設計を行う。

スケマティックの作成後、各セルに含まれる遅延情報を用いて静的タイミング解析をし、信号タイミングの微調整を行う。SFQ 回路は数ピコ秒幅のパルスを用いて情報の伝搬を行うため、このタイミング調整が動作周波数を左右する重要な作業となる。各セルにはクロック入力後にデータ入力が禁止されている時間 (ホールドタイム) とデータ入力後にクロック入力禁止されている時間 (セットアップタイム) があり (図 6)、すべてのセルに対するパルスの入力がこれらのタイミング制約を満たすように配線を調整する。スケマティックの各セルには、遅延、ジョセフソン接合数、バイアス電流量、セル面積、などの情報が含まれており、これらに基づき回路全体のバイアス電流値や回路面積を見積もることができる。

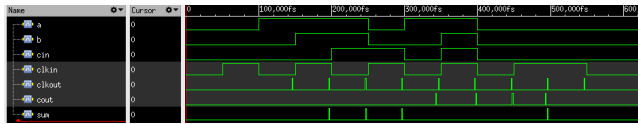


図 7: シミュレーション結果の例

### 5.1.2 ポストレイアウトシミュレーション

設計した回路は同じく Cadence 社の NC-Verilog シミュレータを用いてポストレイアウトシミュレーションを行い動作確認する。シミュレーションの際、設計した回路への入力を Verilog で記述し、シミュレーションの結果は Cadence 社の Sim Vision で波形として確認する (図 7)。

## 5.2 内部構成

設計したビット幅可変加減算器の構造を図 8 に示す。加減算器の低遅延化を目的に桁上げ先見加算方式を採用した。以下 A-I は図中のモジュールを示す。

- (A)  $s$  と  $b$  の排他的論理和を取ることで、 $s$  が 1 の場合  $b$  ビットを反転させる。これは減算をするときに  $b$  の符号を反転させるための演算である。
- (B)  $a$  と  $b$  の和  $h[i] = a[i] \oplus b[i]$ 、各ビットでのキャリーが生成されるかどうかを示す  $g[i] = a[i] \cdot b[i]$ 、各ビットが下位からのキャリーを上位に伝搬するかどうかを示す  $p[i] = a[i] + b[i]$  を求める。
- (C) 2 ビット目から 3 ビット目、5 ビット目から 6 ビット目へのキャリーの伝搬を制御する  $w[0] \cdot w[1]$  および  $\overline{w[0] + w[1]}$  を求める。
- (D) 4 ビット目から 5 ビット目へのキャリーの伝搬を制御する  $w[1]$  および  $\overline{w[1]}$  を求める。
- (E)  $g[3]$ ,  $p[3]$  はそれぞれ  $w[1]$  と、 $g[1]$ ,  $g[5]$ ,  $p[1]$ ,  $p[5]$  はそれぞれ  $w[0] \cdot w[1]$  と論理積をとることで、キャリーの生成・伝搬を  $w$  の値に応じて制御する。
- (F)  $(w[0] + w[1]) \cdot s$  を求める。これは減算をするときに 3 ビット目と 7 ビット目にキャリーを挿入するための信号である。
- (G)  $\overline{w[1]} \cdot s$  を求める。これは減算をするときに 5 ビット目にキャリーを挿入するための信号である。
- (H)  $g[2]$ ,  $g[6]$  と F で求めた信号、 $g[4]$  と G で求めた信号それぞれ論理和を取ることで、 $s$  と  $w$  に応じてキャリー生成信号を挿入する。
- (I) 4 段目までに求めたキャリー生成・伝搬信号を基に桁上げ先見による加算を行う。

### 5.3 設計目標

これまで、ビットパラレル方式のプロセッサとしては細粒度マルチスレッディング方式の 4 ビットプロセッサ [7] が 30 GHz で動作実証されている。これはビット幅固定のプロセッサであり、そのビット幅も 4 ビットと異なること

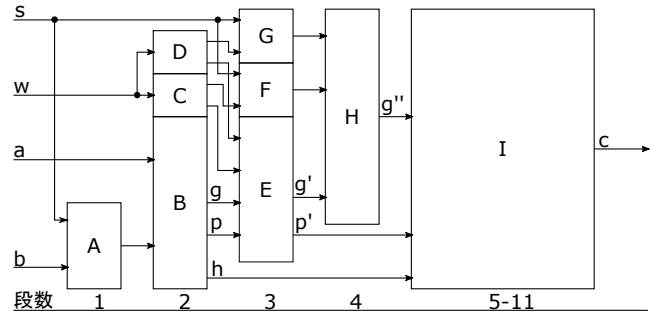


図 8: 加減算器のブロック図

から単純に比較することはできないが、ゲートレベルパイプラインを採用するなど構造的には似ている。このことから、ゲートレベルパイプラインを採用したビットパラレル方式の SFQ プロセッサの動作周波数として、30 GHz を目標とした。

## 5.4 設計手順

設計のフローおよび各作業に使用するツールを図 9 に示す。まず、回路構造を導出し、各配線の距離を考慮しつつフロアプランを決定する。その後、第 5.1 節で述べたように、Virtuoso 上でセルを配置配線してスキマティックを作成し、タイミング調整を行う。タイミングに問題がなければ、NC-Verilog で入力周期を変更しながらテストパターンを入力して、Sim Vision で出力を確認し、正常に動作する周波数範囲を確認する。出力に問題がある場合、または目標動作周波数に達しない場合は再びセルの配置を見直して、タイミング調整をやり直す。これを目標の周波数で正常動作するまで繰り返す。

## 6. 設計回路の評価

### 6.1 動作検証

設計した加減算器のスキマティックを図 10 に示す。ポストレイアウトシミュレーションにおいて表 2 に示すテストパターンを入力し、出力が期待値と一致することを確認した。この結果、設計電圧周辺 ( $2.5 \text{ mV} \pm 20\%$ ) では 50 GHz での正常動作を確認した。

### 6.2 評価

#### 6.2.1 性能評価

第 5.1 節で述べたポストレイアウトシミュレーションにより、正常に動作する周波数を求めた。その結果、50 GHz での正常動作を確認した。これは設計目標値である 30 GHz を上回っており、今回設計した加減算器はベクトルプロセッサの演算器として十分な性能があるといえる。なお、2 ビット幅では 4 つのデータを同時に演算することができるため、理論ピーク性能は 200 GOPS となる。

#### 6.2.2 電力評価

ここでは、ジョセフソン接合数から消費電力を算出す

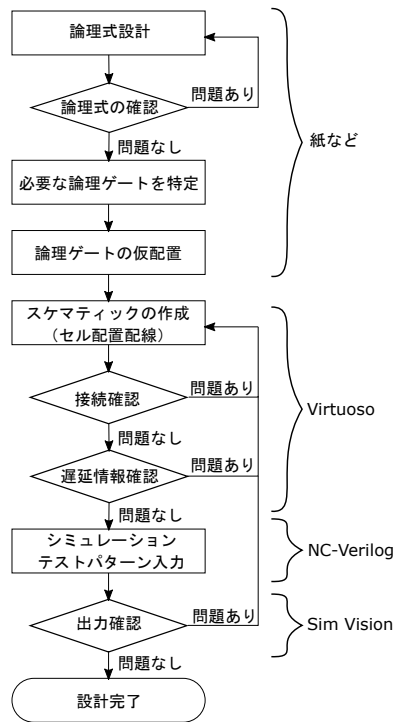


図 9: 設計フロー

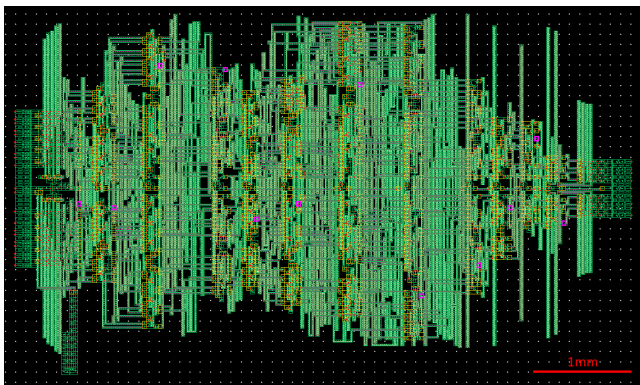


図 10: 加減算器のスキマティック

る. SFQ 回路が全体で消費する電力  $P_{all}$  は動的消費電力  $P_{dynamic}$  と静的消費電力  $P_{static}$  の和で表され,  $P_{dynamic}$  と  $P_{static}$  はそれぞれ式 (1), 式 (2) で表される.

$$P_{dynamic} = \alpha \Phi_0 I_c f N_{JJ} \quad (1)$$

$$P_{static} = V_{bias} I_{bias} N_{JJ} \quad (2)$$

ただし,  $\alpha$  はスイッチング確率,  $f$  は動作周波数,  $V_{bias}$  はバイアス電圧,  $N_{JJ}$  はジョセフソン接合数である. 今回の評価では,  $I_c$  は 0.10 mA,  $V_{bias}$  は 2.5 mV,  $f$  は 50 GHz を用いた. また, 設計の結果,  $N_{JJ}$  は 7,845 接合で  $I_{bias}$  は 871.121 mA であった. よって, 式 (2) より, 静的消費電力は 2.18 mW と求められた. また, 動的電力は式 (1) より,  $\alpha = 1.0$  の最悪ケースで 0.0811 mW であり, 最大消費電力は 2.26 mW となった.

### 6.2.3 面積評価

第 5.1 節で述べたとおり, セルに含まれる情報から Vir-

表 2: 加減算器のテストパターンとそれに対する出力

オペランド a	オペランド b	ビット幅 w	減算 s	出力 c
00 00 00 00	00 00 00 00	00	0	00 00 00 00
11 11 11 11	00 00 00 01	00	0	11 11 11 00
11 11 11 11	00 00 00 10	00	0	11 11 11 01
11 11 11 11	00 00 01 00	00	0	11 11 00 11
11 11 11 11	00 01 00 00	00	0	11 00 11 11
11 11 11 11	01 00 00 00	00	0	00 11 11 11
11 11 11 11	00 00 00 01	01	0	11 11 00 00
11 11 11 11	00 00 00 10	01	0	11 11 00 01
11 11 11 11	00 00 01 00	01	0	11 11 00 11
11 11 11 11	00 01 00 00	01	0	00 00 11 11
11 11 11 11	01 00 00 00	01	0	00 11 11 11
11 11 11 11	00 00 00 01	10	0	00 00 00 00
11 11 11 11	00 00 00 10	10	0	00 00 00 01
11 11 11 11	00 00 01 00	10	0	00 00 00 11
11 11 11 11	00 01 00 00	10	0	00 00 11 11
11 11 11 11	01 00 00 00	10	0	00 11 11 11
00 00 00 00	00 00 00 01	00	1	00 00 00 11
00 00 00 00	00 00 00 10	00	1	00 00 00 10
00 00 00 00	00 00 01 00	00	1	00 00 11 00
00 00 00 00	00 01 00 00	00	1	00 11 00 00
00 00 00 00	01 00 00 00	00	1	11 00 00 00
00 00 00 00	00 00 00 01	01	1	00 00 11 11
00 00 00 00	00 00 00 10	01	1	00 00 11 10
00 00 00 00	00 00 01 00	01	1	00 00 11 00
00 00 00 00	00 01 00 00	01	1	11 11 00 00
00 00 00 00	01 00 00 00	01	1	11 00 00 00
00 00 00 00	00 00 00 01	10	1	11 11 11 11
00 00 00 00	00 00 00 10	10	1	11 11 11 10
00 00 00 00	00 00 01 00	10	1	11 11 11 00
00 00 00 00	00 01 00 00	10	1	11 11 00 00
00 00 00 00	01 00 00 00	10	1	11 00 00 00
11 11 11 11	11 11 11 11	10	0	11 11 11 10
11 00 11 01	01 11 01 11	10	0	01 00 01 00
01 00 01 00	01 10 11 11	10	0	10 11 00 11
11 00 11 01	01 11 01 11	10	1	01 01 01 10
01 00 01 00	01 10 11 11	10	1	11 01 01 01

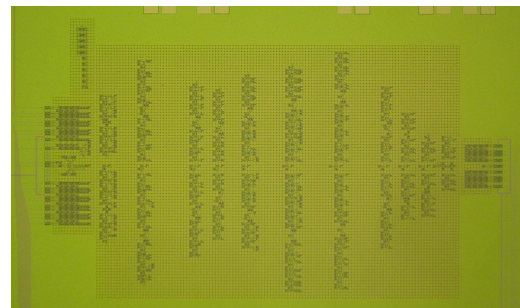


図 11: 試作チップの写真

tuoso を用いて回路面積を求めた. この結果, 回路面積は  $4.34 \text{ mm}^2$  で, うち 70% が配線, 30% が論理ゲートであった. また, 横幅は 4.80 mm, 縦幅は 2.82 mm であった.

### 6.3 試作チップ

今回設計した回路は  $1 \mu\text{m}$  9 層プロセス [14] を使用してチップ試作を行っている. チップの写真を図 11 に示す. 今後, このチップを用いて実際に動作するかを検証する予定である. 動作検証はチップを 4.2 K の液体ヘリウムに浸して行う. 測定装置で直接数十 GHz の信号を入出力することはできないため, 試作チップ中に入力バッファと出力バッファ, 高速テスト用クロックジェネレータ (CG) を備えている. 測定の手順としては, まず, 入力バッファ

に 1kHz 以下の低速で書き込み、その後 CG を動作させて 50 GHz の高速でクロックを供給する。最後に出力バッファから低速で読み出しを行い、期待した出力が得られるかを検証する。

## 7. 関連研究

SFQ 回路の高いポテンシャルが期待され、これまでに様々な回路が設計・試作されてきた。本節では、SFQ 回路の関連研究についてまとめる。

1990 年初頭に RSFQ 論理回路が Likharev らによって体系化され、2002 年に初の SFQ プロセッサである FLUX-1 [15] が開発された。その後、SFQ プロセッサの正常動作に向け、SFQ 回路の設計の難所であるタイミング設計を簡素化することを目的に CORE (COmplexity REduced) アーキテクチャが提案され、CORE1 $\alpha$  が正常動作に成功した [16,17]。ただし、動作実証に重きが置かれており、SFQ 回路の持つ高速性は性能向上のためではなく、回路の簡素化のために用いられ、ビットシリアル処理を採用している。そのため、動作周波数は 15.2 GHz と高いものの、性能は 240 MOPS と半導体プロセッサと比較しても 10 分の 1 以下と低かった。そこで、半導体プロセッサに劣らない性能を目指して CORE1 $\beta$  [6] が開発された。しかしながら、これもビットシリアル処理を採用しており、ピーク性能は 1500 MOPS と向上したものの、半導体プロセッサと同程度となっている。

CORE アーキテクチャにより SFQ プロセッサの動作実証が成功し、設計技術は向上した。しかしながら、性能が低いという問題があり、アーキテクチャが見直された。これにより、SFQ 回路を用いて高性能なプロセッサを実現するためには、CORE アーキテクチャに比べて大規模なゲートレベルパイプラインとビットパラレル処理を採用すべきであるという提案があった [3]。このアーキテクチャを採用した ALU は 56 GHz で動作し、スループットは 56 GOPS、消費電力は 1.6 mW、ジョセフソン接合数は 4,868 接合であった [8]。また、乗算器は 48 GHz で動作し、スループットは 48 GOPS 消費電力は 5.6 mW、ジョセフソン接合数は 20,251 接合であった [9]。マルチスレッディングプロセッサも 30 GHz、6.57 mW での動作が確認されており、スループットは 15 GOPS、ジョセフソン接合数は 23,713 接合であった [7]。

特定のアプリケーションに向けた回路の検討も行われている。SFQ 回路は高速動作が可能であるものの、メモリ密度が小さいという欠点がある。そのため、大きなデータメモリを必要としない特定分野用のアクセラレータとして、SHA-256 エンジンアクセラレータが提案され、半導体デバイスに対し 10 倍以上のエネルギー効率が達成できることが示されている [18]。また、SFQ 回路のパルス論理に着目し、データの遅延時間に基づいて組み合わせ最適化問題を

を解く回路であるレースロジック [19] を SFQ 回路で実現することも提案されている [20]。さらに、近年では、SFQ 回路とは異なるがジョセフソン接合を用いた論理素子である AQFP (Adiabatic Quantum-Flux-Parametron) [21] が、よりエネルギー効率の高い回路方式として研究されている。たとえば、AQFP による超低電力プロセッサの開発 [22] や、確率的コンピューティングに基づくディープニューラルネットワークの提案 [23] などがある。ニューラルネットワークアクセラレータでは SFQ 回路を用いた SuperNPU も提案されており、最新の半導体アクセラレータ [24] の 23 倍の性能が示されている [25]。

## 8. おわりに

本研究では SFQ ベクトルプロセッサの設計を行うためにベクトルプロセッサ用の SFQ コンポーネントの実現可能性を確認することを目的に、ビット幅可変加減算器の仕様策定と設計を行い、その動作検証および性能評価を行った。

設計およびポストレイアウトシミュレーションの結果、動作周波数に関しては 50 GHz と十分な性能を持っていることが実証でき、SFQ ベクトルプロセッサのコンポーネントのうち、演算器について実現可能であることが確認できた。今後、今回設計した回路の試作チップの動作検証を行う予定である。

**謝辞** 本研究を進めるにあたり、活発な議論とご協力を頂いた九州大学サイバーフィジカルコンピューティング研究室のみなさまに心より感謝の意を表します。なお、本研究は、JST 未来社会創造事業、JPMJMI18E1、ならびに、一部 JSPS 科研費 JP19H01105、JP18H05211 の助成を受けたものである。また、本研究は東京大学(旧 VDEC)を通しケイデンス株式会社の協力で行われたものである。チップ試作は、産業技術総合研究所のクリーンルーム (CRAVITY) において、AIST-ADP2 プロセスを用いて行った。

## 参考文献

- [1] Likharev, K. K. and Semenov, V. K.: RSFQ logic/memory family: A new josephson-junction technology for sub-terahertz-clock-frequency digital systems, *IEEE Transactions on Applied Superconductivity*, Vol. 1, No. 1, pp. 3–28 (1991).
- [2] Fujiwara, K., Yamashiro, Y., Yoshikawa, N., Fujimaki, A., Terai, H. and Yorozu, S.: Design and high-speed test of (4 × 8)-bit single-flux-quantum shift register files, *Superconductor Science and Technology*, Vol. 16, No. 12, pp. 14–56 (2003).
- [3] 石田浩貴, 田中雅光, 小野貴継, 井上弘士: 単一磁束量子回路向けマイクロプロセッサのアーキテクチャ探索, 情報処理学会論文誌, Vol. 58, No. 3, pp. 629–643 (2017).
- [4] デイビッド・パターソン, アンドリュー・ウォーターマン: RISC-V 原典オープンアーキテクチャのスズメ, 日経 BP 社 (2018). 成田光彰 訳.
- [5] Waterman, A.: RISC-V “V” Vector Extension (2020). <https://github.com/riscv/riscv-v-spec/>

- releases/download/0.9/riscv-v-spec-0.9.pdf.
- [6] 田中雅光: 単一磁束量子回路に基づく超高速マイクロプロセッサに関する研究, 博士論文, 名古屋大学 (2006).
- [7] 石田浩貴: 単一磁束量子回路を用いた高性能マイクロプロセッサの設計評価, 修士論文, 九州大学 (2018).
- [8] 田中雅光, 石田浩貴, 長岡一起, 村瀬健, 佐野京祐, 小野貴継, 井上弘士, 藤巻朗: 単一磁束量子回路に基づくゲートレベル・パイプライン算術論理演算器の設計とエネルギー効率評価, 研究報告システム・アーキテクチャ (ARC), Vol. 2018-ARC-232, No. 22, pp. 1-8 (2018).
- [9] Nagaoka, I., Tanaka, M., Inoue, K. and Fujimaki, A.: A 48 GHz 5.6 mW gate-level-pipelined multiplier using single-flux quantum logic, *IEEE International Solid-State Circuits Conference (ISSCC2019)* (2019).
- [10] Patterson, D. A. and Hennessy, J. L.: コンピュータの構成と設計～ハードウェアとソフトウェアのインタフェース～, Vol. 下, 日経 BP 社, 第 5 版 edition (2014). 成田光彰 訳.
- [11] ジョン・L・ヘネシー, デイビッド・A・パターソン: コンピュータアーキテクチャ定量的アプローチ, 株式会社 エスアイピー・アクセス (2019). 中條拓伯, 天野英晴, 鈴木貢 訳.
- [12] Inc., C. R.: *CRAY-1 COMPUTER SYSTEM Hardware Reference Manual 2240004* (1977).
- [13] 田中雅光, 長岡一起, 石田浩貴, 佐野京祐, 山下太郎, 小野貴継, 井上弘士, 藤巻朗: 単一磁束量子回路に基づくゲートレベル・パイプライン算術論理演算器の高エネルギー効率化と 0.3 mW, 30 GHz 動作実証, 研究報告システム・アーキテクチャ (ARC), Vol. 2019-ARC-237, No. 3, pp. 1-8 (2019).
- [14] NAGASAWA, S., HINODE, K., SATOH, T., HIDAHA, M., AKAIKE, H., FUJIMAKI, A., YOSHIKAWA, N., TAKAGI, K. and TAKAGI, N.: Nb 9-Layer Fabrication Process for Superconducting Large-Scale SFQ Circuits and Its Process Evaluation, *IEICE Transactions on Electronics*, Vol. E97.C, No. 3, pp. 132-140 (online), DOI: 10.1587/transele.E97.C.132 (2014).
- [15] Dorojevets, M.: A 20-GHz FLUX-1 superconductor RSFQ microprocessor, *Proceedings of the 5th European Workshop on Low Temperature Electronics*, pp. 157-160 (online), DOI: 10.1109/WOLTE.2002.1022472 (2002).
- [16] Tanaka, M., Matsuzaki, F., Kondo, T., Nakajima, N., Yamanashi, Y., Fujimaki, A., Hayakawa, H., Yoshikawa, N., Terai, H. and Yorozu, S.: A single-flux-quantum logic prototype microprocessor, *Tech. Dig. IEEE Int. Solid-State Circuit Conf., San Francisco* (2004).
- [17] Tanaka, M., Kondo, T., Nakajima, N., Kawamoto, T., Yamanashi, Y., Kamiya, Y., Akimoto, A., Fujimaki, A., Hayakawa, H., Yoshikawa, N., Terai, H., Hashimoto, Y. and Yorozu, S.: Demonstration of a single-flux-quantum microprocessor using passive transmission lines, *IEEE Trans. Appl. Supercond.*, Vol. 15, pp. 400-404 (2005).
- [18] Tannu, S. S., Das, P., Lewis, M. L., Krick, R., Carmean, D. M. and Qureshi, M. K.: A Case for Superconducting Accelerators, *Proceedings of the 16th ACM International Conference on Computing Frontiers, CF '19*, New York, NY, USA, Association for Computing Machinery, pp. 67-75 (online), DOI: 10.1145/3310273.3321561 (2019).
- [19] Madhavan, A., Sherwood, T. and Strukov, D.: Race Logic: A Hardware Acceleration for Dynamic Programming Algorithms, *SIGARCH Comput. Archit. News*, Vol. 42, No. 3, pp. 517-528 (online), DOI: 10.1145/2678373.2665747 (2014).
- [20] Tzimpragos, G., Vasudevan, D., Tsiskaridze, N., Micheliogiannakis, G., Madhavan, A., Volk, J., Shalf, J. and Sherwood, T.: A Computational Temporal Logic for Superconducting Accelerators, *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '20*, New York, NY, USA, Association for Computing Machinery, pp. 435-448 (online), DOI: 10.1145/3373376.3378517 (2020).
- [21] Takeuchi, N., Ozawa, D., Yamanashi, Y. and Yoshikawa, N.: An adiabatic quantum flux parametron as an ultra-low-power logic device, *Superconductor Science and Technology*, Vol. 26, No. 3, p. 035010 (online), DOI: 10.1088/0953-2048/26/3/035010 (2013).
- [22] Ayala, C. L., Tanaka, T., Saito, R., Nozoe, M., Takeuchi, N. and Yoshikawa, N.: MANA: A Monolithic Adiabatic iNtegration Architecture Microprocessor Using 1.4-zJ/op Unshunted Superconductor Josephson Junction Devices, *IEEE Journal of Solid-State Circuits*, Vol. 56, No. 4, pp. 1152-1165 (online), DOI: 10.1109/JSSC.2020.3041338 (2021).
- [23] Cai, R., Ren, A., Chen, O., Liu, N., Ding, C., Qian, X., Han, J., Luo, W., Yoshikawa, N. and Wang, Y.: A Stochastic-Computing Based Deep Learning Framework Using Adiabatic Quantum-Flux-Parametron Superconducting Technology, *Proceedings of the 46th International Symposium on Computer Architecture, ISCA '19*, New York, NY, USA, Association for Computing Machinery, pp. 567-578 (online), DOI: 10.1145/3307650.3322270 (2019).
- [24] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Killebrew, D., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snellman, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E. and Yoon, D. H.: In-datacenter performance analysis of a tensor processing unit, *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pp. 1-12 (online), DOI: 10.1145/3079856.3080246 (2017).
- [25] Ishida, K., Byun, I., Nagaoka, I., Fukumitsu, K., Tanaka, M., Kawakami, S., Tanimoto, T., Ono, T., Kim, J. and Inoue, K.: SuperNPU: An Extremely Fast Neural Processing Unit Using Superconducting Logic Devices, *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 58-72 (online), DOI: 10.1109/MICRO50266.2020.00018 (2020).