Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

超伝導ニューラルネットワーク・アクセラレータのアーキテクチャ探索を目的とした電力性能モデリング

石田 浩貴[1,a)]　Ilkwon Byun[2]　長岡 一起[3]　福光 孝介[1]　田中 雅光[3]　川上 哲志[1]　谷本 輝夫[1]
小野 貴継[1]　藤巻 朗[3]　Jangwoo Kim[2]　井上 弘士[1]

概要：本稿では，超伝導単一磁束量子 (SFQ: Single-Flux-Quantum) 回路を用いたニューラルネットワーク・アクセラレータの電力性能モデリングを実施する．まず，SFQ 回路の特性を踏まえニューラルネットワーク・アクセラレータの基本アーキテクチャを定義する．そして，基本アーキテクチャに基づき，アクセラレータの動作周波数，消費電力，面積モデルを構築する．その後，実チップの測定結果やポストレイアウトシミュレーションによるモデルの精度検証を行い，その有効性を明らかにする．さらに，開発したモデルを用いてアーキテクチャ探索，ならびに，最適化を実施し，従来の CMOS によるアクセラレータとの性能を比較することで，改良したアーキテクチャの有効性を議論する．

1. はじめに

ディープニューラルネットワーク（Deep Neural Network: DNN）は、画像認識 [27] や音声認識 [2]，言語翻訳 [30] などの分野に応用され，それぞれにおいて高い認識精度を達成している [8], [16], [29]．これまで，DNN をより高速かつ低消費電力に実行するために，さまざまな専用ハードウェアによるアクセラレーションが提案されてきた [3], [4], [5], [6], [14]．今後も DNN の重要性が増し，低レイテンシ，高スループット，かつ，高電力効率な DNN 実行が求められることが予想される．

一方，現在では半導体技術による計算機システムの劇的な改善は見込めない．たとえば，計算機の頭脳に当たるプロセッサは，2000 年初頭より消費電力問題が露呈し，動作周波数改善による性能向上は困難となった．その後，一つのチップに複数のコアを搭載するマルチコアの導入によって性能向上が維持されたが，これらの根幹技術である半導体の微細化が限界を迎えつつあり，さらなる性能向上は期待できない．

半導体の直面している消費電力問題を解決し，高い動作周波数を狙えるデバイスとして，ジョセフソン接合を用いた超伝導単一磁束量子（SFQ: Single Flux Quantum）回路がある [19]．SFQ 回路は，超伝導リング内に量子化される磁束を情報担体とし，その磁束がジョセフソン接合を通過する際に発生する微弱な電圧パルスの相互作用で論理演算を実現する．ジョセフソン接合のスイッチングに要するエネルギーは，10^{-19} J と非常に小さく，10^{-12} 秒での高速なスイッチングが可能である [19], [20]．また，超伝導伝送路による光速と同程度でのビット情報伝播といった高速性も有する．

これらの特徴から，これまでに様々な SFQ 回路の設計・試作が行われ，数十 GHz という非常に高い周波数での動作実証に成功している [13], [21], [22], [31]．特に，8 bit 乗算器は 48 GHz [21]，8 bit 算術論理演算器（Arithmetic Logic Unit: ALU）は 56 GHz [31] での正常動作が確認されており，DNN の処理の大部分を占める積和演算を効率よく実行できる可能性がある．しかしながら，これらは要素回路レベルの性能であり，これらを組み合わせた SFQ 向けニューラルネットワーク・アクセラレータ（Neural Network Accelerator: NNA）のアーキテクチャやその性能は明らかになっていない．SFQ 回路は従来の CMOS 回路とは動作原理や回路特性が異なり，従来と同じアーキテクチャを採用しても SFQ 回路のポテンシャルを十分に活かすことができない可能性がある．したがって，SFQ 向けの NNA アーキテクチャ探索による適切なトレードオフ設計が必要であると考えられる．

そこで，SFQ 回路の高いポテンシャルを最大限活用可能な NNA のアーキテクチャ実現に向け，SFQ NNA の電

[1] 九州大学
Kyusyu University, Fukuoka-shi, 819–0395, Japan
[2] ソウル大学
Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, Korea
[3] 名古屋大学
Nagoya University, Nagoya-shi, 464–8603, Japan
[a)] koki.ishida@cpc.ait.kyushu-u.ac.jp

Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

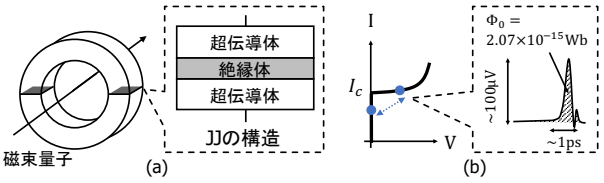

図 1 (a) 超伝導体のリングと JJ 構造 (b) JJ の電気的特性

力性能モデリング，ならびに，アーキテクチャ最適化を実施する．すでに，モデルを用いたボトルネック解析や解消によって，SFQ 向け NNA アーキテクチャは提案しており [12]，本稿では，SFQ NNA の電力性能モデリングの詳細，ならびに，その精度検証について説明する．まず，SFQ 回路のアーキテクチャ的特性に基づき，SFQ NNA の基本アーキテクチャを設計する．その後，SFQ NNA の動作周波数，消費電力，ならびに，面積モデリングを行い，試作したプロトタイプチップの実測データやポストレイアウトシミュレーションによって，その推定精度を検証する．精度検証の結果，アクセラレータを構成するユニットにおける周波数，消費電力，面積の推定誤差はそれぞれ，5.6%，1.2%，1.3%，アクセラレータ全体の推定誤差はそれぞれ，4.7%，2.3%，9.5%での推定が可能であることが明らかになった．さらにモデルを用いて最適化した SFQ 向け NNA アーキテクチャは，6 つの畳み込みニューラルネットワーク（Convolutional Neural Network: CNN）ワークロードにおいて，従来の CMOS によるアクセラレータに比べて平均で 23 倍の性能向上に成功し，最大 522 Tera Multiply-Acumulate operations Per Second （TMAC/s）の性能を達成した．

本稿の構成は以下の通りである．第 2 節では，SFQ 回路の動作原理，ならびにその現状と課題について述べ，第 3 節で NNA の基本アーキテクチャを設計する．そして，第 4 節では，アクセラレータの周波数，消費電力，面積モデリング，ならびにその精度を検証する．その後，第 5 節で開発したモデルを用いてアーキテクチャ探索を行い，SFQ 向け NNA アーキテクチャを評価する．最後に第 6 節でまとめる．

2. 単一磁束量子回路

本節では，SFQ 回路の動作原理，SFQ 回路の特徴の一つであるゲートレベルパイプラン構造，ならびに，SFQ 回路の現状と課題について説明する．

2.1 動作原理

SFQ 回路は，超伝導体でできたリング内に量子化される磁束量子の相互作用を利用して演算を実現する論理回路である [19]．従来の CMOS 回路では電圧レベルがバイナリ信号に対応するが，SFQ 回路ではリング内の磁束量子の有無で ‘1’，‘0’ を表現する．SFQ 回路の基本素子である超伝導体のリング（図 1(a)）は，2 つの超伝導体間に薄い障壁層を挟み弱結合させたデバイスであるジョセフソン接合（JJ: Josephson Junction）を含み，磁束量子の出入りを可能としている．

ジョセフソン接合は図 1(b) のような電気的特性を持ち，磁束量子が通過する（リングに出入りする）際に接合はスイッチし，インパルス状の電圧（以降，SFQ パルスと呼ぶ）が発生する．SFQ 回路では，幅が数ピコオーダの SFQ パルスで情報伝播に用いる．そのため，論理ゲートを構成する際にこれらのパルスを直接作用させることは困難である．そこで，SFQ 回路の論理ゲートは一旦パルスを磁束量子として保持するための超伝導リングを持つ．すなわち，SFQ 回路の論理ゲートはラッチ機能を有する．

SFQ 論理ゲートに到着するパルスを論理値 ‘1’ と扱う場合，論理値 ‘0’ の状態とパルス未到着の状態を区別できない．そのため，SFQ 論理ゲートは参照用のパルス信号（以降，クロックパルスと呼ぶ）と呼ばれる SFQ パルスを用いて論理値を判別する．具体的には，論理ゲートにクロックパルスが入力されるまでに磁束が保持されていれば ‘1’，保持されていなければ ‘0’ と判別する．

2.2 ゲートレベルパイプライン

前述の通り，SFQ 回路で構成される論理ゲートは超伝導体のリング（図 1(a)）から構成され，ラッチ機能を有している．この論理ゲートのラッチ機能を活用するアーキテクチャとしてゲートレベルの深いパイプライン構造が提案されている [32]．SFQ 回路では，従来の CMOS 回路のようにパイプラインレジスタ追加のオーバヘッドはない．また，動的消費電力は SFQ パルスに起因しており，JJ がスイッチングに要するエネルギーは 10^{-19} J と小さいため，電力による発熱も問題にならない．さらに，CMOS 回路のように配線の充放電を必要としないため，スイッチング速度も 10^{-12} s と高速であり，数十〜百 GHz の動作周波数を追求可能である．

これらの特徴から，ゲートレベルパイプライン構造を採用した SFQ 回路の設計・試作が行われ，乗算器や ALU といった要素回路において 50 GHz 程度での高速動作実証に成功している [21], [22], [31]．しかしながら，このような深いパイプライン構造では，パイプラインストールによって性能が著しく低下する恐れがある．これに対し，パイプライン段数分のスレッドを用意し，サイクル毎に実行スレッドを切り替える細粒度マルチスレッディングを採用した 4bit プロセッサの設計が行われ，32 GHz での動作実証に成功し，16 Tera operations per second（TOPs）の性能を達成した [13]．これらの結果より，ゲートレベルパイプライン構造は，高周波数動作だけでなく，パイプライン利用率を高く維持することで高い性能を見込めることが明らかになっており，SFQ 回路の高性能化に有効的な回路構成法で

あるといえる．したがって，本研究で対象とする NNA はゲートレベルパイプライン構造を採用することとする．

2.3 SFQ 向け NNA の実現に向けた課題と研究目的

前述の通り，これまでに SFQ 回路の特性を考慮した回路の最適化が実施され，SFQ 回路の高いポテンシャルが明らかになっている．特に，8 ビット乗算器や ALU はそれぞれ，48 GHz，56 GHz での動作実証に成功しており [21], [31]，DNN の大部分を占める積和演算の処理性能が高く，NNA で高い性能を達成する可能性が高い．しかしながら，これらはそれぞれ要素回路レベルの性能であり，これらを組み合わせた SFQ 向け NNA のアーキテクチャやその性能は明らかになっていない．SFQ 回路は従来の CMOS 回路とは動作原理や回路特性が異なり，従来のアーキテクチャを模倣するだけでは SFQ 回路のポテンシャルを十分に活かすことができない可能性があるため，SFQ 向けの NNA アーキテクチャ探索による適切なトレードオフ設計が必要であると考えられる．

そこで本稿では，SFQ 回路の高いポテンシャルを最大限活用可能な NNA のアーキテクチャを探索すべく，SFQNNA の電力性能モデリングを実施する．具体的には，まず，SFQ 回路のアーキテクチャ的特性に基づき，SFQ 向け NNA の基本アーキテクチャを設計する．その後，SFQNNA の動作周波数，消費電力，ならびに，回路面積モデリングを行い，アーキテクチャの総合評価環境を構築する．開発したモデルは試作したプロトタイプチップの実測データやポストレイアウトシミュレーションにより，推定精度を検証する．そして，SFQ 向け NNA アーキテクチャを探索し，従来の CMOS によるアクセラレータとの性能を比較することで，その有効性を評価する．

3. SFQ NNA の基本アーキテクチャ設計

本節では，SFQ NNA の基本アーキテクチャを決定すべく，NNA の基本要素であるオンチップネットワーク，Processing Element (PE)，オンチップバッファを設計する．図 2(a) に基本アーキテクチャの全体像を示す．SFQ 回路の性質や制約を考慮し，オンチップバッファ，SFQ 向けオンチップネットワーク，PE 構造として，それぞれシフトレジスタ型オンチップバッファ（図 2(a)①），シストリックネットワーク（図 2(b)②），重み保持型 PE（図 2(c)③）を採用した．

3.1 シフトレジスタ型オンチップバッファ（①）

SFQ 回路を用いたメモリの実装に関しては，これまでに幾つかの提案が行われてきたが，最も実用的なのはシフトレジスタ構造を基本とする FIFO メモリである．SFQ パルスは信号の分岐にはスプリッタと呼ばれる専用の配線を必要とするためファンアウトを増やしづらい．また，数ピコ秒の幅の SFQ パルスを用いて，従来の MOS ベースのランダムアクセスメモリのようにビットラインやワードラインを駆動してタイミングを合わせるのは容易ではない．これに対し，シフトレジスタ型メモリは，素子となる Delay flip flops (DFFs) を超伝導リングで効率よく実装可能であり，100 GHz での高速動作実証に成功している [9]．さらに，DNN のメモリアクセスパターンは静的に決まり，ランダムアクセスを必要としないためシフトレジスタ型メモリとの相性は良い．これらの理由より，シフトレジスタ型メモリを用いてオンチップバッファを実装する．また，畳み込みニューラルネットワーク（Convolutional Neural Network: CNN）を行列積に変換するためのデータアライメントユニット（Data Alignment Unit: DAU）も設計した．

3.2 シストリック型ネットワーク（②）

NNA で用いいられる代表的な二種類のネットワーク構造として，1）マルチキャスト型，2）シストリック型があげられる．マルチキャスト型はバスやツリーを用いて複数の PE にデータを同時に供給するネットワーク構造である．一方，シストリック型は PE から一方向に隣接する PE へ，データをサイクル毎に伝播するネットワーク構造である．

SFQ 回路では従来の CMOS 回路で用いられるバスは存在せず，マルチキャスト型の実装にはスプリッタと呼ばれるパルス信号を分岐する専用の配線をツリー状にしたスプリッタ・ツリーが用いられる．一つのスプリッタで一度に分岐できるパルス信号は 2〜3 とファンアウトが少ないため，SFQ 回路においてスプリッタ・ツリーは効率よく実装できない．一方，シストリック型ではデータは隣接する PE にのみデータを分岐すればよく要求ファンアウトは低いため SFQ 回路に向いているネットワーク構造といえる．加えて，ネットワーク実装の際の配線長が PE 数に依存せず，隣接する PE 間の距離で決まるため，PE 数が多い場合でも高い動作周波数を達成できる可能性が高い．これらの理由より，SFQ 向けオンチップ・ネットワークとしてシストリック型を採用する．

3.3 重み保持型 PE（③）

シストリックネットワークでマッピング可能なデータフローは主に Weight Stationary (WS)，Output Stationary (OS)，Input Stationary (IS) がある [4], [25]．データフローによって PE 構造が異なり，図 2(c)(1) に WS に PE 構造，(2) に OS の PE 構造を示す．WS と IS は PE 内部のレジスタで保持するデータが重みか入力特徴マップかの違いだけであり，ハードウェア構造は同じであるため，ここでは WS 型と OS 型の 2 種類について検討する．

図 2(c) に示す通り，WS の PE にはフィードバックループがなく，OS の PE にはフィードバックループが存在する．SFQ 回路では，従来の CMOS とは異なり，フローク

Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

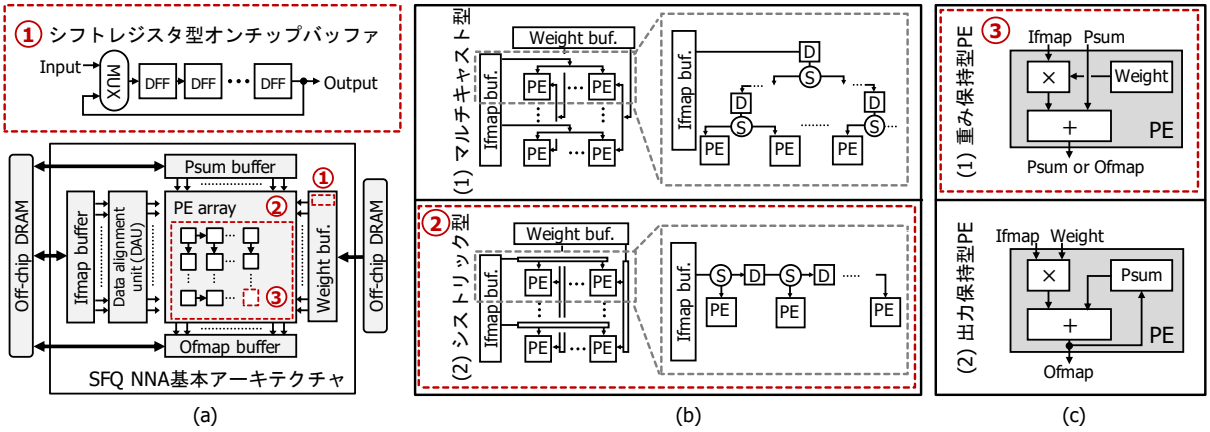

図 2　SFQ NNA 基本アーキテクチャと各ユニットの構造．(a) 基本アーキテクチャの全体像．
(b) 二種類のオンチップネットワーク．(1) マルチキャスト型と，(b) シストリック型．
(c) 二種類の PE 構造．(1) 重み保持型 PE と，(2) 出力保持型 PE.

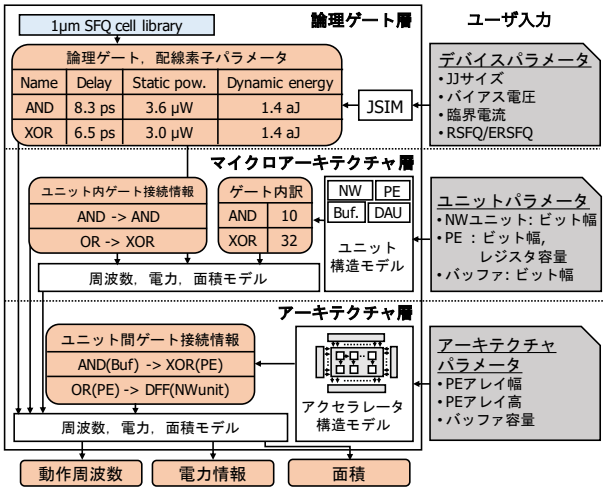

図 3　SFQ NNA の動作周波数・消費電力・面積モデルの全体像

ロッキングと呼ばれる，クロックがデータ同様に回路内を伝播するクロッキング手法が一般的に用いられる．このフロークロッキングでは，クロックとデータを同方向に伝播する場合，データの伝播遅延をクロックの伝播遅延で相殺することが可能であり，結果として高い周波数を狙いやすい．一方で，フィードバックループが回路内部に存在する場合，データとクロックを同方向に伝播できないため，フィードバックループを含まない回路に比べ周波数が低くなるという特徴がある．そこで，基本アーキテクチャではより高い周波数を達成すべく，フィードバックループを含まない WS の PE 構造を採用する．

4. SFQ NNA の電力性能モデリング

SFQ NNA のアーキテクチャ探索をすべく，動作周波数，消費電力，および，面積についてモデリングを行う．本稿では，より高い推定精度を達成すべく，図 3 に示すように，論理ゲートレベル，マイクロアーキテクチャレベル，アーキテクチャレベルの三階層での抽象化を行う．以降の各小節にて，各層の役割，および，実装について詳しく説明し，その後，モデルの検証を行う．

4.1 論理ゲート層

論理ゲート層は，タイミングモデル，電力モデル，面積モデルの三つから構成され，デバイスパラメータ（JJ のサイズ，バイアス電圧，JJ の臨界電流）や対象 SFQ 技術（Rapid SFQ: RSFQ, Energy-efficient: ERSFQ）を入力とし，各論理ゲートや配線素子のタイミング情報（セットアップタイム，ホールドタイム，レイテンシ），電力情報（静的消費電力，動的消費エネルギー），面積を見積もる．各モデルは通常の SFQ 技術である RSFQ 技術に加え，低消費電力化手法が施された ERSFQ 技術もサポートしている．

4.1.1 タイミングモデル

タイミングモデルは，各論理ゲートや配線素子のセットアップタイム，ホールドタイム，レイテンシを出力する．RSFQ 技術による論理ゲートや配線素子のタイミング情報は，実際の回路設計で用いられる 1.0μm SFQ スタンダードセル [23] の構造を基に，JJ 向け SPICE シミュレータである JSIM [7] によって抽出した．タイミングモデルはこれらの値はあらかじめテーブルとして保持しており，入力に応じて適切な値を出力する．

ERSFQ 技術による論理ゲートや配線素子のタイミング情報は，ERSFQ ゲートに関する詳細な情報が不足しているため，RSFQ 技術のタイミング情報，ならびに，ERSFQ 技術の報告されている特性に基づき推定する．RSFQ 技術と ERSFQ 技術の違いはバイアス電流の供給方法であり，ゲートの構造自体に違いがないため，タイミング情報は変わらないとの報告がある [17], [20]．そこで，本タイミングモデルでは，RSFQ 回路と ERSFQ 回路の論理ゲートおよび配線素子のタイミング情報は同じであると仮定する．

SFQ 回路では，1.0〜0.2μm までの範囲において，JJ の

Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

一辺が $1/\alpha$ にスケールすると，セットアップタイム，ホールドタイム，レイテンシも $1/\alpha$ 倍になるという報告がされている [15]．そこで本モデルは，1.0〜0.2μm までの範囲において，対象 JJ サイズに応じてスケールした値を出力するよう実装した．

4.1.2 電力モデル

電力モデルは，各論理ゲートや配線素子の静的消費電力，動的消費エネルギーを出力する．RSFQ 技術による論理ゲートや配線素子では，抵抗を用いて JJ にバイアス電流を供給するため，論理ゲートや配線素子の JJ 数に応じて静的消費電力が発生する．したがって，論理ゲートや配線素子の静的消費電力 $StP_{gate,wire}$ は以下の式 (1) で求められる．

$$StP_{gate,wire} = V_{bias} I_{bias} \times \#JJs_{gate,wire} \quad (1)$$

ただし，V_{bias} はバイアス電圧，I_{bias} はバイアス電流，$\#JJs_{gate,wire}$ は論理ゲートや配線素子に用いたれる JJ 数である．これに対し，論理ゲートや配線素子が活性化（内部の JJ がスイッチ）する際に動的消費エネルギーが発生する．これは論理ゲートの入力によってスイッチする JJ 数が異なるため，本モデルでは，その平均値を論理ゲートの動的消費エネルギーとして算出する．例えば AND ゲートの場合，とりうる入力は 4 パターン（00，01，10，11）であり，各入力パターンにおいてスイッチする JJ 数の平均を，1JJ あたりの消費エネルギーとかけあわせることで，動的消費エネルギーを求める．論理ゲートや配線素子の動的消費エネルギー $DyE_{gate,wire}$ は以下の式 (2) で求められる．

$$DyE_{gate,wire} = I_c \Phi_0 \times \#ActJJs_{gate,wire} \quad (2)$$

ただし，I_c は JJ の臨界電流値，Φ_0 は磁束量子（$2.07mV{\cdot}ps$），$\#ActJJs_{gate,wire}$ は論理ゲートや配線素子のスイッチする平均 JJ 数である．

一方，ERSFQ 回路では代わりに JJ を用いてバイアスを供給する．これにより，ERSFQ 回路では回路の JJ がスイッチするタイミングでバイアス供給用の JJ も同時にスイッチしることでバイアス供給を実現しているため，静的消費電力は発生しない．しかしながら一度のスイッチングに必要な電力（すなわち動的消費電力）は，通常の RSFQ 回路の約二倍であると報告されている [17]．よって本モデルでは，ERSFQ 回路における静的消費電力はゼロ，動的消費エネルギーは RSFQ 回路の二倍の値とした．

4.1.3 面積モデル

面積モデルは，論理ゲートや配線素子の面積を出力する．前述の通り，RSFQ 技術と ERSFQ 技術の違いはバイアス電流の供給方法であり，ゲートの構造自体に違いがないため，RSFQ と ERSFQ 技術の両方において，1.0μm SFQ スタンダードセルのサイズを参考にした．また，JJ の一辺が $1/\alpha$ にスケールすると，セルサイズも $1/\alpha$ 倍になると仮定した．

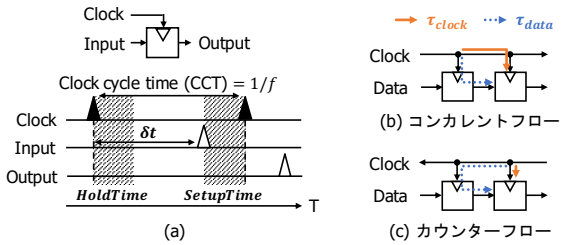

図 4 周波数算出例と二種類のクロッキング手法 (a) DFF における周波数算出．(b) コンカレントフロー・クロッキング．(c) カウンターフロー・クロッキング．

4.2 マイクロアーキテクチャ層

マイクロアーキテクチャ層は，マイクロアーキテクチャ・ユニットの構造モデル，動作周波数モデル，電力モデル，面積モデルから構成され，マイクロアーキテクチャ・ユニットパラメータや倫理ゲート層の出力結果を入力とし，各マイクロアーキテクチャ・ユニットの動作周波数，電力情報（静的消費電力，アクセスあたりの動的エネルギー），面積を見積もる．

4.2.1 ユニット構造モデル

ユニット構造モデルは，アクセラレータを構成する各ユニット（PE，バッファ，オンチップネットワーク，DAU）のパラメータ（ビット幅）に基づき，ユニット内のゲート接続情報，および，ゲート内訳情報を生成する．本モデリングで対象とする NNA はゲートレベルパイプラインを採用しており，より正確な動作周波数を算出するためには，ユニット内の全てのゲートペア（出発元ゲートと到着先ゲートの組み合わせ）において周波数を調査する必要がある．そのため，ユニット構造モデルは，各ユニットの全てのゲートペアおよびその接続間情報（ゲート間の配線の種類や必要な配線素子数）を生成し，周波数モデルに受け渡す．必要な配線素子数は過去の設計事例 [21], [22] より，一般的な SFQ 論理ゲート幅二つ分の距離を配線するために必要な素子数とした．またゲート内訳情報は，ユニットを構成する論理ゲートや配線素子数からなり，電力モデルおよび面積モデルに受け渡される．

4.2.2 周波数モデル

周波数モデルは，ユニット構造モデルで生成されたゲート接続情報，ならびに，論理ゲート層の出力結果をに基づき，各ユニットの動作周波数を見積もる．具体的には，ユニット内のすべてのゲートペアの動作周波数を算出し，その最小値をユニットの動作周波数として出力する．ゲートペアの動作周波数 f は図 4 および式 (3) で求められる．

$$f = 1/\mathrm{CCT} = 1/((SetupTime + M) + \delta t) \quad (3)$$

$SetupTime$ は到着先ゲートのセットアップタイム，M はタイミングマージンであり，M は過去の設計事例 [21], [22]

より JJ 二つ分の遅延とした．また，δt はクロックが入力されてからデータが入力されるまでの時間差であり，クロック線から出発元ゲートにクロックが入力され，到着先ゲートにデータが到達するまでの時間 τ_{data}，および，クロック線から到着先ゲートにクロックが入力されるまでの時間 τ_{clock} の差（$\tau_{data} - \tau_{clock}$）で表される．ただし，到着先ゲートにて正しくデータが認識されるためには，δt は到着先ゲートのホールドタイム $HoldTime$ より大きく（すなわち，クロックに対して $HoldTime$ 分データが後に到着）なければならない．

本モデルは SFQ 回路における代表的な二つのクロッキング手法（コンカレントフロー・クロッキング（図 4 (b)），カウンターフロー・クロッキング（図 4 (c)））をサポートしており，対象ユニットの構造に応じて適切なクロッキング手法を選択する．コンカレントフロー・クロッキングは，クロックをデータと同方向に伝搬する手法であり，ゲート間のデータ伝播遅延をクロック伝播遅延で相殺することで高い周波数を達成しやすい．しかしながら，フィードバックループが存在する場合，ループ部分はクロックとデータの伝播方向が逆となりデータ伝播遅延を相殺できないため，ループの距離に応じて周波数が低くなる．これに対し，クロックをデータと逆方向に伝搬するカウンターフロー・クロッキングでは，フィードバックループ部分はデータ伝播遅延をクロック伝播遅延で相殺することができるため，ループの距離に関わらず周波数を一定に保つことが可能である．しかしながら，フィードフォワード部分ではクロックとデータの伝播方向が逆であるため，フィードバックループを含まないコンカレントフローに比べ周波数は低くなる．つまり，フィードバックループがない場合はコンカレントフロー，ある場合はカウンターフロー・クロッキングによってより高い周波数を達成することが可能であるため，本モデルでは，ユニット内のフィードバックループの有無に応じて適切なクロッキング手法を選択し，動作周波数を算出する．

4.2.3 電力・面積モデル

マイクロアーキテクチャ層の電力，および，面積モデルは，ユニット構造モデルで生成されたゲート内訳情報と論理ゲート層の出力結果をに基づき，各ユニットの静的消費電力，アクセスあたりの動的消費エネルギー，回路面積を見積もる．各ユニットの静的消費電力 StP_{unit}，および，アクセスあたりの動的消費エネルギー DyE_{unit} は以下の式 (4),(5) で求められる．

$$StP_{unit} = \sum (StP_{gate,wire} \times \#gate(\text{or } \#wire)) \quad (4)$$

$$DyE_{unit} = \sum (DyE_{gate,wire} \times \#gate(\text{or } \#wire)) \quad (5)$$

ただし，$\#gate$,，$\#wire$ はそれぞれ対象ユニットに含まれる，ある種類の論理ゲート数，配線素子数である．

また，各ユニットの面積 A_{unit} は以下の式 (6) で求められる．

$$A_{unit} = \sum (A_{gate,wire} \times \#gate(\text{or } \#wire)) \quad (6)$$

ただし，$A_{gate,wire}$ はユニットに含まれる論理ゲートや配線素子の面積である．

4.3 アーキテクチャ層

アーキテクチャ層は，アクセラレータ全体の構造モデル，動作周波数モデル，電力モデル，面積モデルから構成され，アーキテクチャパラメータ，マイクロアーキテクチャ層および倫理ゲート層の出力結果に基づき，アクセラレータの動作周波数，電力情報（静的消費電力，アクセスあたりの動的エネルギー），面積を見積もる．アクセラレータを構成するユニット数に基づいてマイクロアーキテクチャ層の結果を統合するだけでなく，ユニット間の接続も考慮し最終推定結果を出力する．

4.3.1 アクセラレータ構造モデル

アクセラレータ構造モデルは，アーキテクチャパラメータ（PE 数，バッファ容量等）を入力とし，ユニット間のゲート接続情報を生成する．ユニット間のゲート接続情報は，各ユニットのインターフェースゲート，ユニット間の距離，配線の種類の情報を保持しており，ユニット間接続を考慮した周波数推定，および，ユニット接続に必要な配線素子の電力情報や面積の見積もりに用いられる．ユニット間の距離に関しては，シストリック構造のため最大で PE 幅分であり，PE 幅はマイクロアーキテクチャ層で算出された PE 面積の平方根とした．

4.3.2 動作周波数モデル

アーキテクチャ層の動作周波数モデルは，まず，ユニット間のゲート接続情報に基づいてユニット間のゲートペアにおける周波数を算出する．そして，マイクロアーキテクチャ層の結果とユニット間の周波数のうち，最小値をアクセラレータの周波数 f_{acc} として出力する（式 (7)）．

$$f_{acc} = min(f_{PE}, f_{buf}, f_{NW}, f_{DAU}, f_{inter-unit}) \quad (7)$$

ただし，f_{PE}，f_{buf}，f_{NW}，f_{DAU} は各ユニットの周波数，$f_{inter-unit}$ はユニット間における最小周波数である．

4.3.3 電力・面積モデル

電力・面積モデルは，アーキテクチャパラメータを基にマイクロアーキテクチャ層の結果を統合する．そして，アクセラレータ構造モデルで生成されたユニット間のゲート接続情報から，ユニット間接続に必要な配線素子数を算出し，必要素子数分の静的消費電力，アクセスあたりの動的エネルギー，面積を統合結果に加え，最終結果として出力する．アクセラレータの静的消費電力 StP_{acc}，およ

Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

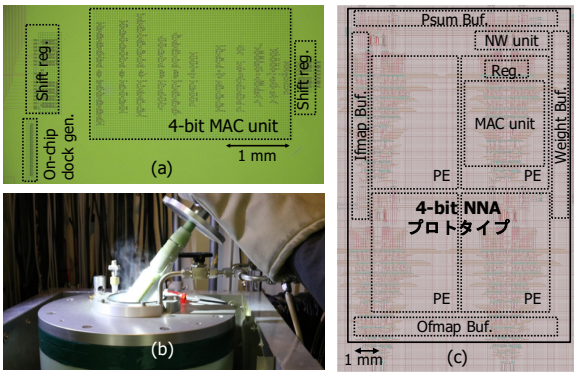

図 5 モデル検証のセットアップ．(a) 試作した 4 ビット MAC ユニット．(b) チップの測定環境．液体ヘリウムデュアーを用いた．(c) 4 ビット 2 × 2PE からなる NNA プロトタイプのレイアウト．

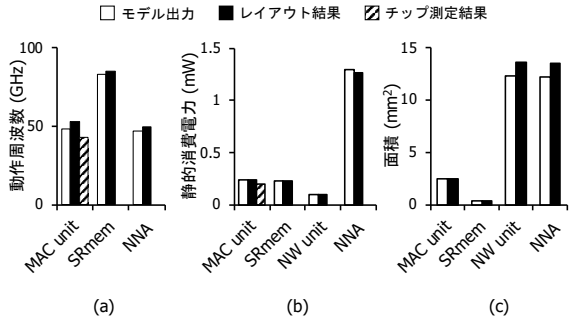

図 6 各モデルの検証結果．(a) 動作周波数モデル．(b) 静的消費電力モデル．(c) 回路面積モデル．

び，アクセスあたりの動的消費エネルギー DyE_{acc} は以下の式 (8),(9) で求められる．

$$StP_{acc} = \sum(StP_{unit} \times \#unit) + StP_{inter-unit} \quad (8)$$

$$DyE_{acc} = \sum(DyE_{unit} \times \#unit) + DyE_{inter-unit} \quad (9)$$

ただし，$\#unit$ はそれぞれ対象アクセラレータに含まれるユニット数，$StP_{inter-unit}$，および，$StP_{inter-unit}$ はそれぞれユニット間接続部分の静的消費電力，アクセスあたりの動的エネルギーである．

一方，アクセラレータの面積 A_{acc} は以下の式 (10) で求められる．

$$A_{acc} = \sum(A_{unit} \times \#unit) + A_{inter-unit} \quad (10)$$

ただし，$A_{inter-unit}$ はユニット間接続部分の面積である．

4.4 検証

本小節では，マイクロアーキテクチャレベル，アーキテクチャレベルの二つの層の，動作周波数，静的消費電力，および，面積の推定精度を，実チップやポストレイアウトシミュレーションにより検証する．検証に用いるチップやその測定環境，ならびにレイアウト設計した NNA プロトタイプを図 5 に示す．論理ゲート層は実チップの製造に用いたれるセルライブラリの情報に基づいており，すでに高い精度が検証されているため，検証の対象外とした．また，本稿では，SFQ 回路全体の 9 割以上を占める静的消費電力のみの検証を実施しているが，ERSFQ 回路技術を考慮する場合，動的消費エネルギーの重要性が増すため，今後の課題である．

4.4.1 マイクロアーキテクチャ層の検証

まずマイクロアーキテクチャ層の出力結果を，試作した 4 ビット MAC ユニット（図 5 (a)）の測定結果（MAC unit），8 ビット 8 エントリのシフトレジスタ型メモリ（SRmem），8 ビットのオンチップネットワークユニット（NW unit）のポストレイアウトシミュレーション結果との比較を行う．4 ビット MAC unit は PE のレジスタを除く演算回路部分であり，チップは図 5 (b) に示した 4 K 環境での測定結果に加え，レイアウト後のデータも検証に用いた．図 6 に検証結果を示す．NW unit は単一の DFF と配線素子のみから構成されており，ユニット内では周波数は決定しないため，周波数を除いた消費電力，面積結果が示されている．各ユニットの周波数，電力，面積の平均推定誤差はそれぞれ 5.6%，1.2%，1.3%であり，全体的に正確に見積もられていることがわかる．MAC unit の周波数の推定誤差は，チップとレイアウトそれぞれにおいて，-9.3%，11.6%と比較的大きい．ただし，実チップとレイアウトデータは同一の MAC unit であり，チップとレイアウトデータ間で 23%の誤差があるため，モデルの推定誤差は製造ばらつきが大きいことが考えられる．一方，レイアウトデータの周波数はモデル出力より高い値となっているが，これは式 (3) のマージン M に起因すると考える．本モデルではマージンを含んだ周波数を出力するが，ポストレイアウトシミュレーションではマージンがゼロ（すなわちゲート，配線遅延とタイミング制約のみ）で周波数が算出されるため，レイアウトデータの周波数がモデルに対し高くなっていると予想される．

これに対し，消費電力，面積はほとんど誤差なく推定できていることがわかる．SFQ 回路の静的消費電力は式 (1) のとおり，JJ 数に依存するため，面積の検証結果と傾向が同じである．特に SRmem，NW unit の誤差が小さいのは，回路構造が比較的単純であり，構成する論理ゲート数だけでなく配線素子数まで正確に見積もられているためと予想される．

4.4.2 アーキテクチャ層の検証

次にアーキテクチャ層の出力結果を，4 ビット 2 × 2PE からなるアクセラレータ（図 5 (c)）のポストレイアウトシミュレーション結果（NNA）と比較する．比較対象は PE 数が四つと比較的小規模であるが，2D シストリックネットワークにおいては隣接するユニット間の接続が重要であ

Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

り，PE 数に依存しないため，モデルの検証には十分であると考える．図 6 に示すとおり，NNA の周波数，電力，面積の平均推定誤差は，それぞれ 4.7%，2.3%，-9.5%であった．周波数に関しては，モデル，レイアウトの両方においてクリティカルパス（ネットワーク部分）が一致していることから，マイクロアーキテクチャ層同様にマージンによる誤差と予想される．

一方，NNA の消費電力と面積に関しては，マイクロアーキテクチャ層の結果と異なる傾向が見受けられる．電力に関しては，マイクロアーキテクチャ層の結果と同じ傾向であることから，PE の電力推定誤差がそのまま結果に反映されていると考えられる．これに対し，面積に関しては，ポストレイアウトシミュレーション結果がモデル出力に比べ，約 10%大きくなっている．消費電力の場合，モデル出力に対してポストレイアウトシミュレーション結果が低いことから，JJ を含まない回路の影響が大きいことが予想される．基本的に SFQ 回路はほぼ全ての回路が JJ から構成されるが，受動伝送線路（PTL:Passive Transmission Line）の配線素子は JJ を含まない．実際の設計では，よりばらつき耐性を向上させるべく，等長配線を実施するが，アクセラレータ・プロトタイプではマイクロアーキテクチャユニットに比べ配線が複雑であり，PTL 配線素子による等長配線のオーバヘッドが大きいことが検証結果の原因であると考える．

5. SFQ 向け NNA アーキテクチャ探索

本節では，第 3 節で設計した基本アーキテクチャを基にアーキテクチャ探索を行い，SFQ 向け高性能 NNA アーキテクチャを評価する．具体的には，まず，第 4 節で開発した SFQ 回路の周波数モデルに加え，自作のサイクルベースのシミュレータを用いてアーキテクチャ探索，ならびに，ボトルネック解析を行う．そして，アーキテクチャの改善を実施し，従来の CMOS による NNA と性能比較を行うことで，その有効性を評価する．詳細は [12] を参照されたい．

5.1 シミュレーション環境

アーキテクチャを評価するためには，開発したモデルの出力に加え，アプリケーション実行時のサイクル数を求める必要がある．そこで，NN の積和演算数から実行にかかるサイクル数を見積もる，サイクルベースのシミュレータを開発した．図 7 にシミュレータの概要を示す．本シミュレータは，入力としてアーキテクチャ記述ファイル（PE 数，バッファサイズ，PE アレイの周波数等），NN 記述ファイル（入力特徴マップサイズ，重みサイズ，重み数等），ならびに，オフチップメモリのバンド幅を入力とし，重みのマッピングごとに，セットアップ（バッファとオフチップメモリのデータのロードストア，重みのマッピング）に要

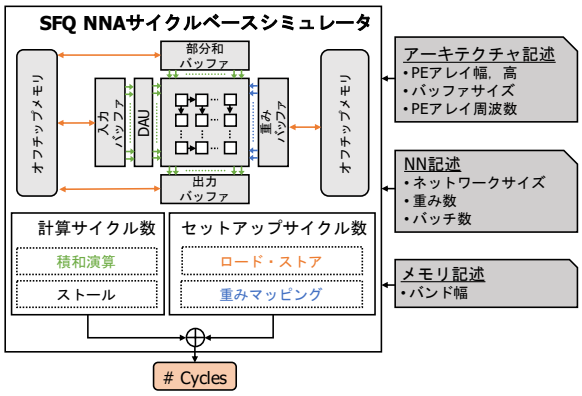

図 7 SFQ NNA サイクルベースシミュレータ

するサイクル数と計算（積和演算，パイプラインストール）に要するサイクル数を見積もる．

5.2 基本アーキテクチャのセットアップ

NNA のアーキテクチャ設計空間は膨大で，そのすべてを探索することは困難である．そこで，基本アーキテクチャのボトルネックを解析し，アーキテクチャを改善することで，高性能な SFQ 向き NNA 実現を目指す．基本アーキテクチャの初期のアーキテクチャパラメータは TPU を参考にして設定した．これは，基本アーキテクチャは TPU と似たアーキテクチャ（シストリックアレイで WS データフロー）を採用しているためである．また，SFQ 回路は冷凍機を用いて 4K まで冷却する必要があるため，エッジデバイスではなくサーバサイドでの運用に適している．TPU もサーバサイドで用いられる NNA であり，さらに，SFQ 回路の JJ が TPU で使用されているテクノロジサイズ（28 nm）までスケールすると仮定した際，同等の面積で実現できる可能性があるため，初期パラメータとして採用することは妥当であると言える．表 1 に基本アーキテクチャの初期パラメータを示す．基本アーキテクチャの周波数は，第 4 節で開発した周波数モデルによって 52 GHz と見積もられた．また，オフチップメモリのバンド幅は TPUv2 で用いられている HBM の値を参考に 300 GB/s とした [1].

5.3 ボトルネック解析

5.3.1 オンチップバッファ内部のデータ移動

シミュレーションの結果，オンチップバッファ内部のデータ移動のオーバヘッドが非常に大きいことが明らかになった．図 8(a) に実行サイクル数の内訳を示す．図 8(a) に示した通り，全てのワークロードにおいてセットアップに要するサイクル数が 9 割以上を占めていることがわかる．これは，オンチップバッファはシフトレジスタ型メモリからなり，計算できる状態にするためにはデータがシフトレジスタの先頭に位置するまでデータをシフトする必要があるためである．図 8(b) にデータ移動の例を示す．ま

Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

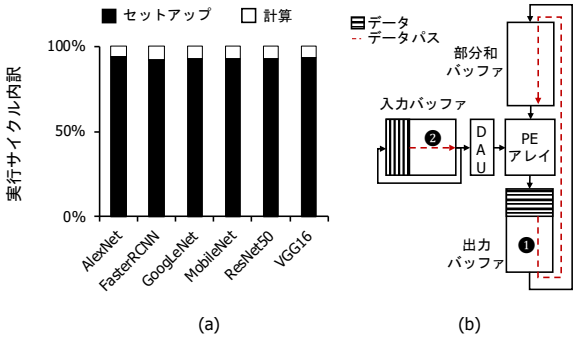

図 8　(a) 実行サイクル内訳，(b) バッファ内のデータ移動例

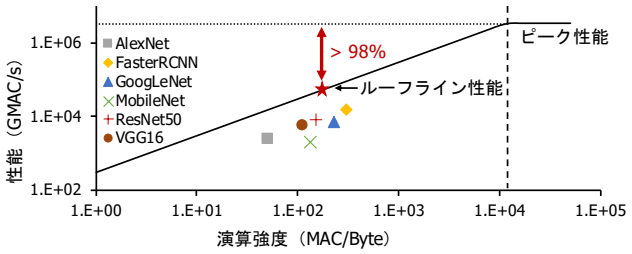

図 9　基本アーキテクチャのルーフラインプロット

ず，出力バッファに格納された部分和の部分和バッファへの移動がある（図 8(b)❶）．この場合，基本アーキテクチャは出力バッファと部分和バッファのシフトレジスタの合計の長さ分データをシフトさせる必要があり，そのサイクル数は 65,536（16 MB÷256 B/cycle）サイクルにもなる．これは，TPU と同じ 8bit 整数の積和演算をサポートしており，各 PE 列が毎サイクル 8bit（=1 B）データを入力として受け取るため，PE アレイの要求バンド幅は 256 B/cycle となるためである．入力特徴マップバッファのデータ再利用時（循環時）にも同様の問題が生じる（図 8(b)❷）．したがって，シフトレジスタの長さを削減し，データ移動のオーバヘッドを緩和する必要がある．

5.3.2　PE の低利用率

シミュレーションの結果，基本アーキテクチャの PE の利用率が低く，実行性能は平均でピーク性能の 2%以下であることが明らかになった．図 9 に，基本アーキテクチャのルーフラインプロットを示す．横軸は計算強度（MAC/Byte），縦軸は性能（GMAC/s）を表す．各ワークロードのプロットはバッチ数が 1 の際の計算強度，性能を示す．図 9 にプロットされいてる星はワークロードの平均の計算強度において達成可能な性能（ルーフライン性能と呼ぶ）であり，ピーク性能の 2%以下となっている．これは，NNA の演算速度（52 GHz）とメモリバンド幅（300 GB/s）のギャップが大きく，性能がメモリバンド幅に律速されているためである．したがって，計算強度を増加させて PE 利用率を上昇させる必要がある．

5.3.3　オンチップバッファの低利用率

計算強度を増加させる手段として，NN のバッチ数の増加があるが，シミュレーションの結果，オンチップのバッファの利用率が低く有効利用できていないため，バッチ数増加による計算強度改善が見込めないことが明らかになった．これは，バッチ数を増やしても，オンチップバッファの容量も同時に増やさない限り，オフチップメモリアクセスが増加してしまうためである．図 10 にバッファを有効活用できていない三つの例を示す．部分和，または，出力特徴マップのデータ量に対して出力バッファの容量が多い場合でも，重みマッピングごとに出力バッファ内のデータを部分和バッファ，あるいは，オフチップメモリに移動させる必要があるため，一度の重みマッピングで使われない容量が無駄になってしまう（図 10(a)）．また，マップされる重みが PE アレイに対して少ない場合，マップされていない PE 列の出力バッファはまったく利用されない（図 10(b)）．入力バッファは，各シフトレジスタは入力特徴マップの各チャネルのユニークなデータを保持するが，そのデータ量がシフトレジスタの容量（長さ）に対して少ない場合，残りの入力バッファ容量は同様に利用されない（図 10(c)）．これらバッファの低利用率は，それぞれ出力バッファのシフトレジスタの長さ，出力バッファのシフトレジスタ数，入力バッファのシフトレジスタの長さが原因であり，改善する必要がある．

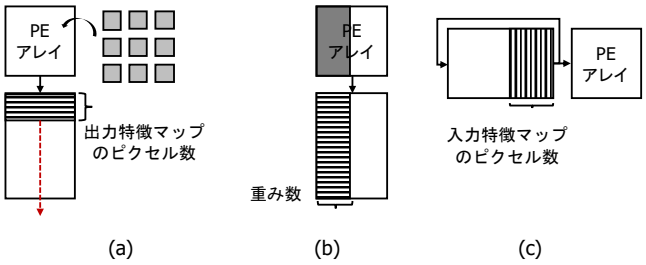

図 10　バッファが有効活用されない例．(a) 出力特徴マップのピクセル数に対して出力バッファが大きい場合，(b) 重み数が少なくすべての PE に重みがマップされない場合，(c) 入力特徴マップのピクセル数に対して入力バッファが大きい場合

5.4　SFQ NNA アーキテクチャの最適化

本節では，第 5.3 節で明らかになったボトルネックに基づき，アーキテクチャ最適化を実施する．図 11 に最適化後の SFQ NNA アーキテクチャの全体像を示す．まず，それぞれオンチップバッファをサブアレイ化し，シフトレジスタの長さを削減した．また，部分和バッファと出力バッファを一つのバッファに統合し，部分和バッファから出力バッファへのデータ移動をなくした．次に，PE アレイの幅を基本アーキテクチャの 1/4 にし，その分オンチップバッファの容量を増加させた．そして，PE あたりのレジスタ数を 1 から 8 に増やし，同時にマッピングできる重みの数を増やすことで PE の利用率の向上を計った．以下の小節において，それぞれのアーキテクチャ最適化の詳細を

Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

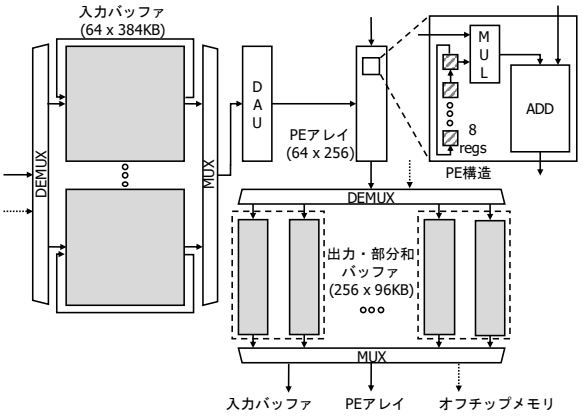

図 11　最適化後アーキテクチャの全体像

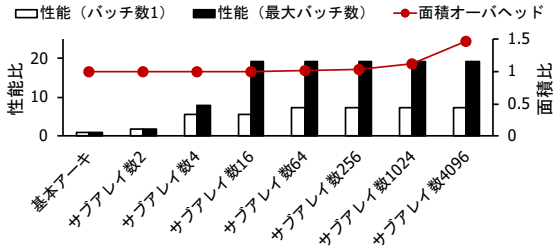

図 12　バッファのサブアレイ数と性能および面積の関係

示す．その後，最適化後のアーキテクチャの有効性を評価すべく，従来の CMOS による NNA との比較を行う．

5.4.1　バッファ・アーキテクチャ最適化

性能向上には，まず，オンチップバッファ内部のデータ移動を削減する必要がある．データ移動は主に 1）シフトレジスタの長さ，2）独立した部分和バッファと出力バッファによるものであるため，1）シフトレジスタのサブアレイ化，2）部分和バッファと出力バッファの統合を行った．シフトレジスタをサブアレイ化する場合，アクセス先を選択するために追加のマルチプレクサ，デマルチプレクサが必要となる．図 12 にサブアレイ化の効果とそのオーバヘッドの関係を示す．縦軸はそれぞれ基本アーキテクチャの性能，面積で正規化されている．サブアレイ化によってバッチ数 1，バッチ数最大（オフチップメモリアクセスが発生しない範囲で増やせる最大のバッチ数）の両方における性能が向上していることがわかる．ここでのサブアレイ数 2 は部分和バッファと出力バッファを統合し，それぞれをサブアレイと見立てた場合と同じである．バッチ数最大における性能向上がより大きいのは，バッファをサブアレイ化することでバッファの利用率も同時に向上しており，結果最大バッチ数も増加しているためである．サブアレイ数が 64 の場合，基本アーキテクチャに比べ性能は 6.3 倍になっており，それ以降は大幅な性能向上見られないため，サブアレイ数を 64 とした．

5.4.2　ハードウェア資源の最適化

第 5.3 節で述べた通り，PE の利用率は低く，ピーク性能

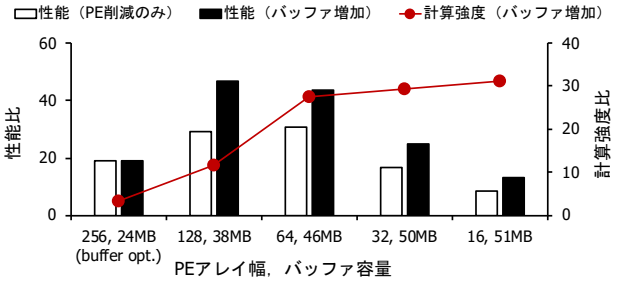

図 13　PE 削減数と性能および計算強度の関係

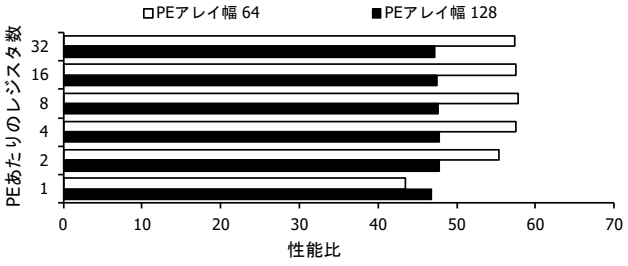

図 14　レジスタ数増加による性能向上

の 2%以下の性能しか達成できていない．そこで，PE を削減して，その分バッファの容量を増やすことで計算強度の向上，PE 利用率の向上を図る．PE 数の削減方針として，PE アレイの高さでなく，幅を削減することとした．PE アレイの幅を削減することで，同時にオンチップバッファの低利用率の例 2（図 10(b)）の改善を試みることができるためである．図 13 に PE 削減した際の性能，ならびに，計算強度向上を示す．縦軸はそれぞれ基本アーキテクチャの性能，計算強度で正規化されている．この性能は無限のオプチップバンド幅を想定した結果である．PE を削減するだけでも，バッファの利用率改善から性能向上を見込めることがわかる．そして，削減した PE 分バッファ容量を増加することで最大バッチ数が上昇し，PE アレイの幅を 1/2（128 PE），1/4（64 PE）にした際にそれぞれ基本アーキテクチャの 47，42 倍の性能を達成している．性能は PE アレイ幅が 128 の方が高いものの，64 より高い計算強度を達成している．次の小節において，バンド幅を考慮した場合のそれぞれの比較を行い，PE 数を決定する．

5.4.3　PE あたりのレジスタ数増加

さらに PE の利用率を向上させるべく，PE 内のレジスタ数を増加させる．PE のレジスタ数を増加させると，同時にマップできる重みの数が増加し，一つの入力データに対して複数の異なる積和演算を実行可能である．結果，マッピングごとの PE あたりの演算数が増加し，深い PE アレイのパイプラインを処理で埋めることができ，PE 利用率向上につながる．図 14 に PE あたりのレジスタ数増加の性能への影響を示す．横軸は基本アーキテクチャの性能で正規化されている．図 14 より，より演算強度の高い PE アレイ幅が 64 の構成がより高い性能を達成していることが

Vol.2021-ARC-244 No.15
Vol.2021-SLDM-194 No.15
Vol.2021-EMB-56 No.15
2021/3/25

表 1　評価対象のアーキテクチャ・パラメータ

	TPU	SFQ NNA 基本アーキテクチャ	最適化後
PE array width	256	256	64
PE array height	256	256	256
Ifmap buf.	24 MB	8 MB	24 MB
Ofmap buf.		8 MB	24 MB
Psum buf.		8 MB	
Weight buf.		64 KB	128 KB
# regs in PE	1	1	8
Frequency (GHz)	0.7	52.6	52.6
Peak perf. (TMAC/s)	45	3366	842
Area (mm^2) (28nm)	<330	~283	~299

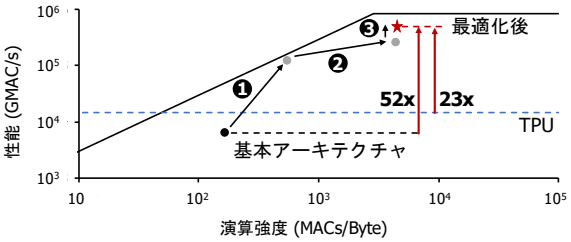

図 15　最適化後アーキテクチャの性能評価

わかる．また，レジスタ数 8 以降は性能向上が見込めなくなるため，PE アレイ幅が 64，PE あたりのレジスタ数が 8 の場合を最適化アーキテクチャのパラメータとした．

5.4.4　性能評価

本小節では，最適化後のアーキテクチャの有効性を評価すべく，基本アーキテクチャ，および，TPU との性能比較を行った．表 1 に評価対象のアーキテクチャ・パラメータを示す．基本アーキテクチャ，ならびに，最適化後のアーキテクチャの性能推定には，現在に SFQ 回路の試作で使用される 1.0μm プロセスを用いた．一方，TPU の性能推定には，シストリックアレイ型 DNN アクセラレータ向けサイクル精度シミュレータである ScaleSIM [25] を用いた．性能比較には，AlexNet [18]，FasterRCNN [24]，GoogLeNet [28]，MobileNet [11]，ResNet50 [10]，VGG16 [26] の 6 つの CNN ワークロードを用いた．

評価結果を図 15 に示す．バッファ・アーキテクチャ最適化（図 15❶），ハードウェア資源の最適化（図 15❷），ならびに，PE 数増加によるパイプライン利用率の向上（図 15❸）によって，基本アーキテクチャ，および，TPU に対して，それぞれ平均で 52 倍，23 倍の性能向上に成功した．これらの結果より，現在試作に使用される 1.0μm プロセスにおいても大幅な性能向上の可能性が明らかになった．加えて，SFQ 回路では微細化技術によるデバイス性能の向上が見込めるため [15]，今後プロセス技術の発展によってさらなる性能向上が期待できる．

6.　おわりに

本稿では，SFQ 回路の高速な積和演算処理性能に着目し，SFQ 向け NNA アーキテクチャを探索すべく，NNA の電力性能モデルを開発した．まず，SFQ 回路の特性を考慮し，SFQNNA の基本アーキテクチャを設計した．その後，論理ゲート層，マイクロアーキテクチャ層，アーキテクチャ層の三階層からなる，NNA の動作周波数，消費電力，面積モデルを構築し，実チップの測定結果やポストレイアウトシミュレーションによる精度検証を実施した．その結果，NNA の動作周波数，消費電力，面積における推定誤差はそれぞれ，4.7%， 2.3%，9.5%と，高い精度での推定を確認した．また，開発したモデルを用いたアーキテクチャ探索，および，最適化によって，従来の CMOS 型 NNA 対し，平均で 23 倍の性能向上に成功し，最大で 522 TMAC/s を達成した．

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井上研究室の皆様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，一部文部科学省科学研究費補助金 JP19H01105，JP18H05211，JP18J21274 の支援を受けたものである．

参考文献

[1] Alcorn, P.: Hot Chips 2017: A Closer Look At Google's TPU v2, `https://www.tomshardware.com/news/tpu-v2-google-machine-learning,35370.html`.

[2] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L. V., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J. and Zhu, Z.: Deep Speech 2: End-to-End Speech Recognition in English and Mandarin, *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, JMLR.org, p. 173–182 (2016).

[3] Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y. and Temam, O.: DianNao: A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning, *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '14, New York, NY, USA, Association for Computing Machinery, p. 269–284 (online), DOI: 10.1145/2541940.2541967 (2014).

[4] Chen, Y., Emer, J. and Sze, V.: Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks, *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pp. 367–379 (2016).

[5] Chen, Y., Luo, T., Liu, S., Zhang, S., He, L., Wang, J., Li, L., Chen, T., Xu, Z., Sun, N. and Temam, O.: DaDianNao: A Machine-Learning Supercomputer, *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 609–622 (2014).

[6] Du, Z., Fasthuber, R., Chen, T., Ienne, P., Luo, T., Feng, X., Chen, Y. and Temam, O.: ShiDianNao: shifting vision processing closer to the sensor, (online), DOI: 10.1145/2749469.2750389 (2015).

[7] Fang, E. and Duzer, T. V.: A Josephson integrated circuit simulator (JSIM) for superconductive electronics application, *Extended Abstracts of 1989 International Superconductivity Electronics Conference*, pp. 407–410 (online), available from ⟨https://ci.nii.ac.jp/naid/10008998489/⟩ (1989).

[8] Farabet, C., Couprie, C., Najman, L. and LeCun, Y.: Learning Hierarchical Features for Scene Labeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp. 1915–1929 (online), DOI: 10.1109/TPAMI.2012.231 (2013).

[9] 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, p. 1456 (2003).

[10] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (2016).

[11] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, *arXiv e-prints*, p. arXiv:1704.04861 (2017).

[12] 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).

[13] Ishida, K., Tanaka, M., Nagaoka, I., Ono, T., Kawakami, S., Tanimoto, T., Fujimaki, A. and Inoue, K.: 32 GHz 6.5 mW Gate-Level-Pipelined 4-Bit Processor using Superconductor Single-Flux-Quantum Logic, *2020 IEEE Symposium on VLSI Circuits*, pp. 1–2 (2020).

[14] 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.-l., 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., Snelham, 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, *Proceedings of the 44th Annual International Symposium on Computer Architecture*, New York, NY, USA, Association for Computing Machinery, p. 1–12 (online), DOI: 10.1145/3079856.3080246 (2017).

[15] Kadin, A. M., Mancini, C. A., Feldman, M. J. and Brock, D. K.: Can RSFQ logic circuits be scaled to deep submicron junctions?, *Applied Superconductivity, IEEE Transactions on*, Vol. 11, No. 1, pp. 1050–1055 (2001).

[16] Karpathy, A. and Fei-Fei, L.: Deep Visual-Semantic Alignments for Generating Image Descriptions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 4, pp. 664–676 (online), DOI: 10.1109/TPAMI.2016.2598339 (2017).

[17] Kirichenko, D. E., Sarwana, S. and Kirichenko, A. F.: Zero Static Power Dissipation Biasing of RSFQ Circuits, *IEEE Transactions on Applied Superconductivity*, Vol. 21, No. 3, pp. 776–779 (online), DOI: 10.1109/TASC.2010.2098432 (2011).

[18] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, Red Hook, NY, USA, Curran Associates Inc., p. 1097–1105 (2012).

[19] 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 (online), DOI: 10.1109/77.80745 (1991).

[20] Mukhanov, O. A.: Energy-Efficient Single Flux Quantum Technology, *IEEE Transactions on Applied Superconductivity*, Vol. 21, No. 3, pp. 760–769 (2011).

[21] Nagaoka, I., Tanaka, M., Inoue, K. and Fujimaki, A.: A 48GHz 5.6mW Gate-Level-Pipelined Multiplier Using Single-Flux Quantum Logic, *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, pp. 460–462 (2019).

[22] Nagaoka, I., Tanaka, M., Sano, K., Yamashita, T., Fujimaki, A. and Inoue, K.: Demonstration of an Energy-Efficient, Gate-Level-Pipelined 100 TOPS/W Arithmetic Logic Unit Based on Low-Voltage Rapid Single-Flux-Quantum Logic, *2019 IEEE International Superconductive Electronics Conference (ISEC)*, pp. 1–3 (2019).

[23] Nagasawa, S., Hinode, K., Satoh, T., Hidaka, 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).

[24] Ren, S., He, K., Girshick, R. and Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *arXiv e-prints*, p. arXiv:1506.01497 (2015).

[25] Samajdar, A., Zhu, Y., Whatmough, P., Mattina, M. and Krishna, T.: SCALE-Sim: Systolic CNN Accelerator Simulator, *arXiv e-prints*, p. arXiv:1811.02883

(2018).
[26] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv 1409.1556* (2014).
[27] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *International Conference on Learning Representations* (2015).
[28] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: Going Deeper with Convolutions, *arXiv e-prints*, p. arXiv:1409.4842 (2014).
[29] Toshev, A. and Szegedy, C.: DeepPose: Human Pose Estimation via Deep Neural Networks, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660 (online), DOI: 10.1109/CVPR.2014.214 (2014).
[30] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. and Dean, J.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation (2016).
[31] 田中雅光，石田浩貴，長岡一起，村瀬　健，佐野京佑，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，技術報告 22，名古屋大学, 九州大学, 名古屋大学, 名古屋大学, 名古屋大学, 九州大学, 九州大学, 名古屋大学 (2018).
[32] 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（オンライン），入手先 ⟨https://ci.nii.ac.jp/naid/170000148470/⟩ (2017).