

# 単一磁束量子回路プロセッサの性能評価と マイクロアーキテクチャの再検討

津秦 伴紀<sup>1,a)</sup> 田中 雅光<sup>2</sup> 井上 弘士<sup>3</sup>

## 概要：

本稿では、単一磁束量子 (SFQ: Single-Flux-Quantum) 回路を用いた超高性能マイクロプロセッサの実現に向けたアーキテクチャ検討を行う。これまでに、25 GHz を越える回路動作速度を実現した SFQ マイクロプロセッサの設計事例が報告されている。しかしながら、従来の設計では 1 ビットずつ演算を行うビットシリアル処理を前提とした 7 段命令パイプライン構成を採用しており、高い回路動作周波数を達成しているものの、マイクロプロセッサ全体としての性能向上効果は明かになっていなかった。また、現在の CMOS を基本とするマイクロプロセッサに対する優位性、さらには、SFQ マイクロプロセッサ・アーキテクチャとしてのあるべき姿に関する議論は殆ど行われていない。そこで本研究では、性能限界を見極める第一次近似として、性能モデリングに基づく SFQ マイクロプロセッサの性能評価を実施した。その結果、1) 従来のビットシリアル処理方式 7 段命令パイプライン構成では現在の CMOS マイクロプロセッサを凌駕する性能とはならない可能性がある、2) 0.3 $\mu$ m プロセスを用いたビットパラレル方式 7 段命令パイプライン構成で 9 GHz 程度の動作周波数を見込める、3) SFQ 回路の特性を考慮したビットパラレル方式細粒度パイプライン構成を採ることによって動作周波数は 87GHz となり、99%以上のパイプラインストールを隠蔽できれば、CMOS マイクロプロセッサに対して 30 倍以上の性能向上を実現できる可能性があることが分かった。なお、本評価ではメモリアクセス・レイテンシの影響や、安定動作を保證する十分な設計マージンの確保は想定しておらず、より精度の高い評価は今後の課題である。

## 1. はじめに

近年の CMOS マイクロプロセッサでは動作周波数が頭打ちとなっている。その理由は、動作周波数の向上による消費電力の増加が問題となるためである。今後この消費電力の壁を打ち破り、超高速動作が可能なプロセッサを実現することは必要不可欠となってくる。そこで我々は単一磁束量子 (SFQ: Single-Flux-Quantum) 回路に着目している。SFQ 回路は、超伝導ループ中の磁束消失時に発生する微弱電圧パルスを情報担体として扱う集積回路である。SFQ 回路は動作原理上、電荷の充放電を必要としないため、論理ゲート当たりの消費電力は CMOS 回路と比較して約 1/1,000 以下という超低消費電力性を持つ。また、超伝導状態により抵抗成分が限りなくゼロに近くなるため、高速なパルス伝搬が可能という特性も有する。

このような特徴に着目し、これまでに SFQ マイクロプロセッサの試作・研究が行われてきた。実際に試作された

マイクロプロセッサである CORE-1 $\beta$  では、25GHz での動作実証に成功している [5][6]。これらの設計事例では、回路面積の縮小やタイミング設計の容易化、そして回路動作の高速化を優先し、ビットシリアル方式を採用している。ビットシリアル方式とは、データ語長 (マイクロプロセッサが扱うことができるデータのビット幅) を 1 ビット単位で処理 (例えば、32 ビット語長の場合には 1 ビット処理を 32 回繰り返す) する方法である。

しかしながら、ビットシリアル方式では、データ語長に比例して処理回数が増加し、計算結果を得るまでのレイテンシが増加する。通常、マイクロプロセッサの性能は、動作周波数のみならず命令実行レイテンシにも大きく依存する。したがって、性能の観点では回路の動作周波数の向上に重きを置いたビットシリアル方式は必ずしも適切であるとは限らない。実際、ビットシリアル方式を採用した CORE-1 $\beta$  の命令パイプラインでは、SFQ 論理回路の動作に必要な駆動パルス信号とパイプライン・ステージ間の同期を制御するクロック信号の 2 つの信号が存在する。先に述べた 25 GHz での動作は駆動パルス信号、すなわち、組合せ回路の動作速度のことである。しかしながら、命令パ

<sup>1</sup> 九州大学大学院システム情報科学府

<sup>2</sup> 名古屋大学大学院工学研究科

<sup>3</sup> 九州大学大学院システム情報科学研究科

a) tsuhata@soc.ait.kyushu-u.ac.jp

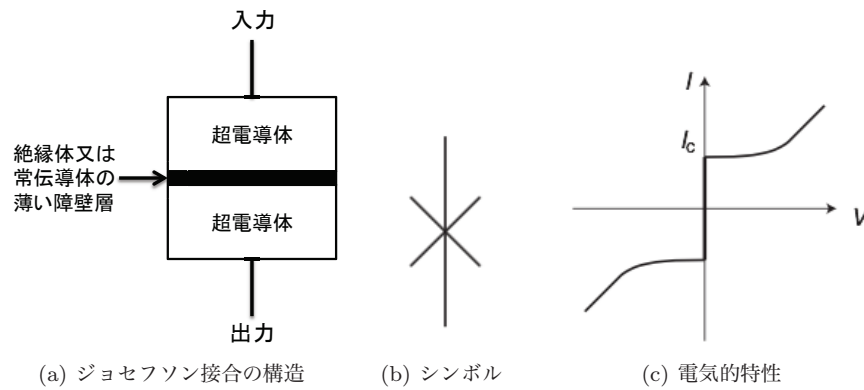


図1 ジョセフソン接合の構造と電気的特性

イプラインを動作させるクロック信号の周波数は高々 1.5 GHz と低い。

以上の事を踏まえ、我々はビットシリアル方式に加え、現在の CMOS マイクロプロセッサが採用しているビットパラレル方式の有効性を調査するため、CORE-1β の命令パイプライン構造に基づき最大動作周波数のモデル化と評価を行った [7]。その結果、データ語長が 64 ビットの時、ビットパラレル方式を用いた場合に最も高いクロック信号動作周波数である 9.27 GHz を達成することが分かった。しかしながら、最新の CMOS マイクロプロセッサは 5GHz 程度の動作周波数を実現しており、これに比べると約 1.85 倍ほどでしかない。文献 [7] で作成したモデルは SFQ マイクロプロセッサのみを対象としているため、CMOS マイクロプロセッサとの比較ができていない。

そこで本稿では、以下の 2 つを研究目的とし、SFQ マイクロプロセッサの性能評価を行う。

- より高速な動作周波数を達成するマイクロアーキテクチャの探索
- CMOS マイクロプロセッサとの性能比較

本評価では、文献 [1] の性能モデルを用いる。SFQ マイクロプロセッサと CMOS マイクロプロセッサの性能評価を行う。このモデルは、マイクロアーキテクチャに依存するため、SFQ マイクロプロセッサと CMOS マイクロプロセッサの比較が可能となる。また、評価結果を用い、今後 SFQ マイクロプロセッサが採るべきアーキテクチャについて議論する。

本稿の構成は以下の通りである。第 2 節では SFQ 回路の動作原理と特徴について述べる。第 3 節では SFQ マイクロプロセッサのアーキテクチャ設計空間について整理する。第 4 節では性能モデルに基づく性能評価と評価結果について述べる。第 5 節で今後 SFQ マイクロプロセッサが採るべきアーキテクチャについて議論し、最後に第 6 節でまとめる。

## 2. SFQ 回路の動作原理

本節では、SFQ 回路の動作原理について述べる。第 2.1 節で SFQ 回路の基本素子となるジョセフソン接合を紹介し、第 2.2 節にてスイッチング動作の原理を説明する。第 2.3 節ではデジタル情報の表現に用いられるパルス理論について述べ、第 2.4 節で 2 入力 1 出力の SFQ AND ゲートの動作例を示す。

### 2.1 ジョセフソン接合

特定の金属や化合物をある一定の温度以下まで冷やすと、電気抵抗が 0 になりマイスナー効果が現れる。この現象を超伝導という。また、超伝導状態になりうる物質を超伝導物質、超伝導状態にある超伝導物質を超伝導体と呼ぶ。SFQ 回路は、超伝導物質であるニオブ (Nb) とアルミ化合物 ( $AlO_x$ ) を用いたジョセフソン接合というデバイスによって構成される。

ジョセフソン接合とは、2 つの超伝導体の間に薄い障壁膜を挟み、弱結合させたデバイスである。図 1(a) にジョセフソン接合の構造を示す。回路図中では図 1(b) に示すシンボルで表され、図 1(c) に示すような電気的特性を持つ。 $I_c$  は、臨界電流値と呼ばれ、ジョセフソン接合を流れる電流が  $I_c$  を下回っていると、ジョセフソン接合間には電圧が発生せず、超伝導体間を直流電流が流れる。いわゆるトンネル現象である。 $I_c$  を上回ると、ジョセフソン接合間に電圧が発生し、常伝導状態 (超伝導を示さない状態) となって交流電流が流れる。この電圧が発生する状態になることを、スイッチする、あるいは、接合が切れる、という。

### 2.2 スwitching 動作の原理

SFQ 回路は、図 2 に示すようなジョセフソン接合を含む超伝導体ループによって構成される。超伝導体で作られたループ内を貫く磁束は、 $\Phi_0 = h/2e = 2.07 \times 10^{-15} \text{ Wb}$  の整数倍に量子化される ( $h$  はプランク定数、 $e$  は電子の電荷)。SFQ 回路では、この磁束量子  $\Phi_0$  を情報担体として用いて

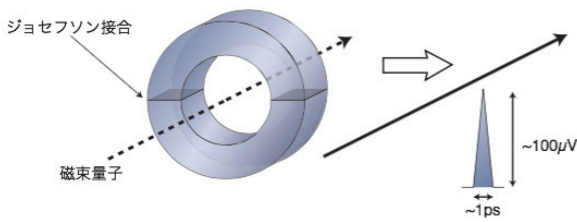


図2 ジョセフソン接合を含む超伝導ループ

いる。具体的には、超伝導ループに磁束がある場合と、磁束がない場合をそれぞれ‘1’、‘0’のビット情報を保持しているとみなす。例として、図2のループ内に量子磁束が1つ入った場合を考える。ループ内に磁束が通ると、周回電流が流れる。この周回電流  $I_L$  は、ループ一周のインダクタンスを  $L$  とすると、 $I_L = \Phi \approx 0/L$  で表される。周回電流値が  $I_c$  を下回る場合は、リングに挿入されているジョセフソン接合には電圧が発生せず、周回電流は手前から見て時計回りにリングを流れ続け、量子磁束はループ内に留まる。つまり、‘1’というビット情報を保持しているとみなすことができる。一方、 $I_c$  を上回った場合、ジョセフソン接合には電圧が発生し、常伝導状態となる。これにより、ループは磁束量子を保持できなくなるため、ループ内の磁束量子はループ外へ出る。この時、ループ内には量子磁束が無くなるので、‘0’というビット情報を保持しているとみなすことができる。このように、ジョセフソン接合をスイッチさせるか、させないかで量子磁束の保持、伝搬が可能になる。また、ジョセフソン接合がスイッチすると、接合間を磁束量子が通過できるようになる。磁束量子がループ外へ飛び出す時、ファラデーの電磁誘導の法則から接合の両端にインパルス状の電圧 (SFQ パルスと呼ぶ) が発生する [9]。SFQ パルスは、高さが数百  $\mu\text{V}$ 、幅が数ピコ秒となる。このパルスは、後に述べるパルス理論において、‘0’、‘1’のビット情報を判別するために用いられている。

### 2.3 パルス理論

SFQ 回路では、パルスの有無で ‘0’、‘1’ を表すパルス理論を用いて情報を表現する。CMOS 回路で用いられているレベル理論との決定的な違いは、パルス自体では状態を表現できない点である。このため、SFQ 回路では ‘0’ という情報を扱うための工夫が必要となる。‘1’ という情報をパルスの到着で表すとすると、‘0’ とパルスが到着していない状態との区別が出来ない。その為、データパルス信号線とは別に駆動パルス信号線を用意し、一定間隔で到着する駆動パルス信号の一周期内にデータパルス信号線上的パルスが到着すれば ‘1’、到着しなければ ‘0’ とする。図3に SFQ 論理ゲートにおける情報の判別方法について示す。駆動パルス信号は一定間隔で各 SFQ 論理ゲートに到着し、動作させる。この時、データパルス信号が到着していれば情報 ‘1’、到着していなければ情報 ‘0’ として扱う。また、こ

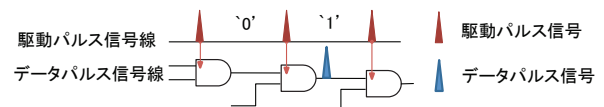


図3 データパルス信号線と駆動パルス信号線

の図からも分かるように、この回路の動作周波数は駆動パルス信号の到着間隔、すなわち、駆動パルス信号の周期によって決まる。

### 2.4 組み合わせ回路の動作例

SFQ 回路を用いた組み合わせ回路の例として AND ゲートを取り上げる。動作を図4に従って説明する。

- (1) まず、2 入力 of データパルスが入力線 a, b にそれぞれ到達する。
- (2) 到着したデータパルスによって  $J_9 - L_{s1} - J_1$  のループと  $J_{10} - L_{s2} - J_2$  のループに電流が流れ、それぞれ磁束が保持される。
- (3) そこへ、駆動パルス信号が分岐して同時に各ループに入ることで  $J_1, J_2$  がスイッチし、それぞれの磁束は放出され、SFQ パルスとなって  $J_7$  に伝搬する。
- (4)  $J_7$  は、SFQ パルス 2 つ分の電流が加わる時、すなわち 2 つとも磁束が入っている状態の時のみスイッチするよう臨界電流値が設計されている。よって、(3) の場合の様に 2 つのパルスが加わると  $J_7$  がスイッチし、出力 ‘1’ を得ることができる。

これにより、2 入力とも論理値 ‘1’ の時のみ ‘1’ を出力する AND ゲート動作が実現する。

ここで示したように、SFQ 回路における組み合わせ回路はパルスの相互作用によって演算を行うため、入力を一旦保持して、SFQ パルス同士の合流や分岐といったタイミングを揃えて演算を行う必要がある。そのため、SFQ 回路の組み合わせ回路は基本的に、駆動パルス信号の入力によって動作する記憶素子付きの論理ゲートとなる。

## 3. SFQ マイクロプロセッサのアーキテクチャ設計空間

### 3.1 命令パイプライン駆動方式

命令パイプラインは実行時に複数の命令をオーバーラップさせる技術である。複数の命令は異なるステージに存在し、同時に各命令を処理することができる。SFQ マイクロプロセッサでは、命令パイプライン実現のために 2 つの SFQ パルス信号を用いている。

- クロック信号：全ての命令パイプラインの動作を同期させるために用いられる信号である。マイクロプロセッサの最大動作周波数はこのクロック信号の周期の逆数と定義され、マイクロプロセッサ内の最もレイテンシが長いパイプラインステージによって決定される。
- 駆動パルス信号：パイプラインステージ内の SFQ 組み

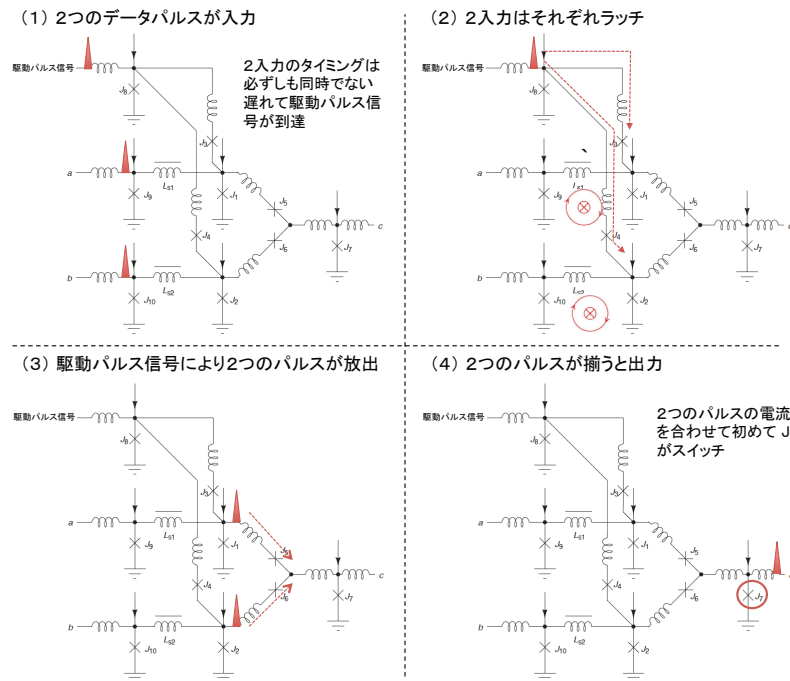


図4 ANDゲートの等価回路と動作

合わせ回路を動作させるために必要な信号である。この信号はCMOSマイクロプロセッサには存在しない。

### 3.2 マイクロアーキテクチャ・パラメータ

本節では、命令パイプライン構造を採る場合に考えられる2つのマイクロアーキテクチャ・パラメータについて述べる。

#### 3.2.1 パイプライン深度

パイプライン深度はパイプラインステージの数を表す。CMOS/SFQのいずれの回路を前提とした場合でも、マイクロプロセッサの性能を決める重要な要因となる。基準として、図5(a)に示す様な5段の命令パイプラインを考える。これをより深くしたパイプライン(図5(b))を持つものはスーパーパイプラインと呼ばれ、1パイプラインステージの遅延が小さくなるために動作周波数を向上させることができる。また、究極には論理ゲートレベルまでパイプラインを深くしたゲートレベルパイプライン(図5(c))も考えられる。

#### 3.2.2 データパスビット幅

データパスビット幅もまたマイクロアーキテクチャデザインパラメータの1つであり、ALUやレジスタファイル等の各ユニットが一度の処理で扱うビット幅を表す。データ語長は、マイクロプロセッサが扱うことができるデータのビット幅である。スライスとは、データ語長を分割したデータを指す。例えば、64ビットを8分割して得られた8ビットの各データをスライスと呼ぶ。また、この場合スライス数は8となる。以上を踏まえ、パイプライン深度と同様に設計空間の定義を行う。データ語長が32ビットの

マイクロプロセッサを例に説明する。ビットパラレル方式はデータパスビット幅とマイクロプロセッサのデータ語長が等しい場合を指す。現行の多くのCMOSマイクロプロセッサはこのデザインを採用している。対して、SFQマイクロプロセッサは、データパスビット幅が1ビットであるビットシリアル方式、もしくは、データパスビット幅が4ビットのようなデータ語長よりも小さいビットスライス方式を採用している。例えば、データパスビット幅が8ビットである場合、ALUやレジスタファイルといった演算ユニットでは4回処理を行うことで8ビットのデータを処理する。

ビットシリアル/スライス方式では、一定の駆動パルス信号間隔ごとにスライスを投入し、SFQ論理ゲートがそのラッチ機能を用いてデータを保持することでユニット内でのパイプライン化が可能となる。図6にビットシリアル/スライス方式でのデータ処理について示す。ここでは、論理ゲート8段、1つのデータを4つのスライスに分割した場合を想定しており、Timeは時間経過を表す。この図からもわかるように、同一ユニット内に複数スライスが存在可能になり、スライスレベルでのオーバーラップによる効果が得られる。ただし、設計者はユニット内での桁上げ信号のためのフィードバックループの存在に注意を払う必要がある。図7にフィードバックループが存在する回路例を示す。初期状態(図7(a))から1駆動パルス信号周期進んだ時(図7(b))、丸で囲んだ最後の論理ゲートではデータの待ち合わせのためのタイミング調整が必要となる。このように、フィードバックループのようにデータの流れと反対方向にデータが流れる箇所が存在すると、データの待ち合



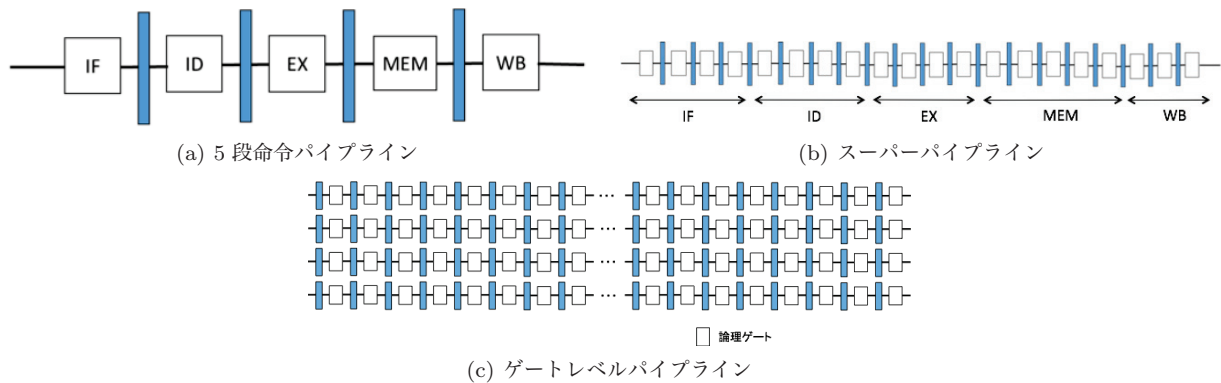


図5 パイプライン深度

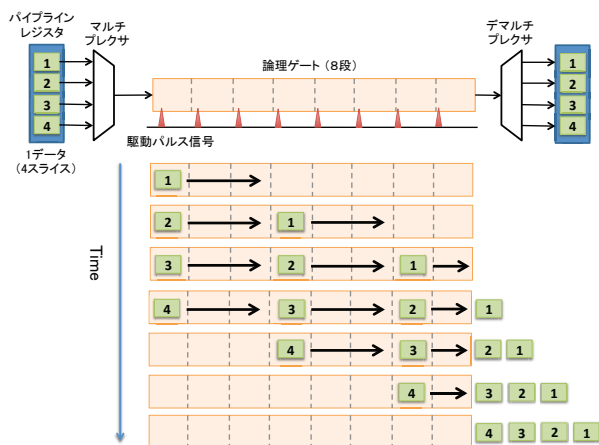


図6 ビットシリアル/スライス方式におけるスライスのオーバーラップ実行

わせのために駆動パルス信号周波数の調整が必要となる。一方、ビットパラレル方式ではユニット内にフィードバックループは存在しない。この時、スキューという回路遅延要素を用いることで、駆動パルス信号周波数を演算ユニットに供給可能な最大値に設定することができる。図8に示すように論理ゲート  $i-1$  段目から  $i$  段目までのデータパルス信号線  $D_{line}$  の遅延時間が駆動パルス信号周期  $T$  より長い場合、対応する駆動パルス信号線にスキューという回路遅延要素を挿入することでデータの到着と駆動信号パルスの到着の整合性を取ることができる。

#### 4. 性能評価

本節では、1命令当たりの実行に必要な時間を表す Time Per Instruction (TPI) に基づき、パイプライン深度とデータパスビット幅が SFQ マイクロプロセッサの性能に与える影響を解析する。また、SFQ マイクロプロセッサの有効性を明かにするため、CMOS マイクロプロセッサとの性能比較を行う。

##### 4.1 性能モデル式

プログラム実行における TPI は次の式で表すことがで

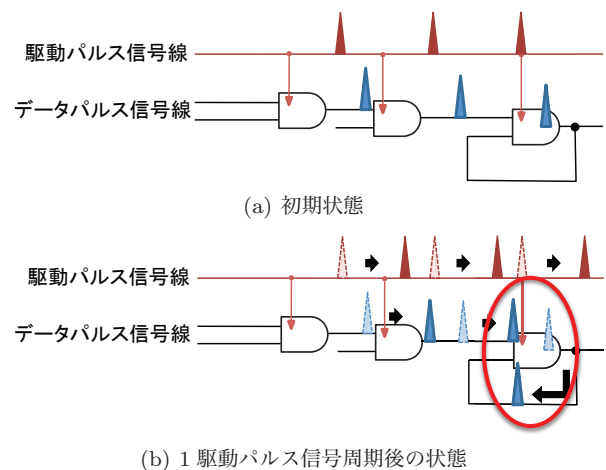


図7 フィードバックループが存在する回路

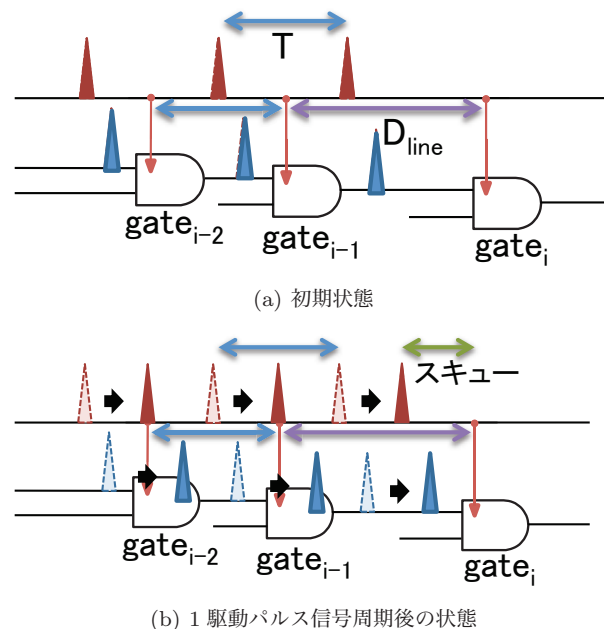


図8 スキューによる駆動パルス信号周波数高速化

きる [1].

$$TPI = \frac{T}{N_I} = \left( \frac{t_o}{\alpha} + \gamma \frac{N_H}{N_I} t_p \right) + \frac{t_p}{\alpha p} + \gamma \frac{N_H t_o}{N_I} p. \quad (1)$$

- $N_I$ : 全実行命令数.

- $N_H$ : 発生するパイプラインハザードの総数.
- $t_o$ : パイプラインレジスタでのデータ保持に要する時間. ラッチのセットアップ/ホールド時間により決まる.
- $t_p$ : 単一命令実行において通過する全論理ゲート遅延の総和.
- $p$ : パイプライン段数.
- $\alpha$ : スーパスカラ度.
- $\gamma$ : パイプラインでの命令実行に要する時間  $t_{op} + t_p$  に対する, パイプラインハザード当たりの平均パイプラインストール時間の割合. 最大値は 1.0 であり, これは最初のパイプラインステージに存在する命令がコミットされるまで後続命令の実行が開始できない状況に相当する. 一方, 最小値は 0 であり, これはパイプラインストールが発生しない状況を表す. 実際には, 各種ハザードに起因するパイプラインストールはマイクロアーキテクチャとハザード発生状況によって様々であるが, 本モデルでは平均値として表している.

第 1 項はパイプライン段数に依存しない, ストールと  $t_o$  によって生じる実行時間の増加である. 第 2 項は 1 命令の処理に必要な時間がパイプライン段数とスーパスカラ度によってどの程度減少するのかを表している. 第 3 項はストールが発生することで生じる  $t_o$  の増加分を示す.

ここで, 上記の TPI は次のように考えることができる. ストールが全く起きない場合, すなわち  $\gamma = 0$  の時, TPI は  $CPI=1$  の場合のクロックサイクル時間と考えることができる. ここでは理想クロックサイクル時間 (逆数は理想動作周波数) と呼ぶ.

#### 4.2 遅延パラメータ $t_o, t_p$ の設定

本評価ではスカラ・マイクロプロセッサ ( $\alpha = 1$ ) を想定する. そして, ビットシリアル, ビットスライス, ビットパラレル方式を採用した SFQ マイクロプロセッサ (それぞれ, SFQ-BSE, SFQ-BSL, SFQ-BP と略す), ならびに, CMOS での設計を前提とした従来のビットパラレル型マイクロプロセッサ (CMOS-BP と略す) に着目した性能比較を行う. マイクロプロセッサとマイクロアーキテクチャに基づく遅延パラメータ  $t_o, t_p$  を表 1 に示す.

##### 4.2.1 SFQ マイクロプロセッサにおける遅延パラメータ

SFQ マイクロプロセッサの遅延パラメータには, 現在多くの SFQ 回路で用いられている  $1.0\mu\text{m}$  プロセスの場合と, SFQ 回路に成り立つスケリング則が限界に達する  $0.3\mu\text{m}$  プロセス [4] の 2 通りを用いる.  $t_o$  は SFQ マイクロプロセッサに用いられている代表的な論理ゲート群のセットアップ/ホールドタイムの算術平均と, 製造ばらつきやジッタ対策のための動作マージンを足し合わせて求めている. また,  $t_p$  は CORE-1 $\beta$  のクリティカルパスである EX ステージを構成している SFQ 加算器の回路遅延

に CORE-1 $\beta$  のパイプライン段数である 7 を掛けて算出した. この SFQ 加算器の回路遅延は駆動パルス信号周期に, SFQ 加算器で処理に要したサイクル数を掛けあわせて求めることができる. 図 9 に加算器のレイアウトと回路遅延パラメータを示す. 回路遅延パラメータには  $1.0\mu\text{m}$  Nb プロセスの SFQ セルライブラリの値を利用している. 駆動パルス信号周期は SFQ 論理ゲートの回路遅延パラメータと SFQ 加算器のレイアウトに基づいて求められる. 以下で SFQ-BP と SFQ-BSE/BSL それぞれの場合の求め方について説明する.

- SFQ-BP では SFQ 加算器内にフィードバックループが存在しないため, 駆動パルス信号周波数は供給可能な最大値である  $t_o$  となる. また, 実際の設計ではワーストケースを想定し,  $1.0\mu\text{m}$  の時 20ps としている.
- SFQ-BSE/BSL では SFQ 加算器内にフィードバックループが存在し, このループにおける入力データの待ち合わせが必要となる. よって, フィードバックループで計算されるデータ到着までの遅延がクリティカルパスとなり, 駆動パルス信号周期を決定する. 図 9 に示すように, ビットシリアル/スライス SFQ 加算器では, フィードバックループで計算されるデータが通るパスが 2 つある. これらをそれぞれパス (A), パス (B) とし, このどちらかがクリティカルパスとなる. この加算器では, パス (A) がパス (B) より長く, パス (A) に含まれる論理ゲート要素は, AND ゲート, コンフルエンスバッファ (Confluence Buffer : CB), 超電導受動線路 (Passive Transmission Line : PTL), そしてスプリッター (Splitter : SPL) である. PTL は更に次の 2 つの要素に分けられる. データの変換, 受信に必要な遅延を  $PTL_{trans}$ , PTL セルユニットあたりの伝搬遅延を  $PTL_{prop}$  と呼ぶ. 以上より, パス (A) の遅延  $T_D$  は下記の式で表される.

$$T_D = AND + CB + PTL_{trans} + (dpw - 1) \times PTL_{prop} \times N_{PTL_{cell}} + SPL. \quad (2)$$

ここで,  $dpw$  はデータパスビット幅を指す. また,  $N_{PTL_{cell}}$  は図 9 に示すような 2 つのビット線の間に存在する PTL セルの数である.

ジョセフソン接合が  $1/a$  にスケールすると, スイッチング速度と PTL の遅延も  $1/a$  となる. 求めた SFQ 加算器の回路遅延は全て 64 ビットのデータ語長の場合を想定している. SFQ-BSL に関しては, 64 ビットのデータ語長の場合に最も性能の良いスライス幅であった 8 ビットスライスの時の回路遅延を採用している. ビットパラレル方式に対し, ビットシリアル/スライス方式の  $t_p$  がそれぞれ単純に 64 倍, 8 倍の遅延時間となっていないのは, 第 3.2.2 節で述べたスライスデータのオーバーラップ実行が遅延時間の短縮に効いているためである.

表 1 遅延パラメータ

	$t_o$	$t_p$
1.0 $\mu$ m SFQ-BP	13.32(ps)	2517.76(ps)
1.0 $\mu$ m SFQ-BSE		13232.8(ps)
1.0 $\mu$ m SFQ-BSL		4565.4(ps)
0.3 $\mu$ m SFQ-BP	3.995(ps)	755.328(ps)
0.3 $\mu$ m SFQ-BSE		3969.84(ps)
0.3 $\mu$ m SFQ-BSL		1369.62(ps)
CMOS-BP	86.76(ps)	4048.58(ps)

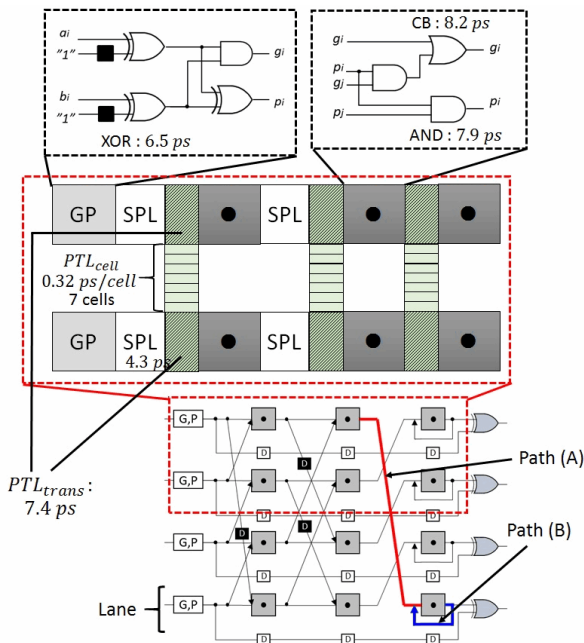


図 9 SFQ ビットスライス加算器の回路図とレイアウト図

#### 4.2.2 CMOS マイクロプロセッサにおける遅延パラメータ

CMOS マイクロプロセッサにおいては、14 段パイプライン、動作周波数 2.66 GHz のもの [2] をモデルとした。  $t_o$  はパイプラインのラッチオーバーヘッドとパイプラインステージ遅延の比を最も良い比率とされている 1.8 対 6 と仮定し [3]、動作周波数から逆算して求めた  $(1/2.66[\text{GHz}] \times 1.8 / (1.8 + 6))$ 。また、  $t_p$  は CMOS マイクロプロセッサのクロック周期にパイプライン段数を掛けたものから、  $t_o$  にパイプライン段数を掛けたものを引いた値  $(1/2.66[\text{GHz}] \times p - t_o \times p)$  である。以上のパラメータについてまとめたものを表 1 に示す。

#### 4.3 評価結果

図 10 に 1.0 $\mu$ m プロセス、0.3 $\mu$ m プロセスの SFQ-BP、SFQ-BSE、SFQ-BSL と CMOS-BP の理想動作周波数をそれぞれ示す。横軸は各マイクロアーキテクチャ、縦軸は理想動作周波数である。パイプライン段数はそれぞれ固定で、SFQ マイクロプロセッサにおいては CORE-1 $\beta$  のパイプライン段数である 7 段、CMOS マイクロプロセッサにおいてはモデルとして用いたマイクロプロセッサの 14 段

(CMOS-BP14 と呼ぶ) としている。また、ストールが発生しない理想動作周波数であるため、  $\gamma$  の値は 0 であり、結果は  $N_H/N_I$  の値に依存しない。

まず、SFQ-BP、SFQ-BSE、SFQ-BSL の 3 つを比較する。1.0 $\mu$ m プロセス時のそれぞれの理想動作周波数は 2.78 GHz、0.52 GHz、1.53 GHz となっている。これは、ビットシリアル/スライス化による  $t_p$  の増加が原因である。このことから、データバスビット幅は SFQ-BP が最も良いことが分かる。次に、CMOS-BP14 との比較を行う。1.0 $\mu$ m プロセス SFQ マイクロプロセッサにおいて、SFQ-BP、SFQ-BSE、SFQ-BSL の理想動作周波数は CMOS-BP14 と同程度、もしくはそれ以下の性能であることがわかる。一方、0.3 $\mu$ m プロセス時においては、SFQ-BSE は 1.76 GHz と CMOS-BP14 よりも低い性能となっているが、SFQ-BSL は 5.11 GHz、SFQ-BP においては 9.27 GHz と CMOS-BP14 より高い。しかしながら、その性能差は約 3.5 倍程度に留まっている。以上より、7 段の浅いパイプライン構造においては、SFQ 回路が有する本来の高速性を十分に活かしきれていないことがわかる。

表 1 から分かるように、SFQ 回路は CMOS 回路と比較して  $t_o$  が非常に小さいといった利点を有する。また、CMOS マイクロプロセッサではスーパーパイプライン化による動作周波数の向上は消費電力問題により極めて難しい、これに対し、SFQ 回路ではその超低消費電力性から消費電力問題は生じない。そこで、パイプライン段数をより深くした場合を考える。図 11 に 0.3 $\mu$ m 時の SFQ-BP、SFQ-BSE、SFQ-BSL と CMOS-BP の理想動作周波数を示す。横軸はパイプライン段数、縦軸は理想動作周波数である。破線は、SFQ-BP、SFQ-BSE、SFQ-BSL の実設計データに基づく駆動パルス信号周波数であり、それぞれ 166.67 GHz、119.90 GHz、76.66 GHz である。SFQ-BP のパイプライン段数を増加させた場合、377 段の時に SFQ-BP の理想動作周波数は 166.67 GHz に達成し、同パイプライン段数の SFQ-BSE、SFQ-BSL と比較しても高い性能である。一方、パイプライン段数を増加させた CMOS-BP は 11GHz 程度である。以上のことから、SFQ マイクロプロセッサで超高速な動作周波数を得るためには、SFQ-BP かつ深いパイプライン構成を取る必要があることがわかる。

ただし、式 (1) からわかるように、パイプライン段数を増加させると同時にストールによる影響が大きくなり、TPI が増加する。そこで、ストールが発生した場合の性能について考える。図 12 に 0.3 $\mu$ m プロセス、  $N_h/N_i = 0.5$  である SFQ-BP の  $\gamma$  の値を 0 から 1 まで変化させた場合の TPI を、CMOS-BP14 の理想クロックサイクル時間で正規化したグラフを示す。横軸はパイプライン段数、縦軸は正規化された TPI であり、  $\gamma = 0$ 、すなわちストールが発生しない場合、SFQ-BP は現行 CMOS マイクロプロセッサに比べ 62.66 倍の性能となる。しかしながら、ストール

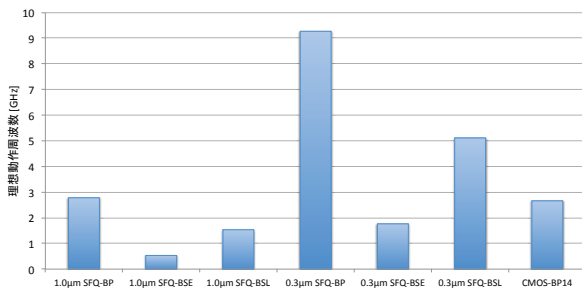


図 10 パイプライン段数を固定した時の理想動作周波数

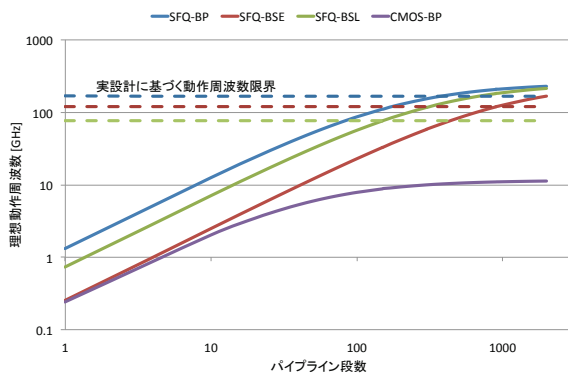


図 11 パイプライン段数が変化した時の理想動作周波数

の影響を反映した  $\gamma = 0.1$  の場合でもパイプライン段数 60 段の時に最高 5.67 倍の性能しか達成できていない。すなわち、ストールによる性能低下を回避しなければ高い動作周波数を達成できたとしても、性能は低いものとなる。

そこで、ストールの隠蔽できる場合を考える。新しくストール隠蔽率  $\theta$  ( $0 \sim 1$ ,  $\theta = 1$  の時ストールが全て隠蔽されていることを表す) を導入し、ストール時間を表す  $\gamma$  と掛け合わせる。  $\theta$  を導入した TPI のモデル式は以下のように表される。

$$TPI = \left( \frac{t_o}{\alpha} + (\gamma \times (1 - \theta)) \frac{N_H}{N_I} t_p \right) + \frac{t_p}{\alpha p} + (\gamma \times (1 - \theta)) \frac{N_H t_o}{N_I} p \quad (3)$$

式 (3) に基づいて算出した  $N_h/N_i = 0.5, \gamma = 0.5$  の場合の SFQ-BP の TPI を CMOS-BP14 の理想クロックサイクル時間で正規化したグラフを図 13 に示す。横軸がパイプライン段数、縦軸は正規化された TPI、グラフは  $\theta$  の値を 0.9 から 1 まで 0.01 ずつ、すなわちストール隠蔽率を 99% から 100% まで 1% ずつ増加させたものである。ストールを 99% 隠蔽できた場合、CMOS マイクロプロセッサに対する性能はパイプライン段数 300 段の時の 32.98 倍が最高となっている。このことから、ストール隠蔽率は極限まで高める必要があることがわかる。次節では、SFQ マイクロプロセッサにおけるストール隠蔽手法について議論する。

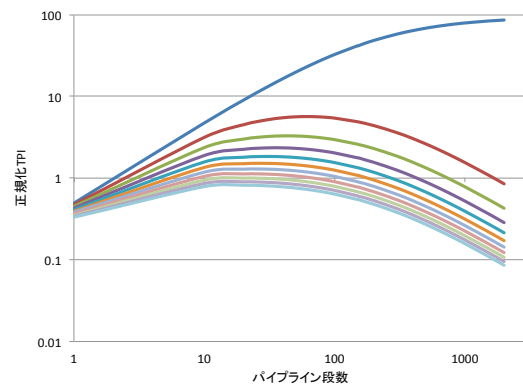


図 12 SFQ-BP と CMOS-BP の性能比較

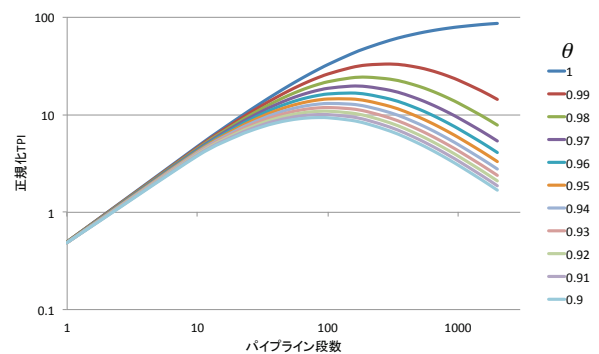


図 13  $\theta$  を考慮した SFQ-BP と CMOS-BP の性能比較

## 5. SFQ マイクロプロセッサ・アーキテクチャの設計方針

SFQ マイクロプロセッサによる飛躍的な性能向上を実現するためには、デバイス/回路特性と各種設計制約を勘案したマイクロアーキテクチャを導入しなければならない。そこで我々は、第 4 節で示した性能評価結果、ならびに、SFQ 回路と設計技術に関する現状を考慮し、マイクロアーキテクチャ決定に向け方針を採ることとした。

- ビットパラレル方式の採用：時間方向に処理を展開するビットシリアル/スライス方式と比較して、ビットパラレル方式は次に示す 3 つの利点を有する。1) 第 3 節で示したように、組合せ回路内部でのフィードバックループの発生を回避し、スキューの挿入により駆動パルス信号周波数を高めることができる。処理完了までのレイテンシは増大するが、これは後述する細粒度パイプライン処理でのオーバーラップ実行により隠蔽可能である。2) ビットレベル並列性を最大限に活用できるため、組合せ回路のレイテンシを低減することができる。3) 時間方向の繰り返し処理が不要なため、回路設計ならびにレイアウト設計においてタイミング調整が比較的容易になる可能性がある。
- ゲートレベル・パイプライン構成の採用：第 4 節で示したように、マイクロプロセッサレベル (すなわち、駆動パルス信号ではなくパイプライン全体の動作速度



を決定するクロック信号レベル)での高速動作を可能にするためには、300段程度と極めて深いパイプライン構成を採る必要がある。そこで、各パイプライン・ステージが一段の論理ゲートで構成されるゲートレベル・パイプライン構成を採る。これにより、従来のSFQマイクロプロセッサで必要であった駆動パルス信号とクロック信号を一本化(つまり、駆動パルス信号でパイプライン全体の動作を制御する)でき、設計容易化の観点からも利点がある。

- 大規模細粒度マルチスレッディング実行方式の採用：第4節で示したように、300段程度のパイプライン構成においてCMOSマイクロプロセッサを大幅に超える性能を実現するためには、ハザードに起因するパイプラインストール時間を99%以上隠蔽する必要がある。現代の高性能マイクロプロセッサではストール隠蔽技術としてアウトオブオーダー実行方式を採用している。しかしながら、パルス論理に基づくSFQ回路ではタイミング調整がクリティカルな設計となり、複雑な論理を要する命令ウィンド、リネーミング機構、高度な分岐予測機構、リオーダーバッファ、パイプラインステージ間を跨いだフィードバックループを形成するフォワーディング機構などを実装することが難しいと予想される。そこで、徹底したハードウェアの簡素化を実現しつつ、極めて深いパイプラインでのストール隠蔽を可能にすべく、パイプライン段数と同程度のスレッドを起動しクロック・サイクル毎に切り替えて実行を進める大規模細粒度マルチスレッディング方式を導入する。
- FIFOメモリを基本とするメモリ階層の採用：大規模細粒度マルチスレッディングを実現するためには、巨大なレジスタファイルを構成し、クロックサイクル毎に適切なレジスタセットを後段パイプラインステージへと出力しなければならない。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を基本とするFIFOメモリである。これは、SFQは回路内部にフィードバックループを持たない単方向データ流の処理を得意とすることに起因する。例えば、 $2.0\mu\text{m}$ プロセスにおいて $20\text{ps}$ で動作可能なシフトレジスタの設計事例があり[8]、SFQマイクロプロセッサCORE-1 $\beta$ のレジスタファイルや、CORE-1 $\gamma$ [6]のキャッシュメモリとして実装されている。そこで、FIFOメモリを用いた大規模細粒度マルチスレッディング向けレジスタファイルや、ストリーム式キャッシュメモリの搭載などを導入する。

図14に、今後検討を進めるSFQマイクロプロセッサ・アーキテクチャのイメージ図を示す。データパスはゲートレベルのパイプライン構成を採っており、レジスタファイ

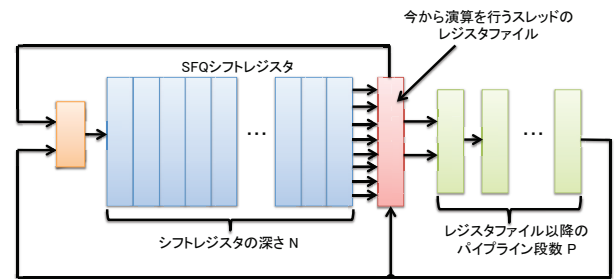


図14 SFQマイクロプロセッサ・アーキテクチャの概念図

ルはFIFOメモリによって実現する。1スレッド分のレジスタセットを全て同時読み出しできる構成であり、循環バッファとして動作させることでクロックサイクル毎のスレッド切り替えを可能にする。また、キャッシュミスなどのメモリアクセスにおいてストールが発生した場合には、パイプライン全体を停止する。これにより、パイプライン全体のタイミング制御を単純化し設計容易化を図る。

なお、現段階ではSFPマイクロプロセッサのアーキテクチャ設計方針を示したに過ぎない。今後は、本稿では言及していないメモリ・サブシステム構成法も踏まえ、さらなる詳細な検討が必要である。

## 6. おわりに

本稿では超高速なマイクロプロセッサの実現に向け、SFQ回路を利用したマイクロプロセッサのアーキテクチャ設計空間と性能モデルを用い、性能評価を行った。その結果、SFQマイクロプロセッサにおいて最も高い性能を発揮するためにはSFQ-BPかつ300段以上の非常に深いパイプライン構成が必要であることを示した。また、ストールが発生しない場合、 $0.3\mu\text{m}$ プロセスにおけるSFQ-BPは $166.67\text{GHz}$ を達成することがわかり、現行CMOSマイクロプロセッサの62.66倍の性能を発揮する。

しかしながら、ストールが発生すると、CMOS-BPに対してSFQ-BPの性能は $\gamma = 0.1$ の時5.67倍の性能しか発揮できない。そこで、ストール隠蔽率を導入し、ストールを隠蔽することによる影響を調査した。その結果、ストールを99%隠蔽することで32.98倍の性能を発揮できることがわかった。深いパイプライン構成を持つSFQマイクロプロセッサにおいては、ストール隠蔽率を極限まで高める必要がある。

さらに我々は、以上の結果を踏まえ、今後SFQマイクロアーキテクチャが採るべきアーキテクチャとして、ゲートレベルパイプラインと細粒度マルチスレッディングを採用したマイクロアーキテクチャの検討を行った。細粒度マルチスレッディングにより、データ依存関係・制御依存関係によって引き起こされるハザードを回避することができる。また、高速動作が可能なSFQシフトレジスタをレジスタファイルに利用することで、SFQマイクロプロセッサ

の高速動作をサポートできる。

今後は、SFQ マイクロプロセッサの設計を行うことで、実際のパイプライン段数はどの程度になるのかを調査し、シフトレジスタの深さと面積と性能の関係を明らかにする予定である。

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井上研究室の皆様にご心より感謝の意を表すと共に厚く御礼申し上げます。なお、本研究は、一部文部科学省科学研究費補助金 26540022 の助成による。

## 参考文献

- [1] HARTSTEIN, A., AND PUZAK, T. R. The optimum pipeline depth for a microprocessor. In *ACM SIGARCH Computer Architecture News* (2002), vol. 30, IEEE Computer Society, pp. 7–13.
- [2] HENNESSY, J. L., AND PATTERSON, D. A. *Computer architecture: a quantitative approach*. Elsevier, 2012.
- [3] HRISHIKESH, M., BURGER, D., JOUPPI, N. P., KECKLER, S. W., FARKAS, K. I., AND SHIVAKUMAR, P. The optimal logic depth per pipeline stage is 6 to 8 for 4 inverter delays. In *ACM SIGARCH Computer Architecture News* (2002), vol. 30, IEEE Computer Society, pp. 14–24.
- [4] 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* 11, 1 (2001), 1050–1055.
- [5] TANAKA, M., KAWAMOTO, T., YAMANASHI, Y., KAMIYA, Y., AKIMOTO, A., FUJIWARA, K., FUJIMAKI, A., YOSHIKAWA, N., TERAI, H., AND YOROZU, S. Design of a pipelined 8-bit-serial single-flux-quantum microprocessor with multiple alus. *Superconductor Science and Technology* 19, 5 (2006), S344.
- [6] TANAKA, M., YAMANASHI, Y., IRIE, N., PARK, H., IWASAKI, S., TAKAGI, K., TAKETOMI, K., FUJIMAKI, A., YOSHIKAWA, N., TERAI, H., ET AL. Design and implementation of a pipelined 8 bit-serial single-flux-quantum microprocessor with cache memories. *Superconductor Science and Technology* 20, 11 (2007), S305.
- [7] TSUHATA, T., YOKOTA, J., INOUE, K., AND TANAKA, M. Architectural design space exploration of single-flux-quantum microprocessors.
- [8] 吉川信行. 超伝導回路用メモリ技術の現状と展望. まぐね = *Magnetics Japan* 5, 1 (2010), 12–18.
- [9] 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. PhD thesis, 名古屋大学, 2006.