クリティカルパス・アイソレーションとビット幅削減を用いた過電圧スケーリング向け省電力設計手法

増田 豊[1,a)]　長山 準[3]　鄭 泰禹[2]　石原 亨[1]　籾山 陽一[3]　橋本 昌宜[2]

概要：本稿は，計算品質の制約下で，過電圧スケーリングの省電力効果を高める設計手法を提案する．提案設計はクリティカルパス・アイソレーション (Critical Path Isolation; CPI) とビット幅削減 (Bit-Width Scaling; BWS) を併用する．CPI を用いて本質的でないクリティカルパス (CP)，すなわち，しきい値電圧の高い論理セルやゲート幅の狭いセルで構成される CP を削減し，BWS により本質的な CP を低減する．両者の協調設計により，回路内の CP を大幅に削減し，低電圧化および省電力化を推進する．GPGPU プロセッサを用いて提案設計の省電力効果を評価したところ，画像処理プログラムにおいて PSNR 30dB の制約下で 42.7%，ニューラルネットワークの推論プログラムでは推論精度 98% の制約下で 51.2%，消費電力を削減できることを実験的に確認した．

1. Introduction

ポストムーア時代において，集積システムの省電力化と高性能化を推進可能な設計技術として，近似コンピューティング (Approximate Computing; AC) に注目が集まっている [1–3]．AC は，重要な計算を正確に実行し，他の演算を近似的に実行する設計指針である．冗長な計算を内包するアプリケーションと特に親和性が高く [1]，機械学習，ディジタル信号処理，画像処理，音声処理などの多様な分野において，省エネルギー化を促進できると期待されている．

省電力化を目的とした AC 技術として，過電圧スケーリング (Voltage Over-Scaling; VOS) が盛んに研究されている [4–7]．従来の電圧スケーリングでは，遅延故障が発生しない範囲で，電源電圧を削減する．一方，VOS では「回路内で遅延故障が発生した場合であっても，集積システムが一定の品質を保って動作していれば問題ない」という思想に基づき，計算品質などの AC の設計制約を満足する範囲で，電源電圧を積極的に低減する．VOS では，電源電圧の積極的な削減により，動的な消費電力を大幅に削減できる一方，遅延故障を起こしうる領域で動作するリスクが高い．従って，VOS を適用するために，設計者は，発生し得るタイミング故障が集積システムの異常動作に影響するかどうかを，慎重に見積もる必要がある．

[1] 名古屋大学 大学院情報学研究科
[2] 大阪大学 大学院情報科学研究科
[3] 株式会社ソシオネクスト
[a)] masuda@ertl.jp

VOS 動作時にシステムの正常動作を支えるための対策は，以下の 2 つに大別される; (1) 故障回復機構の利用，(2) VOS を前提としたタイミング最適化．第一の手法では，VOS 時のタイミング故障の影響を緩和するために，故障回復機構を追加する [4, 8]．この対策では，故障回復機構を注意深く設計することで計算品質の制約を担保できるが，面積オーバヘッドが大きいという問題がある．例えば，文献 [8] では，20% の面積オーバヘッドを要するとの報告がなされている．第二の手法では，タイミング最適化を行い，回路内のクリティカルパス (Critical Path; CP) を削減することで，VOS 時にタイミング故障を起こすパスを削減する [5]．この対策では，追加の故障回復機構を必要としないため，面積オーバヘッドが比較的小さい．

一方，近年，低電圧化の推進を目指した設計技術として，増田らによりクリティカルパス・アイソレーション (Critical Path Isolation; CPI) が提案された [9]．CPI は，活性化する CP に着目し，これらの CP のセットアップスラックを増加することで，低電圧動作時にタイミング故障を起こす CP の数を削減する．この方針は，タイミングクリティカルな FF であっても，活性化されなければ遅延故障は起こり得ない，という [5, 10] 等の手法と同様の考えに基づいている．文献 [9] では，Engineering Change Order (ECO) 再合成を利用して論理セルの置換や論理段数の変更を行うことで，活性化 CP の遅延を削減し，面積オーバヘッドを 1.4% に抑えつつ，電源電圧を 25% 削減した．本稿では，上記の電源電圧削減効果に着目し，CPI を ベースとした VOS 向け省電力設計技術に焦点を絞る．

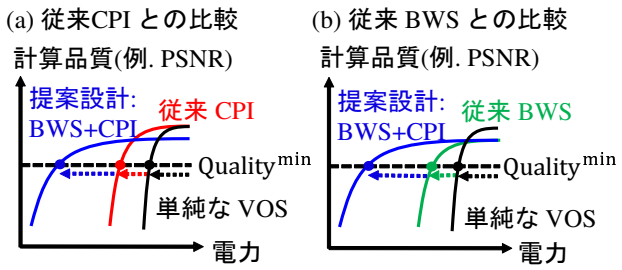

図 1 提案設計手法に期待する効果.

ここで，著者らは，「CPI が本質的な CP の遅延を削減できない」という根本的な課題を発見した．本質的な CP は，しきい値電圧の低いセルやゲート幅の広いセルにより構成され，論理セルの置換や論理段数の変更による遅延削減が困難なパスである．従って，本質的な CP が活性化し，AC の計算品質に影響しうる場合は，CPI の電源電圧削減効果が大幅に低下する可能性がある．以上より，上記の CPI の課題を解決して VOS に展開するためには，本質的な CP を削減可能な対策技術が不可欠である．

本研究では，VOS に向けた省電力設計手法を提案する．提案設計手法のキーアイデアは，CPI [9] とビット幅削減 (Bit-Width Scaling; BWS) [11, 12] の協調設計にある．BWS は，数値の表現に用いるビット幅を削減することで，データパスの実現に要する回路資源を低減する技術であり，回路面積，消費電力や CP の遅延を削減できる．図 1 に提案設計により期待される省電力効果を示す．図 1(a) に従来の CPI との比較図，図 1(b) に従来の BWS との比較図をそれぞれ示す．本稿では，Peak Signal-to-Noise Ratio (PSNR) などの計算品質に対して，制約を与えるものとする．図 1(a) において，従来の CPI は，本質的でない CP の遅延を削減することで，低電圧化動作時の計算品質の劣化を緩和する．一方，提案設計では，BWS と CPI を組み合わせることで，本質的な CP と本質的でない CP の両方を削減する．CP を大幅に削減することで，計算品質の劣化特性を更に改善し，低電圧化と省電力化を大きく促進できる．同様に，図 1(b) において，従来の BWS と比較すると，本質的でない CP の削減により，提案設計の方が消費電力をより削減できると期待される．

ここで，BWS は AC の設計技術の一種であり，計算精度の低下に起因して，計算品質を劣化させ得る．従って，計算品質の制約を満足しつつ消費電力を最小化するためには，設計者はビット幅，CPI 方法，電源電圧などの設計パラメータ組を注意深く設定する必要がある．一方，計算品質，回路構造および電源電圧などのパラメータ間の関係を表す関数は複雑である．また，VOS を前提とした最小動作電圧を評価するためには，回路のタイミング情報と想定するワークロードを用いて，計算時間の長い論理シミュレーションを実行する必要がある．これらの観点から，最適な設計パラメータ組を網羅的に探索することは，計算時間の観点で容易ではない．

以上の考えに基づき，本稿では，BWS と CPI の設計探索空間を大幅に削減可能な手法を提案する．本稿では，BWS と CPI の適用対象がタイミングクリティカルな FF もしくはパスであることに着目し，両適用対象の最小動作電圧が回路全体の最小動作電圧の良い下界となると仮定する．ここで，BWS は本質的な CP，CPI は本質的でない CP を対象としており，両者の遅延削減対象箇所は異なる．また，ビット幅は BWS の本質的な CP の遅延に影響し，CPI 方法は本質的でない CP の遅延に影響するとみなせる．これらの点に着目し，提案設計では，BWS のビット幅と CPI 方法を独立に設計する．両者を独立に設計することで，設計探索空間を大幅に縮小しつつ，VOS 下の電源電圧を最小化し，消費電力を大幅に削減する．

本研究の主な貢献は (1) CPI と BWS の混合設計法と (2) 複数の PVTA コーナーにおける提案設計手法の省電力効果の定量的評価にある．著者らの知る限り，BWS と CPI の協調設計手法の提案は本研究が初である．評価実験により，BWS と CPI は非常に親和性が高く，両者の協調が相乗的に省電力効果を高めることを示す．

本稿の以降の構成は以下の通りである．2 章では，想定する BWS と CPI を説明し，設計最適化問題を定式化する．3 章では，CPI と BWS の協調設計手法を提案する．4 章で提案設計による省電力効果を示し，最後に 5 章で結論を述べる．

2. BWS と CPI の設計方針

提案設計手法は BWS と CPI から構成される．2.1 節では想定する BWS と CPI を説明し，2.2 節において設計最適化問題を定式化する．

2.1 想定する BWS と CPI

まず，BWS の想定について説明する．本研究では，多様なワークロードを実行することを想定し，図 2 に示される，ビット幅を動的に調整可能な BWS 手法を採用する．また，チップ毎に異なるタイミング・マージンを消費電力削減に還元するために，電源電圧およびビット幅をチップ毎，ワークロード毎に調整可能であると想定する．

図 2 の BWS について説明する．ビット幅の削減数 (N_{red}) が，制御信号を通して指示される．指定された下位 N_{red} ビットが ”0” に置換され，BWS 対象の演算器 (例. 浮動小数点演算器) への入力に与えられる．例として，32 ビット入力の浮動小数点演算器において，仮数部 23 ビットに BWS を適用する場合を考え，$N_{\mathrm{red}} = 3$ が指示されたとする．この時，符号ビット 1 ビット，指数ビット 8 ビット，仮数部の上位 20 ビットとして元の論理値が入力され，仮数部の下位 3 ビットに対して ”000” が入力される．ここで，”0”

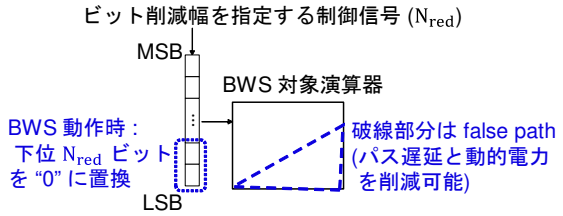

図 2 ビット幅を調整可能な BWS．BWS 動作時には，下位ビットが ”0” に置換される．置換されたビットを始点とするパスが false path になり，CP の遅延と動的電力が削減される．

に置換された入力ビットを始点とするパスは false path になるため，演算器内で活性化しない．従って，CP の遅延と動的電力を削減できると期待される．なお，N_{red} を増加すると，遅延と電力が削減される一方，計算品質の損失も増大する．すなわち，N_{red} は，計算品質の制約，電力，CP の遅延のトレードオフ関係を考慮して，慎重に決定される必要がある．3.2 節において，N_{red} の決定法について後述する．

図 3 に想定する CPI を示す．従来の回路設計フローでは，消費電力と面積を削減するため，CP 以外のパスに含まれるセルを，より小さな/高 Vth セルに置き換える．従って，CP の遅延に近いパスの数が増加する．本稿では，このような CP を本質的でない CP と呼称する．本質的でない CP の増加に伴い，低電圧化時にタイミング違反を起こすパス数が急激に増大するため，VOS 時に計算品質の制約を満足しつつ低電圧化を推進することが困難になる．一方，CPI は，本質的でない CP のタイミングスラックを増加し，CP 数を削減する．この場合，VOS 時にタイミング故障を起こす CP 数を低減できるため，低電圧効果を高めることが出来ると期待される．

ここで，CPI は従来のタイミング最適化の過程で取得していた省電力効果や面積削減効果の一部を手放している．従って，CPI を用いるためには，VOS 時の最小動作電圧，消費電力，面積に関するトレードオフを慎重に考慮する必要がある．以上の観点から，本研究では，CPI による省電力効果と面積削減効果の損失を抑えつつ，電源電圧削減効果を高めるために，活性化する CP 群を対象とする CPI 法に着目する．活性化する CP のスラックを増加することで，実際にタイミング故障を起こしうる CP 数を削減する狙いがある [5,9,10]．また，本研究では，文献 [9] を参照し，FF ベースの CPI を適用する．[9] の FF ベース CPI では，FF 毎に最大遅延制約を指定し，ECO 再合成により，制約を考慮したタイミング最適化を行う．なお，パス単位で制約を付与する CPI は，回路内の膨大なパスに対して設計制約を用意する必要があるため，非常に煩雑である．3.3 節では，CPI 対象 FF への遅延制約の決定法について説明する．

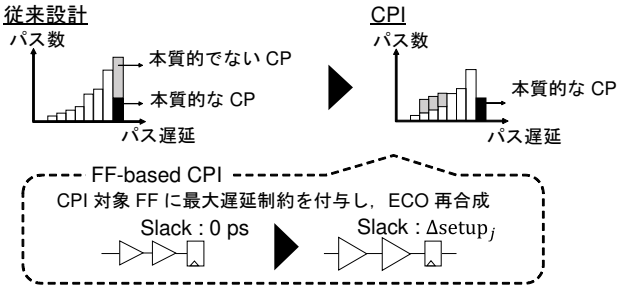

図 3 想定する FF ベースの CPI.

2.2 設計最適化問題の定式化

2.1 節の議論に基づき，本節では VOS 下での BWS と CPI の混合設計最適化問題を定式化する．

- 入力
  - CPI 前回路
  - N_W 個のワークロード
- 出力
  - CPI と BWS を適用した合成後回路
- 目的関数
  - 消費電力の最小化
- 制約
  - $\mathrm{Quality}_i \geq \mathrm{Quality}_i^{\min}$ $(1 \leq i \leq N_W)$
  - $\mathrm{Area} \leq \mathrm{Area}^{\max}$
  - $\mathrm{N_{LowVth}} \leq \mathrm{N_{LowVth}^{max}}$
- 変数
  - N_{red_i} $(1 \leq i \leq N_W)$
  - D_{FF_j} $(1 \leq j \leq N_{\mathrm{FF}})$

本最適化問題において，入力は CPI 前回路と N_W 個のワークロードであり，出力は CPI と BWS を適用した合成後回路である．目的関数は，VOS 下における消費電力の最小化である．設計時の制約は，計算品質 ($\mathrm{Quality}_i^{\min}$)，面積 ($\mathrm{Area}^{\max}$)，および，低 V_{th} セル数 ($\mathrm{N_{LowVth}^{max}}$) である．ここで，上記の設計制約は，設計対象回路の仕様に基づき，設計者に与えられると仮定する．変数 N_{red_i} は i 番目のワークロードを実行する際のビット削減数を指し，D_{FF_j} は ECO 再合成時に回路内の j 番目の FF に与えられる最大遅延制約を表す．N_{FF} は回路内の FF 数である．ここで，Area, $\mathrm{N_{LowVth}}$, $\mathrm{Quality}_i$，および Power は，N_{red_i} と D_{FF_j} に応じて変動する．

3. 提案設計手法

本章では，2.2 節で定式化した問題を解くために，設計手法を提案する．

3.1 概要

2.2 節の最適化問題では，Area, Power, $\mathrm{Quality}_i$, $\mathrm{N_{LowVth}}$, N_{red_i}, D_{FF_j} が非線形の関係を持ち，Area, Power, $\mathrm{Quality}_i$ の評価に比較的長い時間を要する．従って，上記の設計パラメータ組から構成される設計探索空間を網羅的に探索することは，計算時間の観点で困難である．

そこで，本研究では，BWS と CPI が対象とする CP の最小動作電圧が，回路全体の最小動作電圧の良い下界となると仮定し，両対象 CP の最小動作電圧の削減を目指す．動作電圧を低減することで，消費電力を削減する狙いがある．ここで，BWS のビット幅 (N_{red_i}) は本質的な CP，CPI 方法 (D_{FF_j}) は本質的でない CP を対象としているため，BWS と CPI が対象とするパス群は排他的である．従って，両 CP 群の遅延削減に着目すると，ビット幅と CPI 方法を独立に設計できる．この設計方針により，CP を大幅に削減して低電圧化と省電力化を推進しつつ，設計探索空間を大幅に縮小できる．

以上の考えから，本研究では，図 4 に示す 2 段設計手法を提案する．提案設計では，第一に，計算品質の制約を満足する最大の N_{red_i} を探索し，第二に面積と低 V_{th} セル数の制約下で D_{FF_j} 組を決定する．BWS と CPI の設計パラメータを独立に探索することで，設計探索空間を大幅に削減する．BWS と CPI を設計後，消費電力を最小化するために，最小動作電圧を探索する．次節から，N_{red_i} の探索と D_{FF_j} 組の決定法について，それぞれ説明する．

3.2 N_{red_i} の選択

まず，各 i 番目のワークロードに対して，N_{red_i} を決定する．この設計段階での目標は，BWS 対象のモジュールにおける，本質的な CP の遅延および動的な電力を出来るだけ削減することである．本質的な CP の遅延を削減することで，CPI 時の電源電圧削減効果を相乗的に高める狙いがある．以上より，本研究では，CPI との協調動作を踏まえて，計算品質の制約を満足する N_{red_i} の最大値を探索する．

ここで重要な点として，許容可能な N_{red_i} の最大値は，計算品質の制約と実行ワークロードに依存し，機能的検証により上界を得ることができる．従って，本設計では，Register Transfer Level (RTL) シミュレーションや命令セットシミュレーションなどの機能的シミュレーションを実行し，計算品質の制約を満足する N_{red_i} を探索する．この方針により，計算時間の長い論理シミュレーションの実行を省略できる．

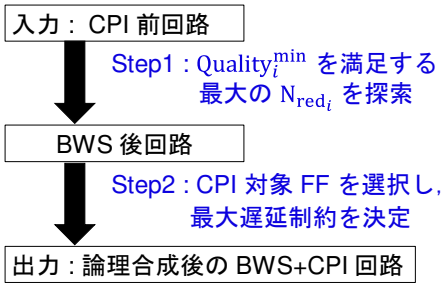

図 4 提案する 2 段設計手法. (1) Quality$_i^{\min}$ を満足する N_{red_i} の最大値を探索し，(2) D_{FF_j} 組を決定する．

3.3 D_{FF_j} 組の決定

次に，提案設計手法は，D_{FF_j} 組を決定する．図 5 に，本研究で用いる CPI フローを示す．なお，D_{FF_j} 組の決定については，文献 [9] などで提案されているような他の手法も，同様に適用できる．

図 5 の詳細について説明する．まず，BWS 適用後の回路と想定するワークロード群を入力し，活性化 FF 群を抽出する．次に，ECO 再合成用に，2 種類の制約を用意する．第一の制約は，従来の CPI [9] と同様に，活性化パスの終点 (活性化 FF) を対象とする．第二の制約は，BWS 対象モジュール内に潜在する本質的でない CP を削減するために，BWS モジュールへの入力上位 k ビットを始点とするパスを対象とする．

第二の制約のモチベーションについて，図 6 を用いて説明する．本稿で想定する BWS はビット幅を動的に調整可能な構成であるため，ビット幅を削減しない正確な演算をサポートする．すなわち，入力の LSB (Least Significant Bit) から出力の MSB (Most Significant Bit) まで，伝搬するパスが回路内に存在する．このパスが，多段のキャリー伝搬などの影響で，本質的な CP になると仮定すると，タイミング最適化時に他のパスが本質的でない CP になる可能性がある．従って，例えば，入力の上位 $N_{\mathrm{bit}} - N_{\mathrm{red}}$ ビットのいずれかを始点とし，出力の MSB を終点とするパスが，本質的でない CP となり得る．なお，N_{bit} は BWS モジュールへの入力信号のビット幅である．

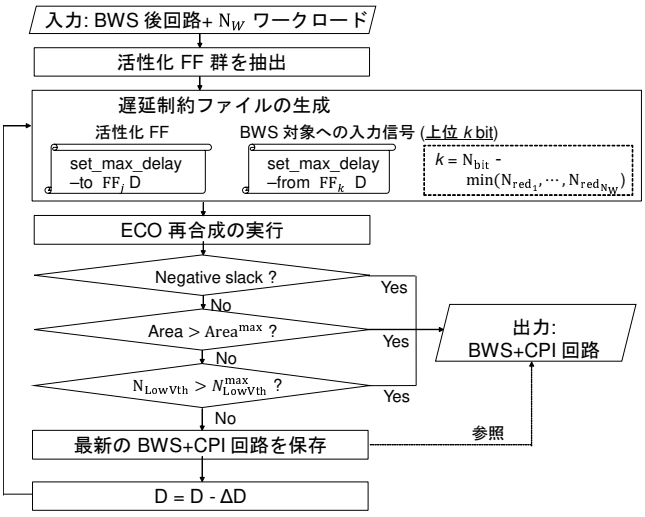

図 5 CPI フローの概要．

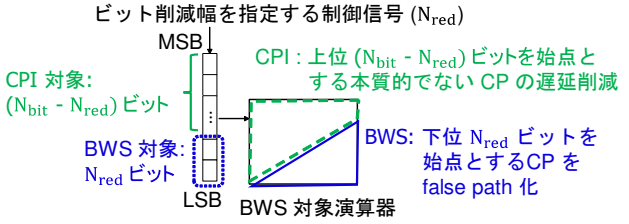

図 6 BWS 対象演算器の入力を始点とする CP を，CPI 対象に追加．

ここで，従来の終端 FF ベースの CPI を用いて，上記の本質的でない CP の遅延を削減する場合，終端 FF (出力の MSB) に対する最大遅延制約を厳しくして ECO 再合成などを行う．しかし，この CPI では，CPI 対象 FF が本質的な CP を終端として持つ場合に，対象 FF やパスの遅延を十分に削減することはできない．従って，従来の終端 FF ベースの CPI では，BWS モジュール内に潜在する CP を削減できず，電源電圧削減効果を十分に引き出せない可能性がある．一方，提案設計で付与する第二の制約では，BWS モジュールへの入力上位 k ビットに対して，最大遅延制約を付与する．k は以下の式に基づき決定される．

$$k = N_{\mathrm{bit}} - min(N_{\mathrm{red}_1}, \cdots, N_{\mathrm{red}_{N_W}}). \quad (1)$$

式 (1) において，$min(N_{\mathrm{red}_1}, \cdots, N_{\mathrm{red}_{N_W}})$ は，各ワークロード実行時のビット削減幅の最小値を指す．すなわち，$N_{\mathrm{bit}} - min(N_{\mathrm{red}_1}, \cdots, N_{\mathrm{red}_{N_W}})$ は，BWS が下位 $min(N_{\mathrm{red}_1}, \cdots, N_{\mathrm{red}_{N_W}})$ ビットを削減する際に，CPI が対象とするべき入力ビットの範囲を指す．式 (1) に基づき第二の制約を付与することで，入力下位ビットを始点とする CP を BWS により false path とし，残りの入力ビットから始まる CP を CPI により削減できるため，BWS モジュール内に潜在する CP を大幅に削減でき，電源電圧削減効果と省電力効果を推進できる．4.2.2 節では，第二の制約による，電源電圧削減効果の向上について説明する．

次に，最大遅延制約の遅延値 (D_{FF_j}) について議論する．ここで，CPI 対象 FF の最大遅延値を削減していくと，ある段階で制約を達成できない FF 群が出現する．本研究では，回路の最小動作電圧がこれらの FF 群により決定されると仮定し，制約を最初に違反する FF 群と違反時の最大遅延を導出する．このアプローチにより，D_{FF_j} の組合せ最適化問題を，D_{FF_j} の最大値探索問題に近似できる．D_{FF_j} の最大値探索問題では，各 CPI 対象 FF に対して同一の最大遅延制約を付与し，その制約を違反する FF が存在するか調査する，という方針を採用できる．すなわち，設計探索時のパラメータを，FF 毎の遅延制約値の組ではなく，単一の最大遅延制約値のみに削減できるため，CPI に要する工数や計算時間を大幅に削減できる．

以上の議論に基づき，図 5 では，各 CPI 対象 FF に対して，同一の遅延制約値 (D) を付与する．この方法により，ワースト遅延を最大限削減した CPI 回路を生成する．本研究では，ΔD を差し引くことで D を更新し，更新後の D を用いて ECO 再合成を実行する．なお，ΔD は，設計工数や設計時間を考慮して，設計者が調整することを想定する．ECO 再合成後に，セットアップ制約を満足しない FF が出現した場合は，CPI 対象 FF が遅延制約を違反しているため，ECO 再合成のループ処理を終了し，前回の合成結果を CPI 回路として出力する．2.2 節の面積や低 V_{th} セル数に対する制約についても，タイミングスラックと同様に判定する．

4. 評価実験

本章では，提案設計手法の省電力効果を定量的に評価する．4.1 節では評価環境を説明する．4.2 節で評価結果を示し，従来の CPI や BWS と比べて消費電力を大幅に削減できることを述べる．

4.1 評価環境

本評価実験では，オープンソースの GPGPU プロセッサである，Nyuzi プロセッサ [13] を対象回路として選択した．この回路を，商用ツールと 45 nm プロセスの Nangate スタンダードセルライブラリを用いて論理合成した．合成後ネットリストは 184,243 個の組合せ論理セルと 29,456 個の FF を持ち，最小クロック周期はワーストコーナーで 1.24 ns であった．

ワークロードとして，Mandelbrot 集合の描画プログラムとニューラルネットワークの推論プログラムの 2 種類を選択した．推論プログラムでは，2 次元識別問題である Fourclass データセット [14] を対象とし，学習済みのデータを利用した．(入力層-隠れ層-出力層) の 3 層構造から構成されるニューラルネットワークに対して，学習後の重みを用いて初期化し，テストデータの実行を通して，識別精度を評価した．なお，各層におけるニューロン数はそれぞれ，入力層 2 個，隠れ層 8 個，出力層 2 個であり，隠れ層の出力に活性化関数として ReLU (Rectified Linear Unit, y = max(x, 0)) が搭載されている．計算品質の制約 ($\mathrm{Quality}^{\mathrm{min}}$) として，Mandelbrot では 30 dB の PSNR，Fourclass では 98% の推論精度をそれぞれ設定した．なお，これらの $\mathrm{Quality}^{\mathrm{min}}$ はあくまで一例であり，他の条件においても，提案設計は全く同様に適用できる．

次に，Nyuzi プロセッサに対して BWS と CPI を適用した．本研究では，32 ビットの浮動小数点演算ユニット (Floating-Point Unit; FPU) に着目した．FPU は消費電力と面積が大きく [15]，しばしば本質的な CP を持つ演算器である．本実験では，FPU の仮数部を削減しながら，RTL シミュレーションを繰り返し実行し，$\mathrm{Quality}^{\mathrm{min}}$ を満足する N_{red} の最大値を評価した．図 7 に評価結果を示す．図 7 より，Mandelbrot プログラムでは $N_{\mathrm{red}} \leq 13$，Fourclass プログラムでは $N_{\mathrm{red}} \leq 20$ において，それぞれ 30 dB の PSNR

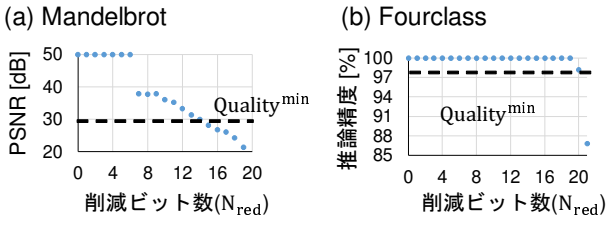

図 7　N_{red} と $\mathrm{Quality}^{\mathrm{min}}$ の関係．(a) Mandelbrot，(b) Fourclass.

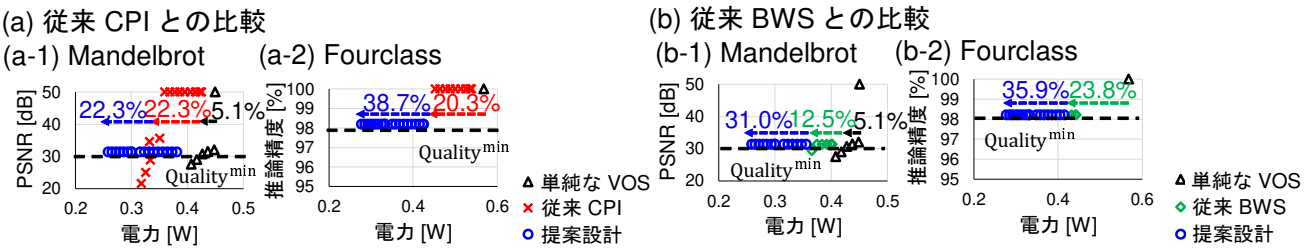

図 8 提案設計の省電力効果．(a) 従来 CPI，および，(b) 従来 BWS との比較．

と 98% の推論精度を達成している．これらの結果から，N_{red} として，Mandelbrot プログラムでは 13，Fourclass では 20 をそれぞれ選択した．なお，PSNR については，出力画像が正解結果と同一の場合に ∞ を取るため，図 7(a) では可視化のために，PSNR の最大値を 50 dB としてプロットした．

次に，CPI を BWS 後回路に適用した．CPI 対象 FF の最大遅延制約を 3.3 節の設計フローに基づき更新しながら，ECO 再合成を繰り返し実行した．面積と低 V_{th} セル数の制約 ($\mathrm{Area}^{\max}$ と $\mathrm{N}^{\max}_{\mathrm{LowVth}}$) として，初期回路 (図 4 における入力回路) の 101.0% と 103.0% の値をそれぞれ設定した．ECO 再合成時の遅延制約の更新幅 (ΔD) として，10 ps を選択した．なお，提案設計手法は，他の条件においても，全く同様に適用可能である．初期回路，BWS 回路，CPI 回路，提案設計後の回路に対して，クロック周期を 1.24 ns に固定しつつ，VOS 論理シミュレーションを実行し，電源電圧と計算品質のトレードオフ関係を評価した．その後，各回路と電源電圧の組に対して，商用の電力評価ツールを用いて消費電力を評価し，消費電力と計算品質のトレードオフ関係を導出した．

4.2 評価結果

本節では，まず提案設計による省電力効果を示し，次に CPI と BWS の効果についてそれぞれ議論する．

4.2.1 提案設計の省電力効果

図 8 に消費電力と計算品質のトレードオフ評価結果を示す．この図において，黒色のプロットは，タイミング最適化を行わずに電源電圧を低減する，単純な VOS の結果を表す．また，赤色，緑色，青色はそれぞれ，従来の CPI，従来の BWS，および提案設計の評価結果を示す．消費電力の基準点として，Mandelbrot では 450.0 mW，Fourclass では 567.5 mW を設定した．これらは，VOS を行わないワーストケース設計の消費電力である．本節では，以下の 2 つの観点から評価結果を議論する; (1) 提案設計全体による省電力効果，(2) 提案設計，従来の CPI，および従来の BWS 回路間の消費電力の比較．

まず，提案設計全体の省電力効果を議論するために，黒色と青色のプロットを比較する．図 8 より，提案設計が計算品質の制約を満足しつつ，消費電力を大幅に削減していることが読み取れる．例えば，図 8(a-1) より，提案設計は消費電力 257.8 mW の時点で PSNR 30 dB を達成しているが，単純な VOS では 427.2 mW の消費電力を必要とする．換言すれば，提案設計は 450.0 mW から 257.8 mW まで 42.7% の省電力効果を達成し，単純な VOS では 450.0 mW から 427.2 mW まで 5.1% の省電力効果にとどまっている．同様に，図 8(a-2) より，Fourclass では，提案設計は 567.5 mW から 277.4 mW まで，消費電力を 51.2% 低減している．初期回路と比較して，低 V_{th} セル数は 0.11% 増加し，面積は 0.58% 減少した．

次に，従来の CPI，従来の BWS，および提案設計の消費電力を比較する．図 8 より，提案設計が，従来の CPI と BWS と比べて，消費電力をさらに削減していることが分かる．例えば，図 8(a) より，従来の CPI と比較して，提案設計は Mandelbrot では 22.3%，Fourclass では 38.7% の省電力効果を上乗せしている．同様に，図 8(b) より，従来の BWS を基準として，提案設計は 31.0% と 35.9%，消費電力を削減している．これらの評価結果から，BWS と CPI の親和性は非常に高く，両者の協調設計最適化が VOS 時の省電力効果を相乗的に高めることを実験的に確認した．

4.2.2 考察

4.2.1 項の評価結果は，提案設計が消費電力を大幅に削減していることを示した．本項では，これらの結果をより詳細に分析する．

まず，提案設計の電源電圧削減効果と BWS の動的電力削減効果を軸に，提案設計の省電力効果を議論する．図 9 に，電源電圧と計算品質のトレードオフ関係を示す．図 9 より，提案設計が，より低い電源電圧において，計算品質の制約を満足していることが読み取れる．例えば，図 9(a) より，提案設計は 0.93 V において $\mathrm{Quality}^{\min}$ を満たしている一方，単純な VOS では 1.07 V の電源電圧を要している．換言すれば，単純な VOS を基準として，提案設計は 13.0% の V_{dd} 削減効果を達成している．このような電源電圧削減効果に起因して，図 8 に示した通り，動的な消費電力が劇的に削減されている．同時に，図 9 の結果から，VOS の電源電圧削減効果を高めるためには，本質的な CP と本質的でない CP の両方を削減することが重要である，という知

見が得られた．

次に，提案設計の CPI フローの効果を確認するために，従来の終端 FF ベースの CPI と BWS の混合設計と，提案設計の電源電圧削減効果を比較する．図 10 に比較結果を示す．図 10 より，提案設計の方が，より大きく電源電圧を低減できていることが分かる．例えば，Mandelbrot プログラムの例では，100 mV から 170 mV まで 70 mV，電源電圧削減効果を高めている．この結果から，BWS 対象回路への入力ビットを始点とする CP を，CPI により削減することで，本質的でない CP を効果的に削減し，低電圧化を推進できることを実験的に確認した．換言すれば，提案設計のような始点への制約を付与した CPI 法が，ビット幅を調整可能な BWS の弱点を補い，低電圧効果が増幅されていることを確認した．

図 11 に，BWS 適用有無での，Nyuzi プロセッサの消費電力比較結果を示す．図 11 より，同一の電源電圧においても，BWS が消費電力を劇的に削減していることが分かる．例えば，電源電圧 0.95 V において，BWS は 398.6 mW から 311.5 mW まで 21.9% の省電力効果を発揮している．この省電力効果は，2.1 節の図 2 で述べた通り，動的電力の削減に起因する．このような動的電力の削減も，提案設計の省電力性を後押ししていると言える．

最後に，異なる PVTA コーナーでの，提案設計の省電力効果を示す．本実験では，遅延と電力のライブラリファイル (liberty ファイル) の情報をワーストコーナーから typical コーナーに置換し，最小動作電圧をスイープした．なお，ゲートレベル・ネットリストとクロック周期は変更せずに，コーナー情報のみ更新した．コーナー情報を変更することで，回路内の論理ゲートの遅延値や感度が大幅に変動する．従って，異なる PVTA コーナーにおいても，提案設計が同様の省電力効果を発揮できるか評価することで，提案設計の PVTA ばらつきへの脆弱性を実験的に評価できる．

図 12 に評価結果を示す．図 12 より，Typical コーナーにおいても，提案設計が消費電力を劇的に削減していることが分かる．例えば，図 12(a) と (b) より，Mandelbrot プログラムでは 218.1 mW から 133.0 mW まで 39.1 %，Fourclass では 273.7 mW から 146.2 mW まで 46.6% の省電力効果を達成している．以上より，提案設計が遅延感度の大きく異なる PVTA コーナー群で，省電力効果を発揮できることを実験的に確認した．今後の課題の一つとして，提案設計を dynamic frequency voltage scaling (DVFS [16]) や adaptive voltage scaling (AVS) [17] などの自律性能制御技術に応用し，多様な PVTA コーナーで自律的に VOS 動作を実現する設計技術を実現することが挙げられる．

5. まとめ

本稿では，VOS 向けの省電力設計手法を提案した．提案設計手法の肝は，CPI と BWS の協調設計にある．従来の

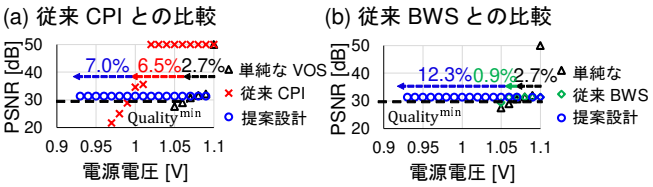

図 9　Mandelbrot 実行時の電源電圧削減効果．
(a) 従来 CPI，および，(b) 従来 BWS との比較．

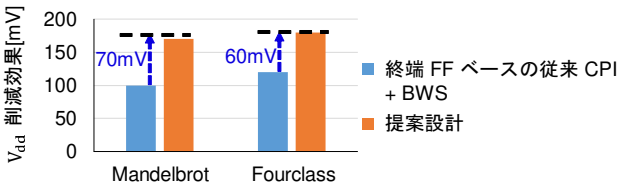

図 10　従来 CPI + BWS と提案設計の V_{dd} 削減効果の比較．

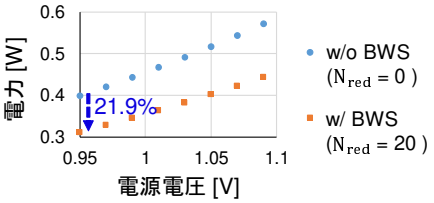

図 11　BWS の省電力効果 (Fourclass 実行時)．
同一の電源電圧においても，BWS 適用時の方が省電力．

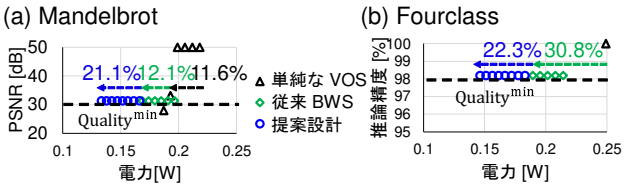

図 12　Typical コーナーでの提案設計の省電力効果．
(a) Mandelbrot，(b) Fourclass．

CPI は本質的な CP の遅延を削減できないという課題を持ち，この課題により CPI の電源電圧削減効果と省電力効果を十分に引き出せない可能性が存在した．一方，提案設計では，BWS と CPI の併用により，本質的な CP と本質的でない CP の両者を大幅に削減し，電源電圧と消費電力を大幅に削減する．評価実験を行ったところ，BWS と CPI の親和性は非常に高く，両者の協調設計により，省電力効果を相乗的に高めることを確認した．GPGPU プロセッサを用いたケース・スタディにより，提案設計の効果を評価したところ，画像処理プログラムにおいて 42.7%，ニューラルネットワークの推論プログラムにおいて 51.2% の省電力効果を実験的に確認した．

参考文献

[1] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” *Proc. ETS*, pp. 1-6, 2013.
[2] V. K. Chippa, S. T. Chakradhar, K. Roy and A. Raghunathan, “Analysis and characterization of inherent application

resilience for approximate computing," *Proc. DAC*, pp. 1-9, 2013.

[3] Q. Xu, T. Mytkowicz and N. S. Kim, "Approximate computing: A survey," *IEEE Design & Test*, vol. 33, no. 1, pp. 8-22, 2016.

[4] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE TVLSI*, vol. 9, no. 6, pp. 813-823, 2001.

[5] A. B. Kahng, S. Kang, R. Kumar and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," *Proc. ASPDAC*, pp. 825-831, 2010.

[6] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan and K. Roy, "IMPACT: Imprecise adders for low-power approximate computing," *Proc. ISLPED*, pp. 409-414, 2011.

[7] R. Ragavan, B. Barrois, C. Killian and O. Sentieys, "Pushing the limits of voltage over-scaling for error-resilient applications," *Proc. DATE*, pp. 476-481, 2017.

[8] B. Shim, S. R. Sridhara and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE TVLSI*, vol. 12, no. 5, pp. 497-510, 2004.

[9] Y. Masuda, M. Hashimoto and T. Onoye, "Critical path isolation for time-to-failure extension and lower voltage operation," *Proc. ICCAD*, pp. 1-8, 2016.

[10] S. Ghosh, S. Bhunia and K. Roy, "CRISTA: A new paradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation," *IEEE TCAD*, vol. 26, no. 11, pp. 1947-1956, 2007.

[11] J. Y. F. Tong, D. Nagle and R. A. Rutenbar, "Reducing power by optimizing the necessary precision/range of floating-point arithmetic," *IEEE TVLSI*, vol. 8, no. 3, pp. 273-286, 2000.

[12] D. Kim, J. Kung and S. Mukhopadhyay, "A power-aware digital multilayer perceptron accelerator with on-chip training based on approximate computing," *IEEE TETC*, vol. 5, no. 2, pp. 164-178, 2017.

[13] J. Bush, NyuziProcessor Source code. https://github.com/jbush001/NyuziProcessor, 2015.

[14] C. Chang and C. Lin, Fourclass, 1996. https://www.csie.ntu.edu.tw/˜cjlin/libsvmtools/datasets/binary.html.

[15] T. Cheng, Y. Masuda, J. Chen, J. Yu, and M. Hashimoto, "Logarithm-Approximate Floating-Point Multiplier is Applicable to Power-Efficient Neural Network Training," Integration, the VLSI Journal, vol. 74, pp. 19-31, 2020.

[16] T. D. Burd, T. A. Pering, A. J. Stratakos and R. W. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE JSSC*, vol. 35, no. 11, pp. 1571-1580, 2000.

[17] K. A. Bowman *et al.*, "A 45 nm resilient microprocessor core for dynamic variation tolerance," *IEEE JSSC*, vol. 46, no. 1, pp. 194-208, 2011.