

# 近似加算器を用いた低消費電力 JPEG 回路に関する研究

濱寛之<sup>1</sup> 佐藤寿倫<sup>2</sup> 請園智玲<sup>2</sup>

**概要** : Digital Signal Processing (DSP) を行う音声処理, 画像処理, データマイニングのようなアプリケーションは, 必ずしも正確な演算を必要としない. JPEG 画像は, その性質を利用した画像処理アプリケーションの 1 つである. しかしながら, JPEG 画像を生成する際多数の計算を要する. 近似演算器は, 電力効率を高める有望な技術として研究者の間で注目を集めている. この演算器は, 正確な演算をする代わりに近似値を出力する. 近似的な演算を行うことで回路面積・消費電力・遅延の削減を実現する. 我々は, 近似計算技術を使用して JPEG 回路の低消費電力化の研究を実施している. しかしながら, 以前の検討では近似計算技術を適用することによる効果は限定的であった. 本報告では, この問題を解決するための訂正回路を提案する. 実験により Ripple Carry Adder と比較して, 画質を維持しつつ最大で約 30% の電力削減を確認している.

**キーワード** : 近似加算器, JPEG, DCT

## 1. はじめに

近似演算器は, 遅延や電力効率を高める有望な技術として研究者の間で注目を集めている[1]. この演算器は, 正確な演算をする代わりに近似値を出力する. 近似的な演算を行うことで回路面積・消費電力・遅延の削減を実現する. エラー耐性のあるアプリケーションに近似演算を適用することで, 近似演算器は, 従来は相反する関係にあったエネルギー効率と性能の両立を実現する可能性を秘めている.

エラー耐性のある代表的なアプリケーションとして, Joint Photographic Experts Group (JPEG) [2]が挙げられる. JPEG は, 静止画像符号化方式の国際標準である. 離散コサイン変換 (Discrete Cosine Transform, DCT) [3]を使用する方式では, 非可逆圧縮方式が採用されており, 画質をある程度犠牲にする代わりに, 高い圧縮率を実現する. そのため, 画質に強いこだわりがない場合, 大量の画像データのデータ容量を抑えて保存することが可能である. また, IoT デバイスではリソースに制約があるため, 通信に利用できる帯域幅が限られることが多い. これにより, JPEG のような高圧縮技術は, リソース制約の厳しい IoT デバイスにおいて非常に有用である. 一方で, DCT 処理には多量の加算を必要とする. これにより, 計算に大量の計算時間を要する[4]. そういうわけで, 加算はエネルギーや時間効率を決定する.

本研究では, 近似加算器を適用することによる JPEG 回路の低消費電力化を図る. 我々は, DCT 処理に多数の加算が行われることに着目した. DCT 内の加算部分に対して, 近似加算を適用することで性能と消費電力のトレードオフを図る. 以前の検討で近似加算器に Carry-Maskable Adder (CMA) [5]を用いたところ, 効果は限定的であった[6]. 本報告ではこの現象を考察し, この問題を解決する手法を提案する.

この節では, 研究背景や本研究の目的について述べた. 残りの節の構成は, 以下のように構成される. 次節では, JPEG のアルゴリズムの 1 つである DCT について解説し, 3 節では本研究のメインの近似加算器として使用した CMA について説明する. 4 節では, DCT に対しての近似適用に関する関連研究や本研究における課題について述べ, この問題を解決するための訂正回路についても紹介する. 5 節では実験方法と評価結果を示し, 考察する. 6 節では, 本研究の総括を述べる.

## 2. JPEG

JPEG が提案する規格には, 離散コサイン変換 DCT を用いる方式と予測符号化を用いる spatial 方式がある. DCT 方式は画質を犠牲にする代わりに高圧縮率を得ることができる非可逆符号化方式である. 必要に応じて圧縮率を選ぶことができる. Spatial 方式は画質の劣化が生じないが低圧縮

1 福岡大学大学院 工学研究科 電子情報工学専攻  
Fukuoka University, Nanakuma, Fukuoka 814-0180, Japan  
2 福岡大学 工学部 電子情報工学科  
Fukuoka University, Nanakuma, Fukuoka 814-0180, Japan

率となる可逆符号化方式である。本研究では、DCT 方式による JPEG 圧縮を使用した。次節では、JPEG 圧縮の手順の 1 つである DCT について説明する。

## 2.1 DCT

DCT では、画像を空間情報から周波数情報に変換する。まず、 $8 \times 8$  ピクセル四方のブロックに分割し DCT によってブロック毎に周波数分解をする。DCT は以下のような式で求める。

$$D(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 S(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \dots (1)$$

ここで、 $u, v$  は DCT 行列の要素番号、 $x, y$  は画像行列の要素番号、 $S(x, y)$  は画素のピクセル値である。また、 $u, v=0$  のとき  $C(u), C(v)=1/\sqrt{2}$  でそれ以外は 1 になる。DCT を施して求められた周波数は  $8 \times 8$  サイズになり、図 1 のように、左上が低周波数成分、右下が高周波数成分になる。

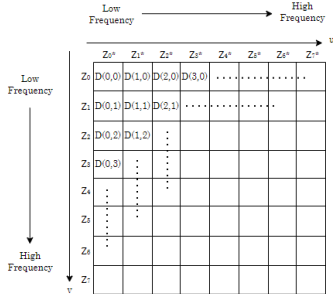


図 1 DCT 係数

## 2.2 Loeffler DCT

一般的には 2-D DCT は、計算の複雑性を緩和させるために、対応する行と列の 2 つの 1-D DCT に分解され、それぞれ行方向と列方向に沿って 2 回変換される。よって 2-D DCT は、図 2 に示すように 2 個の 1-D DCT と Transposition Memory を用いて実装される。ここで  $S$  はピクセル値、 $Z$  は Row DCT 後の DCT 係数、 $Z^T$  は Column DCT 後の DCT 係数、 $D$  は 1 つの DCT 係数を示している。しかしながら、1-D DCT による実装でも、依然として 2D-DCT の計算量は膨大である。

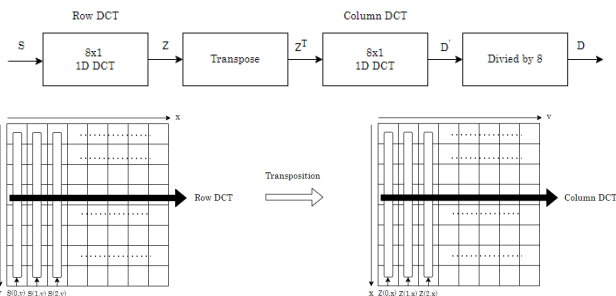


図 2 2-D DCT

この問題を解決するために提案された DCT アルゴリズムが、Loeffler DCT[7]である。図 3 は、1-D Loeffler DCT のアルゴリズムを示している。1-D Loeffler DCT は、Stage0~3 までの 4 個のステージに分けて DCT 係数を計算する。左

端の数字は  $8 \times 1$  画素値の要素番号を表しており、右端の数字は、 $8 \times 1$  1-D DCT 要素番号を表している。各ステージでの入出力の関係は下図で説明されており、記号は下にある式に対応している。このアルゴリズムにより、11 回の乗算と 29 回の加算で 1-D DCT 係数を求めることができる。

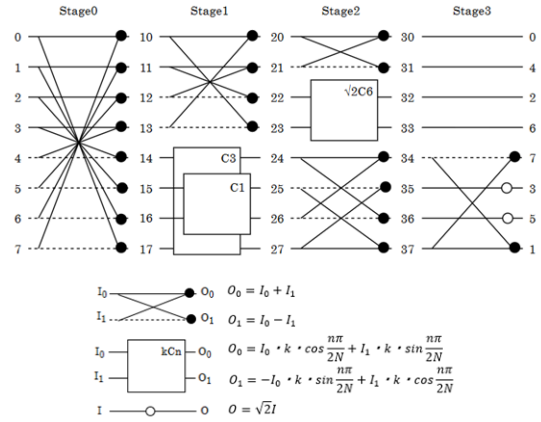


図 3 Loeffler DCT

## 3. Carry Maskable Adder

図 4 は、16-bit CMA の回路構成を示している。CMA は従来の Ripple-Carry-Adder (RCA) に類似した構成を持つ。具体的には、従来の Full-Adder に相当する CMFA と呼ばれる近似加算器を、リプル接続することで構成される。CMA は、加算器を 2 つのパートに分けることができ、それぞれ下位パートと上位パートに分けられる。

さらに、CMA では新たに追加された  $mask\_x$  信号によって近似精度を柔軟に制御することができる。

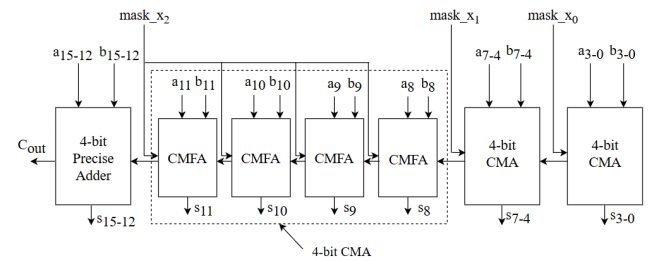


図 4 Carry-Maskable-Adder

図 5 は、CMFA の回路図を示している。2 つの回路には、従来の FA の入出力信号に加えて、新たに Carry 制御信号  $mask\_x$  が追加される。 $mask\_x=1$  ならば、CMFA は従来の FA と同様の動作をする。一方で、 $mask\_x=0$  のとき、CMFA は近似演算を行う。出力  $S$  は 2 入力  $a$  と  $b$  の OR 演算によって計算され、キャリー出力  $Cout$  は 0 に固定される。

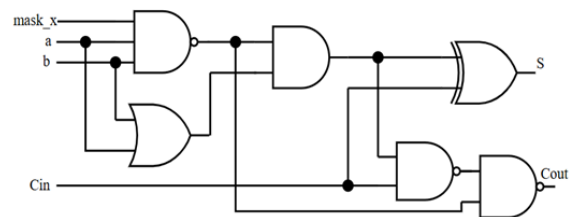


図 5 Carry-Maskable Full Adder

CMA は桁上げをマスクすることで、次の桁上げ入力  $C_{in}$  への伝搬が抑制される。この結果、トランジスタの遷移回数が減少し、動的消費電力の削減につながる。本研究では、CMA をメインの近似加算器として採用した。

## 4. bit 幅選択とエラー訂正方式の検討

### 4.1 本研究における既知の課題

これまでに、CMA を適用することにより消費電力が増大することを確認している[6]。この現象が引き起こされた原因は以下の3つである。1) CMA は、キャリーをマスクしキャリー伝搬を抑制することで電力削減を実現している。キャリーをマスクすることによって、正確値より小さい値が出力されるため、エラーの大きさは負の方向に偏る[8]。2) 画像処理・音声処理といった信号処理のアプリケーションでは、入力値は、符号なし整数の一様分布に従うのではなく、符号あり整数の正規分布に従うことが知られている [9, 10]。つまり、絶対値の小さな値の入力が頻繁に入力される。[6]でも、DCT の入力値の絶対値は非常に小さな値であったことが判明している。3) 2.2 節の Loeffler DCT のフローチャートから見られるように、4 つのステージに分かれており、それぞれのステージの演算結果が次のステージに引き継がれる。

(1) ~ (3) の特徴を考慮すると、通常に加算時と比べて、JPEG アプリケーションに CMA を適用した場合、絶対値が小さい負数が含まれる加算割合が増加する。これにより、マスクすればするほど、負数の加算を実行する確率が増え、正確なパートでもキャリーが発生し、結果として予想しない形で電力増大という結果を引き起こした[6]。この問題を解決するために、4.3 節で Error Correction Scheme (ECS) を提案する。

一方で(2)の特性は、入力値の bit 幅を効率的に削減することに繋がる。これについて、次節で紹介する。

### 4.2 Park's Algorithm

DCT へ近似計算技術を適用する様々な提案がされてきた: DCT 処理のアルゴリズムの改良[11],  $\cos$  定数項のシフト演算化[12, 13] (Canonical Signed Digit (CSD)), Constant Multiplication Factor (CMF), 値, bit 幅の切り捨て[14], 近似加算器の適用[15-21]等である。本研究では、近似加算器と bit 幅の切り捨て[14]の2つの技術を組み合わせて低消費電力化を図る。Park らは、4.1 節で述べた(2)の特性に着目してそれぞれの DCT 係数に適切な bit 幅を割り当てる手法を提案した[14]。具体的には、圧縮画像に影響を与えにくい高周波成分の DCT 係数から順に bit 幅を削減することで画質を維持しつつ低消費電力化を達成している。また、Park らは閾値として PSNR を、34dB, 32dB, 30dB に設定し、閾値を設定しない場合も含めた計 4 つのレベルの bit 幅を提案している。本研究では、近似加算器の特性を最大限に生かすため、PSNR 閾値 34dB での bit 幅を選択した。

Loeffler DCT の加減算部分に近似加算器 CMA を適用し、低消費電力化を目指し、PSNR と SSIM の観点からその性能を評価する。

### 4.3 Error Correction Schemes

上述の問題を解決するために本報告で ECS を提案する。図 6 は、n-bit ECS である。ECS は、Error Detection Circuit (EDC) と Error Correction Circuit (ECC) から構成される。 $S'$  は元の近似和、Flag は制御信号、 $S$  は訂正後の近似和、 $i$  は訂正する近似和  $S'$  の最下位 bit 位置を示している。EDC の Flag 信号が検知されたとき (Flag = 0), 下位 bit から上位 bit にかけて訂正した和  $S$  を出力する。(a)は EDC を示している。ECC-I では、Flag 信号が検知されたとき  $i$  から最上位の和 bit までの値を強制的に 0 にする。それ以外の場合は、元の近似和を保持する。ECC-II では、Flag 信号が検知されたとき出力和は 0 に固定される。それ以外の場合は、元の近似和を保持する。本研究では、 $i=1$  ( $-4 \leq S' < 0$ ) と  $i=3$  ( $-16 \leq S' < 0$ ) における ECS の回路評価と JPEG 回路の評価を行った。 $i=1$  のときの ECC-I と ECC-II の状態をそれぞれ S1-I・II Scheme と、 $i=3$  のときの ECC-I と ECC-II の状態をそれぞれ S3-I・II Scheme と呼ぶことにする。

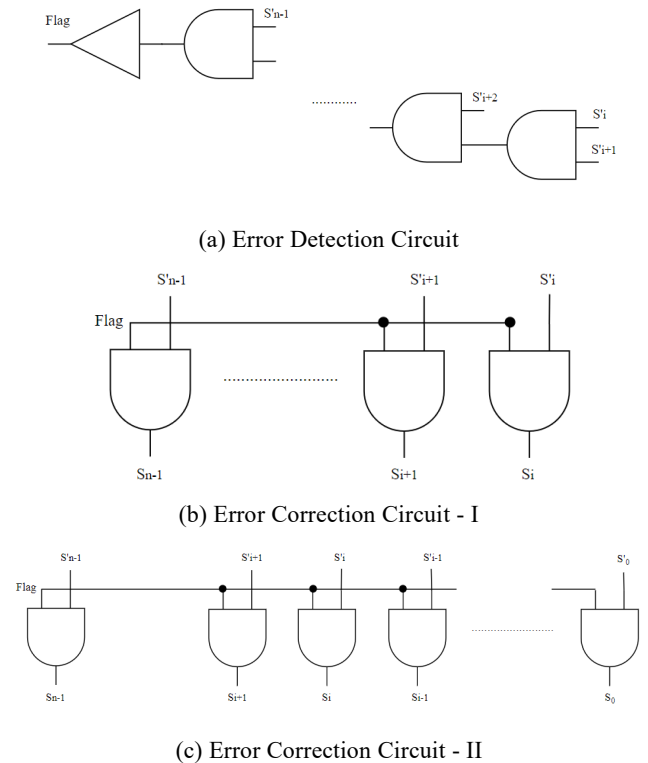


図 6 Error-Correction Schemes (ECS)

## 5. 実験

本節では、CMA を適用しながら bit 幅選択アルゴリズムと ECS を適用した場合の実験結果をそれぞれ報告する。5.1 節では実験方法について述べる。5.2 節で bit 幅選択アルゴリズムと CMA を適用した場合の画像評価と消費電力の結果を説明し、5.3 節で ECS 付き CMA を JPEG に適用した時

の画像評価と消費電力の結果について考察する。最後の5.4節で、CMAを代表的な近似加算器と比較する。

### 5.1 実験方法

本実験では、まずCMAとPark's Algorithmを適用した際の画像評価と消費電力の評価を行う。画像評価では、Loeffler DCTを使用し、C++言語で実装されているJPEG圧縮プログラム[22]を利用する。Loeffler DCT内部のstage0からstage3にかけての加減算を、C言語で実装したCMAによる近似加算に置き換えて画像を圧縮する。よって、近似演算を適用した部分は、図7の赤い部分である。また、4.2節で紹介したbit幅選択を上記のJPEG圧縮プログラム上に実装した。

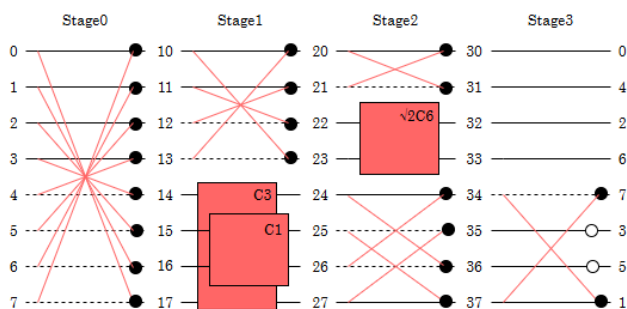


図7 近似加算の適用部分

圧縮前後の2画像を比較して画像の品質を評価する。評価画像には、255×255pixelのlena画像、349×349pixelのbaboon画像、512×512pixelのfl6画像とpepper画像を用いる。評価画像の指標としてPSNR (Peak Signal to Noise Ratio)、SSIM (Structural SIMilarity)を使用する。一般的に、使用されている評価目安を表1に示す。本研究では、PSNRの閾値を30dBに、SSIMの閾値を0.90に設定する。

表1 評価画像の指標[23]

PSNR	SSIM	主観評価
40dB以上	0.98以上	元画像と区別がつかない
30~40dB	0.90~0.98	拡大すれば劣化がわかる
30dB以下	0.90以下	明らかに劣化がわかる

消費電力の評価では、32-bit CMAとRCAをVerilog-HDLで実装し消費電力の評価を行い、比較する。VCSによるシミュレーションを実行し、その結果をもとにDesign Compilerで消費電力の評価をした。合成には45nm Nangateセルライブラリ[24]を利用した。シミュレーションの入力にはDCT内の加算の加数と非加数を与えた。

また、CMAの性能を確かめるために、他の近似加算器と比較した。比較対象として、代表的な近似加算器である、ACA-I [25]、ACA-II [26]、ETA-II [27]、GDA [28]、GeAr [29]、LOA [30]を採用した。ACA-I、ACA-II、ETA-II、GDA、GeAr、LOAをC言語で実装し、CMAと同様に図7の赤い部分に近似加算器を適用させ評価を行う。

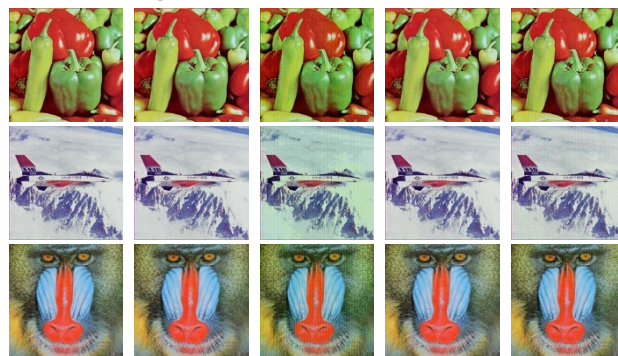
32-bit ACA-I、ACA-II、ETA-II、GDA、GeAr、LOAをVerilog-

HDLで実装し消費電力の評価を行う。ACA-I、ACA-II、ETA-II、GDA、GeArは[31]で公開されているオープンソースライブラリを元に、bit長を32-bitに拡張して実装した。それぞれの近似加算器の性能を最大限に引き出すため、[25-30]にしたがって表2の最適な回路構成パラメータを選んだ。それぞれの構成パラメータの意味は、[25-30]を参考にしてほしい。また、GDAに関してはマルチプレクサを用いて動的にPrevious bitを変更できるが、今回は、静的な設計にしている。

表2 回路構成パラメータ

	k (sub-adder)	Previous bit	Resultant bit	Apx Lower bit
ACA-I	4	4	1	-
ACA-II	4	2	2	-
ETA-II	4	2	2	-
GDA	4	8	4	-
GeAr	12	8	4	-
LOA	-	-	-	4

### 5.2 Park's Algorithm の評価結果



(a) (b) (c) (d) (e)

図8 評価画像

図8はCMAを適用したときの評価画像の結果である。(a)は通常のJPEG画像、(b)はPark's Algorithmを適用したJPEG画像、(c)はRow DCTとColumn DCTの両方に近似加算を適用させたときの画像、(d)と(e)はColumn DCTのみに近似加算を適用させたときの画像である。CMAの動的な近似精度調節機能を活用し、Row DCTのマスクbit数を0(すなわち正確な演算)に設定することでColumn DCTのみに近似演算を適用することを実現した。(c)と(d)はマスクbit数が4、(e)はマスクbit数7の時の評価画像を示している。評価画像の結果から分かるように、Row DCTに近似を適用することで縦のノイズが顕著に表れた。(e)の画像はColumn DCTにのみマスクbit数が7を適用したが、(c)よりもノイズは軽微である。よって、これ以降の結果はColumn DCTにのみ近似を適用した場合の結果を示す。

図9と10はPark's AlgorithmとCMAを適用したときのPSNRとSSIMの結果である。横軸はマスクbit数を、縦軸

はそれぞれ PSNR と SSIM 値を示している。凡例は評価対象の画像に対応している。PSNR に関しては、マスク bit が 6 以降になると閾値を下回る結果となった。SSIM に関しては、lena と pepper 画像において近似を適用しなくとも閾値を下回った。しかし、これらの画像は通常の JPEG 圧縮を施した場合でも SSIM 値が閾値を下回ることが確認されている。したがって、通常の JPEG 画像の SSIM 値と比較すると、Park's Algorithmn や CMA の適用による画像劣化は軽微であるといえる。

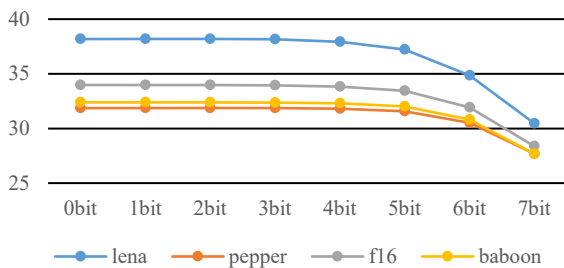


図 9 PSNR (Apx Col DCT)

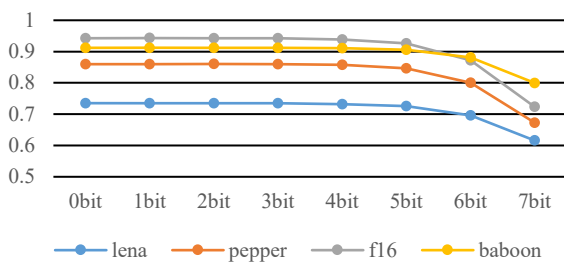


図 10 SSIM (Apx Col DCT)

続いて、Park's Algorithm と CMA を適用したときの消費電力について考察する。図 11 は、消費電力の結果である。縦軸が消費電力の大きさ、横軸はマスク bit 数を示している。消費電力は Park's Algorithm に 32-bit RCA を適用させたときの大きさと正規化している。また凡例は、各画像の消費電力の結果に対応している。どの画像も消費電力を削減することに成功した。先行研究[6]では電力が増大した lena も、Column DCT にのみ近似を適用させることで最大で約 3% 電力削減に成功した。

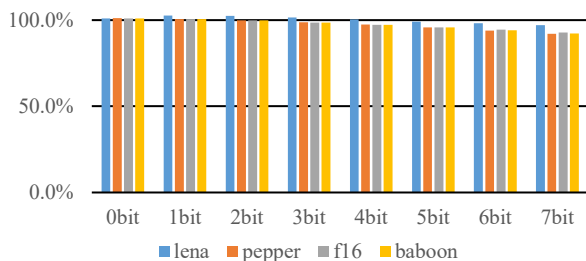


図 11 消費電力 (Apx Col DCT)

評価画像と消費電力の結果の結果より、Row DCT 内の入力に負数が多く含まれている可能性があることを示唆している。図 12 は、CMA の符号計算エラーの一例を示してい

る。この例では 16-bit CMA は上位 8-bit に正確な加算を行い、下位 8-bit に不正確な演算を適用する構成となっている。例えば 128 と -128 を加算する場合、正しい結果は 0 であるにもかかわらず、CMA では -128 が出力される。これにより画像のノイズが顕著に表れたのではないかと推測する。また、この負数の近似和は次の計算でも使用されるため、4.2 節で解説したような Carry 伝搬を引き起こし消費電力が増大した。

	XOR	OR
	00000000	10000000
	128 <sub>10</sub>	
+	11111111	10000000
	-128 <sub>10</sub>	
<hr/>		
	11111111	10000000
	-128 <sub>10</sub>	

図 12 Sign Calculation Error

しかし、Column DCT にのみ近似を適用することで[6]のような電力増大は見られなかったものの、近似適用による消費電力の増大は依然確認でき、RCA と比較して約 2% の電力増大を引き起こしている。

### 5.3 Error Correction Scheme

この節では、ECS を JPEG アプリケーションに適用した場合の性能と消費電力の評価を示す。ECSs を JPEG アプリケーションに適用し、画像評価と電力評価を行った。実験方法は、5.1 節と同じである。

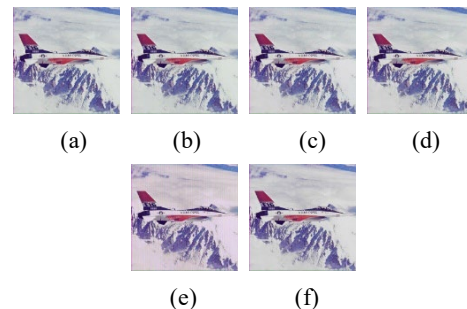


図 13 f16 画像

図 13 は、f16 画像の結果である。(a)は Park's Algorithm, (b)は従来の CMA, (c)~(f)はそれぞれ、S1-I, S1-II, S3-I, S3-II を CMA に適用したときの JPEG 画像を示している。なお(b)~(f)の CMA のマスク bit 数は 4 に設定されている。画像劣化は S3-I を除く方式で確認できなかった。また、これ以降は f16 の結果のみを示しているが他の 3 枚の画像でも ECS 適用による同様の効果が確認できた。

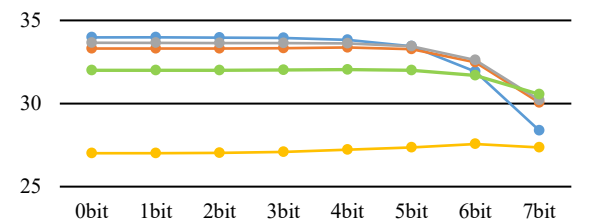


図 14 PSNR (f16)

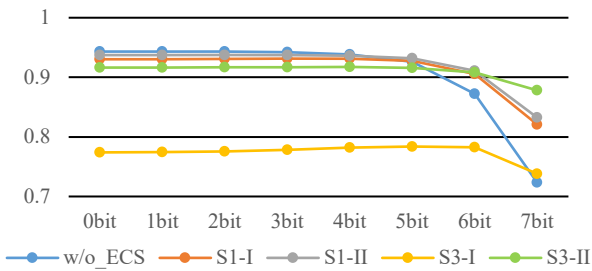


図 15 SSIM (f16)

図 14 と 15 は、f16 画像の PSNR と SSIM である。縦軸は PSNR または SSIM 値を、横軸はマスク bit 数をそれぞれ示している。凡例は誤り訂正方式に対応している。S3-I を除く方式ではマスク bit 数が 5 まで、ほとんど PSNR と SSIM の値は変化しない。マスク bit 数が 5 を超えると、PSNR と SSIM の値が減少し始めている。

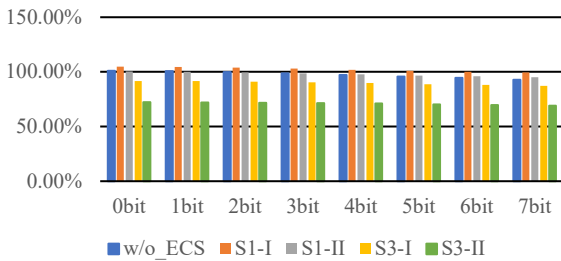


図 16 消費電力 (f16)

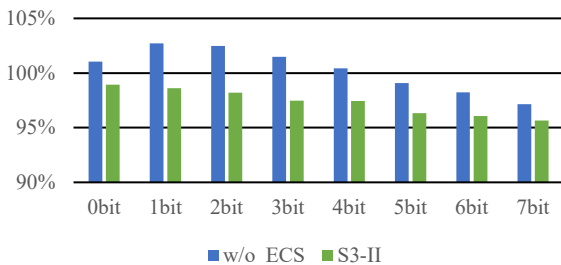


図 17 電力増大の消失 (lena)

図 16 は f16 画像の消費電力である。横軸はマスク bit 数を、縦軸は消費電力の大きさをそれぞれ示している。32-bit Ripple Carry Adder での消費電力で正規化している。どの画像の場合も ECS を適用することで、エラー訂正無し CMA と比較して消費電力を削減することに成功している。最大では約 30% の電力削減に成功している。

図 17 は 5.2 節で電力増大が確認された lena 画像の消費電力の結果を示している。ECS 無しの CMA では消費電力が増大する結果もあったが、S3-II 方式を採用することで、電力が増大するという現象は解消された。DCT 入力値が絶対値の小さな負数である割合が非常に高いことが予想できる。キャリーマスクによって、負数が出力されていたところを 0 に置き換えることによりキャリー伝搬が抑制され消費電力の削減につながったと思われる。ECS 適用により負数によるキャリー伝搬を抑制できた。

S3-I 方式でも、f16 画像の結果のように電力削減に成功している例もある。しかしながら、PSNR や SSIM の評価結果を考慮すると S3-II 方式を採用することが望ましい。

5.2 節と 5.3 節の結果から、Column DCT のみに近似を適用した方が望ましく、CMA に S3-II 方式を採用することで、RCA より消費電力を大幅に削減することができる。また、画質評価の結果を考慮すると CMA のマスク bit 数は 5 が望ましい。

#### 5.4 比較評価

CMA の性能を確かめるため他の近似加算器との比較評価も行った。図 18 は、f16 画像圧縮の結果である：(a) ACA-I, (b) ACA-II, (c) ETA-II, (d) GDA, (e) GeAr, (f) LOA, (g) CMA (Row と Column DCT に近似) (h) CMA (Column DCT に近似) (i) ECS 付き CMA (Column DCT にのみ近似)。LOA を除くすべての近似加算器で画像圧縮に失敗している。画像圧縮できたのは、LOA と CMA を適用した場合のみであった。

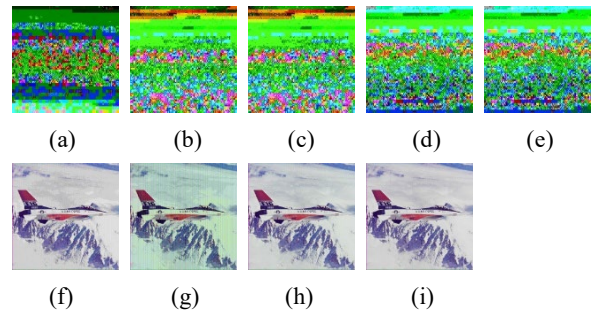


図 18 f16 画像

この原因は、これらの加算器が 2 の補数演算に対して脆弱性を有し、その回路構造に起因している。GeAr は ACA-I, ACA-II, ETAII を包括的に含む構造を持つ。GDA も Carry を予測するための bit 数をマルチプレクサで制限する。そのため、上記のような加算器は似たようなエラー挙動をする。そこで GeAr の回路構造に着目する。

GeAr 等の Carry 予測型加算器の設計における前提条件は、クリティカルパスのキャリーチェーンがめったに発生しないことである。しかし、この前提は符号なし整数が一様分布に従う場合を想定しており、JPEG のような符号あり整数の入力が正規分布に従う場合を考慮していない。

符号あり整数が絶対値の小さな正規分布に従う場合、最大キャリーチェーン長は、全体の加算器長とほぼ同じになる[32]。このような加算器全体のキャリーチェーンを GeAr は考慮しないため、遅延・消費電力、誤差距離の拡大など、設計上の理想性能から大きく乖離する結果を示した。

この現象は、絶対値の小さな異符号同士の加算で特に顕著に発生する。図 19 は、GeAr の  $P=8, R=4, k=12$  の時の回路構成による近似計算の一例である。整数  $A$  と  $B$  を加算するとき、それぞれの  $i$  番目の bit 位置の Carry 生成信号  $g_i$ , Carry 伝搬信号  $p_i$ , kill 信号  $k_i$  を



用し、低消費電力化を図った。CMA は、キャリアをマスクすることで計算精度を調整できる加算器であり、近似計算を行うことで RCA と比較して消費電力を削減できる。しかしながら、近似計算による出力値の誤差の影響で、CMA を適用した効果は限定的であった。この原因は、2 の補数演算に起因する計算エラーである。この問題を解決するためにエラー訂正回路 ECS を提案した。提案した ECS を CMA に適用することで、RCA と比較して画質を維持しつつ最大で約 30% の電力削減を達成した。また、CMA の性能を検証するために、代表的な近似加算器との比較研究も行った。この比較研究では、LOA や CMA 以外の近似加算器が JPEG 圧縮に失敗することが確認された。これは、それらの加算器が符号なし整数同士の加算を想定して設計されているためである。電力効率の評価では、LOA が比較対象の近似加算器の中で最も優れた電力効率を示した。しかし、画質評価を加味した場合、CMA に ECS を適用した方式が、画質と電力のトレードオフの点で最も優れた性能を発揮することが示された。

#### 謝辞

本研究の一部は、福岡大学の研究助成(助成番号:GR2410)および JSPS 科研費 20H00590 の助成を受けたものです。また本研究は、東京大学 VDEC 活動を通してシノプシス合同会社の協力で行われたものである。

#### 参考文献

- [1] H. Jiang et al., "Approximate arithmetic circuits: A survey, characterization, and recent applications," in Proceedings of the IEEE, Vol. 108, No. 12, 2020, pp. 2108-2135.
- [2] G. K. Wallace, "The JPEG still picture compression standard," in IEEE Transactions on Consumer Electronics, 1992, pp. xviii-xxxiv.
- [3] N. Ahmed et al., "Discrete cosine transform," in IEEE Transactions on Computers, 1974, pp. 90-93.
- [4] D. Mohapatra et al., "Design of voltage-scalable meta-functions for approximate computing," Design, Automation and Test in Europe, 2011, pp. 1-6.
- [5] T. Yang et al., "A low-power configurable adder for approximate applications," 19th International Symposium on Quality Electronic Design, 2018, pp. 347-352.
- [6] H. Hama et al., "Negative impact of approximated signed addition on power reduction," International Symposium on Devices, Circuits and Systems, 2023, pp. 1-6.
- [7] C. Loeffler et al., "Practical fast 1-D DCT algorithms with 11 multiplications," International Conference on Acoustics, Speech, and Signal Processing, Vol.2, 1989, pp. 988-991.
- [8] T. Ukezono, "Evaluations of CMA with error corrector in image processing circuit," in International Journal of Networking and Computing, Vol.9, No.2, 2019, pp.301-317.
- [9] A. Cilardo, "A new speculative addition architecture suitable for two's complement operations," Design, Automation and Test in Europe, 2009, pp. 664-669.
- [10] D. Esposito et al., "Approximate adder with output correction for error tolerant applications and Gaussian distributed inputs," IEEE International Symposium on Circuits and Systems, 2016, pp. 1970-1973.
- [11] Z. Wu et al., "An improved scaled DCT architecture," in IEEE Transactions on Consumer Electronics, Vol. 55, No. 2, 2009, pp. 685-689.
- [12] L. Cai et al., "Design of approximate multiplierless DCT with CSD encoding for image processing," IEEE International Symposium on Circuits and Systems, 2021, pp. 1-4.
- [13] B. Garg et al., "Energy scalable approximate DCT architecture trading quality via boundary error-resiliency," 27th IEEE International System-on-Chip Conference, 2014, pp. 306-311.
- [14] J. Park et al., "Dynamic bit-width adaptation in DCT: An approach to trade off image quality and computation energy," in IEEE Transactions on Very Large Scale Integration Systems, Vol. 18, No. 5, 2010, pp. 787-793.
- [15] M. Pashaeifar et al., "Approximate reverse carry propagate adder for energy-efficient DSP applications," in IEEE Transactions on Very Large Scale Integration Systems, Vol. 26, No. 11, 2018, pp. 2530-2541.
- [16] V. Gupta et al., "Low-power digital signal processing using approximate adders," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 32, No. 1, 2013, pp. 124-137.
- [17] R. Nayar et al., "Image compression using approximate addition," IEEE Region 10 Conference, 2021, pp. 1-6.
- [18] H. A. F. Almurib et al., "Approximate DCT image compression using inexact computing," in IEEE Transactions on Computers, Vol. 67, No. 2, 2018, pp. 149-159, 2018.
- [19] Y. Guo et al., "Design of power and area efficient lower-part-or approximate multiplier," IEEE Region 10 Conference, 2018, pp. 2110-2115.
- [20] M. Vasudevan et al., "Image processing using approximate datapath units," IEEE International Symposium on Circuits and Systems, 2014, pp. 1544-1547.
- [21] F. S. Snigdha et al., "Optimal design of JPEG hardware under the approximate computing paradigm," 53rd ACM/EDAC/IEEE Design Automation Conference, 2016, pp. 1-6.
- [22] <https://github.com/richgel999/jpeg-compressor/>, (参照 2025-2-6)
- [23] The Silicon Integration Initiative Inc. Open cell and free PDK libraries, <https://si2.org/open-cell-library/>, (参照 2025-2-6)
- [24] 小箱雅彦, "電子化文書の画像圧縮ガイドライン," 月刊 IM, Vol. 50, No.5, 2011, pp. 21-24.
- [25] A. K. Verma et al., "Variable latency speculative addition: A new paradigm for arithmetic circuit design," Design, Automation and Test in Europe, 2008, pp. 1250-1255.
- [26] A. B. Kahng et al., "Accuracy-configurable adder for approximate arithmetic designs," 49th ACM/EDAC/IEEE Design Automation Conference, 2012, pp. 820-825.
- [27] Ning Zhu et al., "An enhanced low-power high-speed Adder For Error-Tolerant application," the 12th International Symposium on Integrated Circuits, 2009, pp. 69-72.
- [28] R. Ye et al., "On reconfiguration-oriented approximate adder design and its application," IEEE/ACM International Conference on Computer-Aided Design, 2013, pp. 48-54.
- [29] M. Shafique et al., "A low latency generic accuracy configurable adder," 52nd ACM/EDAC/IEEE Design Automation Conference, 2015, pp. 1-6.
- [30] H. R. Mahdiani et al., "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," in IEEE Transactions on Circuits and Systems I: Regular Paper Vol. 57, No. 4, 2010, pp. 850-862.
- [31] <https://sourceforge.net/projects/approxadderlib/files/>, (参照 2025-2-6)
- [32] D. Esposito et al., "Variable latency speculative parallel prefix adders for unsigned and signed operands," in IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 63, No. 8, 2016, pp. 1200-1209.