CMA を用いた画像先鋭化処理専用回路の低消費電力化改善 *

佐　藤　寿　倫 **
ヨウ　ドウキン ***
請　園　智　玲 **

Improving Power Efficiency of CMA-based Circuit for Image Sharpening

Toshinori Sato**, Tongxin Yang***, Tomoaki Ukezono***

This paper revisits to build an approximate circuit for image sharpening, which has been investigated in our previous paper[1]. This is due to the unexpected results of [1], which is that the dynamic range of PSNR is relatively narrow. Using Carry-Maskable Adder, which was also used in [1], we implement an alternative circuit. Observing detailed simulation results unveils that there are not any significant differences between the two regarding area, delay, and power. In contrast, regarding quality of processed images, the two circuits have considerably different results. If the two circuits are compared under the constraint of same image quality, it is found that the implementation in this paper is more power-efficient than that in [1].

Key Words : approximate adder, variable accuracy, low power consumption

1　はじめに

本研究では先に [1] において，論理和で近似化をする加算器 CMA（Carry-Maskable Adder）[2] を，画像先鋭化回路に適用し，それによる消費電力の削減を評価した．ヒトの目で認識可能な精度を持ち，且つ，電力の削減が可能なマスク制御の設定を調査した．6 ビットマスクすることで，約 31% の電力削減効果を確認した．良好な結果と思われるが，出力画像のピーク信号ノイズ比 PSNR（Peak Signal-to-Noise Ratio）の可変範囲が小さいことが意外な結果であった．そこで本稿ではその理由を考察し，画像先鋭化回路の実装方法を見直し，その結果電力削減効果を改善した．

近似計算は，処理の高速化，消費電力の削減，そして回路の小型化を達成できる手法として，近年注目されている [3][4]．現在重要性を増しているアプリケーションの中には厳密に正確な値を必要としないものがあり，それらの処理において近似計算は非常に有効性が高い．画像や音声などのディジタル信号処理や，画像認識などの機械学習がそのようなアプリケーションの代表例である．[1] ではモノクロ画像の場合が紹介されている．モノクロ画像の 1 画素は 8 ビットの 256 階調で表現され，白から黒までの間の灰色を段階的に変化させる符号値となる．画像中で隣接する画素の階調は必然的に近い値となるため，1 画素の処理結果に多少の誤差が含まれても近傍の画素に近い値であれば，ヒトの目で誤差を認識することは困難である．上記のようなアプリケーションが重要性を増していることからも，近似計算は新たな計算パラダイムを開拓する手法として注目に値する．

近似計算はコンピュータ処理における様々な抽象度で利用可能であるが，本稿および [1] では論理回路レベルでの利用を検討した．対象として，画像先鋭化処理 [5] を選んだ．以下の構成は次のとおりである．まず次節で画像先鋭化処理を説明し，3 節でその実装を述べる．4 節で CMA を紹介する．5 節で評価結果を述べ，最後に 6 節でまとめる．

2　画像先鋭化処理

画像先鋭化処理は，元の画像の濃淡を残したままでエッジを強調する．本研究では，元画像とその平滑化処理結果を用いる先鋭化処理 [5] を選択する．これは以下の 2 ステップで処理される．

（ステップ 1）元画像 I に対して平滑化処理を施す．以

* 平成 30 年 11 月受付
** 電子情報工学科
*** 情報・制御システム工学専攻

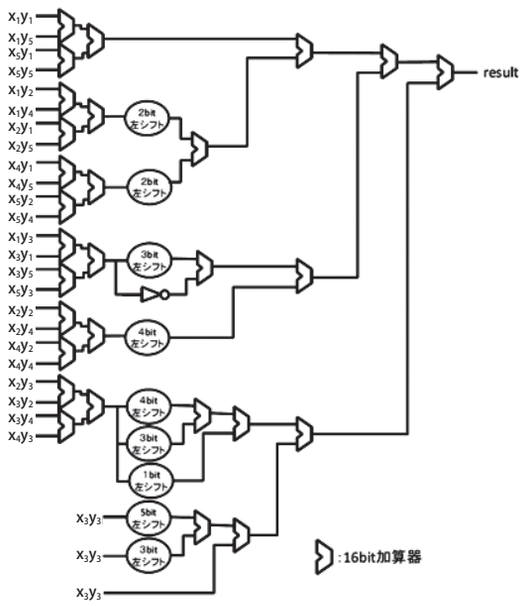


図 1: 畳み込み演算回路[1]

下のガウシアンカーネル G を用いて畳み込み演算によりフィルタ処理を実施する．

$$G = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

平滑化処理後の画素値 R(x,y) は以下により計算される．

$$R(x,y) = \sum_{i=-2}^{2} \sum_{j=-2}^{2} G(i+3, j+3) \cdot I(x+i, y+j)$$

（ステップ 2）元画像 I の 2 倍から平滑化処理後の画像 R を引き算する．鮮鋭化処理後の画素値 S(x,y) は以下により計算される．

$$S(x,y) = 2 \cdot I(x,y) - R(x,y)$$

3　近似回路

本研究ではステップ 1 における畳み込み演算を近似回路化している．乗算は，処理の高速化，消費電力の削減，そして回路の小型化を同時に満足することが困難な処理である．乗算においては，多数の部分積を足し合わせることで最終的な積が得られる．処理を高速化するためには並列に多数の加算をすることが有効であり，必然的に回路規模の増大，ひいては消費電力の増大を引き起こす．逆に，消費電力の削減や回路の小型化を追求しようとすれば，処理の高速化を望むことはできない．本研究では，乗算の近似化により専用回路化の際の消費電力を削減している．

ディジタル回路では数値は二進数表現されていることが一般的である．その場合，乗数を 2 の冪乗値に限定すれば，乗算を容易に処理することが可能である．上位桁方向（左方向）へのシフト処理は 2 の冪乗値を乗数とする乗算と等価である．例えば，2 ビット左シフトは× 4 と，4 ビット左シフトは× 16 と，それぞれ等価である．

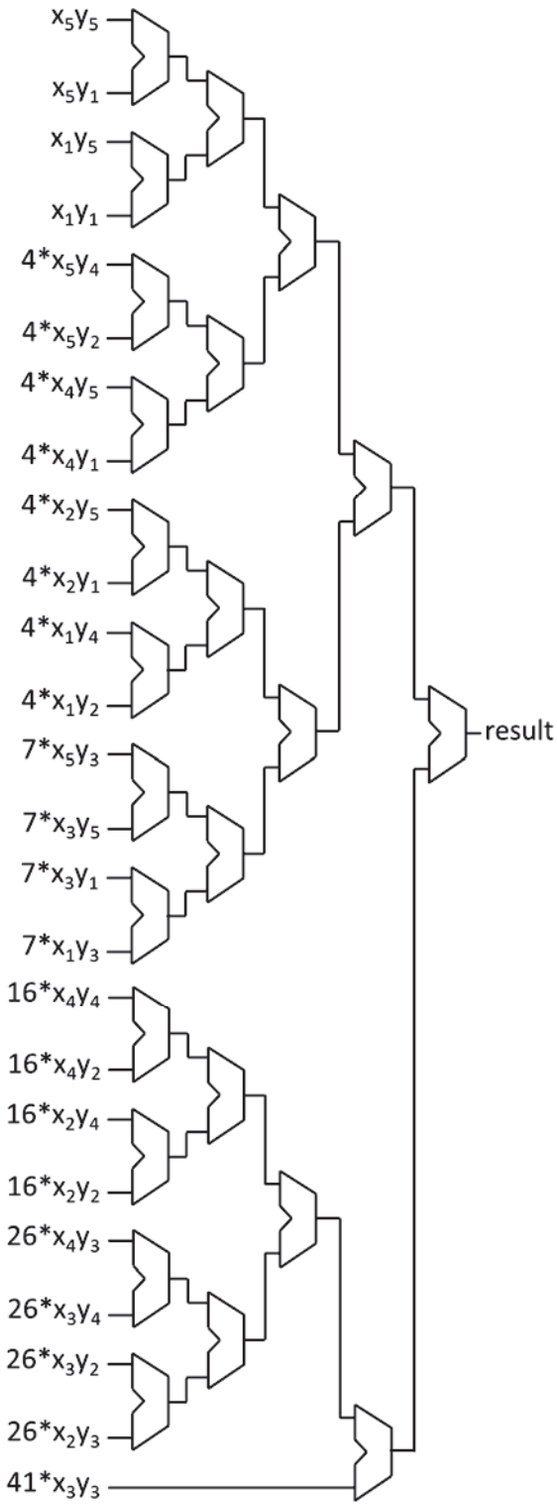


図 2: 演算順序を変えた畳み込み演算回路

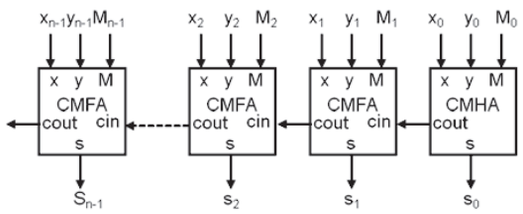
全ての定数乗算を加算とシフト処理に置き換えたのち，加算を CMA で演算することにより近似化している．

図 1 に示すように [1] では，最初に重みが同じである画素値を足し合わせ，続いて重みを掛け，最後にそれらの総和を求めている．全ての加算が近似演算であるため，積算の順序により最終結果が異なる．本稿では，図 2 に示すように，先に全ての画素値と重みとの積を求めた後に，それらの総和を積算するように変更した．図中の 4 と 16 の定数乗算はシフト処理のみで実現できる．26 と 41 の定数乗算は [1] と同様に，16，8，そして 2 の定数乗算の和（16+8+2=26），32，8，そして 1 の定数乗算の和（32+8+1=41）として，それぞれ実現している．8，2，32 の定数乗算もシフト処理で実現可能である．7 の定数乗算のみ [1] での実装方法と異なる．[1] では 8 の定数乗算と元の画素値の差（8 － 1=7）として実現したが，本稿では 4，2，そして 1 の定数乗算の和 (4+2+1=7) として実現している．減算には 2 の補数を用いるのではなく，1 の補数を用いて近似している．後述するように CMA は減算を不得意とするため，差よりも和を利用することが望ましいと思われる．

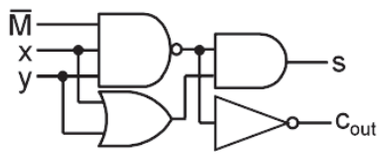
4 Carry-Maskable Adder

加算器は基本的な演算器であり，これまでに様々な近似加算器が提案されているが，それらのほとんどは演算精度が固定のものである．画像によって必要となる演算精度が異なるだろうという予想に基づき，本研究では [2] で提案した演算精度を変更可能な加算器 CMA を利用する．既存の精度可変な加算器では回路規模が大きくその結果消費電力も大きくなるということが課題であるが，提案する加算器はその課題を解決している．

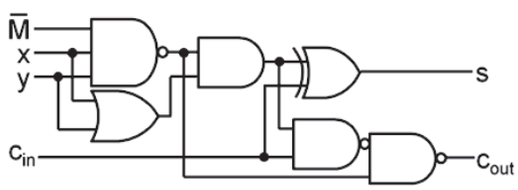
図 3 に CMA と，それを構成する CMHA（Carry-Maskable Half Adder）および CMFA（Carry-Maskable Full Adder）を示す．まず CMHA を説明する．x と y は被加数および加数，s と c_{out} は和および桁上げ出力，そして M が桁上げマスク信号である．M がアサートされる（$\overline{M}=0$）と，s は x と y の論理和（$s = x \cup y$）になり，c_{out} は 0 に固定される．桁上げ出力が 0 固定であるから，桁上げ伝搬がマスクされる．逆に M がネゲートされる（$\overline{M}=1$）時は正確なハーフアダーとして動作する．次に CMFA を説明する．c_{in} は桁上げ入力である．M がアサートされる（$\overline{M}=0$）時は，c_{in} は 0 になっていると仮定する．この時は CMHA と同様で，s は x と y の論理和（$s = x \cup y$）になり，c_{out} は 0 に固定される．桁上げ出力が 0 固定であるから，桁上げ伝搬がマスクされる．逆に M がネゲートされる（$\overline{M}=1$）時は正確なフルアダーとして動作する．最後に，ひとつの CMHA と (n－1) 個の CMFA を直列に接続した n ビット CMA を説明する．CMA ではあるビット位置を境にして，下位側のビットを全てマスクし（$\overline{M}=0$），上位側のビットはマスクしない（$\overline{M}=1$），



(a) n ビット CMA



(b) CMHA



(c) CMFA

図 3: Carry-Maskable Adder

という利用を前提とする．こうすることで下位側の全ての CMFA で c_{in} と c_{out} が 0 になり，それらは桁上げの無い OR ゲートとして動作する．一方，上位側の全 CMA は全加算器として動作する．つまり，上位ビットと比較して下位ビットは重要度が小さいという特徴を利用した近似的な加算を行っている．

現在のディジタル回路は CMOS 回路として実現されるのが一般的である．CMOS 回路が消費する電力の主な要因は，出力信号の変化である．CMA は桁上げ信号をマスクすることで出力信号が変化する確率を低減している．こうして，加算における消費電力を削減出来るわけである．畳み込み演算に現れる加算を全て CMA で実装することで，画像処理における消費電力を削減できる．CMA は桁上げを遮断するビット数を変更可能であるので，処理される画像が要求する演算精度に合わせてマスク信号を設定することで，理想的には必要最小限の消費電力で処理可能である．

5 評価

3 節で説明した畳み込み演算の近似回路を利用する画像先鋭化回路を，シミュレーションにより評価する．処理後の画像品質，回路の消費する電力，処理の遅延時間，そして回路の面積の観点で，[1] の結果と比較する．

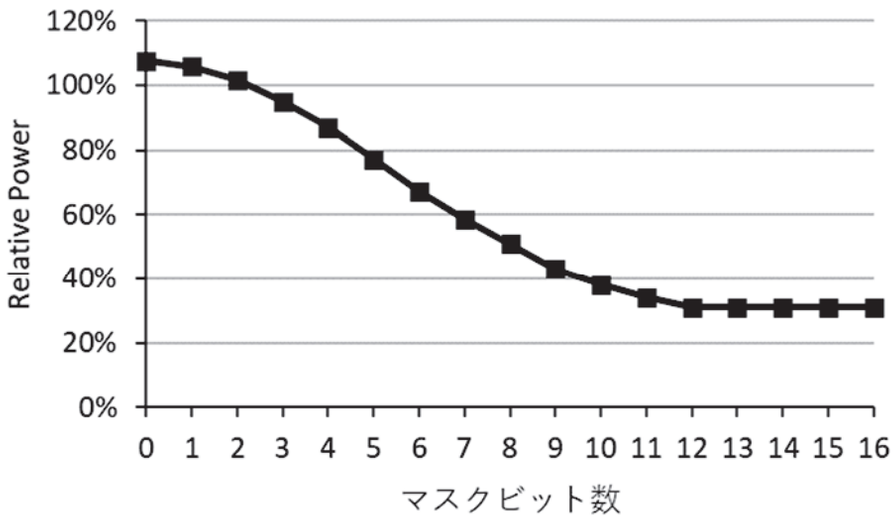


図 4: 消費電力

5.1 評価環境

Verilog HDL を用いて上述した実装を回路記述する．Synopsys 社の DesignCompiler を用いて NanGate 45nm のオープンセルライブラリ [6] で論理合成し，回路規模と遅延時間を見積もる．Mentor Graphics 社の ModelSim で論理シミュレーションを実施し，信号値の変化を VCD ファイルとして獲得する．VCD ファイルから変換された SAIF ファイルを DesignCompiler に与えることで消費電力を見積もる．

処理対象は，サイズが 512 × 512 で 256 階調グレイスケールの Lena 画像である．出力画像の品質は PSNR で評価する．PSNR は以下：

$$PSNR = 10 \cdot \log_{10}\left(\frac{As_{max}^2}{MSE}\right)$$

$$MSE = \frac{1}{X \cdot Y}\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}\{S(x,y) - \tilde{S}(x,y)\}^2$$

で計算され，40dB を超えれば高品質として，20dB よりも低い場合には受け入れ難いとして扱う [7]．上式で，As_{max}，S(x,y)，そして$\tilde{S}$(x,y)は先鋭化処理後画素の最大値，各画素の正確な値，そしてその近似値をそれぞれ表す．X と Y は画像の横縦サイズである．

5.2 結果

桁上げ伝搬加算器 CPA（Carry Propagation Adder）を利用して正確な演算を行う画像先鋭化回路と比較し，評価結果を示す．

回路面積は 15.7% 増加したが，遅延時間は 6.7% 改善された．面積の増加は予想通りであるが，CPA よりも高機能な CMA を使用しているにも関わらず遅延時間が改善している点が興味深い．一般的な全加算器では二つ直列接続された排他的論理和（XOR）ゲートを持つことに対し，CMFA は XOR ゲートをひとつしか持たないことが理由と考えられる．XOR ゲートでは NAND などの基本ゲートの凡そ 2 倍の遅延時間となるからである．[1] での結果は，回路面積が 13.6% の増加であり，遅延時間が 4.3% の改善であった．顕著な違いはなく，回路面積と遅延時間の観点では，本稿の回路と [1] でのそれとは同等である．

図 4 に消費電力を示す．縦軸は正確な画像先鋭化回路で正規化された消費電力である．値が 100% よりも小さければ改善されている．横軸はマスクされるビット数である．マスクされるビット数が 2 ビットまでは正確な回路の電力よりも大きな電力を消費するが，それ以降は着実に減少していき，12 ビット付近で 31.0%にまで削減されて飽和する．桁上げ伝搬をマスクすることで信号の変化確率が低減されていることが確認できる．[1] でも同様に，マスクされるビット数が 2 ビットまでは正確な回路の電力よりも大きな電力を消費するが，それ以降は着実に減少していき，13 ビット付近で 31.4%にまで削減されて飽和する．消費電力の観点でも，両者に顕著な差は無いと言える．

図 5 に PSNR をまとめる．桁上げ伝搬をマスクするビット数が大きくなるにしたがい，PSNR が悪化している．5 ビットまでは高品質と見做される 40dB 以上を満足しているが，8 ビットを越えると受け入れがたいとされる 20dB を下回る．図 6 に処理前後の画像を示す．(a) は元画像で，(b) は正確な演算回路での先鋭化処理結果である．(c-1) ～ (c-16) は近似計算による先鋭化画像で，数

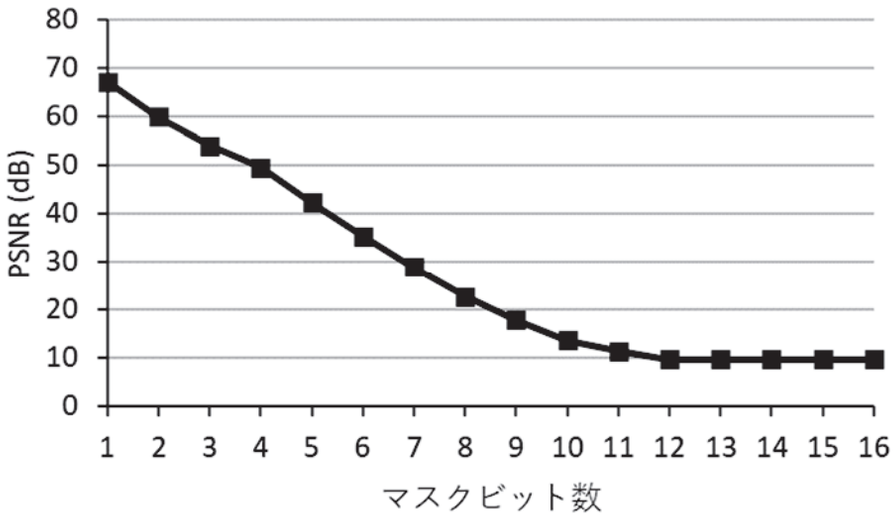


図 5: PSNR

字はマスクビット数である．PSNR が 40dB 以上となる 5 ビットマスクまでは十分満足できる画質である．ヒトの目では正確な処理画像と顕著な違いを識別できない．30dB を下回る 7 ビットマスクの場合でも，画像の違いは明らかであるが，著しく品質が低いとは言えず，客観的な結論では無いが十分受け入れることが可能である．20dB を下回る 9 ビットマスクの場合では，確かに受け入れがたい画質の悪化が見られる．

PSNR については，[1] での結果と本稿とのそれに大きな違いが見られる．まず可変範囲が異なる．[1] でのそれは凡そ 28dB ～ 55dB であるが，本稿では 9.62dB ～ 67.16dB であり，レンジに 2 倍以上の違いがある．PSNR が 40dB あるいは 30dB を越えるマスクビット数にも違いがある．[1] ではそれぞれ 3 ビットと 4 ビットであるのに対し，本稿ではそれぞれ 5 ビットと 6 ビットである．ヒトの目で許容できる限界も，[1] では 6 ビットであるのに対し本稿では 8 ビットである．いずれの場合も [1] と比較して本稿の方でマスクビット数が多く，消費電力の結果に顕著な違いが無かったことから，同等の画像品質で比較すると本稿の実装の方が消費電力を大きく削減可能である．

出力画像に関しても，両者に大きな違いがある．12 ビットマスクの場合，[1] では画像につぶれやにじみが見られるため写真品質とは言い辛いが，本稿の結果では画質は劣っているにせよ十分に写真と認められる．PSNR は前者では 30dB に近い値であり後者では 10dB を下回っているにも関わらず，ヒトの目には後者の方が良質な画像に見える点が興味深い．16 ビットマスクの場合には，[1] ではネガのような画像になってしまっているが，本稿の結果では 12 ビットマスクの場合と大差ない．ネガのような画像になる原因は，[1] では 1 の補数表現で近似した減算を採用していることであると思われる．

6 おわりに

近年近似計算が注目されており，[1] で近似加算器を用いて画像先鋭化回路を実装した．その評価結果において PSNR の可変範囲が小さいということが意外であったため，本稿で畳み込み演算回路の実装方法を再検討し，改めて [1] と同様の評価を行った．

回路面積，遅延時間，そして消費電力の観点では，[1] での実装回路と本稿でのそれとの間に顕著な違いは無かった．ところが出力画像の品質の指標となる PSNR で比較すると，両者には大きな違いが見られた．同等の画像品質で比較すると，[1] とよりも本稿でマスクビット数が大きくすることが可能で，本稿での実装の方が消費電力を大きく削減可能である．同じ CMA を使用しているにもかかわらず顕著な差が見られることから，近似回路の実装には精度の観点で演算順序に細心の注意が必要であると言える．

図 6: 元画像と先鋭化処理後の画像

謝辞

本研究は，JSPS 科研費 JP17K00088 の助成を受け，その一部は福岡大学研究推進部の研究経費（課題番号：175007，177005）による．東京大学大規模集積システム設計教育研究センターを通し，シノプシス株式会社の協力で行われた．

参考文献

1. 田島加織 , 井上晶仁 , 馬場裕之 , ヨウドウキン , 請園智玲 , 佐藤寿倫 , “CMA を用いた画像先鋭化回路の低消費電力化 ,” 信学技報 VLD2017-68, 2018 年 .
2. T. Yang, T. Ukezono, and T. Sato, “A Low-Power Configurable Adder for Approximate Applications,” International Symposium on Quality Electronic Design, 2018.
3. J. Han and M. Orshansky, “Approximate Computing: An Emerging Paradigm for Energy-Efficient Design,” European Test Symposium, 2013.
4. V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, “Analysis and Characterization of Inherent Application Resilience for Approximate Computing,” Design Automation Conference, 2013.
5. M. S. K. Lau, K.-V. Ling, and Y.-C. Chu. “Energy-aware Probabilistic Multiplier: Design and Analysis,” International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, 2009.
6. NanGate Inc., FreePDK45 Open Cell Library, http://www.nangate.com/
7. D. Bull, “Communicating Pictures,” - Academic Press, 2014.