博士論文

ユーザサポート・ハザード対策を目的とした組込み機器向けプロセッサの研究
(A Study on Content Addressable Memory-based Massive-parallel SIMD Matrix Core for User Support and Hazard Prevention)

2022年3月

立命館大学大学院理工学研究科
電子システム専攻博士課程後期課程

蔭山　享佑

立命館大学審査博士論文

ユーザサポート・ハザード対策を目的とした組込み機器向けプロセッサの研究
(A study on Content Addressable Memory-based Massive-parallel SIMD Matrix Core for User Support and Hazard Prevention)

2022年3月
March 2022

立命館大学大学院理工学研究科
電子システム専攻博士課程後期課程
Doctoral Program in Advanced Electrical,
Electronic and Computer Systems
Graduate School of Science and Engineering
Ritsumeikan University

蔭山　享佑
KAGEYAMA Kyosuke

研究指導教員：熊木　武志　教授
Supervisor : Professor KUMAKI Takeshi

目次

1. 主論文

ユーザサポート・ハザード対策を目的とした組込み機器向けプロセッサの研究

(A Research of Content Addressable Memory-based Massive-parallel SIMD Matrix Core for User Support and Hazard Prevention)

蔭山　享佑

2. 副論文

(1) 超並列 SIMD 型演算プロセッサコア MX-1 を利用したモルフォロジカルパターンスペクトラムの並列処理について.
蔭山享佑, 小出哲士, 熊木武志
電気学会論文誌, **139** (3), 237–246 (2019).

(2) Floating-point arithmetic of content addressable memory-based massive-parallel SIMD matrix core.
Kyosuke Kageyama, Akimitsu Hamai, Kensuke Watanabe, Tetsushi Koide and Takeshi Kumaki
RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, 274–277 (2021).

(3) Digital image forensics using morphological pattern spectrum.
Kyosuke Kageyama, Takeshi Kumaki, Takeshi Ogura and Takeshi Fujino
Journal of Signal Processing, **19** (4), 159–162 (2015).

(4) 陸上変温動物を対象としたノーマリオフバイオロギングデータロガーの開発.
蔭山享佑, 大井崇広, 熊木武志
電子情報通信学会論文誌, **J104-D** (4), 415–426 (2021).

3. 参考論文

(1) Proposal of content addressable memory-based massive-parallel SIMD matrix core.
Kyosuke Kageyama, Akira Sekino, Kensuke Watanabe, Akimitsu Hamai, Tetsushi Koide and Takeshi Kumaki
2020 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'20), 77–80 (2020).

(2) Acceleration of arithmetic processing with CAM-based massive-parallel SIMD matrix core.
Kyosuke Kageyama, Kensuke Watanabe, Akimitsu Hamai, Tetsushi Koide and Takeshi Kumaki
2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), 486–489 (2020).

(3) Human-like image manipulation detection using morphological pattern spectrum.
Kyosuke Kageyama, Takeshi Kumaki, Takeshi Ogura and Takeshi Fujino
2015 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'15), 591–594 (2015).

(4) Structuring Element-counting Approach for Morphological Pattern Spectrum-based Image Manipulation Detection.
Kyosuke Kageyama, Tetsushi Koide and Takeshi Kumaki,
2019 2nd International Symposium on Devices, Circuits and Systems (ISDCS) (2019).

(5) 1/f fluctuation-based visible light beacon for Spy-photo Prevention System.
Kyosuke Kageyama, Kohei Sugiyama, Takeshi Kumaki, and Takeshi Fujino,
2016 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'16), 686–689 (2016).

主論文

内 容 梗 概

近年，半導体技術の急速な発展に伴い，スマートフォンを始めとした様々なモバイルデバイスが普及してきている．一般に，半導体技術はムーアの法則に従って，その集積率は 18 か月で 2 倍に向上する．すなわち，半導体により構築される集積回路を多用しているモバイルデバイスは益々高性能かつ小型になっている．その結果，スマートフォンはより身近なものになり，多機能化したことでより便利になり，普及率は急上昇している．当初，スマートフォンや携帯電話は通話や電子メールの機能が重要視されており，通信機器としての役割がほとんどであった．しかし，近年では，動画像編集・視聴，ゲーム，ソーシャルネントワーキングサービスの利用等，娯楽としての役割も担っている．これらの利用目的を実現するためには膨大なデータをリアルタイムに処理する必要があり，より高性能なプロセッサが組込まれている．さらに，スマートフォンを生活の様々な場面で活用するために，搭載センサの種類も増えてきている．例えば，通信機能の面では，通信速度が 1990 年代の 100 Kbps から 2020 年代には 10 Gbps まで向上している．カメラ機能の面では，イメージセンサの画素数は 2000 年の 10 万画素から現在の 1,500 万画素以上まで向上している．このため，スマートフォンのユーザは容易に動画像の撮影，編集及びインターネットへの投稿ができるようになっている．近年のプロセッサは，1 個のプロセッサで様々な処理を実行する “ディジタルコンバージェンス” という考えが一般的である．この考えの基，プロセッサには CPU とは別に DSP，GPU，NPU 等，特定の処理に特化したアクセラレータを組込むことが主流となっている．一方，この様な特定の処理に特化したアクセラレータを複数組込むことで，プロセッサの面積が増加し，モバイルデバイスが大型化する課題がある．すなわち，今後，さらに膨大なデータ量及び演算量を処理する必要があり，特定の処理に特化した複数のアクセラレータを組込むのではなく，汎用性を持った一つのアクセラレータを組込むことで，回路規模を低減しつつ高性能なプロセッサを実現することが重要である．以上の背景から，本論文ではモバイルデバイス向けの汎用性を持った高性能なアクセラレータの研究を行う．

一方，モバイルデバイスが高性能化したことで，社会課題や犯罪におけるモバイルデバイスの利用が問題となっている．例えば，スマートフォンに搭載されているイメージセンサが高画質になったことで，一般のユーザが自由に場所や時間に関係なく動画像を撮影できるようになり，被写体に許可なく隠し撮りする盗撮という新たな犯罪が発生した．また，スマートフォン上で動画像を容易に編集できるようになり，ディープフェイク処理，文書改ざん等を気軽にスマートフォンで

行い，世界に影響を与えるような社会問題が発生した例もある．本論文では，この様なモバイルデバイスを利用することで発生する社会課題や犯罪をソーシャルハザードと呼び，この対策手法及びシステムを提案する．ソーシャルハザードが発生している一方で，さらなる小型化，高性能化が進み，日常生活で利用できるユーザサポート用のウェアラブルデバイスも普及してきている．ユーザのサポートが可能なウェアラブルデバイスは常に身に付けるデバイスであり，日常生活に影響を与えないよう，スマートフォンよりも小型かつリアルタイムな動画像処理等が要求される．ウェアラブルデバイスを利用し，人間の体調管理や動物の生態を解明する様なユーザサポートの研究は活発化しており，小型かつ高性能を維持しつつ，低消費電力なデバイスが必要となっている．すなわち，モバイルデバイスにおけるソーシャルハザードとユーザサポートは表裏一体の関係にあり，今後はモバイルデバイスのユーザサポートに向けた性能向上に併せてソーシャルハザードに対する対策が必要である．

以上の背景や課題に対して，本論文では汎用性のあるモバイルデバイス向けのアクセラレータであり，CPU では処理が難しい膨大なデータを並列に実行可能な連想メモリベース超並列 SIMD 型演算コア（Content Addressable Memory-based massive parallel matriX core: CAMX）を提案し，その処理性能を検証する．CAMX はメモリサイズを自由に調整可能な連想メモリで構成されており，左右の連想メモリの間に小型の演算器を挟み込んだ構造となっている．本論文では，左右の連想メモリのサイズは横方向に 128 ビット又は 256 ビットとし，縦方向に 1,024 エントリとして検証を行った．さらに，演算器は小型かつ単純な構成とするため，主に論理回路と加算器のみで構成している．CAMX はビットシリアル・ワードパラレルの処理方法を採用しており，演算はパイプライン処理を実現することで，高並列な処理を実行可能としている．CPU と CAMX はシステムバスでつながっており，CPU は逐次処理を実行し，並列に処理可能なデータは CPU から CAMX に処理を委託することとなる．特に，CAMX は連想メモリに格納しているデータに対して検索処理が容易かつ並列に実行できるため，検索データと一致した内部のデータのみを書き換えることを高速に実行可能できる．すなわち，CAMX はマルチメディア処理において必要な繰り返し演算処理の並列処理だけでなく，一般的に並列処理が困難であるテーブルルックアップ処理に対しても，並列かつ高速に実行することが可能である．

本論文では第 1 に，CAMX の有効性を検証するために基本演算命令（AND，OR，XOR，加算）及び検索命令についてシミュレーションを行った．それぞれ 128 ビットデータに対して実行した結果，1,024 エントリの全データが正確かつ同時に演算が行われ，基本演算命令は約 130 クロックサイクル，検索命令は 1 クロックサイクルで実行可能であることを確認した．次に，マルチメディア処理等において求めら

れる符号付き乗算処理についても検証を行った．CAMX の演算器は小型かつ単純であることを目指しているため，乗算処理の専用回路は組込んでいない．このため，乗算処理を実行するには，論理演算と加算を組合わせて行う必要がある．本論文では，CAMX における最適な乗算処理手法を検証するため，ビットシリアル乗算，検索加算繰り返し乗算及び Baugh-Wooley 乗算について検証を行った．ビットシリアル乗算は，検索命令を利用せず，乗数と被乗数を 1 ビットずつ加算していく一般的な演算手法である．検索加算繰り返し乗算は，検索命令をと加算命令を繰り返し実行する演算手法である．Baugh-Wooley 乗算は，Baugh-Wooley 演算の公式を利用した演算手法である．まず，ビットシリアル乗算及び検索加算繰り返し乗算を比較するため，4 ビット符号付き乗算を 1,024 エントリに対して実行したところ，ビットシリアル乗算は 1,462 クロックサイクルを要したが，検索加算繰り返し乗算は 237 クロックサイクルであった．すなわち，4 ビット符号付き乗算において，検索加算繰り返し乗算はビットシリアル乗算よりもクロックサイクル数を約 84 %削減できることが確認した．4 ビット乗算において大幅にクロックサイクル数を削減できることから，乗算ビット幅をさらに大きくした場合でも検索加算繰り返し乗算はビットシリアル乗算よりも高速に実行できる．次に，検索加算繰り返し乗算と Baugh-Wooley 乗算を 4 ～ 32 ビット符号付き乗算として 1,024 エントリに対して実行したところ，15 ビット以上の符号付き乗算では Baugh-Wooley 乗算を用いた方が高速に処理可能であった．すなわち，CAMX においては，15 ビット未満の符号付き乗算を実行する場合は検索加算繰り返し乗算により実行し，15 ビット以上の符号付き乗算を実行する場合は Baugh-Wooley 乗算により実行した方が高速に処理可能である．次に，暗号化手法の一つである AES を実装し，CAMX の基になったマトリクスアーキテクチャ型超並列演算プロセッサ MX-1 及び既存のモバイルデバイス向けプロセッサとの性能比較を行った．CAMX と MX-1 の 1 ラウンド当たりのクロックサイクル数について比較した結果，ShiftRows と MixColumns の合計については CAMX が MX-1 よりも約 65 %，AddRoundKey については約 86 %も削減できることが確認できた．既存のモバイルデバイス向けプロセッサの性能比較では，CAMX の方が約 53 %も処理時間が少ないことが確認できた．次に，浮動小数点による加算処理も実装し，その検証を行った．2 の補数削減処理による単精度浮動小数点加算処理とすることで高性能に実行可能であり，ARM コアと 1.5 GHz の動作周波数を想定した CAMX を比較したところ，処理データ数が 4,500 を超えると CAMX の方が高性能であることを確認した．

第 2 に，CAMX を組込むことを想定し，ソーシャルハザードに対する対策手法及びシステムを提案し，その効果を検証した．まず，画像改ざん検知手法であるモルフォロジカルパターンスペクトラム処理を提案し，モバイルデバイスの環境で検証可能な MX-1 に実装し，その効果を検証した．モルフォロジカルパターン

スペクトラムは対象画像に対して繰り返しオープニング処理や減算処理を実行するため，処理量が大きい手法である．すなわち，この手法の処理を検証することで，様々なマルチメディア処理に対応可能であることを確認できる．この処理を総合開発環境である High-performance Embedded Workshop (HEW) シミュレータ及び評価ボードにて処理可能であることを確認し，HEW シミュレータにおいてはユーザマイクロコードを生成することにより約 5 倍の処理速度向上が可能であることを確認した．さらに，MX-1 と既存のプロセッサにおいて処理性能を比較したところ，MX-1 は高性能に実行可能であることを確認できた．また，盗撮防止システムについても構築し，その効果を検証した．本システムは LED 照明とスマートフォンの連携によりスマートフォンを操作する手法であり，スマートフォンのイメージセンサから取得した動画像に対してリアルタイムに膨大なデータを処理する必要がある．LED 照明とスマートフォンの連携のための最適な信号である 1/f ゆらぎに正弦波を組込んだトリガを提案し，この信号に対して周波数分解を行うことでトリガの正弦波を抽出する手法について検証した．この結果スマートフォンにおいてトリガを検出し，スマートフォンのイメージセンサを停止させる等の操作が可能であることを確認した．

第 3 に，モバイルデバイスのさらなる進展を見据え，ユーザサポート用のウェアラブルデバイスに CAMX を組込むことを想定した検証も実施した．近年，動物の生態を解明するために，動物にモバイルデバイスを直接取り付けて行動を調査する研究が行われている．この研究で必要となるバイオロギング用データロガーにおけるノーマリオフ技術による低消費電力化を検証した．この結果，動物の行動に合わせたノーマリオフ動作によるデータロガーを用いることで，一般的に販売されている製品を用いて対象動物の約 1 日間の生態を検証することが可能となった．

目 次

図 目 次

表 目 次

第1章　序論

近年，半導体技術が急速に発展しており，併せて半導体を組込まれたモバイルデバイスの小型化，高性能化が進んでいる．このため，一般のユーザでも気軽に動画像の編集や加工を行い，インターネット上に投稿できるようになっている．モバイルデバイスはこのユーザの要求を満たすよう，動画像処理の様な膨大なデータ量及び演算量を高性能に処理可能なプロセッサを組込んでいる．加えて，近年では，これまでは大規模なパソコンで処理していたAI処理についてもモバイルデバイス上で処理できるよう要求されているところである．すなわち，近年のモバイルプロセッサは小型かつ低消費電力という制約がある中で，マルチメディア処理をより高性能に実行することが求められている．マルチメディア処理は主に繰り返し演算処理とテーブルルックアップ処理により構成されており，これらを高速に実行することがモバイルデバイスの高性能化につながる．

一方，モバイルデバイスの高性能化で，一般のユーザでも気軽に高画質な動画像を扱えるようになり，画像改ざんや盗撮等の新たな犯罪や社会課題が発生している．本論文では，この様なモバイルデバイスを利用して発生する社会課題や犯罪をソーシャルハザードと呼ぶ．このソーシャルハザードに対する対策について，半導体技術から検討することが求められている．さらに，これまではスマートフォンの様なモバイルデバイスが主流であったが，近年では動物の生態や人間の体調を解明するために，ユーザに直接取り付けるウェアラブルデバイスが普及している．このウェアラブルデバイスはユーザサポートを可能とするため，さらなる小型化，低消費電力化及び高性能化が要求されている．

以上の背景や課題に対して，本論文では汎用性のあるモバイルデバイス向けのアクセラレータであり，CPUでは処理が難しい膨大なデータを並列に処理可能な連想メモリベース超並列SIMD型演算コア（Content Addressable Memory-based massive parallel matriX core: CAMX）を提案し，その処理性能を検証する．CAMXはメモリサイズを自由に調整可能な連想メモリで構成されており，左右の連想メモリの間に小型の演算器を挟み込んだ構造となっている．特に，CAMXは連想メモリに格納しているデータに対して検索処理を容易に並列実行できるため，検索データと一致した内部のデータのみを書き換えることが高速にできる．すなわち，CAMXはマルチメディア処理において必要な繰り返し演算処理の並列化だけでな

く，一般的に並列処理が困難であるテーブルルックアップ処理を膨大なデータに対して，並列かつ高速に実行可能である．さらに，CAMX を組込むことを想定し，モバイルデバイスを利用したソーシャルハザードに対する対策手法及びシステムを提案・検証し，モバイルデバイスのさらなる進展を見据えてウェアラブルデバイスに CAMX を組込むことを想定した低消費電力化技術についても検証する．

1.1 節では，近年の半導体技術の発展及び社会課題について議論し，1.2 節にて本論文の目的を示し，最後に 1.4 節において，本論文の全体構成を述べる．

1.1 研究背景

1.1.1 モバイルデバイスにおけるプロセッサ技術の動向

近年，半導体技術が急速に進展している [1]–[3]．一般に，半導体はムーアの法則に沿って発展し続けると言われている．ムーアの法則とは，集積回路の集積率は 18 か月で 2 倍に増加するという法則である．すなわち，トランジスタ数は，3 年で約 4 倍となり，10 年で約 100 倍の集積率で増加していくことになる．図 1.1 は DRAM (Dynamic Random Access Memory)，NAND (Not AND) 型フラッシュメモリ，MPU (Micro Processing Unit)，APU (Accelerated Processing Unit) 及び GPU (Graphics Processing Unit) について，年別のチップ当たりのトランジスタ数の推移を示した概念図である [2]．図からわかるように，どの製品についてもトランジスタ数は右肩上がりで増加し，ムーアの法則に沿って成長を続けていることがわかる．一般に，トランジスタ数は処理能力に影響を与え，トランジスタ数が増加すると処理能力も向上する．一方，トランジスタ数が増加すると処理能力は向上するが，集積することで半導体全体の面積は増加する．このため，より高い集積度でトランジスタを実装するには，トランジスタの微細化も重要となり，微細化も併せて向上することでモバイルデバイスは小型かつ高性能となる [4], [5]．図 1.2 に示す概念図の様に，2010 年頃から量産最小加工寸法は 32nm になっており，さらに年々微細化が進んでいる [4], [5]．トランジスタの微細化が進むことで，モバイルデバイスは小型になり，生産コストも削減できる．このため，各企業はムーアの法則に従って技術革新を進めることが重要となり，各企業の努力が半導体技術の発展につながっている．

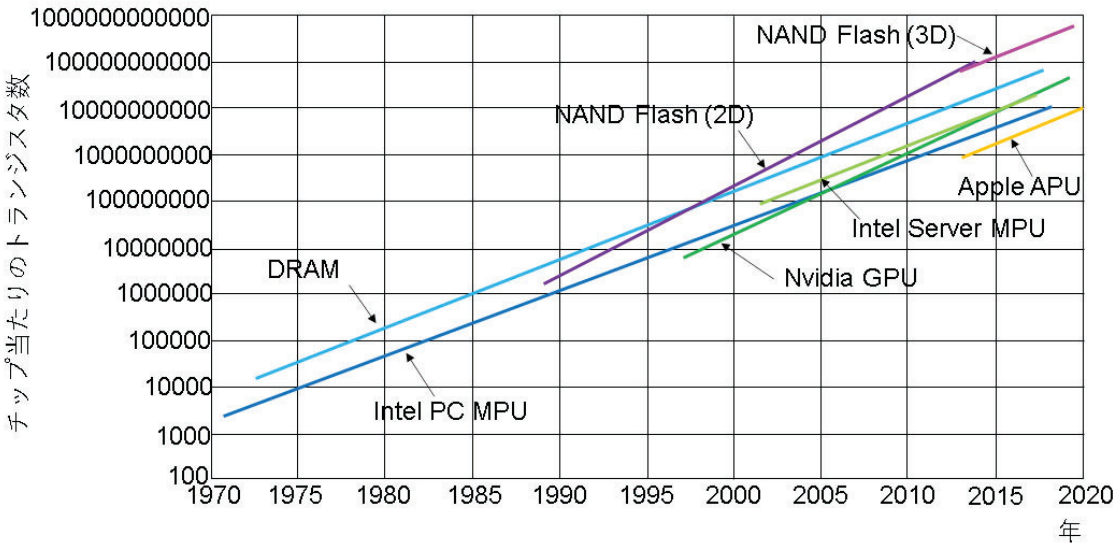


図 1.1: 半導体に組込まれるトランジスタ数の年別推移.

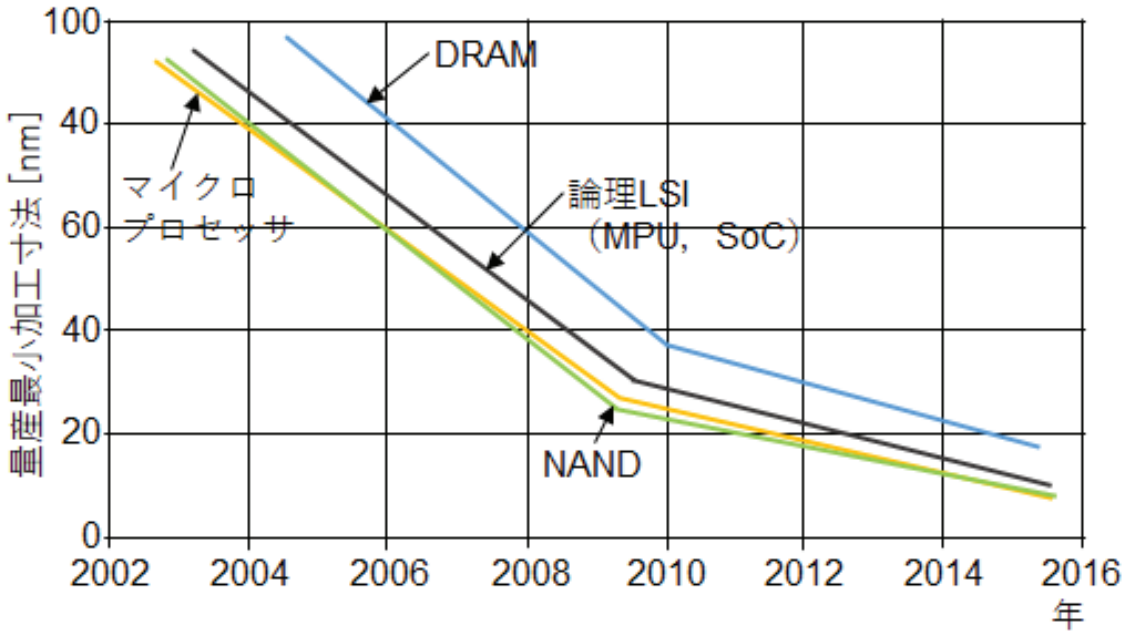


図 1.2: トランジスタの微細化推移.

モバイルデバイスの小型化及び高性能化により，ユーザの利用目的及びデバイスの種類は益々多様化している．例えば，携帯電話においてはより高性能なスマートフォンが登場したことで，持ち運び容易になり通信以外の利用が増えている．2010年からスマートフォンは爆発的に普及し，現在までに国内では約 80 %の世帯が保有している状況にある [6], [7]．例えば，図 1.3 に示すように，通信速度は年々高速化しており，この通信速度に併せてスマートフォン全体の性能も向上し続けている [7]．そして，スマートフォンの性能向上により，電話の音声だけを処理するのではなく，インターネットのブラウジングや動画視聴・編集まで様々なデジタル処理が可能となっている [8]．また，スマートフォンに搭載されているイメージセンサ等の各種センサも性能が向上している．特に，近年ではスマートフォンで撮影した動画や画像をスマートフォン上で編集・加工し，インターネットに投稿するユーザが多い [9]．実際，総務省等において，スマートフォンの利用状況について調査を行ったところ，動画視聴・編集やゲームのプレイが大半を占めている [10], [11]．また，モバイルデバイスの一つとして，ウェアラブルデバイスも 2014 年から普及し始め，保有率は年々上昇している [12]．ウェアラブルデバイスとは，人間の腕や頭等に直接装着するものであり，スマートウォッチやスマートバンドが特に普及している [13]．これらは，人間が身に付けるものであるので，行動等に支障が無いよう，スマートフォンに比べてさらに小型のデバイスとなっている．しかしながら，小型のウェアラブルデバイスにおいても，スマートフォンと同等の性能が要求されている．すなわち，モバイルデバイスは電話やメールの様な通信手段としての役割から，最近では動画視聴，カメラ撮影，画像編集等の娯楽の一種としての役割が増加してきており，さらに人間の生活で活用できるよう小型化が進んでいる．これらの要求を実現するため，画像処理，AI (Artificial Intelligence) 処理，通信処理，暗号処理等，モバイルデバイス上で実行するデータ量及び演算量は膨大になっており，マルチメディア処理を高性能に実行可能なプロセッサが求められている [14]–[17]．ここで，モバイルデバイスに搭載されているプロセッサの性能について確認しておく．上述の背景から，モバイルデバイスのプロセッサは年々高性能化している．図 1.4 は，スマートフォンの一種である iPhone に搭載されているプロセッサのトランジスタ数に関する年別の推移を示している [18], [19]．このトランジスタ数の推移から，モバイルモバイルに搭載されているプロセッサのトランジスタ数についてもムーアの法則に沿って年々増加を続けていることが確認できる．この性能向上により，マルチメディア処理が実行可能となっている．

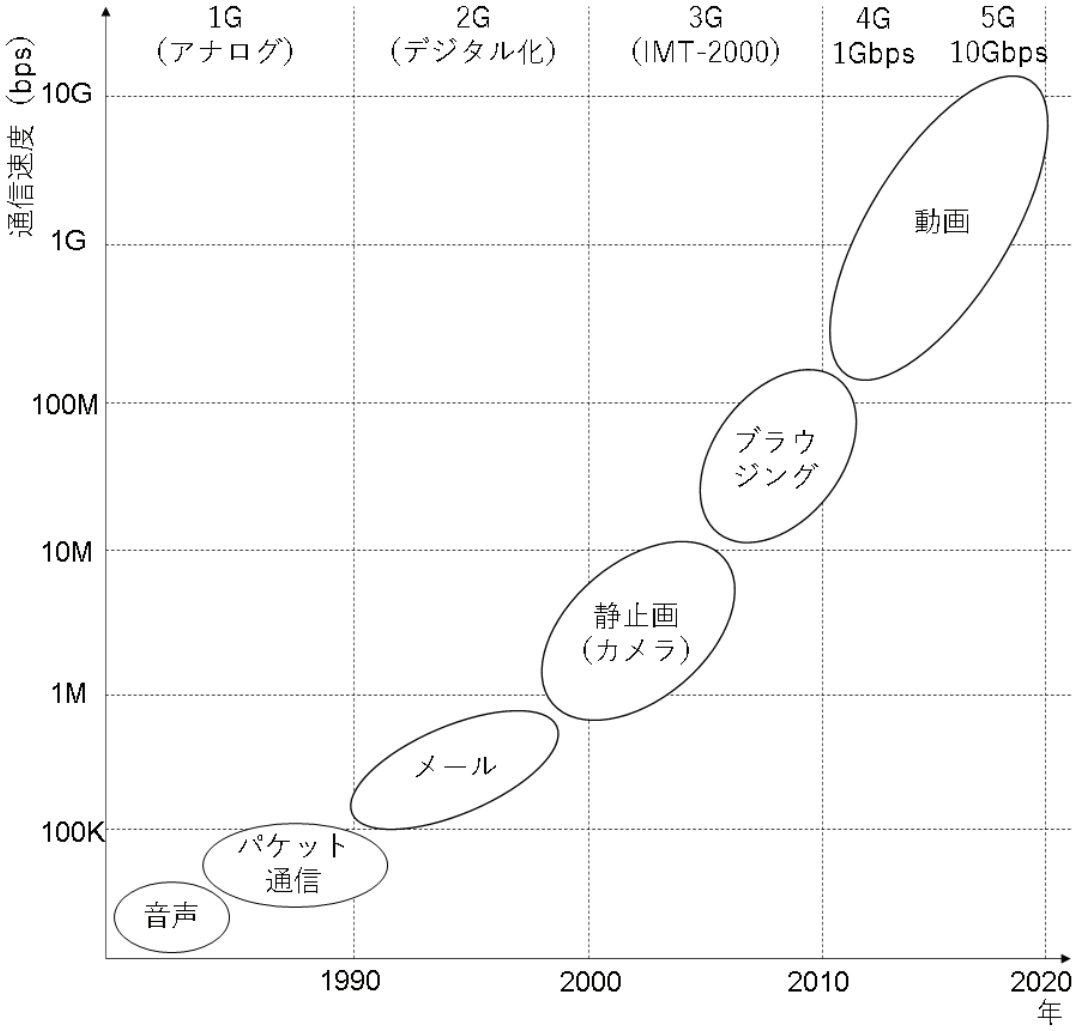


図 1.3: スマートフォンにおける通信速度と要求処理の変化.

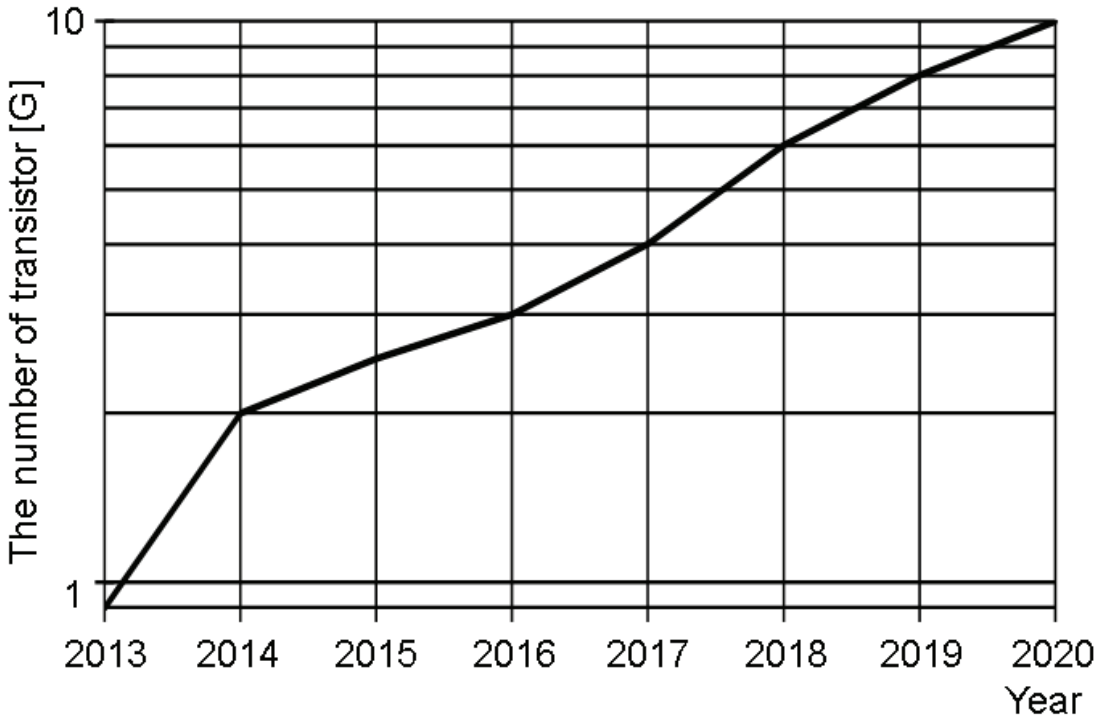


図 1.4: iPhone に搭載されているプロセッサのトランジスタ数変化.

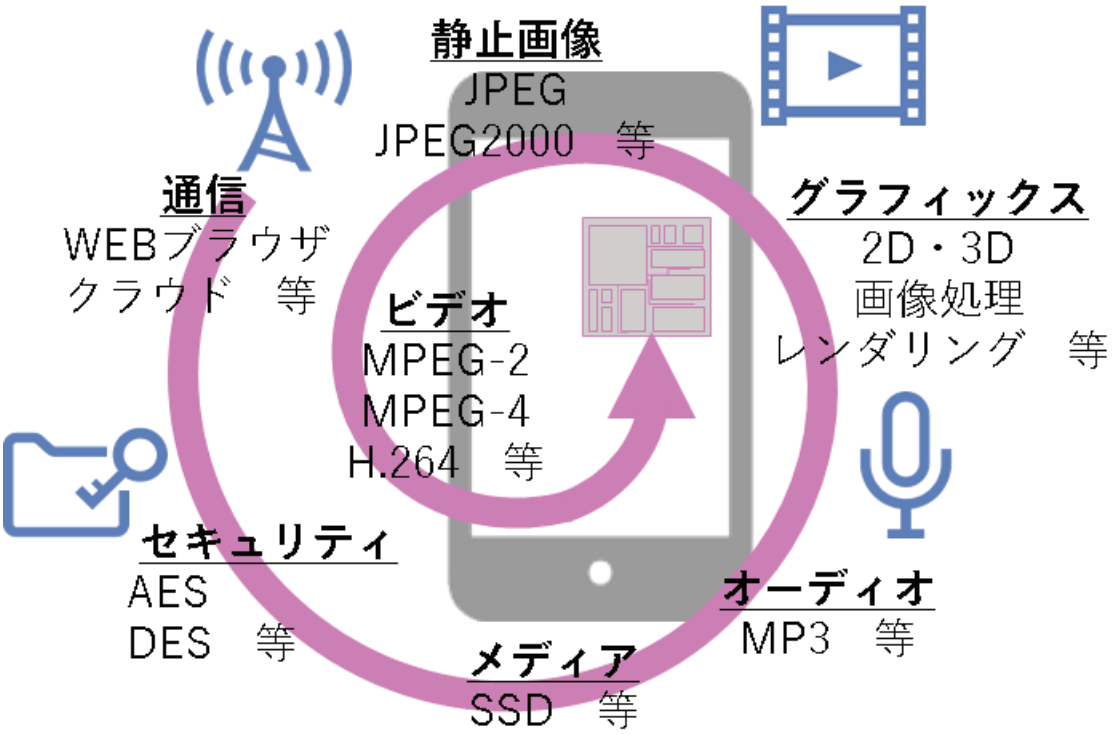


図 1.5: デジタルコンバージェンスの概念.

一方，モバイルデバイスは持ち運びできることが重要であるため，小型かつ低消費電力であることが必須となる．これらの制約があるため，モバイルデバイスにはサイズが大きい複数のプロセッサを搭載することはできず，サイズが小さい 1 個のプロセッサによりマルチメディア処理を実行する手法が一般的である．このコンセプトは，“ディジタルコンバージェンス” と呼ばれている [20], [21]．図 1.5 に示す概念の様に，1 個のプロセッサにおいて通信，セキュリティ（AES (Advanced Encryption Standard)，DES (Data Encryption Standard) 等），オーディオ処理（MP3 (MPEG-1 Audio Layer-3) 等），動画像処理（AI 処理等）を実行する．“ディジタルコンバージェンス’ を実現するため，モバイルデバイスのプロセッサには CPU (Central Processing Unit) とは別に，特定の処理に特化した専用回路を複数組込んで高性能化を実現する手法が一般的になっている．例えば，DSP (Digital Signal Processor) と呼ばれる専用回路は，デジタル信号のデータを高速に処理可能な回路であり，CPU とは別にプロセッサ内に組込まれている [22], [23]．モバイルデバイスに搭載することで，積和演算を CPU 内で処理するのではなく，積和演算に

特化した DSP に処理を一任することで高性能化を実現している．GPU (Graphics Processing Unit) と呼ばれる専用回路は，3D グラフィック等の膨大なデータの処理を高速に実行することが可能であり，CPU とは別にプロセッサ内に組込まれている [24]–[26]．デスクトップパソコンに組込まれることが多かったが，近年ではモバイルデバイス向けの開発も進んでおり，スマートフォンに GPU を組込むことで画像処理が高速に行えるようになり，モバイルデバイス上で VR (Virtual Reality) や AR (Augmented Reality) を実現することが可能となっている．NPU (Neural network Processing Unit) と呼ばれる専用回路は，AI 処理の一種であるニューラルネットワークを高速に実行することが可能となり，CPU とは別にプロセッサ内に組込まれている [27]–[29]．AI 処理は積和演算を繰り返し実行する必要があり，近年では画像処理においても利用されており，処理すべきデータ量が膨大になっている．この膨大な処理をモバイルデバイス上でも実行できるよう NPU が組込まれるようになっている．特に，AI 処理はこれまでクラウド上において実行されていたが，近年ではモバイルデバイス上で完結するエッジ AI という手法も提案されており，小型かつ低消費電力という制約がある中で，益々膨大なデータ量及び演算量を高速に実行する必要がある． [30], [31]．上述の様に，モバイルデバイスの制約がある中で，特定処理に特化した専用回路を複数組込むことで性能向上を実現している．しかし，専用回路を複数組込むことで，回路規模が大きくなり，小型化及び低消費電力化を実現できない可能性も課題となっている．すなわち，マルチメディア処理を複数の専用回路に頼らないプロセッサが必要となっており，1 つのプロセッサで高処理，プログラマビリティ及び汎用性を実現することが重要となる．

1.1.2 モバイルデバイスの技術革新に伴う潜在的な社会ハザード

モバイルデバイスは益々日常生活で利用しやすい機器となっており，日本では 2016 年にモバイルデバイスの一種であるスマートフォンの保有率はパソコンを抜いている [6]．スマートフォンは一般のユーザが容易に使えるように設計されており，日常生活において手放せない機器となっている．さらに，スマートフォンに搭載されている各種センサも急速に性能が向上しており，特にスマートフォンに搭載されているイメージセンサの画素数は年々向上している [32]–[34]．図 1.6 にスマートフォンである iPhone に搭載されているイメージセンサの画素数推移を示す．イメージセンサの画素数は年々増加しており，撮影可能な動画像の解像度が向上している．これに伴って，一般のユーザが高画質な動画像を気軽に撮影できるようになったことで，スマートフォンのカメラ機能を利用した盗撮等の犯罪が急増している．図 1.7 に示すように盗撮の検挙数は年々増加しており，その内，盗

撮ではスマートフォンの利用が最も多い状況にある．さらに，撮影した動画像は気軽にモバイルデバイス上で編集・加工が行えるようになっており，インターネット上に投稿もできるようになったことが原因である．この様に，半導体の技術発展に伴ってモバイルデバイスの利便性が向上した一方で，モバイルデバイスを利用した犯罪が急増しており，技術発展と併せてソーシャルハザードに対する対策についても検討する必要がある．

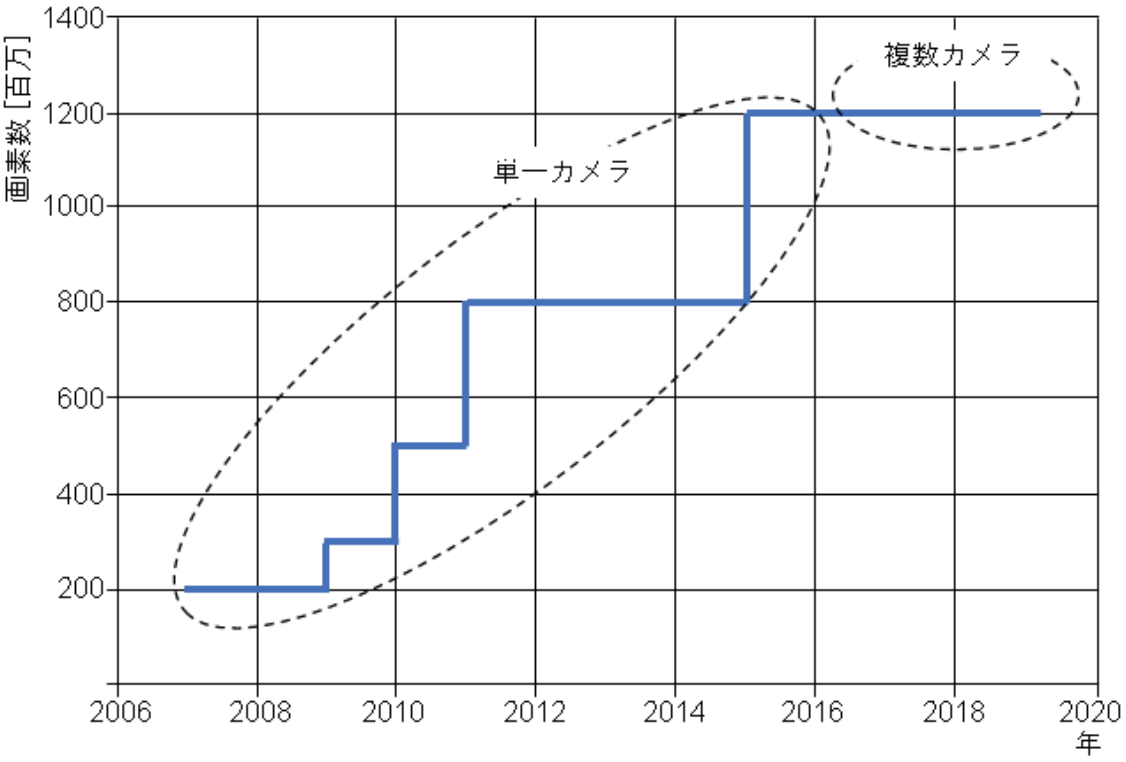


図 1.6: スマートフォンに搭載されているイメージセンサの画素数推移.

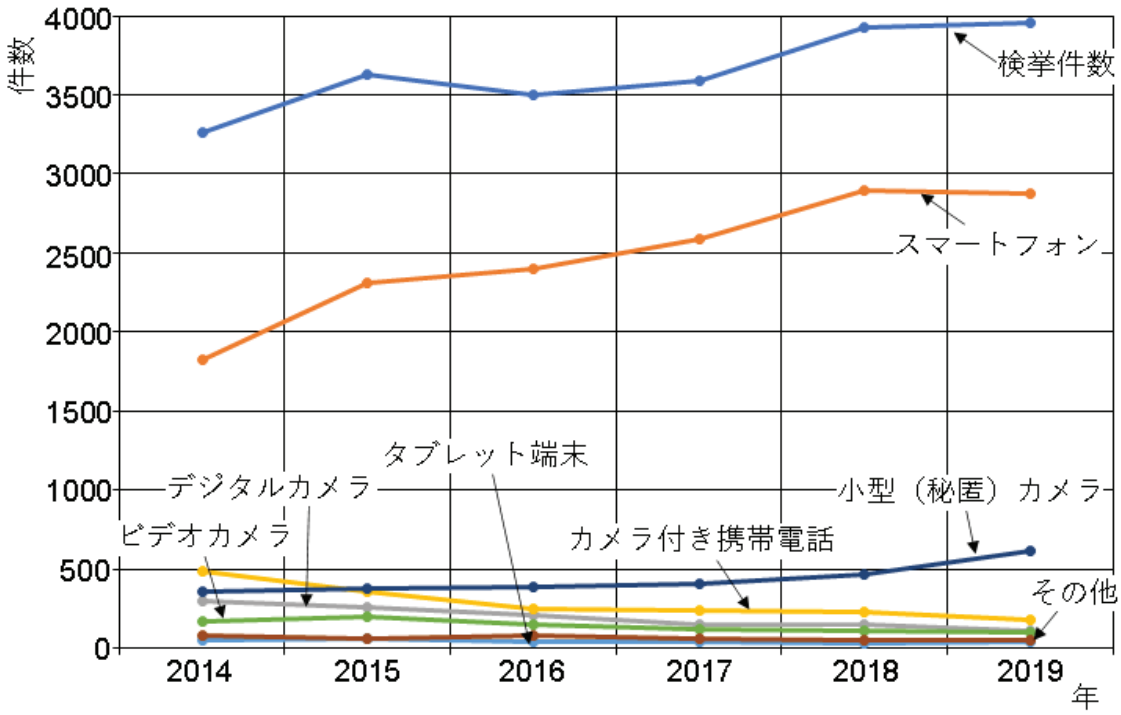


図 1.7: 盗撮犯罪の検挙数及び盗撮に利用された機器の推移.

さらに，犯罪だけではなく，画像改ざんの様なデジタルデータの改ざんも社会課題となっている [35]．画像改ざんは悪意等を持って意図的に画像の編集・加工を行うことであり，モバイルデバイスの性能向上により気軽に動画像を編集・加工が行えるようになったことが原因である．近年では，動画像の改ざんにディープフェイクという技術が知られるようになっている [36]–[38]．この技術は，複数の画像から AI 処理を利用して，画像に写る人物を動画像内で自由に動かすことができ，実際に存在しない動画像を作成できる．改ざんされた動画像はインターネットに投稿され，あたかも事実であるかのように世の中に拡散され，大きく報道されて社会問題になることもある．これまで，これらの動画像の改ざんには大規模な処理を実行できるパソコンにおいて作成されていたが，近年はディープフェイク画像等をモバイルデバイス上で容易に作成できるようになっている．この技術の対策として電子透かし技術の利用が検討されているが [38]，モバイルデバイスにおいて実行可能な対策が必要である．このため，プロセッサの性能向上に併せて，ソーシャルハザードに対する対策も行えるような汎用性のあるプロセッサが求められている．

1.1.3　モバイルデバイスの技術革新に伴う新たな利用価値

半導体技術の発展により，モバイルデバイスは小型かつ低消費電力になってきている．このため，日常生活において利用するウェアラブルデバイスが普及している [13],[39]．ウェアラブルデバイスは常に身に付けることを想定されており，身に付けた対象者の体調や行動を記録する．種類も様々であり，センサ情報を取得するもの，イメージセンサから動画像を取得するもの等がある．そして，ウェアラブルデバイスは今後も普及していくことが予測されている [12]．図 1.8 の様にウェアラブルデバイスの総出荷台数は急速に上昇しており，特にスマートウォッチの普及率は著しくなっている．スマートウォッチは体調管理に加え，音楽再生や動画像撮影等のマルチメディア処理が可能となっている [40]．すなわち，小型かつ低消費電力という制約がより厳しい条件においても，高性能なプロセッサが求められている状況にある．さらに，人間に取り付けることを想定したウェアラブルデバイスだけではなく，動物にもモバイルデバイスを装着する手法も普及している [41]．これは，動物にモバイルデバイスを取り付けることで，動物の体調や行動を記録し，生態を解明しようという手法である．以上の様に，人間や動物の行動をサポートできるようなモバイルデバイスが今後普及していくことになる．この要求を実現するために，モバイルデバイスはさらなる小型かつ低消費電力という制約がある中で，より汎用性があり，高性能なプロセッサが必要となっている．

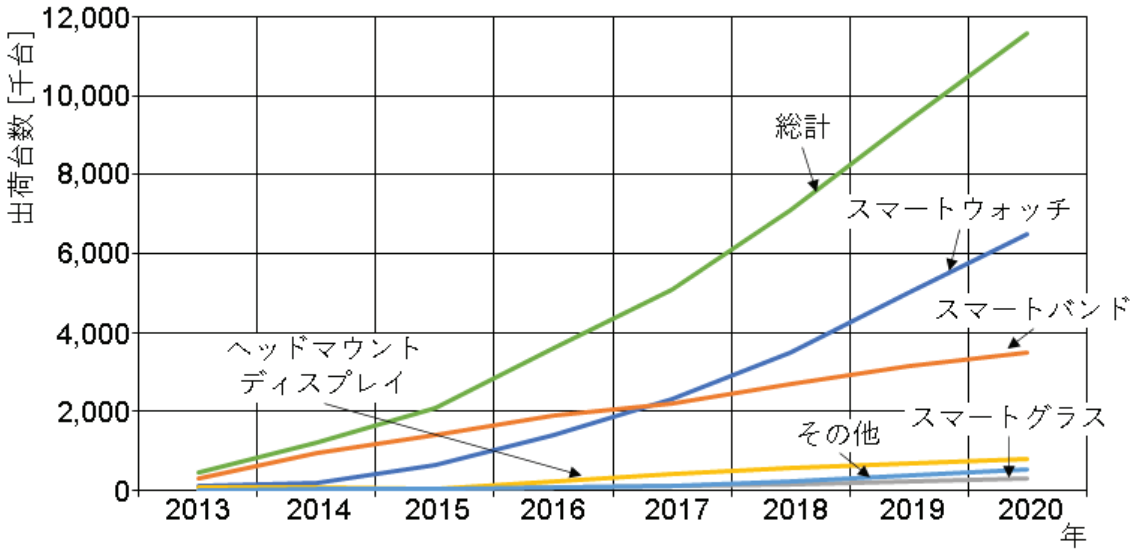


図 1.8: ウェアラブルデバイス種類毎の出荷台数推移.

1.2 本研究の目的

モバイルデバイスは，小型かつ低消費電力という制約がある中でリアルタイムに高性能な処理が求められている．さらに近年では，画像処理や AI 処理といった膨大なデータ量かつ演算量が必要となるマルチメディア処理の実行要求が特に高まってきている．このため，近年では特定の処理に特化した複数の専用回路を組込むことで高性能化を実現することが主流である．しかし，ウェアラブルデバイスの普及により今後もさらなる小型化，低消費電力化が求められ，単一のプロセッサに複数の専用回路を組込むことなく，高処理，プログラマビリティ及び汎用性を兼ね備えたプロセッサを実現することが必要である．

一般に，マルチメディア処理は繰り返し演算処理とテーブルルックアップ処理により構成される [42]–[44]． 繰り返し演算処理は AND，OR，XOR，加算，乗算等のことであり，テーブルルックアップ処理は入力データを参照テーブルにより出力データに変換する処理のことである．このため，モバイルデバイス上におけるマルチメディア処理の性能は，繰り返し演算処理とテーブルルックアップ処理の実行性能に依存する．すなわち，これらの処理を高速かつ高性能に実行できるプロセッサが必要となる．高速かつ高性能に処理する手法の一つとして，膨大なデータ量を並列に処理し，処理速度及び性能を向上させる手法が知られている． 一般に，繰り返し演算処理は容易に並列処理が可能である [45]–[52]．一方，テーブルルックアップ処理は並列に実行することは容易ではない [53]–[55]．すなわち，モバイルデバイスの高性能化には，テーブルルックアップ処理を並列に実行できることが重要となる．

そこで，本論文では連想メモリベース超並列 SIMD 型演算コア（CAMX）を提案する [56]–[60]．CAMX は繰り返し演算処理とテーブルルックアップ処理を並列に実行することができ，AND，OR，XOR，加算，検索等，様々な基本命令を実行できることを確認している．ベースとなった技術はマトリクスアーキテクチャ型超並列演算プロセッサ MX-1 やセルラオートマトン処理用連想メモリ（CAM^2）である [56]–[65]．

1.3 本論文の意義

本論文で提案する CAMX はモバイルデバイスのプロセッサに内蔵され，CPU のアクセラレータとしての役割を想定している． DSP，GPU 及び NPU はそれぞれ画像処理，グラフィック処理及び機械学習処理の個別毎の処理に特化しているが，CAMX は他の専用回路と異なり，これらの様々な処理を汎用的に実行できる機能を持つ．このため，CAMX には主に AND，OR，XOR の論理演算処理及び加算

演算処理で構成されており，乗算等の専用回路は組込んでいない．一方，演算器の種類を最小に抑えることで，CAMX は小面積に実現することが可能となる．

CAMX はモバイルデバイスに組込むことを想定しており，小型，高性能及び汎用性を兼ね備えたアクセラレータとなる．このため，CAMX はモバイルデバイスの一種であるスマートフォンに組込むことを想定しており，スマートフォン上でリアルタイムに膨大なデータ及び処理が必要となる画像処理や AI 処理を実現できることを目標にしている．本論文では，リアルタイムに画像処理が必要となる盗撮防止システム，繰り返し画像処理が必要となるモルフォロジカルパターンスペクトラムを利用した画像改ざん検知手法に関する事前検証についても議論し，今後スマートフォンに CAMX を組込むことを想定して社会ハザードの対策手法を提案する．さらに，近年急速に普及しているウェアラブデバイスに CAMX を組込むことも想定し，ユーザサポートの一種である動物の生態を調査するためのバイオロギング用データロガーの開発について議論する．バイオロギング用データロガーは動物に直接取り付けるウェアラブルデバイスであるため，小型かつ低消費電力であることが要求される．今後 CAMX をデータロガーに組込むことを想定し，事前検証を行う．以上の様に，CAMX が利用できるようなモバイルデバイスは幅広く存在し，CAMX の汎用性及び高性能化を検証しつつ，社会ハザードの対策及びユーザサポートへの導入について検討する．

1.4 本論文の構成

- 第 2 章 (公表論文：1, 2)

  本論文において提案する機能メモリベース超並列 SIMD 型演算コア（CAMX）について提案し，基本演算命令及び検索命令について検証を行い，数千データが並列に演算可能であることを確認する．また，マルチメディア処理において必ず必要となる乗算処理について，CAMX における最適な手法を検証する．さらに，暗号処理の一種である AES を CAMX に実装し，既存のモバイルデバイス向けプロセッサよりも高性能であることを示す．加えて，マルチメディア処理において必要な浮動小数点による加算処理についても実装し，CAMX の有効性を示す．

- 第 3 章 (公表論文：4)

  近年の半導体技術の急速な発展により，社会ハザードとして表面化している画像改ざん及び盗撮に関する対策を提案する．さらに，画像改ざん検知手法については，CAMX の基となった MX-1 において処理の有効性を検証し，

CAMX においても有効であることを確認する．盗撮防止システムについては，今後 CAMX を組込むことを想定し，最適なアプリケーションの検証を行い，リアルタイムに盗撮を防止可能な手法について説明する．

- 第 4 章 (公表論文：3)

  ユーザサポートを目的として，動物にモバイルデバイスを直接取り付けて生態を解明するデータロガー及び人間に直接取り付けて体調を管理するウェアラブルデバイスにも CAMX を展開することを検討する．動物に直接取り付ける低消費電力なデータロガーのためにノーマリオフ手法を提案し，動物の動作と協調可能なデータロガーを開発する．開発するデータロガーについても CAMX を組込むことでさらなる消費電力の削減及び小型化が実現できると考える．

第2章　機能メモリベース超並列SIMD型演算コア（CAMX）のアーキテクチャ概要及び動作検証

本章では，マルチメディア処理に特化した機能メモリベース超並列 SIMD 型演算コア（CAMX）を提案し，基本演算処理，検索処理，符号付き乗算処理，AES 処理及び浮動小数点加算処理を実装する．マルチメディア処理は繰り返し演算処理及びテーブルルックアップ処理により実現されるため，これらの処理を高性能に実現するためには並列処理が重要となる．CAMX はマルチメディア処理を高速かつ高性能に実現するため，繰り返し演算処理及びテーブルルックアップ処理を高並列に処理可能である．CAMX は組込み機器向けのアクセラレータとして高処理，プログラマビリティ及び汎用性を実現する．このため，以降の節では CAMX の概要を説明し，マルチメディア処理において必要となる基本的な演算（基本演算処理，検索処理，符号付き乗算処理，AES 処理及び浮動小数点処理）について検証する．

2.1　はじめに

2.1.1　マルチメディア処理に向けて

1.1 節で述べた様に，近年，半導体技術が急速に発展しており，マルチメディア処理の要求性能も急激に向上している．さらに，モバイルデバイスの特性から，小型かつ低消費電力の制限がある中で，膨大なデータ量及び演算量をリアルタイムに処理する必要がある．マルチメディア処理は，繰り返し演算処理とテーブルルックアップ処理で構成されており，これらの処理を高性能に実行するため並列に処理する手法が一般的である [42]–[44]．繰り返し演算処理は，論理演算と算術

演算で構成されており，一般的にはこれらの演算は並列に処理することは容易である [45]–[52]．一方，テーブルルックアップ処理は，入力データを参照テーブルを用いて出力データに変換することであるが，データの検索や変換が必要となるため，一般的に並列処理することは困難であることが知られている [53]–[55]．本論文では，繰り返し演算処理とテーブルルックアップ処理をどちらも並列に実行することが可能であり，マルチメディア処理を高性能に実現することが容易である機能メモリベース超並列 SIMD 型演算コア（Content Addressable Memory-based massive parallel matriX core: CAMX）を提案する [56]–[60]．このため，CAMX は組込み機器向けアクセラレータとしてプログラマブルな SoC コアを実現可能となっている．以降に，提案する CAMX の概要及び検証を行う．なお，CAMX は主に SRAM (Static Random Access Memory) で構成されるマトリクスアーキテクチャ型超並列演算プロセッサ MX-1 を改良したものであり，この MX-1 から主な構成を連想メモリに変更することで，特にテーブルルックアップ処理に対してより高性能に並列実行が可能となる．

2.1.2 MX-1 の構成

ここで，CAMX の参考となった MX-1 の構成について簡単に説明しておく．MX-1 は組込み機器向けマトリクスアーキテクチャ型超並列演算プロセッサであり，CAMX と同様に組込み機器向けアクセラレータとして開発されている [61]–[65]．そして，MX-1 は動画像等の数値・論理演算を実行する際に，効率的な並列処理が実行可能であり，ルネサスエレクトロニクス社製の SIMD 型演算プロセッサコアとしてこれまでに研究・開発が行われてきた．SRAM を主な構成とし，演算器（Processing Element:PE）の配置やデータの処理ベクトルを工夫することで，従来のプロセッサよりも並列度を大幅に向上させることができるようになった．さらに，演算器と SRAM 領域を密結合し，演算器とメモリ間の I/O 転送にかかる消費電力を低減させている．90 nm 7Cu CMOS テクノロジの実装結果では，面積 3.1 mm^2，消費電力 250 mW となり，16 ビットの加算処理では動作周波数 200 MHz で 40 GOPS（Giga Operation per Second）の性能が実現可能であることを確認している．我々はこれまでに，画像情報変換処理，電子透かし処理，及び暗号化処理等，様々なアプリケーションを実装してその効果を確認してきた [66]–[74]．

MX-1 は C 言語を用いることで様々なアプリケーションを実行可能なハードウェア構成を採用しており，図 2.1 にその構成を示す．MX-1 は SDRAM，制御用 CPU コア，DMA（Direct Memory Address）及び MX コアで構成されており，MX コアは左右の SRAM（512 ビット ×1,024 エントリ）が 1,024 個の 2 ビット PE を挟み込んだ配置となっている．そして，外部メモリとのバスであるインターフェースモ

ジュール及び MX コアを制御するコントローラで構成されている．すなわち，1 命令で 1,024 個のデータを 2 ビット単位で並列に処理することが可能であり，制御用 CPU はプログラムのフロー制御及び逐次処理を実行する．CPU と MX コアを効率よく動作させることが処理速度及び性能に影響を与える．命令の実行は，制御用コントローラ内部の命令メモリから制御線を通して PE に命令が送信され，1,024 個の PE が左右の SRAM からデータを読み出し，演算を行う．MX コアの演算器は，主に加算器，乗算器及び論理演算器で構成され，バリッドフラグ（Valid flag : V フラグ）が追加されている．このバリッドフラグで各エントリの演算の有無を制御する．バリッドフラグに 1 が格納されるとエントリは演算が行われ，0 が格納されると演算が行われない．演算器と左右のエントリは水平チャネル（Horizontal channel）で接続されており，これが 1,024 本垂直に配置されている．このため，1 命令で 1,024 データを並列に演算可能となっている．垂直方向にデータを移動させる際には，垂直チャネル（Vertical channel）を利用する．演算処理の方法はビットシリアル・ワードパラレルになっており，インターフェースモジュールは外部 SDRAM に格納されているデータの処理ベクトルを直行に変換して MX コアに格納する．コントローラは内部に命令メモリを持ち，アプリケーションに合わせてプログラムを入れ替え，柔軟にマルチメディア処理を実行する．これまでに，この MX-1 を用いて，セキュリティ，JPEG 圧縮，顔認識等を実装してきた（表 2.1）．目的としては，デジタルコンバージェンス時代において要求されている，情報・通信機能，マルチメディア機能，及び認識機能を MX-1 における有効性の確認である．

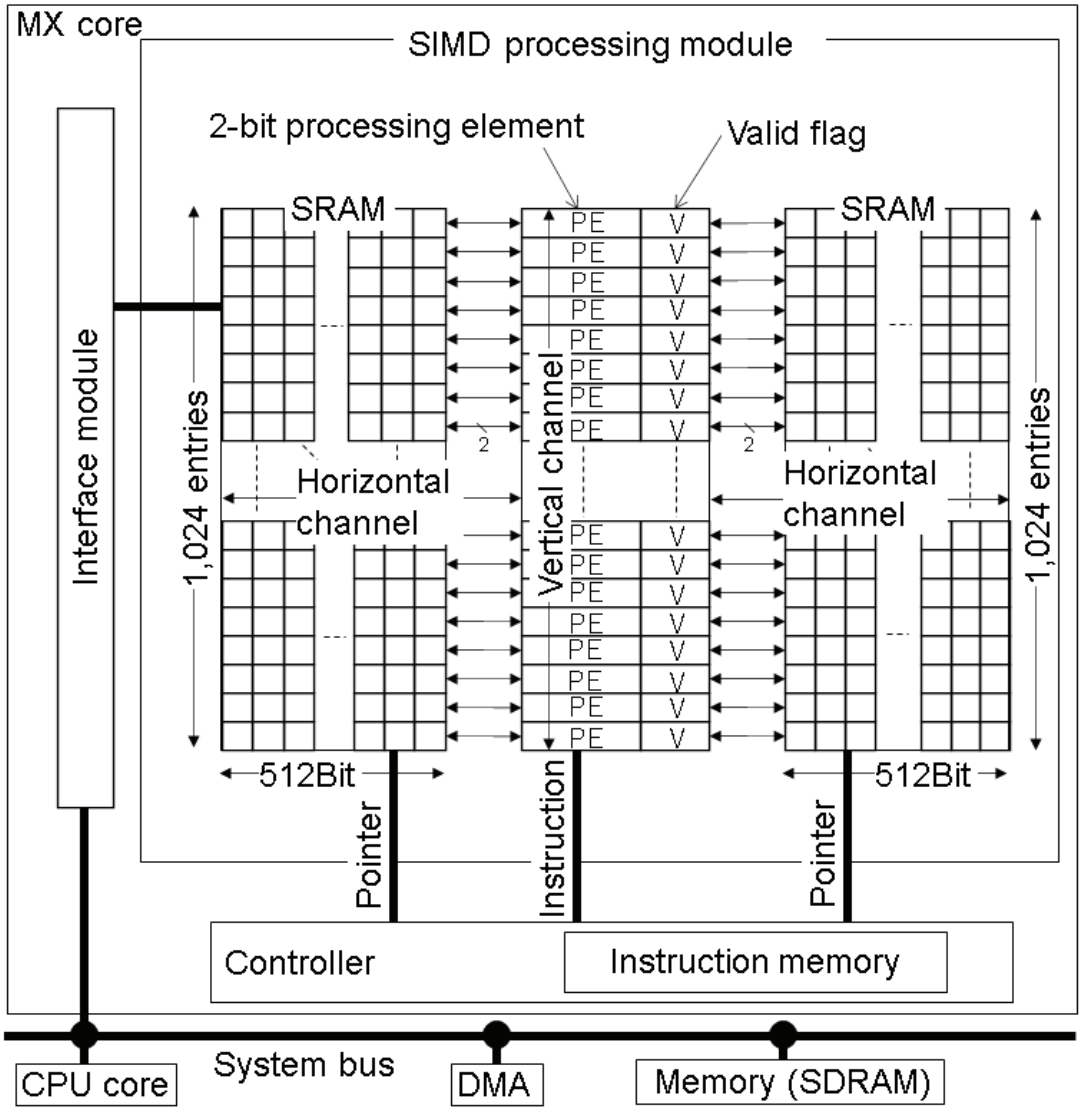


図 2.1: MX-1 のブロック図.

表 2.1: MX-1 を用いたデジタルコンバージェンスに向けた研究成果.

Required functions	Research progress	Research contents	Algorithm
Information and communication function	✓	Security	AES Mersenne Twister Huffman coding
Multimedia function	✓	Data compression	JPEG
Recognition function	✓	Face recognition	Haar-Like features
Graphic function	✓	Spectrum	Morphological pattern spectrum

2.2 CAMX の構成

本節では，CAMX のアーキテクチャについて説明する．図 2.2 は，CAMX のブロック図を表している．CAMX は 2 個の連想メモリ，数千個の演算器，コントローラ及びインターフェースモジュールにより構成されている．連想メモリは，左側と右側に配置され，それらの間に連想メモリを挟み込んだ配置となっている．さらに，コントローラは各演算器に対して処理内容について命令を送信する機能を持っている．本論文では，左右の連想メモリについて，それぞれのサイズを横方向に 128 ビット又は 256 ビット（$= x$）とし，縦方向に 1,024 エントリ（$= n$）とする．なお，これらのサイズはユーザによって容易に変更可能である．また，演算器は小面積かつシンプルな構成であることに重点を置いているため，1 ビット演算とし，主に論理演算器と加算演算器のみで構成されている．以上の構成により，CAMX はビットシリアル・ワードパラレルに実行され，高並列な処理を実現し，演算器における処理についてはパイプライン処理により実行される．なお，CAMX は 1 個のプロセッサ内に CPU とは別に配置され，それぞれシステムバスによりつながっている．CPU は並列に処理可能な命令を CAMX に委託し，CAMX は CPU の命令により並列に処理可能なデータを実行することになる．すなわち，CAMX は CPU のアクセラレータとして機能することになる．

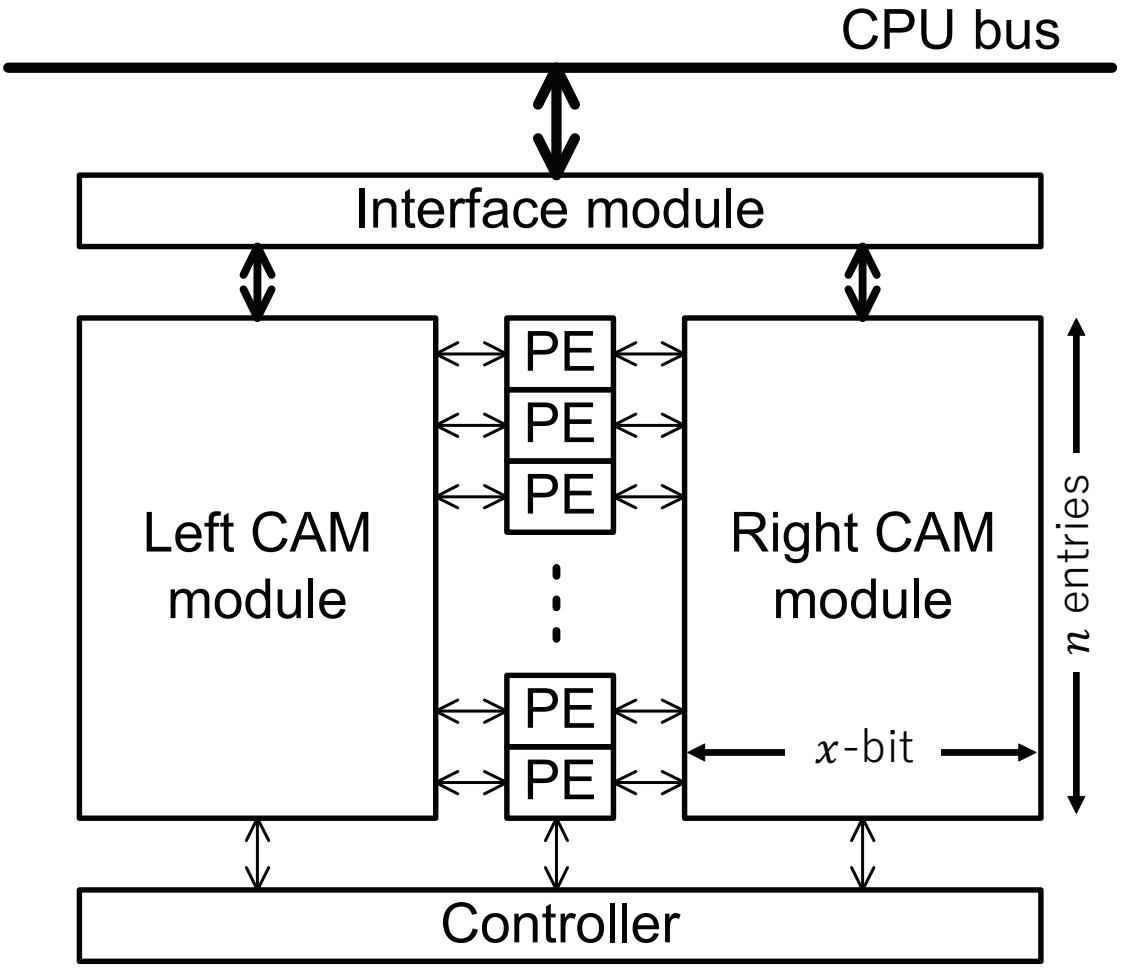


図 2.2: CAMX のブロック図.

2.2.1 ビットシリアル・ワードパラレル処理

CAMX は左右に x ビット × n エントリの連想メモリを搭載しているため，大量のデータをそれぞれに格納でき，格納したデータは並列に処理可能である．CAMX の詳細なアーキテクチャは図 2.3 に示す．この図から，格納するデータは x ビット × n エントリとして左右の連想メモリに格納することになるため，ビット毎に垂直方向へデータが入力され，エントリ毎に水平方向へ演算器（PE）にデータが出力することになる．すなわち，CAMX はビットシリアル・ワードパラレルに格納

データを処理することになる．また，連想メモリから演算器へ出力されたデータは 1 ビット毎に処理され，このデータはパイプライン処理として実行される．なお，演算器はコントローラからの処理命令により，全エントリの演算器が同時に実行され，全データが並列に処理されることになる．

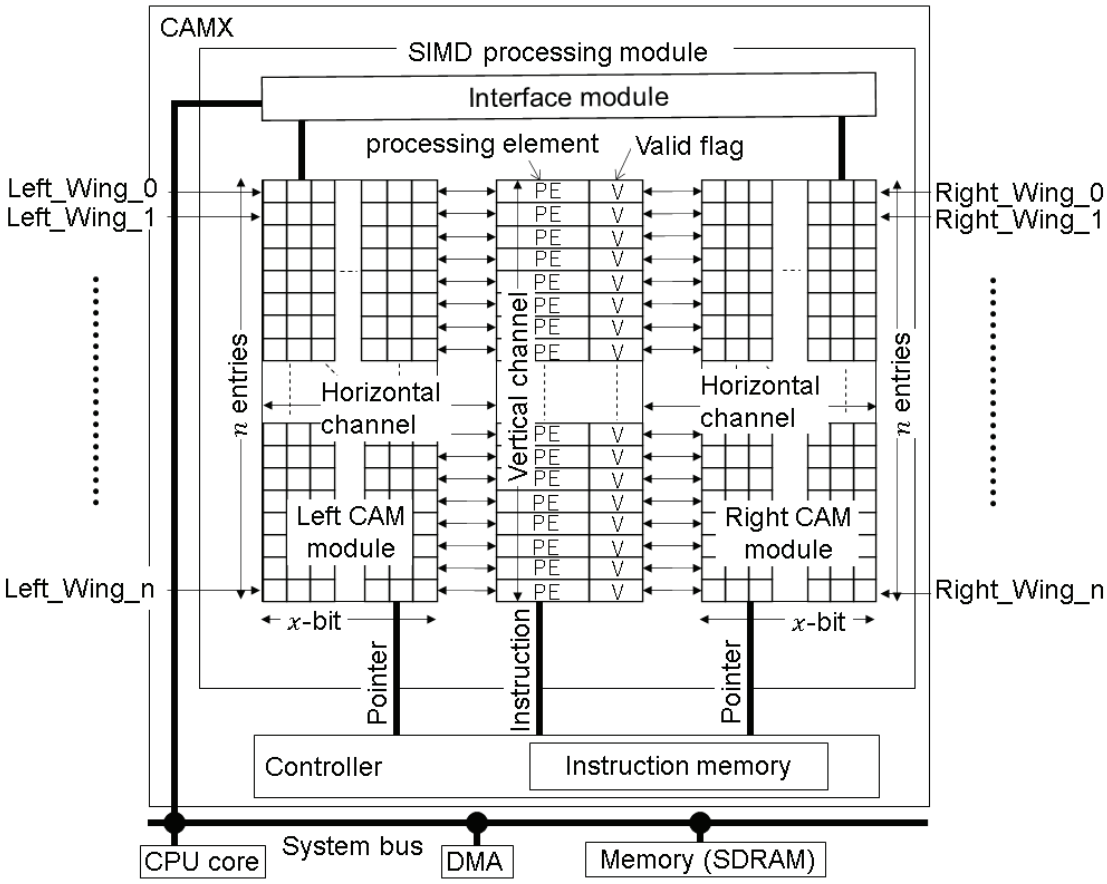


図 2.3: CAMX の詳細アーキテクチャ．

2.2.2 SIMD 処理

CAMX はコントローラの命令により全エントリの演算器が同時に実行される．すなわち，コントローラからの 1 命令により，数千のデータが同時に処理されるため，CAMX は SIMD 処理が可能となる．また，図 2.3 に示すように，演算器である各 PE にはバリッドフラグ（V）が用意されており，このフラグを利用するこ

とで実行するエントリを選択可能となる．バリッドフラグは連想メモリに対して一致検索処理により実行した結果を格納するマッチレジスタと OR 演算を行うことで，出力値が 1 となった場合に任意のエントリを実行する．すなわち，全エントリを同時に実行させることも，個別にエントリを実行させることも，命令により自由に制御可能である．基本的には，コントローラからの 1 命令により全エントリが実行されることになるが，上述のような方法によりバリッドフラグとコントローラで 1 命令をエントリ毎に詳細な操作が可能となる．

2.3 CAMX における基本処理の手順

本節では，CAMX における基本的な処理手順について説明する．CAMX の演算器は小面積かつシンプルな構成な回路を目的としているため，主に論理演算器と加算演算器で構成されている．このため，乗算演算，浮動小数点演算等の専用回路は組込んでおらず，論理演算と加算演算を組合わせて処理することで実行可能となる．

2.3.1 基本的な演算フロー

図 2.4 は基本的な演算の流れを示している．CAMX の演算は，左右の連想メモリに格納されたデータを演算器へ交互に読み出し，演算器で処理された結果を左右どちらかの連想メモリに格納することになる．この基本的な演算の手順は全エントリに対し実行されるため，連想メモリに格納された全てのデータが並列に処理される．

ここで，1 エントリ中の連想メモリ及び演算器の詳細な処理フローについて，図 2.5 を用いて説明する．なお，CPU と CAMX はシステムバスにより接続されており，CPU は並列に処理可能な命令を CAMX に委託する．

1. CPU は外部メモリである SDRAM から対象データを読み込み，CAMX で並列に処理可能な場合，処理を CAMX に一任する．CAMX は CPU からの命令をインターフェースモジュールを通して，CAMX に入出力するデータのアドレスを指定する（図 2.3）．

2. 入出力するデータは連想メモリ上でビット毎に垂直方向で扱われ，CAMX の演算器によって処理される場合はデータ毎に水平方向に読み出される．演算器は 1 ビット毎に処理するため，各エントリの最下位メモリセルから順番にデータを処理する．まずは，左右どちらかの連想メモリのメモリセルから

データを読み出し，読み出したデータを演算器に組込まれたストアレジスタに格納する．次のクロックで，他方の連想メモリのメモリセルからデータを読み出し，論理演算器及び加算演算器を組込んだ ALU (Arithmetic and Logic Unit) にて処理を行う．さらに，次のクロックで，左右どちらかの連想メモリのメモリセルにデータを格納する．これを繰り返すことで，任意のビット幅のデータを処理することになる．なお，演算器でのこの一連の処理はパイプライン処理として実行されるため，あるデータを ALU で処理している間に，次のデータを読み込んでいる．また，PE には加算時の桁上げを格納するプリザーブレジスタも組込まれており，このプリザーブレジスタを連想メモリの 1 ビットデータを一時的に保持しておくレジスタとしても利用することも可能となっている．

3. 以上の処理を実行した後，CAMX は CPU に結果を返し，処理が終了する．

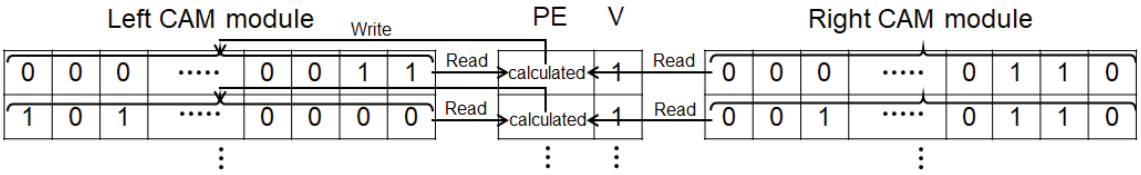


図 2.4: CAMX の基本処理フロー.

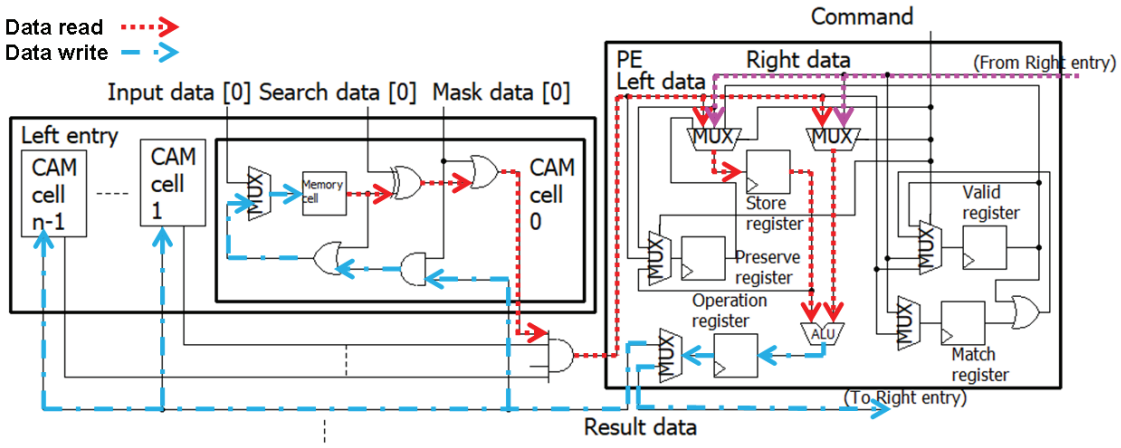


図 2.5: CAMX のメモリセル及び演算器（PE）の詳細.

2.3.2 検索処理のフロー

CAMX は左右に連想メモリが組込まれており，連想メモリに格納されたデータの検索が容易となっている．連想メモリは検索処理に特化したメモリであることが知られており，比較データを用いることで連想メモリ内に格納されたデータに対して一致検索を行うことが可能である [75]–[79]．さらに，マスクデータを併用することで，格納されたデータに対してビット毎に詳細な部分検索も可能である．すなわち，CAMX では一致検索機能に特化した連想メモリを実装しているため，連想メモリ内に格納されたデータに対する検索を得意とする．図 2.6 に CAMX における検索処理の流れを示す．連想メモリのサイズは x ビット $\times$ n エントリである．左右どちらかの連想メモリに対して，格納データに検索データとマスクデータを利用して部分検索を行い，一致した場合に一致信号が出力される．そして，一致した結果はバリッドフラグに格納される．図 2.6 では，マスクデータを利用することで，1 ビットのみを部分検索し，一致したエントリのバリッドフラグにのみ 1 が格納された状態である．バリッドフラグはエントリを動作させるか否かを示すものであり，1 が格納されたエントリは有効化され，次の処理に対して動作可能な状態である．一方，バリッドフラグに 0 が格納されているエントリは，次の処理に対して無動作となる．これにより，CAMX は任意のエントリに対して，データの演算や書き換えが可能となる．

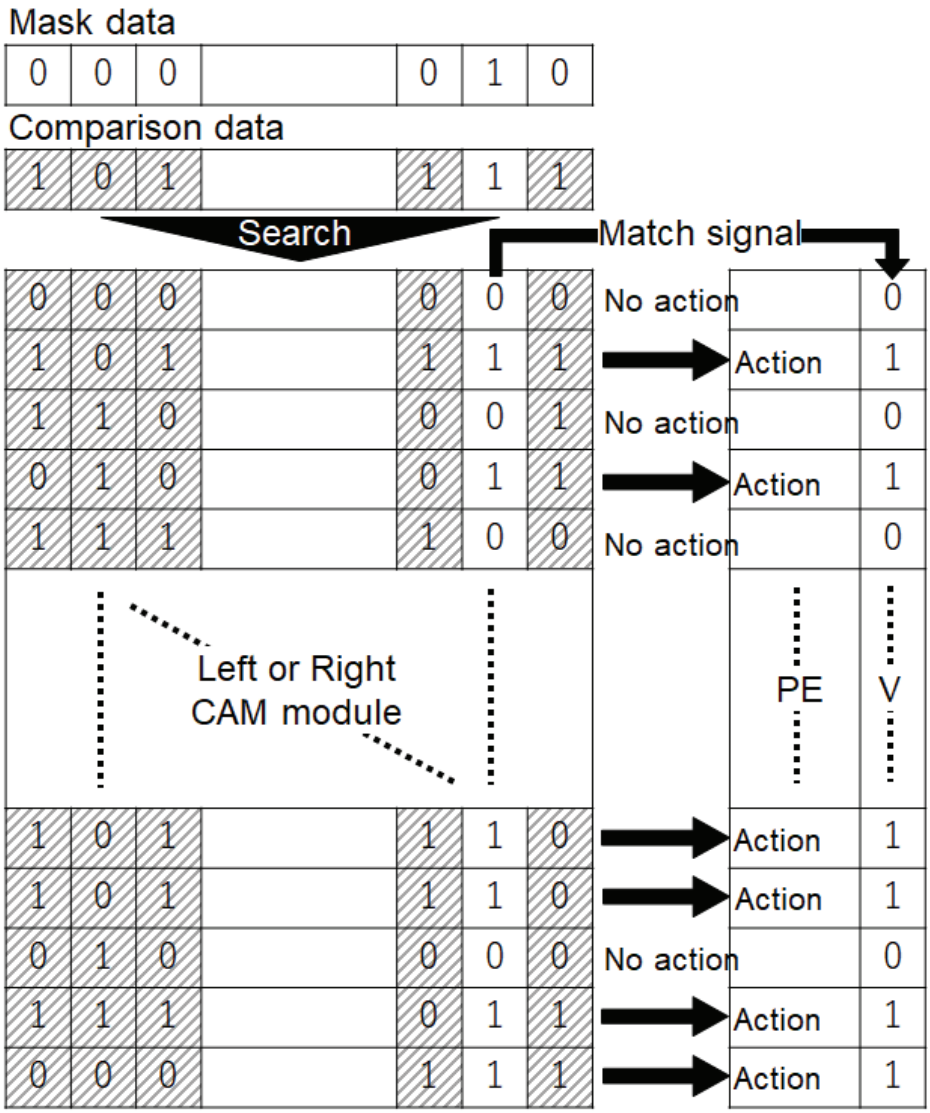


図 2.6: CAMX による検索処理.

2.4 基本演算命令及び検索命令の実装

本節では，CAMX に組込まれている基本演算命令である AND，OR，XOR 及び ADD を実行し，それぞれの処理におけ動作確認及びるクロックサイクル数を確認する．さらに，検索命令も実行し，動作確認及びクロックサイクル数を確認する．これらの検証を行うことで，提案する CAMX の有効性を確認すると共に，正常な動作について確認する．

2.4.1 基本演算命令の実装

本節では，CAMX における基本演算命令の実装を行った．基本演算命令とは，論理演算である AND，OR，XOR 及び加算命令のことであり，これらの命令は CAMX の演算器に組込まれている論理演算器及び加算演算器によって実行される．演算には，Xillinx 社の Vivado v2019.2.1 を利用してシミュレーションを行った．そして，本シミュレーションでは，CAMX に組込まれている連想メモリのサイズを 128 ビット × 1,024 エントリとし，AND，OR，XOR 及び加算それぞれの命令に対して，データの入力を行った後に実行及び演算結果の確認を行う．これらの命令について，演算結果を図 2.7 に示す．CLK はシステムクロック，RST はリセット信号，$BUSY$ は命令実行のタイミング信号，$Lmem$ は左の連想メモリに格納した 1 エントリのデータ，$Rmem$ は右の連想メモリに格納した 1 エントリのデータ，pe_reg は演算器の入力データ，op_reg は演算器の演算結果データである．また，$carry\_reg$ は加算時の桁上げデータ，$search\_data$ は検索命令時の検索データ，$match\_sig$ は検索命令時の一致信号である．なお，演算は 128 ビットデータに対して実行しているが，図 2.7 では簡略化のために 4 ビットで表示している．図 2.7 (a) は AND 命令の実行結果を示し，左の連想メモリに “0011” を入力し，右の連想メモリに “0110” を入力して実行した AND 命令の結果，“0010” が左の連想メモリに出力されていることが確認できた．図 2.7 (b) は OR 命令の実行結果を示し，左の連想メモリに “0011” を入力し，右の連想メモリに “0110” を入力して実行した OR 命令の結果，“0111” が左の連想メモリに出力されていることが確認できた．図 2.7 (c) は XOR 命令の実行結果を示し，左の連想メモリに “0011” を入力し，右の連想メモリに “0110” を入力して実行した XOR 命令の結果，“0101” が左の連想メモリに出力されていることが確認できた．図 2.7 (d) は ADD 命令の実行結果を示し，左の連想メモリに “0011” を入力し，右の連想メモリに “0110” を入力して実行した ADD 命令の結果，“1001” が左の連想メモリに出力されていることが確認できた．以上の結果から，CAMX において，基本的な論理演算及び加算演算が正確に実行可能であることが確認でき，加算時に発生する桁上げ処理も正確に実行され

ていることを確認した．なお，これらの処理は 1,024 エントリ全てに対して同時に実行されるが，図 2.7 では簡略化のためそれぞれ 1 エントリの結果のみ示す．

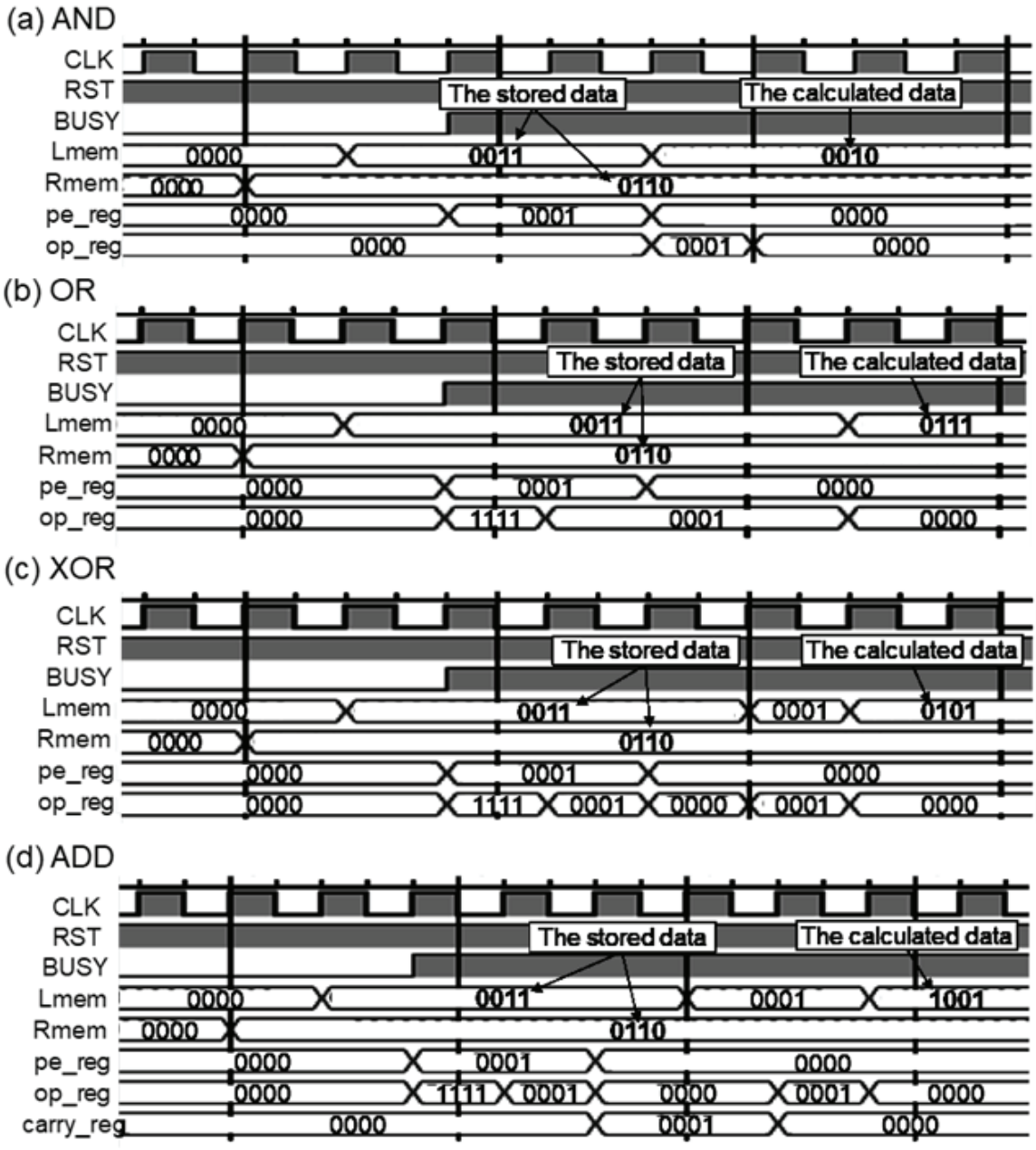


図 2.7: 基本演算命令（AND，OR，XOR 及び ADD）の演算結果.

2.4.2 検索命令の実装

本節では，CAMX を用いた検索命令の実装について検証する．検索命令は，CAMX に組込まれている連想メモリ内において，格納されているデータから検索データを用いて一致検索を実行する処理である．検索命令には，Xillinx 社の Vivado v2019.2.1 を利用してシミュレーションを行った．そして，本実装では，CAMX に組込まれている連想メモリのサイズを 128 ビット × 1,024 エントリとし，検索命令により連想メモリに格納されている 1,024 データ全てを対象に一致検索を行った．検索命令の実行結果を図 2.8 に示す．CLK はシステムクロック，RST はリセット信号，$BUSY$ は命令実行のタイミング信号，$Lmem\_ax$ は左の連想メモリに格納したエントリ毎のデータ（x : エントリの番号を表す）である．また，$search\_data$ は検索命令時の検索データ，$match\_sig\_ax$ は検索命令の一致信号（x : エントリの番号を表す）である．なお，検索命令は 128 ビットデータに対して実行されているが，図 2.8 では簡略化のために 4 ビットで表示している．図 2.8 から，左の連想メモリにはそれぞれのエントリにデータが格納されており，$Lmem\_a0$ には “0011”，$Lmem\_a1$ には “0111”，$Lmem\_a2$ には “0111”，$Lmem\_a3$ には “1000” が格納されている．これらのデータが格納された左の連想メモリ対して，“0111” の検索命令を実行する．すなわち，“0111” が格納されている $Lmem\_a1$ 及び $Lmem\_a2$ のエントリについて一致信号が出力されることになる．図 2.8 の実行結果から，$Lmem\_a1$ 及び $Lmem\_a2$ に格納されている “0111” に対してのみ一致信号である $match\_sig\_a1$ 及び $match\_sig\_a2$ が立ち上がっていることが確認できる．一方，“0011” が格納されている $Lmem\_a0$ 及び “1000” が格納されている $Lmem\_a3$ については，一致信号である $match\_sig\_a0$ 及び $match\_sig\_a3$ は立ち上がっていない．以上の結果から，CAMX において検索命令が実行可能であり，この検索命令は 1,024 エントリに対して並列に検索処理が実行可能であることが確認できた．

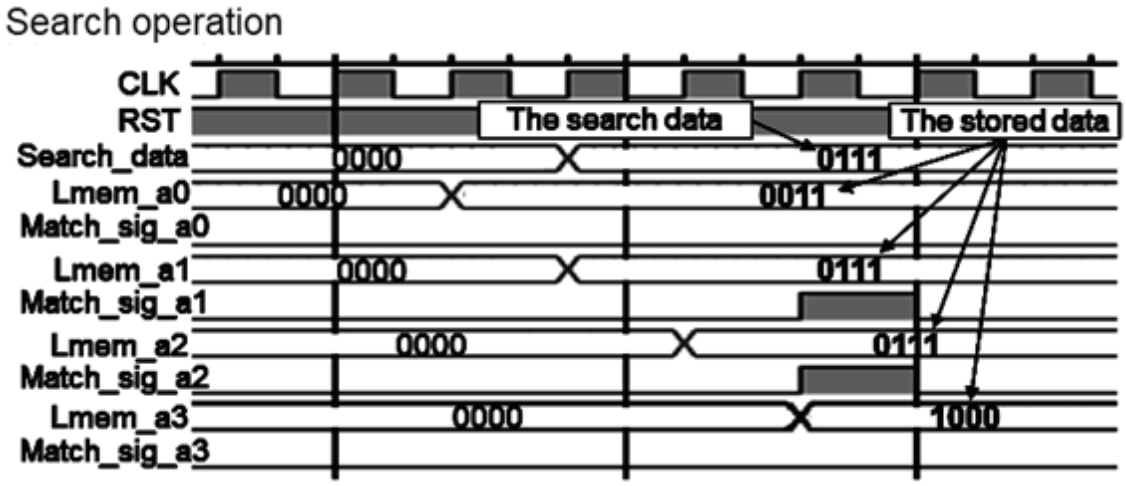
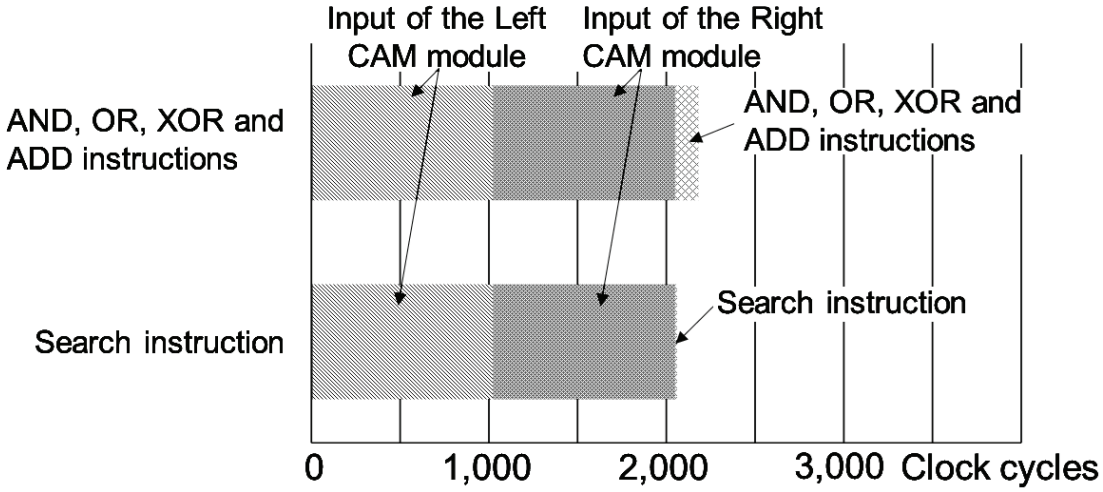


図 2.8: 検索命令の処理結果.

2.4.3 基本演算及び検索命令のクロックサイクル数比較

本節では，2.4.1 節及び 2.4.2 節のシミュレーション結果について，基本演算命令（AND，OR，XOR 及び ADD）及び検索命令それぞれのクロックサイクル数を検証する（図 2.9）．それぞれの命令では，1,024 エントリに対して 128 ビットデータの入力を行っているため，左右の連想メモリにデータを入力するためのクロックサイクル数が必要となり，このデータ入力は左右の連想メモリに対してそれぞれ 1,024 クロックサイクルが必要となる．そして，基本演算命令である AND，OR，XOR 及び ADD 命令を 128 ビットデータに対して実行した場合，それぞれ 130 クロックサイクルが必要であった．さらに，検索命令については，128 ビットデータに対して実行したところ，1 クロックサイクルであった．すなわち，基本演算命令の合計クロックサイクル数は 2,178 であり，検索命令の合計クロックサイクル数は 2,049 であった．なお，それぞれの命令では，1,024 エントリ全てのデータに対して並列に実行されている．よって，CAMX は並列データ数に関係なく，基本演算命令及び検索命令は一定のクロックサイクル数で実行され，基本演算命令は 130 クロックサイクル，検索命令は 1 クロックサイクルであった．



Instructions (1,024 entries and 128-bit data)	Clock cycles
Input of the left CAM module	1,024
Input of the right CAM module	1,024
AND, OR, XOR and ADD	130
Search	1

図 2.9: 基本演算命令（AND，OR，XOR 及び ADD）及び検索命令のクロックサイクル数.

2.5 CAMX における符号付き乗算処理

本節では，CAMX における符号付き乗算処理について検証する．CAMX は小面積かつシンプルな構成であることを研究の目的としているため，演算器には乗算処理のための専用回路は組込んでおらず，2.4 節で示した基本演算命令及び検索命令を組合わせて乗算処理を実行する．一方，マルチメディア処理においては乗算処理が必ず要求される処理であり，乗算処理を高速に実現することは CAMX の有効性を向上させると共に，モバイルデバイスの高性能化に影響を与えることになる．このため，本節では，CAMX における符号付き乗算処理を実行する場合の最適な手法について検討する．

2.5.1 ビットシリアル乗算

本節ではビットシリアル乗算について説明及び検証する．ビットシリアル乗算とは，検索命令を使用せずに，乗数と被乗数を 1 ビットずつ一般的な計算手法により加算していく演算手法である．すなわち，図 2.10 に示すように，x_n と y_n をそれぞれ AND し，シフトしながら加算を繰り返すことになる．なお，符号付き乗算を想定しているため，図 2.10 の様に，4 ビット乗算を実行する場合には拡張した 8 ビット同士の乗算が必要となる．CAMX 上で，このビットシリアル乗算を実行する場合の詳細なアルゴリズムを図 2.11 に示す．図 2.11 は，左の連想メモリに “0011”，右の連想メモリに “0110” を入力した場合の処理手順を示し，“0011” は被乗数，“0110” は乗数となる．

ステップ 1 始めに，乗数及び被乗数を拡張する．“0011” は被乗数であるため，データ毎に左の連想メモリからデータを右の連想メモリへコピーし，データ毎に右の連想メモリに拡張する．そして，“0110” は乗数であるため，ビット毎に乗算のビット幅に合わせて，右の連想メモリから 1 ビットずつ左の連想メモリへコピーすることで，左の連想メモリに拡張する．すなわち，被乗数はデータ毎に右の連想メモリに拡張され，乗数はビット毎に左の連想メモリに拡張される．

ステップ 2 次に，拡張した乗数及び被乗数のデータ同士を AND 演算し，右の連想メモリに演算結果を格納する．すなわち，乗数が 1 である場合は拡張された被乗数のデータがそのまま保持され，乗数が 0 である場合は拡張された被乗数のデータが全て 0 に書き換えられる．

ステップ 3 最後に，右の連想メモリに格納された AND 演算により得られた結果をビッ

トシフトしながら，左の連想メモリに加算していく．全てのデータを加算した結果が乗算処理の結果となる．

上述のビットシリアル乗算処理を実行した結果は図 2.12 となる．このシミュレーション結果は，Xillinx 社の Vivado v2019.2.1 を利用した．そして，本シミュレーションでの CAMX に組込まれた連想メモリのサイズは 128 ビット ×1,024 エントリとした．CLK はシステムクロック，$SETUP$ は処理の準備信号，$BUSY$ は命令実行のタイミング信号，$Lmem$ は左の連想メモリに格納した 1 エントリのデータ，$Rmem$ は右の連想メモリに格納した 1 エントリのデータ，pe_reg は演算器の入力データ，op_reg は演算器の演算結果データである．そして，乗算のビット幅は 4 ビットとし，左の連想メモリに “0011”，右の連想メモリに “0110” を入力し，乗算処理を実行する．なお，図 2.12 は 1 エントリのみ抜粋したシミュレーション結果であり，同様に 1,024 エントリ全てのデータに対しても並列に乗算処理が実行される．この結果，左の連想メモリに乗算結果である “00010010” が格納されていることが確認できた．

$$
\begin{array}{cccccccccc}
 & x_7 & x_6 & x_5 & x_4 & x_3 & x_2 & x_1 & x_0 \\
\times & y_7 & y_6 & y_5 & y_4 & y_3 & y_2 & y_1 & y_0 \\
\hline
 & x_7y_0 & x_6y_0 & x_5y_0 & x_4y_0 & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
 & x_6y_1 & x_5y_1 & x_4y_1 & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 & \\
 & x_5y_2 & x_4y_2 & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 & & \\
 & x_4y_3 & x_3y_3 & x_2y_3 & x_1y_3 & x_0y_3 & & & \\
 & x_3y_4 & x_2y_4 & x_1y_4 & x_0y_4 & & & & \\
 & x_2y_5 & x_1y_5 & x_0y_5 & & & & & \\
 & x_1y_6 & x_0y_6 & & & & & & \\
 & x_0y_7 & & & & & & & \\
\hline
 & p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}
$$

図 2.10: ビットシリアル乗算.

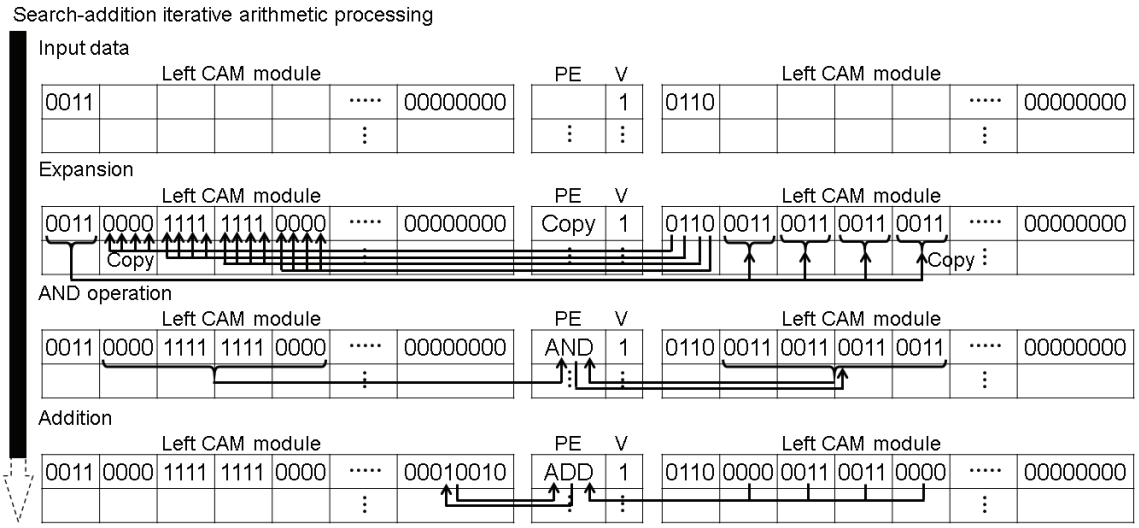


図 2.11: CAMX におけるビットシリアル乗算の手順.

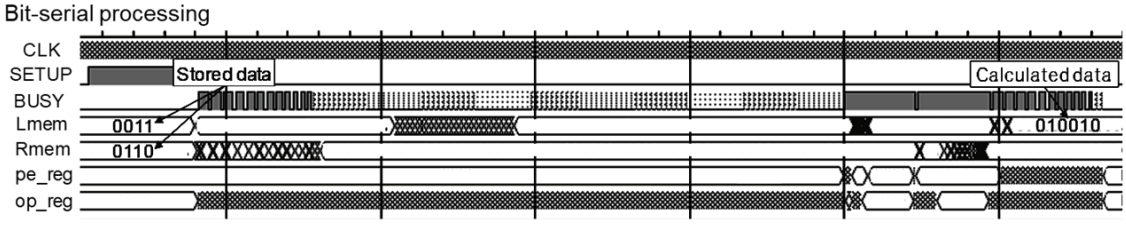


図 2.12: ビットシリアル乗算の結果.

2.5.2 検索・加算繰り返し乗算

本節では，検索・加算繰り返し乗算について確認する．検索・加算繰り返し乗算とは，CAMX の検索命令と加算命令を利用した手法である．CAMX は連想メモリにより構成されているため，全エントリに対して検索命令を並列に実行できる．さらに，検索命令による一致結果を格納できるバリッドフラグも組込まれている．このバリッドフラグを利用することで，CAMX は検索命令により一致したエントリのみを実行することが可能である．この機能を利用することで，乗算処理において，乗数をビット毎に “1” であるか検索命令を実行し，一致した場合にのみ被乗数を加算することが可能となる（図 2.13）．これにより，2.5.1 節で説明した乗算手法よりも命令数及び処理に関するメモリ使用率を削減でき，命令も単純なものとなる．CAMX における処理の流れを図 2.14 に示す．なお，図 2.14 では簡略化のために左の連想メモリに “0011”，右の連想メモリに “0110” を格納し，実行した場合の処理手順を示す．

ステップ 1 始めに，左の連想メモリに被乗数を格納し，右の連想メモリに乗数を格納する．

ステップ 2 次に，右の連想メモリに対して 1 ビットずつ検索命令を実行し，検索命令で検索対象のビットが “1” であるかを確認する．検索命令の結果，一致した場合にはバリッドフラグに “1” を格納し，その該当エントリを次のサイクルで実行可能になるよう有効化する．

ステップ 3 最後に，バリッドフラグに “1” が格納されているエントリに対して加算命令を実行し，結果を右の連想メモリに格納する．

ステップ 4 1~3 の手順を繰り返す．

以上の手順により，4 ビット幅の乗算処理を実行した結果を図 2.15 に示す．CLK はシステムクロック，$SETUP$ は処理の準備信号，$BUSY$ は命令実行のタイミング信号，$Lmem$ は左の連想メモリに格納した 1 エントリのデータ，$Rmem$ は右の連想メモリに格納した 1 エントリのデータ，pe_reg は演算器の入力データ，op_reg は演算器の演算結果データ，$match\_sig$ は検索命令の一致信号である．左の連想メモリに “0011”，右の連想メモリに “0110” を格納し，実行した．なお，図 2.15 は 1 エントリのみ抜粋したシミュレーション結果であるが，1,024 エントリ全てのデータに対して並列に実行される．この結果，右の連想メモリに乗算結果である “00010010” が格納されていることが確認できた．

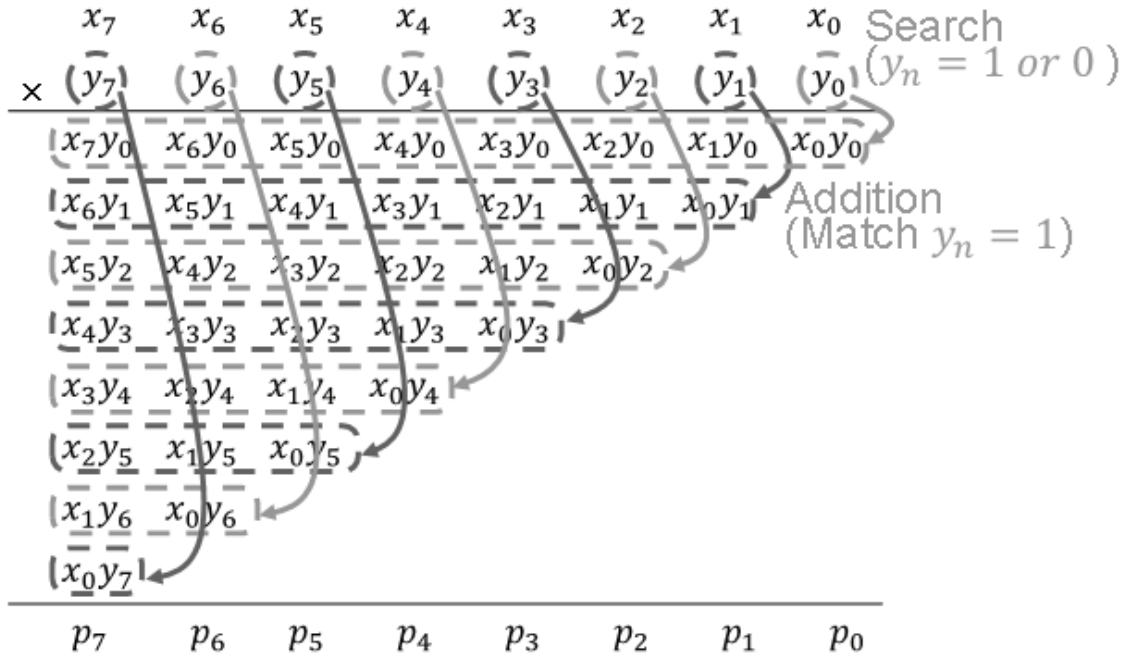


図 2.13: 検索・加算繰り返し乗算.

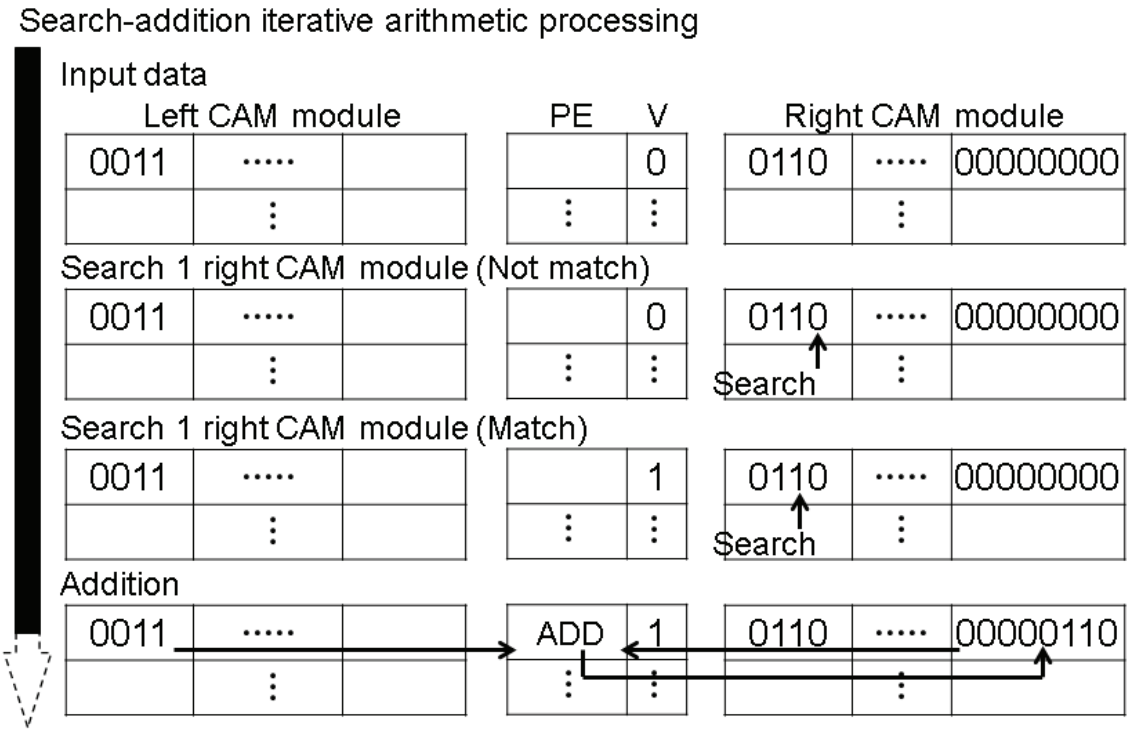


図 2.14: CAMX における検索・加算繰り返し乗算の手順.

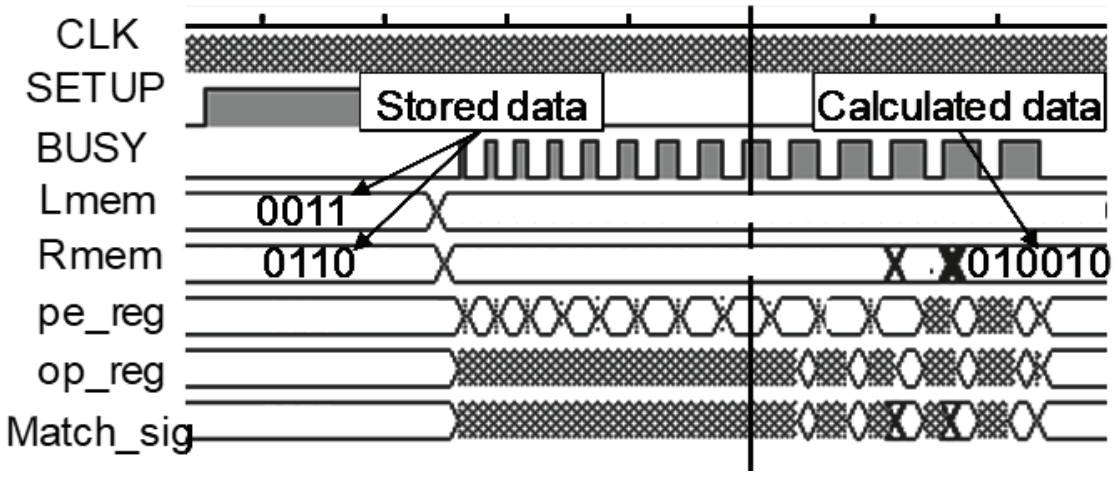
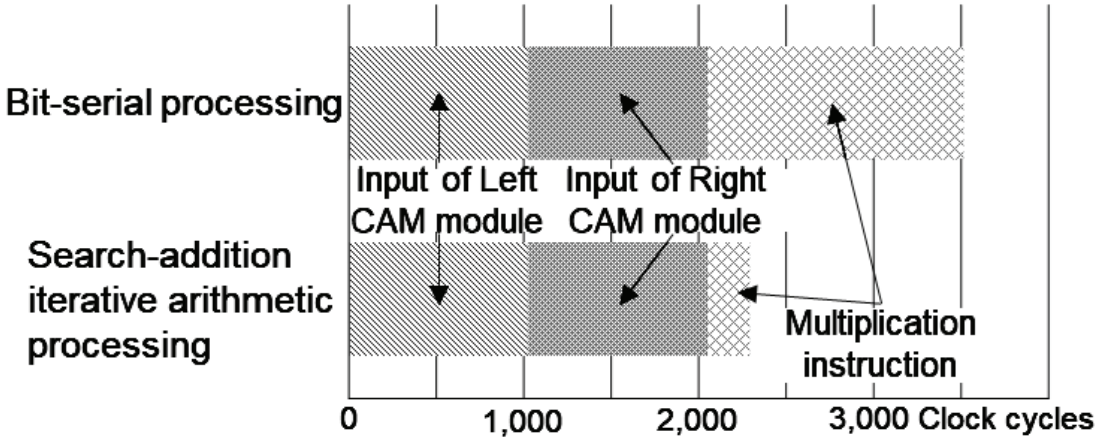


図 2.15: 検索・加算繰り返し乗算の結果.

ここで，ビットシリアル乗算と検索・加算繰り返し乗算の処理速度を比較する．図 2.16 にそれぞれの処理における 4 ビット乗算のクロックサイクル数をまとめる．それぞれの処理では，左右の連想メモリに 1,024 データを格納してから処理しているため，左右の連想メモリにデータを格納するために 1,024 クロックサイクルが必要となる．ビットシリアル乗算は，4 ビット乗算を実行した場合，演算に要するクロックサイクル数は 1,462 クロックサイクルであった．一方，検索・加算繰り返し乗算は，4 ビット乗算を実行した場合，演算に要するクロックサイクル数は 237 クロックサイクルであった．以上の結果から，検索・加算繰り返し乗算はビットシリアル乗算よりも，クロックサイクル数を約 84 %削減できた．大幅にクロックサイクル数を削減できていることから，乗算ビット幅をさらに大きくした場合にも検索・加算繰り返し乗算がビットシリアル乗算よりも高速に処理可能であると考えられる．



Instructions (1,024 entries and 128-bit data)		Clock cycles
Input of the left CAM module		1,024
Input of the right CAM module		1,024
Multiplication	Bit-serial processing	1,462
	Table-lookup operation-used processing	237

図 2.16: ビットシリアル乗算と検索加算繰り返し乗算のクロックサイクル数比較.

2.5.3 Baugh-Wooley 乗算

本節では，Baugh-Wooley 乗算について説明する．Baugh-Wooley 乗算は，n ビットの 2 進数として，A を $a_{n-1}a_{n-2}...a_1a_0$ 及び B を $b_{n-1}b_{n-2}...b_1b_0$ とすると，2 の補数を考慮して式（2.1）で表される [80]–[84]．すなわち，式（2.1）を 4 ビット乗算と考えた場合，図 2.17 に示すように最上位ビットの符号部を考慮した手法となる．

$$
\begin{aligned}
A &= -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i \\
B &= -b_{n-1}2^{n-1} + \sum_{i=0}^{n-2} b_i 2^i \\
A \times B &= a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} a_i b_j \cdot 2^{i+j} \\
&\quad + 2^{n-1}\left(-2^{n-1} + \sum_{j=0}^{n-2} \overline{a_{n-1}b_j} 2^j + 1\right) \\
&\quad + 2^{n-1}\left(-2^{n-1} + \sum_{i=0}^{n-2} \overline{a_{n-1}b_i} 2^i + 1\right)
\end{aligned}
\tag{2.1}
$$

この Baugh-Wooley 乗算を CAMX に実装した場合の処理手順を図 2.18 に示す．

ステップ 1 始めに，左の連想メモリに被乗数を格納し，右の連想メモリに乗数を格納する．

ステップ 2 図 2.17 で示した計算を行うため，まず “1001” を右の連想メモリにおける結果格納用ビットの上位 4 ビットに加算する．その後，以降の計算における前準備として，左の連想メモリに格納した被乗数の最上位ビットを反転する．

ステップ 3 右の連想メモリに格納した乗数をビット毎に検索命令を実行して，一致したエントリに対してのみ左の連想メモリに格納されている被乗数をシフトしながら右の連想メモリに加算命令の結果を格納していく．ただし，検索命令では，0 に一致した場合には左の連想メモリに格納した “1000” を右の連想メモリに加算し，1 に一致した場合には左の連想メモリに格納した被乗数を右の連想メモリに加算する．この処理を乗数の最上位ビットに対して処理を行うまで，繰り返し実行する．

ステップ 4 乗数の最上位ビットに対して検索命令を実行する前に，次の処理を実行する前準備として，左の連想メモリに格納した被乗数の全ビットを反転する．そして，乗数の最上位ビットに対して検索命令を実行し，0 に一致した場合には左の連想メモリに格納した “0111” を右の連想メモリに加算し，1 に一致した場合には左の連想メモリに格納した被乗数を右の連想メモリに加算する．

Baugh-Wooley 乗算を用いた，4 ビット幅の乗算処理を実行した結果を図 2.19 に示す．CLK はシステムクロック，$SETUP$ は処理の準備信号，$BUSY$ は命令実行のタイミング信号，$Lmem$ は左の連想メモリに格納した 1 エントリのデータ，$Rmem$ は右の連想メモリに格納した 1 エントリのデータ，pe_reg は演算器の入力データ，op_reg は演算器の演算結果データ，$match\_sig$ は検索命令の一致信号である．左の連想メモリに “0011”，右の連想メモリに “0110” を格納し，実行した．なお，図 2.19 は 1 エントリのみ抜粋したシミュレーション結果であるが，1,024 エントリ全てのデータに対して並列に実行される．この結果，右の連想メモリに乗算結果である “00010010” が格納されていることが確認できた．

ここで，CAMX における最適な乗算手法を検討するため，2.5.2 節で示した検索・加算繰り返し乗算と Baugh-Wooley 乗算について比較する．図 2.20 にそれぞれのクロックサイクル数の変化を示す．横軸は，乗算のビット幅であり、4 × 32 ビット乗算までの実行結果を示している．縦軸は，乗算処理を実行した時のクロックサイクル数であり，左右の連想メモリにそれぞれ 1,024 データを入力し，出力するまでの数となる．この結果から，15 ビットを境に，乗算処理に要するクロックサイクル数は検索・加算繰り返し乗算と Baugh-Wooley 乗算で反転していることが確認できた．すなわち，CAMX において 15 ビットを超える乗算処理を実行する際は，Baugh-Wooley 乗算を用いて実行した方がより高速になり，15 ビットまでの乗算処理を実行する際は，検索・加算繰り返し乗算を用いて実行した方がより高速となる．これは，乗算処理において，加算時の回数及びビット幅が影響しており，加えて，加算命令や検索命令を実行する時に命令を実行可能な状態にするまでの処理時間が影響していると考えられる．CAMX の処理では，加算命令を実行する場合，加算するデータのビット幅が増加するとクロックサイクル数は増加するため，加算するデータのビット幅が小さいほどクロックサイクル数は減少する [58]．また，加算命令の回数が減少すれば，同様にクロックサイクル数も減少する．すなわち，この加算命令を実行する場合には，命令を実行するための準備処理が必要となり，それらの時間もクロックサイクル数も影響する．この命令を実行するための準備処理時間は検索命令にも影響する．このため，15 ビット以上の乗算では，検索・加算繰り返し乗算は Baugh-Wooley 乗算よりも加算の回数及びビット幅が増加し，全体的なクロックサイクル数も増加したと考えられる．さらに，検

索・加算繰り返し乗算は検索命令の後に加算命令を繰り返し実行することになるが，Baugh-Wooley 乗算は加算命令の後に加算命令時の桁上げデータの一時的な格納処理等を行うことで，命令間のクロックサイクル数が減少したと考えられる．以上の理由から，15 ビットを境に検索・加算繰り返し乗算と Baugh-Wooley 乗算のクロックサイクル数が反転し，15 ビット以上の乗算処理においては Baugh-Wooley 乗算を利用した方が高速な処理になる．

				x_3	x_2	x_1	x_0
$\times$				y_3	y_2	y_1	y_0
				$\overline{x_3y_0}$	x_2y_0	x_1y_0	x_0y_0
			$\overline{x_3y_1}$	x_2y_1	x_1y_1	x_0y_1	
		$\overline{x_3y_2}$	x_2y_2	x_1y_2	x_0y_2		
	$\overline{x_3y_3}$	$\overline{x_2y_3}$	$\overline{x_1y_3}$	$\overline{x_0y_3}$			
1	0	0	1				
p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

図 2.17: Baugh-Wooley 乗算.

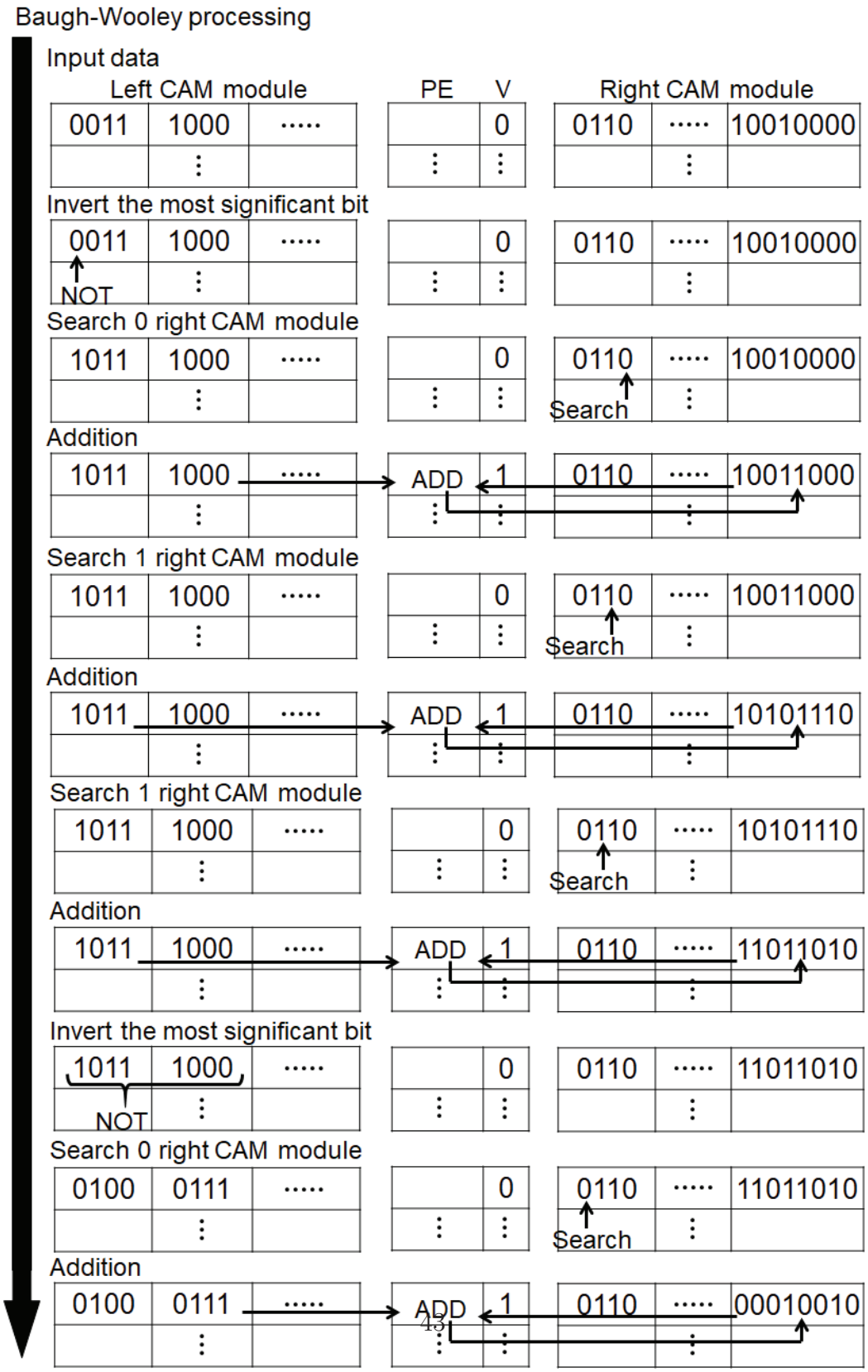


図 2.18: CAMX における Baugh-Wooley 乗算の手順.

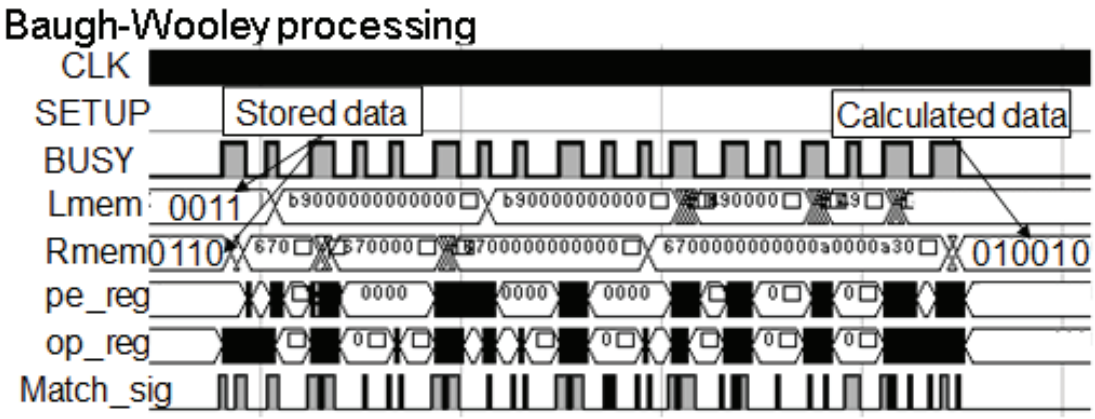


図 2.19: Baugh-Wooley 乗算の結果.

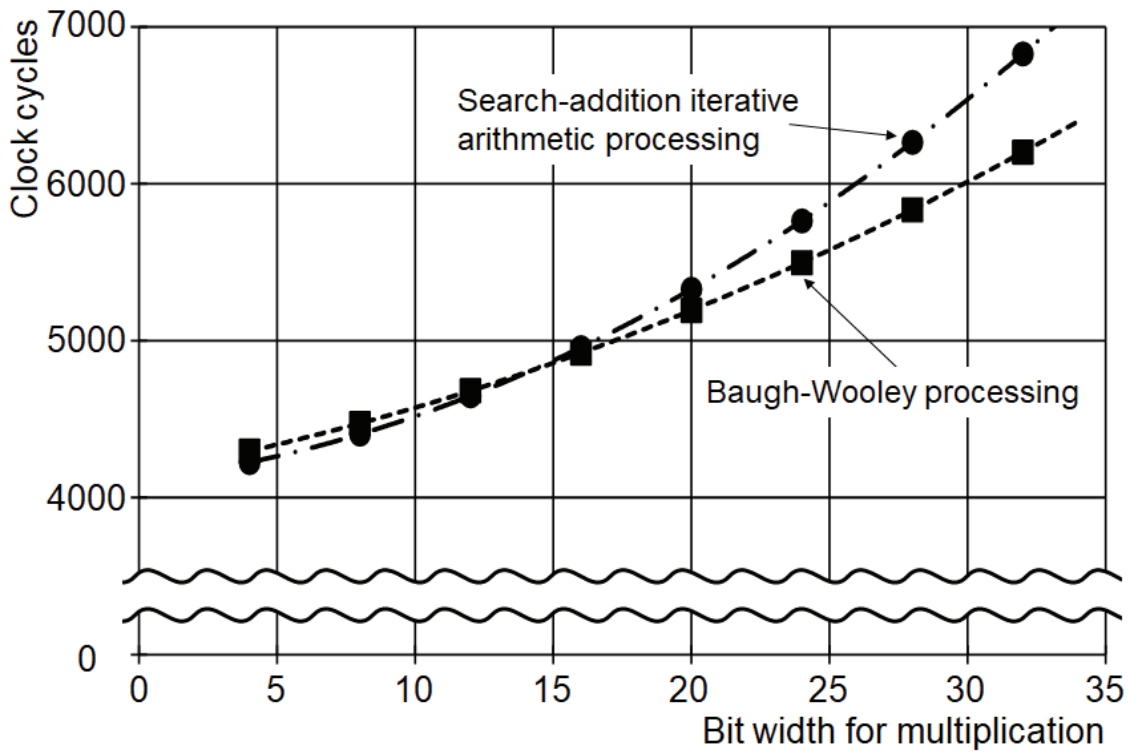


図 2.20: 検索・加算繰り返し乗算と Baugh-Wooley 乗算の乗算ビット幅によるクロックサイクル数の比較.

2.6 暗号化処理（AES）を利用したCAMXとMX-1の性能比較

本節では，暗号化処理として知られているAESをCAMXに実装し，有効性を検証する．なお，MX-1はこれまでにモバイルデバイスと同様の環境でAESを実装及び有効性を検証しているため，MX-1とCAMXを比較することでCAMXの有効性を検証する．

2.6.1 暗号化処理（AES）について

暗号化処理の一つとしてAES (Advanced Encryption Standard) が知られている [44], [85]．AESは共通鍵暗号ブロック暗号化方式であり，通信等において利用さている [86]．AESで利用する暗号化するための鍵は128，192，256ビットがあるが，本節では128ビットの暗号鍵を利用する．

ここで，AESの処理手順について説明する．AESを実行するには4種類の処理を実行する必要があり，それぞれSubBytes，ShiftRows，MixColumns及びAddRoundKeyである．これらの処理を10回繰り返すことでデータを暗号化することが可能となる．以下にそれぞれの詳細について説明する．

1 SubBytes

暗号化対象のデータを8ビット毎に，非線形変換であるS-box（図 2.21）を用いて変換する処理である．この8ビットの上位4ビット（x）と下位4ビット（y）に対してS-boxにより変換を行う．ただし，図 2.21は16進数で表示している．

2 ShiftRows

暗号化対象のデータを並べ替える処理である．図 2.22に示すように，暗号化対象のデータを行列と考え，1行目から4行目を左方向に巡回シフトする．行毎にシフト回数も異なる．

3 MixColumns

暗号化対象のデータを式（2.2）で示すように，ガロア体理論に基づいて行列演算を行う．すなわち，行列式の和はXORされることになり，乗算は規約

多項式を法とする多項式環上により演算が行われる．さらに，オーバフローする場合には，16 進数で “1b” を XOR することになる．

$$\begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \times \begin{pmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{pmatrix} \tag{2.2}$$

4 AddRoundKey

暗号化対象のデータと暗号鍵を排他的論理和（XOR）する処理である．

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	a	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	b	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	c	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	d	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	e	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	f	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

図 2.21: AES で使用する S-box.

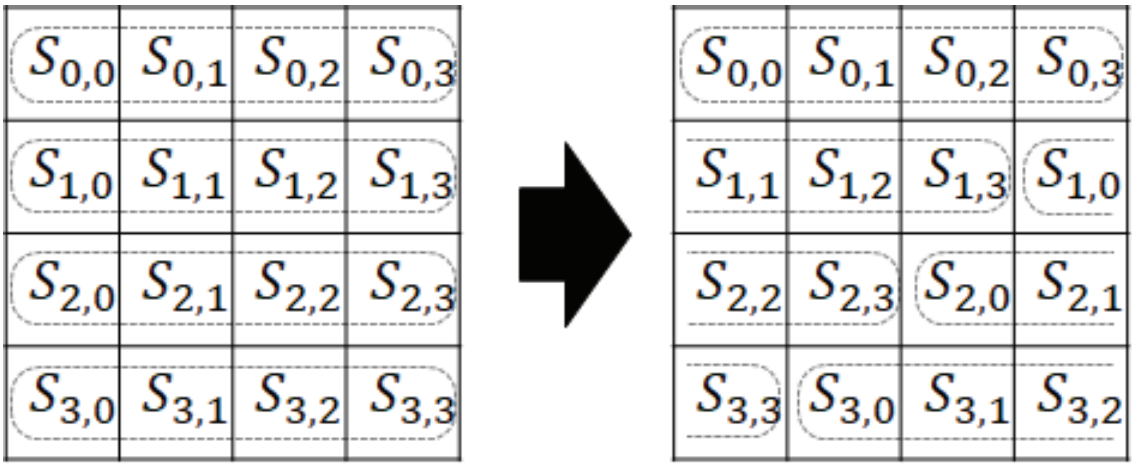


図 2.22: AES における ShiftRows.

2.6.2 CAMX における AES 実装

CAMX における AES の実装手法について説明する．図 2.23 に，AES の処理フローを示す．なお，図 2.23 では，AES における 1 ラウンドのみ示す．この処理を 10 回繰り返すことになる．

ステップ 1 まず初めに，左の連想メモリに暗号対象のデータを入力し，右の連想メモリに暗号鍵を入力する．

ステップ 2 次に AddroundKey を実行するため，左右の連想メモリに対して XOR 処理を実行する．実行した結果は左の連想メモリに格納する．

ステップ 3 AddroundKey の結果に対して，SubBytes 処理を実行する．左の連想メモリに対して，S-box を基に上位 4 ビット及び下位 4 ビットを検索し，一致するエントリに対して S-box に一致する数値に変換する．

ステップ 4 SubBytes において変換された結果に対して，ShiftRows 及び MixColumns を同時に行う．まず，MixColumns を実行するために，左の連想メモリに格納されたデータを 8 ビット毎に，3 倍，2 倍及び 1 倍した後に，XOR 処理により右の連想メモリに結果を格納する．この結果を右の連想メモリに格納する時に，データのシフト処理も行う．

ステップ 5 ステップ 1~4 を繰り返し実行する．

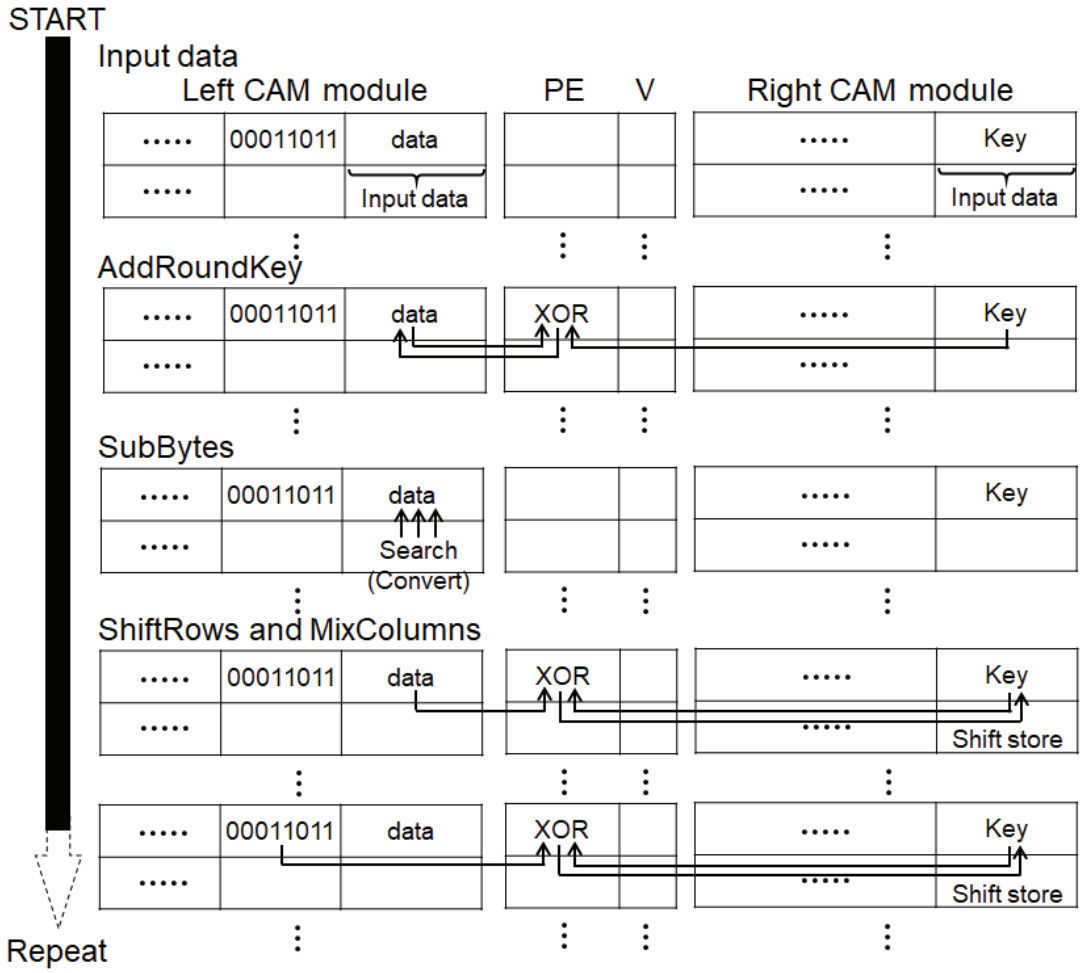


図 2.23: CAMX における AES の手順.

2.6.3 CAMX における AES の実行結果

本節では，CAMX における AES の実装結果を示す．本実験では，Xillinx 社の Vivado v2019.2.1 を利用してシミュレーションを行った．そして，本シミュレーションでは，CAMX に組込まれている連想メモリのサイズを 256 ビット × 1,024 エントリとし，1,024 個のデータに対して並列に AES 処理を実行する．CAMX において AES を実装し，正確に実行できることを確認した（図 2.24）．そして，実行に必要な総クロックサイクル数は 1,362,699 クロックサイクルであることを確認でき

た．さらに，このクロックサイクル数を詳細に検証する．AES 処理における 10 ラウンド合計のクロックサイクル数を表 2.6.3 に示す．SubBytes は 1,312,160 クロックサイクルであり，ShiftRows と MixColumns の合計は 17,161 クロックサイクルであった．ShiftRows と MixColumns は CAMX においては同時実行されるため，表 2.6.3 ではそれぞれの合計クロックサイクル数を示した．そして，AddRoundKey は 2,519 クロックサイクルであり，その他の入出力処理は 30,859 クロックサイクルであることを確認した．すなわち，SubBytes が最もクロックサイクル数を要していることが確認できる．これは，8 ビット毎に検索及び変換が必要となり，本節で実装した処理では 8 ビット毎に S-box 検索を行った後にデータの書き換えを行っているためにクロックサイクル数が大きくなっている．今後この検索及びデータ書き換えを複数のデータに対して同時に実行することでさらなる高速化を実現可能であると考えているが，本論文では，単純な AES 処理に対して比較・検証を行うため，8 ビット毎に検索及びデータ書き換えを行った．

図 2.24: CAMX における AES のシミュレーション結果.

表 2.2: CAMX における AES 処理のクロックサイクル数（10 ラウンド）.

	SubBytes	ShiftRows, MixColumns	AddRoundKey	Other (Input, Output)	Total
CAMX	1,312,160	17,161	2,519	30,859	1,362,699

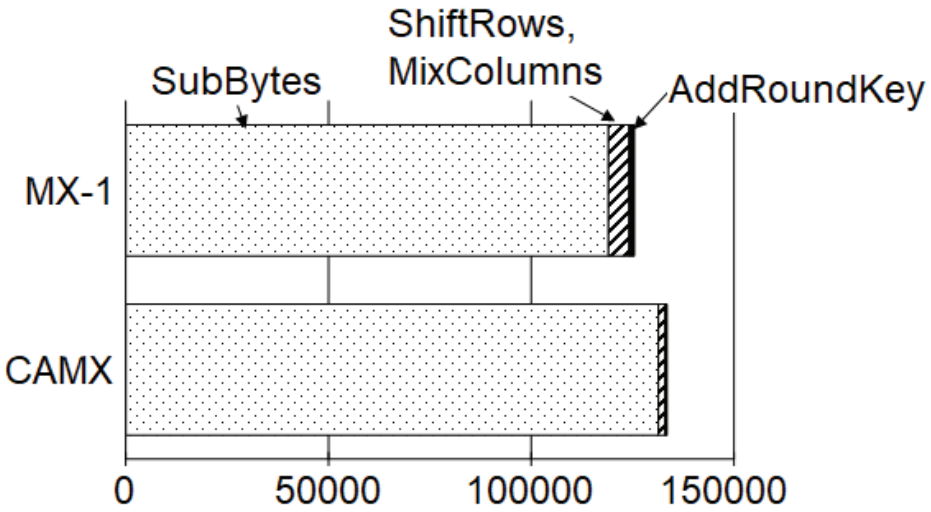
2.6.4 CAMX と MX-1 の実行結果比較

これまで我々は MX-1 を利用した AES 実装をモバイルデバイスと同様の環境で行ってきた [65]．この MX-1 の結果と 2.6.3 節で示した CAMX の結果を比較する．図 2.25 に MX-1 及び CAMX におけるクロックサイクル数の内訳を示す．なお，MX-1 と CAMX に関して，AES 処理の 1 ラウンドにおける各処理の詳細なクロックサイクル数を比較するために，図 2.25 では AES 処理の 1 ラウンドにおけるクロックサイクル数を示している．図 2.25 から，MX-1 については，総クロックサイクル数が 125,345 クロックサイクルであった．そして，SubBytes は 119,009 クロックサイクルで，ShiftRows と MixColumns の合計は 5,350 クロックサイクルで，AddRoundKey は 986 クロックサイクルであった．CAMX については，総クロックサイクル数が 133,232 クロックサイクルであった．そして，SubBytes は 131,216 クロックサイクルで，ShiftRows と MixColumns の合計は 1,880 クロックサイクルで，AddRoundKey は 136 クロックサイクルであった．この結果から，総クロックサイクル数としては，MX-1 の方が CAMX よりも約 5 %の高速となっている．一方，それぞれの内訳について確認すると，ShiftRows と MixColumns の合計クロックサイクル数は CAMX が MX-1 よりも約 65 %も高速に処理できている．さらに，AddRoundKey のクロックサイクル数は CAMX が MX-1 よりも約 86 %も高速に処理できている．このため，CAMX が MX-1 よりも SubBytes において 9 %の処理低下があり，総クロックサイクル数は CAMX の方がクロックサイクル数が増加した．しかし，今回実装した CAMX の AES 処理については，単純な実装を行ったことからクロックサイクル数は大きくなり，今後 SubBytes において検索及びデータ書き換えを同時に実行する等の工夫により高速化は可能であると考えている．以上の結果から，CAMX の総クロックサイクル数は MX-1 より大きくなっているが，基本演算においては CAMX の方が高性能となっることが確認できた．

ここで，モバイルデバイスにおいて利用されている既存のプロセッサと CAMX の性能について比較する．図 2.26 にそれぞれのプロセッサにおける処理バイト当たりのクロックサイクル数を示す．比較対象のプロセッサは，Intel Atom N270 (1.60 GHz)，AMD Geode LX800 (500 MHz)，Advanced RISC Machines (ARM) Cortex A8 core 32K/32K を組込んだ Texas Instruments (TI) DM3730 (1.00 GHz) 及び ARM Cortex A8 core 16K/16K を組込んだ TI OMAP3530 (720 MHz) とする．それぞれのプロセッサにおいて，AES 処理を実装し，処理時間を計測した．その処理時間から処理バイト当たりのクロックサイクル数を算出した．また，CAMX については，Xillinx 社の Vivado v2019.2.1 を利用してシミュレーションを行い，1,024 個のデータを並列に AES 処理を行った．そして，このクロックサイクル数から処理バイト当たりのクロックサイクル数を算出した．

図 2.26 から，CAMX は全てのモバイル向けプロセッサよりも高性能であることが確認できる．特に，TI OMAP3530 （ARM Cortex A8）との比較においては，CAMX の方が約 53 %も高性能であることが確認できた．さらに，図 2.26 において最も高性能であった AMD Geode LX800 と比較しても，CAMX は約 15 %もの性能向上が実現できている．

以上の結果から，CAMX は MX-1 とほぼ同等の性能を実現しつつ，既存のモバイル向けプロセッサよりも高性能に処理可能であることが確認できた．すなわち，CAMX はモバイルデバイスに組込むことで，モバイルデバイスのマルチメディア処理を高速かつ高性能に実行できると考えられる．



	SubBytes	ShiftRows, MixColumns	AddRoundKey	Total
MX-1	119,009	5,350	986	125,345
CAMX	131,216	1,880	136	133,232

図 2.25: CAMX と MX-1 における AES 処理のクロックサイクル数比較（1 ラウンド）．

Processor	The number of AES encryption clock cycles/bytes
Intel Atom N270	115.79
AMD Geode LX800	97.63
TI DM3730 (ARM Cortex A8)	171.27
TI OMAP3530 (ARM Cortex A8)	175.85
CAMX (1,024 parallel), This work	83.17

図 2.26: CAMX と既存のモバイルデバイス向けプロセッサの性能比較.

2.7 CAMXにおける浮動小数点を利用した加算処理の実装及び評価

2.6 節において，CAMX は MX-1 と同等の性能を持ち，既存のモバイルデバイス向けプロセッサよりも高性能であることを確認した．本節では，高性能な CAMX を利用して，単精度浮動小数点を利用した加算処理を実装し，検証する．浮動小数点は様々なマルチメディア処理において必要な処理であり，デジタル演算において小数点計算を扱う際に利用される処理である．

2.7.1 浮動小数点を利用した加算処理について

本節では，単精度浮動小数点を利用した加算処理について説明する．単精度浮動小数点は IEEE754 により定義されており，一般的に式 (2.3) で表される．S は符号部，M は仮数部，E は指数部と呼ばれる．符号部は 1 ビットで表され，正数又は負数を示す．指数部は 8 ビットで表され，小数点の位置を示す．仮数部は 23 ビットで表され，データの大きさを示す．これらの値を 2 進数で 32 ビット表現したものが浮動小数点を利用した表現となる．この浮動小数点表現による加算処理を行う場合には，加算処理対象のデータ同士の指数部を用いて小数点の位置合わせを行い，その後，仮数部を用いて加算を行うことになる．

$$(-1)^S \times (1.M) \times 2^{(E-127)} \tag{2.3}$$

2.7.2 CAMXにおける単精度浮動小数点を利用した基本加算処理の実装

単精度浮動小数点を利用した基本加算処理手順

本節では，CAMX における単純な実装による単精度浮動小数点を利用した加算処理について説明する．図 2.27 に CAMX における単純な実装による浮動小数点の加算処理について示す．CAMX の演算器は，小型化及び汎用性のために，浮動小数点を処理するための専用回路は実装されていない．そのため，基本演算命令及び検索命令を用いることで実現する必要がある．CAMX において，浮動小数点の加算処理を実装する場合，加算時の桁上がり処理に対応するため，符号部は 1 ビット，指数部は 9 ビット，仮数部は 25 ビットとして処理を行う．なお，浮動小数点の加算処理は 1,024 エントリに対して並列に実行される．以下に，CAMX におけ

る浮動小数点の加算処理手順を示す．

ステップ 1 始めに，左右の連想メモリに浮動小数点加算の対象データを入力し，それぞれの指数部及び仮数部を処理が行われるビット位置にコピーする．そして，コピーした指数部同士に対して減算処理を実行する．ただし，CAMX には減算を行う専用回路も組込まれていないため，減算処理を実行するには右の連想メモリの指数部に対して 2 の補数を作成してから左右の指数部を加算する必要がある．この減算処理において，指数部の差分を算出する理由は，浮動小数点の加算処理を実行する対象データ同士に対して小数点位置を揃えるためである．

ステップ 2 ステップ 1 での減算処理結果の正負を検索命令により判断する．そして，小数点位置を揃えるために，減算処理で対象データの大小がわかることから，小さい方のデータに差分結果を加算することで小数点位置を揃える．

ステップ 3 ステップ 1 での減算処理結果における正負に対して別々の処理を行う．ただし，それぞれの処理においても仮数部同士を加算する前には丸め処理を行う．丸め処理は最近接丸めとする．ステップ 1 での減算処理結果が負数の場合には，丸め処理後に仮数部をシフトしながら右の連想メモリに加算処理の結果を格納する．ステップ 1 での減算処理結果が正数の場合には，丸め処理後に仮数部をシフトしながら左の連想メモリに加算処理の結果を格納してから右の連想メモリにコピーする．最後に符号部を加算し，左の連想メモリに格納する．さらに，仮数部の加算結果を IEEE754 に従うよう，仮数部同士の加算結果による桁溢れ等を考慮し，仮数部のシフト及び指数部への加算を行う．単精度浮動小数点を利用した基本加算処理の結果は左の連想メモリに格納される．

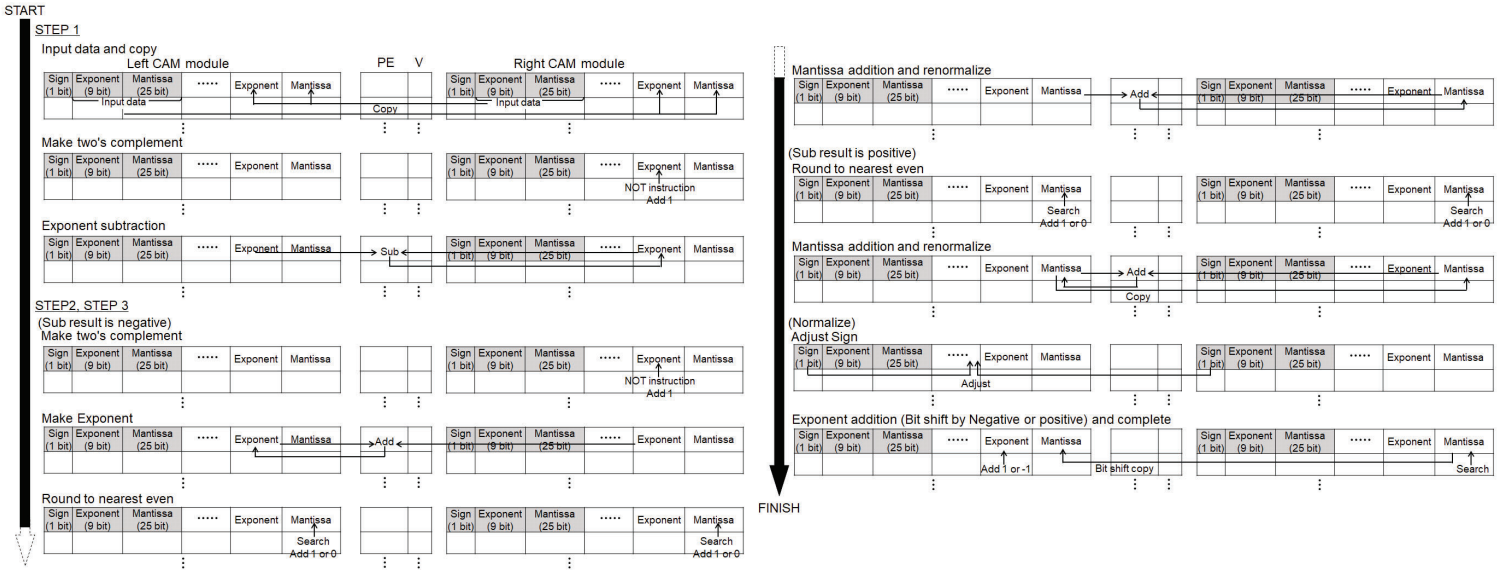
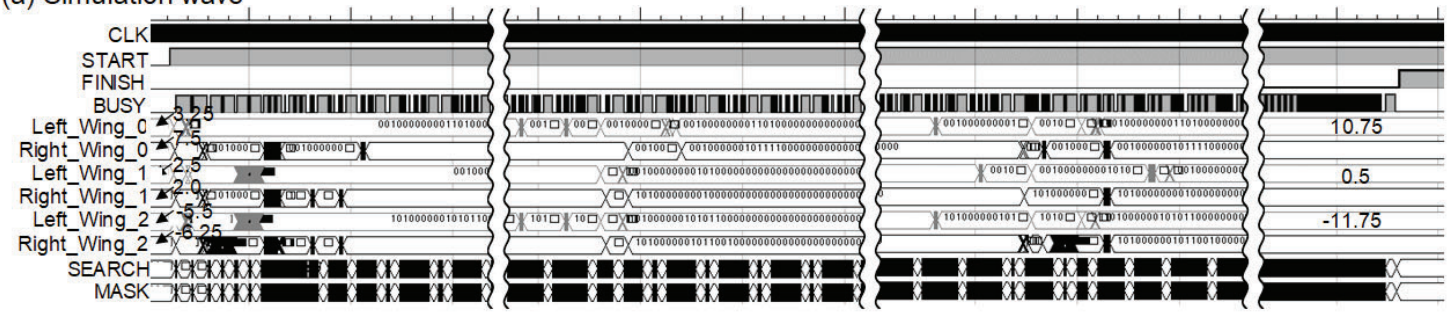


図 2.27: CAMX における単精度浮動小数点による加算処理の手順.

CAMX における単精度浮動小数点を利用した基本加算処理の結果

単精度浮動小数点を利用した基本加算処理を CAMX に実装し，シミュレーションを行う．シミュレーションは Xillinx 社の Vivado v2019.2.1 を利用して実行する．そして，CAMX に組込む連想メモリのサイズは 128 ビット × 1,024 エントリとする．図　2.28 にシミュレーション結果を示す．CLK はシステムクロック，$START$ は処理の開始信号，$FINISH$ は処理の終了信号，$BUSY$ は命令実行のタイミング信号，$Left\_Wing\_n(n$: エントリの番号) は左の連想メモリに格納した 1 エントリのデータ，$Right\_Wing\_n(n$: エントリの番号) は右の連想メモリに格納した 1 エントリのデータ，$SEARCH$ は検索データ，$MASK$ は検索のマスクデータである．シミュレーションでは，3 条件について実行し，それぞれの処理が正確に処理されることを確認する．それぞれ，10 進数で左の連想メモリに “3.25” と右の連想メモリに “7.5” の加算，左の連想メモリに “2.5” と右の連想メモリに “2.0” の加算，左の連想メモリに “-5.5” と右の連想メモリに “-6.25” の加算を実行する．図 2.28 (a) から，それぞれの結果は，“10.75”，“0.5” 及び “-11.75” であり，正確に計算できていることを確認した．図　2.28 (b) に実行した結果のクロックサイクル数を示す．左右の連想メモリにおける入出力時のクロックサイクル数は合計 4,096 クロックサイクルであり，単精度浮動小数点を利用した基本加算処理は 13,580 クロックサイクルであった．この処理を RaspberryPi4 [87], [88] に組込まれている ARM コアと同等の CPU クロック周波数（1.5 GHz）で実行すると考えた場合，113.1 MFLOPS (Mega Floating-point Operation Per Second) で実行可能となる．

(a) Simulation wave



(b) Clock cycle and instruction FLOPS

Instructions (1,024 entries and 128-bit data)	Clock cycles	MFLOPS (1.5 GHz operation frequency)
Input of the left CAM module	1,024	
Input of the right CAM module	1,024	
Output of the left CAM module	1,024	
Output of the right CAM module	1,024	
Other	13	
Floating-point Addition	13,580	113.1
Total	17,689	86.8

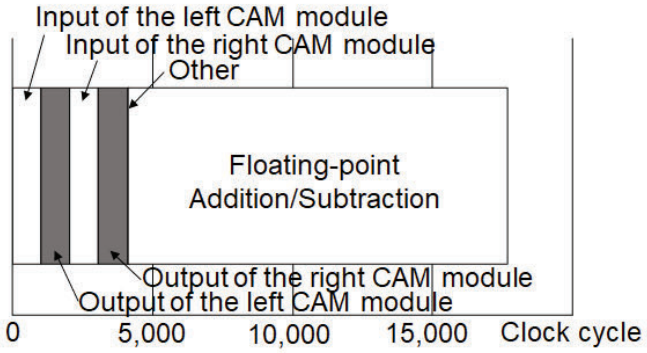


図 2.28: CAMX における単精度浮動小数点による加算処理の結果.

2.7.3 CAMX における 2 の補数削減処理による減算手法

浮動小数点による加算処理においては，減算処理が必要となる．減算処理を実行するには，データに対して 2 の補数を求める必要があり，減算するデータに対してビット反転及び 1 の加算が必要となる．その後，処理対象データ同士を加算する必要がある．しかし，これらの処理を実行すると加算処理とは異なり，減算毎にビット反転及び 1 の加算による余分なクロックサイクル数が発生する．そこで，図 2.29 の様に命令数を削減する手法を提案する．通常，CAMX においては 2 の補数を求めるには図 2.29 (a) の様に処理され，上述したようにビット反転及び 1 の加算後に対象データ同士の加算が必要となる．そこで，CAMX に組込まれているプリザーブレジスタを利用し，2 の補数における処理を削減する手法を提案する．プリザーブレジスタは加算処理を行う際に，加算時の桁上げデータを一時的に保存するレジスタである．このプリザーブレジスタに対して，連想メモリに格納された 1 ビットデータを入力できるようにし，2 の補数を求める際の 1 の加算を減算対象データ同士を加算する時に桁上げと考えて同時に実行する （図 2.29 (b)）．この手法により，CAMX における命令数を減算毎に 1 回削減することが可能となる．

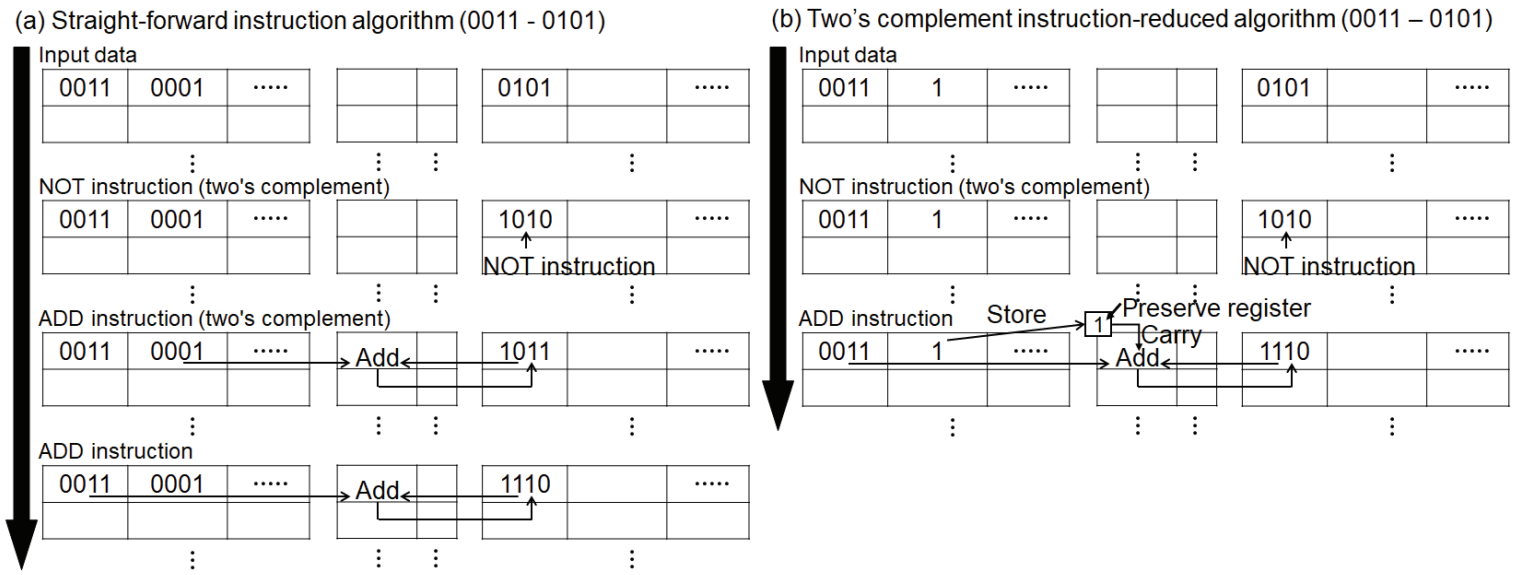


図 2.29: CAMX における 2 の補数処理の削減手法.

2.7.4 2 の補数削減処理による単精度浮動小数点を利用した加算処理について

2 の補数削減処理による単精度浮動小数点を利用した加算処理の手順

2.7.3 節で説明した 2 の補数処理を削減する手法を 2.7.2 節で説明した浮動小数点処理に適用する．図 2.30 に処理手順を示す．本処理においては，減算処理を削減することで命令数を減らすことが実現可能となっている．また，命令の記述も併せて整理する．図 2.30 に示すように，処理手順については，減算処理の削減以外は図 2.27 と同様の手順となる．

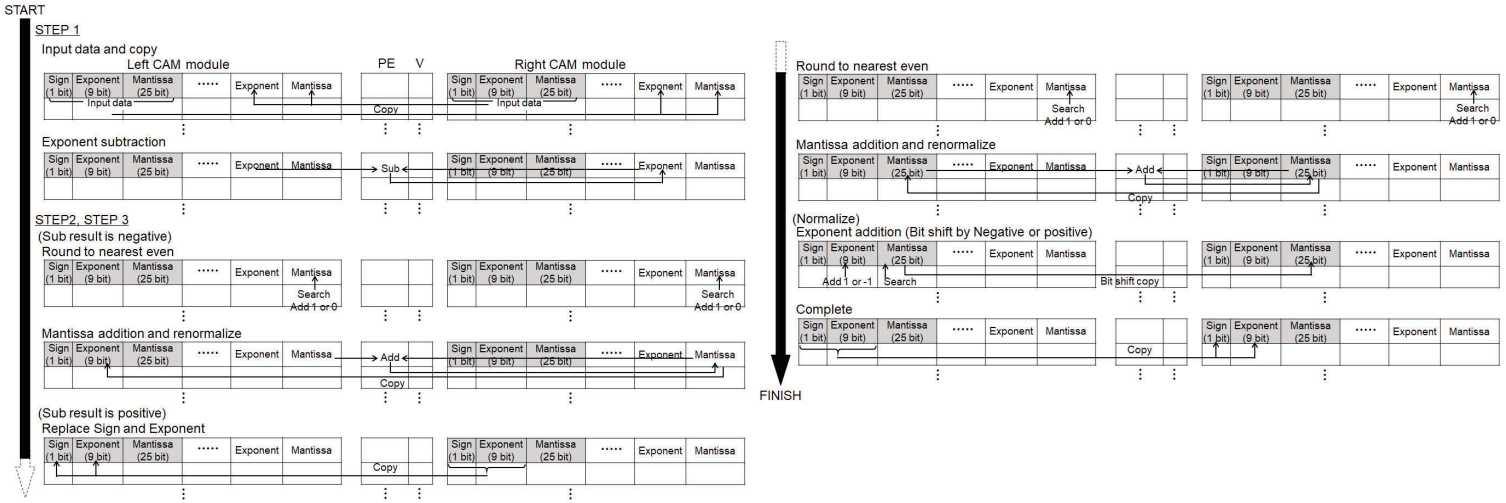
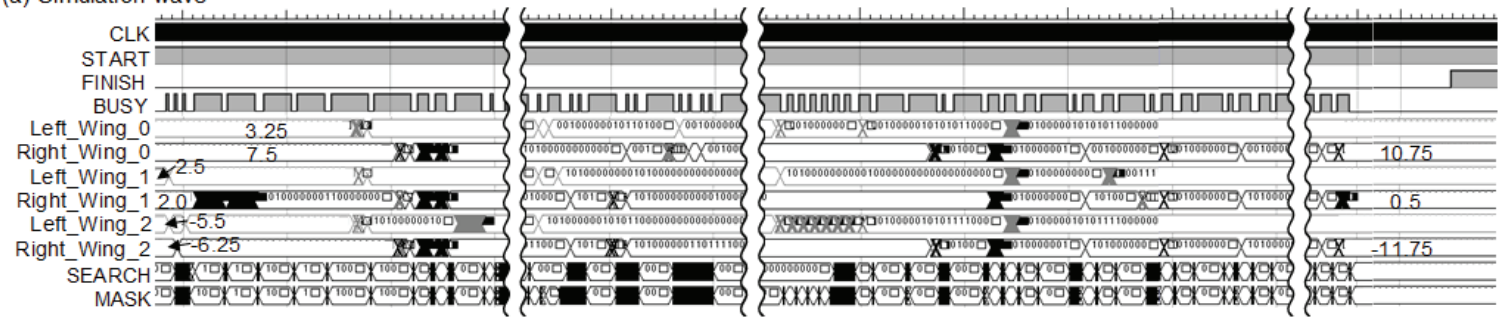


図 2.30: 2 の補数処理削減による単精度浮動小数点を利用した加算処理の手順.

2 の補数削減処理による単精度浮動小数点を利用した加算処理の結果

本節では，2 の補数の命令数を削減する手法により，単精度浮動小数点を利用した加算処理を CAMX に実装し，シミュレーションを行う．シミュレーションは 2.7.2 節と同等の条件で実行し，処理対象データも同様とする．シミュレーション結果を図 2.31 に示す．図 2.31 (a) から，正確に単精度浮動小数点を利用した加算処理が実行されていることを確認した．また，図 2.31 (b) から，入出力におけるクロックサイクル数には変化がないが，浮動小数点加算処理は 5,613 クロックサイクルまで削減できることを確認した．また，RaspberryPi4 に組込まれている ARM コアと同等の CPU クロック周波数（1.5 GHz）で実行すると考えた場合，273.7 MFLOPS (Mega Floating-point Operation Per Second) で実行可能となり，2.7.2 節における処理よりも性能が向上していることが確認できた．

(a) Simulation wave



(b) Clock cycle and instruction FLOPS

Instructions (1,024 entries and 128-bit data)	Clock cycles	MFLOPS (1.5 GHz operation frequency)
Input of the left CAM module	1,024	
Input of the right CAM module	1,024	
Output of the left CAM module	1,024	
Output of the right CAM module	1,024	
Other	13	
Floating-point Addition	5,613	273.7
Total	9,722	158.0

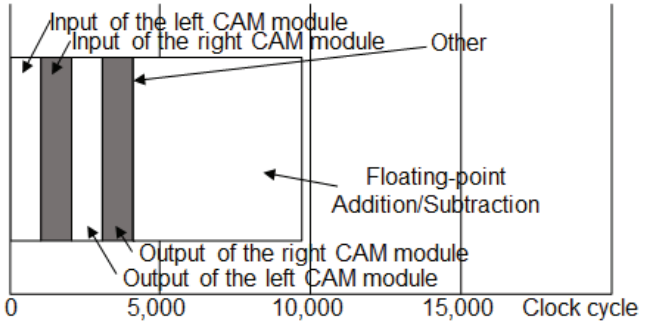


図 2.31: 2 の補数処理削減による単精度浮動小数点を利用した加算処理の結果.

2.7.5 ARM コアと CAMX による単精度浮動小数点を利用した加算処理の比較

本節では，CAMX と RaspberryPi 4 に組込まれている ARM コアに単精度浮動小数点を利用した加算処理を実装し，処理性能を比較する．

CAMX は連想メモリサイズを容易に変更可能であり，並列度に関係なくクロックサイクル数は一定となる．すなわち，CAMX が 0.1，0.5，1.0 及び 1.5 GHz のクロック周波数で動作すると考えた場合，CAMX は並列度に関係なく 5,613 クロックサイクルで実行可能である．

一方，RaspberryPi 4 に組込まれている ARM コアには，SIMD 処理を実行可能な NEON という機能と浮動小数点を処理する専用回路である VFP (Vector Floating Point) が搭載されている．そこで，NEON と VFP の両方を使用する場合，NEON を使用して VFP 使用しない場合，NEON と VFP の両方を使用しない場合において，単精度浮動小数点を利用した加算処理を実装し，処理性能を確認する．実装では，512，1024，2048，4,096 及び 8,192 個のデータに対して，それぞれ 10 億回実行し，処理時間の平均から性能を算出する．なお，RaspberryPi4 に組込まれている ARM コアは 1.5 GHz で動作する．

図 2.32 にそれぞれの性能として FLOPS 値を用いて比較を行う．縦軸に FLOPS 値を表し，横軸に処理データ数を表す．CAMX はクロック周波数を変化させた場合の結果を示した．CAMX は処理データ数の増加に関係なく一定のクロックサイクル数で実行されるため，データ数の増加と共に性能は向上することが確認できる．一方，ARM コアはデータ数が増加する性能は低下していくことが確認できた．1.5 GHz のクロック周波数で動作した場合の CAMX と NEON と VFP の両方使用した場合の ARM コアを比較した場合では，約 4,500 データ以上においては CAMX の方が性能向上する．1.0 GHz のクロック周波数で動作した場合の CAMX と NEON と VFP の両方使用した場合の ARM コアを比較した場合では，約 6,500 データ以上において，CAMX の方が性能向上する．また，0.5 GHz のクロック周波数で動作した場合の CAMX と NEON を使用して VFP を使用しない場合の ARM コアを比較した場合では，約 6,000 データ以上において，CAMX の方が性能向上する．

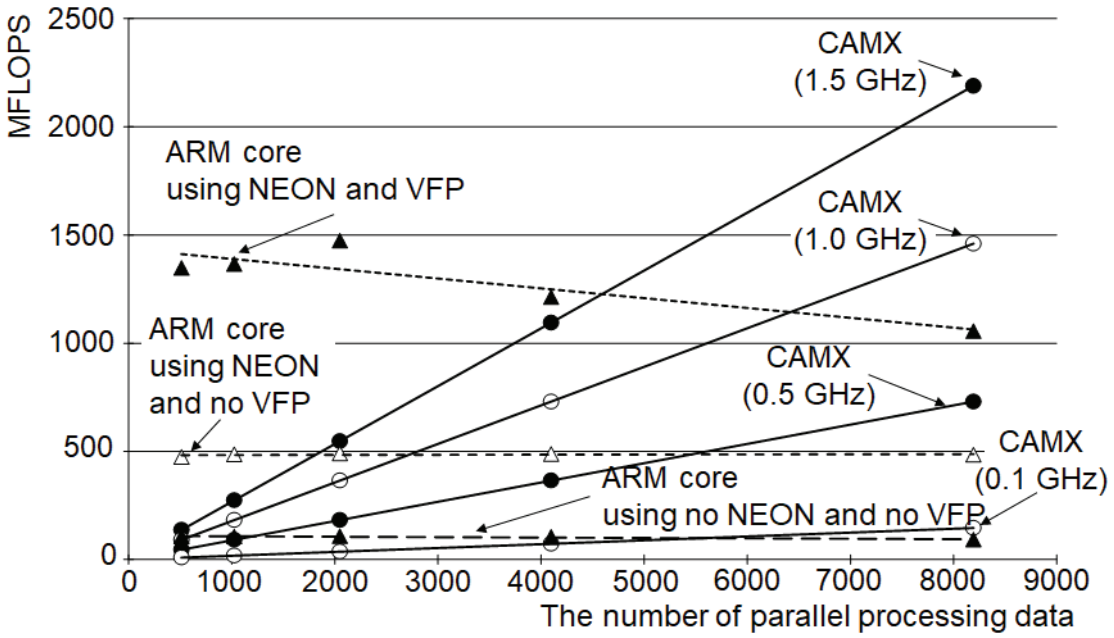


図 2.32: ARM コアと CAMX による単精度浮動小数点を利用した加算処理の比較結果.

2.8 まとめ

本章では，連想メモリをベースとしたモバイル機器向け機能メモリベース超並列 SIMD 型演算コア（CAMX）を提案し，構成について説明した．CAMX は CPU とシステムバスにより接続され，CPU のアクセラレータとして，CPU が並列処理を行う時に CAMX を利用する構成となっている．CAMX は CPU から命令を受けとると，コントローラからの命令を演算器において実行し，数千データを並列に処理することになる．これは，1 命令により数千データを処理可能とする SIMD 処理となる．また，演算器はデータの読み込み，実行，結果の書き込みをパイプラインに処理するることで，高速化を図っている．この様に，CAMX は数千データを並列かつ高速に実行可能となる．さらに，連想メモリをベースとすることで，マルチメディア処理において必要となる繰り返し演算処理及びテーブルルックアップ処理を並列に実行することが可能となる．特に，連想メモリは格納されたデータに対して一致検索処理に特化していることを生かし，CAMX においても一致検索処理の並列化が容易となった．さらに，一致検索処理により，一致したエントリを表すバリッドフラグを利用することで，動作させたいエントリを任意に実行可能であり，検索したデータが一致したエントリのみに対して演算や書き換えを行うことでテーブルルックアップ処理が容易なコアとなる．

この CAMX における基本演算命令及び検索命令の実行状況について確認した．基本演算命令は，AND，OR，XOR 及び ADD であり，CAMX の演算器に実装されているそれぞれの回路により処理することになる．検索命令は，左右どちらかの連想メモリに格納されているデータ対して，検索対象のデータとの一致を検索するものであり，ビット毎に検索を行うことが可能である．これらの命令は連想メモリに格納されている全てのデータに対して実行することが可能であることから並列に処理される．なお，本論文では，CAMX の連想メモリのサイズは 128 ビットで 1,024 エントリとしているため，基本演算命令及び検索命令は 1,024 並列に処理が実行される．基本演算命令及び検索命令を 128 ビットのデータに対して実行した結果，全ての命令が正常に実行されていることを確認でき，基本演算命令である AND，OR，XOR 及び ADD 命令は 130 クロックサイクル，検索命令は 1 クロックサイクルで実行されることを確認した．

次に，マルチメディア処理において必要となる符号付き乗算処理について，CAMX 上での実行手法について検討を行った．CAMX は小型かつ単純であることを研究のコンセプトとしているため，演算器には主に基本演算命令と検索命令を実装しており，乗算処理のための専用回路を実装していない．このため，乗算処理を実行する場合は，基本演算命令と検索命令を組合わせることによって処理する必要がある．本論文では，ビットシリアル乗算，検索･加算繰り返し乗算及び Baugh-Wooley

乗算をそれぞれ実装を行い，処理速度の比較を行った．ビットシリアル乗算とは，検索命令を用いず，乗数及び被乗数を 1 ビットずつ処理する一般的な計算手法である．検索・加算繰り返し乗算とは，乗数に対して検索命令を実行し，検索命令に一致した場合にのみ加算命令の実行を繰り返す手法である．Baugh-Wooley 乗算とは，Baugh-Wooley 演算を CAMX に実装した手法である．この結果，4 ビットデータ同士の乗算を実行した場合，ビットシリアル乗算は 1,462 クロックサイクル必要となるが，検索・加算繰り返し乗算は 237 クロックサイクルであり，乗算処理に必要なクロックサイクルを約 84 %削減できることを確認した．さらに，検索・加算繰り返し乗算と Baugh-Wooley 乗算について，データのビット幅の違いによるクロックサイクル数を比較した．データのビット幅は，4 ビットから 32 ビットまで変更して実行に要するクロックサイクル数を検証した．この結果，15 ビットを境に，検索・加算繰り返し乗算及び Baugh-Wooley 乗算のクロックサイクル数が反転することを確認し，15 ビットを超える乗算を実行する場合は Baugh-Wooley 乗算を実行した方がより高速に処理可能であることが確認できた．

さらに，CAMX の性能を確認するため，CAMX の開発の基になった MX-1 及び既存のモバイルデバイス向けプロセッサと比較を行った．比較を行うために，それぞれのプロセッサに暗号処理の一つである AES 処理を実装した．この結果，CAMX における AES 処理の総クロックサイクル数は 1,362,699 クロックサイクルであり，AES 処理の 1 ラウンドのみについて着目して MX-1 と比較したところ総クロックサイクル数では MX-1 の方が小さい値となった．しかし，処理におけるクロックサイクル数の詳細な内訳で確認すると，ShiftRows と MixColumns の合計クロックサイクル数については CAMX が MX-1 よりも約 65 %も高速に処理でき，AddRoundKey についてはクロックサイクル数は約 86 %も高速に処理できていることが確認できた．一方，SubBytes については CAMX が MX-1 よりも 9 %の処理低下が発生したため，総クロックサイクル数として CAMX の方が MX-1 よりもクロックサイクル数の増加となった．CAMX における SubBytes 処理については，8 ビット毎に逐次検索命令及びデータ書き換えを行っているが，今後，複数ビットを同時に検索及びデータ書き換えを行うことでさらに高速化が見込めると考えている．このため，CAMX は MX-1 と同等の性能を維持できていると確認できた．また，CAMX について，既存のモバイルデバイス向けプロセッサとの処理バイト当たりのクロックサイクル数を比較した．この結果，TI OMAP3530（ARM Cortex A8）との比較においては，CAMX の方が約 53 %も高性能であることが確認できた．その他のプロセッサについても，CAMX がより高性能のであることを確認した．以上の結果から，CAMX は MX-1 と同等の性能を維持しつつ，既存のモバイルデバイス向けプロセッサよりも高性能に処理可能であることが確認できた．

最後に，CAMX は MX-1 と同等の性能であり，既存のモバイルデバイス向けプ

ロセッサよりも高性能であることから，マルチメディア処理において必要となる浮動小数点を利用する加算処理を CAMX に実装して検証した．この結果，2 の補数削減処理による単精度浮動小数点を利用した加算処理を 1.5 GHz のクロック周波数で動作する CAMX に実装した場合，既存のモバイルデバイス向けプロセッサである ARM コアよりも処理データ数が 4,500 を超えると高性能になることが確認できた．以上の結果から，処理 CAMX はマルチメディア処理を有効に実行できると考えられる．

第3章　CAMXの実装を想定した社会ハザードに対する対策の提案

近年，高性能なモバイルデバイスを利用した新たな犯罪や社会課題等の社会ハザードが問題となっている．そこで，これらのモバイルデバイスを利用した犯罪や社会課題に対して，CAMX をモバイルデバイスに組込むことを想定した対策を提案する．CAMX は膨大なデータ量及び演算量が必要なマルチメディア処理に特化しているため，モバイルデバイスを用いた犯罪及び社会課題に対してリアルタイムに処理可能な対策として活用可能と考える．本章では，近年大きな問題となっているディープフェイクの様なデジタルデータの改ざん対策として画像改ざん検知手法を提案する．さらに，モバイルデバイスに搭載されているカメラ機能を利用した盗撮犯罪の対策として盗撮防止システムを提案する．どちらの手法及びシステムについても膨大なデータ量及び演算量に対してリアルタイムに画像処理が要求される．このため，将来的に，犯罪や社会課題に利用されるモバイルデバイスに対して，より高速かつ高性能な対策を行えるよう CAMX を組込むことを想定する．

3.1　はじめに

高性能なモバイルデバイスを用いた犯罪及び社会課題として，画像改ざん及び盗撮に着目する．画像改ざん問題として，近年ディープフェイクを用いた改ざんされた動画像をインターネット上に投稿し，世の中の人々を騙す等，社会的に大きな問題となっている．これは，AI 処理を用いて現実には存在しない動画像を一般のユーザが簡単に作成でき，インターネット上に気軽に投稿できるようになったことから発生した新たな社会課題である．さらに，近年はこの動画像をモバイルデバイスでも簡単に作成できるようになっている．また，盗撮については，被写体に許可なく撮影する行為のことであり，この行為は犯罪となる．これは，モバイルデバイスに搭載されたイメージセンサの高性能化及び高画質化により急増

してきた犯罪であり，撮影した動画像は一般のユーザでも容易にモバイルデバイス上で編集・加工できるようになっていることも影響している．これらの社会課題及び犯罪の対策を行えるよう，マルチメディア処理に特化した CAMX をモバイルデバイスに組込むことで様々な対策が可能になることを想定する．

3.2 CAMX をモバイルデバイスに組込むことを想定した事前研究

近年，モバイルデバイスを用いた新たな社会課題及び犯罪が発生している．この理由として，半導体技術の急速な発展が挙げられ，これに伴ってモバイルデバイスの高性能になっている．このため，一般のユーザでも気軽にデジタルデータを扱えるようになり，特に，モバイルデバイスに搭載されている高性能なイメージセンサから取得した動画像を編集・加工し，インターネットに投稿できるようになっており，ハードウェアによる対策が求められている．そこで，本論文では，今後，モバイルデバイスに CAMX を組込むことを想定し，昨今大きな問題となっている画像改ざん及び盗撮について対策手法及びシステムを提案する．

マルチメディア処理を得意とする CAMX では動画像処理を容易に実行でき，画像改ざん検知手法及び盗撮防止システムについて有効に処理可能であると考える．画像改ざん検知手法はモルフォロジカルパターンスペクトラム処理により，画像の特徴から画像の改ざんを検知する．モルフォロジカルパターンスペクトラム処理は演算量が多く，実行するにはマルチメディア処理を得意とするプロセッサが不可欠となる．このため，CAMX を利用して処理することで，モバイルデバイス上においても容易に処理が可能となる．さらに，盗撮防止システムは LED 照明とスマートフォンを可視光ビーコンにより連携させ，スマートフォンのカメラ機能等を強制的に操作する手法である．このシステムは，スマートフォンに搭載されているイメージセンサから取得した動画に対してリアルタイムに高速フーリエ変換を実行する必要があり，CAMX を利用して処理することで，より確実に可視光ビーコンを検知可能となり，スマートフォンのカメラ機能を強制的に操作可能となる．

3.2.1 モルフォロジカルパターンスペクトラム処理を用いた画像改ざん検知手法の提案

背景

近年，モバイルデバイスの発展に伴い，一般のユーザでも動画像を気軽に編集・加工できるようになっている．さらに，動画像は高画質になっており，データ量は膨大になっているため，取得した動画像のデジタルデータから情報を抽出することが可能となっている．動画像から情報を抽出する手法として，テクスチャ解析やオブジェクト判別が知られている．テクスチャ解析は特徴抽出や画像解析において知られており，画像内のテクスチャの特徴を抽出する処理である [89]．オブジェクト判別はスマートフォンや監視カメラ等で幅広く利用されており，画像中の物体を特定する処理である [90]．これらの技術を利用したシステムは多数知られており，その技術の一つとしてモルフォロジカルパターンスペクトラム処理が知られている [91], [92]．モルフォロジカルパターンスペクトラムは，注目画素とその近傍画素とで非線形な演算処理を行うことで，画像の特徴をスペクトラムとして表現する手法である．本節では，モルフォロジカルパターンスペクトラムに対する並列処理研究 [93] を基に，モバイルデバイス向けコアであり，CAMX の基になった MX-1 に実装し，マルチメディア処理の有効性を検証する．なお，これまでの研究により MX-1 は評価ボードに組込まれ，モバイルデバイスの環境を再現できることから，本論文では MX-1 に対して画像改ざん検知手法を実装した．CAMX は 2.6 節において，MX-1 よりも高速に演算ができることを確認しているため，MX-1 を用いて検証することで，CAMX を実装した場合にはより高性能に実行可能であることが想定できる．

ここで，モルフォロジカルパターンスペクトラムに関する他研究について確認する．モルフォロジカルパターンスペクトラムを利用した研究として，画像から特徴を抽出し，パターンスペクトラム (Pattern Spectrum) により任意のテクスチャの特徴を表すテクスチャ解析がある [94]．さらに，画像中の人物にパターンスペクトラムを用いて，ズボン，スカート，胸部の大きさ，体型等の情報を抽出し，男女識別を行うことも可能である [95]．このように，モルフォロジカルパターンスペクトラムは画像解析技術において様々な応用が期待されている手法である．しかし，モルフォロジカルパターンスペクトラム処理において必要となるオープニング処理，クロージング処理及び画素値計算は繰り返し実行が必要となる．加えて，構造要素を利用してスペクトラムを作成するためには，全画素に対して差分処理を繰り返し実行する必要があり，メモリアクセスが頻繁に発生し，処理時間が増大する原因になっている．このため，モルフォロジカルパターンスペクトラムを

一般的なプロセッサの逐次処理によりリアルタイムに実行することは困難であることが知られている [96]．一方，モバイルデバイスに組込まれているプロセッサは，消費電力，実装面積，コスト等の制限により，演算機能やハードウェア量に限界がある．このため，リアルタイムに実行する手段として，並列処理等が必要となる [97]．すなわち，モルフォロジカルパターンスペクトラムを処理するには，並列処理を実現するためにの回路が必要となり，ハードウェア量，コスト，消費電力の増加するトレードオフの関係が発生する．本節では，C 言語によるソフトウェアベースプロセッサでありながら，効果的な並列処理が可能なプロセッサコア MX-1 においてモルフォロジカルパターンスペクトラム処理を実装する．MX-1 は高並列に処理可能であるため，モルフォロジカルパターンスペクトラムの高い解析能力をモバイルデバイスにおいて実現可能となる．

モルフォロジカルパターンスペクトラム処理

本節では，モルフォロジカルパターンスペクトラム処理について説明する．画像処理の一つであるモルフォロジ演算 [98] は，画像を特徴的な構造の集まりであると考え，対象画像から構造の大きさや形状を抽出する．この構造を “構造要素” と呼び，小図形間の集合演算を行うことになり，画素値が 2 値及び多値により 3 種類（SP : Set Processing，FSP : Function and Set Processing，FP : Function Processing）に分類できる（表 3.1）．それぞれの種類により演算手法は異なるが，演算高速化の観点から SP による処理が多用されている [96], [99]．一方，SP は画像を 2 値化しなければならず，対象画像の情報を劣化することになり，十分な解析を行えない可能性がある．そこで，本節では FSP を高速に実現することを目的に，CAMX と同等の性能がある MX-1 を用いて高速化の検証を行う．

表 3.1: モルフォロジカルパターンスペクトラムの種類.

	Target image	Structuring element	Operation
SP	Binary image	Binary	AND, OR
FSP	Grayscale image	Binary	Maximum value calculated
FP	Grayscale image	Grayscale	ADD，SUB， Maximum value calculated

FSP の基本的な演算は下述の式で定義される [99]．ここで，対象画像を X，その中に含まれる画素を x，オープニング処理画像中の画素を y として，構造要素を B，B の反転を B^S=$(-b : b \in B)$ と定義する．なお，ミンコフスキー和を $\oplus$，ミ

ンコフスキー差を $\ominus$ で表す.

$$\text{ダイレーション} : (X \oplus B^S)(x) = max\{X(y) : y \in (B^S)_x\} \tag{3.1}$$
$$\text{エロージョン} : (X \ominus B^S)(x) = min\{X(y) : y \in (B^S)_x\} \tag{3.2}$$
$$\text{オープニング} : (X \ominus B^S) \oplus B^S(x) \tag{3.3}$$
$$\text{クロージング} : (X \oplus B^S) \ominus B^S(x) \tag{3.4}$$
$$\tag{3.5}$$

ダイレーションは構造要素により画素を膨張する処理であり，エロージョンは画素を縮小する処理である．さらに，オープニングはエロージョン，ダイレーションの順に処理することで，輪郭を内側から滑らかにする平滑処理となり，クロージングはダイレーション，エロージョンの順に処理することで，輪郭を外側から滑らかにする平滑処理である．それぞれのイメージを図 3.1 に表す.

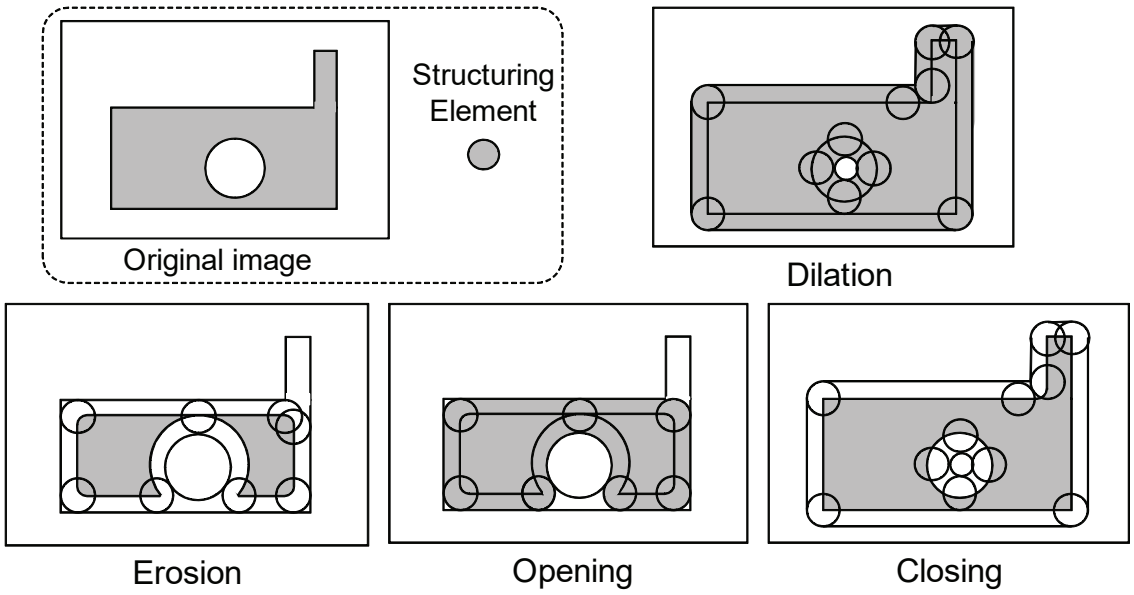


図 3.1: モルフォロジ演算の概念.

次に，パターンスペクトラムアルゴリズムについて説明する．パターンスペクトラムは，構造要素が対象画像にどの程度含まれているかを，形状やサイズの違いで表したものである．図 3.2 にモルフォロジカルパターンスペクトラム処理の手

順を示す．まず始めに，対象画像に対して構造要素を用いてオープニングを実行し，この処理により得られた画像に対して拡大した構造要素によりオープニングを行う処理を繰り返す．ここでは，4 回繰り返した結果を示し，構造要素も 4 回拡大している．これらの結果から得られた画像同士に対して差分をとることで，構造要素と同じ形状及びサイズを抽出することが可能となる．この形状及びサイズをグラフ化したものが，パターンスペクトラムとなる．

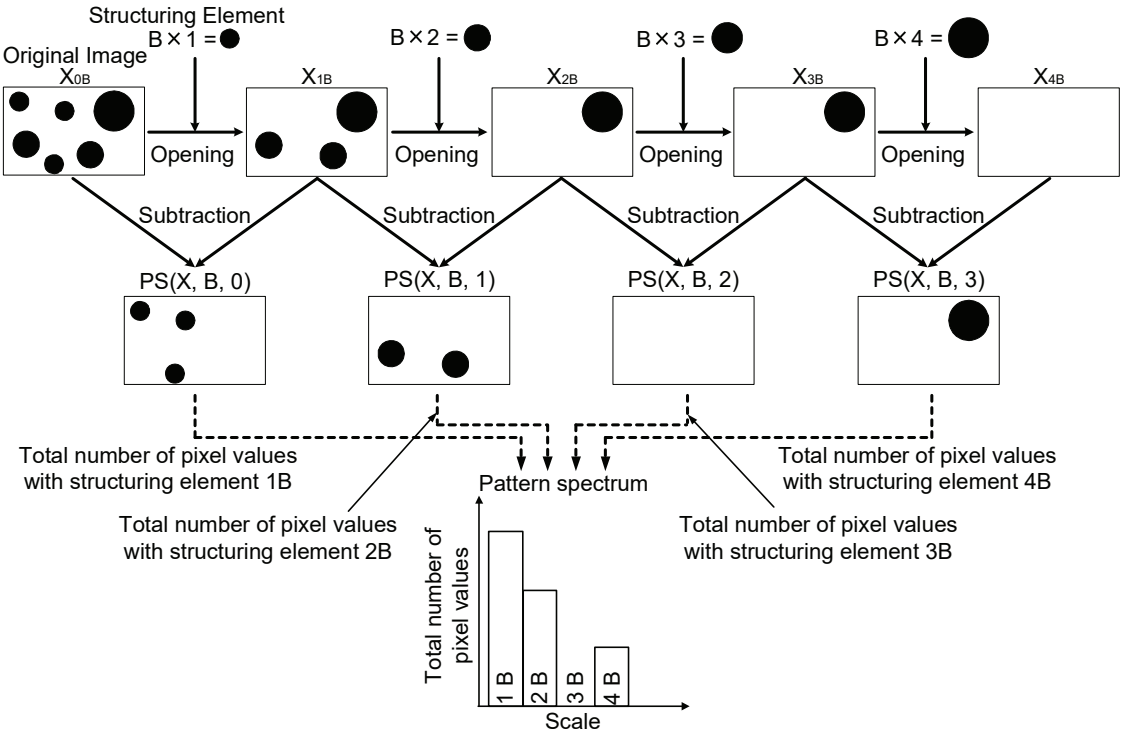


図 3.2: モルフォロジカルパターンスペクトラムの処理手順 $(n = 4)$.

MX-1 におけるモルフォロジカルパターンスペクトラム処理の実装

本節では，CAMX の基となった MX-1 に対してモルフォロジカルパターンスペクトラム処理を実装し，CAMX に実装する前に有効性を検証する．まずは，3 × 3 ピクセルの正方形とした構造要素及び 32 × 32 の対象画像を例として説明し，一般化について後述で議論する．図 3.3 に MX-1 におけるモルフォロジ演算の流れを示す．MX-1 では，対象画像の画素値を SRAM に格納し（ステップ 1），構造要素の形状に合わせて SRAM 内のデータを並列に実行しやすくなるように並び替える（ステップ 2）．次に，モルフォロジ演算に従って，最小値・最大値検索を行い（ステップ 3），モルフォロジ演算のオープニング処理が終了するまで繰り返し実行する．32 × 32 の対象画像（図 3.4 (a)）に対して，MX-1 における画素値の格納手順について説明する．なお，図 3.4 (a) の対象画像中の画素配列は図 3.4 (b) とする．

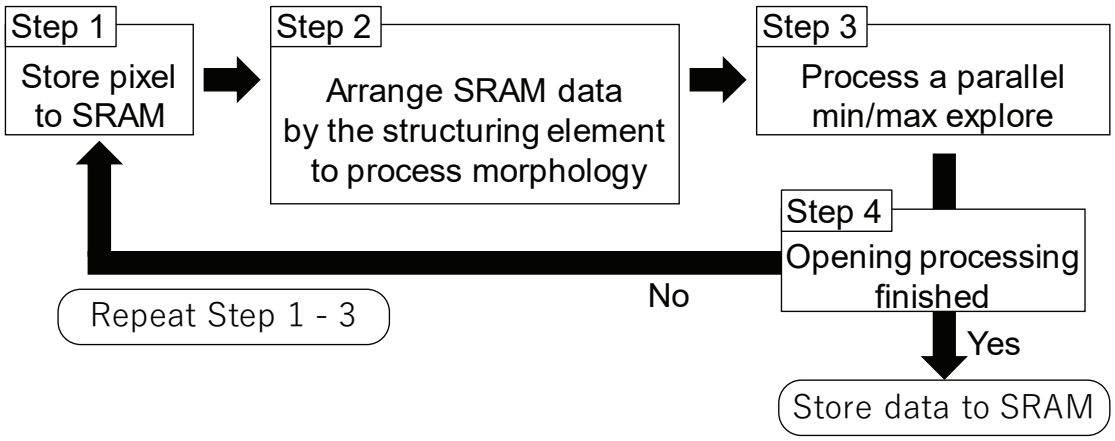


図 3.3: MX-1 におけるモルフォロジカルパターンスペクトラムの処理手順.

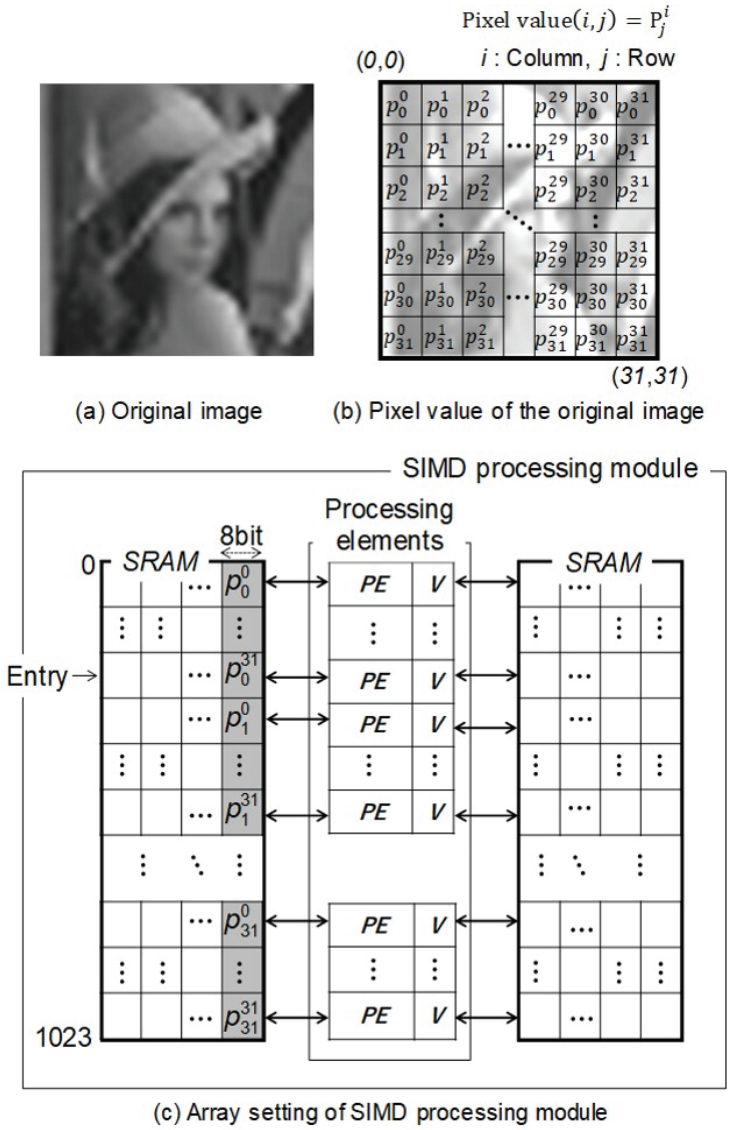


図 3.4: MX-1 内の SRAM における対象画像の画素値格納方法.

ステップ 1 MX コアに組込まれた SRAM に図 3.4 (a) の対象画像中の画素値を格納する．SRAM に図 3.4 (b) に示す画素値 (p_j^i(水平方向 : $i = 0, 1, 2, \cdots, 31$, 垂直方法 : $j = 0, 1, 2, \cdots, 31$)) を 1 行目から順に垂直方向に格納する (図 3.4 (c))．対象画像の各画素は SRAM の各エントリに対応するよう，インターフェースモジュールを介して並び替えられる [100].

ステップ 2 MX-1 で効率的な並列処理を実現するために画素値の配置が重要となるため，構造要素の形状に合わせて，SRAM に格納された画素値を移動及び配置を行う．モルフォロジ演算を並列に処理するため，ステップ 1 で格納した対象画像の画素値を構造要素の形状に合わせて演算を行う．構造要素を 3 × 3 ピクセルの正方形状とした場合，水平・垂直チャネルを利用して図 3.5 (a) の (i) に示すよう，画素値を SRAM 内で横一列に配置する．なお，図中の番号付けは構造要素の対応位置を分りやすくするためのものである．モルフォロジ演算の注目画素は “5” の部分に対応しており，その周りの近傍画素を同じエントリに格納し，この注目画素と近傍画素の間で FSP に基づく最小値・最大値探索を実行する．画素値は SRAM 内で横一列に配置されるため，1 画素に対する近傍画素との探索演算が一つのエントリ内で行うことができる．この処理は，構造要素 1,024 個分の演算を一度に行うことができるため，並列処理となる．32 × 32 の画像に対して 3 × 3 ピクセルの構造要素を適用し，左上から右下まで 1 ピクセルずつずらして処理を行った場合，対象画像に対する構造要素の適用方法は 3 パターンが考えられる．図 3.5 (b) に対象画像中での各パターンの画素位置を示す．パターン A は構造要素の注目画素とその隣接画素の全てが対象画像内に収まる場合であり，3 × 3 ピクセルの構造要素であるので 9 ピクセルが演算対象となる．パターン B は構造要素の上下左右のいずれか一列が画像中からはみ出て演算される場合であり，3 × 3 ピクセルの構造要素であるので 6 ピクセルが演算対象となる．パターン C は構造要素の上下の一列と左右の一列が両方はみ出ている場合であり，3 × 3 ピクセルの構造要素であるので 4 ピクセルが演算対象となる．図 3.5 (b) において，パターン A(処理対象画素 p_1^1)，パターン B(処理対象画素 p_1^{31})，パターン C(処理対象 p_{31}^{31}) の部分を注目画素とすると，SRAM 内では図 3.5 (a) の (ii)∼(iv) のような配置となる．

以上の手順で全画素に対して構造要素を適用すると，注目画素と隣接画素は図 3.6 のように SRAM 内に配置される．なお，MX コアは垂直チャネルを用いて縦方向のデータ転送を行うことができるため，ステップ 1 で入力された画素値を縦方向に移動させ，他のエントリに転送することで配置される．

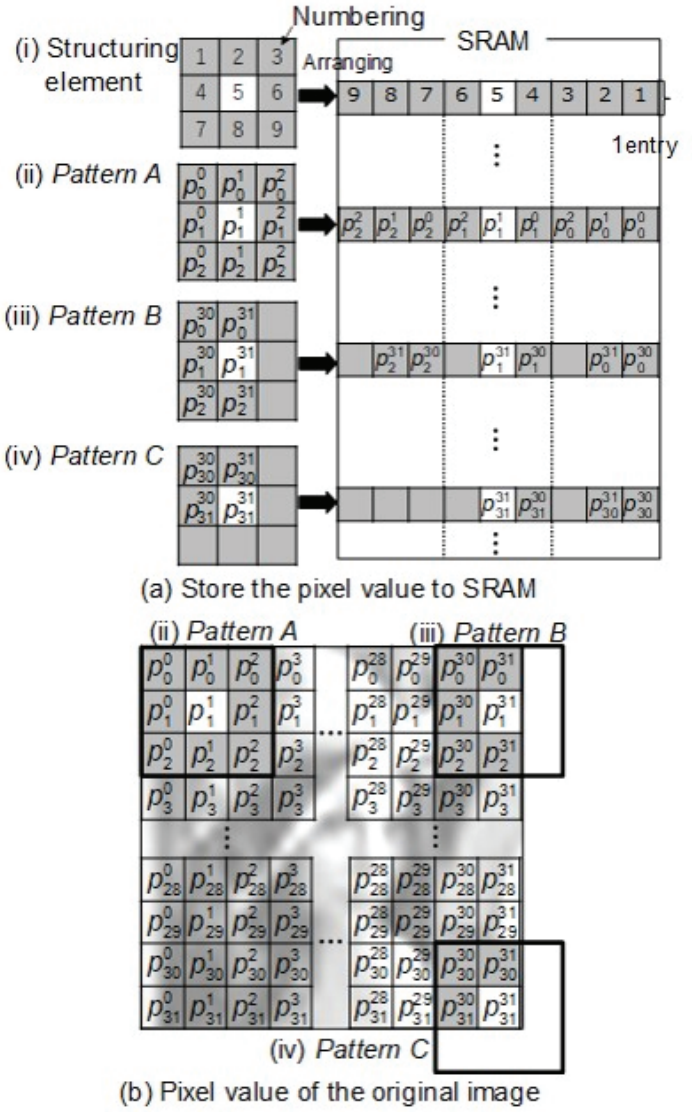


図 3.5: MX-1 内の SRAM における構造要素により抽出した画素値の格納方法.

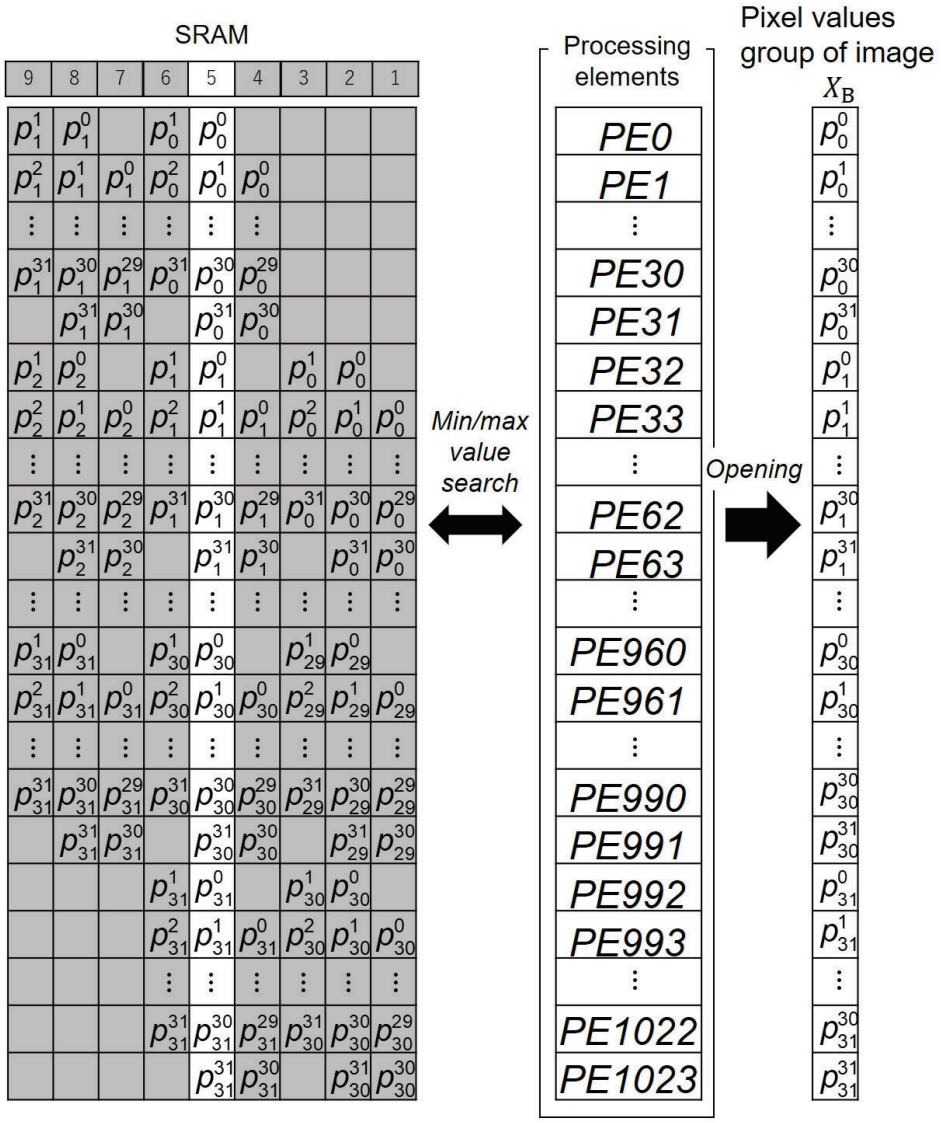


図 3.6: MX-1 内の SRAM における注目画素と画素値の整理及び演算後に出力される画素値 (構造要素:3×3).

ステップ 3 3×3 の構造要素を用いた場合，MX-1 における画素値のメモリ配置は図 3.6 の様になっているため，1 エントリ毎にビットシリアルで水平方向に読み出しを行い，2 ビット PE で演算を実行する．なお，この演算は全てのエントリに対して同時に実行されるので，並列処理となる．演算では，最小値・最

大値探索を行い，モルフォロジ演算である各画素のデータを 1,024 並列で実現できる．

ステップ 4 ステップ 1~3 の手順でエロージョン (最小値探索) を行った後にステップ 1 に戻り，ステップ 3 でダイレーション (最大値探索) を行うことで，オープニング処理が実現可能となる．図 3.6 に示すように，オープニング処理が行われた画素値群 (X_B) は SRAM 内の各エントリに出力される．

ここで，MX-1 を用いたモルフォロジ演算の一般化を行う．上述した手法は，32 × 32 の対象画像に対して，3 × 3 ピクセルの構造要素を用いたモルフォロジ演算であった．一般に，モルフォロジ演算は対象画像に対して正方形とした構造要素の注目画素を中心に 3 × 3，5 × 5，7 × 7，· · · とスケールアップして処理する必要がある．すなわち，スケールアップの度にオープニング処理を実行することになるため，プログラム開発にあたっては，任意の構造要素の大きさに対応する SRAM 内の配置手法を検討する必要がある．ここで，MX-1 における画素値の格納方法を図 3.7 に示すよう一般化する．図 3.7 において，p_j^i (水平方向 : $i = 0, 1, 2, \cdots, S-1$, 垂直方向 : $j = 0, 1, 2, \cdots, T-1$) は画像サイズが $S \times T$ の画像における画素値であり，X_{nB}(n : オープニング処理の回数) は画像の画素値群を表す．E_k^k ($k = 0, 1, 2, \cdots, N-1$) は正方形状の構造要素を表し，図 3.7 の $E_0^0 \sim E_{N-1}^{N-1}$ 様に画素値を横一列として扱う．また，E_r^r は前述と同様に，正方形状の構造要素における注目画素を表す．以上より，$N \times N$ である正方形状の構造要素により，$S \times T$ である対象画像に対し，処理を行うと，MX-1 の SRAM 内では注目画素と隣接画素が図 3.7 の様に横一列に格納される．横一列に格納された画素群は，エントリ毎にモルフォロジ演算である最小値・最大値探索が実行され，オープニング処理結果である画素群 Y_{nB} (q_j^i (水平方向 : $i = 0, 1, 2, \cdots, S-1$, 垂直方向 : $j = 0, 1, 2, \cdots, T-1$)) として出力される．これらの処理は全エントリに対して実行され，並列に処理される．なお，SRAM サイズの影響により，$S \times T$ である対象画像の画素群を SRAM に格納できず，一括で処理できない場合，$S \times T$ の対象画像を任意に分割して SRAM に格納し，分割した領域毎に処理を行うことになる．一方，画像を単純に分割した場合，境界部分にエイリアシングなどのノイズが発生して画質が劣化する可能性があるため，画像の分割時に構造要素の半径分を重複して分割する等の工夫が必要となる．

次に，パターンスペクトラム処理について説明する．モルフォロジ演算においてオープニング処理を実行した後，対象画像の画素群 X_{nB} とオープニング処理で生成した画像の画素群 $X_{(n+1)B}$ で減算処理（$X_{nB} - X_{(n+1)B}$）を実行する（図 3.8）．$S \times T$ である対象画像の画素群 X_{nB} の各画素値を A_j^i (水平方向 : $i = 0, 1, 2, \cdots, S-1$, 垂直方向 : $j = 0, 1, 2, \cdots, T-1$)，$S \times T$ のオープニング処理で生成した画像の画素群 $X_{(n+1)B}$ の各画素値を B_j^i (水平方向 : $i = 0, 1, 2, \cdots, S-1$, 垂直方向 : $j = 0, 1, 2, \cdots, T-1$) とすると，減算処理は各エントリ毎に行われ，画像の画素群 X_{nB}，$X_{(n+1)B}$ に対する各画素毎の演算となる．

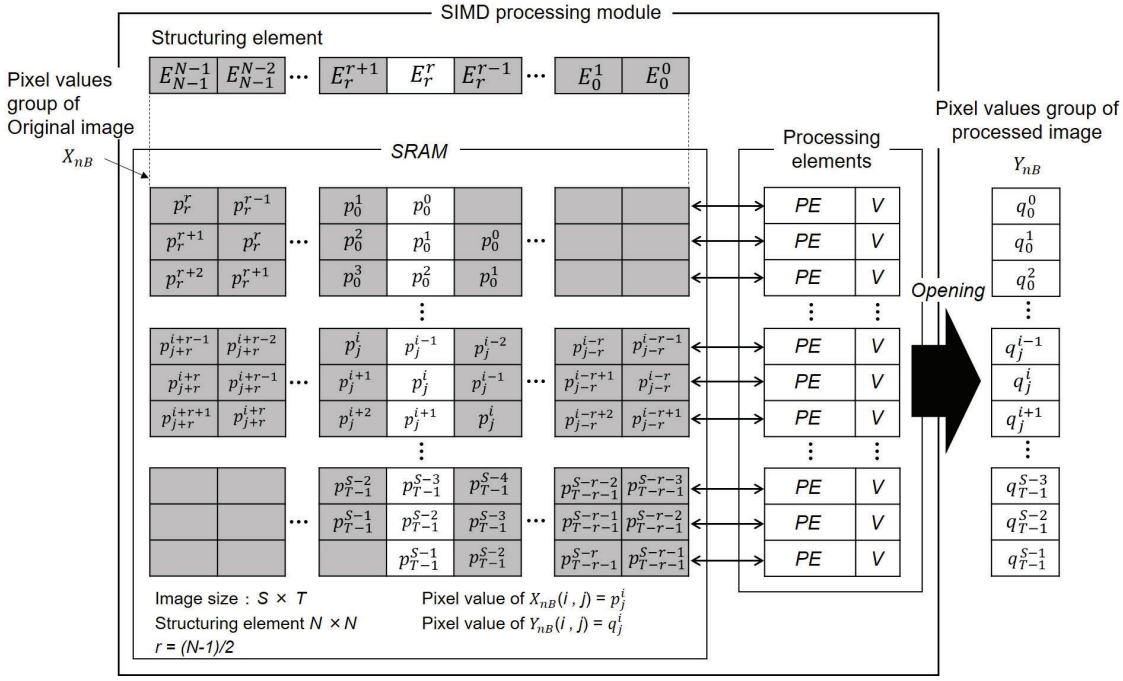


図 3.7: 構造要素による全画素値の配置:$N \times N$.

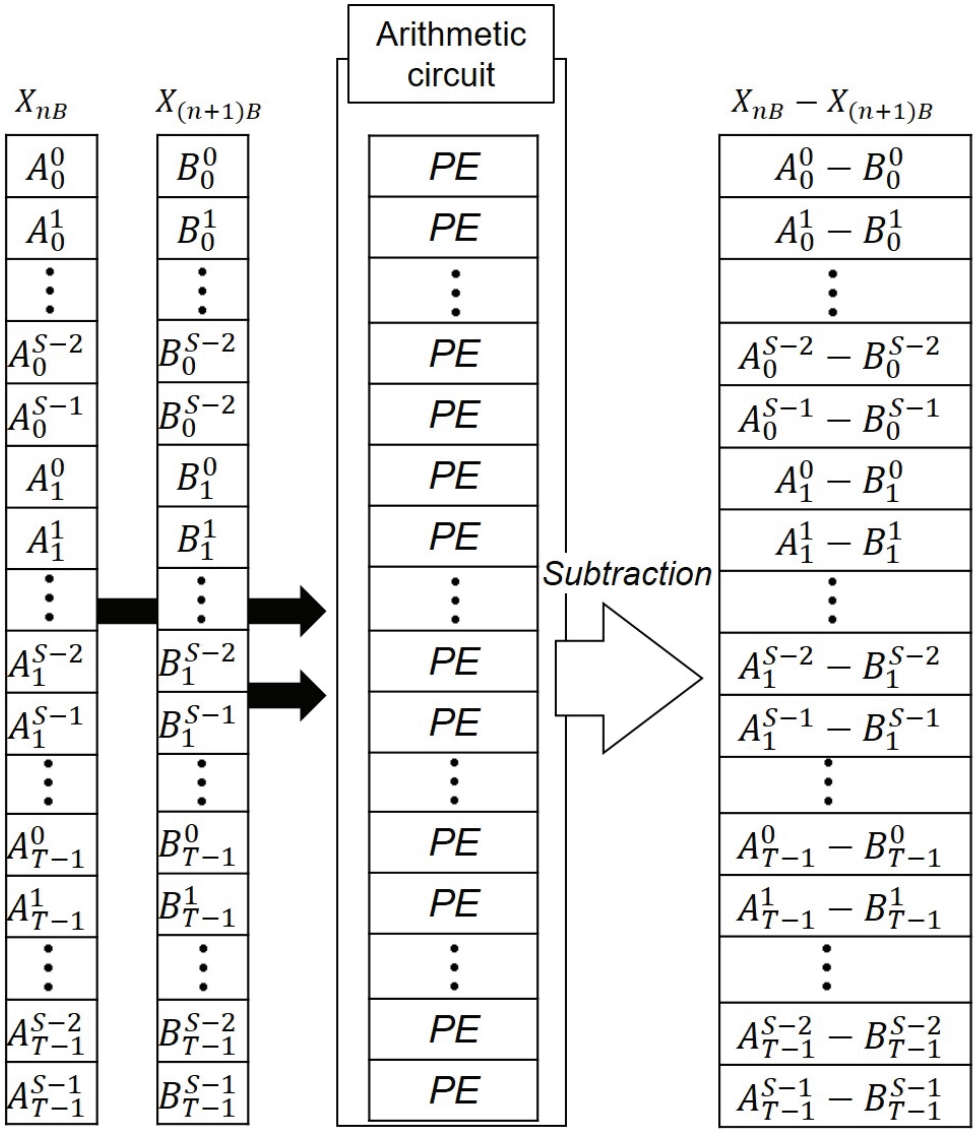

Arithmetic circuit
X_{nB}
$X_{(n+1)B}$
$X_{nB} - X_{(n+1)B}$
PE
Subtraction
A_0^0 A_0^1 A_0^{S-2} A_0^{S-1} A_1^0 A_1^1 A_1^{S-2} A_1^{S-1} A_{T-1}^0 A_{T-1}^1 A_{T-1}^{S-2} A_{T-1}^{S-1}
B_0^0 B_0^1 B_0^{S-2} B_0^{S-2} B_1^0 B_1^1 B_1^{S-2} B_1^{S-1} B_{T-1}^0 B_{T-1}^1 B_{T-1}^{S-2} B_{T-1}^{S-1}
$A_0^0 - B_0^0$ $A_0^1 - B_0^1$ $A_0^{S-2} - B_0^{S-2}$ $A_0^{S-1} - B_0^{S-1}$ $A_1^0 - B_1^0$ $A_1^1 - B_1^1$ $A_1^{S-2} - B_1^{S-2}$ $A_1^{S-1} - B_1^{S-1}$ $A_{T-1}^0 - B_{T-1}^0$ $A_{T-1}^1 - B_{T-1}^1$ $A_{T-1}^{S-2} - B_{T-1}^{S-2}$ $A_{T-1}^{S-1} - B_{T-1}^{S-1}$

図 3.8: 減算処理手法.

MX-1 によるモルフォロジカルパターンスペクトラム処理の時間

MX-1 を用いた場合のモルフォロジカルパターンスペクトラム処理の時間は，“SRAM への画素値格納”，“演算器における計算” 及び “演算結果の出力” の合計時間となる．ここで，“SRAM への画素値格納” の処理時間を T_a，“演算結果の出力” の処理時間を T_b とする．T_a，T_b はモルフォロジ演算における構造要素のスケールアップ毎に，SRAM への画素値の格納及び出力が行われるため，その度に処理時間が発生する．また，“演算器における計算” の処理時間を $T_{st(2k+1)}(k$：スケールアップの回数) とすると，$T_{st(2k+1)}$ は構造要素の大きさにより処理時間が変化する．なお，本節では，構造要素は正方形状とし，3×3，5×5，7×7，$\cdots$ と拡大する．以上から，MX-1 を用いた場合のモルフォロジカルパターンスペクトラム処理の処理時間は T で表すと式 (3.6) となる．

$$T = \sum_{k=0}^{N-1} \Big((T_a + T_b) + T_{st(2k+1)} \Big) \tag{3.6}$$

加えて，モルフォロジカルパターンスペクトラム処理の計算時間は，対象画像のサイズにより，画像を分割して処理する場合と MX-1 に組込まれた SRAM を拡大する場合でも変化する．

画像を分割して処理する場合，対象画像を任意に分割し，分割した画素群毎に SRAM へ格納及び出力することになる．このため，分割回数が増えて SRAM の格納回数が増加すると，“SRAM への画素値格納” の処理時間を T_a も増加する．一方，演算結果を出力しなければいけないため，“演算結果の出力” の処理時間を T_b も増加する．さらに，分割されたそれぞれの画素群に計算が必要となるため，“演算器における計算” の処理時間を $T_{st(2k+1)}$ も増加する．すなわち，本節で扱った 32×32 の画像を考慮し，$S \times T$ の対象画像を処理した場合，32×32 の画像と $S \times T$ の対象画像の比率分処理量が増加する．よって，式 (3.6) は以下のように変化する (式 (3.7))．なお，分割時には構造要素の半径分を重ねて分割することになるが，処理時間においては大きな差が生じないと考えられるため式 (3.7) では考慮しない．

$$\begin{aligned} T &= \sum_{k=0}^{N-1} \Big((T_a + T_b) + T_{st(2k+1)} \Big) \times \frac{S}{32} \times \frac{T}{32} \\ &= \sum_{k=0}^{N-1} \Big((T_a + T_b) + T_{st(2k+1)} \Big) \times \frac{ST}{1024} \end{aligned} \tag{3.7}$$

MX-1 に組込まれた SRAM を拡大する場合，SRAM へ格納できる画素群の容量は大きくなるが，SRAM へ画素群を格納及び出力する時間が増加する．このため，

“SRAM への画素値格納” の処理時間 T_a 及び “演算結果の出力” の処理時間 T_b が増加する．一方，SRAM へ格納できる画素群が増加することでモルフォロジカルパターンスペクトラム処理における計算対象の画素値数も増加するが，SRAM の拡大と同様に演算器も拡大され，モルフォロジカルパターンスペクトラム処理は並列に実行可能であることから “演算器における計算” の処理時間 $T_{st(2k+1)}$ は変化せず一定となる．すなわち，MX-1 に組込まれた SRAM の最大サイズを U とすると，32×32 の画像を格納するための SRAM のサイズ 1,024 を基に拡大率を考えると，T_a 及び T_b の格納時間が U / 1024 分増加する．式 (3.6) は以下のように変化する (式 (3.8))．

$$
\begin{aligned}
T &= \sum_{k=0}^{N-1} \left(\left((T_a + T_b) \times \frac{U}{32 \times 32} \right) + T_{st(2k+1)} \right) \\
&= \sum_{k=0}^{N-1} \left(\left((T_a + T_b) \times \frac{U}{1024} \right) + T_{st(2k+1)} \right) \qquad (3.8)
\end{aligned}
$$

モルフォロジカルパターンスペクトラム処理の結果及び評価

MX-1 による並列処理の効果を検証するため，本節では総合開発環境である High-performance Embedded Workshop (HEW) [101] と MX-1 評価ボードを用いて実装を行った．なお，MX コアの動作を HEW 上にてサポートするため，HEW には専用のデバッカ，コードジェネレータを組込んでいる．

32×32 ピクセルの対象画像に対してモルフォロジカルパターンスペクトラム処理を行った場合の処理画像結果とパターンスペクトラム結果を図 3.9 に示す．本実験では FSP 処理を実装し，画素は 8 ビットのグレースケール値となっている．図 3.9 (a) の様に，オープニング処理によって，各画素の状態遷移は構造要素のサイズ拡大に合わせて変化するオープニング画像が得られる．その変化の差分を視覚的に表したものが，スペクトラム画像になる．図 3.9 (b) では，スペクトラム画像内の全画素値を合計したものをグラフで表している．横軸は構造要素サイズであり，縦軸が画素値の総和である輝度値であり，スペクトラム画像の変化を示している．なお，構造要素は 14 回スケールアップ ($n = 15$) としている．

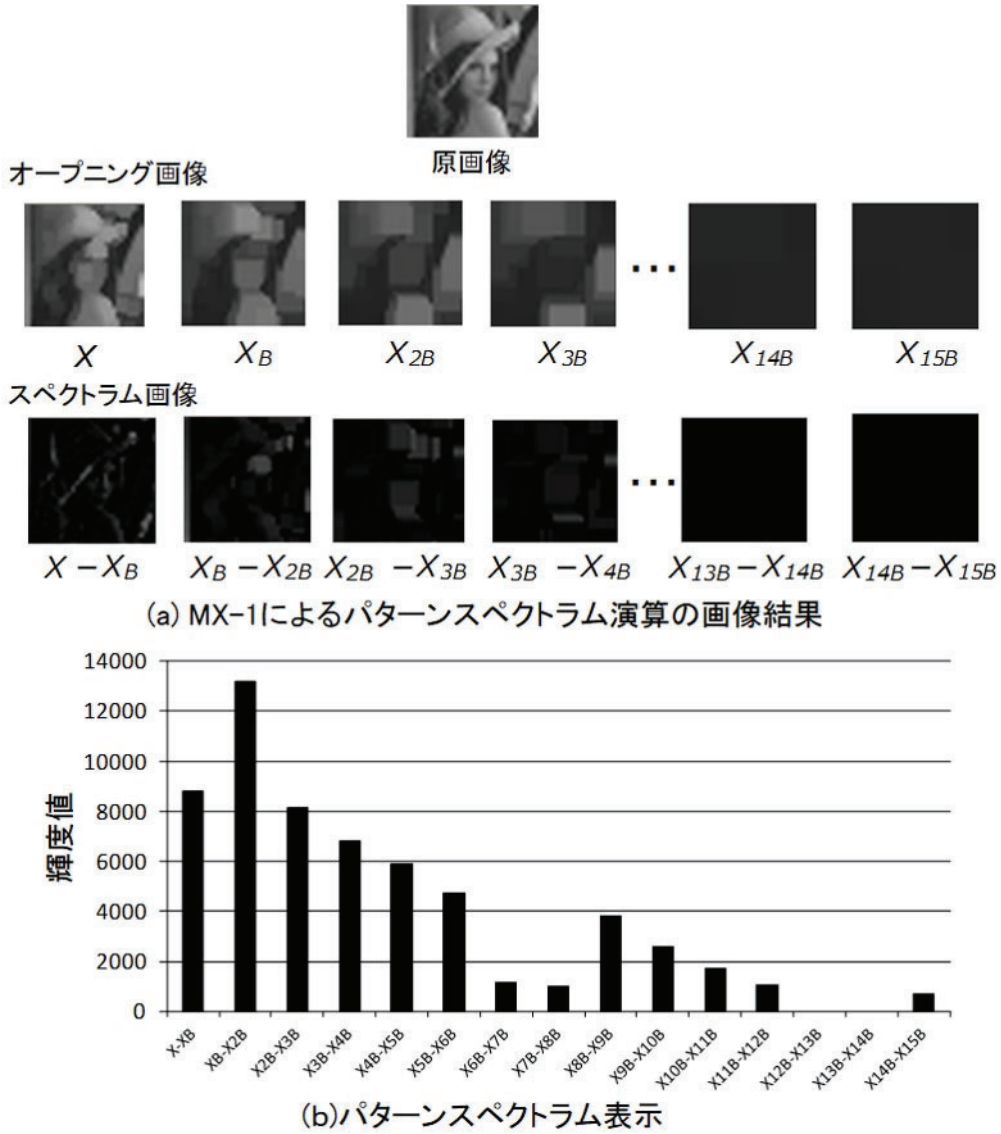


図 3.9: MX-1 における処理結果.

ここで，MX-1 によるモルフォロジカルパターンスペクトラム処理の処理時間について考える．MX-1 内部の制御用 CPU は通常の逐次処理を実行しつつ，MX ライブラリの命令呼び出し動作も逐一実行している．そのため，単純なプログラムだけで制御用 CPU の負担が大きくなり，処理速度に大きく影響を及ぼす．CPU は MX-1 を利用して，処理速度向上に対して並列に処理を行う必要があるため，DMA を利用しデータ転送時の CPU の負担を軽減する．さらに，CPU に対しては，MX-1

命令の呼び出し動作を最小限に留め，プログラム内の変数計算等，並列処理と関係のない処理に集中させることで負荷を軽減させる．これらを実現させるため，画素値配置と比較演算を行うモルフォロジ演算，減算処理と画素値集計を行うパターンスペクトラム演算の MX-1 動作命令をユーザマイクロコードとしてまとめ，制御用 CPU から MX-1 に対する呼び出し命令数の削減を実現させる．ユーザマイクロコードを生成するためには，制御用 CPU，DMA，そして MX-1 の各動作命令をプログラム内にて混在させずに記述する必要があり，各処理のまとまりをそれぞれに担当させ，並列動作を容易に行えるようにする．ここで，CPU と MX-1 による並列動作の概念を図 3.10 に示す．並列動作を考慮しない通常の実装 (Straight forward implementation) では MX 動作と並行して，CPU の制御動作に加え，呼び出し (Polling) 動作を行い，MX-1 動作時に CPU では動作待ちの状態になっているが，MX 動作命令をユーザマイクロコードとしてまとめた最適化実装 (Optimized implementation) では，CPU のポーリング動作が削減されている．その結果，CPU と MX-1 の効率的な並列動作が可能となり，実行時間の短縮が実現できる．

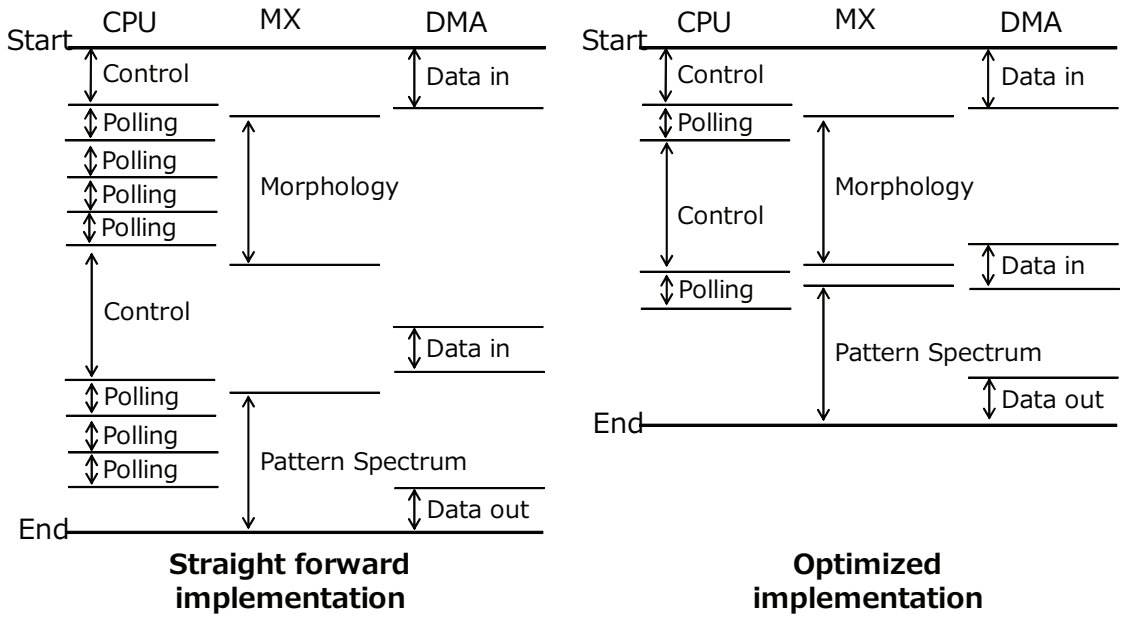
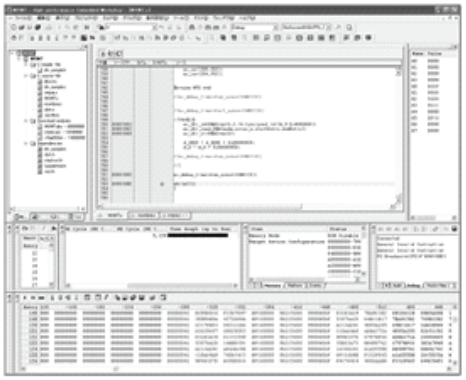


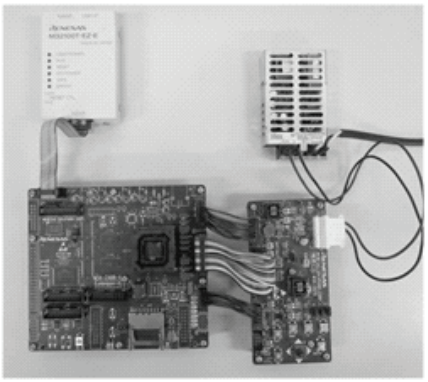
図 3.10: CPU と MX-1 を並列に処理するイメージ.

上述の最適化実装を評価するため，MX-1 によるモルフォロジカルパターンスペクトラム処理を評価ボードとシミュレータ (HEW) で実装し，処理の時間計測を行った．処理の時間計測は，32 × 32 の画像を入力し，スケールアップ回数を $n = 15$ とし，パターンスペクトラム結果が出力されるまでの時間である．HEW シミュレータ及び評価ボードの実装環境を図 3.11 に示す．HEW シミュレータは，これまでに開発した評価ボードにて検証を重ね，より改良を加えることができるよう構築したものであり，HEW シミュレータのみユーザマイクロコードを生成でき，処理の最適化実装が可能となる．なお，HEW シミュレータは，通常実装及び最適化実装を比較して評価することが可能である．評価ボードは，MX-1 のテストチップが組込まれており，ハードウェアでの動作検証が可能となる．評価ボードと HEW 上で同様のプログラムを動作させた場合の処理時間と HEW 上で最適化実装を行った場合の処理時間の結果を表 3.2 に示す．また，HEW 上ではユーザマイクロコードを生成することで，処理の最適化実装について検証をした．表 3.2 から，CPU と MX-1 を並列に実行した場合，約 5 倍の処理速度向上が確認できた．また，通常の実装及び最適化実装の各処理にかかるサイクル数内における MX-1 と CPU の処理比率に関する結果を図 3.12 に示す．この結果から，MX-1 のサイクル数にほとんど変化はないが，全体の占める制御用 CPU のサイクル数が削減でき，全体の処理サイクル数が大幅に削減できていることが確認できる．評価ボードについては，CPU と動作周波数が異なるため，単純な比較はできないが，ハードウェアとして確実に動作することがわかった．

HEW simulator



Evaluation board



Type		SH-2A (HEW)	M32R (Evaluation board)
CPU	Operating frequency [MHz]	200	81
MX core	Operating frequency [MHz]	200	162
Power consumption [mW]		123.5	200

図 3.11: MX-1 の評価ボードと HEW の詳細.

表 3.2: 単純実装と最適化実装による処理結果.

	Program	Processing time [ms]
Evaluation Board	Straight forward implementation	51.71
HEW	Straight forward implementation	21.37
	Optimized implementation	4.4

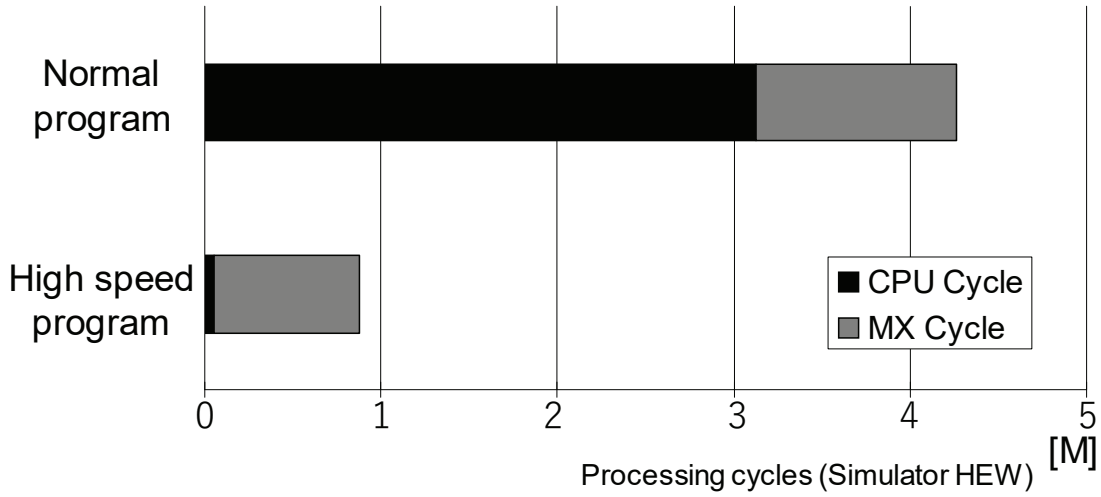


図 3.12: 単純実装と最適化実装によるクロックサイクル数.

さらに，MX-1 の処理性能を客観的に評価することを目的に，既存の組込みプロセッサとの比較を行う．既存の組込みプロセッサとして，実際にモバイル機器に用いられているプロセッサを対象とし，MX-1 と同じくソフトウェアベースのコアであり，低消費電力下で動作可能なハードウェアコストの低いものを選ぶ．比較対象としては，テキサスインスツルメンス社の DM3730，AMD 社の Geode LX800，Intel 社の Atom N450 を選定し，それぞれの仕様を表 3.3 (1)∼(4) に示す．

TI DM3730 は ARM 社のコア ARM Cortex-A8 AM3715 が組込まれており，市場の普及率が高いプロセッサであり，超小型組込みボード BeagleBoard 等にも搭載されている [102]．また，ARM コアには NEON と呼ばれる SIMD 回路も組込まれており，MX-1 と同様に並列処理を実現することが可能である [103]．さらに，ノートパソコンに使われている Geode LX800，Atom (TM) N450 を用いて，モルフォロジカルパターンスペクトラム処理を C 言語で実装した [104], [105]．各プロセッサを用いて，MX-1 で実装したモルフォロジカルパターンスペクトラム処理と同様 (対象画像：32×32 ピクセル，スケールアップ回数 $n = 15$) にプログラムを実行し，処理時間の計測を行う．その結果を表 3.3 (5)∼(8) に示す．まず初めに，MX-1 と逐次処理で実行される AMD Geode LX800 及び Intel Atom(TM) N450 の結果を比較する．なお，逐次処理として実行するため，SSE 命令等は使用せず，通常の使用で測定している．MX-1 は動作周波数が低いにも関わらず並列処理を実行した結果，AMD Geode LX800 と処理時間を比較すると，約 88 倍高速化でき，Intel Atom(TM) N450 と比較すると約 21 倍も高速化できている (表 3.3 (5))．次に，

MX-1 と NEON による並列処理が可能な回路を組込んでいる ARM TI DM3730 の処理時間を比較すると，23 倍高速化されている (表 3.3 (5))．さらに，スループットについて比較すると，32×32 ピクセルである対象画像を処理するのでデータ量は 8,192 ビットを処理したことになり，MX-1 と ARM TI DM3730 を比較すると約 22 倍も向上することが確認できる (表 3.3 (6))．また，消費電力当たりのスループットについても約 81 倍の向上が確認できる (表 3.3 (7))．さらに，プロセスルールの違いを考慮した消費電力の評価をスケーリング則を用いて行う．プロセスルールが $1/s$ となると消費電力は $1/s^2$ となることから，全てのプロセッサが MX-1 と同様の 90 nm プロセスであると仮定し，消費電力を見積もる (表 3.3 (4))．この結果，MX-1 は消費電力当たりのスループットは他のプロセッサよりも大きくなることを確認した (表 3.3 (8))．具体的には，AMD Geode LX800 より約 4000 倍，Intel Atom(TM) N450 より約 145 倍，ARM TI DM3730 より約 20 倍である．以上から，比較対処のプロセッサは組込み機器で利用されており，MX-1 はこれらのプロセッサよりも高性能・小型に実現することが確認できたことから組込み機器向けのプロセッサとして利用可能であると考えられる．さらに，MX-1 から改良を加えた CAMX においても同様の結果が得られると考えられ，CAMX についてもマルチメディア処理を容易に実行できると考えられる．

表 3.3: MX-1 の評価ボードと既存のモバイルデバイス向けプロセッサとの処理性能.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	Operating frequency [MHz]	CMOS process [nm]	Power consumption [mW]	CMOS process-based power consumption [mW]	Processing Time [s]	Throughput [kbps]	(6) / (3) Throughput per power consumption [kbps/mW]	(6) / (4) Throughput per CMOS process-based power consumption [kbps/mW]
MX core	81	90	200	200.00	0.05	158.40	792.02	0.7920
ARM TI DM3730 (ARM AM3715 Cortex-A8 with NEON Circuit)	1,000	45	731	182.75	1.15	7.11	9.73	0.0389
AMD Geode LX800	500	130	3,600	7511.11	4.41	1.86	0.52	0.0002
Intel Atom™ N450	1,660	45	5,500	1375.00	1.09	7.50	1.36	0.0055

3.2.2 まとめ

本節では，マトリクスアーキテクチャ型超並列演算プロセッサ MX-1 を用いて，モルフォロジカルパターンスペクトラムをモバイルデバイスの環境で構築を行った．この結果，MX-1 により，プログラムのユーザマイクロコード化により処理速度が約 5 倍向上し，一般的に普及しているモバイル向けのプロセッサと比較を行ったところ，MX-1 は消費電力当たりのスループットは約 22 倍も高い数値となった．また，プロセスルールの違いを考慮した消費電力当たりのスループットでは約 20 倍もの高い数値となることを確認した．すなわち，MX-1 は低消費電力でも高いスループットを実現でき，特に，モルフォロジカルパターンスペクトラム処理の様な繰り返し演算を必要とするような画像処理に対してはより有効なプロセッサになり得ることが確認できた．また，MX-1 はプログラムを入れ替えることで様々な処理が可能であるため，モバイルデバイスに搭載しやすいことも特徴である．以上のことから，MX-1 を改良し，処理速度を向上した CAMX に実装した場合はより高性能になることが想定される．このため，CAMX においてもモバイルデバイス向けのアクセラレータとして有効に機能することが考えられる．さらに，CAMX は MX-1 と同様の構成をしているため，モルフォロジカルパターンスペクトラム処理の様な繰り返し演算を必要とするマルチメディア処理を高性能に実現することが可能となる．

3.2.3 盗撮防止システムの提案

本節では，将来的に CAMX を組込み機器へ実装することを想定し，組込み機器を用いたアプリケーションの構築について検討する．特に，組込み機器として知られているスマートフォンでのアプリケーションについて議論する．

背景

近年，カメラ機器を搭載した組込み機器が急激に普及しており，その典型としてスマートフォンが知られている．しかし，このスマートフォンの普及に伴い，一般人が瞬時に高画質な写真を容易に撮影できるようになり，犯罪の 1 つである盗撮が社会問題となっている [106]–[111]．この犯罪は，被写体に許可なく，隠れて写真や動画を撮影する行為であり，公共の場で相手に許可なく隠れて撮影 [106]–[109] したり，書籍などを販売する店舗で書籍を購入せずに隠れて書籍の情報を撮影 [110], [111] したり，職場で機密情報を撮影して外部に流出させたりすることである．例えば，韓国やアイルランドでは，旅行先のホテルで宿泊者に許可なくカメラが設置され，宿泊中のプライベートな状況が外部に流出していた [108]．シンガポールでは，大学のバスルームで許可なく入浴中の同僚を撮影していた [109]．これらの行為はホテル，空港，駅等の公共の場でスマートフォンを用いて行われることがほとんどであり，被写体の対象としては女性であることが多い [112]．このため，近年，女性の権利を訴える活動が増えている．2021 年に日本においてオリンピックが開催されたが，元女性アスリートが女性アスリートの権利を守るために盗撮に対する対策をオリンピック委員会に訴えたこともある [113], [114]．さらに，アメリカでは盗撮という犯罪に対する法律も制定するようになった [115], [116]．この様に，盗撮に対する社会問題は大きなものとなってきている．しかしながら，盗撮に対して，法律等の対策は行われているが，技術的な対策については進んでいないのが現状である．

この様に盗撮という犯罪行為は世界的に大きな問題となっており，カメラ機能を直接的に操作する様な対策が求められている．将来，モバイルカメラ機器にハードウェアの面からカメラ機能を制御する新たな対策が必要となっている．しかし，現在まで盗撮に対する対策は以下の様な物理的な方法がほとんどであり，カメラ機能に対する直接的な方法は考えられていないのが現状である．

- 盗撮の対象となりやすい女性の体や書籍に対して何かしらのもので隠す．
- 試着室，トイレ，ホテル等，プライベートな空間を利用する前に，見知らぬモバイルカメラ機器が設置されていないか確認する．

- 盗撮を行わないように注意喚起するポスターを設置する．
- 盗撮行為をに関する罰金付きの法律等を定める．
- 既存の技術である，撮影時のシャッター音，カメラモジュールに貼る盗撮防止シール，赤外線によるノイズ付加技術を利用する．

以上の方法では盗撮を行うとするモバイルカメラ機器に対するハードウェア的な対策はなく，盗撮という行為に明確に対策できていない．このことから考えると，盗撮という行為に対する防止策は，“簡単に構築できること”，“日常生活で利用しやすいこと”，“盗撮の種類毎に防止可能な汎用性があること”，“設置等のコストがかからないこと” 及び “カメラ機能を直接的に操作できること” が重要となる．そこで，我々は組込み機器のカメラに対する直接的な対策として，LED 照明とスマートフォンの連携による盗撮防止手法を構築した．本手法は，LED 照明から特定のパターンである可視光ビーコンを発生させ，スマートフォンで画像処理を行うことで，特定のパターンを検知した場合にスマートフォンのカメラ機能を停止させる手法である．

既存技術について

本節では，既存の技術である，撮影時のシャッター音，カメラモジュールに貼る盗撮防止シール，赤外線によるノイズ付加技術について説明する．

まずは，撮影時のシャッター音について説明する．撮影時のシャッター音はたいていのスマートフォンにあらかじめ組込まれている [117]．しかしながら，このシャッター音は設定から容易に停止させることができ，さらには，シャッター音が出ないアプリケーションが登場している．また，日本では撮影時にシャッター音が出るよう設定されているが，海外ではそのような設定を見つけることはできない．さらに，シャッター音があることにより，静かな空間で風景等を撮影したい時に，シャッター音により静かな空間の妨げとなる．この様な理由から，撮影時のシャッター音は日常生活において利用しづらい技術である．加えて，撮影時にシャッター音は発生するが，シャッター音が発生しても自由に撮影は可能であるため，盗撮という行為に対する防止手段とはならない．また，撮影時にシャッター音を出すだけなので，盗撮の種類毎に汎用性のある防止策を講じることはできない．一方，シャッター音を導入することは，設定等により容易に付加できるため，設置等のコスト低く，構築も容易である．以上の特徴を表 3.4 に示す．

表 3.4: シャッター音の特徴

Ease of construction	✓	- Just changing the camera settings - Just installing an application
Introduction to daily lives	N/A	- Constituting a nuisance in daily lives
Versatility for preventing spy-camera activities	N/A	- Generating near-noise shutter-sound in surrounding
Development cost	✓	- Just changing the camera settings - Just installing an application
Direct control on camera function	N/A	- Unstopping the smartphone camera from taking pictures by only shutter-sound

次に，モバイルカメラ機器のカメラモジュールに貼り付ける盗撮防止シールについて説明する [118], [119]．盗撮防止シールはカメラモジュールに直接シールを貼るため，カメラ機能を起動したとしても画像等が取得できなくなる手法である．そして，シールを外した場合には，シールを外したことがわかるよう，シールの跡が付く仕組みとなっている．この手法は，撮影が禁止されている工場，研究所等に入る前に，モバイルカメラ機器の利用者に自らシールを貼り付けてもらうことで，盗撮を防止する．この手法はシールを利用するため，設置等のコストを安価に抑えることができ，モバイルカメラ機器にシールを貼り付けるだけであるので簡単に設置できる．また，モバイルカメラ機器のカメラモジュールにシールを貼り付けるので，カメラ機能を直接操作できる．しかし，盗撮防止シールをモバイルカメラ機器に都度貼り付けてもらう必要があり，日常生活において導入しづらい．そして，モバイルカメラ機器のカメラモジュールに盗撮防止シールを貼り付けることしかできないため，盗撮行為毎に合わせた対策は難しい．以上の特徴を表 3.5 に示す．

表 3.5: 盗撮防止シールの特徴

Ease of construction	✓	- Just placing seal on camera devices
Introduction to daily lives	N/A	- putting and removing the seal off the camera every now and then
Versatility for preventing spy-camera activities	N/A	- Just be put on the camera
Development cost	✓	- Paper seal is inexpensive
Direct control on camera function	✓	- Placing seal on camera lens

次に，赤外線によるノイズ付加技術について説明する [120]–[123]．本手法は，盗撮が禁止されている対象物に対して，赤外線を照射することで，モバイルカメラ機器のカメラ機能により撮影された画像にノイズが出るようにする技術である．この技術は，盗撮が禁止されている対象物が決まっている映画館のスクリーン，演劇の舞台等で利用される．スマートフォンの一種である iPhone を発売しているアップル社においても，本手法による特許を取得している．赤外線によるノイズ付加技術については，赤外線を照射するための特別な装置を新たに構築する必要があり，これから示す我々が提案する盗撮防止技術の様に LED 照明を取り換えるだけで構築できる手法より，日常生活では利用しづらく，設置等のコストもかかるため，簡単に構築できない．さらに，赤外線は人間の目に対して不快感を与えないが，可視光よりも特別な波長となるため重要文化財等に悪影響を及ぼす可能性もあるため，日常生活では利用しづらい要因ともなる [124]．一方，盗撮する者がカメラ機能を起動した場合，カメラモジュールから取得した画像にノイズが発生するため，盗撮という行為に対してカメラ機能を直接制御できる．また，赤外線に情報等を付加したり，ノイズの強度を変更することで，盗撮の種類毎に様々な処理ができる可能性がある．以上の特徴を表 3.6 に示す．

表 3.6: 赤外線によるノイズ付加技術の特徴

Ease of construction	N/A	- Installing special and new equipment - Not being the daily used equipment
Introduction to daily lives	N/A	- Only using in a few established places
Versatility for preventing spy-camera activities	✓	- Having a wide range of other uses depending on how the application is created
Development cost	N/A	- Installing special and newly equipment - Not being the daily used equipment
Direct control on camera function	✓	- Distorting photos and videos taken by mobile camera

盗撮防止手法（SPS）の構成

本節では，盗撮を防止するための手法（Spy-camera Preven system：SPS）[125]–[132] を提案する．SPS は LED 照明とスマートフォンの様なモバイルカメラ機器を連携することで，モバイルカメラ機器の機能を停止させる手法である。LED 照明は盗撮を防止したい部屋や機密情報を扱う工場に設置されている照明を SPS の機能が搭載された LED 照明に取り換えるだけである．この SPS の機能を搭載した LED 照明は人間の目では一般的に利用する照明と差異がないため，日常生活の照明としても利用できる．一方，モバイルカメラ機器には高速フーリエ変換処理（FFT:Fast Fourier Transform）が可能となるよう組込まれており，この処理により特定の周波数を抽出可能となる．モバイルカメラ機器は LED 照明の光に含まれた可視光ビーコンを FFT 処理により特定の周波数を抽出し，特定の周波数を検知した場合にカメラ機能を停止させたり，警告メッセージを表示したりする．すなわち，LED 照明の光が届く範囲は盗撮を行ってはいけない空間となり，可視光ビーコンが盗撮防止のトリガとなる．さらに，この可視光ビーコンが含まれた光は，人間の目に対して不快感を与えることはないため，日常生活における LED 照明としても利用可能である．

SPS は盗撮を行おうとしているモバイルカメラ機器からプライバシー保護と機密情報の流出防止を目的としており，以下に SPS の特徴をまとめる．

- SPS は LED 照明とモバイルカメラ機器を連携させることによって，強制的にカメラ機能を停止させる．この LED 照明の光が届く空間は盗撮を防止でき，LED 照明を取り換えることで盗撮を防止したい空間を任意に設定可能である．

- SPS の機能が組込まれた LED 照明は日常生活においても利用可能である．LED 照明からの光が人間の目に対して不快感を与えないよう，人間の目に対して快適であると言われる 1/f ゆらぎに正弦波を組合わせた信号をトリガとする．この正弦波をモバイルカメラ機器において検出し，検出した場合にモバイルカメラ機器のカメラ機能が停止する．

- 正弦波の周波数を変更することで，モバイルカメラ機器で様々な処理が可能となる．例えば，強制的にカメラ機能を停止したり，モバイルカメラ機器のモニタに警告メッセージを表示させる等である．

SPS の詳細について説明する．LED 照明から信号を送受信する技術は可視光通信という技術で知られている [133]–[142]．この技術は，人間の目等に影響を与えない通信として知られており，安価にシステムを構築でき，大容量のデータも送

信可能である．さらに，可視光は直進性があるため，壁等により通信を遮ることが可能である．一方，提案する SPS は可視光通信の一種ではあるが，トリガを送信するだけであるため小容量のデータ転送のみで実現可能である．また，モバイルカメラ機器のカメラがどこを撮影していたとしてもこのトリガを受信できるようにする．そして，壁等で通信を遮ることができるため，盗撮防止空間を光が届く範囲でのみ任意に構築可能である．すなわち，この盗撮を防止したい空間で，モバイルカメラ機器のカメラ機能が起動された場合，カメラモジュールからから可視光のトリガを短時間で受信し，カメラ機能を停止させることができる．また，任意の盗撮防止空間でのみこの機能は有効となるため，この空間にないモバイルカメラ機器のカメラ機能は通常通り動作する．図 3.13 に SPS の利用イメージを示す．図 3.13 (a) では，本屋の天井に設置されている証明を SPS が組込まれた照明に変更した場合のイメージである．天井の LED 照明から可視光のトリガが送信されており，顧客が未購入の書籍に記載された情報をスマートフォンのカメラ機能で撮影しようとした時にカメラ機能が停止するイメージである．図 3.13 (b) では，駅や空港等の公共施設に設置されているエスカレータにおいて，撮影される人の許可なく撮影する場合である．エスカレータの天井に設置された LED 照明には SPS が組込まれており，盗撮を行おうとする者がスマートフォンのカメラ機能を起動した時に可視光のトリガを受信し，カメラ機能が停止するイメージである．図 3.13 (c) では，近年普及しているテレワークで利用する WEB カメラを悪意のあるものがインターネット経由でアクセスされた場合である．この場合，天井には SPS が組込まれた LED 照明が設置されているため，もし悪意のある者が WEB カメラを通して盗み見をしようとしても，可視光のトリガを受信してカメラ機能が停止するイメージである．図 3.13 (d) では，工場等において悪意のある従業員等が機密情報を流出させようとした場合である。この場合，工場の照明を SPS が組込まれた LED 照明にすることで，もし悪意のある者がスマートフォンや隠しカメラで撮影しようとしてもカメラ機能が停止し，撮影ができなくなるイメージである．一方，図 3.13 (e) は SPS が組込まれた LED 照明から可視光のビーコンを送信しない場合である．例えば，駅等で管理者が車両等の展示のために撮影を許可したい場合，LED 照明から可視光のビーコンを送信しないようにすることで，スマートフォンで自由に撮影が可能となるイメージである．図 3.14 に SPS の流れを示す．可視光のビーコンは LED 照明で生成され，照明の光と共に送信される．そして，可視光は壁や物体等で反射しながら，スマートフォン等のカメラモジュールに到達する．カメラモジュールで受信した可視光は，カメラモジュール内で画素値として認識される．この画素値の変化が可視光のビーコンであり，スマートフォン等のモバイルカメラ機器内で FFT 処理が実行され，特定の周波数が抽出される．この抽出された周波数があらかじめ設定された値と一致した場合にカメラ機能を停止する．

なお，この周波数を変更することで，カメラ機能を停止させたり，警告メッセージを表示させる等，様々な処理が可能となる．

これまでの研究では，SPS のプロトタイプを構築（図 3.15）しており，本節では，可視光のビーコンについて議論し，CAMX を組込むことを想定しているモバイルカメラ機器の処理について説明する．

LED 照明の可視光ビーコンについては，人間の目に対する不快感を考慮し，1/f ゆらぎと正弦波を組合わせた波形を採用しているが，以前は線形則による波形及びスティーヴンスのべき乗則による波形を採用していた．線形則による波形は LED 照明の照度を時間当たり一定の間隔で変化することで作成できる．スティーヴンスのべき乗則による波形は，人間の目は強い明るさの変化に対して鈍感であることを利用し，LED 照明の照度を対数的に変化することで作成できる．

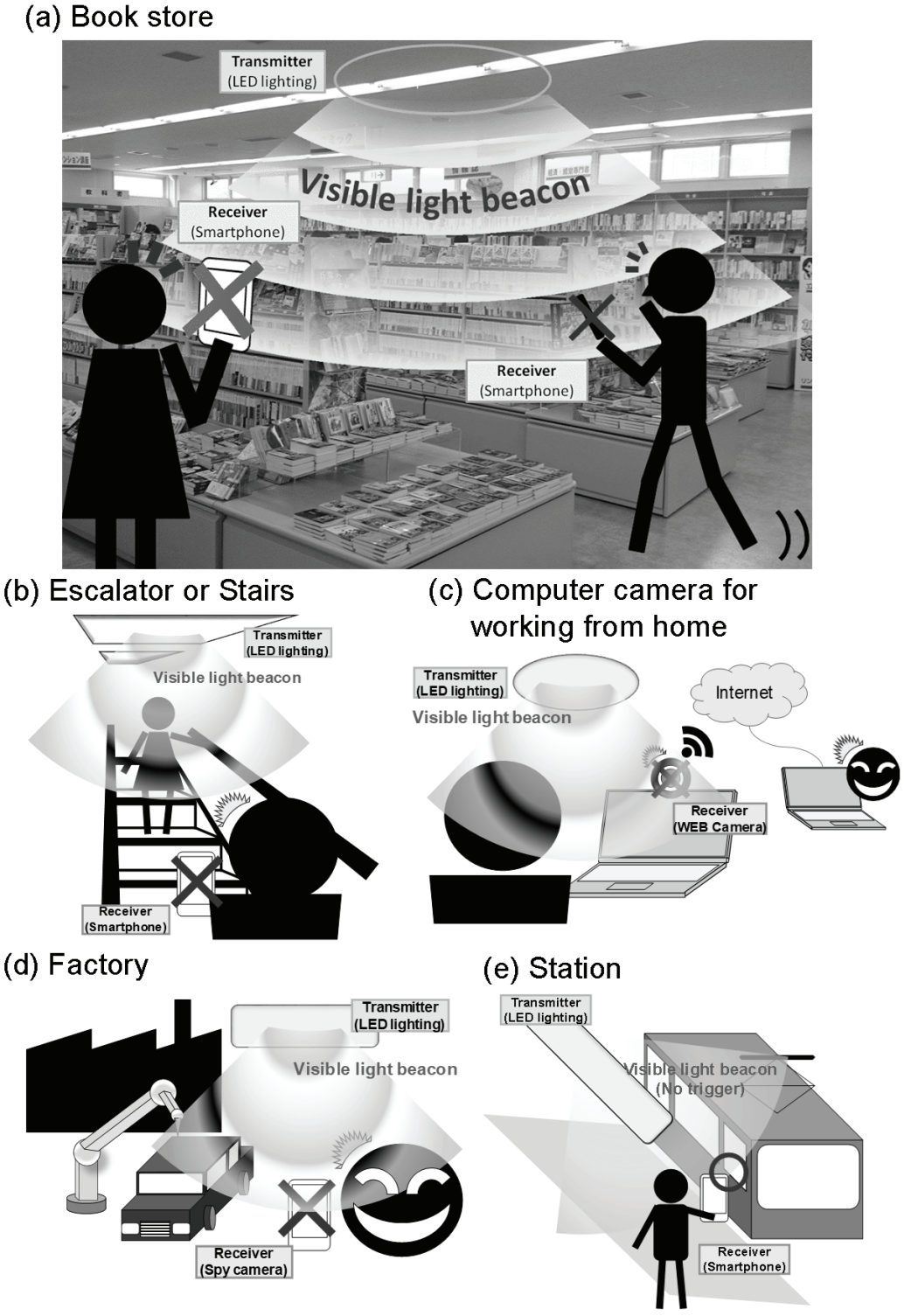


図 3.13: SIPS の使用例

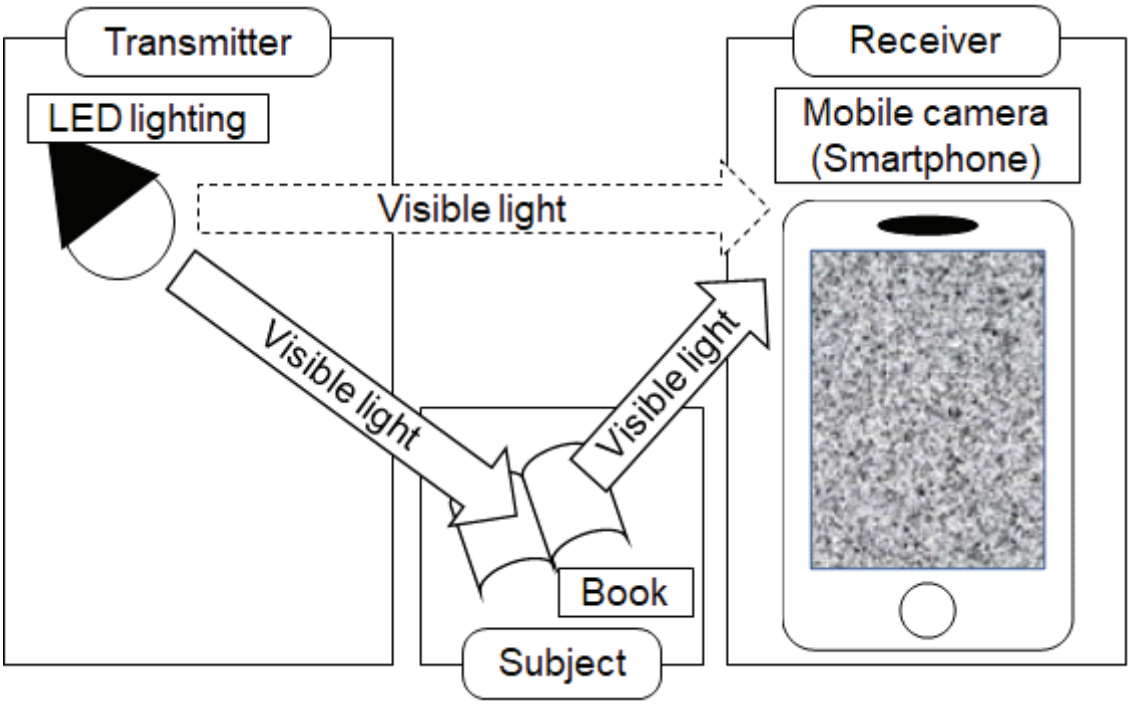


図 3.14: SPS のシステム構成.

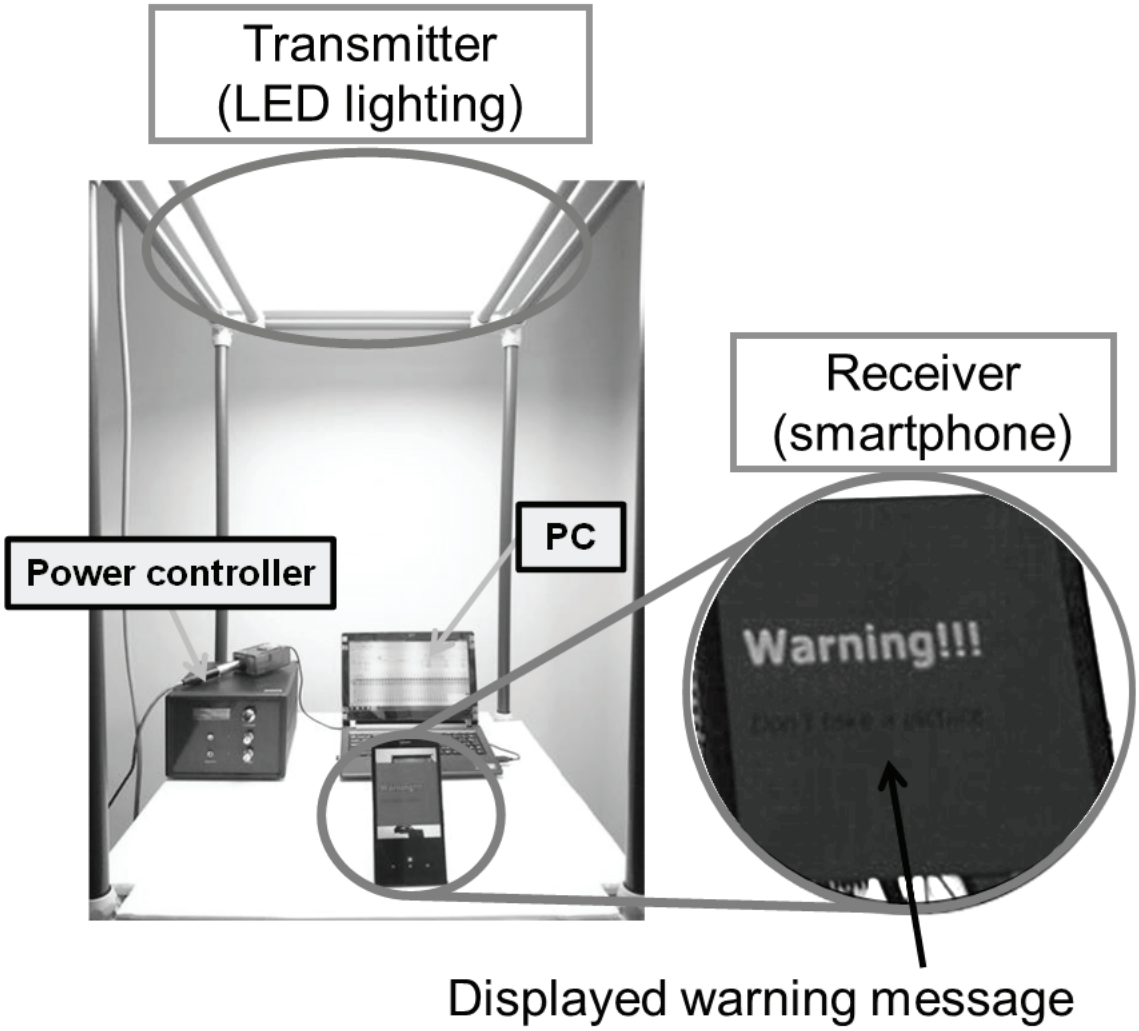


図 3.15: SPS の動作確認

ここで，カオスアルゴリズムを用いた 1/f ゆらぎの波形作成について説明する．1/f ゆらぎは自然現象で発生し，人間の目に対して快適であることが知られている [143]–[145]．例えば，蛍の光やろうそくの炎等である．1/f ゆらぎを照度変化で表し，その波形を周波数分解すると図 3.16 (a) のようになる．図 3.16 (a) の上図は時間に対する照度変化を表す．図 3.16 (a) の下図は，上図を FFT 処理により周波数分解したものとなる．上図の縦軸は照度の振幅であり，横軸が時間である．下図の横軸は周波数であり，縦軸は周波数の強さを示すパワースペクトラル密度である．一般的に，このパワースペクトラル密度の傾きが- 1 に近いほど，人間の目に対して快適であると言われている．理想値である- 1 の傾きは点線で表し，上図の波形から得られたパワースペクトラル密度の傾きを実線で表す．同様に，線形則による照度変化及び周波数分解の結果は図 3.16 (b) となる．図 3.16 (b) の上図は時間に対する照度変化であり，下図は FFT 処理により周波数分解した結果となる．スティーヴンスのべき乗則による照度変化及び周波数分解の結果は図 3.16 (c) となる．図 3.16 (c) の上図は時間に対する照度変化であり，下図は FFT 処理により周波数分解した結果となる．以上の結果から，図 3.16 (a) の 1/f ゆらぎの場合が最も- 1 に近く，線形則による波形が最も- 1 から離れていることを確認した．すなわち，1/f ゆらぎは人間の目に対して快適であることがわかる．さらに，カオスアルゴリズムによる 1/f ゆらぎの生成手法を式 (3.9) に示す．$X(t)$ は 1/f ゆらぎの値であり，t は時間，$X(t+1)$ は $X(t)$ の次時間に変化する値である．

$$
\begin{aligned}
&X(t) < 0.5 \\
&\quad X(t+1) \;=\; X(t) + 2 * X(t)^2 \\
&X(t) \geq 0.5 \\
&\quad X(t+1) \;=\; X(t) - 2 * (1 - X(t))^2
\end{aligned}
\tag{3.9}
$$

本節では，この 1/f ゆらぎに正弦波を組込んだ，人間の目に対して快適な新たな可視光ビーコンを提案する．この正弦波が SPS におけるトリガとなり，モバイルカメラ機器において LED 照明の光を FFT 処理で正弦波の周波数を抽出することになる．さらに，1/f ゆらぎと正弦波を組合わせた可視光ビーコンは 1/f ゆらぎと同様に人間の目に対して不快感を与えない信号となり，日常生活の照明としても利用可能である．1/f ゆらぎと正弦波を組合わせた可視光ビーコンは，式 (3.10) で生成され，SPS の LED 照明ではこの可視光ビーコンを生成し送信することになる．

$$
V(t) = X(t) + sin(2\pi * f * t) \tag{3.10}
$$

$V(t)$: visible light beacon from the LED lighting

beacon embedded with
1/f fluctuation-sine-waves;
f : frequency;
t : time

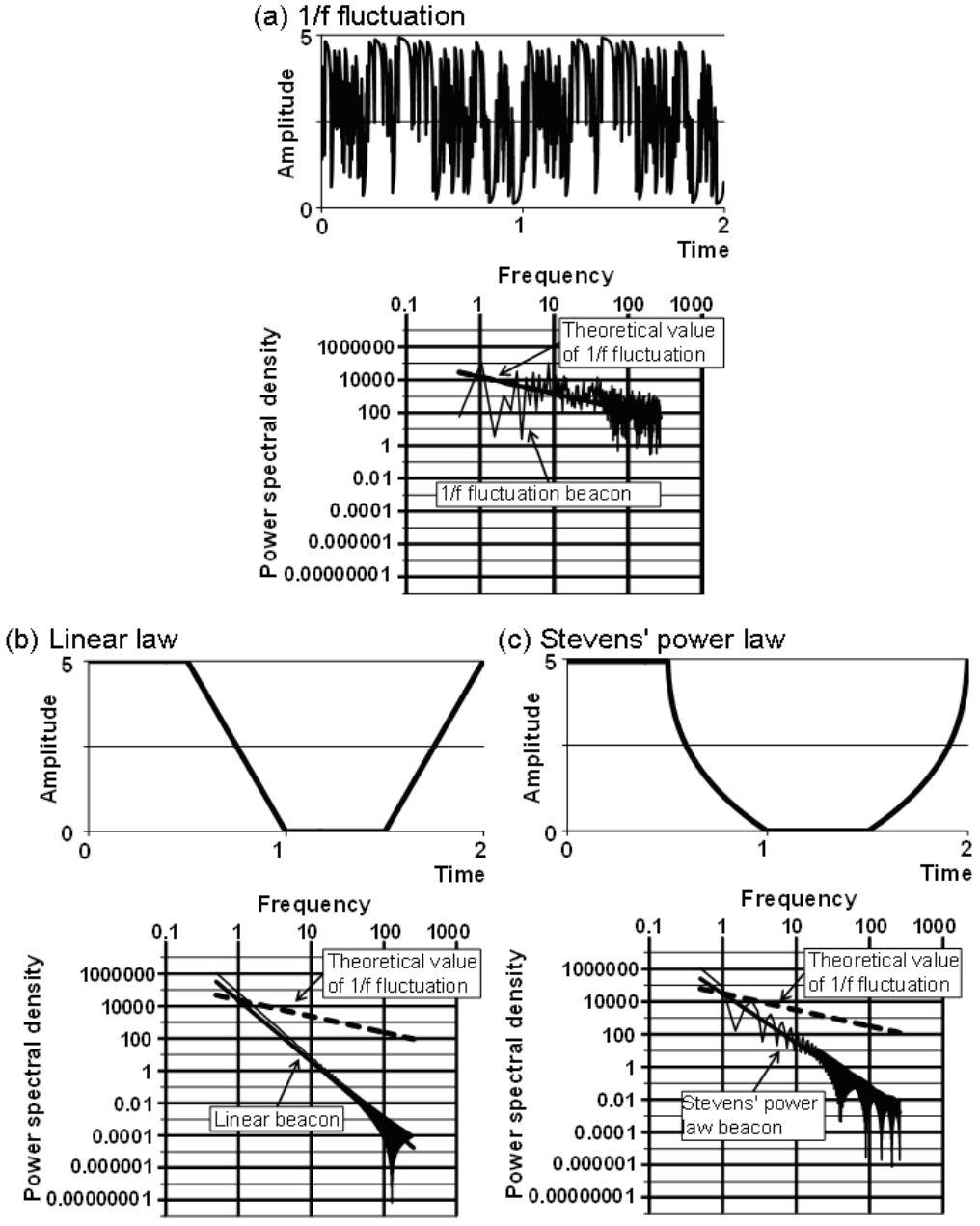


図 3.16: 可視光ビーコンの比較

次に，モバイルカメラ機器における可視光ビーコンの検出手法について説明する．スマートフォン等の様なカメラ機能を備えたモバイルカメラ機器は，一般的に 30 frame per second (fps) でカメラモジュールから画像を取得している．しかし，モバイルカメラ機器ではバックグラウンドにて通信や他アプリケーションの処理が常に動作しており，必ずしも 30 fps で画像を取得できない．また，可視光ビーコンを抽出するために FFT 処理を実行することになるが，FFT 処理は入力データを 2 のべき乗としなければならない．すなわち，SPS の受信機は 2 秒あたり 32 フレームの画像を取得することで，可視光ビーコンを抽出することが可能である．そして，可視光ビーコンを検出した受信機は，あらかじめ設定した周波数に一致することを確認すると，カメラ機能を強制的に停止させる．この処理の流れを図 3.17 に示す．

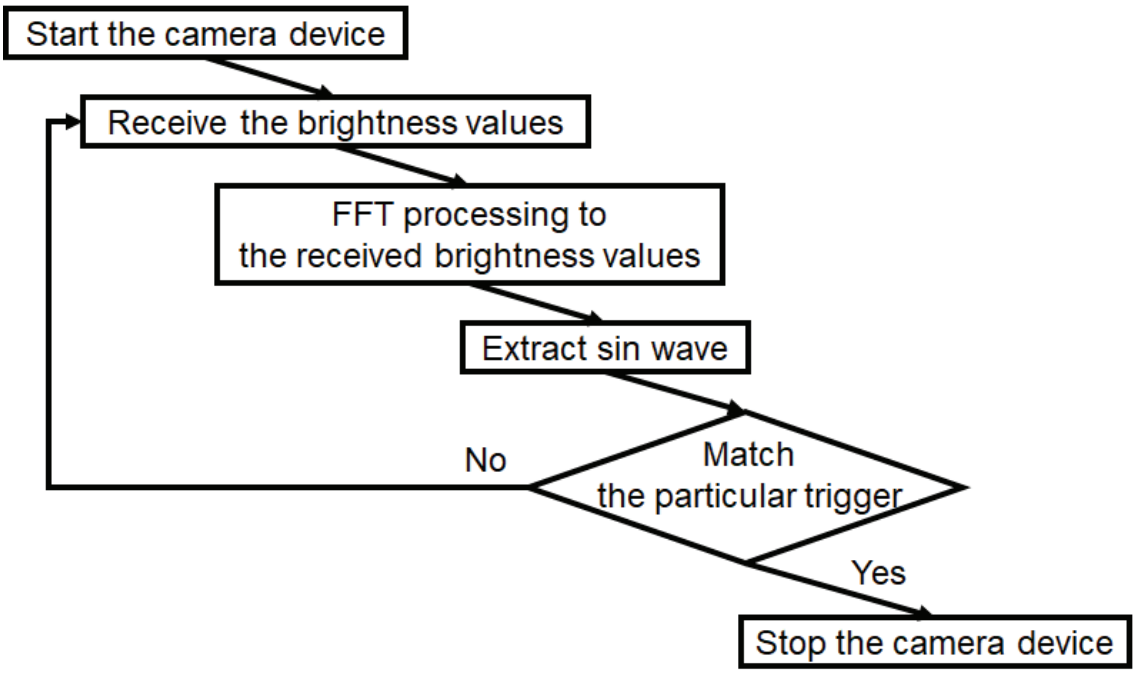


図 3.17: 受信機の処理フロー

SPS の実験結果

本節では，可視光ビーコンの送信機を LED 照明とし，可視光ビーコンを検出し，カメラ機能を停止する受信機としてスマートフォンを利用した．可視光ビーコンは 1/f ゆらぎと正弦波を組合わせ，LED 照明の照度を変化させることで生成及び送信する．受信機では，カメラモジュールにおける画素値の時間的変化を検知し，この値に FFT 処理により，トリガである正弦波の周波数を検出する．

始めに，SPS の受信機は可視光が届く範囲であれば距離や角度に関係なく可視光ビーコンを検出することができることを示す．図 3.18 に距離及び角度による可視光ビーコンの検出結果を示す．なお，本検証においては，受信機においてノイズの影響を受けやすく LED 照明の照度変化の検出がより厳しくなる条件として線形則による波形を用いた可視光ビーコンを用い，LED 照明とスマートフォンの距離及び角度の関係を明確に確認できるようにした．すなわち，線形則による波形を用いた可視光ビーコンを距離及び角度に関係なく受信できると，1/f ゆらぎと正弦波を組合わせた可視光ビーコンも距離及び角度に関係なく受信可能となる．LED 照明とスマートフォンの垂直距離は 2.0 m とし，スマートフォンと地面の垂直距離は 0.2 m とする．なお，スマートフォンの位置は，カメラモジュールの位置とする．X_m は LED 照明とスマートフォンの水平距離であり，0.5，1.0，1.5，2.0，2.5 及び 3.0 m と変化させる．φ はスマートフォンと地面の角度であり，90°，60°，45°，30° 及び 0° と変化させる．上述の条件により SPS の距離及び角度の影響について評価を行い，それぞれの条件において SPS が有効に機能するか確認する．図 3.18 に検証環境及びスマートフォンにいて SPS が機能した時間を示す．SPS の空間で，スマートフォンのカメラ機能を起動してからカメラ機能が停止するまでの時間を検証するため，それぞれの角度及び距離で 10 回の試行を繰り返し行った．図 3.18 のグラフは，縦軸にカメラ機能が停止するまでの時間，横軸に X_m の距離を示し，それぞれの φ の角度（90°，60°，45°，30° 及び 0°）における結果である．この結果から，全ての距離及び角度に関係なく，スマートフォンのカメラ機能が停止することを確認した．さらに，全ての角度において，30 秒以内にカメラ機能が停止している．すなわち，SPS の空間内でカメラ機能を起動した場合，カメラモジュールから可視光である可視光ビーコンが入射し，スマートフォン内で特定の周波数であるトリガを検出でき，スマートフォンを操作できることが確認できた．また，本検証ではノイズ等の影響を最も受けやすく，検出がより厳しい線形則による波形である可視光ビーコンを利用したため，SPS で利用する 1/f ゆらぎと正弦波を組合わせた可視光ビーコンにおいても同様の結果になると考えられる．さらに，1/f ゆらぎと正弦波を組合わせた可視光ビーコンの方がより少ないスマートフォンからの取得画像で特定の周波数を検出可能であるため，30 秒よりもさらに高速にス

マートフォンのカメラ機能を停止させることが可能である.

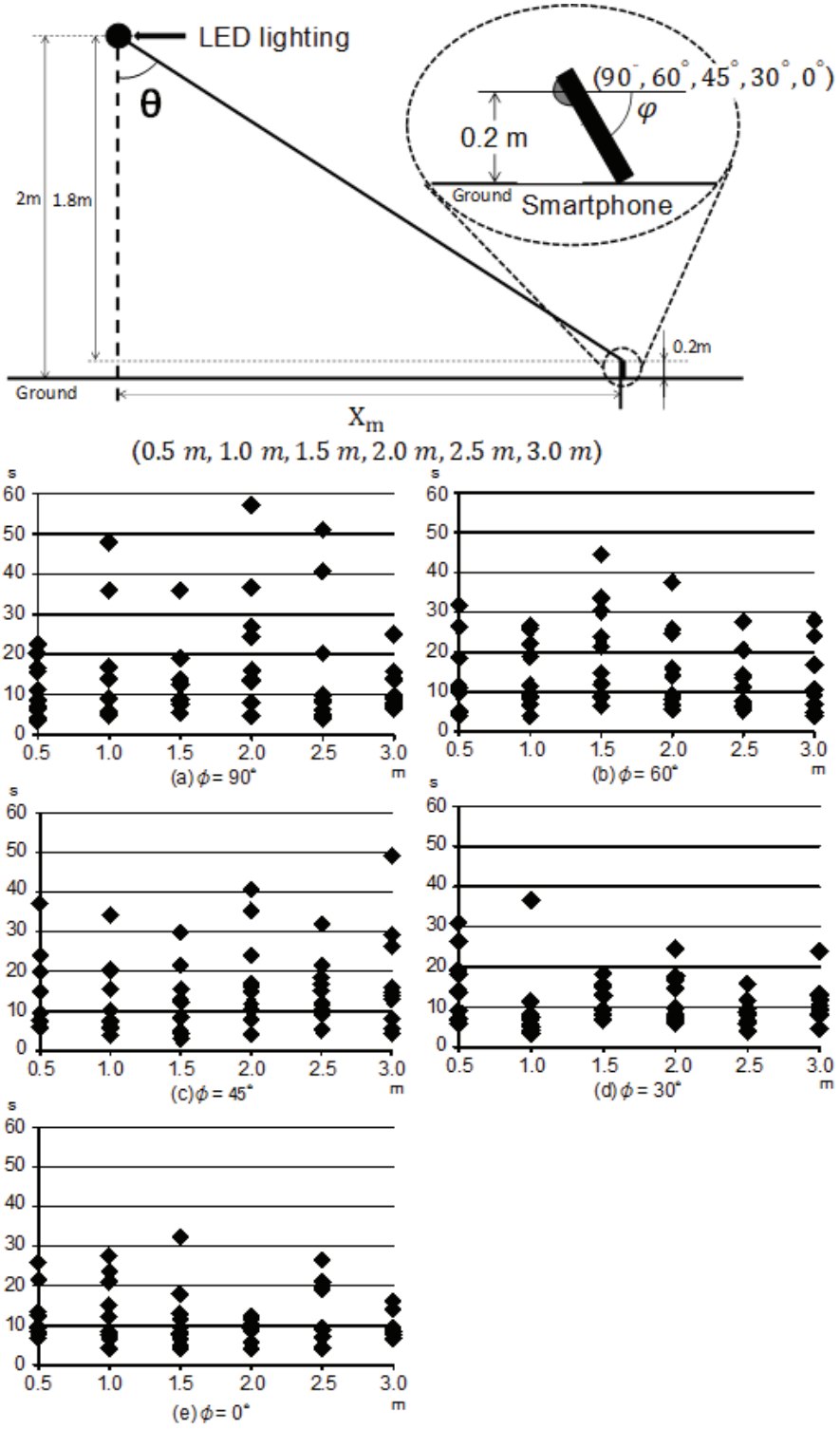


図 3.18: 角度と距離による SPS の有効性検証

次に，可視光ビーコンが人間の目に対して快適であるかを検証する．これまでの研究において，線形則による波形及びスティーヴンスのべき乗則による波形を可視光ビーコンとして利用してきた [146]．しかしながら，これらの可視光ビーコンは人間の目に対して不快感を与える場合があった．そこで，本節ではより人間の目に対して快適であると知られている 1/f ゆらぎを基にし，1/f ゆらぎと正弦波を組合わせた可視光ビーコンを提案する．図 3.19 (a) に線形則による波形及びスティーヴンスのべき乗則による波形を利用した可視光ビーコンを示す．横軸は時間であり，縦軸は LED 照明を操作する時の明るさの強さの設定値である．図 3.19 (b) は (a) の 1 秒から 2 秒を拡大したものである．この図からわかるように，スティーヴンスのべき乗則による波形は線形則による波形よりも対数的に変化し，LED 照明を操作する時の明るさの強さの設定値を 155 から 255 の間とし，4 秒間の間に変化するようにした．この設定で，43 人の被験者（20 代から 60 代）に対して，LED 照明の明るさによる不快感のアンケートを実施した．アンケートの結果を図 3.20 に示す．検証では，白色 LED 照明を用いて，パターン 1 は線形則による波形で点灯し，パターン 2 はスティーヴンスのべき乗則による波形で点灯した．さらに，パターン 3 は白熱 LED を常時点灯させた状態で，パターン 1 も点灯させた．パターン 4 は白熱 LED を常時点灯させた状態で，パターン 2 も点灯させた．43 人の被験者に，これら全てのパターンの LED 照明について，不快感に関する以下のアンケートを実施した（図 3.21）．

- あなたは明るさの変化を感じますか？
- あなたは日常生活においてこの照明を利用できますか？
- あなたはこの照明下において仕事等の作業をできますか？

この結果，50 %の回答者がパターン 1 及び 2 について，快適な照明でないと回答した．特に，パターン 1 及び 2 では，70 %の回答者は “あなたは明るさの変化を感じますか？” という質問に対して，“変化を感じる” と回答している．一方，パターン 3 及び 4 では，“あなたは日常生活においてこの照明を利用できますか？” という質問に対して，パターン 1 及び 2 の 1.5 倍の回答者が “利用できる” と回答している．そして，パターン 3 及び 4 では，“あなたはこの照明下において仕事等の作業をできますか？” という質問に対して，50 %の回答者が “作業できる” と回答している．しかしながら，線形則による波形及びスティーヴンスのべき乗則による波形共に，数人の方には不快感を与えていると確認できた．そのため，SPS においては，自然現象であり，人間の目に対して快適であると知られている 1/f ゆらぎを用いることで，線形則による波形及びスティーヴンスのべき乗則による波

形よりもさらに快適な可視光ビーコンを生成することが可能であると考えられる．また，我々の研究室において，1/f ゆらぎと正弦波を組合わせた可視光ビーコンを確認したところ，線形則による波形及びスティーヴンスのべき乗則による波形よりも照度の変化を感じにくいことを確認している．

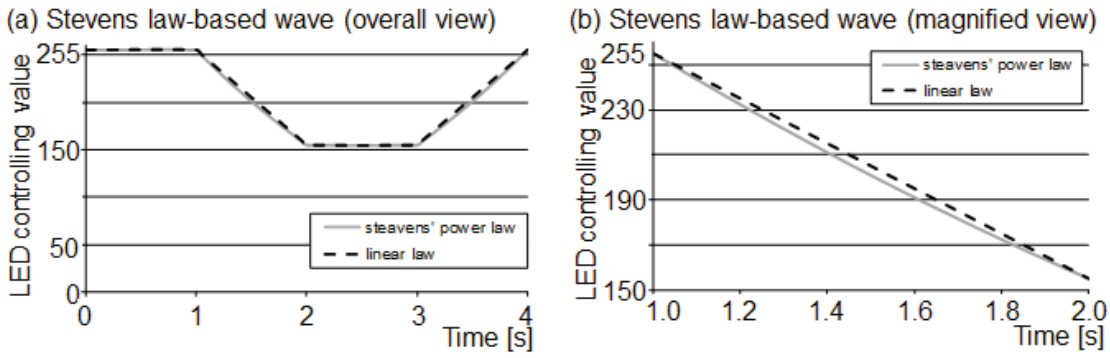


図 3.19: 波形及びスティーヴンスのべき乗則による波形における LED 照明コントローラの値変化

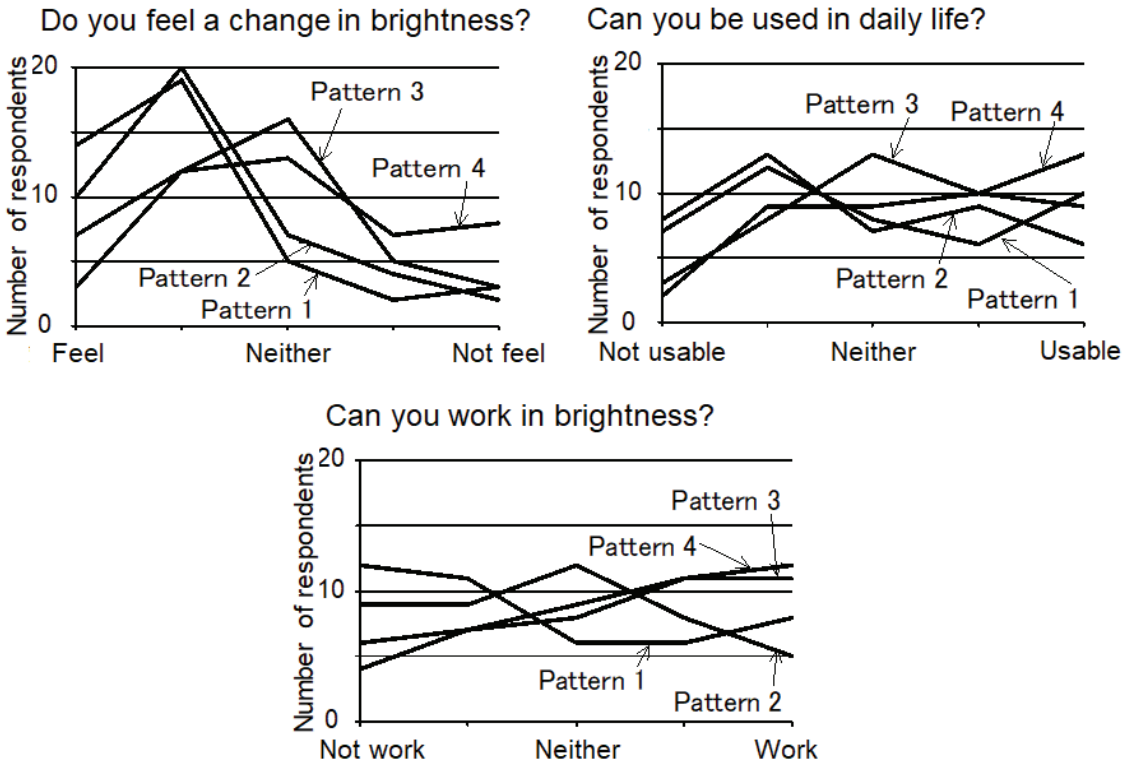


図 3.20: LED 照明の可視光ビーコンパターンによるアンケート結果

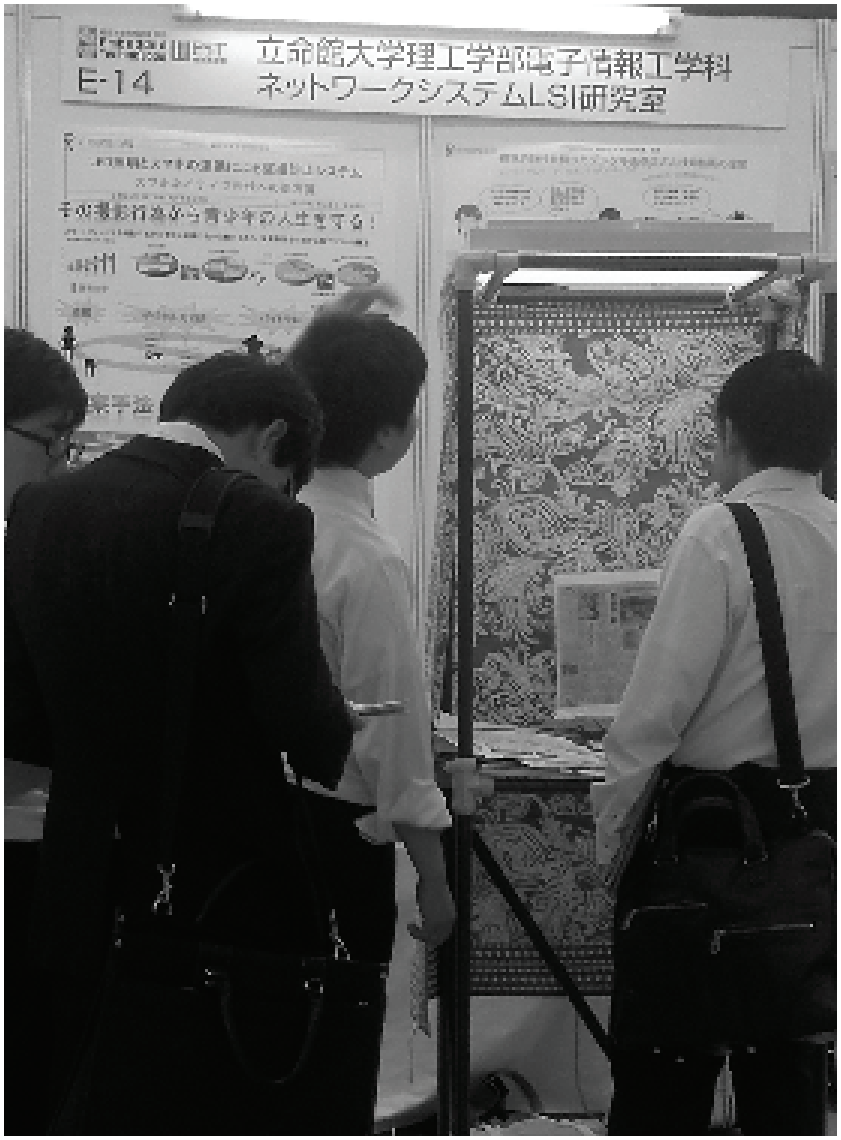


図 3.21: アンケート実施の様子

次に，1/f ゆらぎと正弦波を組合わせた可視光ビーコンから特定の周波数である正弦波の抽出について検証する．本検証では，LED 照明から 1/f ゆらぎと正弦波を組合わせた可視光ビーコンを送信し，スマートフォンでこのビーコンを受信し，受信した画素値の変化から特定の周波数を抽出することになる．また，スマートフォンには FFT 処理を組込んでいる．LED 照明の周波数は検証機器の機械的制限から 12.16，6.08 及び 3.04 秒とする．すなわち，正弦波の周波数は 0.08，0.16 及び 0.32 Hz と考えられる．さらに，1/f ゆらぎに組込む正弦波の振幅を 0，0.2，0.4，0.6，0.8 及び 1.0 と変更し，正弦波をスマートフォンで抽出する．図 3.22 に検証環境である LED 照明とスマートフォンの配置を示す．

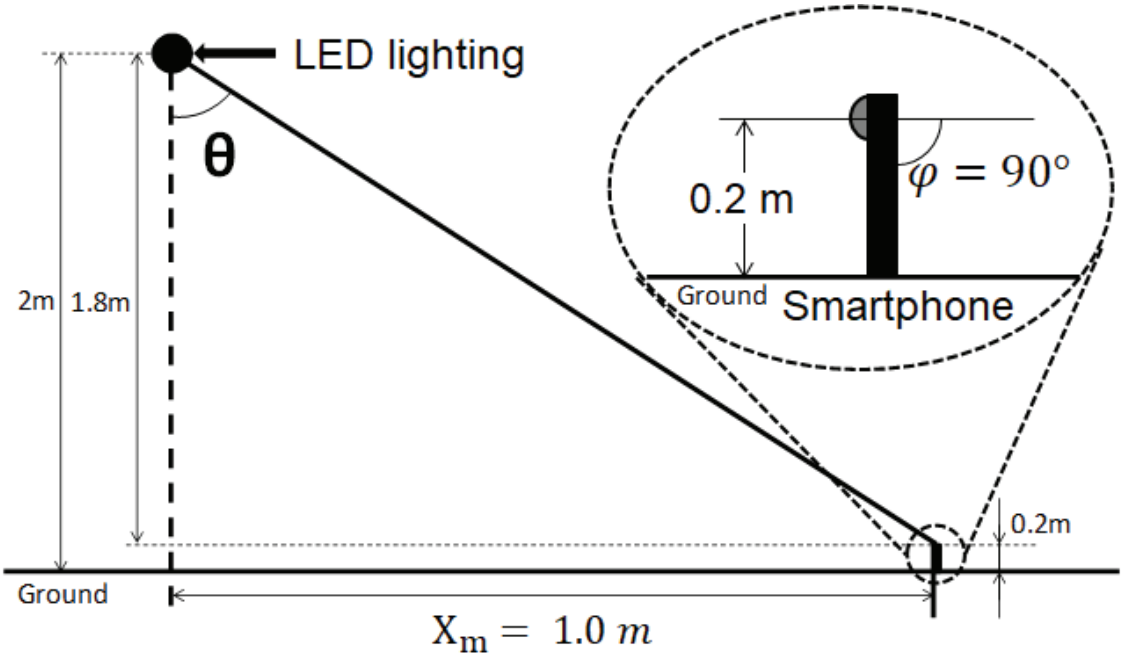


図 3.22: 送信機と受信機の配置

まずはじめに，1/f ゆらぎに 0.08 Hz の正弦波を組込んだ可視光ビーコンについて検証する．0.08 Hz の周波数は 12.5 秒となる．図 3.23 に LED 照明の可視光ビーコン波形及びパワースペクトラル密度を示す．図 3.23 の左図は可視光ビーコンの波形を示し，縦軸が明るさの強度であり，横軸が時間である．図 3.23 の右図は左図の波形に対するパワースペクトラル密度であり，この回帰直線を実線で示している．縦軸がパワースペクトラル密度の強度であり，横軸が周波数である．なお，回帰直線の傾きが-1 に近づくほど人間の目に対して快適な照明となり，点線で-1

の傾きの直線を示している．そして，図 3.23 (a)∼(f) は正弦波の振幅による違いを示している．正弦波の振幅が小さいほど，図 3.23 の右図における回帰直線がより-1 に近づくことが確認できた．すなわち，正弦波の振幅が小さいほど，人間の目に対して快適な可視光ビーコンになる．また，図 3.24 にスマートフォンで受信した画素値の変化とその画素値から抽出した周波数の結果を示す．このため，図 3.23 の左図で示した可視光ビーコンが，図 3.24 の右図で示したスマートフォン上での輝度値として検知される．縦軸は検知した画素値であり，横軸は時間である．そして，図 3.24 の左図に右図で検知した画素値に対して FFT 処理による周波数解の結果を示す．横軸が周波数であり，縦軸が周波数毎による輝度値の強度をを示す．すなわち，0.08 Hz の正弦波を組込んだ可視光ビーコンを検知しているため，0.08 Hz の周波数の強度が大きくなっていれば正弦波を検出したこととなる．なお，本実験では，LED 照明の機械的な制限に併せて，スマートフォンの画像取得速度を 1 fps とした．図 3.24 から，0.08 Hz において輝度値の強度が大きくなっていることから，0.08 Hz の正弦波を抽出できていることが確認できた．また，輝度値の最大値及び最小値が 10 以上の差がある場合，スマートフォン上で FFT 処理により可視光ビーコンが検出できることがわかった．さらに，LED 照明の照度設定が最大値及び最大値の差が 25 以上である場合に，スマートフォンで可視光ビーコンを検知できることも確認した．以上の結果から，1/f ゆらぎに 0.08 Hz の正弦波を組込んだ可視光ビーコンはスマートフォンの FFT 処理により検出可能であり，スマートフォンのカメラ機能も停止できることが確認できた．

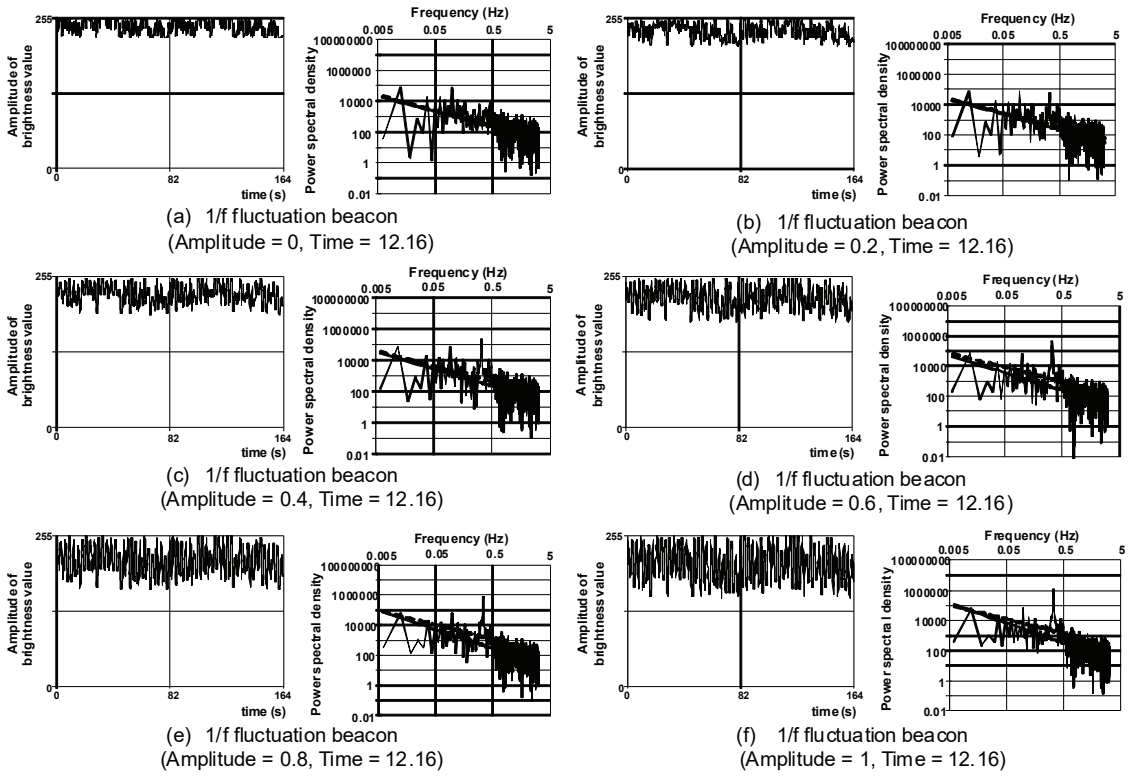


図 3.23: LED 照明の可視光ビーコン変化 (1/f ゆらぎと 0.08 Hz の正弦波を組合わせる)

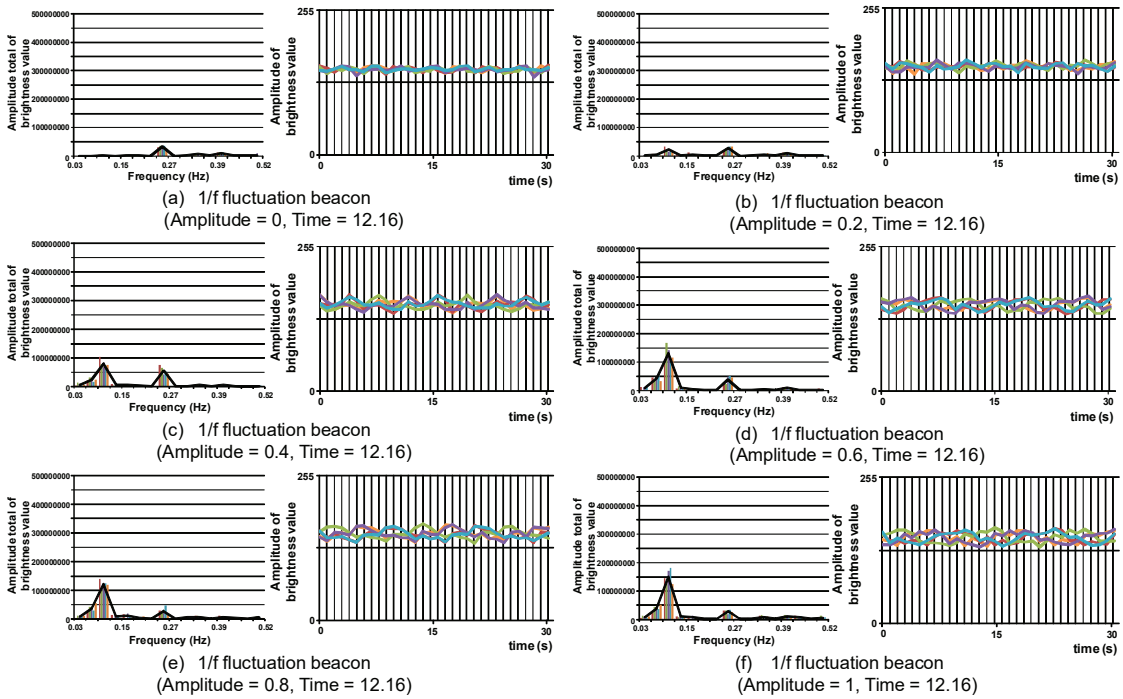


図 3.24: スマートフォンにおける可視光ビーコン受信時の画素値変化及び 0.08 Hz の正弦波を抽出

さらに，1/f ゆらぎに 0.16 Hz の正弦波を組込んだ可視光ビーコンについて検証する．本検証も 0.08 Hz の正弦波を組込んだ場合と同様に実施した．図 3.25 に図 3.23 と同様に，LED 照明の可視光ビーコン波形及びパワースペクトラル密度を示し，図 3.26 に図 3.24 と同様に，スマートフォンで受信した画素値の変化とその画素値から抽出した周波数の結果を示す．図 3.26 から，0.16 Hz において輝度値の強度が大きくなっていることから，0.16 Hz の正弦波を抽出できていることが確認できた．また，輝度値の最大値及び最小値が 10 以上の差がある場合，スマートフォン上で FFT 処理により可視光ビーコンが検出できることがわかった．さらに，LED 照明の照度設定が最大値及び最大値の差が 25 以上である場合に，スマートフォンで可視光ビーコンを検知できることも確認した．以上の結果から，1/f ゆらぎに 0.08 Hz の正弦波を組込んだ可視光ビーコンはスマートフォンの FFT 処理により検出可能であり，スマートフォンのカメラ機能も停止できることが確認できた．

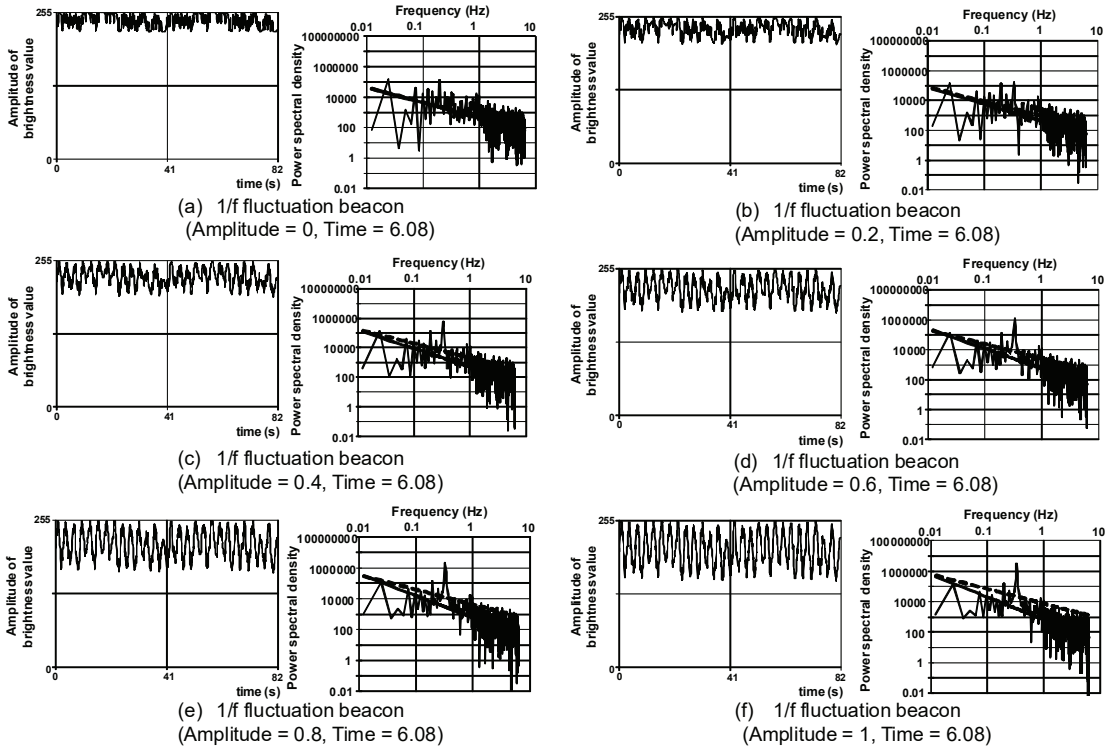


図 3.25: LED 照明の可視光ビーコン変化 (1/f ゆらぎと 0.16 Hz の正弦波を組合わせる)

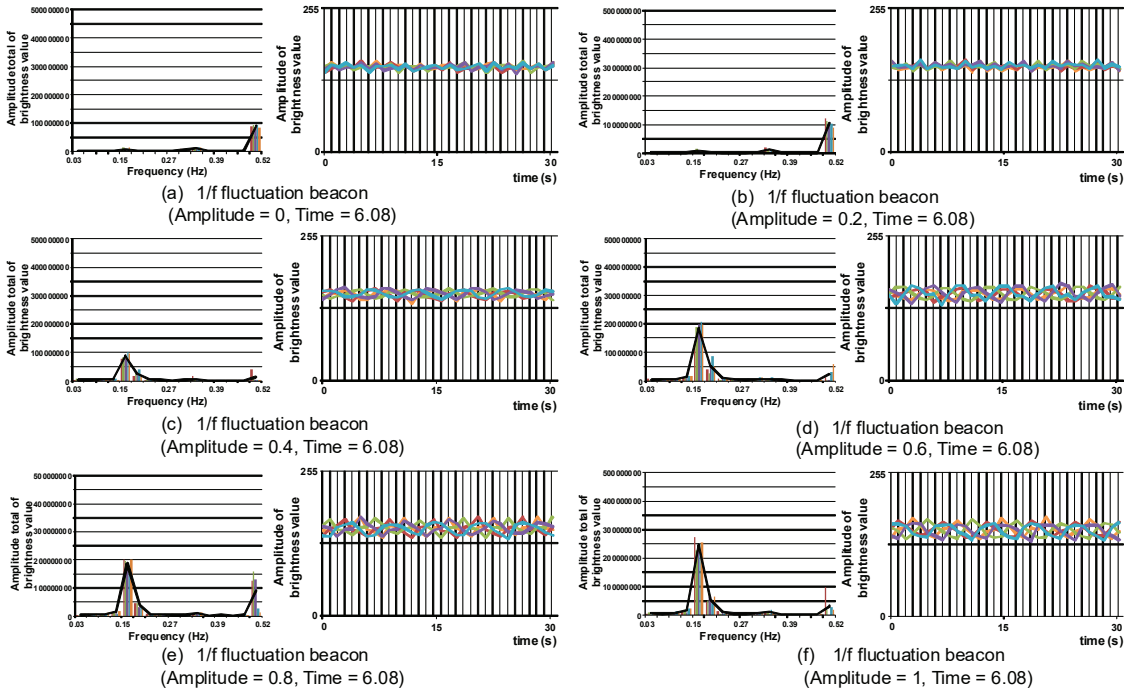


図 3.26: スマートフォンにおける可視光ビーコン受信時の画素値変化及び 0.16 Hz の正弦波を抽出

さらに，1/f ゆらぎに 0.32 Hz の正弦波を組込んだ可視光ビーコンについて検証する．本検証も 0.32 Hz の正弦波を組込んだ場合と同様に実施した．図 3.27 に図 3.23 と同様に，LED 照明の可視光ビーコン波形及びパワースペクトラル密度を示し，図 3.28 に図 3.24 と同様に，スマートフォンで受信した画素値の変化とその画素値から抽出した周波数の結果を示す．図 3.28 から，0.32 Hz において輝度値の強度が大きくなっていることから，0.32 Hz の正弦波を抽出できていることが確認できた．また，輝度値の最大値及び最小値が 10 以上の差がある場合，スマートフォン上で FFT 処理により可視光ビーコンが検出できることがわかった．さらに，LED 照明の照度設定が最大値及び最大値の差が 25 以上である場合に，スマートフォンで可視光ビーコンを検知できることも確認した．以上の結果から，1/f ゆらぎに 0.32 Hz の正弦波を組込んだ可視光ビーコンはスマートフォンの FFT 処理により検出可能であり，スマートフォンのカメラ機能も停止できることが確認できた．

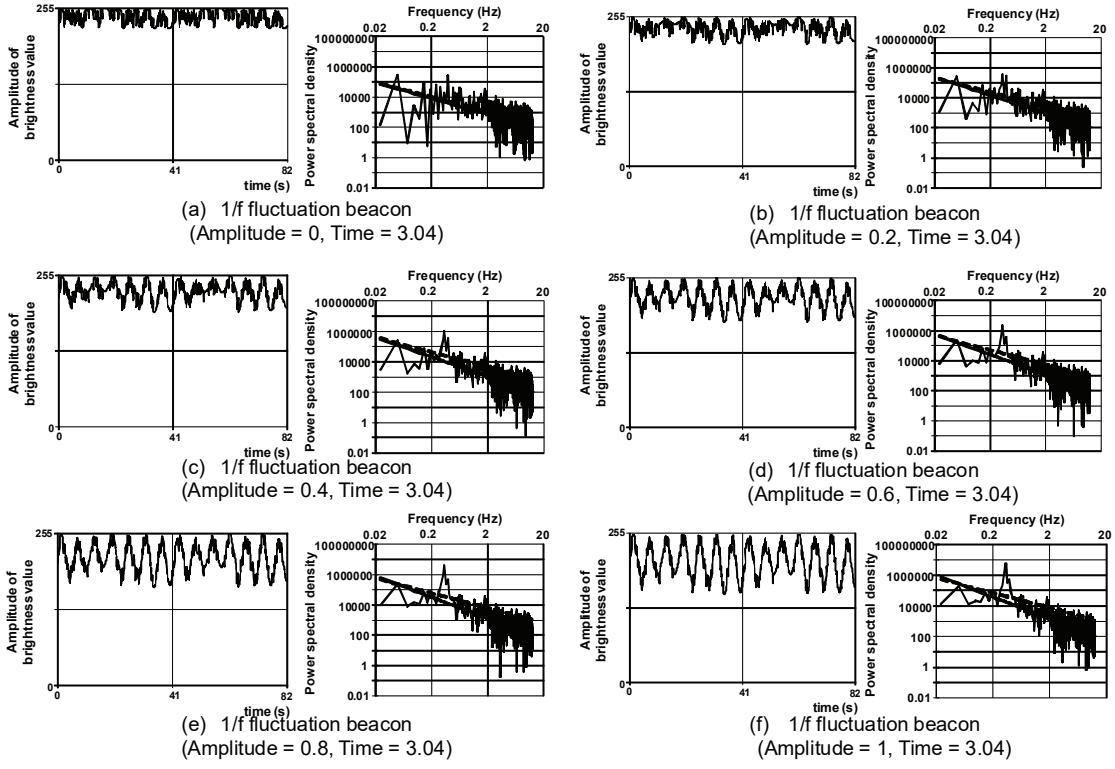


図 3.27: LED 照明の可視光ビーコン変化 (1/f ゆらぎと 0.32 Hz の正弦波を組合わせる)

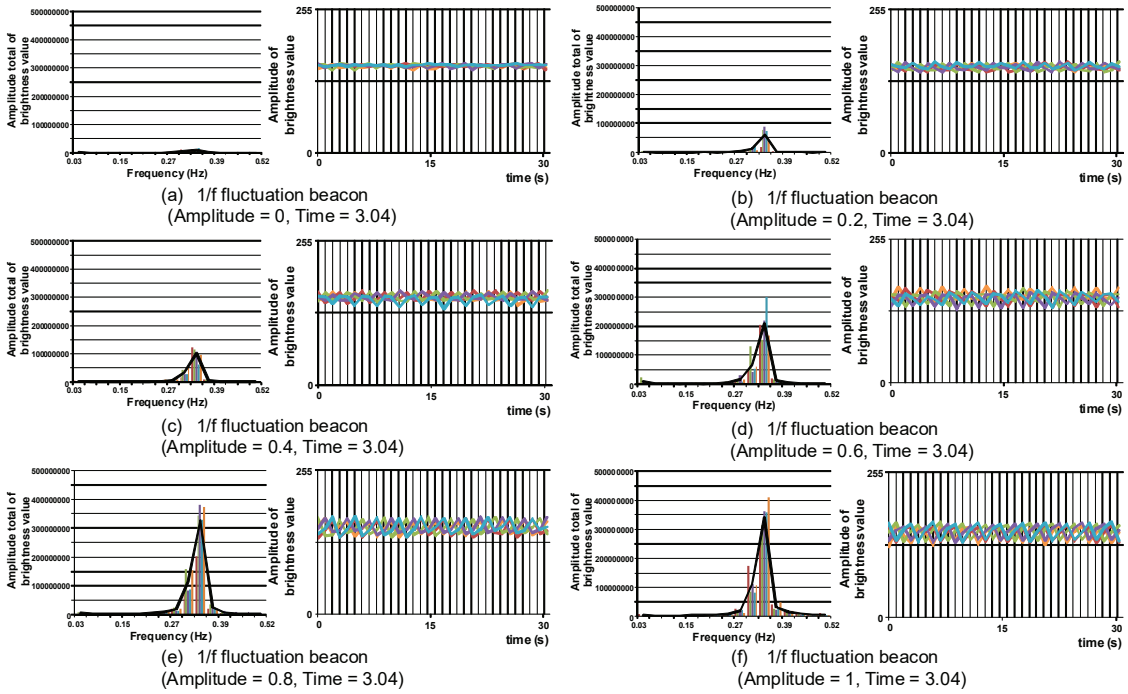


図 3.28: スマートフォンにおける可視光ビーコン受信時の画素値変化及び 0.32 Hz の正弦波を抽出

SPS と既存技術の比較

SPS の有効性について表 3.7 に示し，表 3.8 において SPS と 3.2.3 節において説明した既存技術について比較する．なお，今回行った SPS の検証では FFT 処理を実行可能なソフトウェアをスマートフォンにインストールしたが，将来的にハードウェアに FFT 処理可能な回路を組込むことを想定しているため，これも考慮して有効性の検証を行う．LED 照明は日常的に利用されているものであり，交換も容易である．SPS ではこの日常的に利用される LED 照明の照度を変更が可能な回路を組込むだけで利用できる．そのため，特別な大規模システムは必要としない．このため，赤外線によるノイズ付加技術よりも安価に盗撮防止空間の構築が可能であり，設置には LED 照明を取り換えるだけであるので，盗撮防止空間の構築は容易である．さらに，可視光ビーコンは人間の目に対して快適であることが知られている 1/f ゆらぎに正弦波を組込んであり，人間の目に対して快適なビーコンであることから日常の LED 照明としても利用可能である．また，SPS は可視光を利用しているため，重要文化財等に影響を与えることはない．このことから，SPS は撮影時のシャッター音，カメラモジュールに貼る盗撮防止シール及び赤外線によるノイズ付加技術よりも，日常生活でも利用しやすい．また，モバイルカメラ機器のカメラ機能を起動する場合，必ず可視光がカメラモジュールに入るため，LED 照明からの可視光ビーコンを受信してカメラ機能が停止することになる．すなわち，モバイルカメラ機器のカメラ機能に対して直接的に操作することが可能である．また，可視光ビーコンの周波数を変更することで，カメラ機能を停止したり，警告メッセージを表示したり，様々な処理を行うことが可能であるため，SPS の汎用性は撮影時のシャッター音及びカメラモジュールに貼る盗撮防止シールよりも高い．

表 3.7: SPS の有効性

Ease of construction	N/A	- Installing just the dimming devices - Just replacing LED lighting - Being same as the everyday used LED lighting
Introduction to daily lives	N/A	- Just replacing LED lighting - Being same as the everyday used LED lighting
Versatility for preventing spy-camera activities	✓	- Having a wide range of other uses depending on how the application is created - Controlling various function by changing the LED light beacon
Development cost	N/A	- Installing just the dimming devices - Just replacing LED lighting - Being same as the everyday used LED lighting
Direct control on camera function	✓	- Stopping the mobile camera functions by the LED light beacon - Being Embedded on camera devices as an FFT hardware application

表 3.8: SPS と既存技術における有効性比較

Existing measures and proposed system	Ease of construction	Introduction to daily lives	Versatility for preventing spy-camera activities	Development cost	Direct control on camera function
SPS	✓	✓	✓	✓	✓
Shutter-sound	✓	N/A	N/A	✓	N/A
Photography seal	✓	N/A	N/A	✓	✓
Infrared radiation technique	N/A	N/A	✓	N/A	✓

3.2.4 まとめ

近年，高画質な動画像を撮影可能なモバイルデバイスが急速に普及しており，併せてこのデバイスを利用した盗撮等の犯罪が急増している．このため，世界中で半導体技術の面から対策を講じるよう求められているところである．本節ではこの盗撮という犯罪に対する対策として盗撮防止システムである SPS を提案した．SPS は LED 照明とスマートフォンが連携して，特定の信号を検知した場合にスマートフォンのカメラ機能を強制的に停止させる手法である．LED 照明からは人間の目に不快感を与えない 1/f ゆらぎに正弦波を組合わせた信号を送信し，スマートフォンはこの信号を内蔵されているイメージセンサから受信し，イメージセンサの画素値に画像処理を行って信号に組込まれている正弦波の周波数を抽出する．スマートフォンはあらかじめ設定された特定の正弦波の周波数を検知した場合にカメラ機能を強制的に停止させる．SPS においては，LED 照明の照度変化がシステム設定値で 10 以上の差があると，スマートフォンでは画素値変化が 25 以上となり，この場合にスマートフォンは正弦波の周波数を検知することができた．さらに，既存の盗撮防止手法と SPS を比較したところ，日常生活での使いやすさやコスト面において SPS は優位性があることを確認した．一方，本節ではソフトウェアによる対策について議論したが，今後はハードウェアによる対策についても検討していく予定である．盗撮防止はリアルタイムに膨大なデータ量である動画に対して常に処理が必要となるため，ハードウェアにおいて処理することが高速処理につながる．現在のモバイルデバイスでは一般的にプロセッサに組込まれた DSP を用いて，イメージセンサのデータに対して画像処理を実行することが主流である．しかし，この DSP を既存のモバイルデバイス向けプロセッサよりも高性能であることが確認された CAMX に置き換えることで，より高画質なイメージセンサの膨大なデータ量においてもリアルタイムに処理可能であると考えられる．また，CAMX をイメージセンサから取得したデータを格納するメモリとしても利用することで，この格納されたデータに対して正弦波の検出処理を実行し，データの入出力処理を削減することでさらなる高性能化も考えられる．

第4章　CAMXの実装を想定したユーザサポートのための陸上変温動物向けバイオロギング用ノーマリオフデータロガーの開発及び検証

本章では，CAMX の将来的な実装を目的に，ユーザサポート可能なウェアラブルデバイスの低消費電力手法について検証する．ユーザサポートの一つとして，陸上変温動物の生体を調査でき，直接体に取り付けるバイオロギング用データロガーを構築し，ノーマリオフ手法による低消費電力なデバイスを提案する．

4.1　背景

近年，自然に生きる動物の体に温度センサ，イメージセンサ，GPS センサ等を取り付け，生態や行動のデータを取得するデータロガーの開発が盛んに行われている．この様な手法はバイオロギングと呼ばれ，人間の観察だけでは解明できない動物の生態を常時観察できるようになる．この研究分野が急速に発展した背景には，半導体の技術進歩が挙げられ，LSI の小型化やメモリの大容量化等の要因がある．バイオロギングは，1965 年にアメリカのジェラルド・クーイマンが南極において，アザラシにアナログ式記録計を取り付けたことが始まりと言われている [147], [148]．この時のデータロガーはサイズが大きいものであった．その後，1970 年代に世界各地でデータロガーの開発が盛んに行われ，1980 年代には日本においても開発が始まった [149], [150]．1990 年代には，日本では世界に先駆け，デジタル式データロガーの開発が始まった．これをきっかけに，データロガーの小型化及び多用化が飛躍的に発展し，様々な研究が行われる様になっている [147]–[149], [151]–[157]．データロガーを動物に取り付けることで，生態を知ることだけではなく，地球規

模の問題解決にもつながることがわかってきている．このため，鳥類，魚類等の地球規模で広範囲に行動する動物が観察対象になることが多い．この様な動物は広範囲に行動するため，活動量が多いことから，加速度センサ，GPS センサ等を内蔵したデータロガーを用いることが多い [158], [159]．一方，身近な動物は地球規模に活動する動物よりも活動量が少なく，多様な環境で行動する陸上変温動物多く存在する．現在，これらの陸上変温動物（カメ，トカゲ等）は外来種が生息範囲を拡大するとともに，在来種の個体数が大幅に減少している．例えば，陸上変温動物の一種であり，日本国内の河川，水田等に生息しているニホンイシガメは，行動範囲は局所的であり，ミシシッピアカミミガメ，アライグマ等の外敵に襲われることが多い [160], [161]．そのため，現在ではニホンイシガメは準絶滅危惧種に指定されている [162]．さらに，ニホンイシガメと外来種が交雑することで，新たな種が誕生していることも問題となっている [163], [164].

本論文では，身近に生息している陸上変温動物をターゲットとしたバイオロギング用データロガーの開発について示す．開発するデータロガーは，活動量が少ない動物の動作と連携した各種センサの間歇動作による低消費電力処理を実現する．そして，データロガーを取り付ける動物は，日本の淡水域とその周辺に生息する唯一の固有種であるニホンイシガメとする．なお，ニホンイシガメの生態を解明するために，取り付けるデータロガーに関する研究は，我々が調査した限りでは確認できていない．そのため，本論文では，バイオロギン用データロガーに組込まれているセンサをノーマリオフ動作させた場合の消費電力効果や検知制度を明確にすることを目指す.

4.2　低消費電力を目的とした既存のバイオロギング技術

本論文では低消費電力なバイオロギング用データロガーの開発を目指し，ノーマリオフコンピューティングに着目している [165]–[168].　そして，動物の活動に合わせて各センサを制御することで間歇動作を実現し，データロガーの駆動時間を 1 日以上となるよう開発する．まずは，開発にあたって，バイオロギングに関する低消費電力技術をまとめる.

4.2.1　太陽光発電とタイマー動作を併用したロギング処理

太陽光発電（ソーラパネル）によりデータロガーのバッテリを充電する技術がある [169]．ソーラパネルは太陽光が入射する限り半永久的に発電することが可能であり，データロガーも半永久的に駆動できる．この技術は地面に固定して観察

することを想定しており，動物がカメラに写った時に動画像の撮影を行う．動物の検知では，あらかじめ指定した時間毎に電源が入り，周囲の状況の検知し，カメラを駆動させるか選択することになる．この様に，太陽光発電と間歇動作を利用することで，データロガーの低消費電力を実現している．しかし，太陽光発電は発電に必要な光量を得るために，地上に設置する必要があり，データロガーのサイズが大きくなるという問題がある．

4.2.2 動物の動作に合わせた振動発電

動物が動く時には振動が発生するため，この振動を発電に利用し，この電力をデータロガーに使用することで低消費電化を実現する技術がある [170], [171]．この技術では，魚のブリにデータロガーを取り付けて，ブリの泳ぎにより発生する振動を電力に変換している．振動による発電は，動物のはばたき，尾鰭振動等，体の振動を利用するため，動物が生きている限り発電が可能である．しかし，振動発電で得られる電力は小さく，データロガーを安定的に駆動できない問題がある．活動量が少ない陸上変温動物においては特に発電量が小さくなる．

4.2.3 加速度センサによる動作検知

加速度センサにより動物の動作を検知し，検知時にデータロガーを駆動させる間歇動作を利用した低消費電力なデータロガーがある [172], [173]．このデータロガーは，素早い動きをする魚のマダイ等に取り付け，加速度センサの高加速度を検出する．加速度センサを用いることで，動物の俊敏な動作を検知でき，尾鰭を振った回数や急加速の理由を解析できる．しかし，活動量が多い動物が対象となり，検知可能な動作が多くない小型の動物には利用できない．

4.3 陸上変温動物に取り付けるノーマリオフデータロガーの開発

本論文では，活動範囲が局所的で，俊敏でない低加速度な動作である陸上変温動物のバイオロギングを目的とする [174]–[178]．開発するデータロガーはノーマリオフコンピューティングによる間歇動作を利用した低消費電力の実現であり，1日以上の観察を実現する．なお，間歇動作とは動物の動作に合わせて，観察の必要性をセンサにより判断し，この判断結果からソフトウェアによりその他センサ

類の起動又は停止を行うことである．

活動量が少なく，局所的な地域に生息する陸上変温動物の研究は種の保全のためにも必要である [179], [180]．このため，陸上変温動物をバイオロギングの対象とする．陸上変温動物は水温が低い中で活動した際，体温の調節を目的に陸上でバスキング等を行うことが多い [181]．バスキング中に動作することはほとんどないため，低消費電力なデータロガーの実現に向けてノーマリオフコンピューティングのコンセプトを取り入れ，データロガーに組込んだ加速度センサをトリガとして間歇動作を行う [165]–[168]．すなわち，加速度センサが動物の動作を検知していない間は，マイコン上のソフトウェアでイメージセンサ，GPS センサ等の各種センサを停止させる．なお，実験では陸上変温動物としてニホンイシガメ（メス，背甲長約 23 cm，背甲幅約 16 cm）に開発したデータロガーを取り付ける．

4.3.1 ノーマリオフデータロガーの構成

ノーマリオフデータロガーの検証には，様々なセンサの試験を可能とするため RaspberryPi 3 Model B [182] をベースとしたプロトタイプを構築する．RaspberryPi 3 を利用することで，データロガーのサイズは大きくなるが，複数のセンサを組込んで陸上変温動物の動作を観察することは，今後の効果的なノーマリオフバイオロギング用データロガーの開発において基礎となるデータを収集可能となる．そのため，データロガーを取り付けやすい比較的大きな外骨格を持ち，活動に支障がないニホンイシガメをバイオロギングの観察対象とした．本論文では，ニホンイシガメの生態を観察するために必要なセンサとして，イメージセンサ（Camera Module V1.2 [183]），小型高感度 GPS センサ（GYSFDMAXD [184]），そして 2 種類の加速度センサ（Sense HAT Version 1.0 [185] に組込まれた LSM9DS1 [186], [187] 及び単一センサである GY-291 ADXL345 [188], [189]）を利用する．以下，LSM9DS1 を加速度センサ（LSM9DS1）とし，GY-291 ADXL345 を加速度センサ（GY-291）とする．Sense HAT Version 1.0 は加速度センサ，磁気センサ，ジャイロセンサ，温度センサ，湿度センサ，気圧センサ，LED 及びジョイスティックが組込まれており，本論文では加速度センサのみを使用する．GY-291 ADXL345 は x 軸，y 軸及び z 軸の 3 方向加速度を別々に計測できる．4.4.4 節では防水センサ（DS18B20 [190]）も取り付ける．そして，モバイルバッテリ（700-BTL018BK [191]）を接続することでデータロガーが駆動する．上述のセンサ類を搭載することで，データロガーは以下の機能を備える．

- ニホンイシガメの捕食や周辺の環境等を対象動物の視界で確認することが可能なイメージセンサ

- 屋外におけるニホンイシガメの行動範囲を地図上で俯瞰的に確認できる GPS センサ
- 対象動物の動作を常に検知し，イメージセンサ等を効率よくノーマリオフ動作させるためのトリガとなる加速度センサ
- イメージセンサや GPS センサで得られる情報に追加して，水温の変化を取得して水温による対象動物の動作への影響を確認できる防水温度センサ
- バッテリの残量を確認できる機能が付いており，表示された放電容量から消費電力を確認できるバッテリ

本論文では，活動量が少ない陸上変温動物に取り付けるデータロガーを開発し，その陸上変温動物の視界を確認できるよう，イメージセンサのデータ取得を優先する．ニホンイシガメの 1 年間における生態は知られているが [180]，近年問題となっている外敵や交雑の影響 [192], [193] がどのような状況で発生しているか等の調査は必要である．このため，ニホンイシガメの 1 日の行動を詳細に観察できるデータロガーが求められている．よって，開発するデータロガーの要求性能は，ニホンイシガメの活動に支障を与えることなく，体に直接取り付けることができるデバイスを想定し，1 日以上の観察が可能となるようにする．1 日以上の観察ができるよう，データロガーに組込む加速度センサは常にニホンイシガメの動作を検知し続け，検知した場合には観察すべきタイミングとして他の各種センサを駆動させる．これにより，他の各種センサはノーマリオフ動作となる．本論文では，バイオロギング用データロガーの開発に向けて，消費電力が最も大きいイメージセンサに対するノーマリオフ動作による低消費電力化及び加速度センサの動作検知精度の向上を目的とする．

ノーマリオフバイオロギング用データロガーの諸元を表 4.1 に示す．これらのセンサ等を搭載したデータロガー（図 4.1）をニホンイシガメに取り付ける．図 4.1 (a) はイメージセンサを一定間隔で間歇動作させる 4.4.1 節，GPS センサとイメージセンサを一定間隔で間歇動作させる 4.4.2 節及び加速度センサをトリガとしてイメージセンサを間歇動作させる 4.4.3 節の実験で用いたデータロガーであり，図 4.1 (b) は RaspberryPi 3 に接続した各センサの制御関係である．図 4.1 (a) のデータロガーには加速度センサ（LSM9DS1）を組込んでいる．図 4.1 (c) は加速度センサの動作検知精度を検証する 4.4.4 節の実験で用いたデータロガーであり，図 4.1 (d) は RaspberryPi 3 に接続した各センサの制御関係である．図 4.1 (c) のデータロガーには加速度センサ（GY-291）を組込んでおり，防水温度センサも追加している．このデータロガーニホンイシガメに取り付けイメージを図 4.2 に示す．データロガーは防水性のある箱に収納し，ニホンイシガメの甲羅上に取り付ける．こ

の状態で，図 4.1 (a) ではデータロガーの重量は 421 g であり，図 4.1 (c) ではデータロガーの重量は 496 g である．一般的に，小さい個体であってもニホンイシガメの体重は約 80 g であり [194], [195]，カメが甲羅に重しを載せて動作できる重量は体重の約 20 倍にもなるという調査もある [196]．すなわち，ニホンイシガメの体重をを 80 g と仮定すると，約 1.5 kg のデータロガーを甲羅に載せることができると考えられ，本論文で用いるデータロガーは約 500 g であるのでニホンイシガメの動作に支障ないと考えられる．このデータロガーはイメージセンサを利用することで，ニホンイシガメの目線で動画を撮影することができ，撮影した動画の一部を図 4.3 に示す．人間の目線とは全く異なる動画となっており，対象動物であるニホンイシガメの生態をより詳しく観察することができ，イメージセンサは生態を知る上で非常に有効な手段であることが確認できた．

表 4.1: ノーマリオフバイオロギング用データロガーの諸元

使用機器	型番	仕様
本体	RaspberryPi 3 Model B	動作周波数：1.2 GHz， 動作電源：5 V， メモリ：1 GB， 重量：42 g
イメージセンサ	Camera Module V1.2	画素数：2,592 × 1,944， 最大キャプチャフレームレート：30 fps， 重量：3 g
GPS センサ	GYSFD MAXD	受信感度：-164dBm(typ.)， 重量：8 g
加速度センサ（LSM9DS1）	LSM9DS1（Sense HAT Version 1.0 に搭載）	精度：±2/4/8/16 g， 重量：20.4 g
加速度センサ（GY-291）	GY-291 ADXL345	精度：±2/4/8/16 g 重量：0.03 g
防水温度センサ	DS18B20	測定範囲：-55〜+125 ℃ (0.5 ℃精度)， 重量：22.7 g
モバイルバッテリ	700-BTL 018BK	電圧 3.7 V， 容量：10,400 mAh， 出力 2.4 A， 重量：250 g

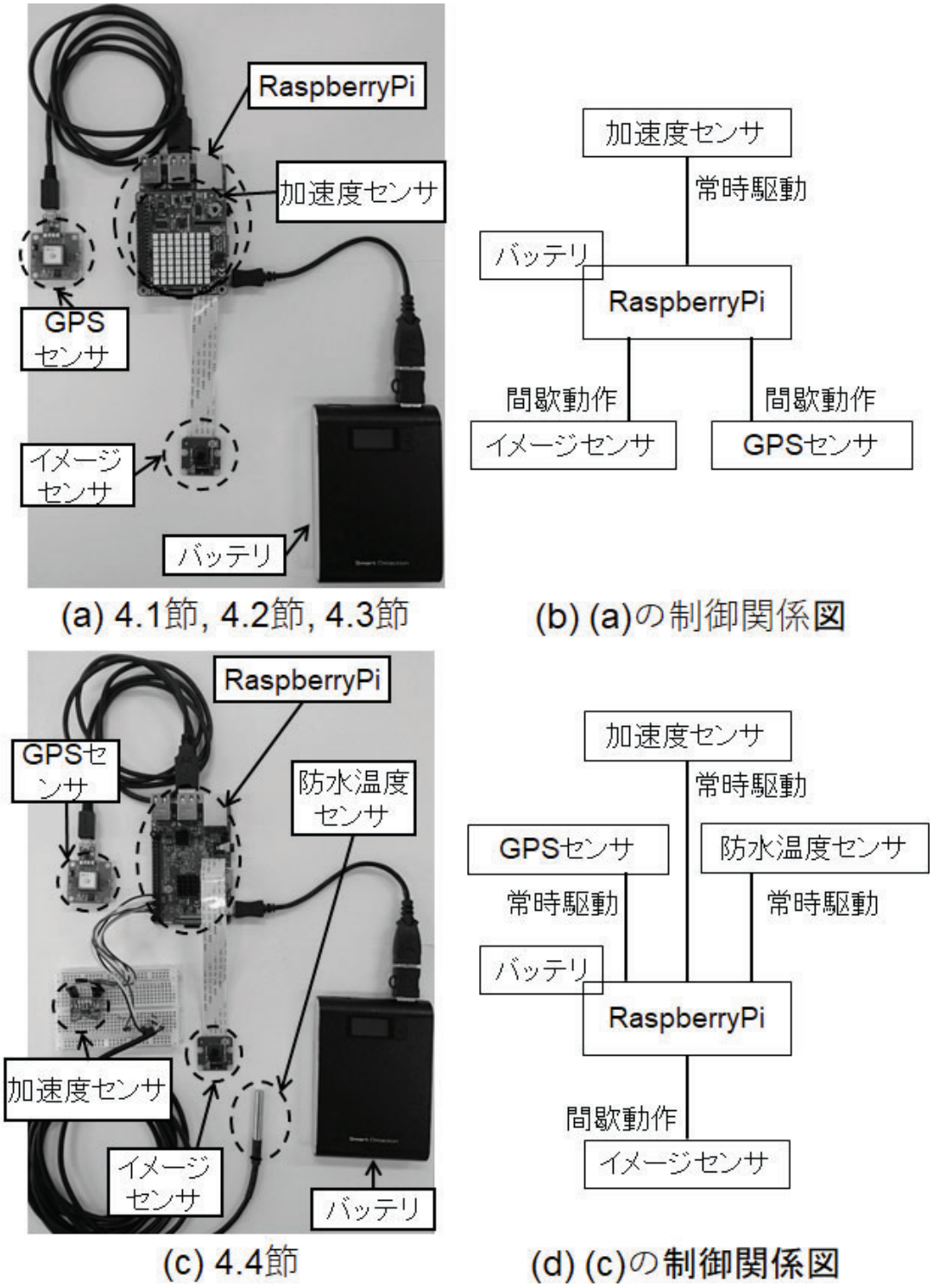


図 4.1: ノーマリオフバイロギング用データロガー.

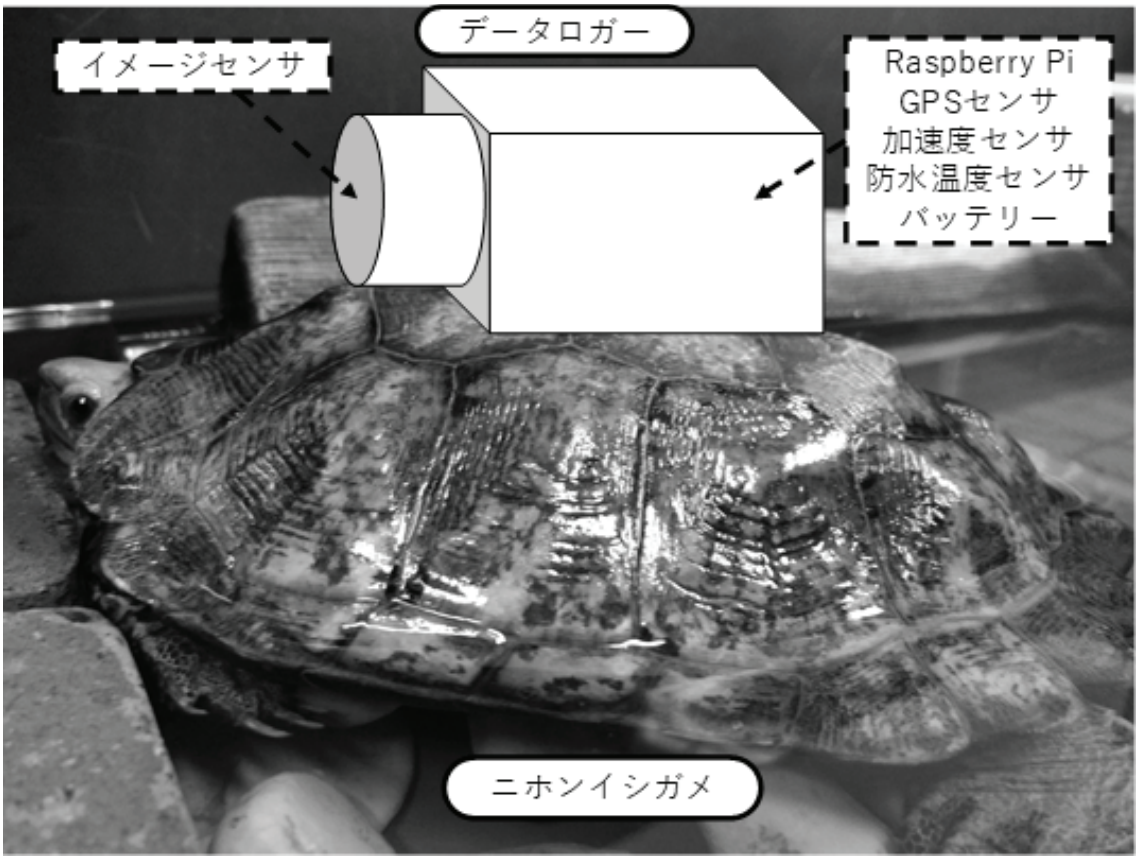


図 4.2: ニホンイシガメにデータロガーを取り付けた状態（イメージ）.

図 4.3: ニホンイシガメの視界を撮影した画像.

4.4　ノーマリオフバイオロギング用データロガーの検証

提案するノーマリオフバイオロギング用データロガーは活動量が少ない陸上変温動物に取り付けることを想定しており，陸上変温動物の動作に合わせてデータロガーを間歇動作させることにより，データロガーの長時間駆動が可能となる．間歇動作を行うためのトリガには加速度センサを使用し，このトリガにより他のセンサを間歇に動作できることを確認するとともに，消費電力の削減についても検証する．なお，活動量が少ない陸上変温動物であっても，観察とは直接関係のない軽微な動作も行っているため，加速度センサの検知タイミングとニホンイシガメの動作が一致していることについても検証する．

ここで，各節において駆動させるセンサの対応表を表 4.2 に示す．4.4.1 節では，イメージセンサにノーマリオフ動作をさせるため，イメージセンサを一定間隔で起動・停止を繰り返すし，他のセンサは常時駆動とする．4.4.2 節では，イメージセンサと GPS センサにノーマリオフ動作をさせるため，それぞれ一定間隔で起動・停止を繰り返し，他のセンサは常時駆動とする．4.4.3 節では，加速度センサ（LSM9DS1）で常時ニホンイシガメの動作を検知し，動作を検知した場合にイメージセンサを 2 分間駆動させ，他のセンサは常時駆動とする．4.4.4 節では，加速度センサ（GY-291）で常時ニホンイシガメの動作を検知し，動作を検知した場合にイメージセンサを起動し，他のセンサは常時駆動とする．また，消費電力の削減率は，バッテリの放電容量から式（4.1）により算出する．各節において，この削減率からノーマリオフ動作させたデータロガーを比較し，削減率が大きくなるほどより低消費電力なデータロガーとなることを確認する．

$$\text{削減率}\,[\%] = \left(1 - \frac{\text{ノーマリオフ消費電力}\,[Wh]}{\text{ノーマリオン消費電力}\,[Wh]}\right) \times 100 \tag{4.1}$$

なお，後述する 4.4.1 節の①，4.4.2 節の⑤，4.4.3 節の⑨はノーマリオン処理であり，後述する 4.4.1 節の② ～ ④，4.4.2 節の⑥ ～ ⑧，4.4.3 節の⑩はノーマリオフ処理である．

表 4.2: 各節において使用する機器

	4.4.1 節	4.4.2 節	4.4.3 節	4.4.4 節
本体	✓	✓	✓	✓
イメージセンサ	✓	✓	✓	✓
GPS センサ	✓	✓	✓	✓
加速度センサ（LSM9DS1）	✓	✓	✓	N/A
加速度センサ（GY-291）	N/A	N/A	N/A	✓
防水温度センサ	N/A	N/A	N/A	✓
バッテリ	✓	✓	✓	✓
ノーマリオフ動作させたセンサ	イメージセンサ	イメージセンサ or GPS センサ	イメージセンサ	イメージセンサ
ノーマリオフ動作のトリガとしたセンサ	N/A	N/A	加速度センサ（LSM9DS1）	加速度センサ（GY-291）

4.4.1 イメージセンサの消費電力とノーマリオフ動作の関係

本実験では，消費電力が大きいイメージセンサをノーマリオフ動作させることで消費電力の削減効果を検証するため，実験条件を以下とする．

① 60 分間動画を撮影し続け，60 分後の消費電力を算出する（撮影時間は 60 分である）．

② 10 分毎にイメージセンサの起動・停止を繰り返し，60 分後の消費電力を算出する（60 分間に合計 3 回の撮影を行う．各撮影時間は 10 分である）．

③ 20 分毎にイメージセンサの起動・停止を繰り返し，60 分後の消費電力を算出する（60 分間に合計 2 回の撮影を行う．各撮影時間は 20 分である）．

④ 30 分毎にイメージセンサの起動・停止を繰り返し，60 分後の消費電力を算出する（60 分間に 1 回の撮影を行う．撮影時間は 30 分である）．

各条件における消費電力及び削減率を図 4.4 に示す．それぞれの条件に対して 3 回の実験を行い，平均値を算出した．図 4.4 では，横軸のそれぞれの条件における平均消費電力と消費電力の削減率を縦軸に示す．なお，平均消費電力は棒グラフ，消費電力の削減率は折れ線グラフで表し，削減率については①に対する②, ③及び④の平均削減率となる．

間歇動作の回数が最も少ない④と比較し，②はイメージセンサの起動・停止の繰り返し回数が 2 回多いにもかかわらず，平均消費電力は約 0.5 Wh 大きくなる．同様に，③は繰り返し回数が 1 回多いが，平均消費電力は約 0.2 Wh 大きくなる．以上の結果から，消費電力削減のため，頻繁にイメージセンサの起動・停止を繰り返しても，起動・停止の繰り返し回数当たりの消費電力は約 0.2 Wh 増加することになることが確認できた．また，実験は AVI 形式で動画を撮影し，データ容量は次の通りであった．①は 179.0 MB で，②は 1 回目の撮影が 23.2 MB，2 回目の撮影が 26.2 MB，3 回目の撮影が 24.8 MB であった．また，③は 1 回目の撮影が 52.2 MB，2 回目の撮影が 62.0 MB で，④は 102.7 MB であった．データ容量は圧縮率に左右されるところではあるが，①のように常に撮影し続ける場合より，間歇動作をさせた方がデータ容量は小さくなることが確認できた．よって，間歇動作によるイメージセンサの消費電力削減には，頻繁にイメージセンサの起動・停止を繰り返すことなく，観察が必要なタイミングにおいて撮影を繰り返すことが有効である．ニホンイシガメのような活動量が少ない動物については，頻繁に動画を撮影し続ける必要がないため，活動が少ない時に，イメージセンサを停止させることで消費電力をより効果的に削減できると考えられる．

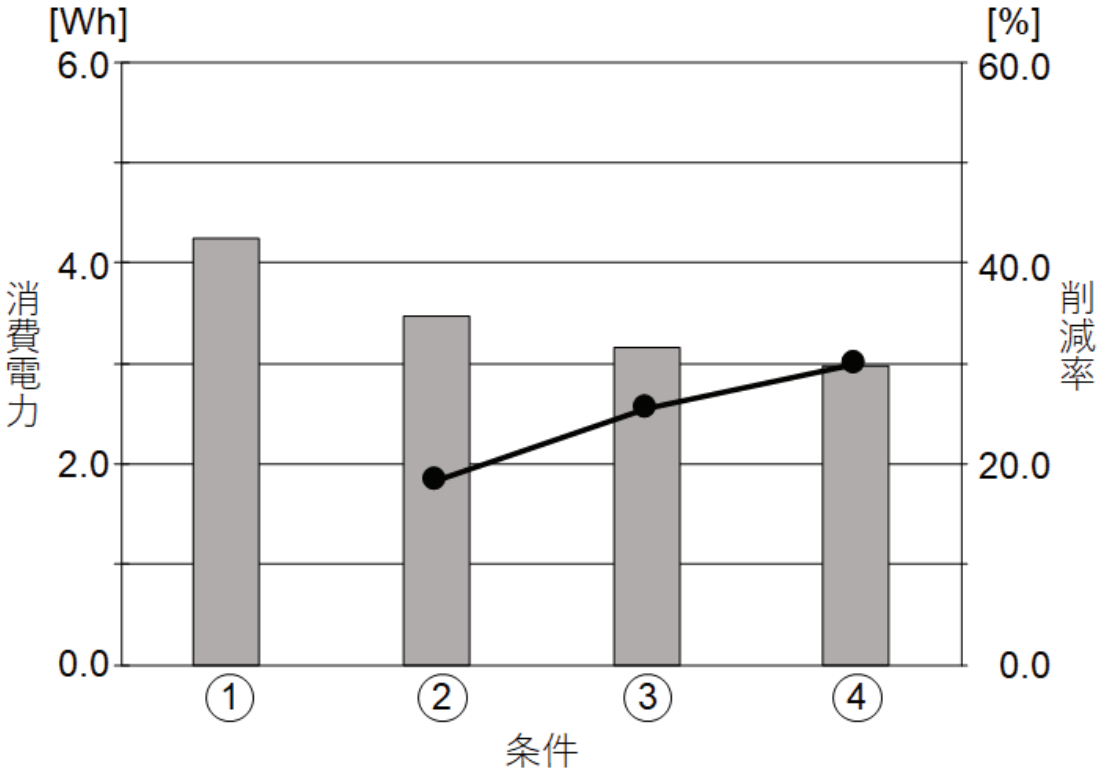


図 4.4: イメージセンサの消費電力及びノーマリオフによる削減率.

4.4.2 GPS センサ及びイメージセンサを用いたノーマリオフ動作

本節では，センサ単体，若しくはセンサの組合わせによる間歇動作が消費電力の削減に与える影響について検証する．実験条件は無間歇動作，GPS センサの間歇動作，イメージセンサの間歇動作及び GPS センサとイメージセンサを同時に間歇動作させた場合の消費電力を計測する．図 4.5 に各条件の時間経過を示す．条件 ⑤は無間歇動作であり，RaspberryPi 3 上で間歇動作は行わず，常時全てのセンサを駆動させて 90 分後の消費電力を計測する．条件 ⑥は GPS センサの間歇動作であり，プログラムを開始し，5 分後に GPS センサを起動してデータ取得を行う．そのデータを保存した後に，GPS センサは停止する．さらに，5 分後に GPS センサを起動して，データの取得・保存を行う．この一連の処理を繰り返し実行し，90 分後の消費電力を計測する．なお，処理中にはイメージセンサ及び加速度センサ（LSM9DS1）は常時駆動となる．条件 ⑦はイメージセンサの間歇動作であり，プログラムを開始し，5 分後にイメージセンサを起動してデータ取得を行う．そのデータを保存した後に，イメージセンサは停止する．さらに，5 分後にイメージセンサを起動して，データの取得・保存を行う．この一連の処理を繰り返し実行し，90 分後の消費電力を計測する．なお，処理中には GPS センサ及び加速度センサ（LSM9DS1）は常時駆動となる．条件 ⑧は GPS センサとイメージセンサの同時間歇動作であり，プログラムを開始し，5 分後に GPS センサ及びイメージセンサを起動してデータ取得を行う．それぞれのデータを保存した後に，GPS センサ及びイメージセンサは停止する．さらに，5 分後に GPS センサ及びイメージセンサを起動して，データの取得・保存を行う．この一連の処理を繰り返し実行し，90 分後の消費電力を計測する．

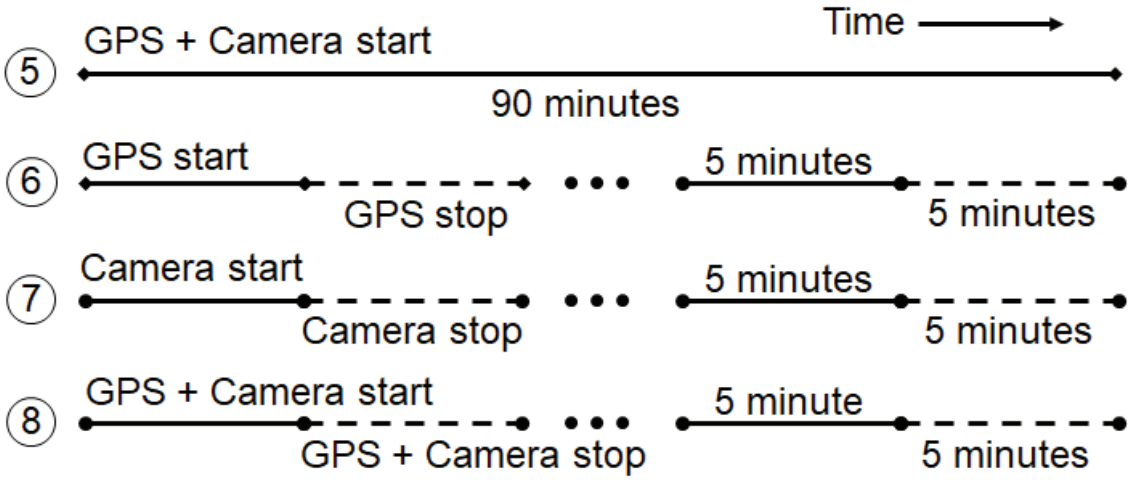


図 4.5: GPS センサ及びイメージセンサを用いたノーマリオフ処理の時間経過.

実験は 4 m × 5 m で区切ったスペースで行い，データロガーを取り付けたニホンイシガメを自由に行動させる．上述の 4 条件をそれぞれ 3 回実施し，モバイルバッテリの消費電力を計測する．ただし，各条件において，実験開始時にはモバイルバッテリを完全に充電する．各条件の消費電力及び削減率の結果を図 4.6 に示す．図 4.6 の各軸は図 4.4 と同様である．なお，消費電力の削減率については，⑤に対する⑥，⑦及び⑧の消費電力の平均削減率を示す．無間歇動作である⑤は，平均消費電力が 3.7 Wh であり，間歇動作を行った⑥ ～ ⑧の条件に比べると消費電力が最も大きくなった．⑥ ～ ⑧の平均消費電力はそれぞれ 3.2 Wh，2.9 Wh，2.8 Wh である．⑤と比較して，平均削減率は約 15 %，22 %，24 %となる．各条件における 3 回の実験は，それぞれの消費電力に多少のばらつきが見られるものの，2 つのセンサを同時に間歇動作させることでデータロガーの長時間駆動に有効であることが確認できた．また，イメージセンサのみを間歇動作させている⑦は，GPS センサのみを間歇動作させている⑥より約 10 %の消費電力が削減できた．すなわち，イメージセンサは消費電力が大きいため，消費電力削減に向けて停止の優先順位が高いと考えられる．ここで，算出した消費電力を基に，一般的な単 3 形電池である Panasonic　エネループ　スタンダードモデル（ニッケル水素電池，電圧 1.2 V，容量 1,900 mAh，電力量 2.3 Wh，重量 13 g） [197]，電子工作用リチウムイオンポリマー電池 (電圧 3.7 V，容量 2,000 mAh，電力量 7.4 Wh，重量 35 g) [198]，モバイルバッテリ用 Anker PowerCore Essential 20000(電圧 3.7 V，容量 20,000 mAh，電力量 100 Wh，重量 343 g) [199] をバッテリとした時の駆

動可能時間を式 (4.2) により見積もった．また，本実験で利用したモバイルバッテリ用 700-BTL018BK（電圧 3.7 V，容量 10,400 mAh，電力量 38.5 Wh，重量 250 g）の駆動可能時間も見積もった．表 4.3 に駆動時間を示す．なお，これらの電池は容易に入手可能という観点から選択した．また，それぞれのバッテリを本実験で使用したデータロガーに実装したと仮定すると，データロガーの重量は 1.5 kg を超えないためニホンイシガメの甲羅に載せたとしてもニホンイシガメの動作に支障ないと考えられる．表 4.3 からバッテリの重量は異なるものの容量が大きいとデータロガーの駆動時間は長くなり，⑤と比較して⑥ ∼ ⑧それぞれは，順に約 17 %，29 %，32 %の割合で駆動時間が長くなっている．この結果より，GPS センサ及びイメージセンサを一定間隔でノーマリオフ動作させることで，常時センサを駆動するよりもデータロガーを長時間駆動できることが確認できた．ニホンイシガメは，日中に甲羅干しをしており，夜間は水中で睡眠 [179], [180] しているため，データロガーにノーマリオフ技術を導入することで一般的な電池と RaspberryPi 3 の組合わせとしても約 1 日の観察が可能となる．

$$
\begin{aligned}
\text{駆動時間}\,[h] \quad &= \frac{\text{電池の容量}\,[Wh]}{\text{データロガーの消費電力}\,[Wh]} \\
&= \frac{\text{電池の電圧}\,[V] \times \text{電池の容量}\,[Ah]}{\text{データロガーの消費電力}\,[Wh]}
\end{aligned}
\tag{4.2}
$$

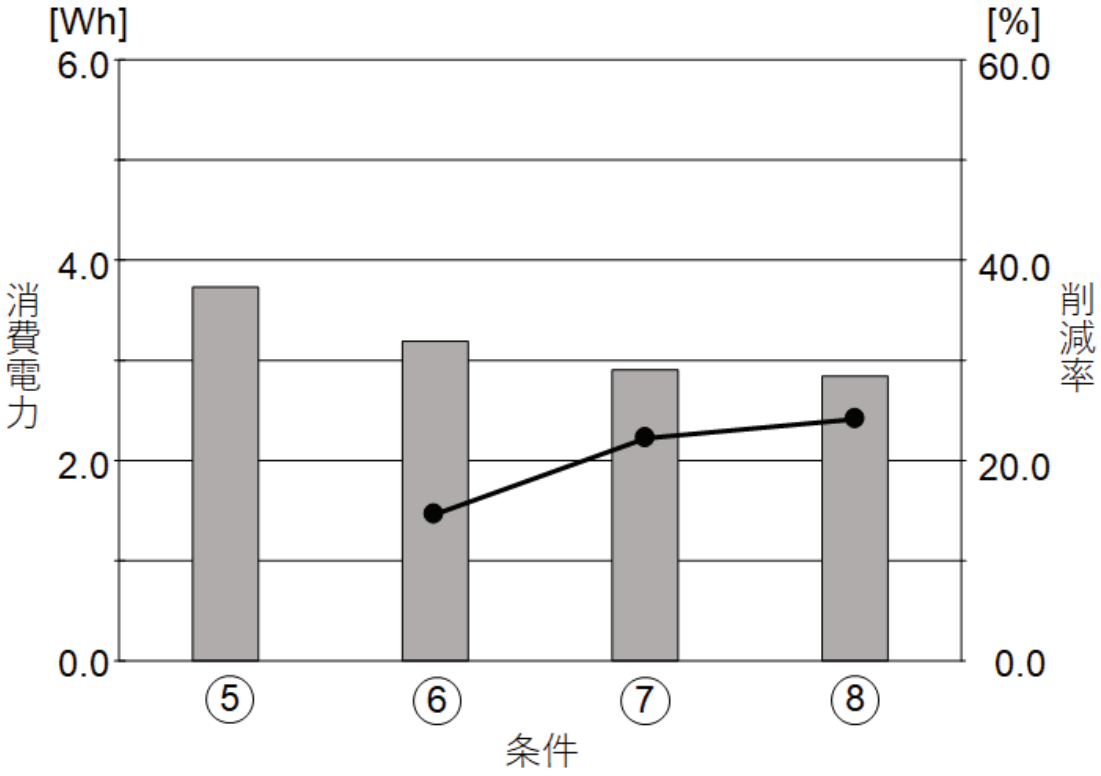


図 4.6: GPS センサ及びイメージセンサのノーマリオフ処理による消費電力と削減率.

表 4.3: GPS センサ及びイメージセンサを用いたノーマリオフ処理におけるデータロガー駆動時間の算出

	駆動時間 [h]			
	Panasonic エネループ スタンダードモデル	リチウムイオンポリマー電池	Anker PowerCore Essential 20000	700-BTL 018BK
⑤	0.6	2.0	19.8	10.3
⑥	0.7	2.3	23.2	12.1
⑦	0.8	2.5	25.4	13.2
⑧	0.8	2.6	26.1	13.6

4.4.3 加速度センサを用いた動作の検知によるノーマリオフ動作

本節では，活動量が少なく，加速度が小さい陸上変温動物の動作を検知することでノーマリオフ動作を行う．各条件の時間経過を図 4.7 に示す．条件 ⑨はイメージセンサで 2 分間の動画撮影を 60 分繰り返し続ける．条件 ⑩は加速度センサによりニホンイシガメの動作を検知し，検知するたびにイメージセンサで 2 分間の動画撮影を行う．これらの条件に付いて比較する．条件 ⑩では，プログラムを開始すると加速度センサでニホンイシガメの動作を常に検知し続け，あらかじめ定めた閾値を超えた動作が発生した場合にイメージセンサを起動して 2 分間の動画撮影を行い，保存する．その後，加速度センサにより再度動作を検知し続ける．この一連の処理を 60 分間繰り返し実行する．なお，4.4.2 節の実験結果より，イメージセンサのノーマリオフ動作による消費電力削減について検証するため，GPS センサは常時駆動とする．また，ニホンイシガメの生態から，一度の動作に費やす時間は数分であるため，動画の撮影時間は 2 分間とした．加速度センサ（LSM9DS1）の検知手法については，起動時に非検知状態の加速度を 100 回計測し，その単純平均値からその場所の重力を算出する．そして，この重力の値と加速度センサ（LSM9DS1）によって検知した動作の加速度との差分があらかじめ定めた閾値以上であればニホンイシガメが動作したとして，イメージセンサを起動する．本実験では，あらかじめ定めた閾値は，複数回の実験から最適であると判断した 0.01 とした．

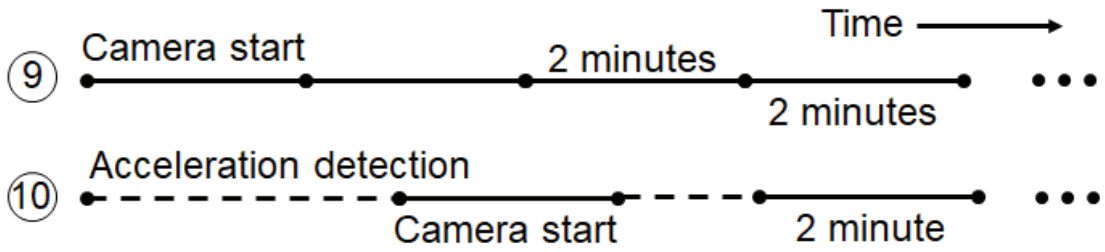


図 4.7: 加速度センサを用いた動作の検知によるノーマリオフ処理の時間経過.

実験環境は 4.4.2 節と同様であり，2 条件についてそれぞれ 6 回実行し，モバイルバッテリから消費電力の計測及び比較を行う．図 4.8 に各条件の消費電力及び削減率を示す．縦軸及び横軸は 4.4.1 節と同様であり，消費電力の削減率は図中の丸点で示す．結果から，⑨の平均消費電力は約 4.9 Wh，⑩の平均消費電力は約 3.1 Wh である．また，⑨に対する⑩の消費電力の削減率は約 37 %となった．すなわち，陸上変温動物に取り付けた市販の加速度センサにより動作を検知でき，イメージセンサを加速度センサの検知タイミングに合わせて間歇動作をさせることにより消費電力の削減が可能であることがわかった．なお，60 分の間にニホンイシガメが動作し，加速度センサが動作を検知した回数は，動画撮影の平均回数から 9.5 回であることを確認した．陸上変温動物は活動量が少なく，データロガーも動作に合わせ起動及び停止を繰り返すことによって消費電力を抑えることが可能である．ここで，4.4.2 節と同様にデータロガーの駆動時間を表 4.4 に示す．表 4.4 から，容量が大きいバッテリを利用した場合にはデータロガーの駆動時間が長くなり，⑨に対する⑩の駆動時間は約 59 %増加している．すなわち，イメージセンサを加速度センサの検知タイミングに合わせてノーマリオフ動作を行うことで，常時駆動よりも長時間データロガーを駆動可能である．本実験についても，一般的な電池により約 1 日の観察が可能となり，陸上変温動物の観察に有効なデータロガーとなることが確認できた．なお，本実験では 4.4.2 節よりも全体的に消費電力が大きくなっているが，加速度センサ（LSM9DS1）を利用する場合には Sense HAT に組込まれている加速度センサ以外のセンサも起動するため，消費電力が大きくなったと考えられる．しかしながら，加速度センサを間歇動作のトリガとすることは，4.4.1 節で確認した②〜④の条件に当てはまり，ノーマリオフ動作を適用する頻度として効果的であると確認できた．よって，活動量が少ない陸上変温動物の観察にノーマリオフバイオロギング用データロガーを用いることで，加速度センサをトリガとして各種センサを間歇動作させることは消費電力削減に有効

な手法である.

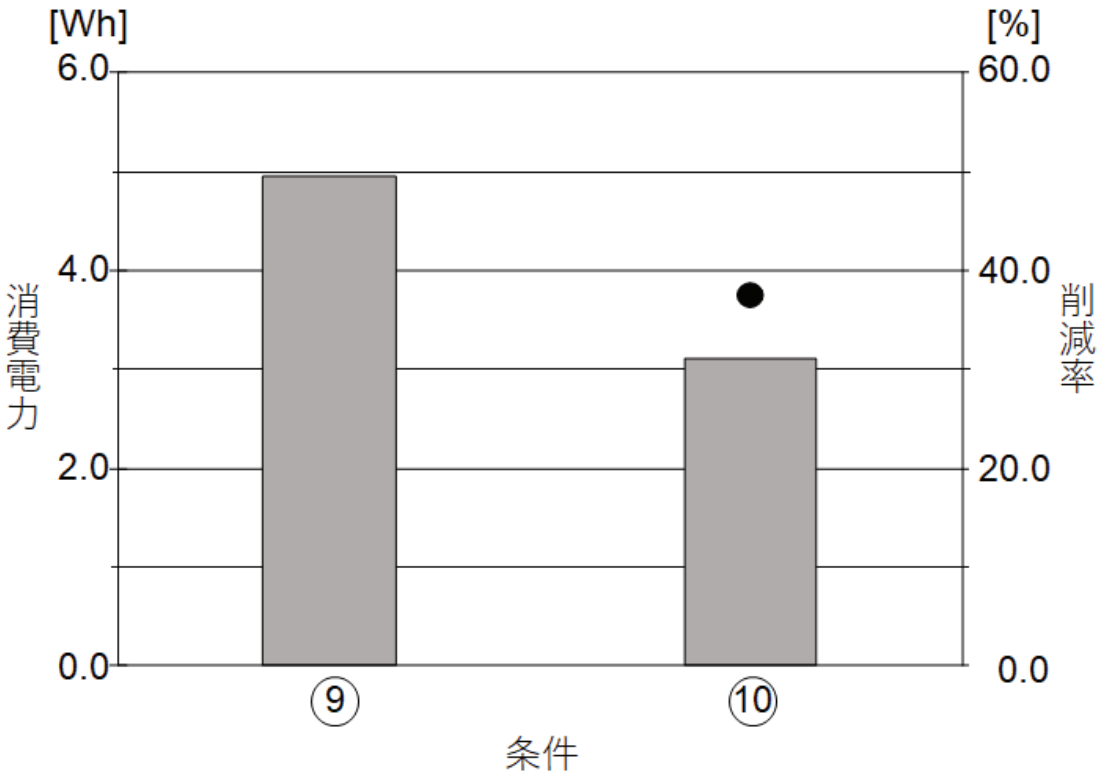


図 4.8: 加速度センサをトリガとしたノーマリオフ処理による消費電力及び削減率.

表 4.4: 加速度センサを用いた動作の検知によるノーマリオフデータロガーの駆動時間算出

	駆動時間 [h]			
	Panasonic エネループ スタンダードモデル	リチウムイオンポリマー電池	Anker PowerCore Essential 20000	700-BTL 018BK
⑨	0.5	1.5	15.0	7.8
⑩	0.7	2.4	23.9	12.4

4.4.4　加速度センサの検知と陸上変温動物の動作

本実験は，加速度センサが動作を検知し場合，実際にニホンイシガメが動作しているかについて調査し，加速度センサによるニホンイシガメの動作検知の精度を確認する．ニホンイシガメの動作は水平方向の動きが多く，一度の動作は数分程度である．このため，加速度センサがニホンイシガメの動作を検知した後に，3 秒又は 5 秒以内に動作を再度検知した場合にイメージセンサを起動する条件と加速度センサの水平方向（x 軸及び y 軸）のみの動作を検知した場合にイメージセンサを起動する条件を比較する．なお，加速度センサが検知した場合にイメージセンサを起動しているが，1 時間における精度を検証する目的で，加速度センサは常時駆動させ，1 時間当たりの加速度センサの検知回数とニホンイシガメの動作回数が一致しているかを観察した．図 4.9 に各実験条件を示す．条件 ⑪は，プログラムを開始した後に加速度センサが駆動し続け，ニホンイシガメの動作を検知した場合にイメージセンサを起動する．イメージセンサは起動後，1 分間の動画を撮影し，保存する．この一連の処理を繰り返し実行する．条件 ⑫及び条件 ⑬は，プログラムを開始した後に加速度センサが駆動し続け，ニホンイシガメの動作を検知した場合に 3 秒又は 5 秒の間に加速度センサで再度動作を検知した時にイメージセンサを起動して 1 分間の動画を撮影・保存する．この一連の処理を繰り返し実行する．条件 ⑭は，加速度センサの水平方向（x 軸及び y 軸）のみの動作を検知した場合にイメージセンサを起動するして 1 分間の動画を撮影・保存する．この一連の処理を繰り返し実行する．

本実験は加速度センサ（GY-291）を利用するが，加速度センサ（GY-291）は加速度センサ（LSM9DS1）より検知精度は低い．しかし，活動量が少ない陸上変温動物を観察する場合，高精度な検知能力は必要がなく，精度が低い方がより正確に動作を検知できるため，本実験では加速度センサ（GY-291）を利用することにする．また，消費電力も考慮すると，加速度センサ（LSM9DS1）は 2.2 V で 600 μA であることから約 1.3 mW であるのに対して [186],[187]，加速度センサ（GY-291）は 2.5 V で 140 μA であることから約 0.4 mW である [189]．すなわち，加速度センサ（GY-291）は加速度センサ（LSM9DS1）よりも消費電力の削減が見込まれ，本実験では前節の実験よりも消費電力が削減されることは明らかであるため，消費電力の比較は行わない．実験環境は，4.4.2 節と同様である．4 条件について，観察者がニホンイシガメの動作と加速度センサ（GY-291）の検知タイミングが一致しているかを確認する．加速度センサ（GY-291）の検知結果を表 4.5 に示し，適合率，再現率及び F 値を以下の式 (4.3), (4.4), (4.5) で算出する．

$$\text{適合率 [\%]} \quad = \frac{\alpha}{\beta} \quad \times 100 \tag{4.3}$$

α ： 加速度センサが検知し，かつニホンイシガメが動作した回数

β ： 加速度センサの検知回数

$$\text{再現率 [\%]} \quad = \frac{\gamma}{\delta} \quad \times 100 \tag{4.4}$$

γ ： ニホンイシガメが動作し，かつ加速度センサが検知した回数

δ ： ニホンイシガメが動作した回数

$$F\text{ 値 [\%]} \quad = \frac{2 \times \text{適合率} \times \text{再現率}}{\text{適合率} + \text{再現率}} \quad \times 100 \tag{4.5}$$

条件⑪は，適合率及び再現率はそれぞれ 86 %のため，F 値も 86 %になっている．条件⑪は加速度センサ（GY-291）を常時駆動させているため，条件⑫，⑬及び⑭が条件⑪の値よりも大きくなった場合，より正確にニホンイシガメの動作を検知できると考えられる．条件⑫は，適合率は 100 %となっているが，再現率は 31 %，F 値は 47 %と小さい数値になっている．条件⑬は，適合率は 100 %となっているが，再現率は 50 %，F 値は 67 %と小さい数値になっている．すなわち，条件⑫及び⑬では，ニホンイシガメの無動作について誤検知を低減できたが，ニホンイシガメの動作に対して検知できない場合が増加した．このため，条件⑫及び⑬の様に，加速度センサ（GY-291）が動作を検知してから，3 秒後又は 5 秒後に動作を再度検知した場合にはニホンイシガメが動作しているとしてデータロガーをノーマリオフ動作させると，加速度センサ（GY-291）を常時駆動させた条件⑪よりも F 値が小さくなり，ニホンイシガメの動作を正確に検知できないことが確認できた．一方で，条件⑭は加速度センサ（GY-291）の水平方向のみを検知することで，適合率，再現率及び F 値は 96 %となり，条件⑪よりもニホンイシガメの動作を正確に検知できることを確認した．以上の結果から，加速度センサ（GY-291）はニホンイシガメの動作をほぼ正確に検知できることが確認できた．

また，本節で利用したデータロガーは，4.4.1 節から加速度センサ（GY-291）に変更し，防水温度センサを追加している．また，イメージセンサによる撮影方式の全条件は同様であるため，条件①の消費電力から条件⑪ ～ ⑭の消費電力を試算する．条件①の消費電力は，60 分間イメージセンサで撮影を続けた場合であり，約 4.3 Wh であった．このため，イメージセンサを 1 回起動した時に 1 分間の撮影を

行うことを考慮すると，条件⑪ ~ ⑭の 1 時間当たりのイメージセンサの稼働率は，それぞれ 30 %，7 %，12 %及び 42 %となる．すなわち，条件①の消費電力は約 4.3 Wh であるため，条件⑪ ~ ⑭の消費電力を試算すると，それぞれ約 1.3 Wh，約 0.3 Wh，約 0.5 Wh 及び約 1.8 Wh となる．これらの結果は，単純に加速度センサ（GY-291）の検知回数とイメージセンサの撮影時間を乗算しており，条件⑪ ~ ⑭を比較した場合，検知回数が増加すると消費電力も大きくなることが確認できる．一方，検知率の精度が最も高かった条件⑭はニホンイシガメが動作した時にのみイメージセンサが起動するため，条件①と条件⑭の消費電力を比較すると約 58 %の消費電力を削減できると考えられる．さらに，加速度センサ（GY-291）は加速度センサ（LSM9DS1）よりも約 77 %の消費電力を削減できることから，条件①よりも条件⑭の消費電力の方がより削減できると考えられる．よって，活動量の少ない陸上変温動物では，水平方向の動作検知を行う加速度センサ（GY-291）を利用することで誤検知が解消され，これをノーマリオフ動作のトリガとすることにより，イメージセンサを頻繁に起動・停止する必要ながなくなり，消費電力の削減が可能となる．これは，陸上変温動物の動作方向は水平移動に限られているためであり，消費電力の削減と陸上変温動物の活動時のみの観察を両立できることとなる．

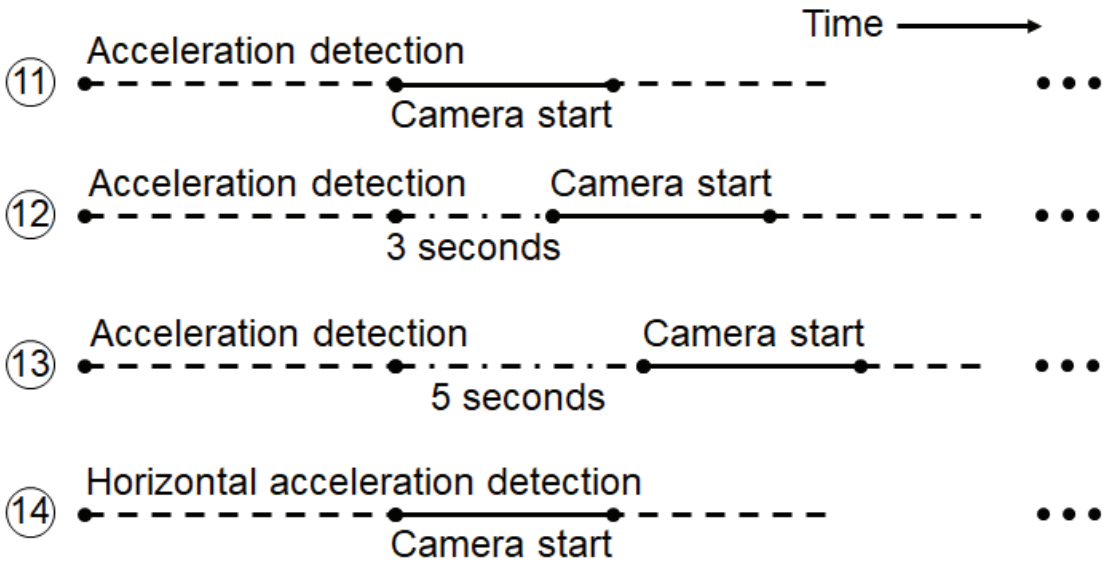


図 4.9: 4.4.4 節の各条件における処理の模式図.

表 4.5: ノーマリオフバイオロギング用データロガーの検知精度

			ニホンイシガメの動作		適合率 [%]	再現率 [%]	F 値 [%]
			動作	無動作			
加速度センサによる検知	⑪	検知	18	3	86	86	86
		非検知	3	-			
	⑫	検知	4	0	100	31	47
		非検知	9	-			
	⑬	検知	7	0	100	50	67
		非検知	7	-			
	⑭	検知	25	1	96	96	96
		非検知	1	-			

4.5 まとめ

本論文において，活動量が少ない陸上変温動物の 1 日以上の生態を観察するバイオロギングを実現するため，ニホンイシガメを観察対象としてデータロガーの開発を行った．活動量が少ない動物に直接取り付けるデータロガーは多く開発されていないが，この様な動物の 1 日の自然な行動を観察することは外敵や交雑の問題を解決できるため重要である．本論文の結果，RaspberryPi 3 の様な比較的大型なコンピュータにおいてもノーマリオフ手法により低消費電力の駆動が可能になることを確認できた．さらに，消費電力が大きいイメージセンサを加速度センサによる動物の行動をトリガとしてノーマリオフ動作を行うことで，消費電力を 37 %削減できることを確認した．特に，水平方向の動作が多いニホンイシガメの行動を考慮して，加速度センサの検知を水平方向に制限することで，間歇動作を 96 %という精度で実現可能であると確認した．これら消費電力の削減及びトリガ検知精度の向上により，一般的に販売されている電池を用いても，陸上変温動物の行動を 1 日間観察可能となった．なお，本論文においては，データロガーの基礎データを取得するために間歇動作を RaspberryPi 3 のソフトウェア上で実装したが，RaspberryPi 3 の電源管理/死活監視モジュールである slee-Pi2 [200] を利用することでより消費電力の削減が見込まれる [174]．加えて，小型のマイコンセンサを利用することもさらなる消費電力の削減につながる．この様な手法により，より小型かつ低消費電力なデータロガーを実現することで陸上変温動物の生態解明や保全への貢献が可能である．

モバイルデバイスは半導体の技術発展と共に動物の生態を解明する分野に広がりを見せており，近年ではこの様な技術は人間の体調管理等をセンシングするウェアラブルデバイスとしても広がりを見せている．これまでのモバイルデバイスは加速度センサや GPS センサ等を利用することで観察等を行ってきたが，より様々な情報を取得するために消費電力は大きくなるが，膨大な情報量を持つイメージセンサを利用するようになっている．このため，動物や人間の観察等を行うためのモバイルデバイスはさらなる低消費電力が求められる一方で，膨大なデータ量を処理可能な性能が要求されている．このため，これらのデータロガーやウェアラブルデバイスに CAMX を組込むことで，小型かつ低消費電力という制約の中で，膨大なデータ量を高速かつ高性能に処理可能なモバイルデバイスを実現可能であると考えている．また，本章において検証したノーマリオフ手法は CAMX にも適用可能であると考える．例えば，CAMX においてはバリッドフラグを用いることで実行の有無をエントリ毎に操作できるため，エントリ毎に細かく実行すべきエントリを操作することで消費電力の削減につながると考える．今後，ユーザサポートなモバイルデバイスに CAMX を組込むことで，センサから取得したデータの処

理及び詳細なエントリ操作により，低消費電力かつ高性能なモバイルデバイスの開発・検証を行う予定である．

第5章　結論

本論文では，モバイルデバイスにおいてアクセラレータとして機能する，高性能な機能メモリベース超並列 SIMD 型演算コアである CAMX を提案し，処理性能について検証を行った．そして，CAMX に組込まれている連想メモリのサイズを 128 ビット又は 256 ビット × 1,024 エントリとし，1,024 個のデータを並列に実行可能であることを確認した．さらに，今後 CAMX を組込むことを想定し，近年モバイルデバイスを利用して発生するソーシャルハザードの一種である社会課題や犯罪が増加していることを踏まえ，画像改ざん検知手法及び盗撮防止システムの提案を行い，その効果を検証した．今後は半導体技術の面からソーシャルハザードに対する対策が必要であり，CAMX をモバイルデバイスに組込むことで有効な対策になると考える．また，近年急速に普及し始めているユーザサポートなウェアラブルデバイスの低消費電力手法を提案し，その効果を検証した．これらの検証により，CAMX はモバイルデバイス向けのアクセラレータとして高性能に機能し，今後普及する様々なアプリケーションに対しても有効に機能することが想定できた．

本論文で得られた研究成果を，5.1 節に示す．5.2 節には，本研究の今後の応用や将来展望を示す．

5.1　研究成果

第 2 章では，機能メモリベース超並列 SIMD 型演算コア（CAMX）のアーキテクチャ概要を説明し，動作確認を行った．マルティメディア処理に特化したアーキテクチャとして，連想メモリを主構成とする CAMX を提案し，基本演算命令及び検索命令が正確に実行できることを確認した．その上で，マルチメディア処理において必須となる乗算処理について検証し，CAMX における最適な処理手法について評価した．さらに，AES 処理を実装し，CAMX の基でありモバイルデバイスの環境で検証可能な MX-1 及び既存のモバイルデバイス向けプロセッサとの性能比較を行った．また，マルチメディア処理において利用される浮動小数点による加算処理についても実装し，その検証を行った．

(2-1) マルチメディア処理を得意とする機能メモリベース超並列 SIMD 型演算コ

ア（CAMX）を提案した

提案したCAMXのアーキテクチャは，主に連想メモリで構成されており，左右の連想メモリが数千個の演算器を挟み込んだ配置となっている．この様な構成とすることにより，数千個のデータを並列に処理可能となる．並列処理にはSIMD処理を適用し，コントローラからの1命令に対して，数千個の演算器が同時に駆動できるようになっている．さらに，連想メモリの機能である一致検索の処理に特化しており，検索処理により一致したデータの部分に対してのみ実行させる等，演算を行うエントリを自由に選択可能となっている．さらに，演算の高性能化に関しては，演算器の処理をパイプラインに実行することで実現する．

(2-2) CAMXにおける基本演算命令を実行し，正確かつ並列な処理を確認した．

提案したCAMXのアーキテクチャにおける基本演算命令が正確に処理されることを確認するため，論理演算であるAND，OR，XOR及び加算を実行した．この結果，それぞれの演算について正確に処理されていることを確認し，全ての演算に対して数千データを並列に処理できることを確認できた．さらに，これらの処理については，128ビットで1,024個のデータに対して実行した場合，演算に必要なクロックサイクル数は130クロックサイクルであった．すなわち，数千データに対して一定のクロックサイクル数で処理可能であることを示した．

(2-3) CAMXにおける検索命令を実行し，正確かつ並列な処理を確認した．

提案したCAMXのアーキテクチャにおける検索命令が正確に処理されることを確認するため，一致検索処理を実行した．この結果，数千データ全てに対して一致検索を行い，一致したデータについて一致信号を出力していることを確認した．さらに，この処理については，1,024個のデータに対して実行した場合，演算に必要なクロックサイクル数は1クロックサイクルであった．これはデータ数に関係なく，連想メモリに格納されたデータ全てに対して実行されるため，データ数が増加しても1クロックサイクルで実行可能となる．

(2-4) CAMXにおける最適な乗算処理について検証した．

CAMX における最適な乗算処理の手法を検討するため，ビットシリアル乗算，検索加算繰り返し乗算，Baugh-Wooley 乗算について比較を行った．ビットシリアル乗算は検索命令を利用せずに 1 ビットずつ乗算を行う手法であり，検索加算繰り返し乗算は検索命令を利用して一致した時にのみ加算処理を行う手法である．これらの演算に対して 4 ビット乗算を実行したところ，ビットシリアル乗算では 1,462 クロックサイクルであり，検索加算繰り返し乗算では 237 クロックサイクルであったため，クロックサイクル数は約 84 %削減可能であることを確認した．Baugh-Wooley 乗算は Baugh-Wooley 演算の公式に沿った処理手法であり，検索加算繰り返し乗算とのクロックサイクル数を比較したところ，乗算対象のデータビット幅が 15 ビット以上の場合には Baugh-Wooley 乗算が有効であることを確認した．すなわち，CAMX においては，15 ビット未満の乗算では検索加算繰り返し乗算がより高速に実行可能であり，15 ビット以上の乗算では Baugh-Wooley 乗算がより高速に実行可能であることを示した．

(2-4) CAMX における AES 処理の実装及び有効性の検証を行った．

CAMX の性能を検証するため，AES 処理を実装し，CAMX の基でありモバイルデバイスの環境で検証可能な MX-1 及び既存のモバイルデバイス向けプロセッサとの性能比較を行った．CAMX において AES 処理を実装したところ，処理に必要な総クロックサイクル数は 1,362,699 クロックサイクルとなった．AES 処理は 10 ラウンドの繰り返し演算で構成されているが，より詳細な検証を行うために，CAMX と MX-1 の 1 ラウンド当たりのクロックサイクル数について比較した．この結果，1 ラウンドの総クロックサイクル数については，MX-1 の方が CAMX よりも小さかったが，ShiftRows と MixColumns の合計クロックサイクル数については CAMX が MX-1 よりも約 65 %，AddRoundKey の合計クロックサイクル数については約 86 %も高速に処理できていることが確認できた．さらに，CAMX と既存のモバイルデバイス向けプロセッサの性能を比較したところ，TI OMAP3530 （ARM Cortex A8）との比較においては，CAMX の方が約 53 %も高性能であることが確認でき，その他のプロセッサについても CAMX の方が高性能であることが確認できた．

(2-5) CAMX における浮動小数点による加算処理の実装及び有効性の検証を行った．

マルチメディア処理において利用される浮動小数点による加算処理を実装し，

その検証を行った．単精度浮動小数点による加算処理について，単純な実装と 2 の補数削減処理による実装を行い，2 の補数削減処理による実装はより高性能に実行可能であることを確認した．さらに，RaspberryPi 4 に組込まれている ARM コアと処理性能を比較したところ，1.5 GHz のクロック周波数で動作したと想定した CAMX は処理データ数が 4,500 を超えるとよりも高性能になることが確認できた．すなわち，CAMX は膨大なデータ量の処理を要求するマルチメディア処理に対して有効であることが確認できた．

第 3 章では，ソーシャルハザードである社会課題及び犯罪に対する対策について，CAMX を組込むことを想定した手法及びシステムを提案した．画像に対して繰り返し演算が必要となるモルフォロジカルパターンスペクトラム処理を利用した画像改ざん検知手法について検討を行った．この手法をモバイルデバイス上で実行するには，大量のデータを繰り返し演算する必要があり，高性能なプロセッサが要求される．そこで，画像改ざん検知手法を提案すると共に，CAMX の基でありモバイルデバイスの環境で検証可能な MX-1 において検証を行い，モバイルデバイス向けのアクセラレータとして有効に機能することを確認した．そして，モバイルデバイスにおいて要求される画像処理及びリアルタイム性を必要とするアプリケーションとして盗撮防止システムを提案した．本システムは特定の信号を検知する必要があるため，一致検索処理に特化した CAMX はより高速に信号を検知可能であると考えらる．

(3-1) モルフォロジカルパターンスペクトラム処理を利用した画像改ざん検知手法を提案した．

近年，半導体の技術が急速に発展したことによって，一般人が気軽に動画像の編集・加工ができるようになり，ディープフェイクの様な動画像改ざんが大きな問題となっている．そこで，モルフォロジカルパターンスペクトラム処理を利用した画像改ざん検知手法について提案した．本手法は，対象画像に対して構造要素を用いてオープニング処理，クロージング処理を繰り返し，それぞれの処理された画像に対して差分処理を施すことで，対象画像の特徴を抽出する手法である．従来の手法では検知できなかった回転画像等に対しても正確に画像改ざんを検知できることを確認した．

(3-2) MX-1 にモルフォロジカルパターンスペクトラム処理を実装し，その検証を行った．

MX-1 はモバイルデバイスの環境で検証でき，CAMX の基となった技術である．このため，MX-1 で有効性を確認したアプリケーションについては，CAMX においても同様の処理となる．この MX-1 に画像処理であり，データ量が多く繰り返し演算が必要となるモルフォロジカルパターンスペクトラム処理を実装した．この結果，プログラムのユーザマイクロコード化により処理速度は約 5 倍に向上し，一般的に普及しているモバイルデバイス向けプロセッサよりも消費電力当たりのスループットは 2 倍になった．さらに，プロセスルールの違いを考慮した場合，消費電力当たりのスループットは約 20 倍にもなった．

(3-3) 盗撮防止システムを提案し，そのシステムの構築を行った．

近年，半導体技術の急速な発展により，モバイルデバイスのイメージセンサは高画質になっている．このため，モバイルデバイスのカメラ機能を利用した一般人の犯罪行為が急増している．そこで，モバイルデバイスにおける盗撮防止システムを提案し，そのシステムを構築した．本システムは LED 照明とスマートフォンを可視光ビーコンにより連携させることで，スマートフォンのカメラ機能を強制的に停止させるものである．LED 照明については，日常的に利用されるものであり，人間の目に対して不快感の無い 1/f ゆらぎに正弦波を組合わせた可視光ビーコンを送信することとした．そして，スマートフォンには，イメージセンサから取得した動画の画素値から可視光ビーコンを検出し，そのビーコンに対して周波数分解を行うことで正弦波を抽出し，特定の正弦波を検出した場合にカメラ機能を停止させるシステムを構築した．この結果，スマートフォンのカメラ機能を強制的に停止可能であることを確認した．このスマートフォンに CAMX を組込むことで，カメラから取得した動画像に対してより高速に処理可能であり，特定の正弦波を容易かつ高速に検出することが可能であると考える．

第 4 章では，モバイルデバイスがデータロガーやウェアラブルデバイスの様により身近になってきていることを踏まえ，これらのデバイスにおいても CAMX を組込むことを想定する．これらのデバイスでは，低消費電力かつ小型という制約がある中で，処理すべきデータ量は増加しているため，性能を維持したまま消費電力の削減手法について検証した．データロガーやウェアラブルデバイスに CAMX を組込むことで，より生活に身近なデバイスとなり，ユーザをサポートできるようなデバイスになると考える．

(4-1) バイオロギング用データロガーを開発し，ノーマリオフ手法を適用した．

近年，人間や動物に直接取り付けて，体調や生態を解明しようという研究が増加している．このため，低消費電力であるモバイルデバイスが求められている．そこで，まずは，動物の生態を解明するためのバイオロギング用データロガーの開発を行った．そして，このデータロガーが低消費電力に駆動するための手法として，ノーマリオフ手法の実装も行った．この結果，低消費電力なデータロガーを開発でき，1 日以上の観察が可能であることを確認した．特に，消費電力は大きいが，近年観察の主流となっているイメージセンサについて，ノーマリオフ手法により約 37 %の消費電力削減になることを確認した．

(4-2) 観察対象動物の動作に協調したデータロガーを提案し，ノーマリオフ動作を正確に実行した

バイオロギング用データロガーの消費電力をさらに削減するために，ノーマリオフ手法のトリガとして観察対象である動物の動作と協調することを提案した．実験では，観察対象であるニホンイシガメの動作は水平方向が多いことに着目し，加速度センサの検知方向を水平方向のみに限定し，この加速度センサで動きを検知した際にデータロガーの駆動を行った．この結果，ニホンイシガメの動作を 96 %という精度で検知できた．検知精度の向上で各種センサの間歇動作は必要に応じて適切に行われるため，より低消費電力なデータロガーを実現可能となる．

5.2 研究成果の応用と将来展望

本研究では，マルチメディア処理において必要となる膨大なデータ量及び演算量を処理可能なモバイルデバイス向けアクセラレータである CAMX について提案，開発及び検証を行った．以下に，この CAMX における今後の研究成果の応用と将来展望について議論する．

5.2.1 研究成果の応用

本研究成果の今後の応用として，主として以下の 3 つが挙げられる．

(1) AI 処理等の機械学習に対する応用．

(2) CAMX を組込んだモバイルデバイスによるソーシャルハザード対策の実現.

(3) マルチメディア処理に対応したユーザサポート向けモバイルデバイスの実現.

(4) 外部メモリとしての応用

以下，それぞれについて述べる.

(1) 近年，AI 処理が急速に普及しており，膨大なデータ量及び演算量の処理が要求され，音声等だけではなく，動画像の処理においても利用されている．このため，より高性能なプロセッサが求められている．加えて，これまでは膨大な演算はクラウド上の大規模パソコンにて処理されていたが，近年は通信状況やリアルタイム性を重視する観点からモバイルデバイス上で AI 処理を完結することが提案されている．本研究で提案した CAMX をモバイルデバイスのプロセッサに組込むことで，AI 処理のアクセラレータとして機能することで高性能化が実現可能となる．CAMX は特に膨大なデータに対する繰り返し演算についてより効果的に実行できるため，AI 処理の様な繰り返し演算を得意とする.

(2) 近年，モバイルデバイスの高性能化に伴って，モバイルデバイスを利用した社会課題や犯罪等のソーシャルハザードが問題となっており，その対策が求められている．本論文では，ソーシャルハザードに対する対策手法及びシステムを提案し，その有効性をソフトウェアにより検証した．これらの手法及びシステムをハードウェアに応用し，CAMX を組込んだモバイルデバイスにおいて実現する．また，現在，様々なソーシャルハザードが発生しており，本論文で提案した対策手法及びシステム以外についても，CAMX を組込んだモバイルデバイスを利用することでより高性能かつ有効性のある対策が実現可能であると考える．今後，CAMX を組込んだモバイルデバイスの実現及び対策手法及びシステムの実装について検討する予定である.

(3) 近年，モバイルデバイスの利用範囲は急速に広がっており，動物の生態観測や人間の体調管理等，より身近なデバイスとなってきている．しかし，ユーザサポートを行うためには，動物及び人間がモバイルデバイスを常に身に付けることになるため，行動や生活に支障のないデバイスが要求される．さらに，ユーザサポート向けのモバイルデバイスは様々な情報を処理するようになっており，小型，低消費電力，高性能が必要となってきている．また，様々な情報を取得するためにイメージセンサを組込むことが必須になってきており，データ量が増加する中で，低消費電力かつ高性能なマルチメディア処理

をリアルタイムに実行する必要性が増加している．これらのモバイルデバイスにマルチメディア処理を得意とする CAMX を組込むことで，より応用範囲が広がると考える．高性能という要求に対して CAMX は高並列に膨大な処理を実行可能である．さらに，CAMX は連想メモリに格納されたデータに対してエントリ毎に検索や演算命令を自由に選択できるため，エントリ毎に別の処理も実行可能であるため，各種センサから取得した様々な情報を同時に実行することが可能になる．また，全エントリを同時に実行する必要がないため，消費電力の削減にもつながると考える．

(4) 本研究で提案した CAMX はシステムバスを介して，CPU や外部メモリに接続されている．処理を実行する場合は，CPU から CAMX へ命令が送信され，CAMX は CPU からの命令により外部メモリのデータを CAMX 内の連想メモリに格納することで実行する．CAMX に組込まれている連想メモリは，大量のデータを格納でき，全ての格納データに対して並列に処理可能である．一方，外部メモリから CAMX 内の連想メモリにデータを格納するには，データ量に合わせた入出力に係る処理時間が必要となるため，データ量が増加すると入出力の処理時間も増加する．すなわち，この処理時間を低減させることにより，さらなる高性能が実現可能となる．このため，現在の CAMX では外部メモリに格納されたデータを連想メモリに都度格納する必要があるが，CAMX の連想メモリに外部メモリの役割を付加することで，外部メモリを介することなく連想メモリに直接データを格納した場合，入出力時の処理速度はより向上すると考える．さらに，モバイルデバイス全体で考えると，外部メモリが不要となることから，より小型かつ低消費電力なデバイスになると考えられる．CAMX 内の連想メモリに格納されたデータはそのまま演算器で処理も可能となる．今後，これらの手法についても検討する予定である．

5.2.2 将来展望

近年，モバイルデバイスにおいて，膨大なデータ量及び演算量を実行可能なプロセッサが求められている．特に，AI 処理等はこれまで高性能なコンピュータにおいて処理されていたが，モバイルデバイス上で実行するようになっている．これまでの AI 処理においては，データ取得はモバイルデバイスのセンサで取得し，取得したデータはクラウド上の AI 学習システムにおいて処理するのが一般的であったが，通信状況の影響が受けやすいことやリアルタイム性を重視されるようになり，近年ではモバイルデバイス上でセンサのデータ取得から AI 学習まで一括して処理することが一般的になっている．モバイルデバイス上で一括した AI 処理が可能に

なることで，通信遅延やリアルタイム性を解決できるようになっている．しかしながら，モバイルデバイス上で一括した AI 処理を行うため，モバイルデバイスにはより高性能なプロセッサが要求されており，近年では NPU という AI 処理に特化した専用回路を搭載したデバイスが登場している．一方，この様な高性能な処理を行うために NPU の様な専用回路を複数搭載することになり，回路規模が増大する問題も考えられる．モバイルデバイスは小型かつ低消費電力という制約がある中で，より高性能に処理する必要がある．このため，専用回路を多数組込むことは，モバイルデバイスの小型化及び低消費電力化を実現できない可能性がある．そこで，本論文では超並列 SIMD 型演算コア（CAMX）を提案し，汎用性を持ちつつ，高性能な処理が可能なモバイルデバイス向けアクセラレータを提案し，その有効性を検証している．CAMX は左右の連想メモリで演算器を挟み込んだ構成となっており，数千データを並列に実行できる特徴がある．このため，マルチメディア処理の一つである画像処理等の様に，データ量及び演算量が多く，繰り返し演算が必要となる処理に対して高速かつ高性能に処理可能となる．さらに，CAMX は主に連想メモリで構成されているため，連想メモリに格納されているデータに対して一致検索処理を高速かつ容易に実行することができ，テーブルルックアップ処理の様な一致検索を行った後に変換が必要な処理に対しても数千データを並列に実行可能となる．以上の様に，近年求められている膨大なデータ量及び演算量に対する処理が容易に行えるため，CAMX は AI 処理等を高性能に処理可能である．すなわち，低消費電力かつ小型という制約があるモバイルデバイスに実装することで，逐次処理については CPU で実行し，CPU では処理しきれないようなデータについては CAMX において処理することで高速かつ高性能に実行が可能となる．さらに，CAMX は特定の処理に特化した回路ではなく，汎用的に様々な処理を高性能に実行できるため，現在のモバイルデバイスのように様々な専用回路を組込むことなく，CAMX のみを組込むだけで同等以上の性能を実現可能となる．よって，CAMX は小型かつ低消費電力を実現しつつ，高性能に演算可能なモバイルデバイス向けアクセラレータとなり得る．

付録

以下に CAMX におけるコマンド一覧を示す．

CAMX を動作させるためには，CAMXLIB コマンドポートに 32 ビットのデータを入力する．CAMXLIB のビット区分は以下の通り．

CAMXLIB = { 翼/PE 選択，動作命令，C ビット，B ポインタ，A ポインタ } = {2'bxxx, 6'bxxxxx_xxxx, 8'bxxx_xxxx, 8'bxxx_xxxx, 8'bxxx_xxxx}

/**—翼/PE 選択——————————————————————-**/

LEFT_WING: 左翼を選択 RIGHT_WING: 右翼を選択 PE_EXE: 演算器を選択 (翼は動作させない)

/**—動作命令 (CAM に対する動作コマンド)——————————————————-**/

・CAMX_CNTRL_CLEAR: CAMX コントローラのレジスタクリア

・CAMX_DATA_WRITE: 左/右翼の CAM 内ワードにデータを書き込む (同時にアドレスとデータを入力)．A，B ポインタ，及び C ビットは 0 にしておく．

(例) 右翼アドレス 10'b00_0000_0011 に，データ 256'h0000_0000_0000_0000 _0000_0000_0000_0000_0000_ffff_0000_ffff_0000_ffff_0000_ffff を書き込む

#(STEP) ADDRESS = 10'b00_0000_0011;

DIN = 256'h0000_0000_0000_0000_0000_0000_0000_0000_0000 _ffff_0000_ffff_0000_ffff_0000_ffff;

CAMXLIB = {RIGHT_WING, CAMX_DATA_WRITE, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_DATA_READ: 左/右翼の CAM 内ワードからデータを読み出す (同時にアドレスを入力)．A，B ポインタ，及び C ビットは 0 にしておく．

DOUT_VALID が H の時の DOUT が読み出されたデータとなる．

(例) 左翼アドレス 10'b00_0000_0011 から，データを読み出す．

#(STEP) ADDRESS = 10'b00_0000_0011;

CAMXLIB = {LEFT_WING, CAMX_DATA_READ, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_UNMASK_SEARCH: 左/右翼の CAM に対し，マスクを用いない検索処理を行う (同時に検索データを入力)．A，B ポインタ，及び C ビットは 0 にしておく．また，検索結果は外部に出力されると共に，PE 内のレジスタに格納される．
ただし，検索結果は PE 内のレジスタにも保存される．
(例) 検索データ 256'h0000_0000_0000_0000_0000_0000_0000_0000
_0000_ffff_0000_ffff_0000_ffff_0000_0000 を右翼に入力し，全データの一致検索処理．
#(STEP) SEARCH_DIN = 256'h0000_0000_0000_0000_0000_0000
_0000_0000_0000_ffff_0000_ffff_0000_ffff_0000_0000;
CAMXLIB = {RIGHT_WING, CAMX_UNMASK_SEARCH, 8'b0000_0000,
8'b0000_0000, 8'b0000_0000};

・CAMX_MASK_SEARCH: 左/右翼の CAM に対し，マスクを用いた検索処理を行う (同時に検索データ，マスクを入力)．A，B ポインタ，及び C ビットは 0 にしておく．マスクは 0 で掛ける．また，検索結果は外部に出力されると共に，PE 内のレジスタに格納される．
ただし，検索結果は PE 内のレジスタにも保存される．
(例) 検索データ 256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_ffff
_0000_ffff_0000_ffff_0000_0000 を右翼に入力し，マスクデータ 256'h0000_0000_0000_0000
_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_ffff
_ffff を用いて一致検索処理．
#(STEP) SEARCH_DIN = 256'h0000_0000_0000_0000_0000_0000
_0000_0000_0000_ffff_0000_ffff_0000_ffff_0000_0000;
MASK_DIN = 256'h0000_0000_0000_0000_0000_0000_0000_0000
_0000_0000_0000_0000_0000_0000_0000_ffff_ffff;
CAMXLIB = {RIGHT_WING, CAMX_UNMASK_SEARCH, 8'b0000_0000,
8'b0000_0000, 8'b0000_0000};

/**——動作命令 (PE も含めた動作コマンド)——————————————————————————————————————**/
・CAMX_DATA_REG: CAM からの読み出し値を PE 内のオペレーションレジスタに書き込む．A ポインタを使用，B ポインタ，及び C ビットは 0 にしておく．
(例) 左翼 8'b0000_1100 の位置にあるビットを PE 内のオペレーションレジスタに格納．

#(STEP) CAMXLIB = {LEFT_WING, CAMX_DATA_REG, 8'b0000_0000, 8'b0000_0000, 8'b0000_1100};

・CAMX_REG_DATA: PE 内のオペレーションレジスタの値を CAM 左/右翼の A ポインタへ格納する．A ポインタを使用，B ポインタ，及び C ビットは 0 にしておく．

(例) PE 内のオペレーションレジスタに格納されているデータを，右翼 8'b0000_0001 へ書き込み．

#(STEP) CAMXLIB = {RIGHT_WING, CAMX_REG_DATA, 8'b0000_0000, 8'b0000_0000, 8'b0000_0001};

・CAMX_DATA_AND: CAM からの読み出し値を PE 内で AND する．始めに左/右翼から A ポインタの値に従って読み出し，次に右/左翼から B ポインタの値に従って読み出し，C ビットの演算を行った後に，左/右翼の A ポインタから書き込む．

(例) 左翼 8'b0100_0000 のデータを読み出しオペレーションレジスタへ格納，次に右翼 8'b0000_0000 のデータを読み出し，8'b0000_0111 ビットの AND 演算を行い，左翼 8'b0100_0000 から結果を格納する．

#(STEP) CAMXLIB = {LEFT_WING, CAMX_DATA_AND, 8'b0000_0111, 8'b0000_0000, 8'b0100_0000}; //C ポインタビット幅は+1 で考える

・CAMX_DATA_OR: CAM からの読み出し値を PE 内で OR する．始めに左/右翼から A ポインタの値に従って読み出し，次に右/左翼から B ポインタの値に従って読み出し，C ビットの演算を行った後に，左/右翼の A ポインタから書き込む．

(例) 右翼 8'b0000_0000 のデータを読み出しオペレーションレジスタへ格納，次に右翼 8'b0000_0000 のデータを読み出し，8'b0111_1111 ビットの OR 演算を行い，左翼 8'b0000_0000 から結果を格納する．

#(STEP) CAMXLIB = {RIGHT_WING, CAMX_DATA_OR, 8'b0111_1111, 8'b0000_0000, 8'b0000_0000}; //C ポインタビット幅は+1 で考える

・CAMX_DATA_XOR: CAM からの読み出し値を PE 内で XOR する．始めに左/右翼から読み出し，次に右/左翼から読み出し，演算を行った後に，左/右翼に

書き込む.

(例) 左翼 8'b0000_0011 のデータを読み出しオペレーションレジスタへ格納，次に右翼 8'b0000_0010 のデータを読み出し，8'b0111_0000 ビットの XOR 演算を行い，左翼 8'b0000_0011 から結果を格納する.

#(STEP) CAMXLIB = {LEFT_WING, CAMX_DATA_XOR, 8'b0111_0000, 8'b0000_0010, 8'b0000_0011}; //C ポインタビット幅は+1 で考える

・CAX_DATA_NOT: 左/右翼 CAM のデータを A ポインタで示した位置から，C ビット分 NOT する．A ポインタ，及び C ビットを使用，B ポインタは 0 にしておく.

(例) 左翼 8'b0110_0000 から，8'b0000_1111 分のデータを反転する.

#(STEP) CAMXLIB = {LEFT_WING, CAMX_DATA_NOT, 8'b0000_1111, 8'b0000_0000, 8'b0110_0000}; //左 (右) 翼のデータを否定，A: 読み込みポインタ，B: 不使用

・CAMX_DATA_ADD: CAM からの読み出し値を PE 内で ADD する．始めに左/右翼から A ポインタの値に従って読み出し，次に右/左翼から B ポインタの値に従って読み出し，C ビットの演算を行った後に，左/右翼の A ポインタから書き込む.

桁上がりのデータは自動的にプリザーブレジスタに格納される.

(例) 左翼 8'b0000_0000 のデータを読み出しオペレーションレジスタへ格納，次に右翼 8'b0000_0000 のデータを読み出し，8'b0111_1111 ビットの ADD 演算を行い，左翼 8'b0000_0000 から結果を格納する.

#(STEP) CAMXLIB = {LEFT_WING, CAMX_DATA_ADD, 8'b0111_1111, 8'b0000_0000, 8'b0000_0000}; //C ポインタビット幅は+1 で考える

・CAMX_DATA_COPY: CAM の左/右翼 B ポインタから読み出したデータを，右/左翼の A ポインタへ C ビット分コピーする．B ポインタは読み出し先，A ポインタは書き込み先，C ビット分コピーする.

(例) 左翼 8'b0000_0000 から，8'b0000_1111 ビット分のデータを，右翼 8'b0100_0000 へコピーする.

#(STEP) CAMXLIB = {RIGHT_WING, CAMX_DATA_COPY, 8'b0000_1111, 8'b0100_0000, 8'b0000_0000};

・CAMX_REG_VALID: PE 内の一致検索結果レジスタデータを PE 内のバリッドレジスタに格納．A ポインタ，B ポインタ，及び C ビットは 0 にしておく．
(例) PE 内のオペレーションレジスタに格納されているデータを，バリッドレジスタに格納．
#(STEP) CAMXLIB = {PE_EXE, CAMX_REG_VALID, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_PE_VALID: PE 内のバリッドレジスタを 1 にする．A，B ポインタ，及び C ビットは 0 にしておく．
(例) PE 内のバリッドレジスタを全て 1 にする．
#(STEP) CAMXLIB = {PE_EXE, CAMX_PE_VALID, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_PE_INVALID: PE 内のバリッドレジスタを 0 にする．A，B ポインタ，及び C ビットは 0 にしておく．
(例) PE 内のバリッドレジスタを全て 0 にする．
#(STEP) CAMXLIB = {PE_EXE, CAMX_PE_INVALID, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_DATA_UP: CAM の左/右翼 A ポインタから読み出したデータを，左/右翼の B ワード上方の A ポインタへ C ビット分垂直移動する．
ただし移動量は 2 ビット 4 つまで．B ポインタは 00〜11 まで．
(例) 左翼 8'b0000_0000 から，16 ビット分のデータを，上方 3 つ移動する．
#(STEP) CAMXLIB = {LEFT_WING, CAMX_DATA_UP, 8'b0000_1111, 8'b0000_0010, 8'b0000_0000};

・CAMX_DATA_DOWN: CAM の左/右翼 A ポインタから読み出したデータを，左/右翼の B ワード下方の A ポインタへ C ビット分垂直移動する．ただし移動量は 2 ビット 4 つまで．B ポインタは 00〜11 まで．
(例) 左翼 8'b0000_0000 から，16 ビット分のデータを，下方 3 つ移動する．

#(STEP) CAMXLIB = {LEFT_WING, CAMX_DATA_DOWN, 8'b0000_1111, 8'b0000_0010, 8'b0000_0000};

・CAMX_NO_OPE: CAM のノーオペレーション
CAMX を動作させたくない時等に用いる．
#(STEP) CAMXLIB = {PE_EXE, CAMX_NO_OPE, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_ACCUM_VALID: PE 内の一致検索結果レジスタデータを PE 内のバリッドレジスタに格納．このとき上書きせず OR することで 1 になった履歴が増えていく．A ポインタ，B ポインタ，及び C ビットは 0 にしておく．
(例) PE 内のオペレーションレジスタに格納されているデータを，バリッドレジスタに OR 格納．
#(STEP) CAMXLIB = {PE_EXE, CAMX_ACCUM_VALID, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_INV_VALID: PE 内のバリッドレジスタを反転させる．A，B ポインタ，及び C ビットは 0 にしておく．
(例) PE 内のバリッドレジスタを反転する．
#(STEP) CAMXLIB = {PE_EXE, CAMX_INV_VALID, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_ALL_WRITE: CAM 内全てのワードに同時書き込み．valid で書き込みたいところと書き込みたくないところを制御できる．CAM の左/右翼 A ポインタにデータ B を C ビット分書き込む．
ただし書き込むデータ B は 4 ビットまで．そのため C ビットも 00〜11 まで．
(例) 左翼 8'b0000_0000 から，データ 0010 を，4 ビット分書き込む．
#(STEP) CAMXLIB = {LEFT_WING, CAMX_ALL_WRITE, 8'b0000_0011, 8'b0000_0010, 8'b0000_0000};

・CAMX_VALID_REG: PE 内のバリッドレジスタの値をオペレーションレジスタへ格納する．A ポインタを使用，B ポインタ，及び C ビットは 0 にしておく．

(例) PE 内のバリッドレジスタに格納されているデータを，オペレーションレジスタへ書き込み．

#(STEP) CAMXLIB = {PE_EXE, CAMX_VALID_REG, 8'b0000_0000, 8'b0000_0000, 8'b0000_0000};

・CAMX_CARRY_DATA: PE 内のキャリーレジスタの値を CAM 左/右翼の A ポインタへ格納する．A ポインタを使用，B ポインタ，及び C ビットは 0 にしておく．

加算時の桁上がりデータは自動的にプリザーブレジスタに格納されて保持される．

加算後直ちに行わずとも最終の結果がプリザーブレジスタに保持されているので，それを CAM セルに書き込むことができる．

(例) PE 内のキャリーレジスタに格納されているデータを，右翼 8'b0000_0001 へ書き込み．

#(STEP) CAMXLIB = {RIGHT_WING, CAMX_CARRY_DATA, 8'b0000_0000, 8'b0000_0000, 8'b0000_0001};

・CAMX_DATA_PRESERVE: CAM からの読み出し値を PE 内のプリザーブレジスタに書き込む．ここに格納された値はリセットと ADD 処理以外は保持される．

A ポインタを使用，B ポインタ，及び C ビットは 0 にしておく．

(例) 左翼 8'b0000_1100 の位置にあるビットを PE 内のオペレーションレジスタに格納．

#(STEP) CAMXLIB = {LEFT_WING, CAMX_DATA_PRESERVE, 8'b0000_0000, 8'b0000_0000, 8'b0000_1100};

・CAMX_PRESERVE_DATA: PE 内のプリザーブレジスタの値を CAM 左/右翼の A ポインタへ格納する．A ポインタを使用，B ポインタ，及び C ビットは 0 にしておく．

プリザーブレジスタは加算処理時に桁上がりのデータを保持する, もしくは CAMX_DATA_PRESERVE で格納されたデータを保持する．

(例) PE 内のプリザーブレジスタに格納されているデータを，右翼 8'b0000_0001 へ書き込み．

#(STEP) CAMXLIB = {RIGHT_WING, CAMX_PRESERVE_DATA, 8'b0000_0000, 8'b0000_0000, 8'b0000_0001};

参考文献

[1] C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, D. Sanchez and T. B. Schardl, "There's plenty of room at the top: What will drive computer performance after moore ' s law," American Association for the Advancement of Science, vol. 368, no. 6495, 2020.

[2] https://news.mynavi.jp/article/20200324-1001886/.

[3] https://www.icinsights.com/news/bulletins/Transistor-Count-Trends-Continue-To-Track-With-Moores-Law/.

[4] https://kobaweb.ei.st.gunma-u.ac.jp/warehouse/2017-10-24nakatani.pdf.

[5] https://kobaweb.ei.st.gunma-u.ac.jp/lecture/20200121_nakatani.pdf.

[6] https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r02/html/nd252110.html.

[7] https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r01/html/nd111110.html.

[8] https://mobile.nuro.jp/mvnolab/articles/202012_87/.

[9] https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r02/html/nd252120.html.

[10] https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/html/nc111120.html.

[11] https://mmdlabo.jp/investigation/detail_1912.html.

[12] https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc131410.html.

[13] https://pc.watch.impress.co.jp/docs/column/kaigai/672590.html.

[14] R. Thabet, R. Mahmoudi and M. H. Bedoui, “Image processing on mobile devices: An overview,” International Image Processing, Applications and Systems Conference, pp. 1–8, 2014.

[15] L. Xiu, B. Ma, K. Zhu and L. Zhang, “Implementation and optimization of image acquisition with smartphones in computer vision,” 2018 International Conference on Information Networking, pp. 261–266, 2018.

[16] K. Kageyama, K. Sugiyama, T. Kumaki and T. Fujino, “1/f fluctuation-based visible light beacon for spy-photo prevention system,” RISP International workshop on Nonlinear Circuit, computer and Signal Processing, 2016.

[17] A. Ahmed and E. Ahmed, “A survey on mobile edge computing,” 2016 10th International Conference on Intelligent Systems and Control, 2016.

[18] https://pc.watch.impress.co.jp/docs/column/kaigai/1208397.html.

[19] https://pc.watch.impress.co.jp/docs/news/1281684.html.

[20] K. Uchiyama, “Processor technology in system lsi,” Journal of the Institute of Electronics, Information and Communication Engineers, vol. 95, no. 7, pp. 582–588, 2012.

[21] K. Uchiyama, “Power-efficient heteregoneous parallelism for digital convergence,” 2008 IEEE Symposium on VLSI Circuits, pp. 6–9, 2008.

[22] https://k-tai.watch.impress.co.jp/cda/article/keyword/9016.html.

[23] https://eetimes.itmedia.co.jp/ee/articles/2104/27/news034.html.

[24] https://news.mynavi.jp/article/20140325-smartphone_word/.

[25] https://www.4gamer.net/games/121/G012181/20180618051/.

[26] https://monoist.atmarkit.co.jp/mn/articles/1605/30/news116.html.

[27] https://k-tai.watch.impress.co.jp/docs/column/keyword/1156507.html/.

[28] https://eetimes.itmedia.co.jp/ee/articles/2011/24/news062.html.

[29] https://xtech.nikkei.com/atcl/nxt/column/18/00582/020500004/.

[30] https://www.sbbit.jp/article/cont1/36532.

[31] https://www.tjsys.co.jp/focuson/edge-ai-approach/index_j.htm.

[32] https://www.kodensha.jp/index/blog/2019/05/15/4077/.

[33] https://time-space.kddi.com/au-kddi/20210517/3110.

[34] https://iphone-mania.jp/news-381538/.

[35] https://www.kyoto-np.co.jp/articles/-/426780.

[36] https://www.jiji.com/jc/article?k=2021050400388&g=soc.

[37] https://www.sankei.com/article/20201120-RXVP5463WFNVZF6Q5IGJ5FQUTQ/.

[38] https://jp.techcrunch.com/2021/04/15/2021-04-14-deep-fake-video-app-avatarify-which-process-on-phone-plans-digital-watermark-for-videos/.

[39] https://mmdlabo.jp/investigation/detail_1958.html.

[40] https://mmdlabo.jp/investigation/detail_1930.html.

[41] https://jp.techcrunch.com/2021/07/27/petvoice/.

[42] M. K. Mandal, “Multimedia signals and systems,” Springer US, 2003.

[43] L. Vincent, “Morphological algorithms,” Mathematical Morphology in Image Processing, pp. 255–288, 1992.

[44] https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm.

[45] G. E. Blelloch and B. M. Maggs, “Morphological algorithms,” Algorithms and theory of computation handbook: special topics and techniques, 2010.

[46] https://www.tutorialspoint.com/parallel_algorithm/parallel_algorithm_introduction.htm.

[47] H. H. Yao and E. E. Swartzlander, “Serial-parallel multipliers,” Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, pp. 359–363, 1993.

[48] B. Chang, B. Goi, R. C.-W. Phan and W. Lee, “Accelerating multiple precision multiplication in gpu with kepler architecture,” 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems, pp. 844–851, 2016.

[49] N. H.-K. C. A. Navarro and L. Mateu, “A survey on parallel computing and its applications in data-parallel problems using gpu architectures,” Cambridge University Press, vol. 15, no. 2, pp. 285–329, 2014.

[50] K. M. Abughalieh and S. G. Alawneh, “A survey of parallel implementations for model predictive control,” IEEE Access, vol. 7, pp. 34348–34360, 2019.

[51] M. Hemnani, “Parallel processing techniques for high performance image processing applications,” 2016 IEEE Students’ Conference on Electrical, Electronics and Computer Science, 2016.

[52] Z. Jovanovic and V. M. Milutinovic, “Fpga accelerator for floating-point matrix multiplication,” IET Computers & Digital Techniques journal, 2012.

[53] L. Hung and Y. Chen, “Parallel table lookup for next generation internet,” 2008 32nd Annual IEEE International Computer Software and Applications Conference, pp. 52–59, 2008.

[54] https://www.tutorialspoint.com/parallel_algorithm/parallel_search_algorithm.htm.

[55] A. M. Fiskiran and R. B. Lee, “Fast parallel table lookups to accelerate symmetric-key cryptography,” International Conference on Information Technology: Coding and Computing, 2005.

[56] T. Kumaki, “Development of content addressable memory-based massive-parallel simd matrix core,” LSI and Systems Workshop 2017, pp. 52–59, 2017.

[57] K. Watanabe, A. Sekino, K. Kageyama, T. Koide and T. Kumakih, “Verification by Simulating Content Addressable Memory-based Massive-parallel SIMD Matrix Core,” LSI and Systems Workshop 2019, 2019.

[58] K. Kageyama, A. Sekino, K. Watanabe, A. Hamai, T. Koide and T. Kumaki, “Proposal of content addressable memory-based massive-parallel SIMD matrix core,” RISP International workshop on Nonlinear Circuit, computer and Signal Processing (NCSP), 2020.

[59] K. Kageyama, K. Watanabe, A. Hamai, T. Koide and T. Kumaki, “Acceleration of arithmetic processing with CAM-based massive-parallel SIMD matrix core,” IEEE International MidWest Symposium on Circuits And Systems (MWSCAS), 2020.

[60] K. Watanabe, K. Kageyama, A. Sekino, A. Hamai, and T. Kumaki, “Basic operation verification of content addressable memory-based massive-parallel SIMD matrix core for multimedia applications,” International symposium on biomedical engineering (ISBE), 2019.

[61] M. Nakajima, H. Noda, K. Dosaka, K. Nakata, M. Higashida, O. Yamamoto, K. Mizumoto, H. Kondo, Y. Shimazu, K. Arimoto, K. Saitoh and T. Shimizu, “A 40GOPS 250mW massively parallel processor based on matrix architecture,” 2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers, 2006.

[62] T. Kumaki, M. Ishizaki, T. Koide, H. J. Mattausch, Y. Kuroda, T. Gyohten, H. Noda, K. Dosaka, K. Arimoto and K. Saito, “Integration architecture of content addressable memory and massive-parallel memory-embedded SIMD matrix for versatile multimedia processor,” IEICE Trans. Electron., vol. E91-C, no. 9, pp. 1409–1418, 2008.

[63] http://www.kumikomi.net/article/news/2008/06/25_01.php.

[64] H. Noda, T. Tanizaki, T. Gyohten, K. Dosaka, M. Nakajima, K. Mizumoto, K. Yoshida, T. Iwao, T. Nishijima, Y. Okuno, and K. Arimoto, “The Circuits and Robust Design Methodology of the Massively Parallel Processor Based on the Matrix Architecture,” 2006 Symposium on VLSI Circuits Digest of Technical Papers, pp. 260–261, 2006.

[65] T. Kumaki, T. Koide, and T. Fujino, “Secure data processing with massive-parallel simd matrix for embedded soc in digital-convergence mobile devices,” IEEJ Transactions on Electrical and Electronic Engineering, vol. 12, no. 1, pp. 96–104, 2017.

[66] T. Kumaki, Y. Murakami, S. Itaya, K. Nakao, T. Ogura, and T. Fujino, “Max-plus algebra-based morphological wavelet transform watermarking for highly-parallel processing with mobile embedded processor,” Journal of Signal processing, vol. 16, no. 6, pp. 547–556, 2012.

[67] T. Kumaki, M. Ishizaki, T. Koide, H. J. Mattausch, Y. Kuroda, T. Gyohten, H. Noda, K. Dosaka, K. Arimoto and K. Saito, “Integration architecture of content addressable memory and massive-parallel memory-embedded SIMD matrix for versatile multimedia processor,” IEICE Transactions on Electronics, vol. E91-C, no. 9, pp. 1409–1418, 2008.

[68] T. Kumaki, T. Koide, H. J. Mattausch, M. Tagami, and M. Ishizaki, “Software-Based parallel cryptographic solution with massive-parallel memory-embedded SIMD matrix architecture for data-storage systems,” IEICE Transrations on Information & Systems, vol. E94-D, no. 9, pp. 1742–1754, 2011.

[69] T. Kumaki, M. Osawa, S. Itaya, T. Ogura, and T. Fujino, “Decomposition/Reconstruction acceleration of max-plus algebra-based morphological wavelet transform with massive-parallel SIMD matrix mobile processor,” Journal of Signal Processing, vol. 15, no. 6, pp. 425–434, 2011.

[70] Y. Mochizuki, N. Yoshida, N. Matsumoto, Y. Murakami, T. Kumaki, and T. Fujino, “Parallel Processing Implementation and Evaluation of Mersenne Twister with SIMD Embedded Processor,” The IEICE transactions on information and systems D, vol. J95-D, no. 3, pp. 376–386, 2012.

[71] T. Honda, Y. Mochizuki, T. Kumaki, and T. Fujino, “Implementation and Evaluation of Data-Parallelized CryptMT Stream Cipher with SIMD Embedded Processor,” The IEICE transactions on information and systems D, vol. J96-D, no. 3, pp. 495–505, 2013.

[72] T. Kumaki, T. Koide, and T. Fujino, “Secure data processing with massive-parallel SIMD matrix for embedded SoC in digital-convergence mobile devices,” IEEJ Transactions on Electrical and Electronic & Engineering, vol. 12, no. 1, pp. 96–104, 2017.

[73] T. Kumaki, M. Ishizaki, T. Koide, H. J. Mattausch, Y. Kuroda, H. Noda, K. Dosaka, K. Arimoto and K. Saito, “Acceleration of DCT processing with

massive-parallel memory-embedded SIMD matrix processor," IEICE Transrations on Information Systems, vol. E90-D, no. 8, pp. 1312–1315, 2007.

[74] H. Hiramoto, T. Kumaki, Y. Imai, T. Koide, H. J. Mattausch, "An Improved Face-Detection Method for a Massive-Parallel Memory-Embedded SIMD Matrix Processor MX-1," IEICE Technical Report, vol. 109, no. 336, pp. 83–88, 2009.

[75] V. R. Datti and P.V. Sridevi, "Performance evaluation of content addressable memories," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 596–598, 2018.

[76] T. Kohonen, "Content-addressable memories," Springer-Verlag Berlin Heidelberg, 1987.

[77] Shruthi G, Abhilash P and Vasunadara Patel K S, "Design of content addressable memory," 2018 International Conference on Networking, Embedded and Wireless Systems, 2018.

[78] X. Fan, A. Ghonem and T. Gemmeke, "Performance evaluation of content addressable memories," Content-Addressable Memory - Overview and Outlook of an Enabler for Modern Day Applications, ANALOG 2018; 16th GMM/ITG-Symposium, pp. 40–145, 2018.

[79] H. Riaz, A. A. Bhatti, M. A. Tahir and M. Sarwar, "High speed content addressable memory with reduced size and less power consumption," 2016 International Conference on Design and Technology of Integrated Systems in Nanoscale Era, 2016.

[80] A. Badawi, A. Alqarni, A. Aljuffri, M. S. BenSaleh, A. M. Obeid, and S. M. Qasim, "Fpga realization and performance evaluation of fixed-width modified baugh-wooley multiplier," 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering, pp. 155–158, 2015.

[81] M. Sjalander, and P. Larsson-Edefors, "High-speed and low-power multipliers using the baugh-wooley algorithm and hpm reduction tree," 2008 15th IEEE International Conference on Electronics, Circuits and Systems, pp. 33–36, 2008.

[82] A. Mukherjee, and A. Asati, “Generic modified baugh wooley multiplier,” 2013 International Conference on Circuits, Power and Computing Technologies, pp. 746–751, 2013.

[83] K. Kageyama, S. Arai, H. Hamano, A. Hamai, X. Kong, T. Koide, and T. Kumaki, “Implementation and evaluation of multiplication using content addressable memory-based massive-parallel SIMD matrix core,” Information Processing Society of Japan Research Report, 2021.

[84] K. Kageyama, S. Arai, H. Hamano, A. Hamai, X. Kong, T. Koide, and T. Kumaki, “Multiplication of Baugh-Wooley arithmetic processing by content addressable memory-based massive-parallel SIMD matrix core,” International symposium on biomedical engineering (ISBE), 2021.

[85] https://news.mynavi.jp/siryou_hikaku/20210225-1751087/.

[86] https://xtech.nikkei.com/atcl/nxt/keyword/18/00002/030800119/.

[87] Nikkei BP, “Raspberry pi magazine,” 2019.

[88] https://www.raspberrypi.org/products/raspberry-pi-4-model-b/.

[89] T. Honma, S. Mikami, “Analyze the texture with smartphone and easily judge the deliciousness of raw sea urchin,” Information Processing Hokkaido Symposium, pp. 221–226, 2014.

[90] C. Nakajima, N. Ito, “Object recognition and position-pose estimation by image processing,” IEEJ journal, vol. 121-C, no. 10, pp. 1516–1523, 2001.

[91] A. Asano, C. Asano, Y. Kimori, M. Muneyasu, H. Nobuhara, M. Fujio, “Non-linear image, signal processing,” Maruzen, pp. 43–67, 2010.

[92] P. Maragos, “Pattern spectrum and multiscale shape representation,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 701–716, 1989.

[93] Z. Rafii, T. Kumaki, T. Fujita, M. Nakanishi, and T. Ogura, “Implementation results and application of real-time morphological pattern spectrum analyzer on cellular automata hardware,” 2012 RISP Int ’ l Workshop on Nonlinear Circuits, Communications and Signal Processing, 2012.

[94] S. Baeg, A. T. Popov, V. G. Kamat, S. Batman, K. Sivakumar, N. Kehtarnavaz, E. R. Dougherty, and R. B. Shah, “Segmentation of mammograms into distinct morphological texture regions,” 11th IEEE Symposium on Computer-Based Medical Systems, pp. 1–6, 1998.

[95] K. Sudo, J. Yamato, A. Tomono, “Determining gender using morphological pattern spectrum,” IEICE transactions D, vol. J80-D2, no. 5, pp. 1037–1045, 1997.

[96] T. Kumaki, T. Fujita, M. Nakanishi, and T. Ogura, “Morphological pattern spectrum and block cipher processing based image-manipulation detection,” Special section on Recent Progress in Nonlinear Theory and Its Applications, vol. 4, pp. 400–418, 2013.

[97] E. C. Pedrino, and M. M. Fernandes, “Automatic generation of custom parallel processors for morphological image processing,” 2014 IEEE 26th International Symposium on Computer Architecture and High Performance Computing, pp. 176–181, 2014.

[98] https://jp.mathworks.com/help/images/morphological-dilation-and-erosion.html.

[99] P. Maragos, “Morphorogical?partii:their relations to median, order statistic, and stack filters,” IEEE Trans. Signal processing, vol. 35, no. 18, pp. 1170–1184, 1987.

[100] T. Tanizaki, T. Gyoten, H. Noda, M. Nakajima, K. Mizumoto, K. DOSAKA, “A Super parallel SIMD processor with Time/Space conversion Bus Bridge on the Matrix Architecture,” IEICE Technical Report, vol. 106, no. 206, pp. 1–6, 2006.

[101] https://www.renesas.com/jp/ja/software-tool/high-performance-embedded-workshop.

[102] S. Yoneda, “Establish the strongest pc with the new “beagleboard”,” Nikkei Linux, vol. 11, no. 7, pp. 55–68, 2009.

[103] https://developer.arm.com/architectures/instruction-sets/simd-isas/neon.

[104] https://japan.cnet.com/article/20083771/.

[105] https://ark.intel.com/content/www/us/en/ark/products/42503/intel-atom-processor-n450-512k-cache-1-66-ghz.html.

[106] Y. Mashiba, “Debate on the criminal regulation of voyeurism,” National Diet Library research and legislative examination stations, pp. 133–142, 2011.

[107] https://www.fox19.com/2018/11/06/is-it-getting-easier-be-voyeur-what-watch-keep-yourself-safe/.

[108] https://www.washingtonpost.com/technology/2019/04/06/they-were-settling-into-their-airbnb-then-they-found-hidden-camera/.

[109] https://www.channelnewsasia.com/news/singapore/singapore-voyeurism-problem-spy-cam-sex-harassment-monica-baey-11487244.

[110] http://news.bbc.co.uk/2/hi/asia-pacific/3031716.stm.

[111] https://www.japantimes.co.jp/opinion/2008/05/04/editorials/thou-shalt-not-steal-books/.

[112] https://www.procamera-app.com/en/blog/smartphone-camera-voyeurism-is-alive-and-clicking-shhhh/.

[113] https://this.kiji.is/719365978188627968?c=39546741839462401.

[114] https://english.kyodonews.net/news/2020/10/ad989aaa0751-joc-to-protect-female-athletes-targeted-by-exploitative-photos.html.

[115] https://privacy.wv.gov/tips/Pages/VideoVoyeurismPreventionAct.aspx.

[116] https://www.rcfp.org/congress-approves-criminal-video-voyeurism-law/.

[117] https://www.digitaltrends.com/mobile/how-to-turn-off-camera-shutter-sound-on-your-android-phone/.

[118] https://www.itmedia.co.jp/mobile/articles/0403/17/news0
01.html.

[119] https://id.misumi-ec.com/vona2/detail/222005567672/.

[120] T. Yamada, S. Gohshi, and I. Echizen, “Enhancement of method for preventing illegal recording of movies to enable it to detect cameras with attached infrared-cut filter,” 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1825–1828, 2012.

[121] T. Yamada, S. Gohshi, and I. Echizen, “Preventing unauthorized copying of displayed information by utilizing differences in spectral sensitivity between humans and imaging devices,” 2012 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 145–150, 2012.

[122] https://www.bbc.com/news/technology-36672001.

[123] https://www.patentlyapple.com/patently-apple/2011/06/
apple-working-on-a-sophisticated-infrared-system-for-ios-
cameras.html.

[124] https://www.canada.ca/en/conservation-institute/services/
agents-deterioration/light.html.

[125] K. Kageyama, T. Honda, K. Nakagawa, T. Kumaki, and T. Fujino, “Study of visible light beacon-based crime prevention system,” IEICE Technical Report, vol. 113, no. 497, pp. 301–306, 2014.

[126] K. Sugiyama, K. Kageyama, T. Kumaki, and T. Fujino, “Prevention peeping system by coordinating led light with smartphones,” IEICE Technical Report, vol. 114, no. 506, pp. 65–70, 2015.

[127] T. Kumaki, K. Kageyama, K. Sugiyama, and Takeshi Fujino, “Development of led-based spy-photo prevention system,” 2016 Symposium on Cryptography and Information Security, 2016.

[128] K. Kageyama, K. Sugiyama, T. Kumaki, and Takeshi Fujino, “Proposal of led-based peeping prevention system,” Proc. IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 616–619, 2015.

[129] K. Kageyama, K. Sugiyama, T. Kumaki, and T. Fujino, “Development of led illumination-based spy photo-prevention system,” IEEE Global Conference on Consumer Electronics (GCCE), pp. 129–130, 2015.

[130] K. Kageyama, K. Sugiyama, T. Kumaki, and T. Fujino, “1/f fluctuation-based visible light beacon for spy-photo prevention system,” RISP International workshop on Nonlinear Circuit, computer and Signal Processing (NCSP), 2016.

[131] http://www.ritsumei-seeds.jp/led.

[132] https://www.sankei.com/life/news/140903/lif1409030053-n1.html.

[133] M. Kamata, K. Takasaka, and C. Katano, “Led lighting control technologies for energy conservation,” Toshiba Review, vol. 65, no. 7, pp. 16–19, 2010.

[134] A. Suzuki, “Overview of lighting market in japan and standardisations,” The Institute of Electrical Installation Engineers of Japan, vol. 35, no. 1, pp. 10–13, 2015.

[135] https://www.grandviewresearch.com/industry-analysis/led-lighting-market.

[136] https://www.ledsmagazine.com/leds-ssl-design/packaged-leds/article/16695951/led-luminaire-growth-looms-over-the-lighting-landscape-magazine.

[137] https://www.ushio.co.jp/jp/technology/lightedge/201203/100436.html.

[138] https://news.panasonic.com/global/stories/2015/32005.html.

[139] M. Oshima, H. Aoyama, K. Nakanishi, and T. Maeda, “Image sensor-based visible light communication technology,” Panasonic Technical Journal, vol. 61, no. 2, pp. 118–123, 2015.

[140] H. Aoyama, and M. Oshima, “Line scan sampling for visible light communication: Theory and practice,” 2015 IEEE International Conference on Communications (ICC), pp. 5060–5065, 2015.

[141] L. E. M. Matheus, A. B. Vieira, L. F. M. Vieira, M. A. M. Vieira, and O. Gnawali, “Visible light communication Concepts, applications and challenges,” IEEE Communications Surveys & Tutorials, vol. 21, pp. 3204–3237, 2019.

[142] T. Sasaki, K. Kobayashi, H. Okada, and M. Katayama, “Data signal modulation scheme based on perceptually uniform color space for image sensor-based visible light communication,” 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS), 2019.

[143] T. Musha, “The fluctuation of the idea: I approach the mystery of the 1 / f fluctuation,” NHK Publishing, 1998.

[144] T. Musha, “1/f fluctuation of living body and information processing,” The Biophysical Society of Japan, vol. 27, no. 5, pp. 206–208, 1987.

[145] T. Musha, “The mystery of “1 / f fluctuation”,” Japan Society for Industrial and Applied Mathematics, vol. 11, no. 2, pp. 168–169, 2001.

[146] T. Irikura, “Vision and lighting,” Shokabo, 2014.

[147] https://hope.c.fun.ac.jp/mod/resource/view.php?id=12819&forceview=1.

[148] https://www.u-tokyo.ac.jp/focus/ja/features/f_00084.html.

[149] Japan Science and Technology Agency, “Forefront of biologging: An approach to wildlife behavior,” JSTnews, 2019.

[150] Y. Naito, “Elephant seal diving behavior and foraging,” Bio-logging Science Report, no. 83, pp. 2–5, 2013.

[151] http://www.nies.go.jp/biwakoi/methods.html.

[152] M. Yoshida, “Alien species fish channel catfish change buoyancy and swimming style according to the flow,” Bio-logging Science Report, no. 130, pp. 2–5, 2017.

[153] Y. Yonehara, Y. Goto, K. Yoda, Y. Watanuki, L. C. Young, H. Weimerskirch, C. Bost, and K. Sato, “Flight paths of seabirds soaring over the ocean surface enable measurement of fine-scale wind speed and direction,” Proceedings of the National Academy of Sciences of the United States of America (PNAS), vol. 113, no. 32, pp. 9039–9044, 2016.

[154] Y. Miyazawa, “Explore ocean currents with streaked shearwaters and freighters,” Blue Earth, vol. 149, pp. 21–28, 2017.

[155] T. Maezawa, Y. Kakuma, “Method for measurement of activity levels of domestic cats using accelerometer: verification of equivalence between one second and two seconds as recording intervals,” Teikyo university of Sicence Report, vol. 13, pp. 137–143, 2019.

[156] T. Yamamoto, K. Yoda, G. S. Blanco, and F.Quintana, “Female-biased stranding in magellanic penguins,” Current Biology Magazine, pp. R12–R13, 2019.

[157] T. Doi, A. Storto, T. Fukuoka, H. Suganuma, and K. Sato, “Impacts of temperature measurements from sea turtles on seasonal prediction around the arafura sea,” frontiers in Marine Science, vol. 6, pp. 1–11, 2019.

[158] Y. Okabe, “Ict in such places,” IEICE short report, vol. 26, pp. 102–115, 2013.

[159] “Bio-logging -knowing the behavior of sea creatures-,” FRA NEWS, vol. 57, 2018.

[160] https://www.honda.co.jp/outdoor/knowledge/adventure/picture-book/nihonishigame/.

[161] https://buna.info/article/1543/.

[162] https://www.nikkei.com/article/DGXLASDG02H0U_S5A101C1CR0000/.

[163] https://www.sankei.com/west/news/130522/wst1305220084-n1.html.

[164] H. Kato, “Global environment found in reptiles: protects organisms from introgression,” TIERRA plus, vol. 120, pp. 24–25, 2017.

[165] https://www.hitachi-hri.com/keyword/k117.html.

[166] https://www.nedo.go.jp/activities/ZZJP_100016.html.

[167] H. Nakamura, T. Nakada, S. Miwa, “Normaly off computing -expectations and challenges-,” Information Processing Society of Japan, vol. 54, no. 7, pp. 654–660, 2013.

[168] T. Honda, K. Nakagawa, T. Kumaki, M. Kimata, T. Fujino, “Development and evaluation of low-power image sensor node using intermittent operation,” 2013 by Information Processing Society of Japan and The Institute of Electronics, Information and Communication Engineers, vol. 12, no. 4, pp. 373–374, 2013.

[169] M. Endo, K. Suwa, “Power saving effect by intermittent operation technology in environmental monitoring system,” Information Media Center Journal, no. 7, pp. 46–51, 2006.

[170] T. Noda, “Development of vibration power generation logger,” Bio-logging Science Report, no. 104, p. 2, 2015.

[171] T. Noda, J. Okuyama, Y. Kawabata, H. Mitamura, and N. Arai, “Harvesting energy from the oscillation of aquatic animals: testing a vibration-powered generator for bio-logging data logger systems,” Journal of Advance Marine Science and Technology Society, vol. 20, pp. 37–43, 2014.

[172] Y. Kawabata, “A new data logger that records only high speed (speed of effect) movements,” Bio-logging Science Report, no. 140, p. 5, 2018.

[173] N. Nishiumi, A. Matsuo, R. Kawabe, N. Payne, C. Huveneers, Y. Watanabe, and Y. Kawabata, “A miniaturized threshold-triggered acceleration data-logger for recording burst movements of aquatic animals,” The Journal of Experimental Biology, pp. 1–6, 2018.

[174] T. Oi, R. Yamakawa, T. Kumaki, “Construction of normaliy-off logging system for get environmental information,” IEICE, p. 57, 2019.

[175] T. Oi, T. Kumaki, “Implementation and evaluation of data logger for low power biologging,” LSI and system workshop 2018, 2018.

[176] T. Oi, Y. Endo, T. Kumaki, “Implementation of npmally-off method according to observation mode for low power consumption,” Bi0-logging science Symposium, 2019.

[177] T. Oi, Y. Endo, T. Kumaki, “Examination of low power consumption by acceleration detection control of data logger,” IEICE, p. 50, 2020.

[178] T. Oi and T. Kumaki, “Implementation and performance evaluation of the low power consumption sensor logger for bio-logging based on normally-off operation,” Society of instrument and control engineers annual conference (SICE), pp. 1404–1407, 2019.

[179] http://www.asahi.com/edu/nie/tamate/kiji/TKY200710230346.html.

[180] T. Yabe, “Let’s take a look at the life of turtles,” nature conservation, no. 534, pp. 32–35, 2013.

[181] https://www5.city.kyoto.jp/zoo/enjoy/breeder-blog/diary/20170929-26621.html.

[182] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[183] https://www.raspberrypi.org/documentation/hardware/camera/.

[184] https://www.yuden.co.jp/jp/product/category/module/GYSFDMAXB.html.

[185] https://www.raspberrypi.org/products/sense-hat/.

[186] https://www.raspberrypi.org/documentation/hardware/sense-hat/images/Sense-HAT-V1_0.pdf.

[187] https://www.st.com/resource/ja/datasheet/lsm9ds1.pdf.

[188] http://wiki.sunfounder.cc/index.php?title=ADXL345_3-Axis_Digital_Acceleration_of_Gravity_Tilt_Module_(GY-291).

[189] https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf.

[190] http://solitonwave.shop/?pid=135615856.

[191] https://direct.sanwa.co.jp/ItemPage/700-BTL018BK.

[192] https://buna.info/article/1543/.

[193] http://isgs.kyushu-u.ac.jp/ISGSseminar/file/11-1suzuki.pdf.

[194] https://pepy.xsrv.jp/90343.

[195] https://pets-kojima.com/library/zukan_small/detail/id=24960.

[196] https://www.shizecon.net/award/detail.html?id=465.

[197] https://panasonic.jp/battery/comparison.html#charge.

[198] https://www.marutsu.co.jp/pc/i/836419/.

[199] https://www.ankerjapan.com/item/A1268.html.

[200] https://mechatrax.com/products/slee-pi/.

謝辞

本研究の遂行にあたり，終始御懇切な御指導と御鞭撻を賜りました立命館大学理工学部電子情報工学科　熊木武志教授，立命館大学理工学部電子情報工学科　藤野毅教授，立命館大学理工学部電子情報工学科　小倉武教授，立命館大学理工学部電子情報工学科　藤田智弘教授，広島大学ナノデバイス・バイオ融合科学研究所 小出哲士教授，に深く感謝の意を表します．また，本研究全般にあたり御指導並びに有益な御討論を頂きました立命館大学理工学部電子情報工学科　孔祥博助教授に深く感謝の意を表します．

研究に対する心構えなどを教えていただきました，泉知論教授，孟林准教授，田中亜実講師，小林正明氏，宮田悠治氏，熊木慧弥氏に心から感謝の意を表します．

日頃から御協力を頂き，お世話になった大井崇広氏，嶋田拓也氏，杉本司氏，関野輝氏，内藤匡志氏，中村浩司氏，名坂純哉氏，法橋渉氏，倉橋和也氏，出口貴大氏，有馬聖氏，下村優太郎氏，杉崎太綱氏，前山勝也氏，渡辺健介氏，高山柊人氏，馬場雄也氏，山川凌平氏，吉泉尊人氏，太田智仁氏，桐原瑠也氏，濱井彰光氏，モハマド・アナ・ビン・ノリザン氏，遠藤雪岳氏，大森大輔氏，後藤壮貴氏，中原祥吾氏，桝谷祥樹氏，安喰豪氏，安部泰雅氏，荒井聡太氏，井上裕介氏，朱岩氏，廣田祐基氏，丸谷佑氏，森武湧太氏，山岸樹氏，山本敦氏，井村菜々子氏，平山遼氏，秋山紗花氏，黒川嵩登氏，齋藤偲勇棋氏，濱野甫氏，林拓実氏，松村大世氏，山高颯太氏，に深く感謝します．

また，研究に関する事務手続き等で特にお世話になりました立命館大学理工学事務室の石井英理香氏，浜田郁子氏に感謝します．

最後になりましたが，日常生活から研究までいろいろ支えていただいた両親，兄弟，友人並びに職場の方々，研究対象であったニホンイシガメのスズに心から感謝の意を表します．

2021 年 12 月　蔭山　亨佑

発表論文リスト

【公表論文】

- Kyosuke Kageyama, Takeshi Kumaki, Takeshi Ogura, and Takeshi Fujino, “Digital image forensics using morphological pattern spectrum,” Journal of Signal Processing, Volume 19, No. 4, pp. 159–162, Jul., 2015.
- 蔭山享佑, 小出哲士, 熊木武志, “超並列 SIMD 型プロセッサコア MX-1 を利用したモルフォロジカルパターンスペクトラムの並列処理について,” 電気学会論文誌, Volume 139, No. 3, pp. 237–246, Mar., 2019.
- 蔭山享佑, 大井崇広, 熊木武志, “陸上変温動物を対象としたノーマリオフバイオロギングデータロガーの開発,” 電子情報通信学会論文誌, Volume J104-D, No. 4, pp. 415–426, Apr., 2021.

【国際会議 (関連論文)】

- Kyosuke Kageyama, Takeshi Kumaki and Takeshi Fujino, “Human-like image manipulation detection using morphological pattern spectrum,” RISP International workshop on Nonlinear Circuit, computer and Signal Processing (NCSP), pp. 591–594, Feb., 2015.
- Kyosuke Kageyama, Kohei Sugiyama, Takeshi Kumaki and Takeshi Fujino, “Proposal of LED-based peeping prevention system,” IEEE International MidWest Symposium on Circuits And Systems (MWSCAS), Aug., 2015.
- Kyosuke Kageyama, Kohei Sugiyama, Takeshi Kumaki and Takeshi Fujino, “Development of LED illumination-based spy photo-prevention system,” IEEE Global Conference on Consumer Electronics (GCCE), pp. 129–130, Oct., 2015.
- Kyosuke Kageyama, Kohei Sugiyama, Takeshi Kumaki and Takeshi Fujino, “1/f fluctuation-based visible light beacon for spy-photo prevention system,” RISP International workshop on Nonlinear Circuit, computer and Signal Processing (NCSP), pp. 686-689, Mar., 2016.
- Kyosuke Kageyama, Takeshi Kumaki and Tetsushi Koide, “Structuring element-counting approach for morphological pattern spectrum-based image manipulation detection,” International Symposium on Devices, Circuits and Systems (ISDCS), Mar., 2019.

- Kyosuke Kageyama, Akira Sekino, Kensuke Watanabe, Akimitsu Hamai, Tetsushi Koide and Takeshi Kumaki, “Proposal of content addressable memory-based massive-parallel SIMD atrix core,” RISP International workshop on Nonlinear Circuit, computer and Signal Processing (NCSP), pp. 77-80, Feb., 2020.
- Kyosuke Kageyama, Kensuke Watanabe, Akimitsu Hamai, Tetsushi Koide and Takeshi Kumaki, “Acceleration of arithmetic processing with CAM-based massive-parallel SIMD matrix core,” IEEE International MidWest Symposium on Circuits And Systems (MWSCAS), pp. 486-489, Aug., 2020.
- Kyosuke Kageyama, Akimitsu Hamai, Kensuke Watanabe, Tetsushi Koide and Takeshi Kumaki, “Floating-point arithmetic of content addressable memory-based massive-parallel SIMD matrix core,” RISP International workshop on Nonlinear Circuit, computer and Signal Processing (NCSP), pp. 274-277, Mar., 2021.

【国際会議 (共著)】

- Kohei Sugiyama, Kyosuke Kageyama, Takeshi Kumaki and Takeshi Fujino, “Information including flicker noise with LED lighting for preventing spy-photos,” RISP International workshop on Nonlinear Circuit, computer and Signal Processing (NCSP), pp. 682-685, Mar., 2016.
- Kensuke Watanabe, Kyosuke Kageyama, Akira Sekino, Akimitsu Hamai and Takeshi Kumaki, “Basic operation verification of content addressable memory-based massive-parallel SIMD matrix core for multimedia applications,” International symposium on biomedical engineering (ISBE), pp. 316-317, Nov., 2019.

【参考論文】

- Kyosuke Kageyama, Takeshi Kumaki, Takeshi Ogura and Takeshi Fujino, “Human-like image manipulation detection using morphological pattern spectrum, ” 2015 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP’15), pp. 31–38, 2015.
- Kyosuke Kageyama, Kohei Sugiyama, Takeshi Kumaki, and Takeshi Fujino, “1/f fluctuation-based visible light beacon for Spy-photo Prevention System,” 2016 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP’16), pp. 686–689, 2016.
- Kyosuke Kageyama, Tetsushi Koide and Takeshi Kumaki, “Structuring Element-counting Approach for Morphological Pattern Spectrum-based Image Manipulation Detection,” 2019 2nd International Symposium on Devices, Circuits and Systems (ISDCS), 2019.
- Kyosuke Kageyama, Kensuke Watanabe, Akimitsu Hamai, Tetsushi Koide and Takeshi Kumaki, “Acceleration of arithmetic processing with CAM-based massive-parallel SIMD matrix core,” 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 486–489, 2020.

- Kyosuke Kageyama, Akira Sekino, Kensuke Watanabe, Akimitsu Hamai, Tetsushi Koide and Takeshi Kumaki, “Proposal of content addressable memory-based massive-parallel SIMD matrix core,” 2020 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP’20), pp. 77–80, 2020.

【国内研究会等発表】

- 蔭山享佑, 本多隼也, 中川和歩, 熊木武志, 藤野毅, “可視光ビーコンを利用した犯罪防止システムの検討, ” 信学技報, Volume 113, No. DC2013-498, pp. 301–306, Mar., 2014.
- 蔭山享佑, 熊木武志, 藤野毅, “可視光ビーコンを利用した犯罪防止システムの実装及び評価, ” LSI とシステムのワークショップ 2014, May, 2014.
- 蔭山享佑，杉山幸平，熊木武志，藤野毅, “可視光信号を利用した犯罪防止システムにおける受信画像の処理について, ” 2014 年度画像符号化シンポジウム・2014 年度映像メディア処理シンポジウム, Nov., 2014.
- 蔭山享佑，杉山幸平，熊木武志，藤野毅, “可視光信号を用いた Android 端末における犯罪防止空間の構築及び実装（レシーバ）, ” 電気関係学会関西連合大会, Nov., 2014.
- 杉山幸平，蔭山享佑，熊木武志，藤野毅, “可視光通信を用いた Android 端末における犯罪防止空間の構築及び実装（トランスミッタ）, ” 電気関係学会関西連合大会, Nov., 2014.
- 杉山幸平, 蔭山享佑, 熊木武志, 藤野毅, “LED 照明を用いたスマー トフォンとの連携による犯罪防止空間の構築, ” 信学技報, Volume 114, No. 507, pp. 65–70, Mar., 2015.
- 熊木武志, 蔭山享佑, 杉山幸平, 藤野毅, 森有生, 佐々木文子, “LED 照明を利用したスマートフォン盗撮防止システムの開発, ” The 29th Symposium on Cryptography and Information Security (SCIS), pp. 1–8, Jan., 2016.
- 渡辺健介, 関野輝, 蔭山享佑, 小出哲士, 熊木武志, “連想メモリベース超並列 SIMD 型演算コアのシミュレーションによる動作検証について, ” LSI とシステムのワークショップ 2019, May., 2019.
- 蔭山享佑，関野輝，渡辺健介，濱井彰光，熊木武志, “超並列マルチメディア処理 CAM ベース SIMD 演算コア, ” VDEC デザイナーズフォーラム, Sep., 2019.
- 山本敦, 蔭山享佑, 熊木武志, “映像脈波抽出と感情検出技術の融合に関する検討, ” 2021 年電気学会全国大会, Mar., 2021.

【特許】

- 熊木武志, 藤野毅,　蔭山享佑, “撮像部搭載装置，機能制限 システム機能制限方法，及びコンピュータプログラム,” 日本特許, 特願 2014-045362, 2014 年 9 月 28 日出願.

副論文

(1) 超並列 SIMD 型演算プロセッサコア MX-1 を利用したモルフォロジカルパターンスペクトラムの並列処理について.
蔭山享佑, 小出哲士, 熊木武志
電気学会論文誌, **139** (3), 237–246 (2019).

(2) Floating-point arithmetic of content addressable memory-based massive-parallel SIMD matrix core.
Kyosuke Kageyama, Akimitsu Hamai, Kensuke Watanabe, Tetsushi Koide and Takeshi Kumaki
RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, 274–277 (2021).

(3) Digital image forensics using morphological pattern spectrum.
Kyosuke Kageyama, Takeshi Kumaki, Takeshi Ogura and Takeshi Fujino
Journal of Signal Processing, **19** (4), 159–162 (2015).

(4) 陸上変温動物を対象としたノーマリオフバイオロギングデータロガーの開発.
蔭山享佑, 大井崇広, 熊木武志
電子情報通信学会論文誌, **J104-D** (4), 415–426 (2021).

参考論文

(1) Proposal of content addressable memory-based massive-parallel SIMD matrix core.
Kyosuke Kageyama, Akira Sekino, Kensuke Watanabe, Akimitsu Hamai, Tetsushi Koide and Takeshi Kumaki
2020 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'20), 77–80 (2020).

(2) Acceleration of arithmetic processing with CAM-based massive-parallel SIMD matrix core.
Kyosuke Kageyama, Kensuke Watanabe, Akimitsu Hamai, Tetsushi Koide and Takeshi Kumaki
2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), 486–489 (2020).

(3) Human-like image manipulation detection using morphological pattern spectrum.
Kyosuke Kageyama, Takeshi Kumaki, Takeshi Ogura and Takeshi Fujino
2015 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'15), 591–594 (2015).

(4) Structuring Element-counting Approach for Morphological Pattern Spectrum-based Image Manipulation Detection.
Kyosuke Kageyama, Tetsushi Koide and Takeshi Kumaki,
2019 2nd International Symposium on Devices, Circuits and Systems (ISDCS) (2019).

(5) 1/f fluctuation-based visible light beacon for Spy-photo Prevention System.
Kyosuke Kageyama, Kohei Sugiyama, Takeshi Kumaki, and Takeshi Fujino,
2016 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP'16), 686–689 (2016).