

画像処理プロセッサの 基本を学ぶ

組み込み用途では「面積効率」と「並列度」が鍵に

中西 圭里 東芝 セミコンダクター社 半導体研究開発センター

組み込み機器では、画像処理に特化した専用プロセッサを用いるケースが増えている。専用ハードウェアアクセラレータを設計するよりも短期間でシステムを構築することができ、汎用プロセッサやDSPよりも高いチップ面積効率が得られるからだ。ただし、専用プロセッサの実力を最大限に発揮するには、画像処理の詳細とプロセッサのアーキテクチャについて、「並列度」の観点から深く理解しておく必要がある。

「画像処理」は多種多様

画像情報を扱う組み込み機器の種類が着々と増えている。(デジタル)テレビや、DVDレコーダ/プレーヤ、パソコン、カメラ付き携帯電話機、デジタルカメラ(DSC)、携帯型カムコーダー(DVC)、車載機器、携帯型ゲーム機など、数え上げれば切りがない(次ページの図1)。

ひと言で画像処理といっても、用途によってその内容はさまざまである(本稿では、画像認識も含め、画像情報に対する処理をまとめて画像処理と呼ぶことにする)。例えば、デジタルテレビやDVDレコーダ/プレーヤでは、MPEG-2、H.264などの動画像符号化/復号化処理(ビデオコーデック)が用いられている。よく知られているように、この種のビデオコーデックでは、16×16画素のブロック(マクロブロック)を中

心としたブロック単位で、動きベクトルの探索やDCT(離散コサイン変換)、量子化などの処理が行われる。

一方、デジタルカメラや、カメラ付き携帯電話機などでは、CCD/CMOSイメージセンサーから出力される生のデータ(RAWデータ)に対してISP(Image Signal Processing)と呼ばれる画像処理が行われる。ISPでは、レンズなど光学系の補正処理や、イメージセンサーのばらつきなどから生じる傷補正など、画素単位での処理が中心となる。

他方、車載機器で用いられる歩行者認識や、デジタルカメラなどで用いられる顔認識などの画像認識処理では、画像データ中の特徴(量)を抽出し、参照データと比較することにより、ある種の判断を下すという流れのことが多い。

ここで画像処理の例として紹介したビデオコーデック、ISP、画像認識は、後述する並列性の構成が大きく異なる。

画像処理プロセッサの 基本を学ぶ

組み込み用途では「面積効率」と「並列度」が鍵に

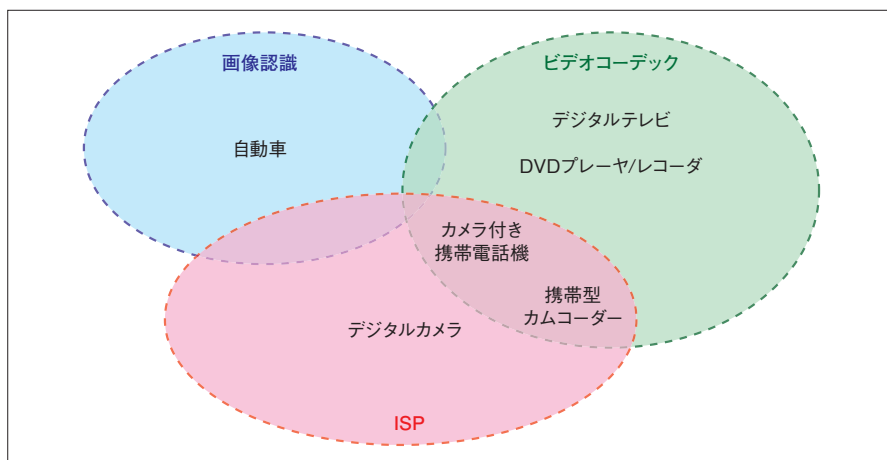


図1 画像処理技術を利用した製品の例
この図に示した以外にも、プリンタ、医療用機器、セキュリティ機器など多数の応用範囲がある。

言い換えれば、この点が画像処理の違いを特徴付ける大きなポイントとなっている。

専用プロセッサの特徴

画像処理を実現するハードウェアとしては、高いプログラマビリティを持つ汎用プロセッサから、高い性能を持つ専用ハードウェアアクセラレータまでさまざまなものがある。汎用プロセッサは高いプログラマビリティを持つので機能変更が容易である。しかし、単位面積/単位消費電力当たりの性能は専用ハードウェアアクセラレータと比べて大きく劣る。逆に、専用ハードウェアアクセラレータは機能変更が行えず、プログラマビリティが非常に低い。

このように、プログラマビリティと(単位面積/単位消費電力当たりの)性能は、トレードオフの関係にある(図2)。なぜなら、プログラマビリティと制御系回路の占める面積(消費電力)、そして性能と演算系回路の占める面積(消費電力)が、それぞれ比例関係にあるからである。

一方で、組み込み機器では、単位面積/単位消費電力当たりの性能について、コストの面での許容領域が存在する。さらに、組み込み画像処理技術が用いられる製品分野が広がる中、各製品分野では短期開発による多品種展開が求められている。そのため、画像処理を実現するハードウェアには、ある程度のプログラマビリティを持つことが強く求められている。すなわち、ソフトウェアの入れ替えにより機能を変更できるプロセッサを用いて、画像処理を実現することが必要なのだ。図2において、これらの要求を満たすことができるのは、プログラマブルアクセラレータの領域である。この領域に属するのが、本稿で焦点を当てる組み込み画像処理専用プロセッサである。

画像処理専用プロセッサは、さまざまな画像処理アプリケーションの中でも、ターゲットとするものに特化したアーキテクチャを備える。それにより、ターゲットアプリケーションで求められる処理において十分なプログラマビリティと高い性能を実現する。具体的に

は、まず必要十分な制御回路により、限られたターゲットアプリケーションにおけるプログラマビリティを保証する。その上で、多数の演算器をアプリケーションの性質に合わせて高い並列性で配置した演算回路構成により、単位面積/単位消費電力当たりの高い性能を実現させようというものである。

上述したように、画像処理専用プロセッサは限られたターゲットアプリケーションに特化したアーキテクチャを有する。そのため、ターゲットから外れた処理に対しては、たとえそれが画像処理の範疇のものであっても十分な性能が得られないことがある。すなわち、汎用プロセッサやDSPのように広い範囲をカバーすることは難しい。しかし、ここ数年、国内外を問わず半導体メーカー各社から、ある範囲の画像処理アプリケーションにターゲットを絞った画像処理専用プロセッサが発表されており^{*1)、*2)、*3)、*4)、*5)}、プロセッサの世界でも注目を浴びる分野の1つになっている。参考文献1～5からわかるように、ISSCC (International Solid State Circuits Conference)においても、ほぼ毎年新しい画像処理専用プロセッサが発表されている。

図3は、ISSCCを中心とした学会で発表されている最近の画像処理専用プロセッサの動作周波数とピーク性能を表したものである。この図において、縦軸のピーク性能を横軸の動作周波数で割ると「並列度」が得られる。図3では、この並列度が1000、500、200に相当するところを点線で表している。この図から、画像処理専用プロセッサは、総じて500～1000程度の非常に高い並列度を有することがわかる。

画像処理専用プロセッサの実力を最

*1) M. Nakajima, et al., "A 40GOPS 250mW Massively Parallel Processor based on Matrix Architecture", ISSCC Dig. Tech. Papers, Session 22.5, Feb, 2006

*2) S. Arakawa et al., "A 512GOPS Fully Programmable Digital Image Processor with Full HD 1080p Processing Capabilities", ISSCC Dig. Tech. Papers, pp.312～313, Feb, 2008

*3) S. Nomura, et al., "A 9.7mW AAC-Decoding, 620mW H.264 720p 60fps Decoding, 8-Core Media Processor with Embedded Forward-Body-Biasing and Power-Gating Circuit in 65nm CMOS technology", ISSCC Dig. Tech. Papers, pp.12～13, Feb, 2008

大限に引き出すには、この非常に高い並列度を使いこなすことがポイントとなる。実現したい画像処理アプリケーションにマッチする最適な画像処理専用プロセッサを選択して使いこなすためには、ターゲットとなるアプリケーションの並列性と画像処理専用プロセッサの並列性の両方について理解しておく必要がある。そこで、次節では画像処理アプリケーションにおける並列性について概観することにする。

3つの並列性

ここでは、まず画像処理に限らず、処理一般における3つの並列性について説明する。一般に処理の並列性としては以下の3種類がある。

- データ並列性
- 命令並列性
- スレッド並列性

以下では、それぞれについて順に説明していく。

■ データ並列性

データ並列性とは、複数のデータに対して同時に同一の演算が行える性質のことを指す。例えば、横が100画素、縦が100画素の1枚の画像データに対して一律に定数Aを乗算する処理を考える。この場合、次の①と②とで処理結果は変わらない。

- ① 1つの画素データに対して定数Aを乗じる命令を、1つずつ1万回実行する
- ② 横方向の100個の画素データに対し、まとめて一律に定数Aを乗ずる命令を100回実行する

このように、複数のデータ（この例では横方向の100個の画素データ）をまとめて（並列に）、1度に同一の演算を施すことができる性質がデータ並列性である（次ページの図4）。

データ並列性がある場合、②のように複数のデータ（横方向100個のデータ）を1つのベクトルデータとし、それに対して一律の演算を実行したと考えることができる。そのため、②の演算をベクトル演算と呼ぶことがある。あ

るいは、1命令で複数のデータに対して演算を行うことから、SIMD (Single Instruction Multiple Data) 演算とも呼ばれる。②の例では、データ並列性による並列度は100である。乗算器を100個横に並べて1命令で1度に100

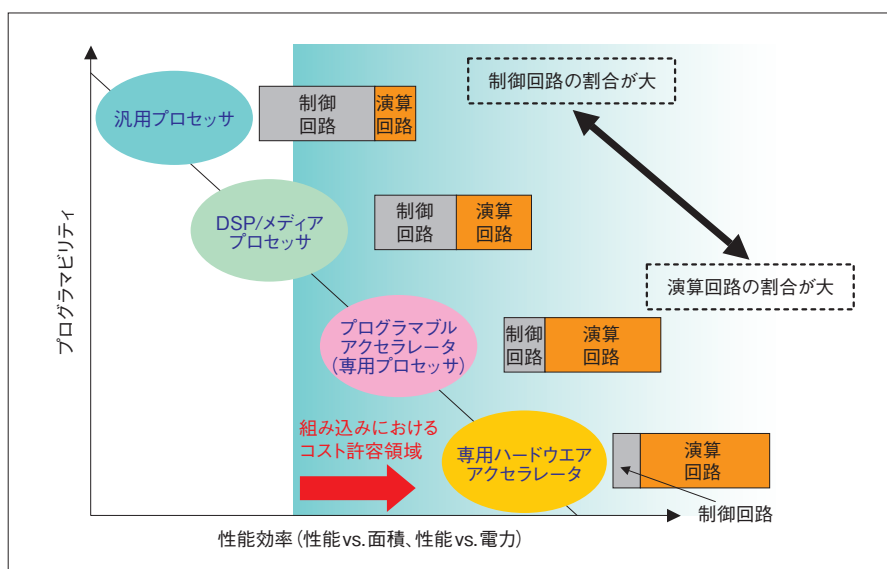


図2 プログラマビリティと性能の関係

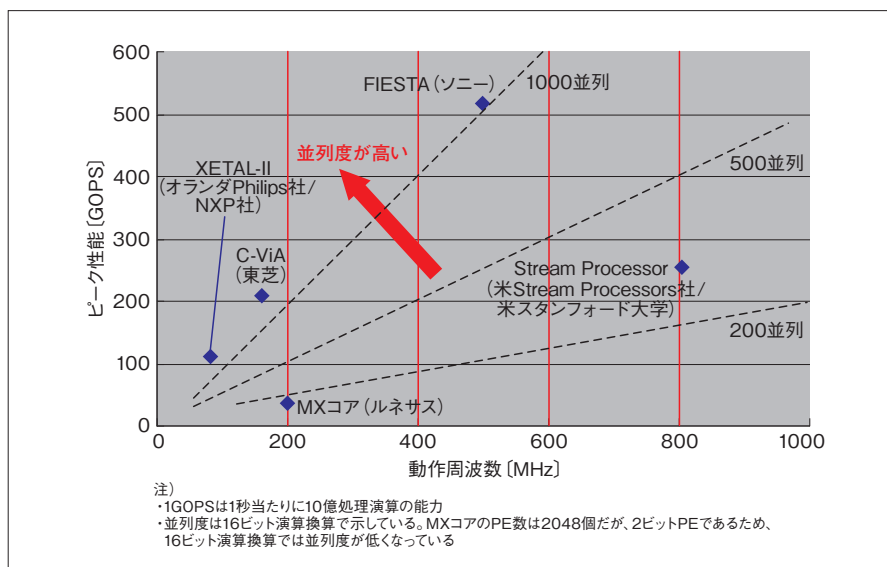


図3 専用プロセッサの動作周波数/ピーク性能

*4) K. Nakanishi, et al., "A 128-Parallel SIMD Image Signal Coprocessor with High Area Efficiency", COOL Chips XII, pp.377 ~ 379, April, 2009

*5) S. Kyo, et al., "IMAPCAR2: A Dynamic SIMD/MIMD Mode Switching Processor for Embedded Systems", HOT CHIPS 21, August, 2009

画像処理プロセッサの 基本を学ぶ

組み込み用途では「面積効率」と「並列度」が鍵に

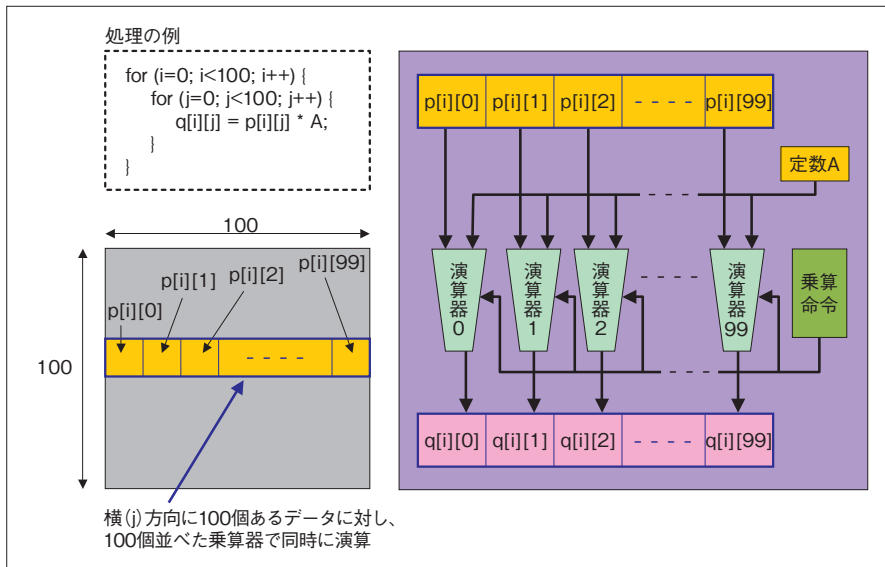


図4 データ並列性

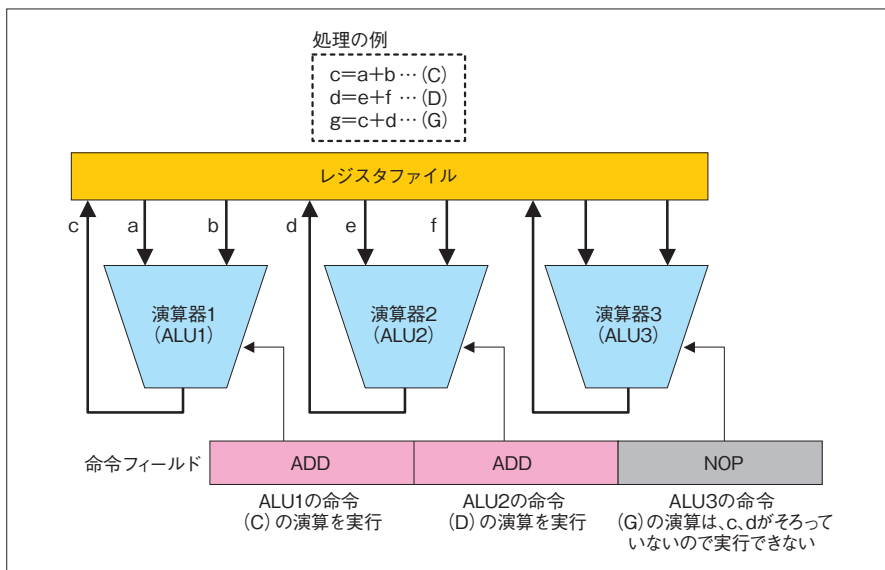


図5 命令並列性

個の乗算を行うので、命令数は①と比べて1/100となる。

画像処理アプリケーションはデータ並列性を持つものが多い。また多くの場合、その並列度が高い。この種の代表的な処理としては、エッジ抽出、エッ

ジ強調、低域通過フィルタなどの線形フィルタ群や、色空間変換などがある。

一方、データ並列性を持たない処理としては、1枚の画像の連結領域に対するラベリング処理などがある。連結領域のラベリング処理の場合、横方向/

縦方向ともに画素データの依存性があるため、ベクトル処理は行えない。

■命令並列性

命令並列性とは、複数の命令を同時に実行できる性質のことである。例として、図5に示した構成のプロセッサで、「処理の例」に示した3つの加算を処理するケースを考える。3つの加算のうち、図5のような構成のプロセッサでは、(C)と(D)の加算は同時に実行できる。この場合、(C)と(D)は命令並列性を持つと言う。それに対し、加算(G)は、(C)と(D)の結果を得てからでなければ実行できない。このように、(G)と(C)と(D)の3演算の組は命令並列性を持たない。ただし、命令並列性は名前のおり、プロセッサがどのような命令セットを持つかによって決まるものである。そのため、プロセッサの構成によっては、(G)と(C)と(D)が命令並列性を持つことがある(詳細は後述)。このように、命令並列性は処理内容とプロセッサの命令セットのペアで考えなければならないので注意が必要だ。

命令並列性を実現する方式としては、同時実行する複数の命令を動的にハードウェアが選んで実行する方式と、コンパイラが静的に解析して選ぶ方式の2通りがある。前者はスーパースカラ型、後者はVLIW (Very Long Instruction Word)型と呼ばれている。

スーパースカラ型の利点は、ハードウェアを拡張する際にオブジェクトコード(バイナリとなった命令列)に互換性があることだ。これは、ユーザー(ソフトウェア開発者)にとっては非常に大きなメリットとなる。ただし、スーパースカラ型には、同時に実行できる命令を管理するための制御系回路の面積が大きくなるという欠点がある。

一方、VLIW型では、ハードウェアを拡張するとオブジェクトコードの互換性がなくなってしまう。すなわち、ソースコードを書き換えたり、再コンパイルしたりする必要が生じ、ユーザーにとってはソフトウェア資産の再利用性が下がることになる。ただし、VLIW型では同時実行命令を管理するハードウェアが不要なので、制御系回路の面積と消費電力は少なく済む。そのため、コスト重視の組み込み用途では、VLIW型のプロセッサが多用される。

■スレッド並列性

スレッド並列性とは、複数のプログラム、関数など、ある大きさの処理単位(スレッド)のレベルで同時に処理できる状態のことを言う。例として、フレーム中のブロックを入出力としてラスタスキャン順に進む、関数A、Bから成る処理を考える(図6)。この場合、入力フレーム中のブロックnに対する関数Aの処理と、中間フレーム中のブロックnに対する関数Bの処理は同時には実行できない。しかし、入力フレーム中のブロックn+1に対する関数Aの処理と中間フレーム中のブロックnに対する関数Bの処理であれば、同時に並列実行することができる。このような状況がスレッド並列性を持つ例である。

このように、スレッドはある程度の大きさの命令列とデータを持つ。そのため、スレッドとハードウェアの対応としては、1つのスレッドに対して1つのプロセッサコアを与えることが多い。すなわち、スレッドを並列処理する方式としては、マルチコアプロセッサを利用して複数のプロセッサコアを同時に動作させるという手法がとられる。マルチコアプロセッサでは、各コアで複数のデータが複数の命令によって処理され

るため、MIMD (Multiple Instruction Multiple Data) 処理の一形態であると考えることができる。マルチコアプロセッサでスレッド並列性を利用して効率良く処理を行う上でポイントとなるのは、スレッドの分割の仕方と各プロセッサコアへのスレッドの分配の仕方である。一連の処理をいくつかのスレッドに分割する作業は、現状ではコンパイラなどのツールではなく、手作業で行うことが多い。分割後のスレッドの各コアへの分配は、全体制御用のプロセッサが各スレッドの依存関係を分析した上で各プロセッサコアに割り当てることなどにより行われる。

各種画像処理の並列性

前節では、処理一般における3つの並列性について説明した。どのアプリケーションでもこれら3つの並列性は多少なりとも存在するが、アプリケーションごとにどの並列性を持つことを特徴とするかは異なる。ここでは、各種画像処理がどのような並列性を持つかを説明する。

■ISP

ISPにおいて顕著なのはデータ並列

性である。画素ごとに依存を持たない処理が非常に多い。前節で紹介したフィルタ系処理が、そうした処理の代表的な例だ。また、データ並列幅(SIMD幅)に関しても制限がない処理が多い。それらの処理は、原理的には1フレームの画素すべてをデータ並列に扱うことも可能である(ただし、ハードウェア量の制約から、実際にはデータ並列幅は制限される)。従って、ISPおよびそれに類する処理をターゲットとしたプロセッサでは、データ並列幅が100を超えるものも多い。

ISPにも命令並列性やスレッド並列性は存在する。だが、ほかと比較してデータ並列性が非常に高いと言える。

■ビデオコーデック

ビデオコーデックで特徴的なのは、スレッド並列性である。先述したように、MPEG-2やH.264では、例えば16×16画素サイズのマクロブロックや、(同じ符号化タイプの)複数のマクロブロックから成るスライス処理単位とする。マクロブロック間には依存関係があるため、スライス単位で処理ごとにスレッド並列処理されることが多い。

一方、MPEG-2での処理の最小単位

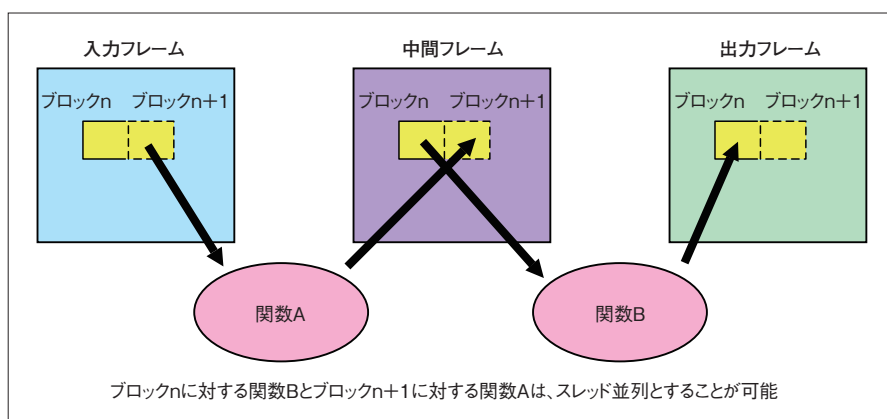


図6 スレッド並列性

画像処理プロセッサの 基本を学ぶ

組み込み用途では「面積効率」と「並列度」が鍵に

は8×8ブロック、H.264での最小処理単位は4×4ブロックである。そのため、ISPのように大きなデータ並列度はない。

なおH.264には、デブロッキングフィルタと呼ばれる処理量の多いフィルタがある。同フィルタの場合、ISPで用いられるフィルタよりも細かい制御を要し、データ依存性があるので、フィルタとはいえどもデータ並列性で性能を稼ぐのは難しい。そのため、スレッド並列と命令並列により性能を稼ぐことが必要となる。

■画像認識

画像認識の処理では、データ並列性、命令並列性、スレッド並列性がバランス良く用いられる。画像認識で行っている処理は、次の3つに分類される^{*6)}。

- 画像前処理/特徴抽出
- 識別処理/統計量計算
- 推論/判断

1つ目の画像前処理/特徴抽出には、各種フィルタ処理やノイズ除去など、ISP処理と類似した処理が多く含まれる。演算量は多いものの、高いデータ並列性を利用できる。

2つ目の識別処理/統計量計算については、データ並列性はあるものの、処理対象領域とデータ並列幅が可変であることが多い。そのため、大きな固定幅でのデータ並列処理は難しい。そこで、短いデータ並列幅によるデータ並列性と命令並列性を組み合わせた並列処理、または短いデータ並列性、命令並列性にスレッド並列性を組み合わせた並列処理を行うのが有効である。

3つ目の推論/判断については、演算量は少ないものの並列化が難しい。この部分は(信号処理に用いるCPUコアとは別の)制御用CPUコアなどで処理を行うことが多い。

最新の専用プロセッサ

ここまで、画像処理における並列性について述べてきた。実際のアプリケーションでは、こうした並列性の特徴を理解して実装を行う必要がある。また、市販の画像処理専用プロセッサには、そのアーキテクチャに依存して、いずれも各種の処理に対する得手不得手がある。すなわち、自分が開発しているアプリケーションに適したプロセッサを選ぶことが重要だ。そこで続いては、ここ数年に発表された各社の画像処理専用プロセッサが、アプリケーションに対応してどのようなアーキテクチャを持っているのかを概観する。

表1は、国内の半導体メーカー各社から最近発表された画像処理専用プロセッサについてまとめたものである。この表では、特に並列構造を中心にアーキテクチャ上の特徴を示している。

■ISP向けのプロセッサ

イメージセンサー、DSC、DVCなどのISPに対する適用が報告されているプロセッサとして、「MXコア」(ルネサス テクノロジー)、「C-ViA」(東芝)、「FIESTA」(ソニー)がある。

ルネサスが2006年に発表したMXコアは、2048個の2ビットPE(Processing Element)をSIMD並列に配置したSIMDアレイプロセッサコアである。先に示した図3では、並列度を16ビット演算で換算した並列度としているため、2ビット構成のPEを持つMXコアの並列度は低くなっている点に注意されたい。2048個のPE間の接続関係としては、一定の距離にあるPE間で同時にデータ転送が可能となっており、FFT(高速フーリエ変換)のパタフライ演算などを効率良く行えるようになっている。

東芝が2009年に発表したC-ViAは、SIMD/VLIW型のプロセッサコアである。カスケード接続された10個の演算器から構成されるVLIW型PEを128個、SIMD並列に配置している。

2008年にソニーが発表したFIESTAは、4コアのマルチコアプロセッサである。各コアは1命令で4つの算術演算を実行できるVLIW型PEを64個、SIMD並列に配置したものとなっている。FIESTAには、SIMD並列化すると面積が巨大化するLUT(Look Up Table)や除算器などの専用回路を、各PEで共有する枠組みが用意されている。

ISPをターゲットとするこれらのプロセッサに共通するのは、高いSIMD並列度(64~2048)を有することである。これは、ISPの高いデータ並列性に対応している。

■ビデオコーデック向けのプロセッサ

ビデオコーデックは、専用ハードウェアアクセラレータで処理することが多い。ただこの分野にも、ビデオコーデックをソフトウェアで実現するプロセッサがある。代表的な例が東芝の「Venezia-EX」である。これは、信号処理を行うMPE(Media Processing Engine)を8個備えるマルチコアプロセッサだ。各MPEは、東芝のコンフィギュラブルプロセッサ「MeP(Media Embedded Processor)」と64ビットのSIMDパイプラインを2本備えるVLIW型コプロセッサ「IVC2」から構成されている。ビデオコーデックで特徴的なのはスレッド並列性であるが、VeneziaではV-Kernelというカーネルにより、8コアを使ったスレッド並列性を実現している。

このように、ビデオコーデック向けプロセッサはスレッド並列性に対応したアーキテクチャを採用している。

^{*6)}枝廣・黒田, "組み込みプロセッサ技術(組み込みシステム基礎技術全集)", CQ出版, 2009

■画像認識向けプロセッサ

画像認識については、代表的なものとしてNECエレクトロニクスの車載向けプロセッサ「IMPCARシリーズ」が挙げられる。2009年夏に発表された「IMPCAR2」が最新版である。

IMPCAR2は16ビットの5ウェイVLIW型PEを128個、SIMD並列に配置したりニアプロセッサアレイ(LPA)に、制御用CPUを加えた構成となっている。IMPCAR2で新たに加わった特徴として、4個のPEを1つのコアとし、MIMD処理を行うモードが追加されたことが挙げられる。これにより、画像認識処理におけるデータ並列性、命令並列性、スレッド並列性をよりバランス良く利用することができる。また、先述したように、画像認識の前処理に関しては高いデータ並列性があるため、ISPで用いられる高いSIMD並列性を

持つプロセッサコアとの親和性が高いことも注目すべき点である。

専用プロセッサの具体例

ここまで、各種画像処理の特徴と、それに対応するプロセッサについて概観してきた。最後に、組み込み向け画像処理専用プロセッサの具体例として、筆者らが開発したISP向けプロセッサC-ViAのアーキテクチャを紹介する。これを通して、ISP向けプロセッサは、アーキテクチャにどのような特徴を持つべきなのかをご確認いただきたい。

その上で、ソフトウェア開発環境にも触れることにする。ここでソフトウェア開発環境を取り上げるのは、それがプロセッサに処理を実装する際の作業効率、作業工数に大きな影響を与えるからだ。すなわち、ソフトウェア開発環境は、実際のSoC(System on Chip)開

発において非常に重要な要素となる。

■C-ViAのアーキテクチャ

前節でも紹介したとおり、C-ViAはSIMD/VLIW型プロセッサコアであり、10個のカスケード接続された演算器から構成されるVLIW型PEが128個、並列に配置されている。C-ViAのPEの構成は、次ページの図7に示したようになっている。

C-ViAはPE内にパイプラインとして3ステージに分割された10個の演算器を備え、それら10個の演算器を1命令で制御するVLIW型プロセッサである。命令並列性の説明で、図5の(G)と(C)と(D)がプロセッサによっては命令並列性を持つ場合があると述べたが、C-ViAがその1つの例だ。実際、図7のPEにおいて、EX0ステージのALUツリーで3つの加算を1命令中で指定することにより同時に実行できる。PE

表1 組み込み向け画像処理専用プロセッサの比較

プロセッサ	応用分野	並列構造			動作周波数	チップ面積	形態	特徴
		データ並列性	命令並列性	スレッド並列性				
MXコア (ルネサス)	CCDセンサー向けISPほか、画像処理一般	2048個の2ビットPEをSIMD並列化	—	—	200MHz	3.1mm ² (90nmプロセス)	プロセッサコアIP。 MXコアに、 ホストCPUとして 「SH-2A」を 集積したチップ 「MX-G」もあり	<ul style="list-style-type: none"> ・2ビットPEで分散演算(Distributed Arithmetic) ・PEごとに512ビットのSRAMがスイッチングネットワークで接続されている
Venezia-EX (東芝)	モバイル向けビデオコーデック、画像認識系	64ビットSIMD演算器	3ウェイVLIW型MPE	8コア	333/ 166MHz	25.6mm ² (65nmプロセス)	チップ、もしくはマルチコアプロセッサIP	<ul style="list-style-type: none"> ・64ビットSIMD ALUはビット幅によりSIMD数が可変 ・V-Kernelによりスレッド並列動作をサポート
FIESTA (ソニー)	DSC/DVC向けISP	64個のSIMD PE	1命令4演算のVLIW型	4コア	250/ 500MHz	152.8mm ² (65nmプロセス)	チップ	<ul style="list-style-type: none"> ・スレッド管理ユニットにより、マイクロコントローラなしでも自律動作可能 ・LUTなどの特殊演算器は64個のPE間で共有可能
C-ViA (東芝)	CMOS/CCDセンサー、DSC/DVC向けISP	128個のSIMD PE	1命令10演算のVLIW型	—	160MHz	9.5mm ² (65nmプロセス)	プロセッサコアIP	<ul style="list-style-type: none"> ・非対称ALUをカスケード接続 ・ISPでのメモリー使用パターンに特化したメモリーコントローラ
IMPCAR2 (NECエレクトロニクス)	車載用画像認識	128個のSIMD PE (SIMDモード)	5ウェイVLIW型	32コア (MIMDモード)	108MHz	100mm ² (90nmプロセス)	チップ	<ul style="list-style-type: none"> ・4SIMD PEを1コアとして動作するMIMDモードあり ・必要なメモリーアクセスパターンを抽出してサポート

画像処理プロセッサの 基本を学ぶ

組み込み用途では「面積効率」と「並列度」が鍵に

内における10個の演算器の構成は、ターゲットとする処理であるISPにおける演算の種類比率、頻出する演算パターンを解析し、ISP向けに最適化したものだ。その結果、図8に示すようにISPの処理で頻出するフィルタ系演

算パターンや分岐系の処理パターンでも、効率良く1命令で実行することが可能となっている。

図9は、典型的な一連のISP処理における単位面積当たりの性能について表したものである。前節で説明した

Veneziaの信号処理プロセッサコアであるMPEとC-ViAとを比較している。グラフの縦軸は各プロセッサの(65nmのCMOSプロセスにおける)単位面積当たりのMOPS値(1MOPSは1秒当たり100万処理演算の能力)だ。コスト面で要求の厳しい組み込みプロセッサでは、コストパフォーマンスが重要である。そのため、ここでは単位面積当たりの性能を比較している。

例えばゲインコントロールの処理では、C-ViAがMPEの4倍の数値を示している。このことは、同処理において、C-ViAの単位面積当たりで1秒間に処理する画素数がMPEの4倍であることを意味している。言い方を変えると、同一の性能を実現するには、MPEを使った場合、C-ViAを使う場合に比べて4倍のチップ面積が必要になるといふことだ。ターゲットとした典型的なISPの各処理において、単位面積当たりで、C-ViAはMPEの4~13倍、処理全体では約10倍の性能を実現している。これは、ISPに最適化されたC-ViAのPE構成によるところが大きい。

なお、C-ViAの特徴として、PE構成以外に、ISPに特化したメモリの使い方をサポートする専用メモリーコントローラを備えていることも挙げられる。この専用メモリーコントローラにより、使用するメモリー量を容易に削減することができる。それだけでなく、複雑なアドレス計算も性能を落とすことなく実現することが可能となっている。

■ C-ViA Cコンパイラ

C-ViAのソフトウェア開発環境は、「C-ViA Cコンパイラ」、GUI(Graphical User Interface)デバッガ、ソフトウェアシミュレータから構成されている。ここでは、その中でも特徴的なC-ViA C

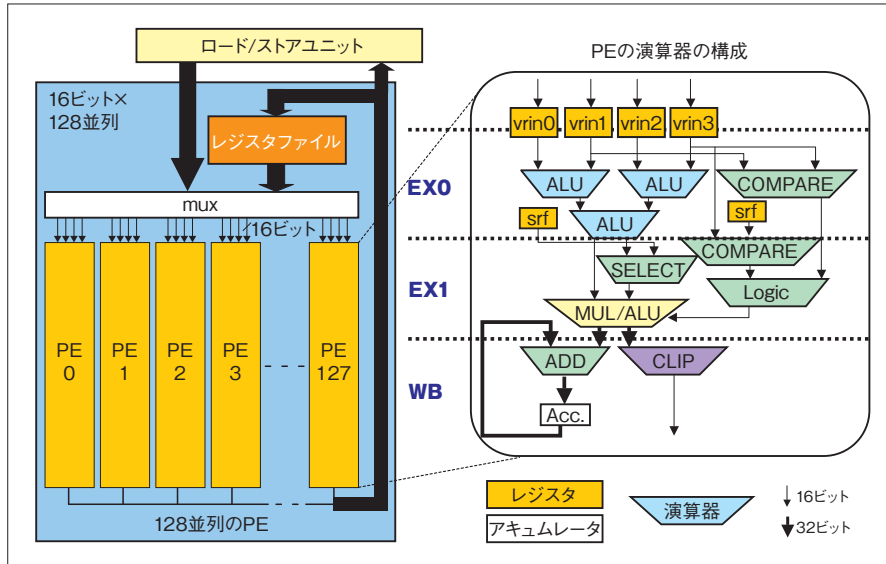


図7 C-ViAのPEの構成

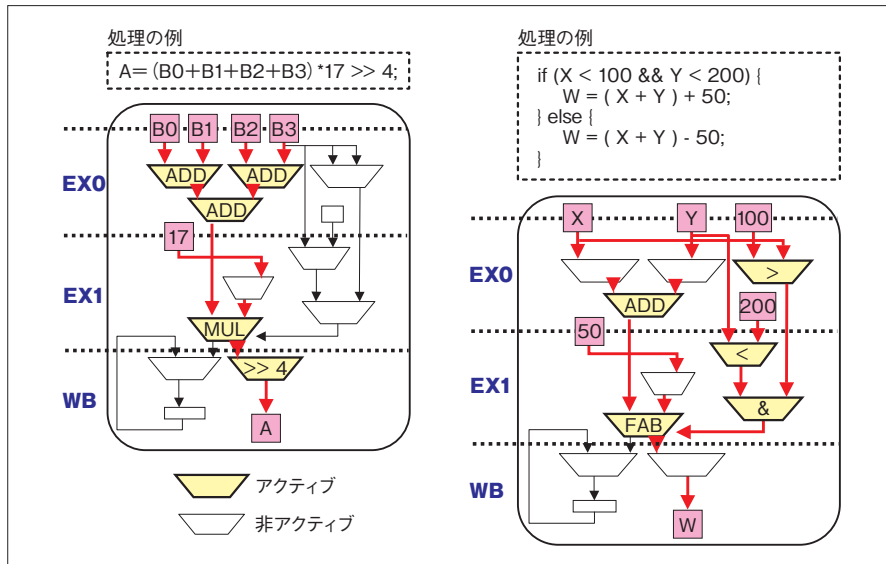


図8 PEへの処理のマッピング

コンパイラについて説明を行う。同コンパイラは、C-ViA用に拡張したC言語(C-ViA C言語)で記述されたソースコード(リスト1)を入力とし、C-ViAの多数の演算器への命令列を出力する。

先に、「画像処理専用プロセッサの実力を最大限引き出すためには、高い並列度を使いこなすことがポイントになる」と述べた。例えば、C-ViAでは10個の演算器が非対称に接続されており(図7)、より多くの演算器を並列動作させるためには、10個の演算器の組み合わせという非常に大きな探索空間の中から、最も効率的に演算器を稼働させることが可能な命令列を探索しなければならない。しかし、探索範囲が非常に大きいため、従来のコンパイラ技術では最適な命令列を現実的な時間内に生成することが難しく、コンパイル後に、部分的に人手によってアセンブリ言語によるコードの最適化を行ったりする必要があった。このような最適化手法は、大きな探索空間の中から人手で最適解を探し出していることと本質的に等価であるため、多大な労力を要すると同時に難易度の高い作業でもある。

C-ViA Cコンパイラでは、このような広い探索空間を効率的に探索する工夫を施すことにより、演算効率の良い、ほぼ最適解に近い命令列を現実的な時間内に探索することができる。実際、ソーベルフィルタやガウシアンフィルタなどの各種フィルタ、SAD(Sum of Absolute Differences)演算、画素補間処理など、40種類程度の基本的な画像処理について、人手でアセンブラを最適化したものに匹敵する性能のコードを出力することができた。このように、C-ViA Cコンパイラを使えば、従来のような人手による作業を行わなくて済むので、高い並列度

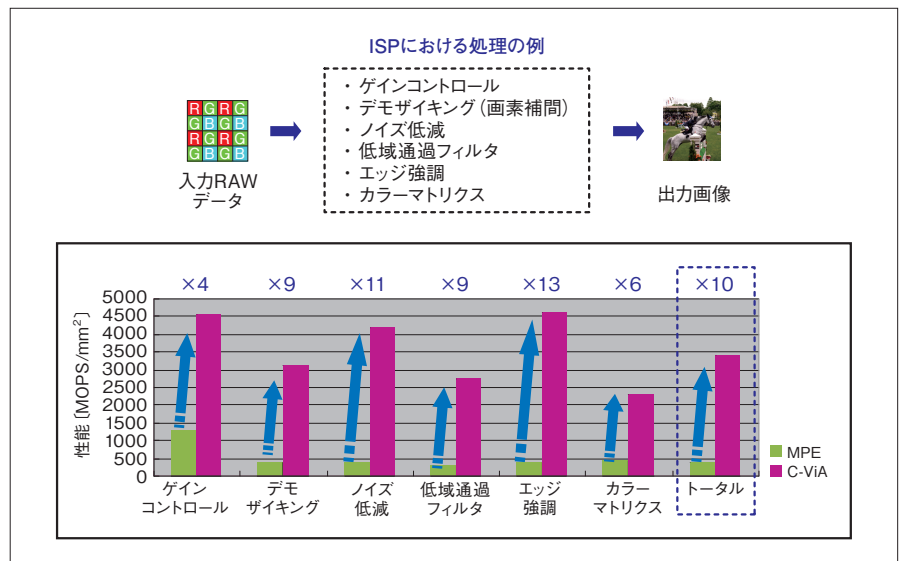


図9 ISP処理におけるC-ViAとMPEの性能比較

```

/* C-ViA C言語による処理の記述例(一部を抜粋) */
for (i=0; i<FRAME_HEIGHT; i++) {
  for (j=0; j<FRAME_WIDTH; j++) {
    ... (略) ...
    delta_x = (b[-1][1] - b[-1][-1]) + (b[0][1] - b[0][-1]) * 2
              + (b[1][1] - b[1][-1]);
    delta_y = (b[1][-1] - b[-1][-1]) + (b[1][0] - b[-1][0]) * 2
              + (b[1][1] - b[-1][1]);
    result = cabs(sats16(delta_x)) + cabs(sats16(delta_y));
    result = satu12(result);
    ... (略) ...
  }
}

```

リスト1 C-ViA C言語によるソースコードの記述例(ソーベルフィルタ)
 青字の「cabs」は絶対値を取るライブラリ関数。赤字の「sats16」、「satu12」は飽和処理のライブラリ関数。

を使いこなすソフトウェアの開発効率を向上することができるのである。

* * *

以上、本稿では、組み込み用画像処理専用プロセッサについて、さまざまな画像処理の並列性とプロセッサアーキテクチャの並列性の関係を中心に解説した。画像処理技術の広がり、進化と

ともに、このような専用プロセッサの活躍の場も増えてくると考えられる。組み込み画像処理システム、または組み込み画像処理SoC設計において、画像処理アルゴリズムの特性に適したアーキテクチャを持つ専用プロセッサを選択することは、コストパフォーマンスを向上させるためにも、今後より重要になってくるであろう。 □