

SpursEngine™ の基本ソフトウェア及びミドルウェア

System Software and Middleware for SpursEngine™

白川 健治 原島 高広 齋藤 奈津子
 ■ SHIRAKAWA Kenji ■ HARASHIMA Takahiro ■ SAITO Natsuko

メディアストリーミング処理プロセッサ SpursEngine™ の基本ソフトウェア (基本 SW) は、ホストプロセッサ上のアプリケーションと、SpursEngine™ に搭載されている SPE (Synergistic Processing Element) 及びビデオデコーダ (ビデオストリームのデコード (復号化) を行う機能)、ビデオエンコーダ (ビデオストリームのエンコード (符号化) を行う機能) とを連携させて動作させる枠組みを提供する。

アプリケーションプログラムは、ストリーム型データ転送及びストリームに対する演算処理に基づくプログラミングモデルとコーデック処理^(注1)ミドルウェア (MW) により、トランスコード処理^(注2)や高次の演算処理を効率よく協調動作させることができる。

The system software for the SpursEngine™, a high-performance stream processor, provides the program execution environment to efficiently coordinate programs running on the host processor, synergistic processor elements (SPEs), and hardware codec engines.

Application programs utilizing the stream-oriented programming model and video codec middleware are able to perform video transcode functions simultaneously with various high-order processing on the video stream.

1 まえがき

SpursEngine™ は、Cell/B.E.™ (Cell Broadband Engine™)^(注3)アーキテクチャに準拠する SPE を 4 基、ビデオデコーダを 2 種、ビデオエンコーダを 2 種搭載するメディアストリーミング処理プロセッサで、ホストプロセッサと協調動作するアクセラレータ^(注4)として機能する。

SpursEngine™ を利用するアプリケーションにとって、アクセラレータの性能を引き出すには、ホストプロセッサ上で動作するプログラムと SpursEngine™ 上で動作するプログラムを連携させる必要がある。

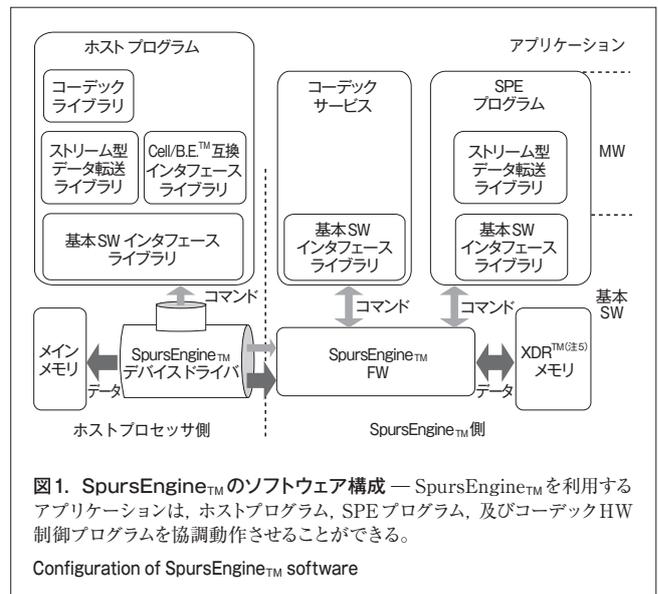
ここでは、SpursEngine™ の基本 SW の概要を述べ、次にビデオストリーム処理と汎用演算処理とを協調動作させる枠組みについて述べる。

2 SpursEngine™ のソフトウェア概要

SpursEngine™ のソフトウェアの構成を図 1 に示す。

SpursEngine™ を利用するアプリケーションは、次の 3 種類

(注1) データのエンコード及びデコードを行う処理。
 (注2) あるストリーム形式に圧縮 (符号化) されたデータなどを、別の符号化形式に変換したり、符号化データの速度だけを変換する処理。
 (注3) IBM, SONYグループ、東芝が共同で開発した高性能プロセッサで、Cell/B.E., Cell Broadband Engineは、(株)ソニー・コンピュータエンタテインメントの商標。
 (注4) コンピュータの特定の機能や処理能力を向上させるハードウェアやソフトウェア。



のプログラムから構成される。

- (1) ホストプロセッサ上で動作するホストプログラム
- (2) SPE 上で動作する SPE プログラム
- (3) コーデックハードウェア (HW) を制御するコーデック HW 制御プログラム (コーデックサービス)

これらを協調動作させる枠組みが SpursEngine™ の基本 SW であり、その中心となるのが制御プロセッサ上で動作する

(注5) XDRは、米国及びその他の国における Rambus 社の商標。

SpursEngine™ ファームウェア (FW) である。

SpursEngine™ のソフトウェアには二つの特徴がある。

第一の特徴は、それぞれのプログラムが、次のように異なる実行環境で動作することである。

- (1) ホストプログラムは、ホストオペレーティングシステムの管理下にあるプロセスとして動作する。
- (2) SPEプログラムは、FWの管理下にあるSPEプロセス内のスレッド (SPEスレッド) として動作する。
- (3) コーデックHW制御プログラムは、FWの一部として動作する。

第二の特徴は、ホストプロセッサとSpursEngine™との間に共有メモリを持たないことである。このため、ホストプログラムとSPEプログラムは、共有メモリを用いた並列処理の記述ができない。

SpursEngine™FWは、SPEプログラム及びコーデックサービスの実行環境の生成と維持をつかさどり、異なる実行環境の間の同期機構とデータ通信機構を提供している。FWは、各実行環境からの要求 (コマンド) を解釈し実行することで、これらの機能を実現している。

- (1) ホストとの間の通信路の生成と維持を行うコマンド群
- (2) メモリなど、リソース管理のためのコマンド群
- (3) データ転送のためのコマンド群
- (4) SPEスレッドを生成及び制御するためのコマンド群
- (5) コーデックサービスを生成及び制御するコマンド群
- (6) ホスト、SPE、及びコーデックの同期のためのコマンド群

ホストプロセッサ側の実行環境には、SpursEngine™ デバイスドライバと、コマンドインタプリタを操作するインタフェースライブラリを提供している。更に、Cell/B.E.™用に開発されたプログラムを移植する利便を図るため、その実行環境をエミュレートするCell/B.E.™互換インタフェースライブラリを提供している。

3 SpursEngine™によるストリーム型演算処理

コーデックHWやSPEを動作させるプログラムを記述するにあたり、各プロセッサ間のデータ転送や同期処理を直接記述する低レベルのプログラミングは可能ではあるが、記述の難易度が高く、また汎用性が低い。SpursEngine™では、ストリーム型のデータ転送とストリームに対する演算処理に基づくプログラミングモデルを推奨している。

ホストプロセッサ側の共通ライブラリとして、ストリーム型データ転送ライブラリ、コーデックライブラリなどが基本SW上に構築されている。トランスコード処理や高次の画像処理などを行うMWは、これらの共通ライブラリを利用する。

3.1 ストリーム志向のプログラミングモデル

SPE間での同期は、HWの機能により実現されているが、

SPEとコーデックサービスやSPEとホストプロセッサ間などの異なる実行環境間での同期はFWで実現している。

ストリームに対する演算処理は、データにより処理が変わるため、処理時間にばらつきがでることが多いので、受け側の状況に依存せずにメッセージを送ることができるメッセージキューを主に用いる。メッセージキューとは、指定した大きさのメッセージをキューイングできる同期オブジェクトである。ホストプロセッサとSPEが非同期で動作して、高い性能を引き出すための有用な手段となっている。

3.2 ストリーム型データ転送ライブラリ

SpursEngine™基本SWとして提供するストリーム型データ転送ライブラリは、メインメモリとローカルメモリにそれぞれリングバッファ状のメモリを確保し、それらの間のデータ転送とフロー制御を行っている。これをFIFO型 (先入れ先出し型) の共有メモリという抽象型データ構造としてプログラミングインタフェースとして提供している。このFIFOは、書込み側と読出し側を非同期で動作させるためメッセージキューを用いて実装している。ホストプログラムとSPEプログラムを結合するFIFOは処理遅延やデータ転送遅延の緩衝材となり、それぞれのプログラムを独立して動作させることができるため、互いの待ち時間を軽減させ、処理効率が向上する。SPEプログラムでの処理結果を直後の処理の入力に反映させる必要がないアプリケーションに適した転送方式である。

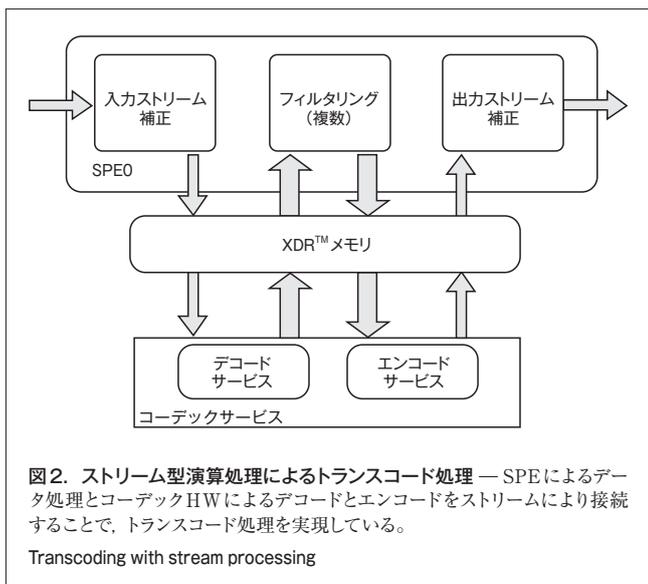
ストリーム型のデータ転送はスループットを重視したプログラミングスタイルに向いているが、FIFOにあまりデータをためないプログラムにすることで、即応性を要求するアプリケーションにも対応できる。

重要なストリームデータとして、地上デジタル放送などの動画ストリームが挙げられる。SpursEngine™はホストプロセッサとPCI (Peripheral Component Interconnect) Expressで接続されているが、これは汎用バスであるため、コンテンツ保護の観点からデータ転送にあたっては暗号化による保護が必須である。SpursEngine™の基本SWでは、ストリーム型データ転送ライブラリでデータ暗号化をオプションで行うことができる。FIFOにデータを書き込むときにAES (Advanced Encryption Standard) 暗号化し、転送後のデータを取り出すときにAES復号化が可能である。暗号に使用する鍵の管理は基本SWにより安全に行われる。アプリケーションがSpursEngine™を利用するときに暗号鍵の生成が行われ、またストリーム中途での鍵の更新も行われる。

このライブラリを使用することで、処理遅延やデータ転送遅延の緩衝材としての機能と、AES暗号化による安全なデータ転送の機能を同時に利用することができる。

3.3 SpursEngine™におけるコーデック処理

ストリーム型演算処理の例として、映像データのトランスコード処理を図2に示す。



コーデックHWにはデコーダとエンコーダがあり、それぞれMPEG-2 (Moving Picture Experts Group-phase2) 用とH.264^(注6)用がある。SpursEngine™では、デコーダ及びエンコーダとSPEを協調動作させることでエンコード処理及びデコード処理を行う。デコード結果を用いてエンコードすることで、トランスコード処理が行える。

SpursEngine™を使ったトランスコード処理における速度性能は、フルHD (High Definition: 1,920×1,080画素) サイズのMPEG-2⇒H.264変換で2.0倍速、SD (Standard Definition: 720×480画素) サイズの同変換で8.0倍速の高速処理を実現している^(注7)。

SPEは、デコーダとエンコーダの間でコーデックサービスの通信メッセージを使って、制御及び同期処理を行う。コーデックサービスは、基本SWが提供するサービスとして実装している。SPEとコーデックサービスとの通信には、基本SWの機能であるメッセージキューを用いる。

コーデックサービスにはデコードサービスとエンコードサービスがあり、以下に述べる。

3.3.1 デコードサービス デコードサービスは、ストリームデータをデコーダに入力し、デコード結果のYUV^(注8)画像とサイド情報 (解像度情報や再生時のフレームレートなど) を出力する。

デコードサービスでは、SPEからデコーダに通知するコマンドメッセージと、デコーダの状態をSPEに通知するイベントメッセージの2種類の通信用メッセージを規定している。SPE

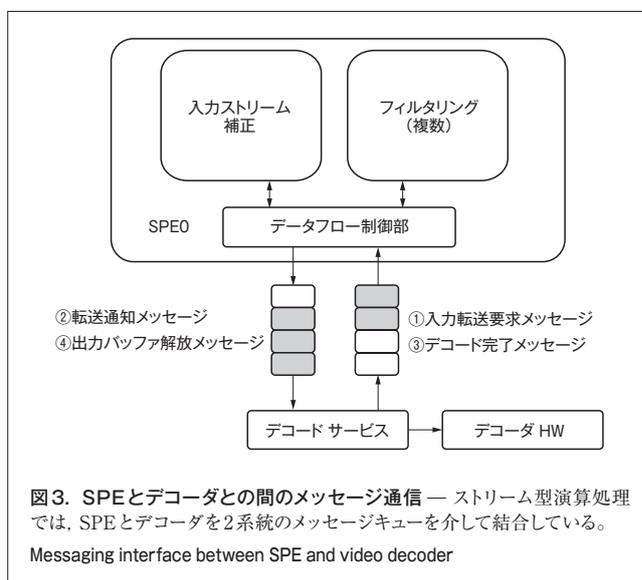
(注6) ITU-T (国際電気通信連合—電気通信標準化部門) とISO (国際標準化機構) が2003年に共同で勧告した動画圧縮符号化の国際標準規格。

(注7) ITU標準画像を使った速度測定における代表的な性能値。

(注8) Y (輝度), U (青との色差), V (赤との色差) によって画像を表現する方法。

から“デコード開始メッセージ”を通知した後、“デコード終了メッセージ”を通知するまでの間、デコードサービスはデコーダの状態に応じたメッセージをSPEに返し、またSPEからの通知メッセージをデコーダに送る。SPEは、取得したメッセージに基づいて処理を進める。

ビデオストリームのデコードは、図3の構成で次のように行われる。SPEは、“入力転送要求メッセージ (Transfer Request)”を取得すると、ストリームデータをデコーダの入力バッファにDMA (Direct Memory Access) 転送し、“転送通知メッセージ (Notify Transfer)”を通知する。通知によりデコーダはデコード処理を行い、“デコード完了メッセージ (Decode Completion)”をSPEに通知する。SPEは、画像データと関連情報をDMA転送し、“出力バッファ解放メッセージ (Release Buffer)”を通知する。



3.3.2 エンコードサービス エンコードサービスは、YUV画像データとエンコード設定をエンコーダに入力し、エンコーダのエンコード結果のストリームを出力する。

エンコードサービスにおいても、デコードサービスと同等の2種類の通信用メッセージを規定しており、エンコーダとSPEとのエンコード処理を円滑に行うことを可能にしている。

3.4 SPEからのコーデックサービスの利用

SpursEngine™でのデコード処理及びエンコード処理において、デコード前後及びエンコード前後のデータのフィルタリング処理をSPEで行う。SpursEngine™では、ホストアプリケーションと共有する付加情報を入出力データに持たせるため、ES (Elementary Stream) に独自のヘッダを付加したフォーマット (SPES) を入出力データに使用している。SPESヘッダは、SPEでの内部処理状況や入出力されるストリームの区切り情報などで構成されている。

入力ストリーム変換部ではSPESからデコード入力データフォーマットへの変換を、出力ストリーム変換部ではエンコード出力データフォーマットからSPESへの変換をそれぞれ行っている。

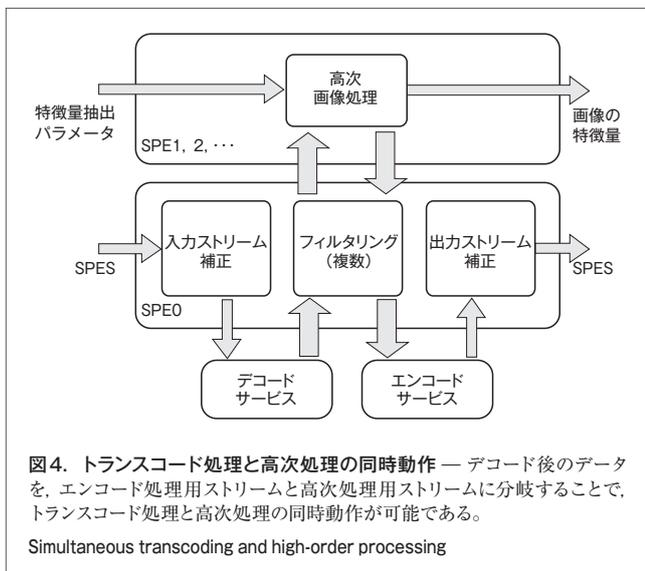
フィルタ処理部では、YUVデータに対して解像度変換、フレームレート変換、及びインタレース入力のプロGRESSIVE化などの処理を行っている。デコードした結果に対してフィルタ処理を行い、これをエンコードすることでトランスコードの自由度を高めている。

なお、フィルタ部のAPI (Application Programming Interface) を公開し、ユーザーが作成したフィルタ処理をトランスコード処理に組み込むことができる環境を提供することも検討している。

3.5 複数SPEを利用した高次処理

入力ストリームに対して、複数の出力ストリームを生成するSPE処理を用いることで、コーデック処理と並行してほかのSPEプログラムにより、同一のビデオストリームに対する演算を行うことが可能である(図4)。一般の1入力1出力のトランスコード処理だけでなく、エンコード出力に加えて、画像に対する処理結果(例えば画像の特徴量)を出力するといった、高次の演算処理をビデオストリームに対して行うことができる。

ストリームの入出力の形態は自由にプログラムすることができるため、複数のストリームを入力とするSPE処理を用いることで、各入力に対する演算処理を統合する形の高次処理も可能である。



(注9) データ転送要求を出してから、データ転送処理が完了するまでの遅れ時間。

4 汎用演算処理への応用と展開

SpursEngine™は、コーデックHWを搭載したことにより、動画像処理で優れた性能を発揮するが、SPEを活用することで汎用演算処理でも高い性能を引き出せる可能性を持っている。メインメモリとローカルメモリとの間のデータ転送のレイテンシ^(注9)はSPEの性能に対して長い間、レイテンシを隠蔽(いんべい)するだけの演算量をSPEに対して供給し、演算性能を最大限に生かすプログラミングスタイルが望まれる。

最近のGPU (Graphics Processing Unit) では、物理シミュレーションや動画像圧縮処理など3次元グラフィックス以外の様々な処理を行えるようになってきている。SpursEngine™は、大きなローカルメモリを持ち、複数の汎用演算コアがある点でGPUと共通している。GPUは、細粒度のデータ並列処理に特化しているが、SPEは粗粒度のタスク並列処理を得意とし、更にデータ並列処理モデルにも適用可能である。GPUでは高並列化が難しい処理に適しており、互いに補完する関係といえる。GPUとSpursEngine™を並列処理用のプロセッサとして共通に扱えるプログラミング言語とランタイムシステムによる協調分散処理で、いっそうの性能向上が期待できる。

5 あとがき

SpursEngine™ではSPEを活用して、デコード画像に対して高画質化や画像認識といった高次処理を行うことで、ホストプロセッサの負荷を軽減できる。更に、基本SWや開発環境を拡張、拡充することで、SpursEngine™やSPEであることを意識せずに、より汎用的な処理を効率的に行うことができるようになる。



白川 健治 SHIRAKAWA Kenji

セミコンダクター社 システムLSI事業部 先端SoC開発センター 参事。システムLSI基本ソフトウェアの設計・開発に従事。System LSI Div.



原島 高広 HARASHIMA Takahiro

セミコンダクター社 システムLSI事業部 先端SoC開発センター 主務。システムLSI応用ソフトウェアの設計・開発に従事。System LSI Div.



齋藤 奈津子 SAITO Natsuko

セミコンダクター社 システムLSI事業部 先端SoC開発センター 主務。システムLSI基本ソフトウェアの設計・開発に従事。System LSI Div.