

# Cell プラットフォーム向け アプリケーションフレームワークの開発

武内 和昭† 前田 誠司‡ 野末 浩志\* 雨宮 治郎\*\*

†株式会社 東芝 コアテクノロジーセンター

‡株式会社 東芝 研究開発センター

\*株式会社 東芝 ソフトウェア技術センター

\*\*株式会社 東芝 ブロードバンドシステム L S I 開発センター

## 1. はじめに

Cell プラットフォームにおいて、複数のプロセッサコアを効率的に動作させるプログラムを容易に開発することができるアプリケーションフレームワークを開発した。本論文では、アプリケーションフレームワークの概要と、それを用いたアプリケーションの適応例について述べる。

## 2. Cell プラットフォーム

Cell プラットフォームにおけるソフトウェア構造を、図 1 に示す。

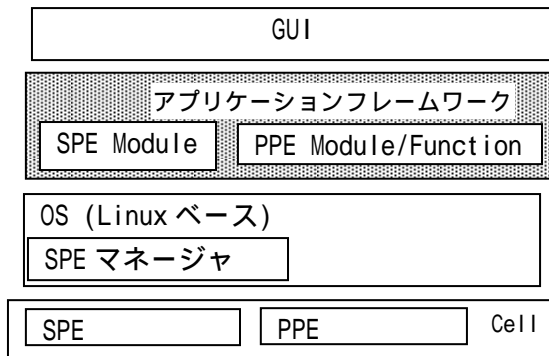


図 1 Cell プラットフォームにおけるソフトウェア構造

Cell は 1 つの PPE と 8 つの SPE からなるマルチコアプロセッサである。複数ある SPE を効率的に動作させるために、その上で動作するプログラムは SPE マネージャにより制御される。アプリケーションを構成するモジュールは PPE のみで動作する PPE モジュールと、大部分が SPE で動作する SPE モジュールに分けられる。アプリ

Development of Application Framework for Cell Platform  
 †Kazuaki Takeuchi, Core Technology Center, TOSHIBA Corporation  
 ‡Seiji Maeda, Corporate Research & Development Center, TOSHIBA Corporation  
 \*Hiroshi Nozue, Software Engineering Center, TOSHIBA Corporation  
 \*\*Jiro Amemiya, Broadband System LSI Development Center, TOSHIBA Corporation

ケーションフレームワークは、モジュールとアプリケーションを操作する GUI の間に位置し、モジュールの組み合わせを指定する手段を提供する。これにより、Cell で動作するさまざまなアプリケーションの構築が可能になる。

## 3. アプリケーションフレームワーク

アプリケーションフレームワークにより構成されたアプリケーションの構成を図 2 に示す。

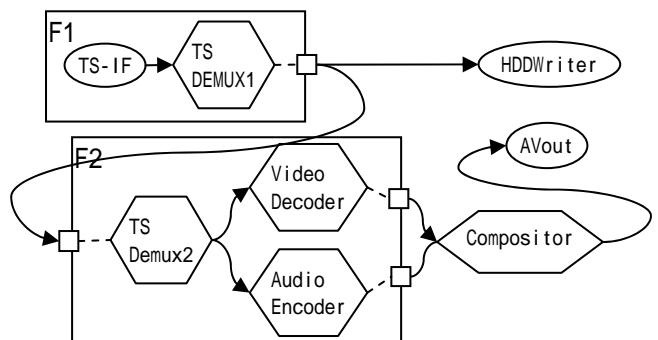


図 2 アプリケーションの構成例

このアプリケーションは、デジタル放送の視聴、録画の基本的な部分から構成される。図 2 において、丸い部分は PPE モジュールを、六角形の部分は SPE モジュールを表す。PPE モジュールは主にデバイスをアクセスする処理を担当し、SPE モジュールはコーデック処理などの計算量の大きい処理を担当する。また、各モジュールは、Port と呼ばれるデータ入出力の口を持つ。図 2 でモジュール間をつなぐ線を Connection という。図 2 において、四角で囲った部分は、ある機能を実現するモジュールをまとめて扱うファンクションを表す。図 2 のアプリケーションでは、チューナからの受信機能を実現するファンクションとストリームをデコードする機能を実現するファンクションが存在する。ファンクションの内部構成は外側から見えないため、ファンクションの中に含まれるモジュールの Port と外側のモジュールの Port は直接接続できない。この

ため、ファンクションの内部の Port を外側に見せる PortForwarding と呼ばれる仕組みを提供する。

アプリケーションを構成するファンクション、モジュールやそれらの接続関係はアプリケーション構成ファイルで指定する。ファンクションの構成は、ファンクション構成ファイルで指定する。個々のモジュールは、モジュール構成ファイルによりパラメータを指定する。SPE モジュールと PPE モジュールのモジュール構成ファイルは、実行環境や操作方法の違いを反映して、異なる形式で表現される。アプリケーションフレームワークはフレームワークコアと関連部から構成され、PPE および SPE 固有の部分を関連部が吸収する。

#### 4. アプリケーションへの適応例

今回開発したアプリケーションフレームワークを用いて、デジタルメディア処理を行うアプリケーションを構築した。アプリケーションフレームワークは、アプリケーション構成ファイル、ファンクション構成ファイルの SubComponent タグを再帰的に調べて、必要なファンクションやモジュールを生成する。その後、Connection のタグを調べて、Connection を生成する。

図 2のアプリケーションを実行する場合を例にして具体的に説明する。GUI や上位アプリケーションの指示により、アプリケーションフレームワークは図 3のアプリケーション構成ファイルをもとにアプリケーションを構成する。このファイルでは、AVout、HDDWriter、Compositor の3つのモジュールと F1、F2 の2つのファンクションからアプリケーションが構成されることを表す。このファイルを処理したときに、F1、F2、AVout、HDDWriter、Compositor が生成される。F1、F2 の構成はファンクション構成ファイルで表す。例として、F1 の構成を示すファンクション構成ファイルを 図 4に示す。F1 を生成の中で SubComponent タグに指定されたファイルが読まれて、アプリケーションフレームワークは TS-IF、TSDEMUX1、Connection、Port Forwarding を生成する。同様に、F2 の処理中に、アプリケーションフレームワークは TSDEMUX2、Video Decoder、Audio Decoder、Connection、Port Forwarding を生成する。その後、アプリケーションフレームワークは F1、F2、AVOut、Compositor、HDDWriter を接続する Connection を生成する。本フレームワークを用いてアプリケーションを構築すると、以下のような利点がある。

- 構成ファイルによりモジュール構成を指定

できるので、アプリケーションの構築が容易になる。

- モジュールの組み合わせでアプリケーションが構築されるため、モジュール単位での再利用が可能になる。

```
<Function name="digitalTV" type="PPE">
  <SubComponents>
    <SubComponent name="F2" xmlfile="F2.xml"/>
    <SubComponent name="Composite" xmlfile="comp.xml"/>
    ...
  </SubComponents>
  <Connections>
    <Connection>
      <Source module="F2" port="output"/>
      <Destination module="Composite" port="input" />
    </Connection>
    ...
  </Connections>
```

図 3 アプリケーション構成ファイル

```
<Function name="TSIn" type="PPE">
  <SubComponents>
    <SubComponent name="TS-IF" xmlfile="TSIF.xml" />
    <SubComponent name="TSDEMUX1" xmlfile="Dmux1.xml"/>
  </SubComponents>
  <Connections>
    <Connection>
      <Source module="TS-IF" port="out" />
      <Destination module="TSDEMUX1" port="in" />
    </Connection>
  </Connections>
  <Ports>
    <PortForwarding dst_module="TSDEMUX1" dst_port="out1" name="out" />
  </Ports>
</Function>
```

図 4 ファンクション構成ファイル

#### 5. おわりに

Cell プラットフォームにおいて、アプリケーションを効率的に開発できるフレームワークを開発した。デジタルメディア処理を行うアプリケーションを試作し、フレームワークの有効性を確認した。今後、様々なアプリケーションに本フレームワークを適用していく予定である。

#### 参考文献

前田他、"Cell の真価を引き出すソフトウェア実行環境を開発"、日経エレクトロニクス 2005/7/4

Maeda,S,et.al, "A CELL Software Platform for Digital Media Application", CoolChips VIII Apr. 2005