

# GPU の行方とプログラマブル GPU

～ GPU の進化が与えた衝撃～

小林 賢二, 今里 和弘

Kenji KOBAYASHI, Kazuhiro IMAZATO

## 1 はじめに

近年, CG などグラフィックス技術の進歩が脚光を浴びている. グラフィックス技術の進歩は, 映像表現の質を著しく高めたが, それと共に多大な計算負荷も招くことになった<sup>1)</sup>.

そこで, 多大な計算負荷がかかるグラフィックス処理を行うために, GPU(Graphics Processing Unit)<sup>2)</sup> が開発されてきた. 本報告では, 固定処理型 GPU からプログラマブル GPU への歴史と次世代 GPU について述べる.

## 2 GPU と DirectX

GPU とは, 高度な (主に 3D) グラフィックス処理を行うためのコントローラ・チップであり, ビデオチップが進化したものである. GPU の進化には DirectX の多大な影響があった. DirectX とは, Windows の標準開発環境を提供する API 群であり, 3 次元グラフィックスの高速な描画を可能にする Direct3D や, 音声を再生するための機能を備えた DirectSound などの Windows の基礎機能を支える様々な API が存在する. このようなプラットフォームの提供は, ソフトウェア開発者にハードウェア固有の情報を考慮しなくてもよいという影響を与えた.

### 2.1 初期の GPU アーキテクチャ

初期の GPU のアーキテクチャは, Fig. 1 に示すような, 固定機能パイプライン処理を行うものであった. 固定機能パイプライン処理とは, 前工程の処理の終了次第, 次工程の処理へ移るパイプライン処理と, 各工程が Fig. 1 の中に示すような固定機能により構成されている処理である.

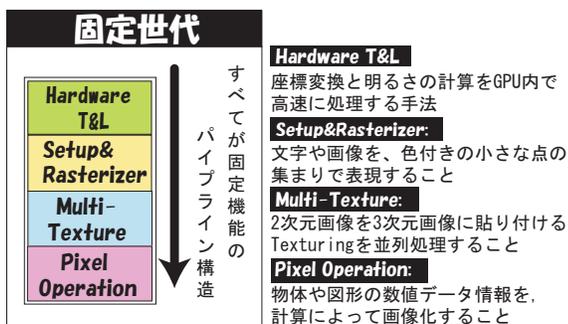


Fig. 1 初期の GPU アーキテクチャ

Fig. 1 に示した固定機能パイプラインアーキテクチャ

には大きな問題がある. このアーキテクチャでは開発時に埋め込まれたグラフィックス処理しか行うことが出来ない. そのため, 新しく開発された技術を適用したグラフィックスは処理できないため, 新しい技術の処理を行うためには, その技術を次期 GPU に埋め込まなければならない. 以上のことから, 新技術の実装の遅延という問題を生じる.

### 2.2 プログラマブル GPU

#### 2.2.1 プログラマブルシェーダ

従来の問題を解決するために, プログラマブルシェーダと呼ばれる新しい技術が実装される. プログラマブルシェーダとは, GPU の固定機能パイプラインのシェーダ (陰影) 処理をプログラム可能にする技術である. シェーダ処理は, 3D グラフィックスにおいて映像表現をより繊細でダイナミックなものにするものであり, 頂点処理を行う「Vertex Shader」と色や明暗, 質感の処理を行う「Pixel Shader」の 2 種類の処理が存在する. シェーダ処理がプログラム可能になることにより, 映像に与える効果を自由に定義できるようになる.

#### 2.2.2 Vertex Shader と Pixel Shader

Vertex Shader と Pixel Shader が物体に与える効果について述べる. Vertex Shader は, 頂点に処理を行うことにより主にオブジェクトを変形, もしくはオブジェクトの移動を行う. Fig. 2 に, Vertex Shader による変化を示す. Fig. 2(a) は処理を行う前の画像であり, Vertex Shader の処理を与えることにより, Fig. 2(b) のように変形する.



Fig. 2 VertexShader による物体の状態変化

一方 Pixel Shader は, オブジェクトの表面に, 影や色, 質感などの効果を自由に与える. Fig. 3 に, Pixel Shader による変化を示す. Fig. 3(a) は処理を行う前の物体の状態である. これに Pixel Shader の処理を行い, 光によるオブジェクトの表面の変化および, 質感を変化させ

ることにより、Fig. 3(b) のようになる。



Fig. 3 PixelShader による物体の状態変化

### 2.3 プログラマブル GPU のアーキテクチャ

プログラマブル GPU では、Vertex Shader や Pixel Shader などの機能が実装されることにより GPU のアーキテクチャは変化する。Fig. 4 にプログラマブル GPU のアーキテクチャを示す。

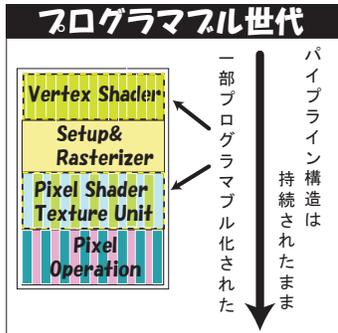


Fig. 4 プログラマブル GPU のアーキテクチャ

初期の GPU アーキテクチャでは、GPU の開発時に埋め込まれたグラフィクス処理しか行うことができなかった。しかし、プログラマブル GPU アーキテクチャでは、システムをアップデートすることにより、新しく開発された技術を即時に適用可能である。これにより、CG クリエータは最新の技術を適用したグラフィクス処理を早期に使用することが出来る。

またこのアーキテクチャでは、各工程内での処理の並列化が進められ、処理効率の向上が行われている。

## 3 次世代 GPU(1年後の GPU)

GPU は、初期の GPU の固定機能パイプライン構造から、DirectX の進化と共に、プログラマブルシェーダなどの機能を持つ、柔軟性を有したアーキテクチャに進化してきた。次世代 GPU では DirectX が一新され、Windows のサブグラフィックシステム「Windows Graphics Foundation」(WGF) と改名される。本章では、次世代 GPU に革新的な影響を与える WGF2.0 と WGF2.0 を実装した次世代 GPU について述べる。

### 3.1 WGF2.0(Unified-Shader)

DirectX は、次期 OS「Windows Vista」のユーザインタフェースの 3D ベース化により、WGF に一新される。WGF2.0 では、シェーダが拡張され、Vertex Shader, Pixel Shader を一つに統合した Unified-Shader(Fig. 5)

が導入される。これにより、従来のシェーダは、シェーダ間に機能上の相違点があったが、WGF2.0 ではこれが解消されるため、シェーダの統合が可能になる。

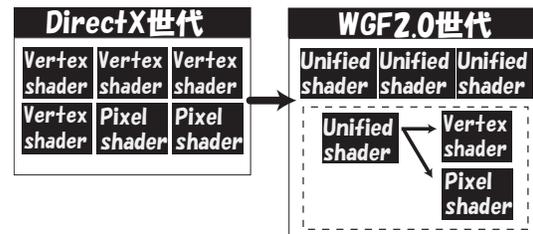


Fig. 5 Unified-Shader

### 3.2 WGF2.0 世代 GPU のアーキテクチャ

WGF2.0 世代 GPU では Unified-Shader を用いることにより、WGF2.0 世代の GPU アーキテクチャは、Fig. 6 のように DirectX 世代 GPU(Fig. 4) のパイプライン構造を崩し、プログラマブルユニットの並列化が可能になる。これにより、状況に応じてシェーダを動的に Vertex Shader や Pixel Shader に割り当てることで、処理能力を向上させることができるだろう。

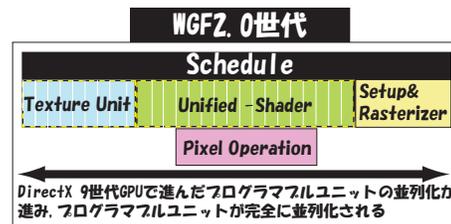


Fig. 6 WGF2.0 世代 GPU アーキテクチャの構造

## 4 まとめ

GPU アーキテクチャは、固定機能のみのパイプライン処理から、シェーダ処理のプログラマブル化などの柔軟性を持つ構造に進化してきた。そして WGF2.0 では、Unified-Shader の導入により、プログラマブル化部分の並列性を高め、従来よりさらに柔軟性を持つ構造に進化していくと考えられる。<sup>3)</sup>

## 5 今後の展望

現在の GPU は、CPU と GPU を用いてグラフィクスに関する処理を行っているため、すべての処理を単独で行うことが出来ない。そのため、将来的にはグラフィクス処理におけるこのような混合的な処理を完全に分離し、GPU 単独でグラフィクス処理を行うことを目指す。このように、GPU の汎用プロセッサ化を目指した進化を一層進めていくことになると思われる。

### 参考文献

- 1) <http://pcweb.mycom.co.jp/articles/2004/01/01/3dgraph/>
- 2) <http://d.hatena.ne.jp/keyword/DirectX>
- 3) 月刊アスキー, ASCII, 2005 7月号 p185