

DVE アドレス生成回路を事例とする 演算回路の最適化

本書の詳細はこちら：

<https://www.cqpub.co.jp/hanbai/books/52/52531.htm>

第6章までで四則演算から初等超越関数、浮動小数点演算まで、各種数値演算回路を解説しました。本章ではまとめとして、デジタル・ビデオ・エフェクト (DVE) の設計を例に、実際の回路設計における数値演算回路について考えてみることにしましょう。

7-1 デジタル・ビデオ・エフェクトとは

デジタル・ビデオ・エフェクトは、ビデオ画像に変形効果を加える手段として、現在、主流となっています。例えば、図7-1に示すようなビデオ・エフェクトによる処理には、テレビなどでおなじみのものもあると思います。このような効果は、 x - y 平面にある画像を原画像として x 、 y 方向に変形をかけることで得られます。画像をデジタル・データとすることで、メモリ上に書き込んだ原画像を自在に読み出すことができるようになり、これによってさまざまな変形を生み出すわけです。

パソコン向けのソフトウェアの中に上記と似たような機能を備えているものがあることからわかるように、こうした効果はあえてハードウェア化しなくてもソフトウェアで実現できます。

しかし、ニュースやスポーツなどのテレビの生放送では、ハードウェアによるビデオ・エフェクト処理が欠かせなくなっています。その理由は、リアルタイムで十分な画質を保って効果を得るには、今のところハードウェアに頼らざるをえないからです。ハードウェアによるビデオ・エフェクト処理は、多くの場合、画面の切り換えに使われることが多いようです。例えば、野球中継などで、ホームランがスタンドに入るシーンのVTRから次のバッターの生映像へ、ページをめくるように切り換えるといったぐあいです。図7-1のページめくり効果において、黒い背景となっている部分に別の画像がはまっているイメージと考えればよいでしょう。また、モザイクや砂嵐は特定の部分にその効果をかけることで、画面に出せない部分を隠す効果でおなじみでしょう。

ここでは、デジタル・ビデオ・エフェクトのメモリ読み出しアドレス生成部を例にとり、数値演算回路の最適化事例を紹介します。なお、ここで言う最適化とは、簡潔で規模の小さな実現構造を指します。

● 保守性や配置配線性の良い「スマートな設計」を目指す

デジタル・ビデオ・エフェクトのアドレス生成部は、数値演算の塊といってよいでしょう。しかし、

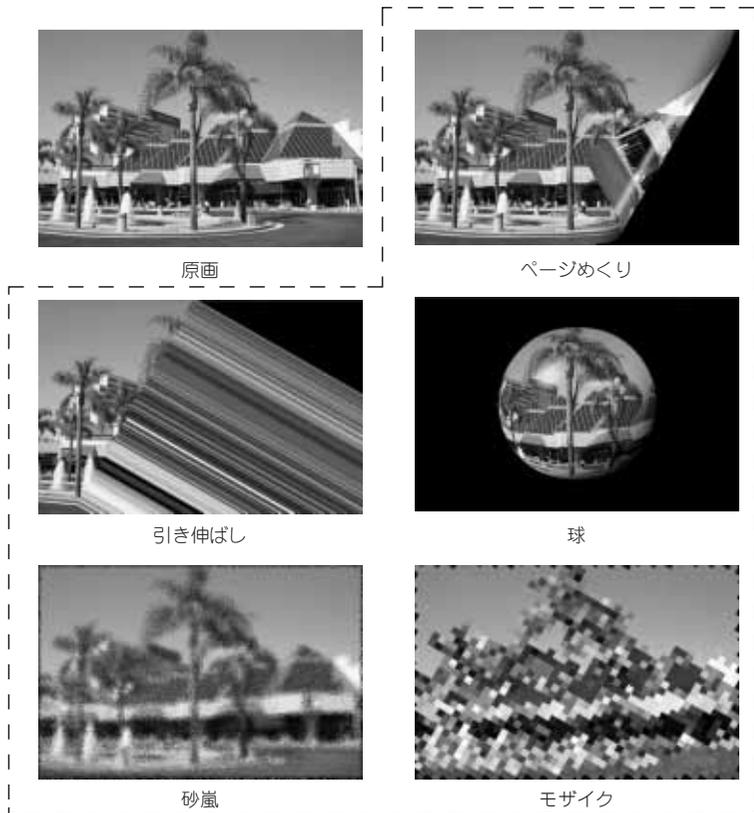


図7-1
デジタル・ビデオ・エフェクトの効果の例
中にはテレビでおなじみの処理もある

機能別に見ていくと、目的ごとに性格の異なる部分があります。本章では、ここに着目して実現手段を選択することで回路の最適化を図り、回路規模を抑制できることを示します。

ただし、回路規模を気にするあまり、回路の共用を極度に進めると、非常に難解で保守性の悪い回路になってしまいます。また、そのような回路は往々にしてターゲットとなるFPGA上で配置配線性の悪い回路になりがちです。力技も時には必要ですが、まずはスマートな実現方法を目ざしましょう。一步引いて見ることで物事を的確にとらえることができるのは、回路設計でも同じです。

また、効率の良い実装に必要とされる条件として、ハードウェアとソフトウェアの共存があります。例えば、ビデオ・アプリケーションでは、1フレーム(あるいは1フィールド)に1回更新されればよい制御パラメータが数多くありますが、これらを算出するのにハードウェアを適用するのは好ましくありません。ソフトウェアに演算の余裕がある限りソフトウェアで処理するのが最適なシステムの構造であると筆者は考えます。余分なハードウェア・リソースはほとんどん排除しましょう。

ここでは要求仕様が以下のものとして設計を始めます。

- ビデオ・フォーマット：NTSC (COLUMN7-1「ビデオ信号の方式」を参照)
- ターゲット・デバイス：米国Xilinx社のFPGA「Spartan-3 (XC3S400-4)」(システム・ゲート数は40万)

設計がひととおり終わるまでの道のりは長いので、ここで本章の説明の流れを示しておきます。

COLUMN 7-1 ビデオ信号の方式

私たちが日本でふだん目にする（ハイビジョンではない）ビデオ映像（放送）は、NTSC（National Television Standards Committee）と呼ばれる方式によるものです。

NTSCをはじめとするビデオ信号のほとんどが、ラスタ・スキャンという方式を採用しています。図7-Aにその原理図を示します。1画面（フレーム）を水平方向の線状の画像（走査線＝ライン）に分解し、1走査線を左から右へ走査（スキャン）することを走査線の数だけ繰り返すことで、画面を構成します。走査線の繰り返しを1本の線とみなすことで、1画面を時間軸方向で変化する1次元のデータとして扱えることとなります。これが、ラスタ・スキャン方式の最大のメリットといえます。NTSCでは、垂直方向に486本の走査線、1走査線当たり720画素で画面を構成しています。

またNTSCは、1秒間当たり30枚の画像を構成しています。テレビのブラウン管はその構造上、画面がつねに1画面を表示しているわけではなく、スキ

ャンしている走査線上の画素を順次表示しているに過ぎません。つまり、画面上の画素は30Hzの周期で表示されていることになります。30Hzという周波数は、人間の目には「ちらつき」として知覚される程度に遅いものです。NTSCではこのちらつきを抑えるために、インターレースという方式を採用しています（図7-B）。1フレームを構成する486本の走査線を偶数番目と奇数番目の二つに分け、偶数番目を連続したものを偶数フィールド、奇数番目のものを奇数フィールドと呼びます。偶数フィールドと奇数フィールドを順次表示することで、密度は半分になりますが、表示を見かけ上60Hzにすることでちらつきを抑えているわけです。インターレース方式は、撮影時において偶数フィールドと奇数フィールドに1/2フレームの時間差をもたせることで、フィールド間の画像の変化を滑らかにする効果もあります。

フレームを二つのフィールドに分けない方式は、プログレッシブ（ノンインターレース）方式と呼ばれます。これは最初に示した図7-Aにあたります。

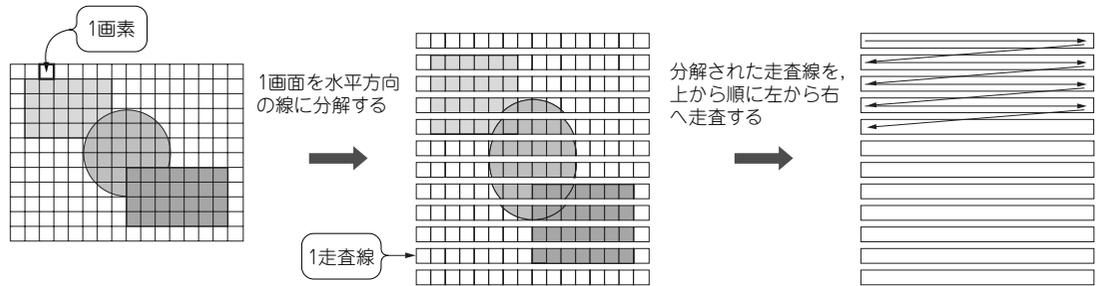


図7-A ラスタ・スキャン方式による画像の表現（プログレッシブ方式）

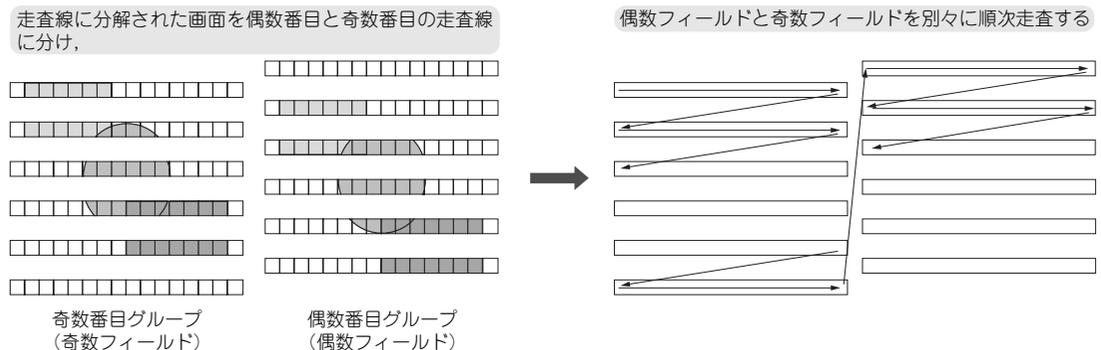


図7-B インターレース方式

- 1) デジタル・ビデオ・エフェクトの原理(何を作るのか, 基本的な部分を押さえる)
- 2) 各効果別の実現(各効果を実現するために必要な演算を数学的に理解する)
- 3) 各効果の結合(別々の機能をもつ効果を統合させる際の考えかたを理解する)
- 4) トップ・レベルの構造構築(スマートな構造の構築や規模の抑制にあたっての基本的な指針を理解する)
- 5) サブモジュール単位の具体的な回路構造

ここで重要な箇所は2)と3)です。一見無関係な複数の機能が一つにまとまっていく過程を読み取っていただきたいと思います。

7-2 効果の原理——逆写像が必要

図7-2にデジタル・ビデオ・エフェクトの効果の基本原則を示します。なお、ここでは、元となる原画像をソース画面、変形後の画像をデスティネーション画面と呼ぶことにします。デジタル・ビデオ・エフェクトを直感的に考えると、「ソース画面の画素データをデスティネーション画面へ貼り付けるために必要な変換(写像)を求める」ということになりそうです。

しかし、実際に私たちが見るのは変形後の画面になるので、デスティネーション画面の上をラスタ・スキャンして見ることになります。このラスタ・スキャンしている座標に貼り付けるソース画面の画素データの座標を逆変換(逆写像)によって求めるという方法が一般的に用いられます。少しまわりくどくなりましたが、ここでは「逆写像が必要である」ということだけ覚えておいてください(詳しく知りたい方はコンピュータ・グラフィックスなどの参考書を参照のこと。巻末の参考文献(6)はお勧め)。

図7-3に、これを実現するためのブロック図を示します。必要に応じて前処理フィルタをかけたソース画面を、一度フレーム・バッファ・メモリに書き込み、アドレス生成回路によって生成された逆写像アドレスによって画素データを読み出し、走査線に貼り付けること(レンダリング)でデスティネーション画面を構成します。今回着目するのは図の中で灰色に塗ったブロックです。なぜアドレス生成回路と呼ぶかといえば、メモリの読み出しアドレス(逆写像アドレス)を生成するという理由からです。

なお、今回ここで紹介するのは2次元(2D)のデジタル・ビデオ・エフェクトです。陰影を付けることで立体的に見せていますが、例えばページめくり効果を斜めに見たところで、めくった部分は持ち上がって見えません。あくまでも正面から見た平面を表現しているに過ぎないためです。これに対し、最近の主流は3次元(3D)のデジタル・ビデオ・エフェクトです。3Dのコンピュータ・グラフィックスと同じ手法で画像を生成します。貼り付ける素材(テクスチャ)が動画である部分が違うだけです。

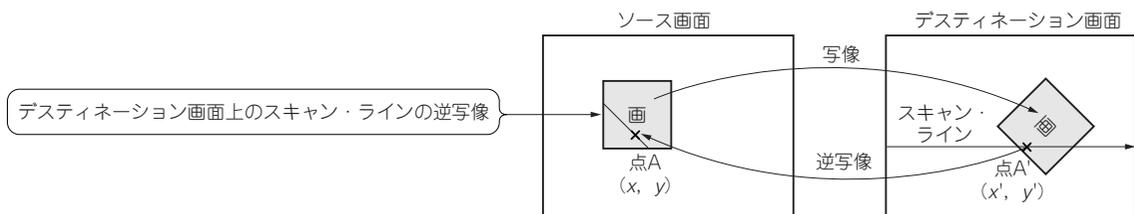


図7-2 デジタル・ビデオ・エフェクト実現の基本原則

ラスタ・スキャンしている座標に貼り付けるソース画面の画素データの座標を逆変換(逆写像)によって求めるという方法が一般的に用いられる

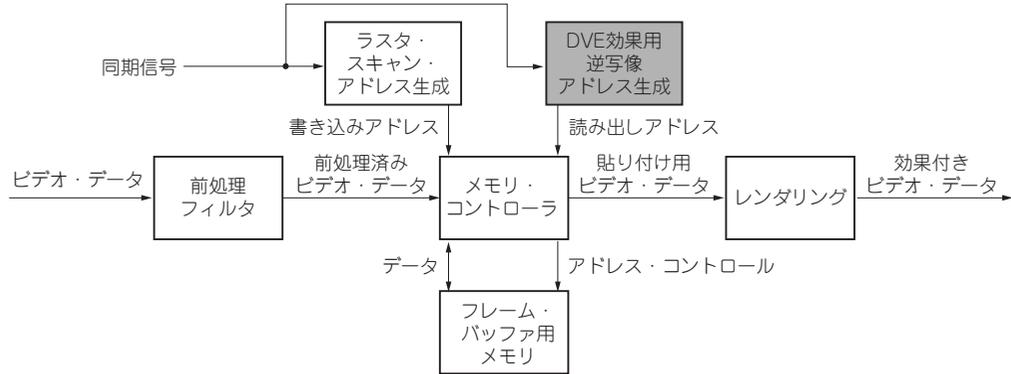


図7-3 デジタル・ビデオ・エフェクト実現の基本ブロック図

必要に応じて前処理フィルタをかけたソース画面を、一度フレーム・バッファ・メモリに書き込み、アドレス生成回路によって生成された逆写像アドレスによって画素データを読み出し、走査線に貼り付けること(レンダリング)でデスティネーション画面を構成する

7-3 最初の一步は数学的アプローチ

ここからは、各効果を実現するにあたって必要な演算を検討します。あくまでも算術的なアプローチとなりますが、この段階で複数の機能をいかに統合するかを検討しておくことは、非常に重要です。これをやっておかないと、最悪の場合、機能ごとに一つ一つ回路を作って、出力をマルチプレクスする、といった非常にみっともない結果になりかねません。以下の説明では、このアプローチにより特殊な構造を採用します。

● ページめくり効果——四つの領域に分けて考える

図7-4に、ページめくり効果の実現方法を示します。制御パラメータは次の三つです。

- めくり半径： R
- めくり量： Δx
- めくり方向： θ_r

図7-4(c)は画面を上から見た図で、 Δx と θ_r を定義しています。図7-4(d)はめくり方向の真横から見た図で、 R と座標の関係を示しています。視点から見ためくられた形がデスティネーション画面上に描画されており、このめくりを戻してやったときに対応する点がソース画面上の画素データになります。例えば、図7-4(d)の点P0はめくりの起点となる場所で、点P1は視点から見て点P0の右に R だけ離れた位置に見えます。めくりを戻していくと、ソース画面上の対応する点P1'はP0から円周の1/4の距離にあることになります。

このようにすると、領域を四つに場合分けして考えればよいことになります。めくる前の領域をarea 0とし、後は 90° ずつarea 1, area 2と続き、area 3では図7-4(a)のページめくり1と図7-4(b)のページめくり2で別処理になります。ページめくり2の場合、area 4まで考える必要がありそうですが、この部分は不可視ですので必要ありません。

これをまとめると、表7-1のようになります。なお、表の中で x , y と表記されているのは、 x - y 平面上

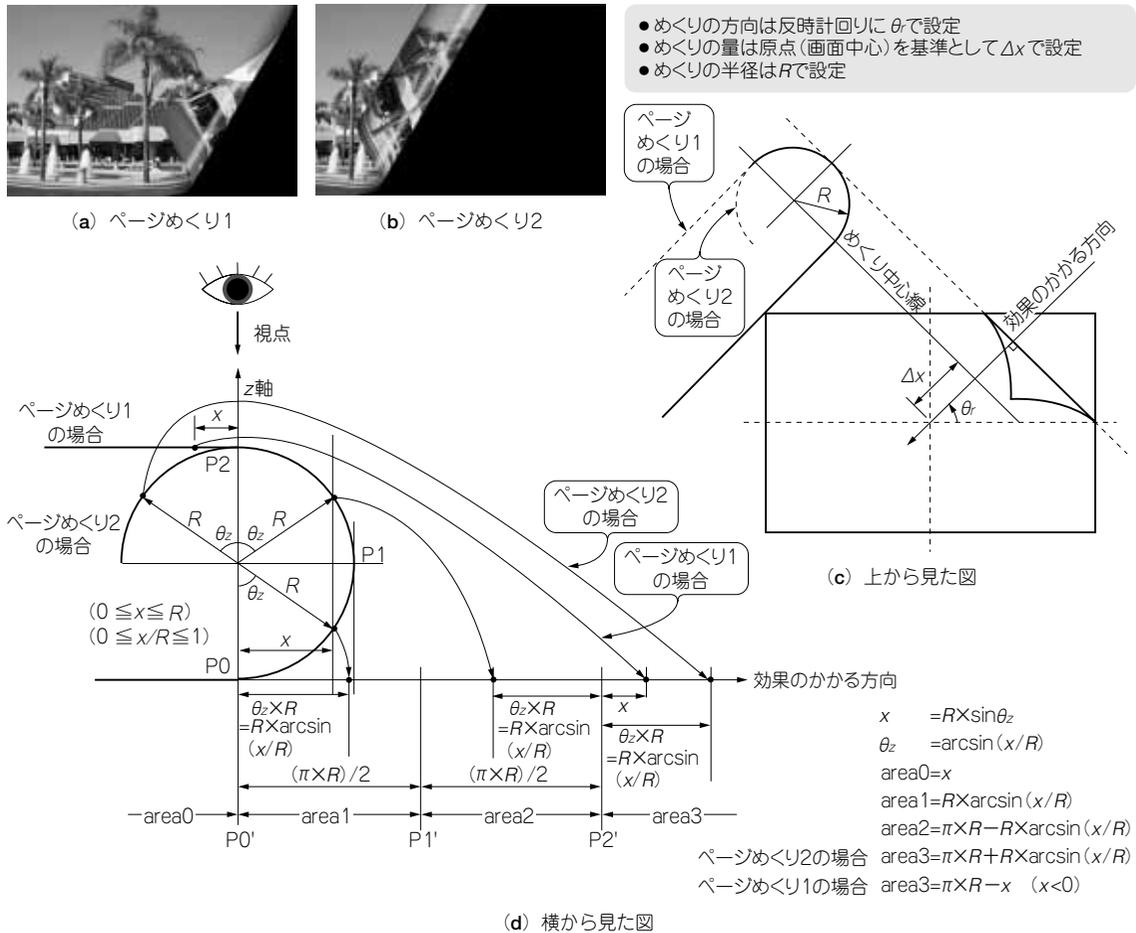


図7-4 ページめくり効果の実現

(a)と(b)にページめくりの効果の例を示す。こうした効果を実現するには(c)と(d)に示す数学的なアプローチをとる。(c)は画面を上から見た図であり、 Δx と θ_r を定義している。(d)はめくり方向の真横から見た図(効果のかかる方向をx軸方向にとってみた場合)であり、 R と座標の関係を示している

の座標そのものではなく、図7-4(c)におけるめくり中心線をy軸、効果のかかる方向をx軸とした直交座標によるものです。効果のかかる方向がx軸に一致し、めくりの中心が原点を通るときにかぎり、本来のx-y座標と一致します。

ここまでできれば、x軸に沿ってめくることが出来ます。しかし、紙をめくるときは角からめくことも珍しくないので、任意の方向にめくれるようにしておかなければなりません。そこで、図7-4(c)の原点周りに回転する操作が必要になります。

以上のことから、次の二つの工程が必要になります。

- z軸方向の情報をx-y平面上に展開していく
- めくり方向に回転させる

しかし、単純に二つの工程を縦方向に並べたのでは全体が回転してしまい、ここで目ざすものとは異なってしまいます。例えば、めくられていない部分に回転操作があってはならないのです。したがって、