

## アーケードゲームのテクノロジー (その1) 増補改訂版

宮沢 篤† 駒野目 裕久‡

† 日本アイ・ビー・エム株式会社 東京基礎研究所

〒242 神奈川県大和市下鶴間1623番14

‡ 池上通信機株式会社 R&Dセンター

〒251 神奈川県藤沢市小塚400

ピンポンをもとにした、商業的に成功した初めてのアーケードゲーム「ポン」が、米アタリゲームズ社で発明されてから、既に20年以上の歳月が流れている。当時のゲームは、汎用ロジックICを組み合わせて設計されており、技術的に見ても未発達で、最も単純な対話型コンピュータグラフィックスの一応用分野でしかなかった。それから現在までに、世界中のさまざまな会社から、その時代の最も進んだコンピュータ技術を取り入れた、非常にたくさんのゲームが発表されてきた。今日のアーケードゲームは、幾多の技術革新を経て進化してきた、全く新しいインタラクティブなメディアである、と言えるかもしれない。本稿では、ゲームマシンのハードウェアを中心に、アーケードゲームを構成するいくつかの基本的な技術について解説する。

### ARCADE GAME TECHNOLOGIES PART 1

Expanded and revised edition

Atsushi Miyazawa † and Hirohisa Komanome ‡

† IBM Research, Tokyo Research Laboratory IBM Japan, Ltd.

1623-14, Shimotsuruma, Yamato-shi, Kanagawa-ken 242, Japan

‡ R & D Center Ikegami Tsushinki Co., Ltd.

400, Kozuka, Fjjsawa-shi, Kanagawa-ken 251, Japan

Over 20 years have passed since the first commercially successful arcade game "Pong," a simulated game of ping-pong, was invented by Nolan Bushnell at Atari. The game machines at that time were designed to use standard logic ICs. From the technological point of view, they were all immature, and as applications of interactive computer graphics they were also very primitive. Since then, a huge number of games incorporating state-of-the-art computer technologies have been released by companies all over the world. Recent arcade games, which have evolved through several stages of technical innovation, constitute what may be called a brand-new interactive medium. In this paper, we discuss several fundamental technologies used in arcade games mainly in terms of hardware architecture.

ピンポンをもとにした、商業的に成功した初めての業務用ビデオゲーム「ポン」(Pong, 1972年)が、米アタリゲームズ社を設立したノラン・ブッシュネルによって発明されてから、既に20年以上の歳月が流れている。当時のゲームは、TTL (Transistor-Transistor Logic)を中心とした多量の論理ICを組み合わせて設計されており、技術的に見ても未発達で、最も単純な対話型(インタラクティブ)コンピュータグラフィックスの一応用分野でしかなかった。それから現在までに、世界中のさまざまなゲーム会社から、その時代の最も進んだコンピュータ技術を取り入れた、非常にたくさんのビデオゲームが発売されてきた。ゲームデザイナーの優れた感性と、コンピュータエンジニアの高度な技術の両方が融合することによって、今日のビデオゲームは、全く新しいインタラクティブなメディアにまで進化している、と言えるかもしれない。そしてこの新しいメディアは、出版界や映画、テレビといった、伝統的なメディア産業にも大きな影響を与えている。例えばビデオゲーム関連の出版物は、攻略本や情報誌から始まって、ゲームの人気キャラクターを主人公にした小説やコミック、アニメまでありとあらゆるジャンルに及んでいる。今まで単なる暇つぶし、または子供向けの遊びやおもちゃの感覚で扱われていたビデオゲームが、一つの文化として語られる時代になってきたのである。

ビデオゲームを学問的に分析しようとする試みは、アタリ社の家庭用ゲーム機「アタリ2600、通称VCS (Video Computer System, 1977年)」がアメリカで爆発的な人気を呼んだ70年代後半から、主に心理学や教育学、メディア論などの歴史社会的な領域で進められてきたが、コンピュータを含む電子的な映像技術の応用という立場からの調査研究は、これまでほとんど行われたことがなかった。コンピュータグラフィックス草創期の技術を、広い範囲にわたって解説した書物として有名な、ウィリアム・ニューマンとロバート・スプロールによる“PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS, 2/e” (McGraw-Hill, 1979年)は、上記のビデオゲーム「ポン」のルール説明と、「ビデオゲームは、

家庭で最初に使われた対話型グラフィックスの代表である。」という印象的なメッセージで始まっている。残念なことに、技術的に未発達だった当時のビデオゲームにそれ以上の出番は与えられなかったが、このところ人気を集めている、いわゆる新世代ゲーム機の特徴となった3次元グラフィックス機能は、多角形(ポリゴン)の走査変換(スキャンコンバージョン)、幾何計算、隠面消去アルゴリズム、グーロー(Gouraud)シェーディングやテキストチャマッピング、そしてTLMIMI (Tri-Linear MIP-Mapped Interpolation)に及んでいる。今日のビデオゲームは、コンピュータグラフィックスの最新技術をふんだんに使いながら、なおかつ究極的なインタラクティブ映像への発展途上にある、と言えるかもしれない。本稿では、70年代後半から現在に至るまでのゲームマシンのハードウェアを中心に、ビデオゲームを構成するいくつかの基本的な映像表示技術について(一部推測を交えながら)解説する。

## ゲームマシンのアーキテクチャ

ファミコンやメガドライブを代表とする家庭用ゲーム専用機(コンシューママシン)から、業務用ゲーム機(アーケードマシン)に至るまで、ゲームマシンのハードウェアには、これが標準といった規格はなく、各社の各種機種ごとにその仕様はバラバラである。しかし、強力なグラフィックス機能や高品位なサウンドは、高度なゲームを作り上げるための必要条件であり、これらのマシンに共通する特徴となっている。多くの場合、ゲームの実行や画面処理(グラフィックス)、サウンドの発生などは、それだけを専門に担当するCPU(6502、8080、Z-80、6809などに始まって、現在では68000系が主流)によって行なわれている。この意味で、ほとんどのゲームマシンは、複数のCPUを搭載したMIMD (Multiple Instruction Multiple Datastream)型の並列計算機、という見方ができる。

もともと、ビデオゲームの黎明期における作品、例えば上記の「ポン」や、「ブロックくずし」という呼び名のもと大ブームとなった「ブレイクアウ

ト」(Breakout, 1972年/アタリゲームズ)などは、TTL-ICを組み合わせて設計された大掛かりなもので、当時の電卓技術を流用したと言われている。ビデオゲームに汎用CPU(6502)が初めて使われたのが、「風船割り」の通称で有名な「サーカス」(Circus, 1977年/エキシディ)においてであり、以降のゲームは例外なくマイクロプロセッサを用いている。

## 画面処理

ビデオゲームにとって必要なのは、高速で動く、たくさんの色鮮やかなキャラクタである。グラフィックスの表示速度が遅ければ、いくら緻密で美しい画面を出したとしても、もはやゲームとしては成り立たない。IBM-PCやMacintoshのような、一般的なパソコンで主流を占めているビットマップ方式のグラフィックスは、速度上の問題から、現在では全く使われていない。過去の例としては、有名なタイトー社の「スペース・インベーダー」(Space Invaders, 1978年)[4]や「QIX」(1981年)、ナムコ社の「リブルラブル」(Libble Rabble, 1984年)などがある。ゲームマシンの画面構成をひとこと言えば、PCG(Programmable Character Generator)のような疑似グラフィック画面(通常、2面から7面ぐらいあって、重ね合わせて表示される)と、それらを独立して上下左右に1ドット単位で動かせるスクロール機能である(通常のPCGは8ビットしかないので、256種類のキャラクタしか表示することができない。そこで実際のゲームマシンでは、キャラクターコードを10ビット以上に拡張して使っている)。これに、後述するスプライトの機能が加わって、あのスピード感あふれる映像が作り出される[5,6](図1)。

## スプライト

スプライトはオブジェクトとも呼ばれ、ハードウェアによる特殊な表示機能を持ったキャラクターパターンと言える。通常、ソフトウェアにおいてグラフィックス画面上のキャラクタを移動させる場合、現在表示しているパターンを消してから、移動

先に新しいパターンを表示する必要がある。従って、高速に多数のキャラクタを同時に移動させることはほとんど不可能である。(かつてのインベーダーゲームでは、キャラクタパターンの書き換えをソフトウェアで行っていたため、1/60秒ごとに移動できるキャラクタの数に制限を受けていた。つまり、ゲームの最初ではキャラクタの数が多いため移動スピードが遅く、キャラクタの数が減るに従って移動スピードが早くなる効果として現れている。)

このスプライト機能の特徴は、座標の変更のみで表示画面内を自由に移動することができ、ソフトウェアによるパターンの再定義は一切必要としない。また、スプライトには優先順位があり、2つ以上のスプライトが重なった場合、優先順位の低いスプライトは自動的に消えるようになっており、手前にある物体が後方の物体の前を通り過ぎるように見え、画面に奥行き感を持たせる表現が簡単に行える。さらに、スプライト同士の1画素が一致するとプロセッサ内部の衝突フラグがセットされるなど、ビデオゲームの制作には欠かせない機能の一つである。スプライトは、もともとMSXパソコンなどに採用されたテキサスインスツルメンツ社のグラフィックLSI(TMS-9918-A)にユニークな機能であったが、当時のゲームボードには性能面ではかき越えるハードウェアが搭載されていた。

スプライトの表示回路のブロック線図と表示処理の概略を図2に示す。スプライトは、画面に表示するパターンデータが登録されているスプライト・キャラクタ・パターンROMと、スプライトの位置、パターン番号及びカラーコードをセットするスプライト属性テーブルから構成される。CPU側からは、このスプライト属性テーブルにデータをセットするだけで、任意のスプライトを画面に表示することができる。スプライトを表示するVRAMとしては、表示画面のドットに対応した2次元の画面メモリを持っているわけではなく、一走査線分のラインバッファのみを持っている。スプライトの表示処理は次のとおりである。

まず、CPU側からスプライト属性テーブルに表示したいスプライトのデータ(x, yアドレス、パ

ターン番号、カラーコード等)を書き込みハードウェアに渡す。通常、この処理は、表示画像の乱れを防ぐために垂直ブランキング期間(約1.2ms)に行われる。

スプライト属性テーブルに書き込まれたデータは、水平ブランキングのタイミングごとに、表示判定アドレス演算部で、各スプライトのyアドレスと走査線アドレスY<sub>a</sub>(TV水平信号カウンタ)をもとに、走査線Y<sub>a</sub>上に存在するスプライトの判定を行う。走査線アドレスY<sub>a</sub>上にスプライトが存在すれば、パターン番号より該当パターンをパターンROMから読みだし、走査線が読み出すラインバッファに書き込みを行う。このラインバッファへの書き込みは、走査線アドレスとスプライトx、yアドレスから、該当する一ライン分のデータをパターンROMから読みだして行なわれる。この処理を、すべてのスプライト(スプライト属性テーブル)に対し水平ブランキング期間(11.4μs)で行う。水平ブランキング期間中は処理時間が短いためラインバッファを2つ設け、表示と処理を並列にできるようにし、水平期間(63.55μs)で行っているものもある。最終的にできあがった、ラインバッファの内容は、ビデオ表示装置によりディスプレイに表示される。この一連の処理を走査線ごとに行う事により、画面上に2次元のスプライトの表示が行われる。

スプライトの表示処理は一走査線単位(水平期間)に行われているため、この処理時間の間に幾つものスプライトが処理できるかにより表示制限が決定される。TI社のTMS-9918-Aでは、一走査線上に最大4個までの表示が行える。アーケードゲームの例として「ドンキーコング」(Donkey Kong, 1981年/任天堂)に使用したハードウェアのスプライト制御仕様を示す。

- ・スプライト表示アドレス 256×256  
(表示画面有効区間 224×256)
- ・スプライトのサイズ 16×16ドット
- ・パターン数 128個
- ・一走査線スプライト表示最大数 16個
- ・一画面スプライト表示最大数 96個
- ・左右、上下反転機能

- ・カラー 3色+黒(カラーコードにより16種)  
(スプライトのサイズは、ゲーム機により異なり、8×8、16×16、32×32ドットなどの種類がある。)

「ドンキーコング」では、最初にすべてのスプライトに対し、表示走査線上に存在するかのチェックを行い、該当するスプライトにマークを付ける処理を行っている。その後、マークの付いたスプライトデータのみを処理している。存在チェックでは、96個のスプライト属性テーブルから、一走査線上で表示できる最大の16個のスプライトの抽出処理を行っている。

## スクロール機能

通常のグラフィック画面でスクロール処理を行うには、背景画のVRAMを数画素ずつ、行または列ごとにずらして直接書き換える必要があり、リアルタイムに背景を動かすことはできない。そこで、ゲームマシンには、背景VRAMの表示開始位置を指定するのみで背景画が表示できるハードウェアが設けられており、リアルタイムのスクロール処理を実現している。

スクロール制御は、CPU側からハードウェアに対し背景画面の表示開始位置(表示開始アドレス)を定期的に更新することで実現される。この更新処理は、スプライトの時と同様に、表示画像の乱れを防ぐために垂直ブランキング期間に行う必要がある。Y座標を固定してX座標を垂直ブランキング期間ごとに更新すれば、横スクロール、逆ならば縦スクロールとなる。このとき、背景VRAMはリングバッファ構造としているため、図3の(b)に示すように、背景VRAMを越えた表示画面領域はラップして表示される。スクロール方向の表示画面領域外の背景VRAMを更新することにより、同じ背景画の繰り返しではなく、変化ある背景画のスクロールが行える。

実際の表示機構では、ゲーム空間のすべての背景画をそのままデータとして持つには、データ容量が大きくなってしまいうため、通常、8×8の基本的なセル・パターン(通常、256個)を組み合わせて

背景画を構成し、データ容量の削減を図っている。

スクロール処理の概要を図4に示す。ここで、CPU側が背景画のセル・パターン番号をハードウェアに指示する領域を、背景マップRAMと呼ぶことにする。この背景マップRAMは、画面アドレスを8×8のセルに分割した表示位置と1対1に対応している。すなわち、予めセル・パターンをROM化しておき、CPU側からはセル・パターンの番号を背景マップRAMに書き込むことで表示が行える機構をとっている。また、背景画面の表示開始位置をハードウェアに渡す表示開始アドレスポイントを設けてある。表示開始アドレス、及び表示アドレスは1ドット単位の制御を指定するため、背景マップRAMのアドレスを示す部分とセル・パターン内のアドレスを示す下位3ビット部分の2つよって構成される。

背景画の表示処理はスプライトの表示処理と似ている。しかし、表示するためのVRAMは、スプライトよりさらに小さく、セル・パターンの一ライン分（実際には2ライン以上）のシフトレジスタみである。次に背景画の基本的な表示処理の流れについて説明をする。

(1) CPU側からは、あらかじめ背景マップRAMにセル・パターンを書き込み、表示開始アドレスをセットしておく。垂直ブランキングの割り込みが発生時点で、表示開始アドレスの更新を行う。

(2) ハードウェア側では、表示開始アドレスと走査線の水平、垂直カウンタの値から、次に表示する表示アドレスを計算し、該当する背景マップRAMから、格納されているセル・パターン番号を読み出す。

(3) 読み出したセル・パターンの番号とY座標の表示アドレスYbの下位3ビットにより、背景パターンROMから該当セル・パターンの表示すべき8ドット分のデータを取り出しシフトレジスタにセットする。

(4) シフトレジスタの内容は順次読み出され、ビデオ信号優先選択器に送られ、キャラクタ、スプライト、背景画の表示優先判定を行った後、画面に表示される。

(5) X座標の表示アドレスXbが8ドット加算して背景マップRAMのアドレスを更新し、次に表示するセル・パターンのアドレスを読みだす。

以後は、前項(3)、(4)を繰り返して走査線分の表示を行う。(6)以上の処理を画面表示領域全体にわたり行うことにより背景画面が表示される。

以上の様に表示機構により、表示開始位置を垂直ブランキング期間ごとに変えることで、背景画面のスクロール処理が行える。

縦、横のスクロール制御を組み合わせることで、斜め方向のスクロールも行える。この斜め方向のスクロールを横方向のスクロール制御のみで行っている特別なゲームとして「ザクソン」(Zaxxon, 1982年/セガ・エンタープライゼス)がある。このゲームは表示画面に3次元の立体感を与えるために、立体を斜め方向から見た画面構成とし、スクロール処理も斜め方向に行っているものである。これは、もともと8×8の正方形のセル・パターンを、読み出すときに斜めのセル・パターンとして読み出しを行っている。これは、背景画全体が斜めの絵となっているため、基本となるセル・パターンを斜めにするにより、パターン利用効率を上げることからこのような特殊な処理を行っている。また、スクロールの動きをなめらかにするため、移動の単位はX方向に2、Y方向に1を移動の単位として行っている。

スクロール処理の中で変わったものにラスタースクロールと呼ばれるものがある。これは、通常のスクロール処理が表示開始アドレスの変更を画像の乱れを防ぐために垂直ブランキング期間に行っていたものを、1走査線ごと(水平ブランキング期間)に表示位置を変化させるものである。表示開始アドレスを1走査線ごとにサインカーブ状に指定すれば、画面がゆらゆらとした表現ができる。このラスタースクロールの技術を応用した代表的なものとしては「ダライアスII」(Darius II, 1989年)や「ガンフロンティア」(Gun Frontier, 1991年/いずれもタイトー)などがあり、面白い例としては「ポールポジション」(Pole Position, 1982年/ナムコ)に代表される初期のドライブ・ゲームがある。これは、ラスタースクロールを道路のカーブに

うまく応用し3Dを表現したものである。

## 特殊効果

画面処理に触れたついでに、背景やキャラクタの拡大・縮小や回転機能など、ビデオゲームの表現力を高めるための特殊効果についても、見逃すわけにはいかない。これらの機能は、数学で言うところの座標幾何の応用であるが、定義通りまともに座標計算をしていたのでは、とても限られた時間内に絵を作り終えることができない。そこで、高速化のためのさまざまな工夫が必要になってくる。まず拡大・縮小については、キャラクタパターン内の1画素を、定期的に繰り返したり間引いたりしながら表示することでとりあえず実現できて、いわゆる最近隣法 (Nearest-neighbor Interpolation) による0次の補間と同じ結果になる。この処理が、実は直線発生アルゴリズムを用いて高速に実現できることについては、Braccini と Marino が示している [7]。

直線発生器は、もともとコンピュータグラフィックスの分野において、直交座標系の格子点上に誤差最小に離散化された直線を、微分解析器 (Digital Differential Analyzer) の原理を用いて高速に発生させるために使われるものである。このとき得られる傾き  $N/M$  の直線を、 $N$  個の画素列と  $M$  個の画素列間の対応付けのための規則ともみることができる。これは、画像データに対して倍率  $N/M$  の拡大・縮小をすることと等価である。アルゴリズムは加減算とシフト操作の範囲内で行えるために、ハードウェア化も容易である。ひとたび、画像の拡大・縮小が実時間で実行できるようになると、これを使ってゲーム画面に疑似的な3次元効果を出すことが可能になる。その結果、例えば前述の「ボール・ポジション」や、セガ社の大型可動筐体による「体感ゲーム・シリーズ」として、「ハングオン」(Hang On, 1985年)や「スペース・ハリヤー」(Space Harrier, 1985年)、「アウトラン」(Out Run, 1986年)など、数々の名作と呼ばれた疑似3次元ゲームが次々と制作されることになる。

回転については、2次の変換行列をたかだか4つの単純な行列の積に分解し、それぞれを整数演算の

範囲内で、高速に計算するようなアルゴリズムがいくつか知られている。しかし、現実のゲームマシンでは、各パターンを実行時にリアルタイムで回すのではなく、あらかじめ回転しておいたパターンを適当に選びながら、それらを決められた位置に表示することで、あたかも回転しているかのように見せているものがほとんどである。戦闘機のドッグファイトをシミュレートした「アフターバーナーII」

(After Burner II, 1987年/セガ・エンタープライゼス)では、敵の攻撃を避けながら、空中旋回のグラフィックが360度ぐるっと回る。これは、ゲーム画面に表示されるのほとんどすべてものをスプライトだけで構成し、スプライトの表示位置を調整することによって、背景そのものが回っているような雰囲気を出すことに成功している。同様の技法を用いたのが、迷路の中を下に落ちていくボールを画面全体を回転させてゴールへ進むアクションゲーム「キャメルトライ」(Camel Try, 1989年/タイトー)である。このゲームでは、迷路を構成する一つ一つのブロックのすべての回転パターンをスプライトとして用意することで、回転機能を持たないシステムへの実装を可能にしている。もちろんこの技法を用いることができるのは、あらかじめ用意しておく回転パターンの数が少ないときに限られる。初めてハードウェアによるフレームバッファの回転機能を実現したのが、「A-JAX」(1987年/コナミ)においてであり、その後も戦車やヘリコプターを操るシューティングゲーム、「アサルト」(Assault, 1988年)や「メタルホーク」(Metal Hawk, 1988年/いずれもナムコ)などの名作が生まれている。

また、数学では射影変換 (Projective Transformation) と呼ばれる、長方形の背景パターンを台形に変形することによって、奥行き感を持った画面を作る手法が、「ゴルフンググレイツ」(Golfing Greats, 1991年/コナミ)や任天堂のスーパーファミコン用ソフト、「スーパーマリオカート」(Super Mario Kart, 1992年)などで使われている (図5)。これは一見すると、長方形パターンの上部を縮小、下部を拡大していくことによって作

れそうな気がするが、射影変換自体が非線形なため、実際にはうまくいかない。この変換にも、やはり整数演算の範囲内で行なえるアルゴリズムが存在する [8]。

なお、ビデオゲームを支える技術に関しては、この他にも立体視やポリナイザー、画像処理やサウンドなどの興味深い話題もあるが、また別の機会とさせていただきます。

### 参考文献

1. テレビゲーム・ミュージアム・プロジェクト編：テレビゲーム／電視遊戯大全／TV-GAMES, UPU (1988年)。
2. 月刊ゲームスト増刊：ザ・ベストゲーム, 新声社 (1991年)。
3. コンピュータ局開発：新世代ゲームビジネス, 日経BP社 (1995年)。
4. テレビジョン学会編：応用マイクロコンピュータ, 昭晃堂 (1980年)。
5. 資料「ドンキーコング プログラム仕様書」, 池上通信機株式会社 (1981年)。
6. ラジオライフ別冊：ゲームマシン大研究, 三才ブックス (1989年)。
7. Carlo Braccini and Giuseppe Marino : Fast Geometrical Manipulations of Digital Images, Computer Graphics and Image Processing, Vol.13, 1980, pp.127-141.
8. A. Miyazawa : SYSTEM FOR SOURCE IMAGE TRANSFORMATION, U. S. Patent 5, 077, 681 (Dec. 31, 1991)。

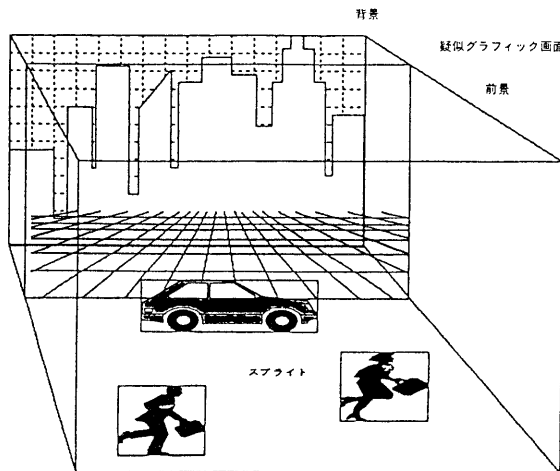


図 ビデオゲームの画像処理①

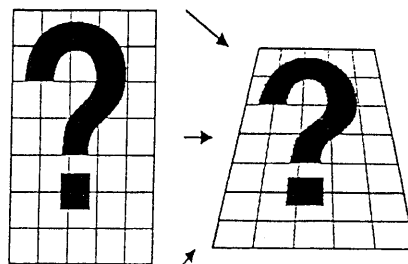
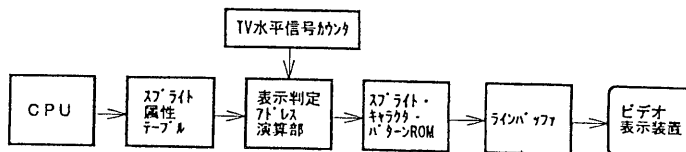
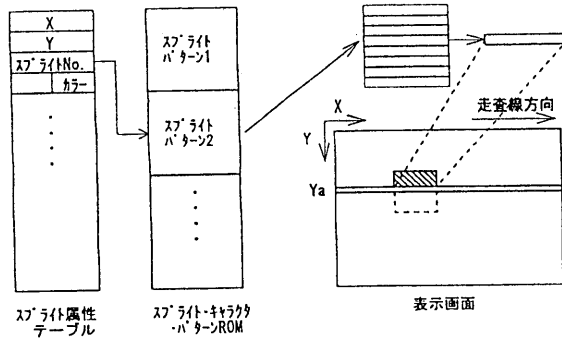


図 ビデオゲームの画像処理②



表示回路ブロック線図



スプライト表示処理

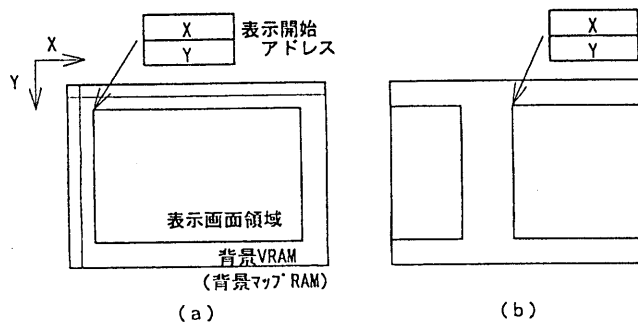


図 表示領域

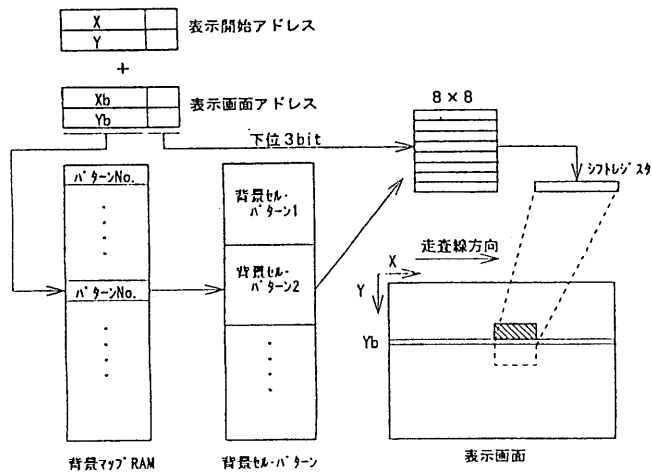


図 スクロール処理