

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5093553号  
(P5093553)

(45) 発行日 平成24年12月12日(2012.12.12)

(24) 登録日 平成24年9月28日(2012.9.28)

(51) Int.Cl.

A63F 7/02 (2006.01)

F 1

A 6 3 F 7/02 320

請求項の数 3 (全 39 頁)

(21) 出願番号 特願2006-144731 (P2006-144731)  
 (22) 出願日 平成18年5月25日 (2006.5.25)  
 (65) 公開番号 特開2007-312949 (P2007-312949A)  
 (43) 公開日 平成19年12月6日 (2007.12.6)  
 審査請求日 平成21年5月25日 (2009.5.25)

(73) 特許権者 000148922  
 株式会社大一商会  
 愛知県北名古屋市沖村西ノ川1番地  
 (74) 代理人 100084227  
 弁理士 今崎 一司  
 (74) 代理人 100174182  
 弁理士 古田 広人  
 (72) 発明者 市原 高明  
 愛知県北名古屋市沖村西ノ川1番地株式会  
 社大万内  
 (72) 発明者 後藤 将仁  
 愛知県北名古屋市沖村西ノ川1番地株式会  
 社大万内

審査官 河本 明彦

最終頁に続く

(54) 【発明の名称】遊技機

## (57) 【特許請求の範囲】

## 【請求項 1】

遊技盤面に設けられた表示装置の表示領域に遊技中に所定の演出表示を行う遊技機であつて、

前記遊技の状況に応じて前記演出表示を制御する表示コマンドを出力する遊技制御手段と、

前記遊技制御手段からの表示コマンドを受けて、前記表示装置を駆動するための表示データを生成し、該表示装置に出力する表示制御手段とを有し、

前記表示制御手段は、

前記表示装置に表示すべき画面の構成を規定する画面データを記憶する画面データ記憶部と、

前記表示領域をN (Nは自然数)分割して分割領域を設定し、前記表示コマンドに応じて、各分割領域に表示すべき画面を決定するとともに、前記画面データに基づいて、予め定めた周期の割り込みが発生するごとに、前記分割数Nの値だけ存在する分割領域のうち、一の分割領域に応じた描画コマンドを出力し、当該分割数Nの値だけ前記予め定めた周期の割り込みが発生すると、すべての分割領域に応じた描画コマンドの出力を完了する描画制御部と、

前記画面に表示される所定のスプライトを前記表示装置の画素単位で表したスプライトデータを記録するキャラクターメモリと、

仮想のキャンバス上に前記描画コマンドで指定されたスプライトデータを指定された位

10

20

置に配置して、前記画面データに対応した画像を描画し、前記キャンバス上に前記一の分割領域に対応した大きさで予め設定された表示エリア内の描画結果に基づいて表示データを生成する表示データ生成部と、

前記生成された前記一の分割領域の表示データを、前記表示領域上の画素と対応づけて記憶するための第1の表示データ記憶部と第2の表示データ記憶部とを有する表示データ記憶部と、

前記予め定めた周期の割り込みが発生するごとに、前記第1の表示データ記憶部と前記第2の表示データ記憶部とのうち、いずれか一方に表示データを格納するとともに、格納完了した表示データを前記表示装置に出力する表示データ管理部とを備え、

前記描画制御部は、前記表示コマンドに基づいて、前記スプライトデータの解像度を低くするときには前記分割数Nの値を小さく設定する一方、前記スプライトデータの解像度を高くするときには当該スプライトデータの解像度を低くするときに設定された前記分割数Nの値と比べて当該分割数Nの値を大きく設定することにより、当該分割数Nを切り換える、

前記表示データ管理部は、前記予め定めた周期の割り込みが発生するごとに、前記表示データ生成部が生成した前記分割数Nの値だけ存在する分割領域のうち、前記一の分割領域の表示データを前記第1の表示データ記憶部に格納し、当該分割数Nの値だけ当該予め定めた周期の割り込みが発生すると、当該分割数Nの値だけ存在する分割領域のすべての表示データを当該第1の表示データ記憶部に格納完了し、次の当該予め定めた周期の割り込みが発生すると、当該表示データ生成部が生成した当該分割数Nの値だけ存在する分割領域のうち、前記一の分割領域の表示データを、当該第1の表示データ記憶部から前記第2の表示データ記憶部へ切り替えて格納開始するとともに、当該第2の表示データ記憶部に当該分割数Nの値だけ存在する分割領域のすべての表示データが格納完了するまでの間、当該第1の表示データ記憶部に格納完了した表示データを前記表示装置に出力する遊技機。

### 【請求項2】

請求項1記載の遊技機であって、

前記描画制御部は、前記スプライトデータの解像度を低くするときには前記分割数Nの値を値1に設定する非分割モードと、前記スプライトデータの解像度を高くするときには前記分割数Nの値を値2以上に設定する分割モードとを切り換える遊技機。

### 【請求項3】

請求項1または2記載の遊技機であって、

前記分割数Nを切り換える際の移行期間において、前記表示データの前記表示データ記憶部への新たな格納が禁止される遊技機。

### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本発明は、遊技盤面に設けられた表示装置の表示領域に遊技中に所定の演出表示を行う画面表示方法に関する。

#### 【背景技術】

#### 【0002】

パチンコ機やスロットマシンなどの遊技機では、遊技盤に備えられた表示装置を用いて、遊技中に種々の演出表示が行われる。この演出表示には、液晶パネルが用いられることが多い。液晶パネルは、マトリックス状に配置された画素によって画像を表示する。画像を表示するための表示データは、例えば、特許文献1記載の手順で生成される。まず、演出表示を制御するためのCPUは、表示コマンドを受け取り、その内容を解析し、予め用意されたスケジュールデータを参照して、表示すべき画面の内容を決定する。そして、このスケジュールデータに基づいて描画コマンドをVDP (Video Display Processor) に出力する。VDPは、この描画コマンドをビットマップ展開して画素単位での表示データ

10

20

30

40

50

を生成し、液晶パネルに出力する。表示データを生成する回路には、生成されたデータを一時的に保持しておくためのバッファが設けられるのが通常である。

### 【0003】

【特許文献1】特開2005-27833号公報

#### 【発明の開示】

##### 【発明が解決しようとする課題】

### 【0004】

近年、遊技機の興趣を高めるため、液晶パネルが大型化し、画素数が増大する傾向にある。このような液晶パネルを用いるためには、その画素数に応じて、VDPの処理能力を向上させることができ望ましい。しかし、画素数は液晶パネルの面積に比例して増大するため、VDPには液晶パネルの面積増に応じた能力向上が要求されることになる。この能力向上にはVDPが表示データを生成する際に使用するメモリ容量や表示データの管理情報を格納するためのレジスタサイズの増大、描画に必要なスプライトのデータを格納するキャラROMの容量、キャラROMへのアクセス速度などが要求され、いずれの面でも能力向上には限界があり、また仮に能力向上が技術的に可能であったとしても、VDPのコスト増大、ひいては遊技機自体のコスト増大という看過できない弊害を招くことになる。

10

### 【0005】

特許文献1では、VDPの能力向上に代えて、スケーラと呼ばれる素子を用いる技術が開示されている。スケーラを用いれば、VDPが生成した1画素分の表示データを、複数画素で共通に使用することによって、表示データを拡大し、大画面の液晶パネルに画像を表示することが可能となる。しかし、この技術は、複数の画素に同じ表示データを出力するため、液晶パネルの解像度を実質的に低下させ、表示される画像の画質を低下させることになる。

20

### 【0006】

このように、従来の技術では、VDPの能力およびコストを抑制しつつ、十分に高い解像度で大画面の液晶パネルへの画像表示を行うことができなかった。かかる課題は、液晶パネルを用いる場合のみならず、種々の表示装置に共通の課題であった。本発明は、この課題の解決を図り、遊技機において、VDPの能力増を要さず、高解像度の表示装置への画像表示を実現することを目的とする。

#### 【課題を解決するための手段】

30

### 【0007】

本発明は、遊技盤面に設けられた表示装置の表示領域に遊技中に所定の演出表示を行う遊技機を対象とする。遊技機には、パチンコ機や回胴式遊技機などが該当する。表示装置は、液晶パネル、プラズマディスプレイ、有機ELなどを利用可能である。遊技機には、遊技の状況に応じて演出表示を制御する表示コマンドを出力するサブ制御基板と、サブ制御基板からの表示コマンドを受けて、表示装置を駆動する表示制御基板とが備えられている。この他に、遊技機全体の制御を統合する主制御基板、賞球やメダル等の払出しを制御する払出基板などを備えてよい。

### 【0008】

本発明の表示制御基板は、表示装置への演出表示を実現するために画面データ記憶部、描画制御部、キャラクターメモリ、表示データ生成部、表示データ管理部が設けられている。画面データ記憶部は、表示装置に表示すべき画面の構成を規定する画面データを記憶している。画面データは、表示コマンドと対応づけられているため、画面データを用いることによって表示コマンドに応じた画面が決まることになる。描画制御部は、大きく2つの機能を実現する。第1に、描画制御部は、表示コマンドに基づいて表示領域をN（Nは自然数）分割して分割領域を設定する。本明細書では、「分割」とは、分割領域と表示領域とが同一となる場合も含む広い概念で用いる。描画制御部は、第2に、表示コマンドに応じて、各分割領域に表示すべき画面を決定するとともに、画面データに基づいて、予め定めた周期の割り込みが発生するごとに、分割数Nの値だけ存在する分割領域のうち、一つの分割領域に応じた描画コマンドを出力し、分割数Nの値だけ予め定めた周期の割り込み

40

50

が発生すると、すべての分割領域に応じた描画コマンドの出力を完了する。例えば、分割領域が表示領域と一致している場合には、表示領域に表示すべき画面全体を描画するための描画コマンドを出力する。また、表示領域を2分割して分割領域を定義している場合には、1回目の処理で一方の分割領域を描画するための描画コマンドを出力し、2回目の処理で他方の分割領域を描画するための描画コマンドを出力する。3分割以上されている場合も同様である。

#### 【0009】

ここで、描画コマンドとは、画面を、表示装置の画素単位で展開したデータを生成するためのコマンドである。本発明では、キャラクターメモリに、画面に表示される所定のスプライトを表示装置の画素単位で表したスプライトデータが記録されている。描画コマンドには、例えば、このスプライトの配置、複数のスプライトの重ね合わせ方が含まれる。スプライト以外の図形や線分の描画を指定するコマンドを含めても良い。

10

#### 【0010】

本明細書では、「キャラクター」および「スプライト」を次の意味で用いる。スプライトとは、遊技機の画面にまとまった単位として表示されるイメージを意味する。例えば、画面上に種々の人物を表示させる場合には、それぞれの人物を描くためのデータを「スプライト」と呼ぶ。複数の人物を表示させるためには、複数のスプライトを用いることになる。人物のみならず背景画像を構成する家、山、道路などをそれぞれスプライトとすることもできる。また、背景画像全体を一つのスプライトとしてもよい。遊技機は、これらの各スプライトの画面上の配置を決め、スプライト同士が重なる場合の上下関係を決めることで、種々の画像を表示させることができる。

20

#### 【0011】

遊技機では、データを扱う便宜上、各スプライトは縦横それぞれ64ピクセルなど一定の大きさの矩形領域を複数組み合わせて構成される。この矩形領域を描くためのデータを「キャラクター」と呼ぶ。小さなスプライトの場合は、一つのキャラクターで表現することができるし、人物など比較的大きいスプライトの場合には、例えば、横2×縦3などで配置した合計6個のキャラクターで表現することができる。背景画像のように更に大きいスプライトであれば、更に多数のキャラクターを用いて表現することができる。キャラクターの数および配置は、スプライトごとに任意に指定可能である。

30

#### 【0012】

表示データ生成部は、仮想のキャンバス上に描画コマンドで指定されたスプライトデータを指定された位置に配置して、画面データに対応した画像を表す表示データを生成する。表示データは、ビットマップ展開された画像のうち、キャンバス上に分割領域に対応した大きさで予め設定された表示エリア内の描画結果に基づいて生成される。本発明では、このキャンバスのサイズは、各分割領域よりも大きく表示領域を構成する全画素数よりも縦横の一方が小さい。分割領域よりも大きいサイズとすることにより、いずれの分割領域についても、それぞれキャンバス上に画像を展開することが可能となる。表示領域の画素数よりも小さいサイズとすることにより、表示データ生成部に要求される処理能力を抑制することができる。キャンバスのサイズは、表示データ生成部の能力によって定まるものである。例えば、表示データ生成部が描画用に有するメモリ容量上の制約で定まる場合もあれば、キャラクターメモリへのアクセスや描画処理の速度上の制約で定まる場合もある。また、表示データの管理情報を格納するためのレジスタサイズ上の制約で定まる場合もある。キャンバスのサイズは、表示データ生成部の物理的な性能の制約とは無関係に、設定された値であってもよい。キャンバスは、仮想的なものであるため、必ずしもフレームメモリのようにキャンバス全体の画素を記憶可能なメモリが物理的に用意されている必要はない。

40

#### 【0013】

表示データ記憶部には、こうして生成された各分割領域の表示データを、表示領域上の画素と対応づけて記憶するための第1の表示データ記憶部と第2の表示データ記憶部とを

50

有する。つまり、表示データ記憶部の第1の表示データ記憶部と第2の表示データ記憶部とには、表示領域に表示すべき画面全体に対応したデータが格納可能であり、各分割領域の表示データが生成されるたびに、画面のデータが部分的に満たされていき、全ての分割領域についての処理が完了すると、画面全体のデータが満たされることになる。表示データ管理部は、予め定めた周期の割り込みが発生するごとに、第1の表示データ記憶部と第2の表示データ記憶部とのうち、いずれか一方に表示データを格納するとともに、格納完了した表示データを表示装置に出力し、予め定めた周期の割り込みが発生するごとに、表示データ生成部が生成した分割数Nの値だけ存在する分割領域のうち、一の分割領域の表示データを第1の表示データ記憶部に格納し、分割数Nの値だけ予め定めた周期の割り込みが発生すると、分割数Nの値だけ存在する分割領域のすべての表示データを第1の表示データ記憶部に格納完了し、次の予め定めた周期の割り込みが発生すると、表示データ生成部が生成した分割数Nの値だけ存在する分割領域のうち、一の分割領域の表示データを、第1の表示データ記憶部から第2の表示データ記憶部へ切り替えて格納開始するとともに、第2の表示データ記憶部に分割数Nの値だけ存在する分割領域のすべての表示データが格納完了するまでの間、第1の表示データ記憶部に格納完了した表示データを表示装置に出力する。表示データと表示領域上の各画素との対応関係は1対1に限らず、i対j(i, jは自然数)でよい。1対1以外の対応関係では、表示装置への出力時に、表示領域上の画素数に適合するよう表示データを拡大または縮小すればよい。表示データの出力タイミングは、種々の設定が可能である。例えば、画面の一部のデータが満たされた時点での出力を開始してもよい。この場合には、表示データの格納と、出力とを並行して行うことになる。

10

20

#### 【0014】

本発明では、描画制御部は、更に表示コマンドに基づいて、スプライトデータの解像度を低くするときには分割数Nの値を小さく設定する一方、スプライトデータの解像度を高くするときにはスプライトデータの解像度を低くするときに設定された分割数Nの値と比べて分割数Nの値を大きく設定することにより、分割数Nを切り換える。遊技機では、多彩な演出表示が行われる。例えば、パチンコ機の場合、リーチ後には画像 자체を楽しませる内容の画面よりも、画面が移り変わるスピード感によって興奮を高めることが多い。一方、その他の場面では、画像 자체を楽しませる内容の画面が用いられることがある。前者では、表示される各フレームの画面は、比較的簡易な処理で生成することが好ましく、後者では、高解像度で高画質の画面を生成することが好ましい。本発明では、表示コマンドに基づいて、分割数Nを切り換えることにより、狙いの異なる演出画面にかかる切り換えを実現している。

30

#### 【0015】

分割数Nの切り換えには次の利点がある。分割数Nの切り換えは、分割領域のサイズ変更、ひいては表示データ生成装置が表示データを生成する際の解像度の変更を意味する。解像度が高い場合には、画質が向上するものの、スプライトデータの解像度も同様に高める必要があるため、キャラクターメモリの容量が増大する。また、表示データの画素数が増大するため、表示データ生成時の処理負荷が増大し、所要時間が増大する。解像度が低い場合には、逆に、画質が低下するものの、キャラクターメモリの容量は抑制でき、所要時間も抑制することができる。従って、比較的高速に移り変わる画面を表示する場合には、分割数Nを低減することにより、低容量のキャラクターメモリを用い、短時間で表示データを生成することが可能となる。逆に、高画質が要求される画面を表示する場合には、分割数Nを増大させればよい。

40

#### 【0016】

本発明では、一例として、分割の数Nが1の非分割モードと、分割数Nが2以上の分割モードとを切り換える態様を探ることができる。分割モードで用いる分割数Nは1通りであってもよいし、更に複数の分割数に切り換えてよい。非分割モードおよび分割モードの使い分けは任意に決定可能である。一例として、画質への要求が低い画面に対して非分

50

割モードを用い、高画質が要求される画面に対して分割モードを用いるように使い分けることができる。

【0017】

先に説明した通り、表示装置は種々の装置を用いることが可能であるが、例えば、画素の配列に沿った一方に各画素の表示状態を設定する主走査と、一方と交差する方向に主走査を繰り返し実行する副走査とによって駆動される装置とすることができる。この場合において分割数  $N$  が 2 以上の場合は、分割領域は、主走査が行われる一方（以下、「主走査方向」と呼ぶ）に平行な境界線で定義することが好ましい。例えば、左右方向に主走査が行われる場合には、左右方向の境界線によって分割領域を定義することが好ましく、各領域は上下方向に配置されることになる。分割領域は、更に、主走査に交差する方向（以下、「副走査方向」と呼ぶ）の境界線を含めて定義しても良い。例えば、上述の場合において、上下方向に配置された各領域を更に、左右に分割して各分割領域を定義してもよい。

10

【0018】

いずれの場合においても、主走査方向に平行な境界線で分割領域を定義することにより、主走査方向に配列された一列分の画素（以下、「ラスタ」と呼ぶ）に対応するデータを速やかに生成することができる。仮に主走査方向に交差する境界線で分割領域が定義されている場合には、1 ラスタ分のデータを用意するために、複数回の表示データ生成を繰り返す必要が生じるが、上述の態様では表示データの生成が行われる度に、速やかに 1 ラスタ分の表示データが完成することになる。この結果、各ラスタの表示データの格納、全ラスタの表示データの生成が完了したか否かの判断その他の管理が容易となる利点がある。また、表示データの格納と出力とを並行して行うことが可能となる利点もある。

20

【0019】

分割数が 2 以上の場合は、表示データ記憶部へのデータ格納の際、表示データと分割領域との対応関係は、表示データ管理部が自律的に判断可能としてもよい。表示データ管理部にとって、各分割領域の位置および処理順序が既知となっている場合には、この方法が可能である。これに対し、描画制御部が、描画コマンドを表示データ生成部に出力する際に、フィールド信号、即ち描画コマンドに対応する分割領域を特定するための制御信号を、表示データ管理部に出力するようにしてもよい。予め分割領域に付されたインデックスを表す信号をフィールド信号として用いることができる。こうすることにより、表示データ管理部は、容易かつ確実に、表示データと分割領域との対応関係を把握することが可能となる。

30

【0020】

分割数  $N$  が 2 以上の場合には、先に説明した通り、分割領域ごとに表示データを生成し、これらの表示データを集めて画面全体に対応する表示データを生成する。これは、キャンバスに展開すべき描画コマンドが分割領域ごとに異なることを意味する。分割領域ごとに異なる描画コマンドを表示データ生成装置に出力するためには、種々の方法を探ることができる。

【0021】

第 1 の態様として、分割領域が同一の形状となっている場合には、描画制御部は、分割領域に関わらず、画面の全体を描画するための描画コマンドを出力する。この際、描画コマンドで指定されるスプライトデータの位置と表示エリアとの相対的な位置関係を分割領域に応じて変更する。描画コマンドに含まれる座標値、表示エリアの位置を規定する座標値の一方または双方を変更すればよい。この態様では、描画コマンドを複数の分割領域で共通して使うことができるため、用意しておくべき画面データの容量を抑制することができる。

40

【0022】

第 2 の態様として、分割領域が同一の形状となっている場合には、描画制御部は、分割領域のうち最初の分割領域に対して、画面の全体を描画するための描画コマンドを出力し、2 番目以降の分割領域に対しては、キャンバスにおける表示エリアの位置を変更するコ

50

マンドを出力するようにしてもよい。この態様では、分割領域の位置に応じて表示エリアの位置を変更するため、描画コマンドは2以上の分割領域で共通化することができる。もっとも、この態様においても、2番目以降の分割領域においても、適宜、描画コマンドを出力しても差し支えない。

#### 【0023】

第3の態様として、画面データ記憶部に、一つの画面に対し分割領域単位で画面の一部を表す複数通りの画面データを用意しておいてもよい。画面データは、分割領域ごとに個別に用意してもよいし、2以上の分割領域に共通のデータとしてもよい。このように複数通りの画面データが用意されている場合、描画制御部が、分割領域に対応した画面データを選択して描画コマンドを出力することにより、座標値の修正などの処理を要するまでなく、各分割領域に応じた表示データを生成することができる。

10

#### 【0024】

分割数Nが2以上の場合、表示データの生成時には、描画コマンドとは別に、分割領域の全てに共通して描画時の条件を設定する条件設定コマンドが要求されることがある。条件設定コマンドには、例えば、描画コマンドが複数のレイヤから構成されている場合に、各レイヤの重ね合わせ順序、表示/非表示の設定などが含まれる。このような場合には、描画制御部が、条件設定コマンドを生成して表示データ生成部に出力すればよい。条件設定コマンドの出力は種々のタイミングで行うことができる。一例として、最初の分割領域に対する描画コマンドを出力した後に出力する方法を探ることができる。こうすることにより、描画コマンドとほぼ同期して条件設定コマンドを表示データ生成部に受け渡すことができる利点がある。

20

#### 【0025】

また、上述のタイミングで条件設定コマンドを出力するか否かに関わらず、条件設定コマンドは、2番目以降の分割領域のうち少なくとも一つに対する処理時を除いて出力することが好み。つまり、最初の分割領域の処理時には必然的に出力されるが、2番目以降の分割領域の処理時には、一部または全部で出力が省略される。こうすることで表示データ生成部へのデータ出力を簡素化することができる。

#### 【0026】

分割数Nの切り換えを行う際の移行期間においては、表示データ記憶部に新たに表示データを格納することを禁止することが望ましい。分割数Nが切り換えられると、表示データのサイズ等が変化するため、一時期、描画コマンドの出力、表示データの出力、格納等において不安定な時期が生じるおそれがある。上述の態様によれば、切り換えを行う際の移行期間においては、新たなデータ格納を禁止するため、この期間では、表示データ記憶部に従前の処理で記憶済みの表示データを用いて画面の表示が行われる。従って、表示データの出力および格納等が不安定な時期においても、画面表示に違和感が生じることを回避できる。ここで、移行時期は、分割数Nの切り換えが完了するまでに許容される時間範囲、および切り換え時に画面に生じる違和感の双方における影響を考慮して、任意に設定可能である。

30

#### 【0027】

表示データの格納禁止には、種々の方法を探ることができる。例えば、表示データ管理部が、表示データ生成部から出力される所定の同期信号をトリガとして表示データの格納を開始するよう構成されている場合には、この同期信号が表示データ管理部に伝達されるのを停止してもよい。例えば、同期信号を表示データ生成部が出力している場合には、その出力自体を停止してもよい。また、表示データ生成部と表示データ管理部との間の同期信号の伝達経路に、信号の伝達可否を切り換えるスイッチ回路を設け、描画制御部がこのスイッチ回路のオン・オフを制御する方法を探っても良い。

40

#### 【0028】

本発明では、表示データ生成部は、分割数Nの切り換えに応じて、表示データを出力する周期を切り換えるようにしてもよい。分割数Nの切り換えによって、表示データ生成部が1回の処理で生成すべき表示データのサイズが変動する。従って、周期を切り換えるこ

50

とにより、表示データ生成部の処理能力の制限に関わらず、柔軟に分割数Nを設定可能となる。表示データの出力周期は任意に設定可能であるが、特に、分割数N×周期が一定となるよう切り換えることが好ましい。こうすることにより、各フレームを表示するために必要となる表示データは、一定の周期でそろうことになり、表示装置への表示データの出力制御の簡素化、安定化を図ることが可能となる。

【0029】

表示データ記憶部は、複数枚の画面に対応する表示データを格納可能、即ち複数フレーム分の容量を備えるようにしてもよい。この場合、描画制御部は、分割数Nを切り換える際の移行期間においては、複数枚の画面に対応する表示データが同一となるよう、描画コマンドの出力を制御することが好ましい。先に説明した通り、移行期間においては、描画コマンドの出力、表示データの生成、格納が不安定になることがある。かかる状況で、複数フレーム分の記憶領域に、異なる表示データを格納していると、これらの表示データが表示装置に出力される順序によっては、表示画面の内容が不自然に変動し、違和感を与える画面となるおそれがある。上述の通り、複数のフレームに格納される表示データを同一のものとしておけば、かかる弊害が生じる可能性を抑制することができる。

10

【0030】

複数フレーム分の表示データを同一とする制御は種々の態様で実現できる。第1の方法として、同一の描画コマンドを繰り返し表示データ生成部に出力してもよい。第2の方法として、移行期間は描画コマンドの出力を保留し、表示データ生成部に従前に出力済みの描画コマンドに基づく表示データを繰り返し生成させてもよい。

20

【0031】

本発明では、表示データ記憶部は、分割数Nの値に応じて個別に設けられた領域に、表示データを格納するよう構成することが好ましい。分割数Nが変わると表示データ生成装置から出力される表示データのサイズおよびその後の管理方法も変動するため、格納領域を変えることにより表示データの格納、管理を安定的に行うことが可能となる。分割数Nに応じた格納領域の使い分けは、表示データ記憶部が自律的に行うようにしてもよいし、描画制御部が表示データを格納すべき格納領域を指定するようにしてもよい。

【0032】

本発明では、上述した種々の特徴を全て備えている必要はなく、一部を省略してもよいし、適宜、組み合わせて適用してもよい。また、本発明における上述の特徴部分は、ハードウェア的に実現してもよいし、ソフトウェア的に実現してもよい。

30

【発明を実施するための最良の形態】

【0033】

本発明の実施例について以下の順序で説明する。本実施例では、パチンコ機としての構成例を示すが、遊技機は、回胴式遊技機としてもよい。

A. 遊技機の構成：

B. 制御用ハードウェア構成：

C. 表示制御処理：

D. 非分割表示制御処理：

E 1. 分割表示制御処理：

E 2. 分割表示制御処理の変形例：

F. 分割移行処理：

G. 非分割移行処理：

40

【0034】

A. 遊技機の構成：

図1は実施例としてのパチンコ機1の正面図である。パチンコ機1は、中央に遊技領域6を備えた遊技盤4が取り付けられている。遊技者は、ハンドル8を操作して遊技領域6内に遊技球を打ち込み、入賞口に入賞させる遊技を行うことができる。入賞口の一つである始動入賞口9に遊技球が入賞すると、パチンコ機1は抽選を行い、その結果に応じて「大当たり」か否かが決まる。大当たり発生時には、大入賞口10が所定期間開放するなどの大

50

当たり遊技が行われる。

【0035】

上述の抽選の結果は、4つのランプで構成された特別図柄表示装置41に表示される。遊技領域6の中央には、LCD16が備えられており、遊技中に種々の演出画面（装飾図柄と呼ぶこともある）が表示される。始動入賞口9への入賞時、大当たりの発生時などにも、それぞれ遊技の状態に応じた演出画面が表示される。装飾図柄は、遊技の状況に応じて種々変化する。抽選結果を表示する過程で、大当たりが発生する可能性が高いリーチ状態になると、スピード感あふれる画面が表示され、遊技者をハラハラドキドキさせることによって興奮を高めるようになっている。また、リーチ以前の状態や、遊技が行われていない時のデモ画面では、種々のスプライトを組み合わせて画面自体にも楽しめる要素を多分に含んだ画面が表示される。装飾図柄には、このように主としてスピード感が重視される画像（以下、「高速画像」という）と、主として画質が重視される画像（以下、「高画質画像」という）とが含まれる。以下の実施例は、スピード感および画質のいずれを重視すべきか微妙なものも含めて、高速画像、高画質画像のいずれかに分類したとして説明する。本実施例では、この分類に応じて2つの表示モードを使い分けるからである。装飾図柄を3通り以上に分類し、3つ以上の表示モードを使い分けるようにしてもよい。

10

【0036】

B. 制御用ハードウェア構成：

図2はパチンコ機1の制御用ハードウェア構成を示すブロック図である。パチンコ機1は、メイン制御基板3、払出制御基板25、サブ制御基板35、装飾図柄制御基板30などの各制御基板の分散処理によって制御される。メイン制御基板3、払出制御基板25、サブ制御基板35は、それぞれ内部にCPU、RAM、ROMなどを備えたマイクロコンピュータとして構成されており、ROMに記録されたプログラムに従って種々の制御処理を実現する。

20

【0037】

実施例のパチンコ機1では、種々の不正を防止するため、メイン制御基板3への外部からの入力が制限されている。メイン制御基板3とサブ制御基板35とは単方向のパラレル電気信号で接続されており、メイン制御基板3と払出制御基板25とは、制御処理の必要上、双方向シリアル電気信号で接続されている。払出制御基板25、サブ制御基板35は、それぞれメイン制御基板3からのコマンドに応じて動作する。装飾図柄制御基板30は、サブ制御基板35からのコマンドに応じて動作する。パチンコ機1には、メイン制御基板3が直接に制御する機構もある。図中には、メイン制御基板3が制御する装置の一例として、大入賞口10を駆動するための大入賞口ソレノイド18、および特別図柄表示装置41を例示した。メイン制御基板3は、この他にも、普通図柄表示装置、特別図柄保留ランプ、普通図柄保留ランプ、大当たり種類表示ランプ、状態表示ランプなどの表示を制御することができる。また、メイン制御基板3には、遊技中の動作を制御するため、種々のセンサからの検出信号が入力される。図中には一例として入賞検出器15aからの入力を例示した。入賞検出器15aとは、始動入賞口9への入賞を検出するためのセンサである。メイン制御基板3は、入賞検出器15aからの信号に応じて、先に説明した抽選を行い、大当たり遊技を実行することができる。メイン制御基板3には、他にも種々の入力がなされているが、ここでは説明を省略する。

30

【0038】

遊技時におけるその他の制御は、払出制御基板25、サブ制御基板35を介して行われる。払出制御基板25は、遊技中の遊技球の発射および払い出しを次の手順で制御する。遊技球の発射は、直接的には発射制御基板47によって制御される。即ち、遊技者が、発射ハンドル8を操作すると、発射制御基板47は操作に応じて発射モータ49を制御し、遊技球を発射する。遊技球の発射は、タッチ検出部48によって、発射ハンドル8に遊技者が触れていることが検出されている状況下でのみ行われる。払出制御基板25は、発射制御基板47に対して、発射可否の制御信号を送出することで、間接的に球の発射を制御する。

40

50

## 【0039】

遊技中に入賞した旨のコマンドをメイン制御基板3から受信すると、払出制御基板25は、賞球払出装置21内の払出モータ20を制御し、払出球検出器22によって球数をカウントしながら規定数の球を払い出す。払出モータ20の動作は、モータ駆動センサ24によって監視されており、球ガミ、球切れなどの異常が検出された場合、払出制御基板25は、表示部4aにエラーコードを表示する。エラー表示された時には、係員が異常を除去した後、操作スイッチ4bを操作することで復旧させることができる。

## 【0040】

サブ制御基板35は、遊技中における音声、表示、ランプ点灯などの演出を制御する。これらの演出は、通常時、入賞時、大当たり時、エラー時、不正行為その他の異常が生じた時の警報など、遊技中のステータスに応じて変化する。メイン制御基板3から、各ステータスに応じた演出用のコマンドが送信されると、サブ制御基板35は、各コマンドに対応したプログラムを起動して、メイン制御基板3から指示された演出を実現する。

10

## 【0041】

本実施例では、図示する通り、サブ制御基板35はスピーカ29を直接制御する。LCD16は、装飾図柄制御基板30を介して制御する。装飾図柄制御基板30の回路構成は後述する。サブ制御基板35の制御対象となるランプには、遊技盤面に設けられたパネル装飾ランプ12と、枠に設けられた枠装飾ランプ31がある。サブ制御基板35は、ランプ中継基板32、34を介して、パネル装飾ランプ12および枠装飾ランプ31と接続されており、各ランプを個別に点滅させることができる。

20

## 【0042】

図3は装飾図柄制御基板30の回路構成を示す説明図である。装飾図柄制御基板30は、サブ制御基板35から受けた表示コマンドに応じて、LCD16に画面を表示するための駆動データを出力する。駆動データは、LCD16にマトリックス状に備えられたR、G、Bの各画素の表示階調値を示すデータである。本実施例のLCD16は左右方向に800画素、上下方向に600画素を有している。駆動データは、主走査として、LCD16の左から右に順次、画素ごとに出力される。一ラスタ分の出力が完了すると、副走査として直下のラスタに移行し、同様に左から右に向けて順次、画素ごとに駆動データが出力される。以下では、左右方向を主走査方向と呼び、上下方向を副走査方向と呼ぶこともある。

30

## 【0043】

装飾図柄制御基板30には、表示コマンドに応じた駆動データを生成する機能を実現するため図示する種々の回路が用意されている。装飾図柄制御基板30には、まず、駆動データの生成を制御するためのマイクロコンピュータとしてCPU381、RAM382、ROM383が備えられている。ROM383には、駆動データを生成するための表示プログラム、表示コマンドに対し表示すべき画面、表示の時間、表示の順序を規定するスケジューラ、各画面の構成を規定する画面データが記憶されている。画面データの内容については後述するが、この段階では、LCD16の画素に対応したデータとはなっていない。CPU381は、ROM383を参照して、表示コマンドに応じた画面データを抽出し、VDP(Video Display Processor)385に出力する。

40

## 【0044】

キャラROM386は、スプライトデータ、即ち画面に表示されるスプライトをビットマップで表したデータを格納している。VDP385は、CPU381から受け取った画面データに基づいて、表示すべきスプライトデータをキャラROM386から抽出し、表示データ、即ち表示すべき画像をビットマップ展開したデータを生成して、スケーラ390に出力する。以下、ビットマップ展開するための記憶領域を「キャンバス」と呼ぶ。

## 【0045】

VDP385には、CPU381からの画面データを受け取り保持しておくためのレジスタとして、スプライトレジスタ385sおよびVDPレジスタ385vを備えている。スプライトレジスタ385sは、画面データのうち、スプライトの配置や重ね合わせの順

50

序などを示す描画コマンドを受け取るためのレジスタであり、ダブルバッファとして構成されている。つまり、第1バッファ、第2バッファという二つの等しい容量のバッファが備えられている。従って、VDP385は、CPU381から出力された描画コマンドが第1バッファに書き込まれている間、第2バッファに保持されている描画コマンドを読み出して表示データの生成処理を実行することができる。VDPレジスタ385vは、表示データを生成する際の条件設定を指定するコマンド（以下、「条件設定コマンド」と呼ぶ）を記憶するためのレジスタである。条件設定コマンドには、例えば、描画コマンドが複数のレイヤから構成されている場合に、各レイヤの重ね合わせ順序、表示／非表示の設定などが含まれる。条件設定コマンドは比較的低容量であり、書き込みの所要時間が短いことから、VDPレジスタ385vはダブルバッファとはなっていない。

10

#### 【0046】

スケーラ390は、VDP385からの表示データをフレームメモリ397に格納する。また、フレームメモリ397から表示データを読み出し、LCD16に出力する。図示する通り、本実施例では、フレームメモリ397の内部は397[0]～397[5]の6つのフィールドに分かれている。4つのフィールド397[0]～397[3]は800×300画素分を単位とするメモリ領域（以下、「SVGAフィールド」と呼ぶこともある）である。フィールド397[4]、397[5]は640×480画素分を単位とするメモリ領域（以下、「VGAフィールド」と呼ぶこともある）である。このようにサイズの異なる2種類のフィールドが用意されている意義について説明する。

#### 【0047】

本実施例では、VDP385は2つの表示モードでLCD16の表示データを生成する。一つは、表示データを、640×480画素のVGAサイズで生成するモードである。このモードでは、VGAサイズの表示データをスケーラ390で800×300画素に拡大してLCD16に出力する。以下、この表示モードを非分割モードと称する。非分割モードは、VGAサイズの画像を拡大してLCD16に表示するため、1フレーム分の表示データを速やかに準備することができる反面、画質が比較的低いという特徴がある。本実施例では、高速画像の表示に非分割モードを用いる。

20

#### 【0048】

2つ目のモードは、LCD16の表示データを2回に分けて生成する表示モードである。本実施例では、図中に示すようにLCD16の表示領域を中央でそれぞれ800×300ドットの分割領域に分割する。以下、上側の領域を上画像、下側の領域を下画像と呼ぶものとする。VDP385はまず上画像の表示データを生成し、フレームメモリ397に格納し、次に下画像の表示データを生成して、フレームメモリ397に格納する。スケーラ390は、上画像と下画像の両方の表示データが格納された時点で、両者をLCD16に順次、出力する。こうすることで、VDP385が800×600ドットの表示データを一度に生成するだけの能力を有しない場合でも、解像度を低下させることなく画面を表示させることが可能となるのである。スケーラ390は、本来、LCD16への出力過程で、VDP385によって生成された表示データの画素数を増減しLCD16の画素数に適合させたり、いわゆるインターレース形式で生成された表示データをノンインターレース形式に形式変換したりするための素子であり、フレームメモリ397の内部を4つのフィールドに分けて管理する機能も、本来は画素数の増減や形式変換を実現するために用意されている機能である。本実施例では、この機能を以下で説明する通り流用することで、分割して生成された表示データを結合して1画面分の表示データを生成する機能を実現する。以下、このモードを分割モードと称する。分割モードは、高解像度で表示データを生成可能であるため、本実施例では、高画質データの表示に用いる。

30

#### 【0049】

フレームメモリ397に設けられた各フィールド397[0]～397[3]には、分割モードにおいてVDP385によって1回に生成される表示データを格納することができる。フィールド397[0]、397[2]はそれぞれ上画像の表示データを格納し、フィールド397[1]、397[3]はそれぞれ下画像の表示データを格納する。従つ

40

50

て、フィールド397[0]、397[1]には併せて1フレーム分のデータが格納され、フィールド397[2]、397[3]にも併せて1フレーム分のデータが格納される。つまり、フレームメモリ397は、フィールド397[0]～397[3]を用いて分割モードにおける2フレーム分の表示データを格納することができる。

#### 【0050】

フィールド397[4]、397[5]は $640 \times 480$ 画素の画像データ、即ち非分割モードで生成される表示データを格納可能である。つまり、フレームメモリ397は、フィールド397[4]、397[5]を用いて、非分割モードにおける2フレーム分の表示データを格納することができる。このように、本実施例では、分割モード、非分割モードでそれぞれフレームメモリ397内に個別に用意されたフィールドを用いている。こうすることにより、表示モードを切り換える過程を含めて、表示データの簡易かつ安定的に管理することができる。もっとも、このように表示モードによってフィールドを使い分けることは必然的な構成ではなく、双方のモードで共通のフィールドに表示データを格納する構成としてもよい。また、本実施例では、それぞれの表示モードで2フレーム分の表示データを格納可能としたが、1フレーム分のデータのみを格納可能としてもよいし、3フレーム以上のデータを格納可能としてもよい。

10

#### 【0051】

フレームメモリ397の各フィールド397[0]～397[5]に表示データを誤り無く格納するため、本実施例では、CPU381がVDP385に描画コマンドを出力する際に、フィールド信号を生成し、スケーラ390に出力する。フィールド信号は、非分割モードにおいてはフレームの切り換えを表す信号となり、分割モードにおいては、上画像／下画像のいずれの描画コマンドを示す信号ともなる。スケーラ390は、この信号に応じて内部的に制御信号IFLDを生成し、フィールド397[0]～397[5]のうち、VDP385からの表示データを格納すべきフィールドを特定する。

20

#### 【0052】

VDP385からスケーラ390には、表示データと併せて同期信号VSYNCが出力される。非分割モードでは、同期信号VSYNCは1フレーム分の表示データが生成される度に出力され、分割モードでは上画像、下画像が生成される度に出力される。本実施例では、非分割モードのデータは32msc周期で生成し、分割モードにおける上画像、下画像の生成は、16msc周期で行うものとした。この同期信号VSYNCは、信号停止回路389を介してスケーラ390に出力される。信号停止回路389は、CPU381が出力する制御信号VENに応じて、同期信号VSYNCの伝達／停止を切り換え可能な回路である。本実施例では、東芝社製のTC7SZ125AFE（商標）なる素子を用いた。この素子は、A端子、Y端子、G端子を有しており、G端子がロウの時のみA端子の入力信号をY端子から出力する機能を有している。この素子を用いる場合には、A端子をVDP385、Y端子をスケーラ390、G端子をCPU381に接続すればよい。本実施例では、同期信号VSYNCの伝達／停止の切り換えは、後述する通り、表示モードの移行時期に行う。本実施例では、このようにハードウェア的に伝達／停止を切り換えたが、この切り換えはVDP385からの同期信号VSYNCの出力／停止をソフトウェア的に切り換える方法で実現してもよい。

30

#### 【0053】

スケーラ390は、LCD16に16mscのフレームレートで同期信号および駆動データを出力し、LCD16の画像を更新する。同期信号は、スケーラ390自身が生成する。先に説明した通り、本実施例では、1フレーム分の表示データは32msc周期でしか更新されない。非分割モードでは表示データの生成は32mscで行われ、分割モードでは、上画像、下画像の両方の表示データをそろえるためには、16mscの処理を2回行う必要があるからである。従って、本実施例では、スケーラ390は、既に生成済みの表示データを、フレームメモリ397から16mscで読み出し、LCD16に出力する処理を2回繰り返すことにより、同一画像を16mscで2回表示しつつ、表示内容を32mscで更新しながらLCD16への表示を行う。

40

50

## 【0054】

本実施例のLCD16は800×600画素を備えている。従って、分割モードでは、スケーラ390は上画像／下画像の表示データをそのまま駆動データとして出力すればよい。非分割モードでは、表示データが600×480画素で生成されているため、スケーラ390は、この表示データを1.25倍に拡大して、800×600画素相当の駆動データを生成し、LCD16に出力する。このようにスケーラ390はVDP385によって生成された表示データを加工してLCD16に出力してもよい。例えば、スケーラ390は、上画面／下画面に分割して生成された表示データを結合する過程で上画面／下画面の画素数の増減を行っても良いし、結合した後、LCD16に出力する段階で画素数の増減を行っても良い。分割モードでは、原理的には無制限に高解像度の画面を表示することも可能となるが、現実には、高解像度の画面に対応した膨大なキャラクタデータを用意する必要が生じ、キャラROM386の容量が膨大になってしまうという弊害や、キャラROM386へのアクセス回数の増加、スプライトを描画する際の処理負荷の増大などに伴う制約が生じるのが通常である。上述の通り、画面の分割と、スケーラ390本来の機能である画素数の増減とを組み合わせて適用することにより、これらの弊害を抑えながら、高解像度での画面表示を実現することが可能となる。

10

## 【0055】

## C. 表示制御処理：

図4は表示制御処理のフローチャートである。CPU381がVDP385に対して描画コマンド等を出力し、表示データを生成させるために実行する処理である。この処理は、16msの周期の割込処理として実行される。この処理を開始すると、CPU381は、処理を実行するための準備として、多重割込みを許可し(ステップS1)、ノイズキヤンセル・判定処理を行う(ステップS2)。そして、割込端子の端子レベルを確認し(ステップS3)、端子レベルが異常の場合には、ノイズ等の影響による異常なトリガに基づいて表示制御処理が開始されたものと判断し、そのまま処理を終了する。端子レベルが正常の場合には、以下に示す処理によって、描画コマンド、条件設定コマンドの出力を行う。

20

## 【0056】

端子レベルが正常の場合には、CPU381は現在の表示モードが非分割モード／分割モードのいずれであるかを読み込み(ステップS4)、サブ制御基板35からのコマンドに基づいて次に非分割モード／分割モードのいずれの表示モードを探るべきかを設定する(ステップS5)。そして、両者の比較に基づいて表示制御モードを判断する(ステップS6)。表示制御モードは、表示モードに関する制御処理の分類である。非分割モード、分割モードという2つの表示モードの他、非分割モードから分割モードへの移行モード、分割モードから非分割モードへの移行モードが含まれる。CPU381は、現在の表示モードおよび次に探るべき表示モードに応じて、以下の通り4通りの表示制御モードに対応する制御処理を実行する。

30

(現在) 非分割モード (次) 非分割モードの場合、非分割表示制御処理(ステップS100)；

(現在) 分割モード (次) 分割モードの場合、分割表示制御処理(ステップS200)；

(現在) 非分割モード (次) 分割モードの場合、分割移行処理(ステップS300)；

(現在) 分割モード (次) 非分割モードの場合、非分割移行処理(ステップS400)；

40

以下では、各表示制御モードにおける処理内容について順に説明する。

## 【0057】

なお、本実施例では、表示制御処理において上述の表示制御モードを設定し、それぞれ対応する処理に振り分ける制御方法を説明した。各表示制御モードを使い分ける方法は、図4に例示した方法に限らない。本実施例も含め、上述の4つの表示制御モードは、非分

50

割表示制御処理（ステップ S 1 0 0）、分割移行処理（ステップ S 3 0 0）、分割表示制御処理（ステップ S 2 0 0）、非分割移行処理（ステップ S 4 0 0）の順にシーケンシャルに行われるのが通常である。従って、それぞれの表示制御モードを繰り返し実行する過程で、次の表示制御モードに移行するか否かの判定を組み込むようにしてもよい。例えば、非分割表示制御処理（ステップ S 1 0 0）の中に、分割移行処理に移行するか否かの判断処理を組み込めばよい。こうすることにより、上述の処理（図 4）よりも簡易な処理で表示制御モードの使い分けができる利点がある。ただし、上述の処理（図 4）には、表示制御モードを柔軟に切り替え可能という利点がある。例えば、分割モードとして更に、3 分割の表示モードが増えた場合など、表示モード間の移行態様が多様になった場合でも、上述の処理はステップ S 6 の分岐条件を修正することにより、比較的容易に対応できる利点がある。10

#### 【 0 0 5 8 】

##### D. 非分割表示制御処理：

図 5 は非分割表示処理における表示データの生成例を示す説明図である。図の右下に示す画像（640 × 480 ドット）（以下、「VGA 画像」という）を表示する手順を示した。図の左側には、VDP 385 内での処理概要を示し、図の右側にはフレームメモリ 397 に格納されるデータ内容を示した。図の左側に示すように、VDP 385 内では、描画データをビットマップ展開するためのキャンバス CV01 が用意されている。回路上では、キャンバス CV01 内の全画素に対応するメモリ領域が用意されることになる。キャンバス CV01 のサイズは、VGA 画像（640 × 480 ドット）よりも大きく、LCD 16 の表示領域（800 × 600 ドット）よりも小さく縦方向は 512 ドットとなっている。以下、説明の便宜上、キャンバス CV01 の左上を原点として左から右に向かう主走査方向を x 方向、上から下に向かう副走査方向を y 方向と呼ぶこともある。20

#### 【 0 0 5 9 】

本実施例では、キャンバス CV01 に対応する物理的なメモリが用意されている場合を例にとって説明するが、キャンバス CV01 は仮想的なものでもよい。例えば、キャンバス CV01 上への描画をラスタ単位で行う場合には、1 ラスタ分の描画データをビットマップ展開するためのラインメモリ、およびキャンバス CV01 のどのラスタを展開しているのかを管理するためのレジスタを用意しておけば足りる。VDP 385 が描画可能なキャンバス CV01 のサイズは、このレジスタの上限値で決定されることになる。本実施例は、このレジスタの上限値が 512 に制限されている場合に相当すると言ふこともできる。30

#### 【 0 0 6 0 】

キャンバス CV01 内には、VGA 画像相当の表示エリア VA が定義される。VDP 385 は、キャンバス CV01 上に画像をビットマップ展開した後、表示エリア VA 内の画像を切り出し、表示データとしてスケーラ 390 に出力する。本実施例において、キャンバス CV01 を表示エリア VA よりも大きいサイズとしたのは、スプライトが表示エリア VA から一部はみ出す状態で配置された場合でも比較的簡易な処理でビットマップ展開可能とするためである。このような状態でスプライトが配置されている場合の処理方法としては、キャラ ROM から取得したスプライトデータのうち、表示エリア VA 内に存在する部分のみを切り出してビットマップ展開する方法も可能ではある。しかし、この方法では、描画時の処理負荷が非常に高くなる。これに対し、キャンバス CV01 を表示エリア VA よりも大きくしておけば、スプライトが表示エリア VA からはみ出す場合でも、上述の切り出しを行うまでなく、ビットマップ展開できるため、処理の簡素化を図ることができる。こうした効果を重視しない場合は、キャンバス CV01 と表示エリア VA とを同じ形状とすることも可能である。40

#### 【 0 0 6 1 】

CPU 381 から上画像の描画データを受け取り、VDP 385 が画像を描画すると、キャンバス CV02 の状態が得られる。VDP 385 は、キャンバス CV02 から、表示エリア VA 内の画像を切り出し、スケーラ 390 を介して（図示を省略）、表示データと50

して出力する。この表示データは、フレームメモリ 397 の V G A 画像用のフィールド 397 [4] に格納される。この結果、フレームメモリ 397 内には、状態 F M 0 1 で示す内容の表示データが格納されることになる。この表示データを出力すれば、L C D 1 6 には右下に示した画像を表示させることができる。ただし、本実施例では、スケーラ 390 は、この表示データを L C D 1 6 の 800 × 600 画素相当のサイズとなるよう 1.25 倍に拡大して出力する。

#### 【 0 0 6 2 】

図 6 は非分割モードにおける V D P 385 への画面データ出力の様子を示すタイミングチャートである。図 5 で説明した手順で V G A 画像の表示データがフレームメモリに格納され、L C D 1 6 に出力される様子を示した。V D P 385 のスプライトレジスタ 385 s には、32 msec 周期で繰り返し描画コマンドが書き込まれる。図中では、L C D 1 6 に表示されるフレームに対応して「 V G A 1 」、「 V G A 2 」のように示した。

10

#### 【 0 0 6 3 】

図 3 で説明した通り、スプライトレジスタ 385 s はダブルバッファとなっている。第 1 バッファへの「 V G A 1 」の描画コマンドの書き込みが完了すると、この描画コマンドは、D M A 信号に同期して、第 2 バッファに D M A 転送され、V D P 385 によるビットマップ展開が開始される。この結果、図中に示すように、「 V G A 1 」の書き込みが完了し、D M A 信号が出力された後で、V D P 出力として「 V G A 1 出力 」が得られる。V G A 2 以降の描画コマンドおよび表示データの出力も同様のタイミングで繰り返される。また、V D P 出力と併せて V D P 385 からは同期信号 V S Y N C が出力される。

20

#### 【 0 0 6 4 】

C P U 381 から V D P 385 には、描画コマンドの他、描画条件を指定する条件設定コマンドも出力される。このコマンドは V D P レジスタに格納される。また、C P U 381 は、V D P 385 に描画コマンドを出力する際に、フィールド信号を出力する。フィールド信号は、図示する通り、フレームごとに O N / O F F が切り替わる信号である。スケーラ 390 は、フィールド信号に連動する形でデータ読み書きをする対象となるフィールドを指定する制御信号 I F L D を内部的に生成する。分割表示モードでも使用できるよう制御信号 I F L D は上位桁を I F L D 1 、下位桁を I F L D 0 とする 2 ビットの信号として設定されている。

30

#### 【 0 0 6 5 】

下位桁 I F L D 0 の O N · O F F は、フィールド信号を受けて変化する。ただし、C P U 381 から V D P 385 への描画コマンド書き込みと、V D P 385 からの表示データ出力のタイミングのずれに応じて、フィールド信号および制御信号 I F L D 0 の出力タイミングにはずれが生じる。下位桁 I F L D 0 の O N · O F F が変化すると、1 周期ごとに上位桁 I F L D 1 の O N · O F F が変化する。つまり、制御信号 I F L D は、フィールド信号の変化に連動して、2 ビットで「 00 H 」～「 03 H 」の 4 値を表す信号となっている。制御信号の下位桁 I F L D 0 の O F F / O N は、それぞれフレームメモリ 397 のフィールド 397 [4] ～ 397 [5] に対応する。

40

#### 【 0 0 6 6 】

例えば、V D P 385 から「 V G A 1 出力 」がなされている時点では、制御信号 I F L D は、「 10 H 」を表しているから、書き込みメモリはフィールド 397 [4] となる。「 V G A 2 出力 」がなされている時点では、書き込みメモリはフィールド 397 [5] となる。この時点では、フィールド 397 [4] には V G A 1 の表示データが格納されているため、V D P 385 はフィールド 397 [4] からデータを読み出し、L C D 1 6 に出力する。読み出しメモリは、制御信号 I F L D 0 を反転させた信号で指定すればよい。「 V G A 2 出力 」がなされている時点では、制御信号 I F L D 0 は O N となっているから、その反転信号は O F F である。従って、下位桁が「 0 」となるフィールド、つまりフィールド 397 [4] が読み出しメモリとなる。V G A 1 出力が完了すると、下位桁 I F L D 0 が O F F となるから、読み出しメモリはフィールド 397 [5] に切り替わる。

50

## 【0067】

以上で示した手順により、VDP385はCPU381から描画コマンドを受け取り、表示データを生成してフレームメモリ397に書き込む。フレームメモリ397の書き込みメモリは、32 msec周期で切り替わる。スケーラ390は、また、フレームメモリ397から、格納済みの表示データを読み出し、LCD16に駆動データとして出力する。図示する通り、LCD16への出力は、同一の読み出しメモリから2回ずつ繰り返して行われる。つまり、LCD16には16 msec周期で同じ駆動データが出力され、同じ画面が表示される。画面内容は、32 msec周期で切り替わることになる。本実施例では、遊技中に表示される画面のスケジュール管理を容易にするため、分割モードと非分割モードの両者で、LCD16の画面内容が更新される周期を32 msecで統一した。こうすることにより、VDP385に過度の処理負荷をかけることを回避できる利点もある。もっとも、非分割モードにおけるLCD16の画面更新周期は、分割モードと必ずしも合わせる必要はない。例えば、VDP385が処理可能であれば、非分割モードでは図6に示した描画コマンドの出力、表示データの出力等を16 msec周期で実行し、LCD16の画面内容を16 msecで更新してもよい。こうすることにより、非分割モードは、高速画像を滑らかに表示することが可能となる。

## 【0068】

図7は非分割表示制御処理のフローチャートである。表示制御処理(図4)のステップS100に相当する処理である。CPU381がVDP385に対して描画コマンド等を出力し、非分割モードで表示データを生成させるための処理である。この処理は、32 msec周期の割込処理として実行される。16 msecの割込処理として構成し、2回の割込に対して1回の割合で処理全体をスキップするようにしてもよい。

## 【0069】

この処理を開始すると、CPU381は、VDP385を初期化する(ステップS102)。この処理には、VDP385が表示データの生成に使用する種々のレジスタ等の値の初期化が含まれるが、CPU381からの描画コマンド等を保持しておくべきスライトレジスタおよびVDPレジスタの初期化は含まれない。スライトレジスタ等の初期化は、別途実行することになる。

## 【0070】

次に、CPU381はVDP385のスライトレジスタを初期化する(ステップS104)。また、VDPレジスタに条件設定コマンドを設定する(ステップS106)。これは、表示データ生成時に使用される種々の設定をVDPレジスタに書き込む処理である。これらの初期化およびVDPレジスタ設定が完了すると、CPU381は、次にサブ制御基板35から受信した表示コマンドの内容を解析し(ステップS108)、LCD16に表示すべき画面データを特定する。そして、この画面データに基づいてVGA画像の描画コマンドをVDP385に出力する(ステップS110)。

## 【0071】

CPU381は、描画コマンド出力の後、スライトの最適化を行う(ステップS112)。これは、次の手順で、VDP385に定義された表示エリア(図5のVA参照)から全体が外れるようなスライトについては、描画コマンドから省略する処理である。先に説明した描画コマンド出力時には、CPU381は、スライトが表示エリアに含まれるか否かを判定することなく、画面データで特定された全てのスライトを描画対象としてVDP385のスライトレジスタに書き込む。各スライトについては、描画位置の指定の他、表示/非表示を切り換えるためのフラグが設けられており、この段階では、このフラグは全て「表示」に設定されている。CPU381は、スライトの最適化では、それぞれのスライトレジスタについて、書き込んだスライトレジスタのサイズ、位置に基づいて、表示エリアから全体が外れるか否かを判定する。全体が外れるスライトに対しては、CPU381は、上述のフラグを切り換え「非表示」に設定する。

## 【0072】

スライト最適化は、上述の通り、全体が表示エリアから外れるスライトをキャンバ

10

20

30

40

50

ス上に描画するという無駄な処理を回避するためである。描画コマンドを出力する際に、一つ一つのスプライトについて表示エリアから外れるか否かの判定を行う方法を探ることも可能ではあるが、本実施例では、描画コマンドの出力と、スプライト最適化に分けることにより、同種の処理を集約することができ、処理効率を向上させている。もっとも、スプライト最適化は、VDP385の処理効率を向上させるための処理であるため、省略することも可能である。CPU381は、以上の処理が完了すると、フィールド信号（図6参照）を切り換えて（ステップS114）、非分割表示制御処理を終了する。

#### 【0073】

##### E1. 分割表示制御処理：

図8は分割モードにおける表示データの生成例を示す説明図である。図の右下に示す画像（800×600ドット）を表示する手順を示した。図の左側には、VDP385内の処理概要を示し、図の右側にはフレームメモリ397に格納されるデータ内容を示した。図の左側に示すように、VDP385内では、描画データをビットマップ展開するためのキャンバスCV1が用意されている。回路上では、キャンバスCV1内の全画素に対応するメモリ領域が用意されることになる。キャンバスCV1のサイズは、上画像／下画像（800×300ドット）よりも大きく、LCD16の表示領域（800×600ドット）よりも小さく、縦方向に512ドットとなっている。

10

#### 【0074】

キャンバスCV1内には、表示エリアVAが定義されている。これは上画像／下画像と同一形状の領域である。VDP385は、キャンバスCV1上に画像をビットマップ展開した後、表示エリアVA内の画像を切り出し、表示データとしてスケーラ390に出力する。

20

#### 【0075】

CPU381から上画像の描画データを受け取り、VDP385が画像を描画すると、キャンバスCV2の状態が得られる。VDP385は、キャンバスCV2から、表示エリアVA内の画像を切り出し、スケーラ390を介して（図示を省略）、表示データとして出力する。この表示データは、フレームメモリ397の上画像用のフィールド397[0]に格納される。この結果、フレームメモリ397内には、状態FM1で示す内容の表示データが格納されることになる。

#### 【0076】

30

次に、CPU381から下画像の描画データを受け取り、VDP385が画像を描画すると、キャンバスCV3の状態が得られる。本実施例では、表示エリアVAの位置は固定したまま、画像の位置を-y方向に300ドット分ずらして描画する。この移動を実現するためには、描画コマンドで指定されるy座標値から300を引けばよい。画像の位置をずらして描画した状態で、表示エリアVA内の画像を切り出せば、下画像の表示データが得られることになる。この表示データは、フレームメモリ397の下画像用のフィールド397[1]に格納される。この結果、フレームメモリ397内には、状態FM2で示す通り、画像全体を表す表示データが格納されることになる。この表示データを出力すれば、LCD16には右下に示した画像を表示させることができる。

#### 【0077】

40

図9は分割モードにおいてVDP385への画面データ出力の様子を示すタイミングチャートである。図8で説明した手順で、上画像、下画像の表示データがフレームメモリに格納され、LCD16に出力される様子を示した。VDP385のスプライトレジスタ385sには、上画像、下画像の順で繰り返し描画コマンドが書き込まれる。図中では、LCD16に表示されるフレーム1に対応した上画像、下画像をそれぞれ「上1」、「下1」のように略称で示した。フレーム2、フレーム3に対応する上画像、下画像も同様である。

#### 【0078】

図3で説明した通り、スプライトレジスタ385sはダブルバッファとなっている。第1バッファへの「上1」の描画コマンドの書き込みが完了すると、この描画コマンドは、

50

DMA信号に同期して、第2バッファにDMA転送され、VDP385によるビットマップ展開が開始される。この結果、図中に示すように、「上1」の書き込みが完了し、DMA信号が出力された後で、VDP出力として「上1出力」、即ち「上1」に対応した表示データの出力が得られる。「下1」以降の描画コマンドおよび表示データの出力も同様のタイミングで繰り返される。また、VDP出力と併せてVDP385からは同期信号VSYNCが出力される。

#### 【0079】

CPU381からVDP385には、描画コマンドの他、描画条件を指定する条件設定コマンドも出力される。このコマンドはVDPレジスタに格納される。本実施例では、CPU381からVDP385への上画像の描画コマンドの出力が完了した時点で、条件設定コマンドを出力するものとした。従って、図示するように、例えば、「上1書き込み」後のDMA信号に同期して、VDPレジスタ更新、つまりVDPレジスタへの条件設定コマンドの出力が行われる。この条件設定コマンドは、「上1出力」および「下1出力」のVDP出力に共通して用いられる。従って、「下1書き込み」後のVDPレジスタの更新は省略することができる。

10

#### 【0080】

CPU381は、VDP385に描画コマンドを出力する際に、上画像／下画像を特定するフィールド信号を出力する。フィールド信号は、図示する通り、上画像に対してON、下画像に対してOFFとなる信号である。スケーラ390は、フィールド信号に連動する形でデータ読み書きをする対象となるフィールドを指定する制御信号IFLDを内部的に生成する。制御信号IFLDは2ビットの信号であり、図中では上位桁をIFLD1、下位桁をIFLD0で表した。

20

#### 【0081】

下位桁IFLD0のON・OFFは、フィールド信号を受けて変化する。例えば、「上1書き込み」時にCPU381からフィールド信号としてONが出力されると、VDP385は「上1出力」時に制御信号IFLD0としてONを出力する。CPU381からVDP385への描画コマンド書き込みと、VDP385からの表示データ出力のタイミングのいずれに応じて、フィールド信号および制御信号IFLD0の出力タイミングにはずれが生じる。下位桁IFLD0のON・OFFが変化すると、1周期ごとに上位桁IFLD1のON・OFFが変化する。つまり、制御信号IFLDは、フィールド信号の変化に連動して、2ビットで「00H」～「03H」の4値を表す信号となっている。制御信号IFLDで表される4値は、それぞれフレームメモリ397のフィールド397[0]～397[3]に対応している。

30

#### 【0082】

例えば、VDP385から「上1出力」がなされている時点では、制御信号IFLDは、「10H」を表しているから、書き込みメモリはフィールド397[2]となる。「下1出力」がなされている時点では、書き込みメモリはフィールド397[3]となる。つまり、フレーム1の上画像、下画像の表示データは、フィールド397[2]、397[3]に書き込まれることになる（図3参照）。この時点では、フィールド397[0]、397[1]にはフレーム1よりも前に書き込み済みの表示データが格納されているため、VDP385はフィールド397[0]、397[1]からデータを読み出し、LCD16に出力する。読み出しメモリは、制御信号IFLD1を反転させた信号で指定すればよい。上述の時点では、制御信号IFLD1はONとなっているから、その反転信号はOFFである。従って、上位桁が「0」となるフィールド、つまり「00H」、「01H」に対応するフィールド397[0]、397[1]が読み出しメモリとなる。下1出力が完了すると、上位桁IFLD1がOFFとなるから、読み出しメモリは「10H」、「11H」に対応するフィールド397[2]、397[3]に切り替わる。また、下位桁IFLD0に応じて、フィールド397[0]、397[1]が書き込みメモリとなる。

40

#### 【0083】

以上で示した手順により、VDP385はCPU381から描画コマンドを受け取り、

50

表示データを生成してフレームメモリ 397 に書き込む。フレームメモリ 397 の書き込みメモリは、16 msec 周期で切り替わる。スケーラ 390 は、また、フレームメモリ 397 から、格納済みの表示データを読み出し、LCD 16 に駆動データとして出力する。図示する通り、LCD 16 への出力は、同一の読み出しメモリから 2 回ずつ繰り返して行われる。つまり、LCD 16 には 16 msec 周期で同じ駆動データが出力され、同じ画面が表示される。画面内容は、32 msec 周期で切り替わることになる。

#### 【0084】

図 10 は分割モードにおける表示制御処理のフローチャートである。CPU 381 が VDP 385 に対して描画コマンド等を出力し、表示データを生成させるために実行する処理である。この処理は、16 msec 周期の割込処理として実行される。

10

#### 【0085】

先に説明した通り、本実施例では、表示制御処理を実行する度に、上画像、下画像を順次、切り換えて描画コマンドを出力することになる。上画像、下画像のいずれを処理対象とすべきかは、上下フラグによって管理される。上下フラグは、上画像を処理対象とすべき時には値 1 となり、下画像を処理対象とすべき時には値 0 となるフラグである。

#### 【0086】

CPU 381 は上下フラグが値 0 の場合には（ステップ S202）、以下に示す通り、上画像に対する処理を実行する。まず、CPU 381 は VDP 385 を初期化する（ステップ S204）。この処理には、VDP 385 が表示データの生成に使用する種々のレジスタ等の値の初期化が含まれるが、CPU 381 からの描画コマンド等を保持しておくべきスライドレジスタおよび VDP レジスタの初期化は含まれない。スライドレジスタ等の初期化は、別途実行することになる。

20

#### 【0087】

CPU 381 は次に上半分表示処理を実行する（ステップ S210）。上半分表示処理の内容は後で詳述し、ここでは概要のみを説明する。この処理は、新たなフレームの上画像の描画コマンドを VDP 385 に出力する処理ではない。新たな上画像の描画コマンド出力に先立って、直前のフレームの下画像の描画コマンドの座標を修正する処理である。この下画像の描画コマンドは既に VDP 385 のスライドレジスタに蓄積されているデータであり、本来、上画像に対する処理では不要のデータである。しかし、スライドレジスタの初期化を行っても、そのままスライドレジスタに残っている場合がある。初期化では、レジスタに全て「0」や「Null」を入力するとは限らず、描画コマンドを残したまま、単にその描画コマンドの設定を「無効」、「非表示」などにする簡易な方法が採られることがあるからである。この場合には、設定の誤り等によって、直前のフレームのスライドが再度、表示対象となる可能性がないとは言えない。そこで、本実施例では、仮にこのような事態が生じたとしても、上画像の表示データに極端な違和感を与えることを回避するための処理として、上半分表示処理を実行している。この処理では、例えば、図 8 のキャンバス CV3 に示された画像の位置は、キャンバス CV2 に示した上画像の描画時の位置に戻される。次のフレームは、直前のフレームと近似した画像となることが多いため、このように位置を修正しておくことにより、例え直前の描画コマンドが表示データに影響を与えたとしても、違和感を十分に抑制することが可能となる。上半分表示処理は必須の処理ではないため、省略することも可能である。

30

#### 【0088】

上半分表示処理が完了すると、CPU 381 は VDP 385 のスライドレジスタを初期化する（ステップ S220）。この時点で、併せて VDP レジスタの初期化も行う。初期化が完了すると、CPU 381 は、次にサブ制御基板 35 から受信した表示コマンドの内容を解析し（ステップ S222）、LCD 16 に表示すべき画面データを特定する。そして、この画面データに基づいて上画像の描画コマンドを VDP 385 に出力する（ステップ S224）。先に図 8 で説明した通り、本実施例では、上画像、下画像のいずれを処理するかに関わらず、LCD 16 に表示すべき画面全体の描画コマンドを VDP 385 に出力している。上画像、下画像の生成時で異なるのは、描画コマンドの座標値のみである

40

50

。本実施例では、描画コマンドは、上画像生成時を基準として用意しておくものとした。従って、ステップ S 2 2 4 の処理では、C P U 3 8 1 は描画コマンドの座標値を修正するまでなく、そのまま V D P 3 8 5 に出力すれば足りる。

#### 【 0 0 8 9 】

C P U 3 8 1 は、描画コマンド出力と並行して、スプライトの最適化を行う（ステップ S 2 2 6 ）。

#### 【 0 0 9 0 】

C P U 3 8 1 は、以上の処理が完了すると、下画像が次の処理対象となることを示すため、フィールド信号（図 9 参照）を O F F とする（ステップ S 2 2 8 ）。また、上下フラグを値 1 だけ増加させる（ステップ S 2 5 0 ）。上下フラグは 1 ビットのフラグであるため、この処理によって上下フラグの値は 1 となり、次の処理では下画像が処理対象と判断されることになる。

10

#### 【 0 0 9 1 】

ステップ S 1 3 において、上下フラグが値 1 の場合には、C P U は以下の手順で下画像に対する処理を実行する。まず、C P U は V D P 3 8 5 を初期化し（ステップ S 2 3 0 ）、V D P レジスタに条件設定コマンドを設定する（ステップ S 2 3 2 ）。これは、図 9 に示した V D P レジスタの更新に相当する処理であり、上画像、下画像に共通して表示データ生成時に使用される種々の設定を行う処理である。V D P 3 8 5 は、この処理を受けて、スプライトレジスタに設定済みの描画コマンドに基づき、上画像の表示データを生成してフレームメモリ 3 9 7 に出力する。

20

#### 【 0 0 9 2 】

C P U 3 8 1 は、次に下半分表示処理を実行する（ステップ S 2 3 4 ）。この処理は、先に概要を説明した上半分表示処理（ステップ S 2 1 0 ）と同様、描画コマンドの座標値を修正する処理である。この処理によって、例えば、上画像の描画時には、図 8 中のキャンバス C V 2 の状態で配置される描画コマンドは、キャンバス C V 3 の状態に移動されることになる。本実施例では、上画像、下画像は y 方向に 3 0 0 ドットのサイズとなっているから、下半分処理では、描画コマンドの y 座標をそれぞれ 3 0 0 ずつ減じればよい。

#### 【 0 0 9 3 】

C P U 3 8 1 は、下半分表示処理が完了すると、スプライトの最適化を行い（ステップ S 2 3 6 ）、フィールド信号を O N とし（ステップ S 2 3 8 ）、上下フラグを値 1 だけ増加させる（ステップ S 2 5 0 ）。上下フラグは 1 ビットのフラグであるから、この演算によって値 0 となる。この結果、フィールド信号、上下フラグとともに、次の処理対象が上画像であることを示す状態となる。この後、V D P 3 8 5 内では、下画像の描画コマンドが読み込まれ、表示データが生成される。条件設定コマンドはステップ S 5 1 で V D P レジスタに設定された内容がそのまま使用される。本実施例では、上画像の描画コマンドが設定された後、その表示データの生成を開始する前の時点で、条件設定コマンドを設定することにより、下画像に対する一連の処理内で、条件設定コマンドを上画像、下画像に流用することができる。

30

#### 【 0 0 9 4 】

図 1 1 は上半分表示処理のフロー チャートである。表示制御処理（図 1 0 ）のステップ S 2 1 0 に相当する処理である。この処理は、先に説明した通り、V D P 3 8 5 のスプライトレジスタに格納されている描画コマンドの y 座標を 3 0 0 増加させる。本実施例では、スプライトごとに、その配置、表示 / 非表示の設定することで描画コマンドが構成されていることから、スプライト単位で順次、その座標を修正する手法を探った。以下では、それぞれのスプライトに対して、一義的な識別番号として「スプライト N o . 」が付されているものとして説明する。

40

#### 【 0 0 9 5 】

まず、C P U 3 8 1 は全スプライトを順番に処理するための初期設定として、処理対象となるべき「スプライト N o . 」を 0 に設定する（ステップ S 2 1 1 ）。そして、処理対象となるスプライトの設定データを読み込む（ステップ S 2 1 2 ）。図中に設定データの

50

内容を例示した。識別番号としての「スプライトNo.」、キャラROM内の「データ格納場所」を表すアドレス、スプライトを配置すべきキャンバス上の配置座標(XPOS, YPOS)、およびスプライトの表示/非表示を切り換えるためのフラグが含まれる。この他のデータを含めても差し支えない。

#### 【0096】

CPU381は、処理対象のスプライトが、表示/非表示フラグによって「表示」設定されている場合には(ステップS213)、配置座標の修正を施す(ステップS214)。つまり、y座標(YPOS)を値300だけ増大させる。図中にこの処理の意味を示した。図の左側に示す通り、表示エリアVAおよび画像が配置されているとする。この状態で画像のy座標を値300だけ増大させると、右側に示すように画像の位置が表示エリアVAに対して相対的に下側にずれる。表示エリアVAは縦方向に300ドットであるから、左側の画像で表示エリアVAから上側にはみ出していた部分が表示エリア内に入る。つまり、左側の下画像に対応する上画像が表示エリアVA内に現れることになる。CPU381は、表示/非表示フラグが非表示設定となっている場合には(ステップS213)、上述した配置座標の修正を施す必要はないと判断し、ステップS214をスキップする。

#### 【0097】

以上の処理を、CPU381は「スプライトNo.」が所定の上限値N1imに至るまで(ステップS215)、「スプライトNo.」を順次、値1ずつ増加させながら(ステップS216)、繰り返し実行する。上限値N1imは、画面データで用いられる全スプライトに付された「スプライトNo.」の最大値とすることができます。遊技状態によって、一部のスプライトしか用いないことが決まっている場合には、その中の「スプライトNo.」の最大値としてもよい。

#### 【0098】

以上で説明した実施例によれば、800ドット×600ドットというLCD16用の表示データを上画像、下画像の2回に分けて生成することができる。この結果、VDP385の処理能力がLCD16の画素数に見合うほどに高くはない場合であっても、解像度を活かした高い画質で画像を表示することが可能となる。

#### 【0099】

E2. 分割表示制御処理の変形例：

実施例では、図8で説明した通り、上画像、下画像の表示データを生成する際に、表示エリアVAに対する位置を相対的に移動させながら、画面全体の描画コマンドをVDP385に出力する例を示した。分割モードにおけるVDP385への描画コマンドの出力は、この様に依らず、種々の様相を採ることができる。例えば、描画コマンドの座標値は固定としたまま表示エリアの位置を移動させてもよい。また、上画像用、下画像用に個別に描画コマンドを用意してもよい。以下では、これらの2つの様相について、それぞれ第1変形例、第2変形例として説明する。

#### 【0100】

図12は第1変形例としての表示制御処理のフローチャートである。図10に代わる処理であり、上画像、下画像の生成時に、描画コマンドの座標値は固定としたまま表示エリアの位置を移動させる様相に相当する。この処理では、図10と同様、CPU381は、上下フラグの値に応じて、以下に示す通り、上画像、下画像を対象とするそれぞれの処理を実行する。

#### 【0101】

上下フラグが値0の場合(ステップS202)、即ち上画像の処理を行う場合には、CPU381はVDPの初期化を行った後(ステップS204)、表示エリアVAの位置を設定する(ステップS204A)。表示エリアVAは、描画コマンドで描かれる全体画像のうち上画像を切り出すことができる位置に設定される(図8のキャンバスCV2参照)。そして、CPU381は、スプライトレジスタを初期化し(ステップS220)、コマンド解析処理を実行し(ステップS222)、画像全体の描画コマンドをVDP385に出力する(ステップS224)。そして、スプライトを最適化し(ステップS226)、

10

20

30

40

50

次の処理対象が下画像となることを示すため、フィールド信号をOFFにするとともに(ステップS228)、上下フラグの値を1だけ増加させて(ステップS250)、表示制御処理を終了する。

#### 【0102】

上下フラグが値1の場合(ステップS202)、即ち下画像の処理を行う場合には、CPU381はVDPレジスタに条件設定コマンドの設定を行う(ステップS230)。この設定を受けて、VDP385は上画像の描画コマンドおよび表示エリアVAの設定に基づき、上画像の表示データを生成、出力する。その後、CPU381は、表示エリアVAを下方に300ドット移動させる(ステップS234A)。図中に移動の様子を示した。左側に示すように、上画像を切り出し可能な状態で配置されていた表示エリアVAは、この処理によって、右側に示すように下画像を切り出し可能な状態に移動される。実施例(図10)と異なり描画コマンドの座標の修正は行わないため、画像の位置は固定されたままである。CPU381は、スプライトの最適化を行い(ステップS236)、フィールド信号をONした後(ステップS238)、上下フラグを値1だけ増加させて(ステップS250)、表示制御処理を終了する。VDP385は、これに応じて、下画像の表示データを生成する。

10

#### 【0103】

第1変形例によれば、描画コマンドの座標値を修正する必要がなく、上画像、下画像それぞれの生成に要する処理負荷を軽減することができる利点がある。

#### 【0104】

20

図13は第2変形例としての表示制御処理のフローチャートである。図10に代わる処理であり、上画像用、下画像用に個別に描画コマンドを用意する態様に相当する。この処理では、図10と同様、CPU381は、上下フラグの値に応じて、以下に示す通り、上画像、下画像を対象とするそれぞれの処理を実行する。

#### 【0105】

上下フラグが値0の場合(ステップS202)、即ち上画像の処理を行う場合には、CPU381はVDP初期化(ステップS204)、スプライトレジスタの初期化を行った後(ステップS220)、コマンド解析処理を実行してLCD16に表示すべき画像の画面データを特定した後(ステップS222)、上半分の描画コマンドを出力する(ステップS224A)。変形例では、コマンド解析処理によって特定される画面データには、上画像用の画面データ、下画像用の画面データが個別に用意されているものとした。例えば、上画像用には図中に示すハッチング部分に対応する描画コマンドが含まれている。図の例では、この画面にはスプライトSP1、SP2が含まれるが、上画像で必要とされるのは、スプライトSP1のみである。従って、画面データはスプライトSP2を省略した状態のデータとすることができる。CPU381は、実施例と同様、スプライトの最適化(ステップS226)、フィールド信号OFF(ステップS228)、上下フラグのインクリメント(ステップS250)の各処理を実行して、表示制御処理を終了する。第2変形例では、上画像用に最適化された画面データを用意することができるため、スプライトの最適化(ステップS226)は省略することも可能である。

30

#### 【0106】

40

上下フラグが値1の場合(ステップS202)、即ち下画像の処理を行う場合には、CPU381はVDPレジスタに条件設定コマンドを設定し(ステップS232)、スプライトレジスタを初期化して(ステップS233)、下半分の描画コマンドを出力する(ステップS234A)。下画像用の画面データには図中に示すハッチング部分に対応する描画コマンド、即ちスプライトSP1、SP2の双方に対する描画コマンドが含まれることになる。描画コマンドのy座標値は、表示エリアVAによって下画像を切り出すことができるよう設定された値となっている。従って、CPU381は座標値の修正や表示エリアVAの位置修正などを施すまでなく、単に用意された描画コマンドをVDP385に出力すれば足りる。CPU381は、次に、実施例と同様、スプライトの最適化(ステップS236)、フィールド信号ON(ステップS238)、上下フラグのインクリメント(

50

ステップS250)の各処理を実行して、表示制御処理を終了する。スプライトの最適化(ステップS236)は省略することも可能である。

#### 【0107】

第2変形例によれば、上画像、下画像のそれぞれについて描画コマンドを出力する際に、座標修正や表示エリアの移動などの処理が不要となるため、処理負荷を軽減することができる利点がある。

#### 【0108】

上述の実施例および変形例では、分割モードにおいてLCD16の画面を上下2つに分割して表示データを生成する場合を例示した。画面の分割は、かかる様に依らず、種々の様子を探ることができる。例えば、上下方向に3以上の領域に分割してもよい。また、上画像、下画像を更に左右方向に分割してもよい。

10

#### 【0109】

F. 分割移行処理：

図14は分割移行処理におけるVDP385への画面データ出力の様子を示すタイミングチャートである。分割移行処理とは、非分割モードから分割モードへの切り換えを行う処理である。最上段には、定常処理としてサブ制御基板35が実行するスケジューラ更新のタイミングを示した。スケジューラ更新とは、サブ制御基板35が遊技状態に応じてLCD16にシーケンシャルに表示すべき画面を順次、更新していく処理であり、サブ制御基板35から装飾図柄制御基板30に表示コマンドを出力する処理に対応する。本実施例では、非分割モード、分割モードとともに、32 msecでスケジューラ更新が行われる。

20

#### 【0110】

非分割モードで描画コマンドの出力が行われているものとして説明を行う。図中の例は、VDP385のスプライトレジスタにVGA2出力がなされた状態を示している。この時点では、VDP385からは、従前の描画コマンドに対応するVGA1の表示データが出力されている。このVGA1出力は、VDP385から出力される同期信号VSYNCをトリガとして、フレームメモリ397のフィールド397[4](以下、「VGA用エリア0」と呼ぶこともある)に書き込まれる。

#### 【0111】

次に、サブ制御基板35がスケジュール更新を行い、分割モードへの移行が指示されたとする。この移行指示は、例えば、サブ制御基板35が「移行コマンド」を出力する様子をとってもよいし、サブ制御基板35が指定する装飾図柄が、高速画像から高画質画像に切り替わった時点でCPU381が自律的に移行指示がなされたものと判断するという様子をとってもよい。これらの様子で移行指示がなされると、CPU381は、VDPレジスタに対して「上下設定」、即ち分割モードで処理するための設定を行う。また、スプライトレジスタには、「上1」画像の描画データ書き込みを行う。ただし、この間、VDP385は従前に書き込まれたVGA2に対応する表示データを出力している。

30

#### 【0112】

CPU381は、上述の処理と並行して、スケーラ390に対して上下用、即ち分割モード用の書き込みアドレスを設定する。本実施例では、図3に示した通り、非分割モードと分割モードでは個別に用意されたフィールドが用いられる。書き込みアドレスの設定は、表示データの格納先を、非分割モードで使用されているフィールド397[4]、397[5]から、分割モードで使用されるフィールド397[0]~397[3]に切り換える処理である。

40

#### 【0113】

書き込みアドレスの変更は、VDP385からのVGA2出力と並行して行われるため、この変更を直ちに反映させると、フレームメモリ397に表示データを格納すべきフィールドが、VDP385からのデータ出力中に変化することになり、表示データの格納が不安定となるおそれがある。また、VDP385から出力される表示データは、非分割モードの描画データに対応したVGA2出力であるから、データ容量的にも分割モード用のフィールドに適合しない。本実施例では、これらの不整合によって表示データの格納が不

50

安定となる事態を回避するため、書き込みアドレスの変更時には、表示データをフレームメモリに書き込むことを禁止している。フレームメモリ 397 への表示データの格納は、VDP385 からの同期信号 VSYNC をトリガとして行われるから、書き込みアドレス変更時には、図中に破線 A で示すように、VSYNC の出力を停止するのである。本実施例では、CPU381 が VEN にハイを出力することにより、信号停止回路 389 の機能によって VDP385 からの VSYNC の出力を停止する。この方法に代えて、CPU381 が VDP385 に制御信号を出力することで、ソフトウェア的に同期信号 VSYNC の出力を停止させてもよい。同期信号 VSYNC の出力が停止される結果、図中に示す通り、VDP385 から表示データが出力されてはいるものの、この表示データは、書き込みメモリには格納されなくなる。

10

#### 【0114】

この時点では、書き込みアドレスの変更は行われているものの、フレームメモリ 397 からの読み出しアドレスの変更は行われていない。従って、スケーラ 390 は、従前の非分割モードで設定された読み出しアドレスに従い、フィールド 397 [5] (「VGA 用エリア 1」と呼ぶこともある) から表示データを読み出し、LCD16 に出力する。LCD16 には、非分割モード時の画像が表示され続けることになる。

#### 【0115】

書き込みアドレスの変更が完了すると、次の 16 msec の処理では、CPU381 はスプライトレジスタに「下 1」の書き込みを行う。また、分割モードでの処理に従い、下 1 書き込みと並行して VDP レジスタの設定を行う。VDP385 からは、既に設定済みの「上 1」画像に対応した表示データが出力される。この時点では、書き込みアドレスの変更が完了しているため、この表示データは、フィールド 397 [0] (「エリア 0 上」と呼ぶこともある) に格納される。この時点でも、読み出しアドレスは未変更であるから、スケーラ 390 は、VGA 用エリア 1 から表示データを読み出し LCD16 に出力する。

20

#### 【0116】

「下 1」画像の書き込みが完了すると、以後は、分割モードに従って、安定的に描画コマンドの書き込み、および表示データの出力が行われるようになる。CPU381 は、この時点で、フィールド 397 [0] ~ 397 [3] からのデータ読み出しが行われるよう、スケーラ 390 の読み出しアドレスを分割モード用に変更する。書き込みアドレスは変更済みであるから、フィールド 397 [0] (エリア 0 上)、397 [1] (エリア 0 下) には、「上 1」「下 1」画像の表示データが格納されている。読み出しアドレスの変更中は、非分割モードに従って、フィールド 397 [4] (VGA 用エリア 0) からの表示データ読み出しが行われる。なお、VGA 用エリア 0 には、VGA 用エリア 1 よりも以前に生成された表示データが格納されているため、この時点では、一瞬とはいえ、厳密には LCD16 の表示は過去の表示内容に逆戻りしていることになる。従って、遊技者にちらつきなどの違和感を与えないようするため、非分割モードから分割モードへの切り換え前には、黒画面などの同一画面を 2 フレーム以上継続する期間を設けておくことが好ましい。

30

#### 【0117】

読み出しアドレスの変更が完了した後は、フィールド 397 [3] (エリア 1 上)、397 [1] (エリア 1 下) へのデータ書き込みが行われるとともに、フィールド 397 [0]、397 [1] からの表示データの読み出しが行われるようになり、分割モードへの移行が完了する。以上で説明したように、分割モードへの移行は、16 msec での割込処理を一つのフェーズとして考えると、書き込みアドレスの変更 (フェーズ 1)、分割モードでの「上 1」「下 1」画像の描画コマンド出力およびフィールド 397 [0] (エリア 0 上) への表示データの格納開始 (フェーズ 2)、読み出しアドレスの変更 (フェーズ 3) という 3 つのフェーズで行われる。

40

#### 【0118】

図 15 および図 16 は分割移行処理のフローチャートである。CPU381 が実行する

50

処理であり、図14のタイミングチャートで示した処理に相当する処理である。CPU381は、まず信号停止回路389を制御して、VDP385からのVSYNCの出力をディセーブルする(ステップS302)。次に、VDP385を上下表示出力に設定し、分割モードで処理すべきことを指示する(ステップS304)。そして、次に処理すべき画像が上画像であることを示すため、上下フラグに値0を初期値として設定し(ステップS306)、上表示用座標補正処理を実行する(ステップS308)。上表示用座標補正処理とは、分割モードにおける上半分表示処理(図11参照)と同じく、描画コマンドの座標値を補正する処理である。ここでは、非分割モードで指定されている描画データの座標値を補正することになるため、「上半分表示処理」との名称を避けたが、処理内容は同等である。

10

#### 【0119】

上表示用座標補正処理が完了すると、CPU381は、「上1」画像に対応する描画コマンドを出力し(ステップS310)、スプライトの最適化を行う(ステップS312)。この時点で、サブ制御基板35からのコマンドを解析して、分割モードでの表示内容に従った描画コマンドを出力してもよいが、本実施例では、非分割モードで指定済みの描画コマンドを流用するものとした。分割移行処理を行う時点では、暗転など改めてコマンド解析をする必要性に乏しい画像が使われることが多いからであり、コマンド解析を行ったとしても分割移行中は表示データの出力、格納が不安定なため、有効活用されない可能性があるからである。

#### 【0120】

20

CPU381は、VDP385からの表示データがフレームメモリ397の分割モード用のフィールド397[0]～397[3]に格納されるよう、スケーラ390の書き込みアドレスを設定し(ステップS314)、信号停止回路389を制御して、VSYNCをイネーブルする(ステップS316)。この処理によって、VDP385から次に出力される同期信号VSYNCはスケーラ390に伝達されるようになり、表示データがフレームメモリ397に格納されるようになる。以上が、図14で説明したフェーズ1の処理に対応する。

#### 【0121】

30

次に、CPU381は、下画像の処理に移行すべく、フィールド信号をOFFとし(図16のステップS320)、上下フラグに値1を設定する(ステップS322)。そして、下表示用座標補正処理を行い(ステップS324)、描画コマンドを出力して(ステップS326)、スプライトの最適化を行う(ステップS328)。下表示用座標補正処理は、上表示用座標補正処理(ステップS308)と同様、描画コマンドの座標値を下画像用に300減じる処理である。以上が、図14で説明したフェーズ2の処理に対応する。本実施例では、この時点でも、サブ制御基板35からのコマンド解析は省略した。

#### 【0122】

40

次に、CPU381は、再び上画像の処理に移行すべき、フィールド信号をONとし(ステップS330)、上下フラグに0を設定する(ステップS332)。そして、上表示用座標補正処理を行う(ステップS334)。この時点では、書き込みアドレスの変更も完了しており、分割モードでの表示データを安定的にフレームメモリ397に格納可能となっている。従って、CPU381は、スプライトレジスタを初期化して(ステップS336)、サブ制御基板35からのコマンドを解析し(ステップS338)、これに応じた描画コマンド出力(ステップS340)、スプライトの最適化を行う(ステップS342)。

#### 【0123】

その後、CPU381は、フレームメモリ397の分割モード用のフィールド397[0]～397[3]から表示データが読み出されるよう、スケーラ390の読み出しアドレスを設定し(ステップS344)、分割移行処理を終了する。以上の処理が図14のフェーズ3に相当する。この処理が完了した後は、分割モードで表示データの格納、読み出しが可能な状態となる。

50

## 【0124】

## G. 非分割移行処理：

図17は非分割移行処理におけるVDP385への画面データ出力の様子を示すタイミングチャートである。非分割移行処理とは、分割モードから非分割モードに表示モードを切り換える処理である。最上段には、サブ制御基板35でのスケジューラ更新のタイミングを示した。後述する通り、本実施例では、移行期間の途中で、スケジューラの更新を行わない期間が存在する。これは、移行期間において表示データの生成、格納が不安定となることによって、LCD16に表示される画面にちらつきが生じることを回避するために行われる処理であり、意図的に同一画面を継続的に表示させる処理である。

## 【0125】

10

分割モードが実行されている途中にサブ制御基板35がスケジュール更新を行い、非分割モードへの移行が指示されたとする。この移行指示は、例えば、サブ制御基板35が「移行コマンド」を出力する様子をとってもよいし、サブ制御基板35が指定する装飾図柄が、高画質画像から高速画像に切り替わった時点でCPU381が、自律的に移行指示がなされたものと判断するという様子をとってもよい。これらの様子で移行指示がなされると、CPU381は、VGA1書き込みを行った後、VDPレジスタに対して「VGA設定」、即ち非分割モードで処理するための設定を行う。ただし、この間、VDP385は従前に分割モードで書き込まれた「下2」画像に対応する表示データを出力している。

## 【0126】

20

次に、CPU381は非分割モードに従って表示データを格納するよう、フレームメモリ397の書き込みアドレスをVGA用、即ち非分割モード用に設定する。また、スライドレジスタには、VGA2の書き込みを行う。VDP385は、従前に書き込まれたVGA1画像に対応する表示データを出力している。ただし、VDPレジスタに「VGA設定」を行った直後なので、VGA1の表示データの出力動作は不安定となっているおそれがある。本実施例では、このような不安定な状態で出力された表示データが表示に弊害を与えることを回避するため、この時点での表示データのフレームメモリ397への格納を禁止する。図中の破線Bに示すように、CPU381は信号停止回路389を制御して、VDP385からの同期信号VSYNCをディセーブルするのである。こうすることにより、VDP385から出力されるVGA1出力の表示データはフレームメモリ397に格納されずに破棄される。

30

## 【0127】

書き込みアドレスの変更が完了した後は、VDP385から出力されるVGA2画像の表示データは、非分割モードに従って、フィールド397[4](VGA用エリア0)に格納されるようになる。この時点では、読み出しアドレスは変更されていないため、スケーラ390は分割モードに従って、フィールド397[2]、397[3]の表示データをLCD16に出力する。

## 【0128】

40

サブ制御基板35は、VGA2書き込みがなされた後は、スケジューラの更新を省略する。この結果、スライドレジスタへの新たな描画コマンドの書き込みも省略される。従って、VDP385からは書き込み済みの「VGA2」画像の描画コマンドに従って、VGA2出力が引き続き出力される。こうして出力されたVGA2の表示データは、非分割モードに従って、フィールド397[5](VGA用エリア1)に格納される。この結果、非分割モード用のフィールド397[4]、397[5]の双方に、VGA2画像の表示データが格納されることになる。

## 【0129】

フィールド397[5](VGA用エリア1)へのVGA2の表示データの書き込みが行われている間、並行して、スライドレジスタには新たにVGA3画像の書き込みが行われる。また、CPU381は、非分割モードで表示データの読み出しが行われるように、スケーラ390に対してVGA拡大出力設定を行う。つまり、フレームメモリ397の読み出しアドレスを、非分割モード用に変更し、読み出し時の拡大倍率を1.25倍に設

50

定する。非分割モードで生成される表示データのサイズと、LCD16の画素数との関係によっては、拡大倍率の設定は省略してもよい。読み出しあドレスの変更が完了すると、スケーラ390は、非分割モードに従って、フィールド397[4](VGA用エリア0)の表示データを読み出し、LCD16への出力を開始する。これにより、以後、非分割モードによる表示データの生成、出力が可能となる。

#### 【0130】

以上で説明した通り、非分割モードへの移行は、非分割モードでの画面更新周期である32 msecを一つのフェーズとして考えると、書き込みアドレスの変更(フェーズ1)、非分割モードでの描画コマンド出力およびフィールド397[4](VGA用エリア0)への表示データの格納開始(フェーズ2)、読み出しあドレスの変更(フェーズ3)という3つのフェーズで行われる。

10

#### 【0131】

図18は非分割移行処理のフローチャートである。CPU381が実行する処理であり、図17のタイミングチャートで示した処理に相当する処理である。CPU381は、信号停止回路389を制御して、VSYNCをディセーブルにする(ステップS402)。そして、VDP385を非分割表示出力、即ち非分割モードに設定する(ステップS404)。その後、図17にVGA2書き込みとして示した通り、描画コマンドを出力し(ステップS406)、スプライトの最適化を行う(ステップS408)。また、この処理と並行して、非分割モードで表示データの格納が行われるように、スケーラの書き込みアドレスを設定し(ステップS410)、VSYNCをイネーブルする(ステップS412)。以上の処理が図17で説明したフェーズ1に対応する処理であり、この処理によって、以後の表示データは、非分割モード用のフィールド397[4]、397[5]に格納されるようになる。

20

#### 【0132】

次に、CPU381は表示データを格納するフィールドを切り換えるべくフィールド信号をOFFとする(ステップS420)。これは図17のフェーズ2に対応する処理である。本実施例では、先に説明した通り、非分割移行処理の移行期間において、サブ制御基板35がスケジューラの更新を省略するため、フェーズ2ではCPU381は、他の処理を行うことなく時間経過を待つことになる。この間、VDP385からは、図17に示した通り、従前に書き込まれた描画コマンドVGA2に従って、VGA2出力が行われている。もっとも、スケジューラの更新の省略は非分割移行処理に必須のものではなく、移行期間においてもスケジュールの更新を行っても良い。この場合、フェーズ2では、CPU381はサブ制御基板35からの表示コマンドに従って、スプライトレジスタへの書き込みを行えばよい。

30

#### 【0133】

次に、CPU381は表示データを格納するフィールドを切り換えるべくフィールド信号をONとする(ステップS430)。そして、描画コマンドの出力(ステップS432)、スプライト最適化(ステップS434)を行う。また、非分割モードで表示データの読み出しが行われるよう、スケーラ390の読み出しあドレスを設定して、非分割移行処理を終了する(ステップS436)。以上が図17で示したフェーズ3の処理に相当する。この処理が完了した後は、非分割モードで表示データの格納、読み出しが可能な状態となる。

40

#### 【0134】

以上で説明した実施例の遊技機では、LCD16への表示データを、非分割モード、分割モードの2通りで生成することができ、これらの表示モードを遊戯中に切り換えることができる。分割モードには、LCD16の解像度を活かし、高画質な画像を提供することができる。非分割モードは、画像の解像度が比較的低いため、描画に用いるスプライトデータのデータ量を抑制することができる利点がある。従って、遊戯中に表示されるべき画面内容に応じて、双方の表示モードを使い分けることにより、それぞれ画面内容の特徴を活かした表示を実現することができ、遊戯の興奮を高めることができる。

50

## 【0135】

また、本実施例の遊技機では、書き込みアドレスの変更（フェーズ1）、移行後の表示モードでの表示データの格納開始（フェーズ2）、読み出しあドレスの変更（フェーズ3）という3フェーズで表示モードの切り換えを行うことにより、切り換え時に、LCD16への表示画面がちらつくなどの弊害を抑制することができる。こうした弊害をより確実に抑制するため、切り換え前後で暗転などの同一画面を数フレーム継続して表示させることが更に好ましい。移行期間において、VDP385からの同期信号VSYNCを停止し、フレームメモリ397に不安定な表示データの書き込みが行われるのを回避することも、ちらつき等の弊害防止に効果的である。

## 【0136】

10

以上、本発明の種々の実施例について説明したが、本発明はこれらの実施例に限定されず、その趣旨を逸脱しない範囲で種々の構成を探ることができることはいうまでもない。実施例では、LCD16を表示装置として用いる場合を例示したが、本発明は他の種類の表示装置を用いる場合にも適用可能である。

## 【図面の簡単な説明】

## 【0137】

【図1】実施例としてのパチンコ機1の正面図である。

【図2】パチンコ機1の制御用ハードウェア構成を示すブロック図である。

【図3】装飾図柄制御基板30の回路構成を示す説明図である。

【図4】表示制御処理のフローチャートである。

20

【図5】非分割表示処理における表示データの生成例を示す説明図である。

【図6】非分割モードにおけるVDP385への画面データ出力の様子を示すタイミングチャートである。

【図7】非分割表示制御処理のフローチャートである。

【図8】分割モードにおける表示データの生成例を示す説明図である。

【図9】分割モードにおいてVDP385への画面データ出力の様子を示すタイミングチャートである。

【図10】分割モードにおける表示制御処理のフローチャートである。

【図11】上半分表示処理のフローチャートである。

30

【図12】第1変形例としての表示制御処理のフローチャートである。

【図13】第2変形例としての表示制御処理のフローチャートである。

【図14】分割移行処理におけるVDP385への画面データ出力の様子を示すタイミングチャートである。

【図15】分割移行処理のフローチャート(1)である。

【図16】分割移行処理のフローチャート(2)である。

【図17】非分割移行処理におけるVDP385への画面データ出力の様子を示すタイミングチャートである。

【図18】非分割移行処理のフローチャートである。

## 【符号の説明】

## 【0138】

40

1 … パチンコ機

3 … メイン制御基板

4 … 遊技盤

4 a … 表示部

4 b … 操作スイッチ

6 … 遊技領域

8 … 発射ハンドル

9 … 始動入賞口

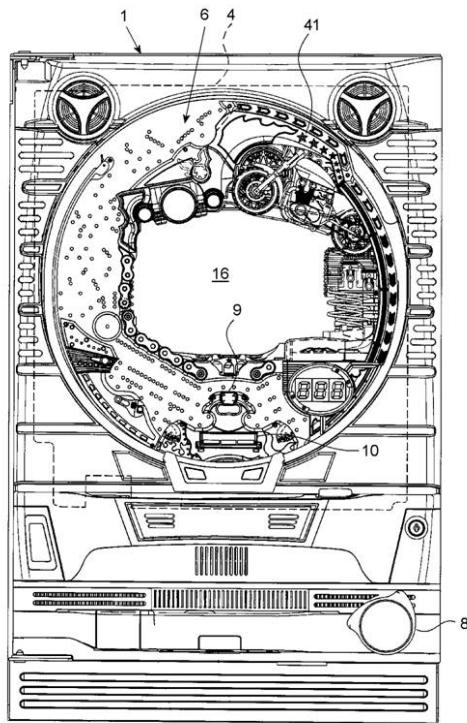
10 … 大入賞口

12 … パネル装飾ランプ

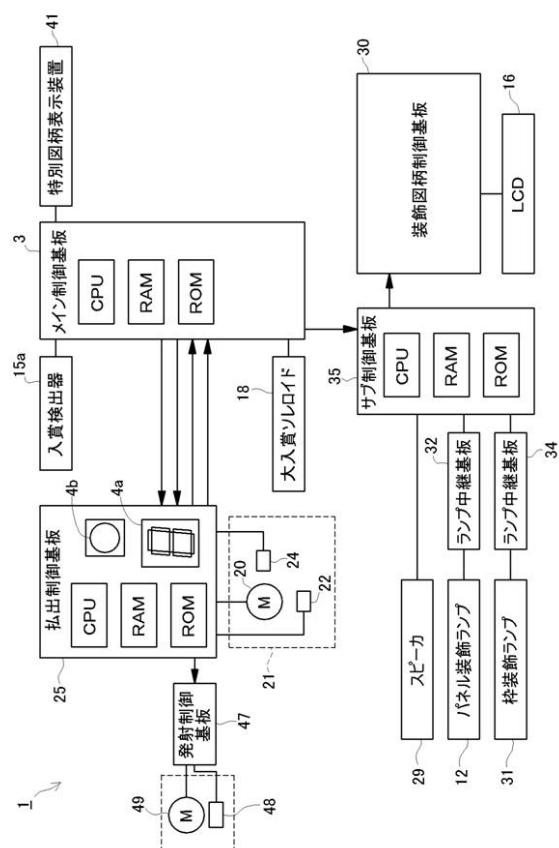
50

1 5 a ... 入賞検出器	
1 6 ... L C D	
1 8 ... 大入賞口ソレノイド	
2 0 ... 払出モータ	
2 1 ... 賞球払出装置	
2 2 ... 払出球検出器	
2 4 ... モータ駆動センサ	
2 5 ... 払出制御基板	
2 9 ... スピーカ	
3 0 ... 装飾図柄制御基板	10
3 1 ... 枠装飾ランプ	
3 2、3 4 ... ランプ中継基板	
3 5 ... サブ制御基板	
4 1 ... 特別図柄表示装置	
4 7 ... 発射制御基板	
4 8 ... タッチ検出部	
4 9 ... 発射モータ	
3 8 1 ... C P U	
3 8 2 ... R A M	
3 8 3 ... R O M	20
3 8 6 ... キャラ R O M	
3 8 5 ... V D P	
3 8 5 s ... スプライトレジスタ	
3 8 5 v ... V D P レジスタ	
3 8 9 ... 信号停止回路	
3 9 0 ... スケーラ	
3 9 7 ... フレームメモリ	
3 9 7 [ 0 ] ~ 3 9 7 [ 5 ] ... フィールド	

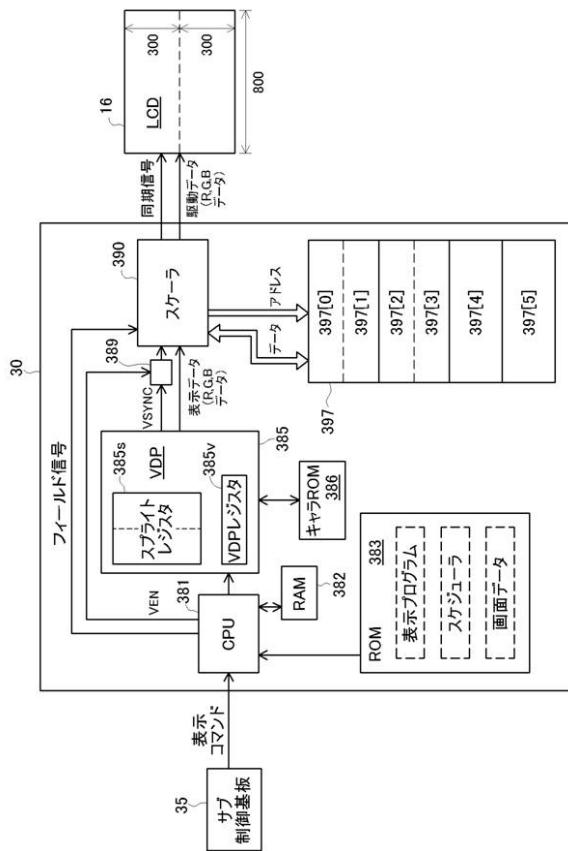
【図1】



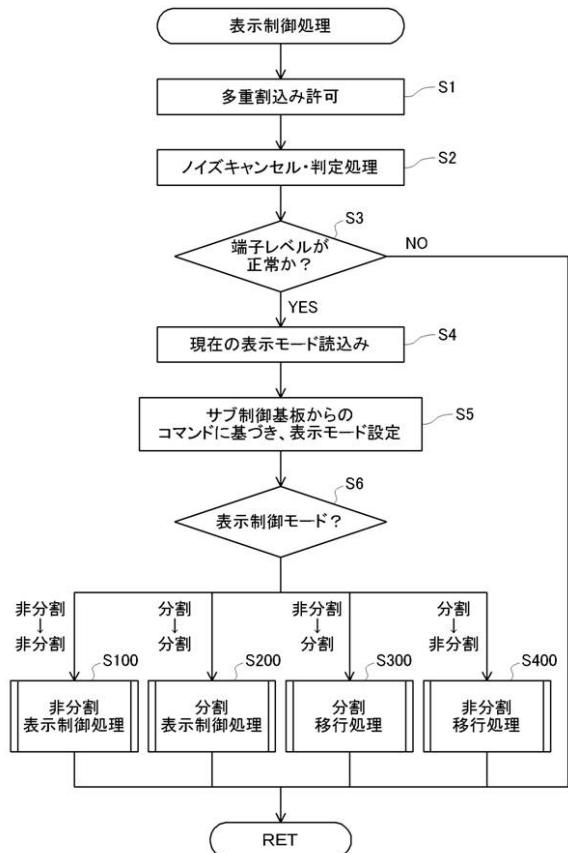
【図2】



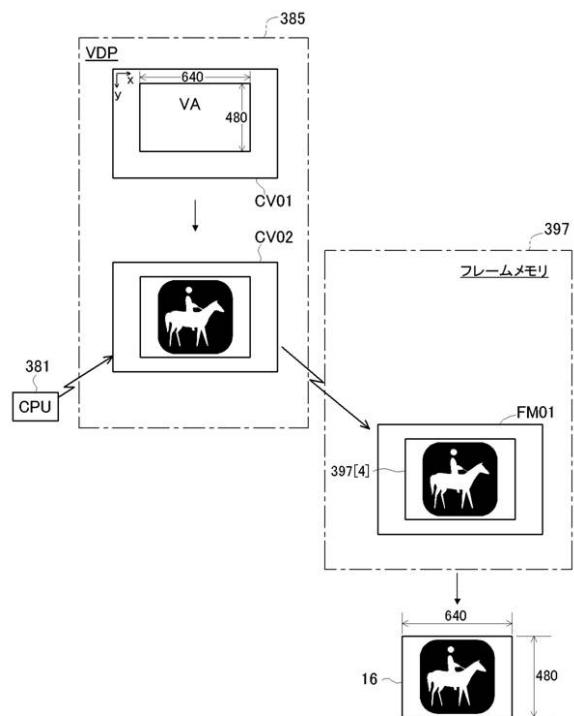
【図3】



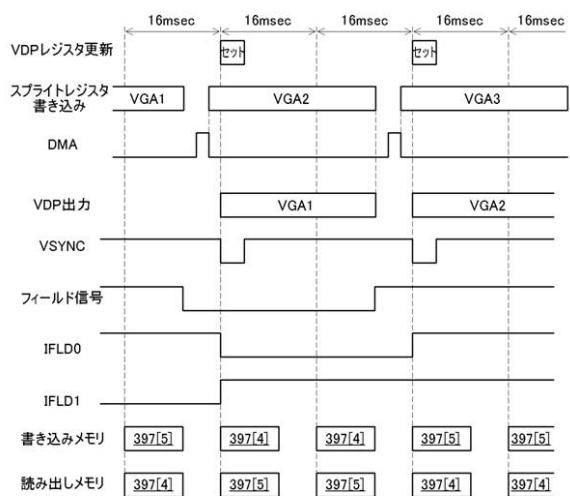
【図4】



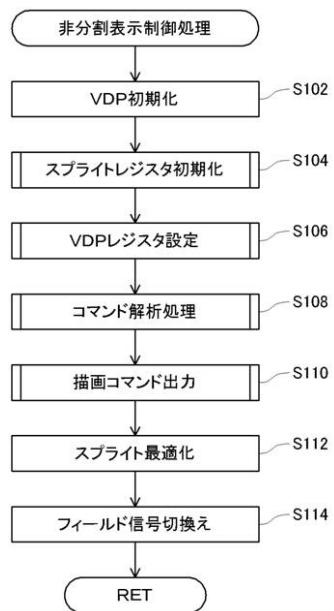
【図5】



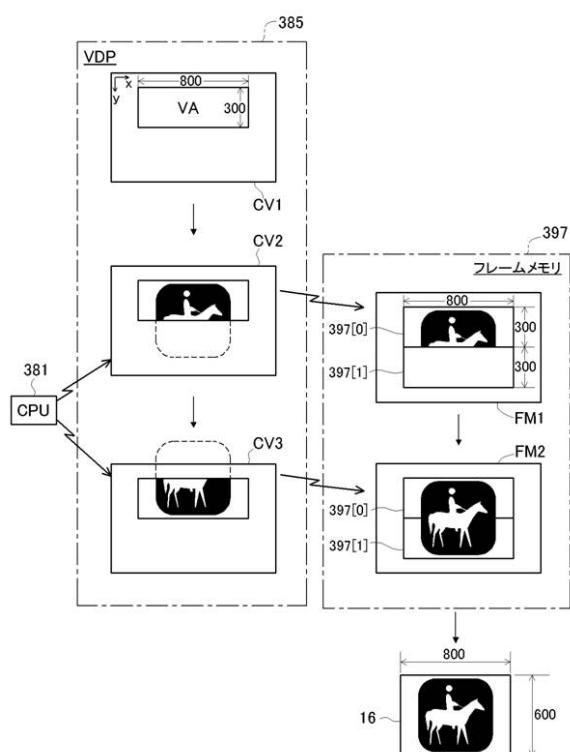
【図6】



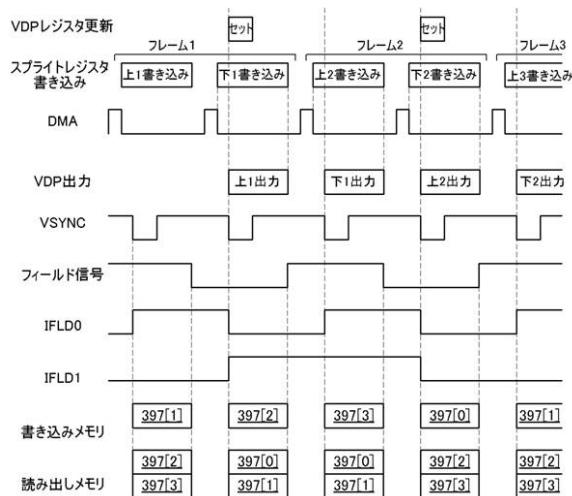
【図7】



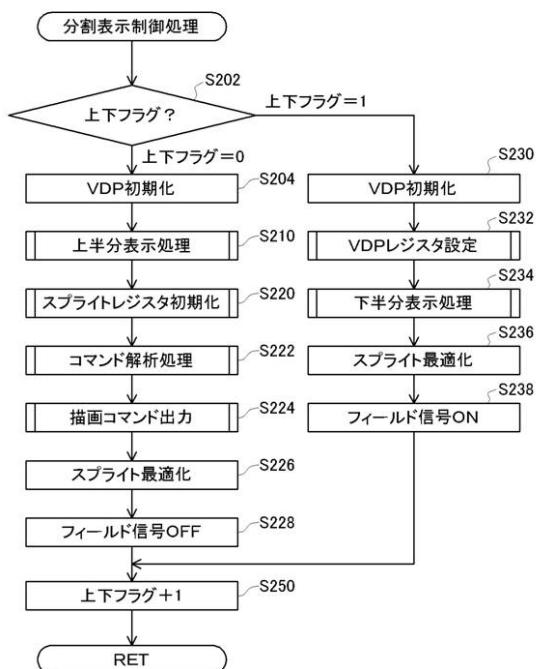
【図8】



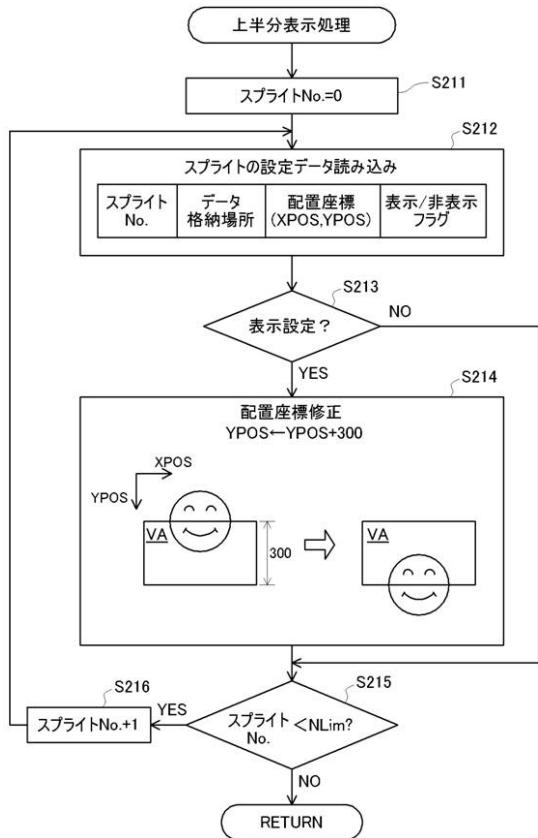
【図9】



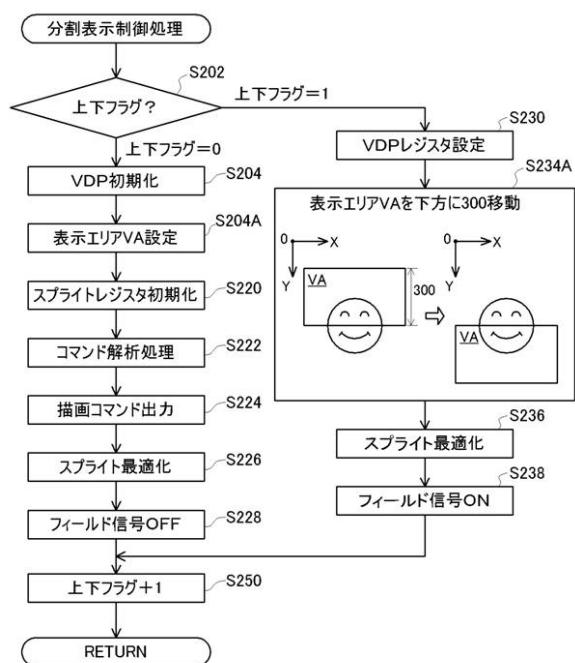
【図10】



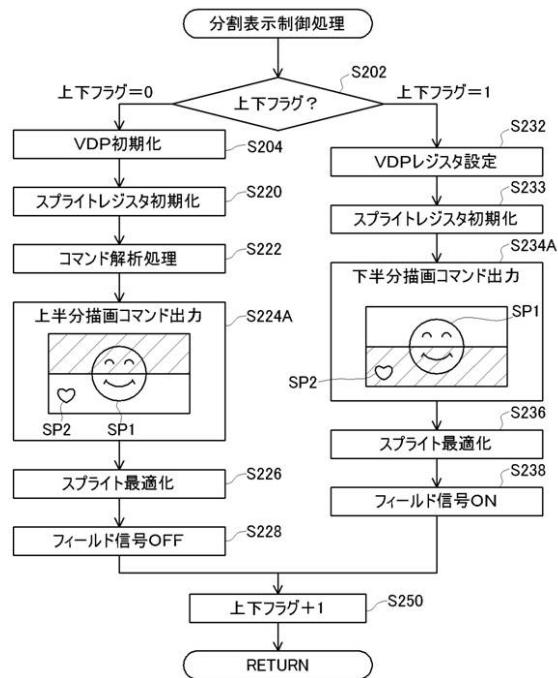
【図1-1】



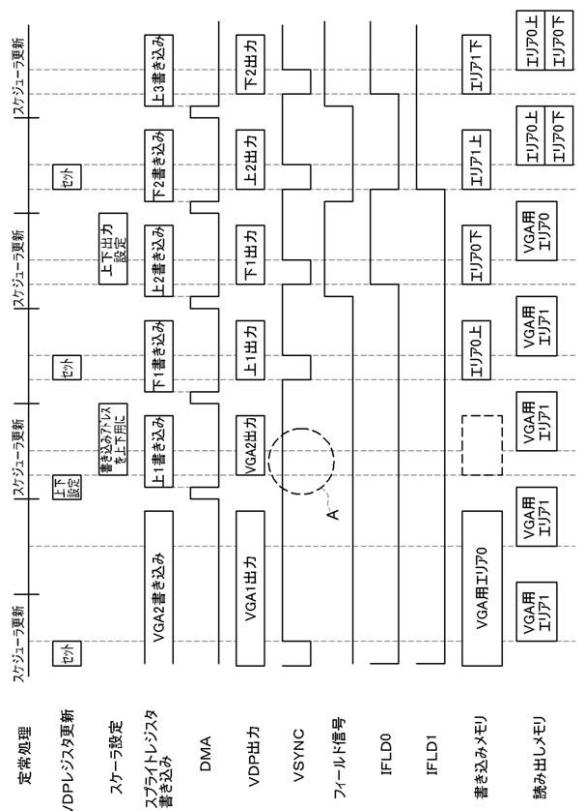
【図1-2】



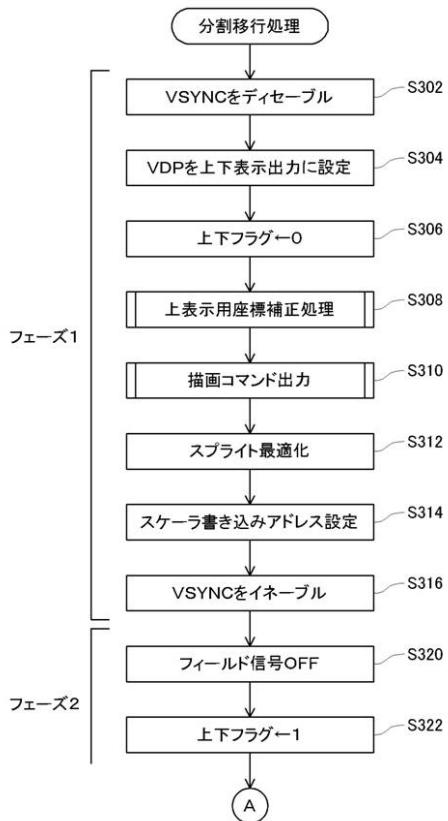
【図13】



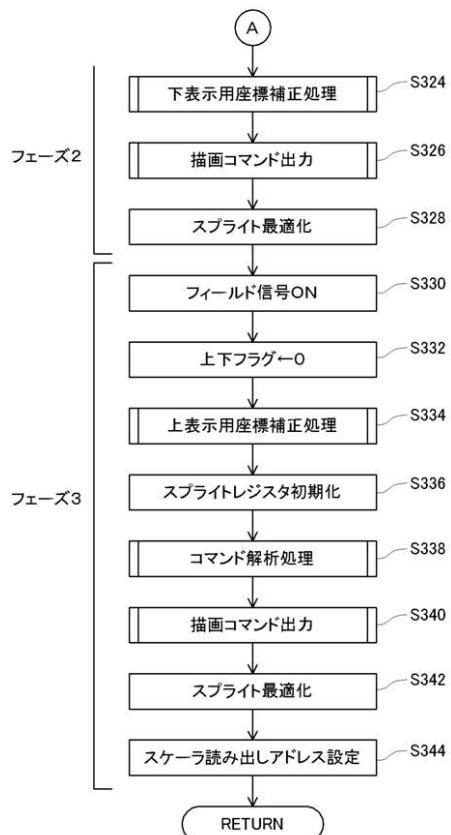
【図14】



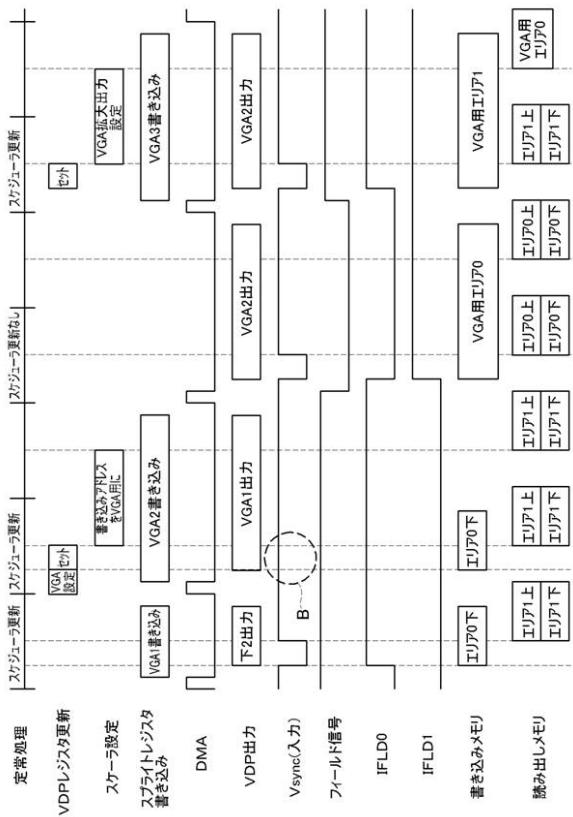
【図15】



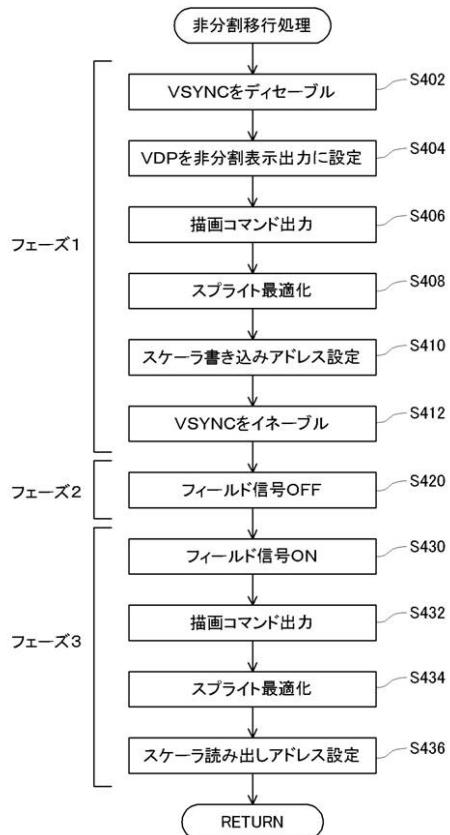
【図16】



【図17】



【図18】



---

フロントページの続き

(56)参考文献 特開2005-027833(JP,A)  
特開2004-242953(JP,A)  
特開2006-061629(JP,A)  
特開2002-126320(JP,A)

(58)調査した分野(Int.Cl., DB名)

A63F 7/02  
A63F 5/04