(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7527683号
(P7527683)

(45)発行日　令和6年8月5日(2024.8.5)

(24)登録日　令和6年7月26日(2024.7.26)

(51)国際特許分類
Ａ６３Ｆ　5/04 (2006.01)

ＦＩ
Ａ６３Ｆ　5/04　６１１Ｂ
Ａ６３Ｆ　5/04　６１１Ａ
Ａ６３Ｆ　5/04　６０３Ｅ

請求項の数　１　（全67頁）

(21)出願番号　特願2023-27049(P2023-27049)
(22)出願日　令和5年2月24日(2023.2.24)
(62)分割の表示　特願2018-130382(P2018-130382)の分割
原出願日　平成30年7月10日(2018.7.10)
(65)公開番号　特開2023-54299(P2023-54299A)
(43)公開日　令和5年4月13日(2023.4.13)
審査請求日　令和5年2月24日(2023.2.24)

(73)特許権者　598098526
株式会社ユニバーサルエンターテインメント
東京都江東区有明三丁目７番２６号 有明フロンティアビルＡ棟
(74)代理人　110001520
弁理士法人日誠国際特許事務所
(72)発明者　坂本 孝夫
東京都江東区有明３丁目７番２６号
(72)発明者　▲高▼橋 和紀
東京都江東区有明３丁目７番２６号
(72)発明者　久保田 税
東京都江東区有明３丁目７番２６号
(72)発明者　西田 篤
東京都江東区有明３丁目７番２６号

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】

遊技の演出の制御を行う演出制御手段と、

前記演出制御手段の制御により発音を行う発音手段と、

前記発音手段を駆動制御するための発音駆動手段と、を備え、

前記演出制御手段は、

演算処理を行う演算処理手段と、

前記演算処理手段を動作させるためのプログラム及びデータが記憶された第１記憶手段と、

前記第１記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第２記憶手段と、を有し、

前記演算処理手段は、前記発音手段の発音制御を行う発音制御タスクを実行し、

前記発音制御タスクは、前記第１記憶手段に記憶されたプログラムに含まれ、前記発音制御タスクで起動後に実行される複数の発音制御プログラムのうち、特定の発音制御プログラムを前記第２記憶手段に転送する転送処理を起動時に実行し、

前記特定の発音制御プログラムは、サウンドデータに基づいてサウンド制御データを生成する処理と、生成された前記サウンド制御データを登録し、前記発音駆動手段に前記サウンド制御データを送信する処理と、を実行するためのプログラムを含み、

前記特定の発音制御プログラム以外の発音制御プログラムは、前記第１記憶手段に記憶された状態で実行される

ことを特徴とする遊技機。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、パチスロ等の遊技機に関する。

【背景技術】

【０００２】

従来、複数の図柄がそれぞれの表面に配された複数のリールと、遊技メダルやコイン等（以下、「遊技媒体」という）が投入され、遊技者によりスタートレバーが操作されたことを検出し、複数のリールの回転の開始を要求するスタートスイッチと、複数のリールのそれぞれに対応して設けられたストップボタンが遊技者により押されたことを検出し、該当するリールの回転の停止を要求する信号を出力するストップスイッチと、複数のリールのそれぞれに対応して設けられ、それぞれの駆動力を各リールに伝達するステッピングモータと、スタートスイッチ及びストップスイッチにより出力された信号に基づいて、ステッピングモータの動作を制御し、各リールの回転及びその停止を行うリール制御装置とを備え、スタートレバーが操作されたことを検出すると、乱数値に基づいて抽籤を行い、この抽籤の結果（以下、「内部当籤役」という）とストップボタンが操作されたことを検出したタイミングとに基づいてリールの回転の停止を行う、いわゆるパチスロと称される遊技機が知られている。

【０００３】

この種の遊技機として、ＲＯＭに格納されたプログラムをＣＰＵに実行させる遊技機が特許文献１に提案されている。また、ＲＯＭに格納されたプログラムを起動時のブート処理によりＲＡＭに転送して、ＲＡＭに転送したプログラムをＣＰＵに実行させる遊技機が特許文献２に提案されている。

【先行技術文献】

【特許文献】

【０００４】

【文献】特開２０００－２９６２２３号公報

【文献】特開２００８－１４８８９１号公報

【発明の概要】

【発明が解決しようとする課題】

【０００５】

ＲＯＭに格納されたプログラムをＣＰＵなどの演算処理手段に実行させる従来の遊技機は、プログラムを実行するたびに演算処理手段がＲＯＭにアクセスするため、ＲＯＭのアクセス速度がボトルネックとなり、演算処理手段の処理速度を低下させていた。一方、ＲＡＭに転送したプログラムをＣＰＵなどの演算処理手段に実行させる従来の遊技機は、必要なＲＡＭの容量が増加するため、コストがかかってしまう。

【０００６】

本発明は、このような課題を解決するためになされたもので、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することを目的とする。

【課題を解決するための手段】

【０００７】

本発明に係る遊技機は、

遊技の演出の制御を行う演出制御手段（副制御回路４２）と、

前記演出制御手段の制御により発音を行う発音手段（スピーカ２０Ｌ，２０Ｒ）と、

前記発音手段を駆動制御するための発音駆動手段（サウンドＩＣ８８）と、を備え、

前記演出制御手段は、

演算処理を行う演算処理手段（サブＣＰＵ８１）と、

前記演算処理手段を動作させるためのプログラム及びデータが記憶された第１記憶手段（サブＲＯＭ８２）と、

前記第１記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第２記憶手段（サブＲＡＭ８３）と、を有し、

前記演算処理手段は、前記発音手段の発音制御を行う発音制御タスク（サウンド制御タスク）を実行し、

前記発音制御タスクは、前記第１記憶手段に記憶されたプログラムに含まれ、前記発音制御タスクで起動後に実行される複数の発音制御プログラムのうち、特定の発音制御プログラムを前記第２記憶手段に転送する転送処理（サウンド関数ＲＡＭ転送処理）を起動時に実行し、

前記特定の発音制御プログラムは、サウンドデータに基づいてサウンド制御データを生成する処理（サウンド制御データ生成処理）と、生成された前記サウンド制御データを登録し、前記発音駆動手段に前記サウンド制御データを送信する処理（サウンドドライバ）と、を実行するためのプログラムを含み、

前記特定の発音制御プログラム以外の発音制御プログラムは、前記第１記憶手段に記憶された状態で実行される

構成を有している。

【発明の効果】

【０００９】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

【図面の簡単な説明】

【００１０】

【図１】本発明の一実施形態の遊技機における機能フローを説明する説明図である。

【図２】本発明の一実施形態の遊技機における外観構成例を示す斜視図である。

【図３】本発明の一実施形態の遊技機におけるフロントパネルを外した状態の正面図である。

【図４】本発明の一実施形態の遊技機におけるＬＥＤ配置ポート図である。

【図５】本発明の一実施形態の遊技機における内部構造を示すものであり、フロントドアを開いた状態の斜視図である。

【図６】本発明の一実施形態の遊技機が備える回路の全体構成を示すブロック図である。

【図７】本発明の一実施形態の遊技機における副制御回路の内部構成を示すブロック図である。

【図８】本発明の一実施形態の遊技機における第１モードの制御データのデータ構造を示す図である。

【図９】本発明の一実施形態の遊技機における第２モードの制御データのデータ構造を示す図である。

【図１０】本発明の一実施形態の遊技機における第３モードの制御データのデータ構造を示す図である。

【図１１】本発明の一実施形態の遊技機における図柄配置表の一例を示す図である。

【図１２】本発明の一実施形態の遊技機における月別日テーブルを示す図である。

【図１３】本発明の一実施形態の遊技機における日時格納領域を示す図である。

【図１４】本発明の一実施形態の遊技機における再生状態管理格納領域を示す図である。

【図１５】本発明の一実施形態の遊技機におけるランプデータが格納された再生状態管理格納領域の具体例を示す図である。

【図１６】本発明の一実施形態の遊技機におけるランプデータの再生例を示す図である。

【図１７】本発明の一実施形態の遊技機における中断復帰用ランプデータが格納された再生状態管理格納領域の具体例を示す図である。

【図１８】本発明の一実施形態の遊技機における中断復帰用ランプデータの再生例を示す図である。

【図１９】本発明の一実施形態の遊技機における主制御回路の電源投入処理を示すフローチャートである。

【図２０】本発明の一実施形態の遊技機におけるメインＣＰＵの割込処理を示すフローチャートである。
【図２１】本発明の一実施形態の遊技機における副制御回路の電源投入処理を示すフローチャートである。
【図２２】本発明の一実施形態におけるサブＣＰＵにより行われる主基板通信タスクを示すフローチャートである。
【図２３】本発明の一実施形態におけるサブＣＰＵにより行われる演出登録タスクを示すフローチャートである。
【図２４】本発明の一実施形態における演出内容決定処理を示すフローチャートである。
【図２５】本発明の一実施形態における無操作コマンド受信時処理の例を示すフローチャートである。
【図２６】本発明の一実施形態における日時更新処理を示すフローチャートである。
【図２７】本発明の一実施形態におけるサブＣＰＵにより行われるサウンド制御タスクを示すフローチャートである。
【図２８】本発明の一実施形態におけるサウンド関数ＲＡＭ転送処理を示すフローチャートである。
【図２９】本発明の一実施形態におけるサウンド制御タスク及びサウンド関数ＲＡＭ転送処理の具体例をソースコードで表す概要図である。
【図３０】本発明の一実施形態におけるサブＣＰＵにより行われるランプ制御タスクを示すフローチャートである。
【図３１】本発明の一実施形態におけるランプデータ読み込み処理を示すフローチャートである。
【図３２】本発明の一実施形態におけるランプ関数ＲＡＭ転送処理を示すフローチャートである。
【図３３】本発明の一実施形態におけるランプ制御タスク、ランプデータ読み込み処理及びランプ関数ＲＡＭ転送処理の具体例をソースコードで表す概要図である。
【図３４】本発明の一実施形態におけるサブＣＰＵに実行させるプログラムの製造方法を説明するための概念図である。
【発明を実施するための形態】
【００１１】
以下、本発明の一実施形態を示す遊技機であるパチスロについて、図１～図３４を参照しながら説明する。
なお、本実施形態では、特定の図柄組合せが表示された場合にリプレイの当籤確率が通常時より高くなる遊技状態であるリプレイタイム（以下、「ＲＴ」という）が作動する機能を備えたパチスロについて説明する。
【００１２】
＜機能フロー＞
まず、図１を参照して、パチスロの機能フローについて説明する。
本実施形態のパチスロでは、遊技を行うための遊技媒体としてメダルを用いる。なお、遊技媒体としては、メダル以外にも、コイン、遊技球、遊技用のポイントデータ又はトークン等を適用することもできる。
【００１３】
遊技者によりメダルが投入され、スタートレバーが操作されると、予め定められた数値の範囲（例えば、０～６５５３５）の乱数から１つの値（以下、乱数値）が抽出される。
【００１４】
内部抽籤手段は、抽出された乱数値に基づいて抽籤を行い、内部当籤役を決定する。この内部抽籤手段は、後述する主制御回路が担う。内部当籤役の決定により、後述の入賞判定ラインに沿って表示を行うことを許可する図柄の組合せが決定される。なお、図柄の組合せの種別としては、メダルの払い出し、再遊技の作動、ボーナスの作動等といった特典が遊技者に与えられる「入賞」に係るものと、それ以外のいわゆる「ハズレ」に係るもの

とが設けられている。

【００１５】

また、スタートレバーが操作されると、複数のリールの回転が行われる。その後、遊技者により所定のリールに対応するストップボタンが押されると、リール停止制御手段は、内部当籤役とストップボタンが押されたタイミングとに基づいて、該当するリールの回転を停止する制御を行う。このリール停止制御手段は、後述する主制御回路が担う。

【００１６】

パチスロでは、基本的に、ストップボタンが押されたときから規定時間（１９０ｍｓｅｃ又は７５ｍｓｅｃ）内に、該当するリールの回転を停止する制御が行われる。本実施形態では、この規定時間内にリールの回転に伴って移動する図柄の数を「滑り駒数」と呼ぶ。規定期間が１９０ｍｓｅｃである場合には、滑り駒数の最大数を図柄４個分に定め、規定期間が７５ｍｓｅｃである場合には、滑り駒数の最大数を図柄１個分に定める。

【００１７】

リール停止制御手段は、入賞に係る図柄の組合せ表示を許可する内部当籤役が決定されているときは、通常、１９０ｍｓｅｃ（図柄４コマ分）の規定時間内に、その図柄の組合せが入賞判定ラインに沿って極力表示されるようにリールの回転を停止させる。また、リール停止制御手段は、例えば、第２種特別役物であるチャレンジボーナス（ＣＢ）及びＣＢを連続して作動させるミドルボーナス（ＭＢ）の動作時には、１つ以上のリールに対して、規定時間７５ｍｓｅｃ（図柄１コマ分）内に、その図柄の組合せが入賞判定ラインに沿って極力表示されるようにリールの回転を停止させる。さらに、リール停止制御手段は、遊技状態に対応する各種規定時間を利用して、内部当籤役によってその表示が許可されていない図柄の組合せが入賞判定ラインに沿って表示されないようにリールの回転を停止させる。

【００１８】

こうして、複数のリールの回転がすべて停止されると、入賞判定手段は、入賞判定ラインに沿って表示された図柄の組合せが、入賞に係るものであるか否かの判定を行う。この入賞判定手段は、後述する主制御回路が担う。入賞判定手段により入賞に係るものであるとの判定が行われると、メダルの払い出し等の特典が遊技者に与えられる。パチスロでは、以上のような一連の流れが１回の遊技として行われる。

【００１９】

また、パチスロでは、前述した一連の流れの中で、表示装置により行う映像の表示、各種ランプにより行う光の出力、スピーカにより行う音の出力、或いはこれらの組合せを利用して様々な演出が行われる。

【００２０】

スタートレバーが操作されると、上述した内部当籤役の決定に用いられた乱数値とは別に、演出用の乱数値（以下、演出用乱数値）が抽出される。演出用乱数値が抽出されると、演出内容決定手段は、内部当籤役に対応づけられた複数種類の演出内容の中から今回実行するものを抽籤により決定する。この演出内容決定手段は、後述する副制御回路が担う。

【００２１】

演出内容が決定されると、演出実行手段は、リールの回転開始時、各リールの回転停止時、入賞の有無の判定時等の各契機に連動させて対応する演出を実行する。このように、パチスロでは、内部当籤役に対応づけられた演出内容を実行することによって、決定された内部当籤役（言い換えると、狙うべき図柄の組合せ）を知る機会又は予想する機会が遊技者に提供され、遊技者の興味の向上を図ることができる。

【００２２】

＜パチスロの構造＞

次に、図２～図４を参照して、本実施形態におけるパチスロの構造について説明する。

【００２３】

［外観構造］

図２は、パチスロ１の外部構造を示す斜視図である。図３は、本実施形態におけるパチ

スロ１のフロントパネル１０を外した状態の正面図である。

【００２４】

図２に示すように、パチスロ１は、外装体２を備えている。外装体２は、リールや回路基板等を収容するキャビネット２ａと、キャビネット２ａに対して開閉可能に取り付けられるフロントドア２ｂとを有している。

キャビネット２ａの両側面には、把手７が設けられている（図２では一側面の把手７のみを示す）。この把手７は、パチスロ１を運搬するときに手をかける凹部である。

【００２５】

キャビネット２ａの内部には、３つのリール３Ｌ，３Ｃ，３Ｒが横並びに設けられている。以下、各リール３Ｌ，３Ｃ，３Ｒを、それぞれ左リール３Ｌ、中リール３Ｃ、右リール３Ｒという。各リール３Ｌ，３Ｃ，３Ｒは、円筒状に形成されたリール本体と、リール本体の周面に装着された透光性のシート材と、リール本体の内側からシート材に光を照射するリール用光源とを有している。シート材の表面には、複数（例えば２１個）の図柄が周方向に沿って所定の間隔をあけて描かれている。ここで、赤７図柄（不図示）の一部分は半透明部分とされている。赤また、リール本体には、シート材の背面に光を照射するリールバックライトが設けられている。このリールバックライトは、後述の副制御回路４２によって制御されて点灯及び消灯する。

【００２６】

フロントドア２ｂは、ドア本体９と、フロントパネル１０と、発光表示装置１１とを備えている。

ドア本体９は、ヒンジ（不図示）を用いてキャビネット２ａに開閉可能に取り付けられている。ヒンジは、パチスロ１の前方からドア本体９を見た場合に、ドア本体９における左側の端部に設けられている。

【００２７】

図２に示すように、発光表示装置１１は、ドア本体９の上部に設けられている。この発光表示装置１１は、マトリクス状に配置された複数の光源部によって形成されたドットマトリクス部１１９（図３参照）と、フロントパネル１０のドットマトリクス部１１９に対向する部分から構成されている。

【００２８】

ドットマトリクス部１１９は、任意の箇所の光源部を点灯（点滅）することで、フロントパネル１０に施されたデザインの任意の箇所（本実施形態では花火の絵柄）を背面から照明する。これにより、フロントパネル１０に施されたデザインの任意の箇所を発光させる演出が行われる。

【００２９】

発光表示装置１１の下方には、３つのリール３Ｌ，３Ｃ，３Ｒに描かれた図柄を表示する表示窓４Ｌ，４Ｃ，４Ｒが設けられている。以下、各表示窓４Ｌ，４Ｃ，４Ｒを、それぞれ左表示窓４Ｌ、中表示窓４Ｃ、右表示窓４Ｒという。

【００３０】

表示窓４Ｌ，４Ｃ，４Ｒは、例えばアクリル板等の透明な部材で形成されている。この表示窓４Ｌ，４Ｃ，４Ｒは、正面（遊技者側）から見て、３つのリールの配置領域と重畳する位置に設けられ、かつ、３つのリールより手前（遊技者側）に位置するように設けられる。したがって、遊技者は、表示窓４Ｌ，４Ｃ，４Ｒを介して、表示窓４Ｌ，４Ｃ，４Ｒの背後に設けられた３つのリールを視認することができる。

【００３１】

本実施形態では、表示窓４Ｌ，４Ｃ，４Ｒは、その背後に設けられた対応するリールの回転が停止したとき、各リールに描かれた複数種類の図柄のうち、連続して配置された３つの図柄を表示できる大きさに設定されている。すなわち、表示窓４Ｌ，４Ｃ，４Ｒの枠内には、リール毎に上段、中段及び下段の各領域が設けられ、各領域に１個の図柄が表示される。

【００３２】

フロントパネル１０は、ドア本体９の上部に取り付けられている。このフロントパネル１０は、上表示部１０１と、リール照明部１０２と、リールサイド演出表示部１０３Ａ，１０３Ｂと、エッジ演出表示部１０４Ａ，１０４Ｂと、リール下表示部１０５とを有している。

【００３３】

上表示部１０１は、発光表示装置１１の上方に配置されており、リール照明部１０２は、リール３Ｌ，３Ｃ，３Ｒと発光表示装置１１との間に配置されている。リールサイド演出表示部１０３Ａ，１０３Ｂは、リール３Ｌ，３Ｃ，３Ｒの側方に配置されており、エッジ演出表示部１０４Ａ，１０４Ｂは、リール横演出表示部１０３の側方に配置されている。リール下表示部１０５は、リール３Ｌ，３Ｃ，３Ｒの下方に配置されている。

【００３４】

上表示部１０１、リール照明部１０２、リールサイド演出表示部１０３Ａ，１０３Ｂ、エッジ演出表示部１０４Ａ，１０４Ｂ及びリール下表示部１０５は、ドア本体９に設けられた後述する各種ランプ群１１１～１１７を覆っている。そして、これら上表示部１０１，１０２，１０３Ａ，１０３Ｂ，１０４Ａ，１０４Ｂ，１０５は、各種ランプ群１１１～１１７からの光が照射されて発光する。

【００３５】

例えば、リールサイド演出表示部１０３Ａには、上下方向に並ぶ３つのＢＥＴ発光部が設けられている。３つのＢＥＴ発光部の点灯する数は、１回の遊技に使用するメダルの数を示す。

【００３６】

本実施形態では、１回の遊技に使用するメダルの数を１～３に設定している。例えば、１回の遊技に使用するメダルの数を「１」にした場合は、１つのＢＥＴ発光部（例えば一番下に位置するＢＥＴ発光部）が点灯し、その他ＢＥＴ発光部（真ん中と一番上に位置するＢＥＴ発光部）が消灯する。

【００３７】

図２に示すように、ドア本体９の中央には、台座部１２が形成されている。この台座部１２には、遊技者の操作対象となる各種装置（メダル投入口１３、ＭＡＸベットボタン１４、１ベットボタン１５、スタートレバー１６、ストップボタン１７Ｌ，１７Ｃ，１７Ｒ）が設けられている。

【００３８】

メダル投入口１３は、遊技者によって外部からパチスロ１に投下されるメダルを受け入れるために設けられる。メダル投入口１３から受け入れられたメダルは、予め設定された枚数（例えば３枚）を上限として１回の遊技に使用され、予め設定された枚数を超えた分は、パチスロ１の内部に預けることができる（いわゆるクレジット機能）。

【００３９】

ＭＡＸベットボタン１４及び１ベットボタン１５は、パチスロ１の内部に預けられているメダルから１回の遊技に使用する枚数を決定するために設けられる。なお、図２には示さないが、台座部１２には、精算ボタンが設けられる。この精算ボタンは、パチスロ１の内部に預けられているメダルを外部に引き出す（排出する）ために設けられる。

【００４０】

スタートレバー１６は、全てのリール（３Ｌ，３Ｃ，３Ｒ）の回転を開始するために設けられる。ストップボタン１７Ｌ，１７Ｃ，１７Ｒは、それぞれ、左リール３Ｌ、中リール３Ｃ、右リール３Ｒに対応づけて設けられ、各ストップボタンは対応するリールの回転を停止するために設けられる。以下、ストップボタン１７Ｌ，１７Ｃ，１７Ｒを、それぞれ左ストップボタン１７Ｌ、中ストップボタン１７Ｃ、右ストップボタン１７Ｒという。

【００４１】

また、台座部１２には、７セグメントＬＥＤ（Light Emitting Diode）からなる７セグ表示器６が設けられている。この７セグ表示器６は、特典として遊技者に対して払い出すメダルの枚数（以下、払出枚数）、パチスロ１の内部に預けられているメダルの枚数（

以下、クレジット枚数）、遊技を行うためのメダルの投入枚数（以下、ＢＥＴ枚数）等の情報をデジタル表示する。

【００４２】

ドア本体９の下部には、メダル払出口１８、メダル受皿１９、スピーカ２０Ｌ，２０Ｒ等が設けられている。メダル払出口１８は、後述のメダル払出装置３３の駆動により排出されるメダルを外部に導く。メダル受皿１９は、メダル払出口１８から排出されたメダルを貯める。また、スピーカ２０Ｌ，２０Ｒは、演出内容に対応する効果音や楽曲等の音を出力する。

【００４３】

また、ドア本体９には、腰部パネル表示部１０６が設けられている。この腰部パネル表示部１０６は、スピーカ２０Ｌ，２０Ｒの上方に配置されている。腰部パネル表示部１０６は、ドア本体９に設けられた後述する光源部（Ｎｏ.５３のポート）を覆っている。そして、腰部パネル表示部１０６は、光源部（Ｎｏ.５３のポート）からの光が照射されて発光する。

【００４４】

［各種ランプ群］

図４は、パチスロ１におけるＬＥＤ配置ポート図である。

【００４５】

図４に示すように、ドア本体９には、３３７ポートの光源部が設けられている。各光源部には、発光体として少なくとも１つのＬＥＤが配置されている。なお、各光源部には、有機エレクトロルミネッセンス等の他の発光体を配置してもよい。

【００４６】

Ｎｏ.０～Ｎｏ.３０のポートに係る光源部は、第１ランプ群１１１を形成している。この第１ランプ群１１１は、上表示部１０１（図３参照）に光を照射する。Ｎｏ.４９～Ｎｏ.５２のポートに係る光源部は、第２ランプ群１１２を形成している。この第２ランプ群１１２は、リール照明部１０２（図３参照）に光を照射する。

【００４７】

Ｎｏ.６６～Ｎｏ.７０のポートに係る光源部は、第３ランプ群１１３を形成し、Ｎｏ.７１～Ｎｏ.７５のポートに係る光源部は、第４ランプ群１１４を形成している。第３ランプ群１１３は、リールサイド演出表示部１０３Ａ（図３参照）に光を照射し、第４ランプ群１１４は、リールサイド演出表示部１０３Ｂ（図３参照）に光を照射する。第３ランプ群１１３のＮｏ.６８～Ｎｏ.７０のポートに係る光源部は、上述した３つのＢＥＴ発光部に対向し、点灯することでそれぞれ対向するＢＥＴ発光部を発光させる。

【００４８】

Ｎｏ.５４～Ｎｏ.５９のポートに係る光源部は、第５ランプ群１１５を形成し、Ｎｏ.６０～Ｎｏ.６５のポートに係る光源部は、第６ランプ群１１６を形成している。第５ランプ群１１５は、エッジ演出表示部１０４Ａ（図３参照）に光を照射し、第６ランプ群１１６は、エッジ演出表示部１０４Ｂ（図３参照）に光を照射する。

【００４９】

Ｎｏ.１０８～Ｎｏ.１５６のポートに係る光源部は、第７ランプ群１１７を形成している。第７ランプ群１１７は、リール下表示部１０５（図３参照）に光を照射する。第７ランプ群１１７のＮｏ.１０８～Ｎｏ.１２１のポートに係る光源部は、例えば、パチスロ１の内部に預けられているメダルの枚数である貯留枚数を表示する。

【００５０】

第７ランプ群１１７のＮｏ.１２２～Ｎｏ.１４２のポートに係る光源部は、例えば、ボーナス中のメダルの獲得枚数を表示する。第７ランプ群１１７のＮｏ.１４３～Ｎｏ.１５６のポートに係る光源部は、例えば、メダルの払出枚数を表示する。

【００５１】

Ｎｏ.３１～Ｎｏ.４８及びＮｏ.７６～Ｎｏ.９３のポートに係る光源部は、第８ランプ群１１８を形成している。第８ランプ群１１８のＮｏ.３１～Ｎｏ.３６のポートに係る光源

部は、左リール３Ｌのリール用光源（リールバックライト）として機能する。また、Ｎｏ.３７～Ｎｏ.４２のポートに係る光源部は、中リール３Ｃのリール用光源（リールバックライト）として機能し、Ｎｏ.４３～Ｎｏ.４８のポートに係る光源部は、右リール３Ｒのリール用光源（リールバックライト）として機能する。

【００５２】

Ｎｏ.５３のポートに係る光源部は、腰部パネル表示部１０６（図３参照）に光を照射する。Ｎｏ.１５７のポートに係る光源部は、ＭＡＸＢＥＴボタン１４に光を照射する。左ストップボタン１７Ｌに光を照射する。Ｎｏ.１５８のポートに係る光源部は、左ストップボタン１７Ｌに光を照射する。Ｎｏ.１５９のポートに係る光源部は、中ストップボタン１７Ｃに光を照射する。Ｎｏ.１６０のポートに係る光源部は、右ストップボタン１７Ｒに光を照射する。

【００５３】

Ｎｏ.１６１～Ｎｏ.３３７のポートに係る光源部は、ドットマトリクス部１１９を形成している。このドットマトリクス部１１９は、発光表示装置１１の光源として機能する。ドットマトリクス部１１９の各ポートには、１個のＬＥＤが設けられている。

【００５４】

［内部構造］

次に、パチスロ１の内部構造を、図５を参照しながら説明する。図５は、パチスロ１の内部構造を示す斜視図である。

【００５５】

キャビネット２ａは、正面側の一面が開口された略直方体状に形成されている。このキャビネット２ａ内の上部には、後述の主制御回路４１（図６参照）を構成する主基板３１が設けられている。主制御回路４１は、内部当籤役の決定、各リールの回転及び停止、入賞の有無の判定等の、パチスロ１における遊技の主な動作及び該動作間の流れを制御する回路である。なお、主制御回路４１の具体的な構成は後述する。

【００５６】

キャビネット２ａ内の中央部には、３つのリール（左リール３Ｌ、中リール３Ｃ及び右リール３Ｒ）が設けられている。なお、図５には示さないが、各リールは、所定の減速比を有する歯車を介して対応する後述のステッピングモータ（図６中のステッピングモータ６１Ｌ，６１Ｃ，６１Ｒのいずれか）に接続される。

【００５７】

キャビネット２ａ内の下部には、多量のメダルを収容可能であり、かつ、それらを１枚ずつ排出可能な構造を有するメダル払出装置３３（以下、ホッパー３３という）が設けられている。また、キャビネット２ａ内における、ホッパー３３の一方の側部（図５に示す例では左側）には、パチスロ１が有する各装置に対して必要な電力を供給する電源装置３４が設けられている。

【００５８】

フロントドア２ｂの裏面側（表示画面側とは反対側の部分）における上部には、後述の副制御回路４２（図６及び図７参照）を構成する副基板３２が設けられている。副制御回路４２は、映像の表示等による演出の実行を制御する回路である。なお、副制御回路４２の具体的な構成は後述する。

【００５９】

さらに、フロントドア２ｂの裏面側における略中央部には、セレクタ３５が設けられている。セレクタ３５は、メダル投入口１３（図２参照）を介して外部から投入されたメダルの材質や形状等が適正である否かを選別する装置であり、適正であると判定したメダルをホッパー３３に案内する。また、図５には示さないが、セレクタ３５内においてメダルが通過する経路上には、適正なメダルが通過したことを検出するメダルセンサ３５Ｓ（図６参照）が設けられている。

【００６０】

＜パチスロが備える回路の構成＞

次に、パチスロ１が備える回路の構成について、図６及び図７を参照して説明する。
図６は、パチスロ１が備える回路全体のブロック構成図である。図７は、副制御回路の内部構成を示すブロック構成図である。
【００６１】
パチスロ１は、主制御回路４１、副制御回路４２、及び、これらの回路と電気的に接続される周辺装置（アクチュエータ）を備える。
【００６２】
［主制御回路］
主制御回路４１は、主に、回路基板（主基板３１）上に設置されたマイクロコンピュータ５０により構成される。それ以外の構成要素として、主制御回路４１は、クロックパルス発生回路５４、分周器５５、乱数発生器５６、サンプリング回路５７、表示部駆動回路６４、ホッパー駆動回路６５、及び、払出完了信号回路６６を含む。
【００６３】
マイクロコンピュータ５０は、メインＣＰＵ５１、メインＲＯＭ（Read Only Memory）５２及びメインＲＡＭ（Random Access Memory）５３により構成される。
【００６４】
メインＲＯＭ５２には、メインＣＰＵ５１により実行される各種処理の制御プログラム、内部抽籤テーブル等のデータテーブル、副制御回路４２に対して各種制御指令（コマンド）を送信するためのデータ等が記憶されている。メインＲＡＭ５３には、制御プログラムの実行により決定された内部当籤役等の各種データを格納する格納領域が設けられている。
【００６５】
メインＣＰＵ５１には、クロックパルス発生回路５４、分周器５５、乱数発生器５６及びサンプリング回路５７が接続されている。クロックパルス発生回路５４及び分周器５５は、クロックパルスを発生する。なお、メインＣＰＵ５１は、発生されたクロックパルスに基づいて、制御プログラムを実行する。また、乱数発生器５６は、予め定められた範囲の乱数（例えば、０～６５５３５）を発生する。そして、サンプリング回路５７は、発生された乱数の中から１つの値を抽出する。
【００６６】
マイクロコンピュータ５０の入力ポートには、各種スイッチ及びセンサ等が接続される。メインＣＰＵ５１は、各種スイッチ等からの入力信号を受けて、ステッピングモータ６１Ｌ，６１Ｃ，６１Ｒ等の周辺装置の動作を制御する。
【００６７】
ストップスイッチ１７Ｓは、本発明に係る停止操作検出手段の一具体例を示すものであり、左ストップボタン１７Ｌ、中ストップボタン１７Ｃ、右ストップボタン１７Ｒのそれぞれが遊技者により押されたこと（停止操作）を検出する。スタートスイッチ１６Ｓは、本発明に係る開始操作検出手段の一具体例を示すものであり、スタートレバー１６が遊技者により操作されたこと（開始操作）を検出する。精算スイッチ１４Ｓは、精算ボタンが遊技者により押されたことを検出する。
【００６８】
メダルセンサ３５Ｓは、本発明に係る投入操作検出手段の一具体例を示すものであり、メダル投入口１３に投入されたメダルがセレクタ３５内を通過したことを検出する。また、ベットスイッチ１２Ｓは、ベットボタン（ＭＡＸベットボタン１４又は１ベットボタン１５）が遊技者により押されたことを検出する。
【００６９】
また、マイクロコンピュータ５０により動作が制御される周辺装置としては、３つのステッピングモータ６１Ｌ，６１Ｃ，６１Ｒ、７セグ表示器６及びホッパー３３がある。また、マイクロコンピュータ５０の出力ポートには、各周辺装置の動作を制御するための駆動回路が接続される。
【００７０】

モータ駆動回路６２は、左リール３Ｌ、中リール３Ｃ、右リール３Ｒに対応してそれぞれ設けられた３つのステッピングモータ６１Ｌ，６１Ｃ，６１Ｒの駆動を制御する。リール位置検出回路６３は、センサ発光部とセンサ受光部とを有する光センサにより、リールが一回転したことを示すリールインデックスをリール毎に検出する。

【００７１】

３つのステッピングモータ６１Ｌ，６１Ｃ，６１Ｒのそれぞれは、その運動量がパルスの出力数に比例し、回転軸を指定された角度で停止させることが可能な構成を有する。また、各ステッピングモータの駆動力は、所定の減速比を有する歯車を介して、対応するリールに伝達される。そして、各ステッピングモータに対して１回のパルスが出力されるごとに、対応するリールは一定の角度で回転する。３つのリール３Ｌ，３Ｃ，３Ｒ及び３つのステッピングモータ６１Ｌ，６１Ｃ，６１Ｒは、本発明に係る変動表示手段の一具体例を示すものである。

【００７２】

メインＣＰＵ５１は、各リールのリールインデックスを検出してから対応するステッピングモータに対してパルスが出力された回数をカウントすることによって、各リールの回転角度（具体的には、リールが図柄何個分だけ回転したか）を管理する。

【００７３】

ここで、各リールの回転角度の管理を具体的に説明する。各ステッピングモータに対して出力されたパルスの数は、メインＲＡＭ５３に設けられたパルスカウンタ（不図示）によって計数される。そして、図柄１個分の回転に必要な所定回数（例えば１６回）のパルスの出力がパルスカウンタで計数されるごとに、メインＲＡＭ５３に設けられた図柄カウンタ（不図示）の値に、「１」が加算される。なお、図柄カウンタは、リール毎に設けられる。そして、図柄カウンタの値は、リール位置検出回路６３によってリールインデックスが検出されるとクリアされる。

【００７４】

すなわち、本実施形態では、図柄カウンタの値を管理することにより、リールインデックスが検出されてから図柄何個分の回転動作が行われたのかを管理する。それゆえ、各リールの各図柄の位置は、リールインデックスが検出される位置を基準として検出される。

【００７５】

なお、表示部駆動回路６４は、７セグ表示器６の動作を制御する。ホッパー駆動回路６５は、ホッパー３３の動作を制御する。払出完了信号回路６６は、ホッパー３３に設けられたメダル検出部３３Ｓが行うメダルの検出を管理し、ホッパー３３から外部に排出されたメダルが所定の払出枚数に達したか否かをチェックする。また、主制御回路４１には、外部端子板１８Ｓが接続されている。主制御回路４１は、外部端子板１８Ｓを介してホールコンピュータ又は呼出装置１００に接続されている。

【００７６】

［副制御回路］

図７に示すように、副制御回路４２は、主制御回路４１と電気的に接続され、主制御回路４１から送信されるコマンドに基づいて演出内容の決定や実行等の処理を行う。副制御回路４２は、サブＣＰＵ８１、サブＲＯＭ８２、サブＲＡＭ８３（演算記憶手段）、発振回路８４、バックアップＲＡＭ８５、ＲＴＣ（Real-Time Clock）８６、バッテリ８７及びサウンドＩＣ８８を含んで構成される。

【００７７】

サブＣＰＵ８１は、主制御回路４１から送信されたコマンドに応じて、サブＲＯＭ８２に記憶されている制御プログラムに従い、音、光の出力制御を行う。サブＣＰＵ８１は、制御部及び演算処理手段を構成する。

【００７８】

サブＲＯＭ８２は、本発明の遊技機に係る記憶手段の一具体例を示すものであり、基本的には、プログラム記憶領域及びデータ記憶領域を有する。このように、サブＲＯＭ８２は、サブＣＰＵ８１を動作させるためのプログラム及びデータが記憶された第１記憶手段

を構成する。

【００７９】

プログラム記憶領域には、サブＣＰＵ８１が実行する各種制御プログラムが記憶される。なお、プログラム記憶領域に格納される制御プログラムには、例えば、主制御回路４１との通信を制御するための主基板通信タスク、演出用乱数値を抽出して演出内容（演出データ）の決定及び登録を行うための演出登録タスク、決定した演出内容に基づいて発光表示装置１１のドットマトリクス部１１９及び各種ランプ群による光の出力を制御するためのランプ制御タスク、スピーカ２０Ｌ，２０Ｒによる音の出力を制御するためのサウンド制御タスク等のプログラムが含まれる。

【００８０】

データ記憶領域には、例えば、各種データテーブルを記憶する記憶領域、各種演出内容を構成する演出データを記憶する記憶領域、ＢＧＭや効果音に関するサウンドデータを記憶する記憶領域、光の点消灯のパターンに関するランプデータを記憶する記憶領域等の各種記憶領域が含まれる。なお、ランプデータについては、図１４～図１８を参照して後述する。

【００８１】

サブＲＡＭ８３は、サブＲＯＭ８２よりもアクセス速度が高く、かつ、書き換え可能な第２記憶手段を構成し、例えば、ＤＲＡＭ（Dynamic Random Access Memory）、又は、ＳＲＡＭ（Static Random Access Memory）によって構成されている。

【００８２】

サブＲＡＭ８３は、決定された演出内容や演出データを登録する格納領域や、主制御回路４１から送信される遊技状態や内部当籤役等の各種データを格納する演出状態格納領域などを有する。発振回路８４は、サブＣＰＵ８１を動作させるための所定周波数、例えば、２２ＭＨｚのクロックを生成（発振）するクロック生成手段を構成する。

【００８３】

バックアップＲＡＭ８５は、例えば、電源が不要な書き換え可能な不揮発性メモリであるＦＲＡＭ（Ferroelectric Random Access Memory）（登録商標）によって構成されている。なお、バックアップＲＡＭ８５は、ＲＴＣ８６に電源を供給するバッテリ８７をＳＲＡＭにも接続することにより構成してもよい。

【００８４】

バックアップＲＡＭ８５には、ＲＴＣ８６が計測する日時を記憶する日時記憶領域と、副制御回路４２の遊技状態に関する情報、及び演出に関する情報が記憶される遊技情報領域と、エラー情報履歴を格納する格納領域と、ホールメニューで設定するパチスロ１の各種設定を格納する格納領域と、を有する。

【００８５】

なお、バックアップＲＡＭ８５の遊技情報領域は、任意のタイミング（例えば、副制御回路４２への電源供給が断たれた時（電断発生時）、スタートコマンドを受信した時など）で、サブＲＡＭ８３の演出状態格納領域をバックアップＲＡＭ８５の遊技情報領域にコピーされ、電源投入処理（図１９参照）の直後に、サブＲＡＭ８３の演出状態格納領域にコピーされる。パチスロ１は、バックアップＲＡＭ８５の遊技情報領域をサブＲＡＭ８３の演出状態格納領域にコピーすることで、電源投入前に実行されていた遊技の演出を再開することが可能となる。

【００８６】

ＲＴＣ８６は、パチスロ１の電源がオン状態であれば、副制御回路４２に搭載された電子部品と同じく電源基板（不図示）から供給される電力で動作し、パチスロ１の電源がオフ状態であれば、バッテリ８７から供給される電力によって動作し、日時を計時する計時手段を構成する。

【００８７】

ＲＴＣ８６は、サブＣＰＵ８１とＩ２Ｃ（Inter-Integrated Circuit）によって接続され、サブＣＰＵ８１からの要求に応じて計測した日時を表す日時データをサブＣＰＵ８１

に送信する。バッテリ８７は、コイン形リチウム電池などの一次電池、又は、リチウムイオン二次電池などによって構成される。

【００８８】

サウンドＩＣ８８には、スピーカ２０Ｌ，２０Ｒが接続されている。サウンドＩＣ８８は、サブＣＰＵ８１から送信されたサウンドデータにしたがってＢＧＭ等の音声信号をスピーカ２０Ｌ，２０Ｒから出力させる。

【００８９】

副制御回路４２は、第８ランプ群１１８を駆動するためのドライバＩＣ１２８ａ～１２８ｃと、第１ランプ群１１１を駆動するためのドライバＩＣ１２１ａ～１２１ｂと、第７ランプ群１１７を駆動するためのドライバＩＣ１２７ａ～１２７ｄと、ドットマトリクス部１１９を駆動するためのドライバＩＣ１２９ａ～１２９ｇとを更に含んで構成される。

【００９０】

これらのドライバＩＣとサブＣＰＵ８１は、シリアルバス通信によって接続されている。サブＣＰＵ８１にはシリアルバス通信用の通信回路（不図示）が２回線内蔵され、チャネル０にはドライバＩＣ１２８ａ～１２８ｃ、ドライバＩＣ１２１ａ～１２１ｂ及びドライバＩＣ１２７ａ～１２７ｄに接続され、チャネル１にはドライバＩＣ１２９ａ～１２９ｇが接続されている。

【００９１】

本実施形態において、第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９は、発光部を構成する。ドライバＩＣ１２８ａ～１２８ｃ、ドライバＩＣ１２１ａ～１２１ｂ、ドライバＩＣ１２７ａ～１２７ｄ及びドライバＩＣ１２９ａ～１２９ｇ（以下、単に総称して「ドライバＩＣ」ともいう）は、発光駆動手段を構成する。

【００９２】

本実施形態において、各ドライバＩＣは、２４個の出力端子を有する。したがって、各ドライバＩＣは、チャネル０からチャネル２３の２４系統のＬＥＤを駆動することができる。

【００９３】

各ドライバＩＣは、出力設定端子を有する。各ドライバＩＣは、出力設定端子の状態（Ｈｉｇｈ／Ｌｏｗ）に応じて、出力端子の出力を定電流出力とシンク出力とのいずれかに設定する。したがって、各ドライバＩＣの出力端子は、ＬＥＤの接続状態に応じて、定電流出力とシンク出力とのいずれかに設定することができる。

【００９４】

各ドライバＩＣは、各チャネルに対応する６ビットのレジスタを内蔵し、レジスタに設定された値に応じたデューティ比の駆動信号を各チャネルに対して６３μｓ周期で出力する。

【００９５】

例えば、レジスタの値が６３の場合には、ドライバＩＣから該当チャネルにデューティ比が１００％（すなわち、パルス幅が６３μｓ）の駆動信号が出力され、該当チャネルのＬＥＤが最も高い輝度で発光する。

【００９６】

一方、レジスタの値が０の場合には、ドライバＩＣから該当チャネルにデューティ比が０％（すなわち、パルス幅が０）の駆動信号が出力され、該当チャネルのＬＥＤが消灯する。

【００９７】

本実施形態では、サブＣＰＵ８１と各ドライバＩＣとがシリアルバス通信方式により通信を行う。シリアルバス通信は、データ線・クロック線の２線、又は、データ線・クロック線・セレクト線の３線を用いた同期式シリアル通信により、各ドライバＩＣに各ランプ群１１１，１１７，１１８及びドットマトリクス部１１９のＬＥＤの発光態様を制御させるための制御データが送信される。

【００９８】
サブＣＰＵ８１は、制御データを出力するための複数の出力ポートを有し、各出力ポートから複数のドライバＩＣに、制御データを送信することができる。本実施形態において、サブＣＰＵ８１は、制御データを出力するための第１ポートと第２ポートとを有する。第１ポートには、ドライバＩＣ１２８ａ～１２８ｃ、ドライバＩＣ１２１ａ～１２１ｂ及びドライバＩＣ１２７ａ～１２７ｄが接続されている。第２ポートには、ドライバＩＣ１２９ａ～１２９ｇが接続されている。
【００９９】
各ドライバＩＣは、複数のアドレス端子を有し、各アドレス端子のレベル（Ｈｉｇｈ／Ｌｏｗ）によって、サブＣＰＵ８１の出力ポート内で、ユニークなアドレスが割り当てられている。
【０１００】
図８～図１０を参照してそれぞれ説明するように、サブＣＰＵ８１の各出力ポートから送信される制御データは、各ドライバＩＣに全チャネルのＬＥＤの輝度を設定する第１モードと、指定したチャネルのＬＥＤの輝度を設定する第２モードと、全チャネルのＬＥＤを消灯させるコマンドを含む第３モードとの３つのモードがある。
【０１０１】
なお、図８～図１０において、１行目は、制御データにおけるビット順序を表し、２行目は、制御データに含まれる各パラメータの種別を表し、３行目は、後述するエラー検出用データＥを表している。
【０１０２】
エラー検出用データＥは、制御データにおける各パラメータの位置によらずに、制御データの所定の位置に挿入される。このため、図８～図１０においては、制御データに含まれる各パラメータの種別と、エラー検出用データＥとを２行に分けて示している。すなわち、図８～図１０において、３行目に「Ｅ」が示されている各ビットは、エラー検出用データＥに相当し、３行目に「Ｅ」が示されていない各ビットは、２行目に示すパラメータに相当する。
【０１０３】
図８～図１０において、４行目は、サブＣＰＵ８１の各出力ポートから実際に送信される制御データを表す。なお、４行目に示す制御データにおいて、「＊」は、制御データに応じて設定される値（「０」又は「１」）を表す。
【０１０４】
（第１モード）
図８に示すように、第１モードにおける制御データは、ヘッダとして、先頭検出ビットＨＤと、デバイス識別子ＤＥＶと、アドレスＡＤＲと、モードＭＤと、所定の位置に挿入されるエラー検出用データＥ（例えば、固定値「０」）とを含む。
【０１０５】
具体的には、第１モードにおける制御データのヘッダは、２２ビットよりなる。先頭検出ビットＨＤは、１ビット目から９ビット目の９ビットに割り当てられている。先頭検出ビットＨＤは、図示したように、固定値である。
【０１０６】
デバイス識別子ＤＥＶは、１１ビット目から１４ビット目の４ビットに割り当てられている。デバイス識別子ＤＥＶは、各ドライバＩＣの種別を表す。したがって、同種のドライバＩＣであれば同一な値となる。
【０１０７】
アドレスＡＤＲは、１５ビット目から１８ビット目及び２０ビット目の５ビットに割り当てられている。アドレスＡＤＲは、ドライバＩＣを識別するためのアドレスを表す。すなわち、アドレスＡＤＲは、制御データの送信先とするドライバＩＣのアドレス端子に設定されたアドレスを表す。
【０１０８】

具体的には、ドライバＩＣ１２８ａ～１２８ｃのそれぞれにアドレス設定用の５つの端子（不図示）があり、５つの端子それぞれを＋５Ｖ電源側（ＰｕｌｌＵｐ）又は、グランド側（ＰｕｌｌＤｏｗｎ）に接続すると、ドライバＩＣを識別するためのアドレスが設定される。

【０１０９】

モードＭＤは、２１ビット目の１ビットに割り当てられている。モードＭＤは、制御データが第１モードであるか、第２モードであるかを表す。本実施形態においては、モードＭＤには、第１モードの場合には「０」が設定され、第２モードの場合には「１」が設定される。

【０１１０】

エラー検出用データＥは、１０ビット目、１９ビット目及び２２ビット目の各１ビットに割り当てられている。本実施形態において、エラー検出用データＥは、固定値とし、例えば、「０」とする。なお、エラー検出用データＥは、固定値以外に、所定の法則に基づく値（例えば、所定周期の循環値）でもよく、パリティビットでもよい。

【０１１１】

第１モードにおける制御データは、ペイロードとして、チャネル０からチャネル２３の各輝度データＬ（図中、Ｌの添え字は、チャネル番号を表す）と、エラー検出用データＥと、エンドビットＥＮＤとを含む。

【０１１２】

チャネル０からチャネル２３の各輝度データＬ０～Ｌ２３は、ペイロードの先頭から６ビット単位で割り当てられている。エラー検出用データＥは、ヘッダのエラー検出用データＥと同様であり、ペイロードの９ビット目から９ビット周期で挿入されている。エンドビットＥＮＤは、制御データの最後のビットを表す。本実施形態において、エンドビットＥＮＤは、固定値とし、例えば、「０」とする。

【０１１３】

このように、第１モードにおける制御データのペイロードは、エラー検出用データＥの周期を輝度データＬの周期に同期させないことにより、輝度データＬのデータパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができるように規定されている。

【０１１４】

第１モードの制御データを受信したドライバＩＣは、割り当てられたアドレスと制御データのアドレスＡＤＲとが等しく、予め登録されたデバイス識別子とデバイス識別子ＤＥＶとが等しければ、チャネル０からチャネル２３に対応するレジスタの値を輝度データＬ０～Ｌ２３に更新する。

【０１１５】

（第２モード）

図９に示すように、第２モードにおける制御データのヘッダは、第１モードにおける制御データのヘッダと同様であるため、説明を省略する。

【０１１６】

第２モードにおける制御データは、ペイロードとして、チャネル数ＣＨＮと、チャネル番号ＣＨ（図中、ＣＨの添え字は、ａから順次付されている）と、輝度データＬ（図中、Ｌの添え字は、ａから順次付されている）と、エラー検出用データＥと、エンドビットＥＮＤとを含む。

【０１１７】

チャネル数ＣＨＮは、ペイロードの先頭から５ビットに割り当てられている。チャネル数ＣＨＮは、更新対象のチャネルの数を表す。すなわち、チャネル数ＣＨＮは、以降に続く、チャネル番号ＣＨと輝度データＬとの組の数を表す。

【０１１８】

チャネル番号ＣＨは、５ビットで表され、チャネル０からチャネル２３のいずれかのチャネルの番号を表す。各輝度データＬａ～Ｌｇは、チャネル番号ＣＨに続く６ビットで表

されている。

【０１１９】

エラー検出用データＥは、第１モードの制御データにおけるエラー検出用データＥと同様であり、ペイロードの９ビット目から９ビット周期で挿入されている。エンドビットＥＮＤは、第１モードの制御データにおけるエンドビットＥＮＤと同様である。

【０１２０】

このように、第２モードにおける制御データのペイロードは、エラー検出用データＥの周期を輝度データＬの周期に同期させないことにより、輝度データＬのデータパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができるように規定されている。

【０１２１】

第２モードの制御データを受信したドライバＩＣは、割り当てられたアドレスと制御データのアドレスＡＤＲとが等しく、予め登録されたデバイス識別子とデバイス識別子ＤＥＶとが等しければ、各チャネル番号ＣＨに対応するレジスタの値を各輝度データＬに更新する。

【０１２２】

（第３モード）

図１０に示すように、第３モードにおける制御データは、先頭検出ビットＨＤと、リセットコマンドＲＳＴと、エラー検出用データＥと、エンドビットＥＮＤとを含む。

【０１２３】

先頭検出ビットＨＤは、第１モードの制御データのヘッダにおける先頭検出ビットＨＤと同様であり、１ビット目から９ビット目の９ビットに割り当てられている。リセットコマンドＲＳＴは、１１ビット目から１８ビット目の８ビットに割り当てられている。本実施形態において、リセットコマンドＲＳＴは、図示したように、固定値である。

【０１２４】

エラー検出用データＥは、第１モードの制御データのヘッダにおける先頭検出ビットＨＤと同様であり、１０ビット目の１ビットに割り当てられている。エンドビットＥＮＤは、第１モードの制御データにおけるエンドビットＥＮＤと同様である。第３モードの制御データを受信したドライバＩＣは、全チャネルに対応するレジスタの値を０に更新する。

【０１２５】

このように、チャネルを個別に制御する場合には、第２モードで制御データを送信することによって、第１モードと比較して制御データのデータ長を短くすることができる。一方、チャネルを全体的に制御する場合には、チャネル番号を必要としない第１モードで制御データを送信することによって、第２モードと比較して制御データのデータ長を短くすることができる。

【０１２６】

また、全てのチャネルのＬＥＤを消灯させる場合には、第３モードで制御データを送信することによって、第１モード及び第２モードと比較して制御データのデータ長を短くすることができる。

【０１２７】

以上のように、ドライバＩＣが駆動する発光体の用途に応じて、制御データのデータ長が短くなるモードを選択することにより、サブＣＰＵ８１とドライバＩＣとの間の伝送負荷を低減させることができ、ＬＥＤの応答性を向上させることができる。

【０１２８】

本実施形態において、ドットマトリクス部１１９を構成するＬＥＤは、第１発光体群を構成し、各第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８を構成するＬＥＤは、第２発光体群を構成する。

【０１２９】

サブＣＰＵ８１は、第１発光体群のＬＥＤを駆動するドライバＩＣには、第２ポートから制御データを第２モードで送信する。サブＣＰＵ８１は、第２発光体群のＬＥＤを駆動

するドライバＩＣには、第１ポートから制御データを第１モードで送信する。サブＣＰＵ８１は、電源投入時や演出の切り替え時などのようにＬＥＤを全消灯させる場合、該当ポートから制御データを第３モードで送信する。

【０１３０】

なお、本実施形態に係る第２ランプ群１１２、第３ランプ群１１３、第４ランプ群１１４、第５ランプ群１１５、第６ランプ群１１６、及び、その他のポートに係る光源部は、それぞれ不図示のドライバＩＣを介してサブＣＰＵ８１によって制御される。

【０１３１】

＜メインＲＯＭに記憶されているデータテーブルの構成＞

次に、メインＲＯＭ５２に記憶されている各種データテーブルの構成について説明する。

【０１３２】

［図柄配置表］

まず、図１１を参照して、図柄配置表について説明する。図柄配置表は、左リール３Ｌ、中リール３Ｃ及び右リール３Ｒのそれぞれの回転方向における各図柄の位置と、各位置に配された図柄の種類を特定するデータとの対応関係を規定する。

【０１３３】

図柄配置表では、リールインデックスが検出されたときに、表示窓４Ｌ，４Ｃ，４Ｒの枠内における中段領域に配置される各リールの図柄の位置を「０」と規定する。そして、各リールにおいて、図柄位置「０」を基準としてリールの回転方向（図１１における下方向）に進む順に、図柄カウンタに対応する「０」～「２０」が、図柄位置として、各図柄に割り当てられる。

【０１３４】

すなわち、図柄カウンタの値（「０」～「２０」）と、図柄配置表とを参照することにより、表示窓４Ｌ，４Ｃ，４Ｒの枠内における各リールの上段、中段及び下段の領域に表示されている図柄の種類を特定することができる。例えば、左リール３Ｌに対応する図柄カウンタの値が「７」であるとき、表示窓４の枠内における左リール３Ｌの上段、中段及び下段の領域には、それぞれ、図柄位置「８」の「ドン１」、図柄位置「７」の「ベル２」及び図柄位置「６」の「リプレイ」に対応する図柄が表示されている。

【０１３５】

＜サブＲＯＭに記憶されているデータテーブル＞

［月別日テーブル］

図１２に示すように、月別日テーブルは、閏年であるときの日数と閏年でないときの日数とが月ごとに対応付けられている。月別日テーブルは、後述する日時更新処理（図２６参照）を実行するサブＣＰＵ８１によって参照される。このように、月別日テーブルを記憶するサブＲＯＭ８２は、月別日テーブル記憶手段を構成する。

【０１３６】

＜サブＲＡＭに割り当てられる格納領域＞

［日時格納領域］

図１３に示すように、日時格納領域は、「年」、「月」、「日」、「曜日」、「時」、「分」及び「秒」をそれぞれ格納するための各領域がサブＲＡＭ８３に割り当てられる。すなわち、日時格納領域には、日時データが記憶される。

【０１３７】

以下の説明において、日時格納領域を構成する各領域の値を日時格納領域（年）、日時格納領域（月）、日時格納領域（日）、日時格納領域（曜日）、日時格納領域（時）、日時格納領域（分）、日時格納領域（秒）という。日時格納領域は、後述する日時更新処理（図２６参照）を実行するサブＣＰＵ８１によって使用される。

【０１３８】

＜ランプデータ＞

本実施形態において、サブＲＯＭ８２には、光の点（増減含む）消灯のパターンに関する演出データであるランプデータ（「発光データ」とも称す）と、複数のパーツデータと

が格納されている。このように、サブＲＯＭ８２は、発光データ記憶手段及びパターンデータ記憶手段を構成する。

【０１３９】

各パーツデータは、パーツデータを識別するための識別情報と、パーツデータの属性を示す属性データと、光の点（増減含む）消灯のパターンを表すパターンデータとを含む。各ランプデータは、パーツデータの識別情報の順序を表す。

【０１４０】

パターンデータは、再生順に順序付けられた複数の輝度パターンよりなる。各輝度パターンは、制御対象とする各ドライバＩＣの各チャネルに設定する輝度を表す。各輝度パターンにおいて、ドライバＩＣのチャネルに設定する輝度は、パターンデータの汎用性が考慮され、８ビットで表されている。

【０１４１】

なお、ドライバＩＣのチャネルに設定する輝度は、パターンデータの汎用性が担保されれば、１６ビット、３２ビット又は６４ビットなどで表されていてもよい。パターンデータにおける最後の輝度パターンは、エンドブロックを表す。

【０１４２】

パーツデータの属性データとしては、ショットと、ショット＋チェインと、ループとがある。属性データがショットのパーツデータは、１のランプデータにおいて、一度しか再生されない。

【０１４３】

ショット＋チェインは、連続再生を表す。属性データがショット＋チェインのパーツデータは、１のランプデータにおいて、連続再生される。したがって、１のランプデータにおいて、属性データがショット＋チェインのパーツデータが連続する場合には、属性データがショット＋チェインの一連のパーツデータが繰り返し再生される。

【０１４４】

すなわち、ランプデータにおける最後の識別情報に対応するパーツデータが再生された後、ランプデータにおける最後の識別情報に対応するパーツデータに含まれる属性データがショット＋チェインを表す場合には、ランプデータが表す識別情報の先頭から属性データがショット＋チェインを表すパーツデータが検索され、検出されたパーツデータから連続再生が開始される。

【０１４５】

ループは、繰り返し再生を表す。属性データがループのパーツデータは、１のランプデータにおいて、繰り返し再生される。すなわち、属性データがループのパーツデータの識別情報がランプデータに含まれている場合、このパーツデータが繰り返し再生される。

【０１４６】

本実施形態におけるランプデータの具体例を説明する。実行する演出に応じたランプデータは、サブＣＰＵ８１によって選択され、図１４に示すサブＲＡＭ８３の再生状態管理格納領域に読み込まれる。

【０１４７】

図１４において、再生状態管理格納領域には、再生中のパーツデータを識別するためのパーツ番号を格納するエントリパーツ番号格納領域と、パーツデータを識別するための識別情報であるパーツ番号を格納するパーツ番号格納領域と、ランプデータの最後を表すエンドコードを格納するエンドコード格納領域と、ランプデータが表す識別情報のパーツデータに、属性データがショット＋チェイン又はループであるパーツデータが含まれているか否かを表すチェイン／ループ設定領域とが含まれる。なお、再生状態管理格納領域に含まれるパーツ番号格納領域の数は、ランプデータが表すパーツデータの数と等しくなる。

【０１４８】

図１５は、再生状態管理格納領域の具体例を示している。図１５に示す例において、エントリパーツ番号格納領域には、エントリパーツ番号として「１」（イントロが割り当てられた番号）が格納され、パーツ番号格納領域には、パーツ番号として「１」～「８」が

それぞれ格納され、エンドコード格納領域には、エンドコードが格納され、チェイン／ループ設定領域には、「ＴＲＵＥ」が格納されている。

【０１４９】

なお、ランプデータが表す識別情報のパーツデータに、属性データがショット＋チェイン又はループであるパーツデータが含まれていない場合には、チェイン／ループ設定領域には、「ＦＡＬＳＥ」が格納されている。また、チェイン／ループ設定領域に、「ＦＡＬＳＥ」が格納されている場合、属性データを検索することなく、パーツデータの繰り返し再生は行われない。（１回限りの再生）

【０１５０】

図１６は、図１５に示した再生状態管理格納領域に基づくランプデータの再生例を示す。パーツ番号が「１」のパーツデータは、属性データにショットが設定され、パターンデータに、例えば３０秒分のイントロ演出を表す輝度パターンが設定されている。

【０１５１】

パーツ番号が「２」～「８」の各パーツデータは、属性データにショット＋チェインが設定され、パターンデータに、例えば３０秒分の連続演出を表す輝度パターンが設定されている。なお、図示の例では、パーツ番号が「１」～「８」の全てのパーツデータの再生時間が３０秒となっているが、パーツデータの長さは、互いに異なっていてもよい。

【０１５２】

図中、矢印で示すように、ランプデータの再生が開始されると、パーツ番号が「１」のパーツデータが３０秒にわたって再生され、次いで、パーツ番号が「２」～「８」の各パーツデータが３０秒にわたって再生される。その際、パーツ番号が「１」から「２」への更新は、再生状態管理格納領域のエントリパーツ番号格納領域の値が、「１」から「２」へとの更新されることで管理される。

【０１５３】

エントリパーツ番号格納領域の値を基にエンドコードが検出されると、最後に再生されたパーツ番号が「８」のパーツデータの属性データがショット＋チェインであるため、ランプデータの先頭であるパーツ番号が「１」のパーツデータから、属性データがショット＋チェインであるパーツデータが検索される。

【０１５４】

図示の例では、パーツ番号が「２」のパーツデータが検出される。したがって、エントリパーツ番号格納領域に「２」がセットされ、図中、矢印で示すように、パーツ番号が「２」のパーツデータからパーツ番号が「２」～「８」の一連のパーツデータが連続して再生される。以降、ランプデータが更新されない限り、パーツ番号が「２」～「８」の一連のパーツデータが繰り返し再生される。

【０１５５】

ランプデータの再生中にパチスロ１のエラーが検出された場合、サブＣＰＵ８１は、エラー用のランプデータを再生する。したがって、エラー検出時に再生していたランプデータの再生は、中断する。エラー復帰時には、サブＣＰＵ８１は、エラー検出時に再生していたランプデータを再生する。

【０１５６】

このときに、中断したランプデータを最初から再生すると、演出内容が遊技状態にそぐわなくなってしまうことがある。このため、サブＲＯＭ８２には、ランプデータに対応して、中断復帰用ランプデータが格納されている。中断復帰用ランプデータは、通常のランプデータに対して、属性データがショット＋チェインを表していないパーツデータのパーツ番号が除かれている。

【０１５７】

図１７は、中断復帰用ランプデータが格納された再生状態管理格納領域を示している。図１７に示す再生状態管理格納領域には、再生状態管理格納領域に「２」が格納され、図１５に示した再生状態管理格納領域に格納されたランプデータに対して属性データがショット＋チェインを表していないパーツデータのパーツ番号である「１」（イントロが割り

当てられた番号）が除かれている。

【０１５８】

したがって、図１８に示すように、中断復帰用ランプデータは、属性データがショット＋チェインを表す先頭のパーツ番号が「２」のパーツデータから再生が開始される。このように、サブＣＰＵ８１は、エラー復帰時に中断復帰用ランプデータの再生を開始することによって、演出内容が遊技状態にそぐわなくなってしまうこと（演出の途中にも関わらずイントロが再生すること）を防止している。

【０１５９】

なお、サブＣＰＵ８１は、後述の属性検索頭出処理（図３０のステップＳ６１１）で再生状態管理格納領域を使用して、繰り返し再生を行うための頭出しを行っている。図１５から図１８は、属性検索頭出処理の具体例でもある。

【０１６０】

なお、サブＲＯＭ８２にランプデータに対応する中断復帰用ランプデータを格納するのに代えて、サブＣＰＵ８１は、エラー復帰時に、エラー検出時に再生していたランプデータを中断復帰用ランプデータに変換するようにしてもよい。

【０１６１】

具体的には、サブＣＰＵ８１は、再生状態管理格納領域に格納されたランプデータに対して属性データがショット＋チェインを表していないパーツデータのパーツ番号を除くことにより、中断復帰用ランプデータに変換するようにしてもよい。

【０１６２】

＜主制御回路の動作説明＞

次に、図１９を参照して、主制御回路４１のメインＣＰＵ５１が、プログラムを用いて実行する各種処理の内容について説明する。

【０１６３】

［電源投入処理］

まず、メインＣＰＵ５１の制御で行うパチスロ１の主制御回路４１の電源投入処理を、図１９を参照しながら説明する。

【０１６４】

まず、パチスロ１に電源が投入されると、メインＣＰＵ５１は、電源投入時の初期化処理を行う（Ｓ１）。この初期化処理では、バックアップが正常に行われたか、設定変更が適切に行われたか等が判定され、その判定結果に対応した初期化が行われる。

【０１６５】

次いで、メインＣＰＵ５１は、一遊技終了時の初期化処理を行う（Ｓ２）。この初期化処理では、メインＲＡＭ５３における指定格納領域のデータをクリアする。なお、ここでいう指定格納領域は、例えば、内部当籤役格納領域や表示役格納領域などの１回の遊技ごとにデータの消去が必要な格納領域である。

【０１６６】

次いで、メインＣＰＵ５１は、メダル受付・スタートチェック処理を行う（Ｓ３）。この処理では、メダルセンサ３５Ｓやスタートスイッチ１６Ｓの入力のチェック等が行われる。

【０１６７】

具体的には、メインＣＰＵ５１は、前回の遊技で再遊技（リプレイ役）に係る表示役が成立したときの自動投入による投入枚数のカウント、メダル投入口１３から投入されたメダルによる投入枚数のカウント、及び、ＢＥＴボタン（ＭＡＸＢＥＴ、１ＢＥＴ）の押下に基づく投入枚数のカウント等が実行され、投入枚数が遊技開始可能枚数に達している場合には、スタートスイッチの状態を判別する。

【０１６８】

次いで、メインＣＰＵ５１は、乱数発生器５６の第１乱数レジスタ（不図示）から乱数値（０～６５５３５）を抽出し、該抽出した乱数値をメインＲＡＭ５３に設けられた乱数値格納領域（不図示）に格納する（Ｓ４）。次いで、メインＣＰＵ５１は、乱数発生器５

６の第２乱数レジスタ（不図示）からリール演出及びロックの制御で用いる演出用乱数値を抽出し、該抽出した演出用乱数値をメインＲＡＭ５３に設けられた演出用乱数値格納領域（不図示）に格納する（Ｓ５）。なお、本実施形態では、演出用乱数値は、０～１２７の範囲から抽出される。

【０１６９】

そして、抽出した各種乱数値が所定の乱数値格納領域に格納されると、メインＣＰＵ５１は、内部抽籤処理を行う（Ｓ６）。この処理では、Ｓ４で抽出した乱数値に基づいた抽籤により内部当籤役の決定が行われる。

【０１７０】

次いで、メインＣＰＵ５１は、遊技ロック抽籤処理を行う（Ｓ７）。遊技ロック抽籤処理において、メインＣＰＵ５１は、遊技状態に応じた遊技ロック抽籤テーブル（不図示）を参照し、演出用乱数値に基づいて、遊技ロック抽籤を行う。

【０１７１】

次いで、メインＣＰＵ５１は、リール停止初期設定処理を行う（Ｓ８）。リール停止初期設定処理において、メインＣＰＵ５１は、内部当籤役に基づいて、リール停止初期設定テーブル（不図示）からルールの停止制御に係る各種情報を取得する。

【０１７２】

次いで、メインＣＰＵ５１は、スタートコマンド送信処理を行う（Ｓ９）。具体的には、メインＣＰＵ５１は、スタートコマンドを副制御回路４２に送信する。なお、スタートコマンドは、内部当籤役等を特定するパラメータ、遊技ロックの種別及びロック時間等を含んで構成される。

【０１７３】

次いで、メインＣＰＵ５１は、ウェイト処理を行う（Ｓ１０）。この処理では、メインＣＰＵ５１は、前回の遊技開始から所定時間（例えば、４．１秒）を経過していない場合、該所定時間が経過するまで待ち時間を消化する。

【０１７４】

次いで、メインＣＰＵ５１は、リール回転開始処理を行う（Ｓ１１）。この処理において、メインＣＰＵ５１は、全リールの回転開始を要求する。リール回転開始処理において、メインＣＰＵ５１は、全リールの回転開始を要求する。全リールの回転開始が要求されると、一定の周期（１．１１７２ｍｓｅｃ）で実行される後述の割込処理（図２０参照）により、３つのステッピングモータ６１Ｌ，６１Ｃ，６１Ｒの駆動が制御され、左リール３Ｌ、中リール３Ｃ及び右リール３Ｒの回転が開始される。

【０１７５】

次いで、メインＣＰＵ５１は、引込優先順位格納処理を行う（Ｓ１２）。この処理では、メインＣＰＵ５１は、引込優先順位データを取得して、引込優先順位データ格納領域（不図示）に格納する。

【０１７６】

次いで、メインＣＰＵ５１は、リール停止制御処理を行う（Ｓ１３）。この処理では、左ストップボタン１７Ｌ、中ストップボタン１７Ｃ及び右ストップボタン１７Ｒがそれぞれ押されたタイミングと内部当籤役とに基づいて該当するリールの回転が停止される。すなわち、本実施形態では、リール停止制御処理を実行する主制御回路４１は停止制御手段を構成する。

【０１７７】

次いで、メインＣＰＵ５１は、遊技ロック処理を行う（Ｓ１４）。この処理において、メインＣＰＵ５１は、遊技の進行を無効化する又は遅延させる。メインＣＰＵ５１は、決定された遊技ロック種別を実行する期間に相当する値をロックタイマにセットし、ロックタイマの値が「０」になるまで待機する。その間、メインＣＰＵ５１は、遊技者のあらゆる操作を受け付けない。

【０１７８】

次いで、メインＣＰＵ５１は、入賞検索処理を行う（Ｓ１５）。この処理では、メイン

ＣＰＵ５１は、図柄コード格納領域（不図示）のデータを表示役格納領域（不図示）に格納する。また、この処理では、左リール３Ｌ、中リール３Ｃ及び右リール３Ｒが全て停止した後に有効ライン（入賞判定ライン）に表示された図柄の組合せと、図柄組合せテーブル（不図示）とを照合する。そして、メインＣＰＵ５１は、有効ラインに表示役が表示されたか否かを判定し、その判定結果を表示役格納領域に格納するようにしてもよい。

【０１７９】

次いで、メインＣＰＵ５１は、メダル払出処理を行う（Ｓ１６）。メダル払出処理は、本発明に係る遊技媒体付与手段の一具体例を示す。この処理では、Ｓ１５において決定された表示役の払出枚数に基づいて、ホッパー３３の駆動やクレジット枚数の更新が行われ、メダルの払い出しが行われる。この際、本実施形態では、図柄組合せテーブル（不図示）に示すように、メダルの投入枚数が１～３枚であり、メダルの払出枚数は表示役に応じて異なるが、その最大払出枚数（払出上限）は１５枚である。

【０１８０】

次いで、メインＣＰＵ５１は、ＲＴ制御処理を行う（Ｓ１７）。この処理では、メインＣＰＵ５１は、ＲＴ遊技状態を管理する。メインＣＰＵ５１は、ＲＴ遷移テーブル（不図示）を参照し、移行元（現在）のＲＴ遊技状態において成立し得るＲＴ遊技状態の移行条件をチェックし、ＲＴ遊技状態の移行条件が成立していると判別したとき、ＲＴ遷移テーブル（不図示）を参照し、移行条件に基づいて、移行先のＲＴ遊技状態に移行する。

【０１８１】

次いで、メインＣＰＵ５１は、払出終了コマンド送信処理を行う（Ｓ１８）。具体的には、メインＣＰＵ５１は、払出終了コマンドを副制御回路４２に送信する。

【０１８２】

次いで、メインＣＰＵ５１は、ボーナス終了チェック処理を行う（Ｓ１９）。この処理では、メインＣＰＵ５１は、ボーナスゲームの終了契機を管理するための各種カウンタを参照して、ボーナスゲームの作動を終了するか否かをチェックする。

【０１８３】

次いで、メインＣＰＵ５１は、ボーナス作動チェック処理を行う（Ｓ２０）。この処理では、メインＣＰＵ５１は、ボーナスゲームの作動を開始するか否か、及び、再遊技を行うか否かをチェックする。ボーナス作動チェック処理が終了すると、メインＣＰＵ５１は、処理をＳ２に戻し、Ｓ２以降の処理を繰り返す。

【０１８４】

［メインＣＰＵの制御による割込処理（１．１１７２ｍｓｅｃ）］

次に、図２０を参照して、メインＣＰＵ５１に内蔵されたタイマ（不図示）の制御による定周期（１．１１７２ｍｓｅｃ毎）に行われる割込処理について説明する。

【０１８５】

まず、メインＣＰＵ５１は、レジスタの退避を行う（Ｓ３５１）。次いで、メインＣＰＵ５１は、入力ポートチェック処理を行う（Ｓ３５２）。この処理では、ストップスイッチ１７Ｓ等の各種スイッチから入力される信号がチェックされる。

【０１８６】

次いで、メインＣＰＵ５１は、タイマ更新処理を行う（Ｓ３５３）。この処理では、メインＣＰＵ５１は、例えば、割込処理毎にロックタイマの値を減算する処理を行う。次いで、メインＣＰＵ５１は、通信データ送信処置を行う（Ｓ３５４）。この処理では、主に、各種コマンドを主制御回路４１及び副制御回路４２に適宜送信する。

【０１８７】

次いで、メインＣＰＵ５１は、リール制御処理を行う（Ｓ３５５）。この処理では、メインＣＰＵ５１は、全リールの回転開始が要求されたときに、左リール３Ｌ、中リール３Ｃ及び右リール３Ｒの回転を開始し、その後、各リールが一定速度で回転するように、３つのステッピングモータ６１Ｌ，６１Ｃ，６１Ｒを駆動制御する。また、滑り駒数が決定されたときは、メインＣＰＵ５１は、該当するリールの図柄カウンタを滑り駒数分だけ更新する。そして、メインＣＰＵ５１は、更新された図柄カウンタが停止予定位置に対応す

る値に一致する（停止予定位置の図柄が表示窓の有効ライン（入賞判定ライン）上の領域に到達する）のを待って、該当するリールの回転の減速及び停止が行われるように、対応するステッピングモータを駆動制御する。また、本実施形態では、Ｓ３５５の処理において、前述した通常の加速処理、定速処理及び停止処理だけでなく、加速処理時にリール演出パターンが設定されている場合には、該リール演出パターンに対応するリール演出（リールアクション）及びロックの制御処理も行う。

【０１８８】

次いで、メインＣＰＵ５１は、ランプ・７セグ駆動処理を行う（Ｓ３５６）。この処理では、メインＣＰＵ５１は、７セグ表示器６を駆動制御して、払出枚数やクレジット枚数などを表示する。次いで、メインＣＰＵ５１は、レジスタの復帰処理を行う（Ｓ３５７）。そして、その後、メインＣＰＵ５１は、割込処理を終了する。

【０１８９】

＜副制御回路の動作説明＞

次に、図２１～図３２を参照して、副制御回路４２のサブＣＰＵ８１が、プログラムを用いて実行する各種処理（タスク）の内容について説明する。

【０１９０】

［電源投入処理］

図２１は、電源投入時に実行される副制御回路４２の電源投入処理を示すフローチャートである。

【０１９１】

まず、サブＣＰＵ８１は、初期化処理を実行する（Ｓ３６１）。初期化処理において、サブＣＰＵ８１は、サブＲＡＭ８３等のエラーチェックに加えて、各種ドライバの初期化を行う。

【０１９２】

Ｓ３６１で初期化されるタスクは、コマンド受信割込同期のタスクグループである主基板通信タスク（図２２参照）、並びに、タイマ割込同期のタスクグループであるサウンド制御タスク（図２７参照）及びランプ制御タスク（図３０参照）等を含む。

【０１９３】

次に、サブＣＰＵ８１は、ランプ制御タスクを起動する（Ｓ３６２）。ランプ制御タスクでは、サブＣＰＵ８１は、２ｍｓ毎に送信されるタイマ割込イベントメッセージが受信されるのを待ち、タイマ割込イベントメッセージを受信したことに応じて、ＬＥＤ群２１などの各種ランプの点灯状態を制御する。このように、ランプ制御タスクを実行するサブＣＰＵ８１は、制御手段を構成する。

【０１９４】

次に、サブＣＰＵ８１は、サウンド制御タスクを起動する（Ｓ３６３）。サウンド制御タスクでは、サブＣＰＵ８１は、ランプ制御タスクと同様に、スピーカ２０Ｌ、２０Ｒなどの各種スピーカの出音状態を制御する。

【０１９５】

次に、サブＣＰＵ８１は、主基板通信タスクを起動する（Ｓ３６４）。主基板通信タスクでは、サブＣＰＵ８１は、主制御回路４１から送信されるコマンドに対する受信及び解析と解析したコマンドに基づいて演出を決定（抽籤）する。

【０１９６】

次に、サブＣＰＵ８１は、ＲＴＣ８６から日時データを取得し、サブＲＡＭ８３に割り当てられた日時格納領域（図１３参照）に格納する（Ｓ３６５）。このように、サブＣＰＵ８１は、日時取得手段を構成する。

【０１９７】

次に、サブＣＰＵ８１は、サブＣＰＵ８１に備えられたクロックカウンタ（不図示）からカウント値を取得しクロックカウントＰとしてサブＲＡＭ８３に格納する（Ｓ３６６）。Ｓ３６６の処理を実行した後、サブＣＰＵ８１は、電源投入処理を終了する。なお、クロックカウンタとは、サブＣＰＵ８１に内蔵され発振回路８４（図７参照）が発振する所

定周波数のクロックをカウントするカウンタ回路であり、発振回路８４のクロックをカウントするクロックカウンタを備えるサブＣＰＵ８１は、カウント手段を構成する。

【０１９８】

［主基板通信タスク］

次に、図２２を参照して、サブＣＰＵ８１により行われる主基板通信タスクについて説明する。

【０１９９】

まず、サブＣＰＵ８１は、主制御回路４１から送信されたコマンドの受信チェックを行う（Ｓ５０１）。次いで、サブＣＰＵ８１は、受信チェックの結果に基づいて受信したコマンドが有効であるか否かを判別する（Ｓ５０２）。

【０２００】

例えば、サブＣＰＵ８１は、受信したコマンドのデータ長が８Ｂｙｔｅである第１条件と、受信したコマンドがスタートコマンド、リール停止コマンド、表示コマンド、払出終了コマンド、ボーナス開始コマンド、ボーナス終了コマンド及び無操作コマンドなど予め登録されたコマンドである第２条件と、受信したコマンドの８Ｂｙｔｅ目に格納されたサム値が正しい第３条件との３つの条件が成立した場合には、受信チェックのチェック結果に受信したコマンドが有効であると反映され、いずれかの条件が成立しなかった場合には、受信チェックのチェック結果に受信したコマンドが有効でないと反映される。

【０２０１】

Ｓ５０２において、サブＣＰＵ８１が、受信したコマンドが有効でないと判別したとき（Ｓ５０２がＮＯ判定の場合）、サブＣＰＵ８１は、処理をＳ５０１に戻し、Ｓ５０１以降の処理を繰り返す。

【０２０２】

一方、Ｓ５０２において、サブＣＰＵ８１が、受信したコマンドが有効であると判別したとき（Ｓ５０２がＹＥＳ判定の場合）、サブＣＰＵ８１は、受信したコマンドに基づいて、メッセージキューにメッセージを格納する（Ｓ５０３）。なお、メッセージキューとは、プロセス間で情報を交換するための機構である。そして、Ｓ５０３の処理後、サブＣＰＵ８１は、処理をＳ５０１に戻し、Ｓ５０１以降の処理を繰り返す。

【０２０３】

［演出登録タスク］

次に、図２３を参照して、サブＣＰＵ８１により行われる演出登録タスクについて説明する。

【０２０４】

まず、サブＣＰＵ８１は、メッセージキューからメッセージを取り出す（Ｓ５１１）。次いで、サブＣＰＵ８１は、メッセージキューにメッセージが有るか否かを判別する（Ｓ５１２）。Ｓ５１２において、サブＣＰＵ８１が、メッセージキューにメッセージが無いと判別したとき（Ｓ５１２がＮＯ判定のとき）、サブＣＰＵ８１は、後述のＳ５１５の処理を行う。

【０２０５】

一方、Ｓ５１２において、サブＣＰＵ８１が、メッセージキューにメッセージが有ると判別したとき（Ｓ５１２がＹＥＳ判定のとき）、サブＣＰＵ８１は、メッセージから遊技情報を複写する（Ｓ５１３）。この処理では、例えば、パラメータによって特定される、内部当籤役、回転が停止したリールの種別、表示役、遊技状態フラグ等の各種データがサブＲＡＭ８３に設けられた格納領域（不図示）に複写される。

【０２０６】

次いで、サブＣＰＵ８１は、演出内容決定処理を行う（Ｓ５１４）。この処理では、サブＣＰＵ８１は、受信したコマンドの種別に応じて、演出内容の決定や演出データの登録等を行う。なお、演出内容決定処理の詳細については、後述の図２４を参照しながら後で説明する。

【０２０７】

次いで、サブＣＰＵ８１は、サウンドデータの登録を行う（Ｓ５１５）。次いで、サブＣＰＵ８１は、ランプデータの登録を行う（Ｓ５１６）。なお、これらの登録処理は、Ｓ５１４の演出内容決定処理において登録された演出データに基づいて行われる。また、本実施形態において、サブＣＰＵ８１は、サウンドデータのリクエスト番号が決定されていた場合に限りサウンドデータのリクエスト番号をサブＲＡＭ８３の登録領域に記憶することでサウンドデータを登録し、ランプデータのリクエスト番号が決定されていた場合に限りランプデータのリクエスト番号をサブＲＡＭ８３の登録領域に記憶することでランプデータを登録する。Ｓ５１６の後、サブＣＰＵ８１は、処理をＳ５１１に戻し、Ｓ５１１以降の処理を繰り返す。

【０２０８】

［演出内容決定処理］

次に、図２４を参照して、演出登録タスクのフローチャート（図２３参照）中のＳ５１４で行う演出内容決定処理について説明する。

【０２０９】

まず、サブＣＰＵ８１は、スタートコマンド受信時であるか否かを判別する（Ｓ５２１）。

【０２１０】

Ｓ５２１において、サブＣＰＵ８１が、スタートコマンド受信時であると判別したとき（Ｓ５２１がＹＥＳ判定の場合）、サブＣＰＵ８１は、スタートコマンド受信時処理を行う（Ｓ５２２）。この処理では、サブＣＰＵ８１は、演出用乱数値を抽出し、内部当籤役等に基づいて演出番号を抽籤により決定して登録する。ここで、演出番号は、今回実行する演出内容を指定するデータである。

【０２１１】

次いで、サブＣＰＵ８１は、登録されている演出番号に応じて、スタート時の演出データを登録する（Ｓ５２３）。演出データは、アニメーションデータ、サウンドデータ及びランプデータを指定するデータである。それゆえ、演出データが登録されると、対応するアニメーションデータ等が決定され、表示装置による表示等の演出が実行される。そして、Ｓ５２３の処理後、サブＣＰＵ８１は、演出内容決定処理を終了し、処理を演出登録タスク（図２３参照）のＳ５１５に移す。

【０２１２】

一方、Ｓ５２１において、サブＣＰＵ８１が、スタートコマンド受信時でないと判別したとき（Ｓ５２１がＮＯ判定の場合）、サブＣＰＵ８１は、リール停止コマンド受信時であるか否かを判別する（Ｓ５２４）。

【０２１３】

Ｓ５２４において、サブＣＰＵ８１が、リール停止コマンド受信時であると判別したとき（Ｓ５２４がＹＥＳ判定の場合）、サブＣＰＵ８１は、登録されている演出番号及び作動ストップボタンの種別に応じて、停止時の演出データを選択する（Ｓ５２５）。その後、サブＣＰＵ８１は、演出内容決定処理を終了し、処理を演出登録タスク（図２３参照）のＳ５１５に移す。

【０２１４】

一方、Ｓ５２４において、サブＣＰＵ８１が、リール停止コマンド受信時でないと判別したとき（Ｓ５２４がＮＯ判定の場合）、サブＣＰＵ８１は、表示コマンド受信時であるか否かを判別する（Ｓ５２６）。

【０２１５】

Ｓ５２６において、サブＣＰＵ８１が、表示コマンド受信時であると判別したとき（Ｓ５２６がＹＥＳ判定の場合）、サブＣＰＵ８１は、表示コマンド受信時処理を行う（Ｓ５２７）。この処理では、サブＣＰＵ８１は、表示役等に基づいて演出番号を抽籤により決定して登録する。ここで、演出番号は、今回実行する演出内容を指定するデータである。その後、サブＣＰＵ８１は、演出内容決定処理を終了し、処理を演出登録タスク（図２３参照）のＳ５１５に移す。

【０２１６】
一方、Ｓ５２６において、表示コマンド受信時でないと判別したとき（Ｓ５２６がＮＯ判定の場合）、サブＣＰＵ８１は、払出終了コマンド受信時であるか否かを判別する（Ｓ５２８）。Ｓ５２８において、サブＣＰＵ８１は、払出終了コマンド受信時であると判別したとき（Ｓ５２８がＹＥＳ判定の場合）、サブＣＰＵ８１は、払出終了コマンド受信時処理を行う（Ｓ５２９）。その後、サブＣＰＵ８１は、演出内容決定処理を終了し、処理を演出登録タスク（図２３参照）のＳ５１５に移す。
【０２１７】
一方、Ｓ５２８において、サブＣＰＵ８１が、払出終了コマンド受信時でないと判別したとき（Ｓ５２８がＮＯ判定の場合）、サブＣＰＵ８１は、ボーナス開始コマンド受信時であるか否かを判別する（Ｓ５３０）。
【０２１８】
Ｓ５３０において、サブＣＰＵ８１が、ボーナス開始コマンド受信時であると判別したとき（Ｓ５３０がＹＥＳ判定の場合）、サブＣＰＵ８１は、ボーナス開始用の演出データを登録する（Ｓ５３１）。その後、サブＣＰＵ８１は、演出内容決定処理を終了し、処理を演出登録タスク（図２３参照）のＳ５１５に移す。
【０２１９】
一方、Ｓ５３０において、サブＣＰＵ８１が、ボーナス開始コマンド受信時でないと判別したとき（Ｓ５３０がＮＯ判定の場合）、サブＣＰＵ８１は、ボーナス終了コマンド受信時であるか否かを判別する（Ｓ５３２）。Ｓ５３２において、サブＣＰＵ８１が、ボーナス終了コマンド受信時であると判別したとき（Ｓ５３２がＹＥＳ判定の場合）、サブＣＰＵ８１は、ボーナス終了用の演出データを登録する（Ｓ５３３）。その後、サブＣＰＵ８１は、演出内容決定処理を終了し、処理を演出登録タスク（図２３参照）のＳ５１５に移す。
【０２２０】
Ｓ５３２において、サブＣＰＵ８１が、ボーナス終了コマンド受信時ではないと判別したとき（Ｓ５３２がＮＯ判定の場合）、サブＣＰＵ８１は、無操作コマンド受信時であるか否かを判別する（Ｓ５３４）。Ｓ５３４において、サブＣＰＵ８１が、無操作コマンド受信時ではないと判別したとき（Ｓ５３４がＮＯ判定の場合）、サブＣＰＵ８１は、演出内容決定処理を終了し、処理を演出登録タスク（図２３参照）のＳ５１５に移す。
【０２２１】
一方、Ｓ５３４において、サブＣＰＵ８１が、無操作コマンド受信時であると判別したとき（Ｓ５３４がＹＥＳ判定の場合）、サブＣＰＵ８１は、無操作コマンド受信時処理を行う（Ｓ５３５）。なお、無操作コマンド受信時処理の詳細については、後述の図２５を参照しながら後で説明する。Ｓ５３５の処理後、サブＣＰＵ８１は、演出内容決定処理を終了し、処理を演出登録タスク（図２３参照）のＳ５１５に移す。
【０２２２】
［無操作コマンド受信時処理］
次に、図２５を参照して、演出内容決定処理のフローチャート（図２４参照）中のＳ５３５で行う無操作コマンド受信時処理について説明する。
【０２２３】
無操作コマンド受信時処理において、サブＣＰＵ８１は、図２６に示す日時更新処理を実行する（Ｓ５４０）。Ｓ５４０の処理を実行した後、サブＣＰＵ８１は、無操作コマンド受信時処理を終了する。
【０２２４】
無操作コマンドは、約１０ｍｓｅｃ周期で、主制御回路４１から送信される。図２６に示す日時更新処理は、無操作コマンド受信時処理に限らず、周期的に実行される他の処理やタスクから実行されるようにしてもよい。
【０２２５】
［日時更新処理］

次に、図２６を参照して、日時更新処理について説明する。

【０２２６】

まず、サブＣＰＵ８１は、クロックカウンタからカウント値を取得しクロックカウントＣとしてサブＲＡＭ８３に格納する（Ｓ５４１）。次に、サブＣＰＵ８１は、クロックカウントＣからクロックカウントＰを減じたクロックカウントＤを算出する（Ｓ５４２）。このように、サブＣＰＵ８１は、経過時間算出手段を構成する。

【０２２７】

次に、サブＣＰＵ８１は、クロックカウントＤが１秒に相当するクロックカウント値（以下、単に「１秒カウント」という）以上であるか否かを判断する（Ｓ５４３）。本実施形態において、サブＣＰＵ８１は、２２ＭＨｚのクロックで動作しているため、１秒カウントは、２２，０００，０００（パルス）となる。

【０２２８】

Ｓ５４３において、クロックカウントＤが１秒カウント以上でないと判断した場合には（ＮＯ）、サブＣＰＵ８１は、日時更新処理を終了する。Ｓ５４３において、クロックカウントＤが１秒カウント以上であると判断した場合には（ＹＥＳ）、サブＣＰＵ８１は、クロックカウントＤから１秒カウントを減算する（Ｓ５４４）。

【０２２９】

Ｓ５４４の処理を実行した後、サブＣＰＵ８１は、クロックカウントＰに１秒カウントを加算する（Ｓ５４５）。次に、サブＣＰＵ８１は、日時格納領域（秒）に１を加算する（Ｓ５４６）。

【０２３０】

次に、サブＣＰＵ８１は、日時格納領域（秒）が６０以上であるか否かを判断する（Ｓ５４７）。Ｓ５４７において、日時格納領域（秒）が６０以上でないと判断した場合には（ＮＯ）、サブＣＰＵ８１は、Ｓ５４３の処理を実行する。

【０２３１】

Ｓ５４７において、日時格納領域（秒）が６０以上であると判断した場合には（ＹＥＳ）、サブＣＰＵ８１は、日時格納領域（秒）を０に更新し、日時格納領域（分）に１を加算する（Ｓ５４８）。

【０２３２】

Ｓ５４８の処理を実行した後、サブＣＰＵ８１は、日時格納領域（分）が６０以上であるか否かを判断する（Ｓ５４９）。Ｓ５４９において、日時格納領域（分）が６０以上でないと判断した場合には（ＮＯ）、サブＣＰＵ８１は、Ｓ５４３の処理を実行する。

【０２３３】

Ｓ５４９において、日時格納領域（分）が６０以上であると判断した場合には（ＹＥＳ）、サブＣＰＵ８１は、日時格納領域（分）を０に更新し、日時格納領域（時）に１を加算する（Ｓ５５０）。

【０２３４】

Ｓ５５０の処理を実行した後、サブＣＰＵ８１は、日時格納領域（時）が２４以上であるか否かを判断する（Ｓ５５１）。Ｓ５５１において、日時格納領域（時）が２４以上でないと判断した場合には（ＮＯ）、サブＣＰＵ８１は、Ｓ５４３の処理を実行する。

【０２３５】

Ｓ５５１において、日時格納領域（時）が２４以上であると判断した場合には（ＹＥＳ）、サブＣＰＵ８１は、日時格納領域（時）を０に更新し、日時格納領域（日）及び日時格納領域（曜日）に１を加算する（Ｓ５５２）。

【０２３６】

Ｓ５５２の処理を実行した後、サブＣＰＵ８１は、日時格納領域（曜日）が７以上であるか否かを判断する（Ｓ５５３）。Ｓ５５３において、日時格納領域（曜日）が７以上であると判断した場合には（ＹＥＳ）、サブＣＰＵ８１は、日時格納領域（曜日）を０に更新する（Ｓ５５４）。

【０２３７】

Ｓ５５３において、日時格納領域（曜日）が７以上でないと判断した場合（ＮＯ）、又は、Ｓ５５４の処理を実行した後、サブＣＰＵ８１は、日時格納領域（年）が表す年が閏年であるか否かを算出する（Ｓ５５５）。

【０２３８】

なお、閏年の算出は、西暦年を「４」、「１００」及び「４００」で除算して余りがない年（例えば、２０００年等）、及び、西暦年を「４」を除算して余りがない、且つ、西暦年を「１００」で除算して余りがある年（例えば、２０２０年等）が閏年となる。

【０２３９】

次に、サブＣＰＵ８１は、月別日テーブル（図１２参照）によって、Ｓ５５５の算出結果と、日時格納領域（月）とに対応付けられている日数（以下、「月日数」ともいう）を特定し、日時格納領域（日）が月日数以上であるか否かを判断する（Ｓ５５６）。

【０２４０】

Ｓ５５６において、日時格納領域（日）が月日数以上でないと判断した場合には（ＮＯ）、サブＣＰＵ８１は、Ｓ５４３の処理を実行する。Ｓ５５６において、日時格納領域（日）が月日数以上であると判断した場合には（ＹＥＳ）、サブＣＰＵ８１は、日時格納領域（日）を１に更新し、日時格納領域（月）に１を加算する（Ｓ５５７）。

【０２４１】

Ｓ５５７の処理を実行した後、サブＣＰＵ８１は、日時格納領域（月）が１２以上であるか否かを判断する（Ｓ５５８）。Ｓ５５８において、日時格納領域（月）が１２以上でないと判断した場合には（ＮＯ）、サブＣＰＵ８１は、Ｓ５４３の処理を実行する。

【０２４２】

Ｓ５５８において、日時格納領域（月）が１２以上であると判断した場合には（ＹＥＳ）、サブＣＰＵ８１は、日時格納領域（月）を１に更新し、日時格納領域（年）に１を加算する（Ｓ５５９）。Ｓ５５９の処理を実行した後、サブＣＰＵ８１は、Ｓ５４３の処理を実行する。

【０２４３】

このように、日時更新処理を実行するサブＣＰＵ８１は、経過時間算出手段を構成する。日時更新処理は、日時格納領域に記憶された日時データに規定時間として１秒を加算していくことにより、日時データを更新する。したがって、本来であれば、ＲＴＣ８６から日時データを取得し、取得した日時データを日時格納領域に記憶することで、日時格納領域の日時データを更新するが、日時更新処理は、ＲＴＣ８６とシリアル通信を行うことなく日時格納領域の日時データを更新することにより、サブＣＰＵ８１はＲＴＣ８６との通信に係る処理負荷を省略することができる。

【０２４４】

一般的に、クロックカウントから経過日時を算出する場合には、クロックカウントを１日相当のクロックカウントで除算し、その商から、経過年数、経過月数及び経過日数を算出し、その余りを１時間相当のクロックカウントで除算して経過時間を算出し、その余りを１分相当のクロックカウントで除算して経過分数を算出し、その余りを１秒相当のクロックカウントで除算して経過秒数を算出する。

【０２４５】

ＣＰＵによる除算処理は、浮動小数点演算又は繰り返し演算によって実行される。このため、除算は、四則演算のなかで、ＣＰＵにかかる処理負荷が最も高い。本実施形態における日時更新処理は、処理負荷が高い除算処理を行わずに加算処理だけで日時格納領域に記憶された日時データを更新することにより、サブＣＰＵ８１にかかる負荷を低減させている。

【０２４６】

また、本実施形態では、電源投入処理（図２１）でＲＴＣ８６から日時データを取得し、クロックカウンタを用いて日時データを更新する説明をしたが、本発明はこれに限定されない、例えば、ステップＳ５５０の換わりに１時間毎にＲＴＣ８６から日時データを読み込むようにしてもよい。

【０２４７】
［サウンド制御タスク］
　次に、図２７を参照して、サブＣＰＵ８１により行われるサウンド制御タスクについて説明する。
【０２４８】
　まず、サブＣＰＵ８１は、サブＲＯＭ８２に記憶されている制御プログラムのうち所定のプログラムをサブＲＡＭ８３に転送するサウンド関数ＲＡＭ転送処理を実行する（Ｓ５８１）。サウンド関数ＲＡＭ転送処理については、図２８を参照して後述する。
【０２４９】
　次に、サブＣＰＵ８１は、演出登録タスク（図２３のＳ５１５参照）で登録されたサウンドデータが更新されたかた否かを判断する（Ｓ５８２）。このサウンドデータは、一定の期間におけるスピーカ２０Ｌ，２０Ｒ（図２参照）を駆動制御するためのデータである。
【０２５０】
　Ｓ５８２において、サブＣＰＵ８１は、登録されたサウンドデータが更新されたと判断した場合には（ＹＥＳ）、Ｓ５８３の処理を実行し、登録されたサウンドデータが更新されなかったと判断した場合には（ＮＯ）、Ｓ５８４の処理を実行する。
【０２５１】
　Ｓ５８３において、サブＣＰＵ８１は、登録されたサウンドデータを取得する（Ｓ５８３）。Ｓ５８３の処理を実行した後、サブＣＰＵ８１は、Ｓ５８４の処理を実行する。Ｓ５８４において、サブＣＰＵ８１は、サウンドデータとサウンドデータの再生位置とに応じてサウンド制御データを生成するサウンド制御データ生成処理を実行する。
【０２５２】
　サウンド制御データは、所定の周期毎のスピーカ２０Ｌ，２０Ｒを駆動制御するためのデータである。すなわち、サブＣＰＵ８１は、一定の期間におけるスピーカ２０Ｌ，２０Ｒを駆動制御するためのデータであるサウンドデータを、所定の周期毎に分けてサウンド制御データを作成する。
【０２５３】
　Ｓ５８４の処理を実行した後、サブＣＰＵ８１は、Ｓ５８５の処理を実行する。Ｓ５８５において、サブＣＰＵ８１は、サウンド制御データ生成処理（Ｓ５８４）で生成したサウンド制御データをサウンドドライバ（ソフトウェア）に登録する。
【０２５４】
　このように、サウンド制御データをサウンドドライバに登録することによって、サウンドドライバの処理を実行するサブＣＰＵ８１は、サウンド制御データをサウンドＩＣ８８に送信する。また、サウンド制御データをサウンドＩＣ８８に送信する際に、サブＣＰＵ８１は、サウンドＩＣの再生状態等を受信する。
【０２５５】
　Ｓ５８５の処理を実行した後、サブＣＰＵ８１は、Ｓ５８６の処理を実行する。Ｓ５８６において、サブＣＰＵ８１は、再生中のサウンドが再生終了である否かを判断する。Ｓ５８６において、サブＣＰＵ８１は、再生中のサウンドが再生終了であると判断した場合には（ＹＥＳ）、Ｓ５８７の処理を実行し、再生中のサウンドが再生終了でないと判断した場合には（ＮＯ）、Ｓ５８２の処理を実行する。
【０２５６】
　Ｓ５８７において、サブＣＰＵ８１は、再生中のサウンドがループ再生である否かを判断する。Ｓ５８７において、サブＣＰＵ８１は、再生中のサウンドがループ再生であると判断した場合には（ＹＥＳ）、Ｓ５８８の処理を実行し、再生中のサウンドがループ再生でないと判断した場合には（ＮＯ）、Ｓ５８２の処理を実行する。
【０２５７】
　Ｓ５８８において、サブＣＰＵ８１は、サウンドデータ頭出処理を実行する。サウンドデータ頭出処理において、サブＣＰＵ８１は、サウンドデータの再生位置を先頭に戻す。Ｓ５８８の処理を実行した後、サブＣＰＵ８１は、Ｓ５８２の処理を実行する。

【０２５８】
［サウンド関数ＲＡＭ転送処理］
次に、図２８を参照して、サウンド関数ＲＡＭ転送処理について説明する。
【０２５９】
まず、サブＣＰＵ８１は、サウンド制御タスクにおけるサウンド制御データ生成処理（図２７のＳ５８４参照）を実行するためのプログラムをサブＲＯＭ８２からサブＲＡＭ８３に転送する（Ｓ５９１）。
【０２６０】
次に、サブＣＰＵ８１は、サウンドドライバをサブＲＯＭ８２からサブＲＡＭ８３に転送する（Ｓ５９２）。Ｓ５９２の処理を実行した後、サブＣＰＵ８１は、サウンド関数ＲＡＭ転送処理を終了する。
【０２６１】
サブＣＰＵ８１は、上位プログラムであるサウンド制御タスクのプログラムからサブＲＡＭ８３に転送したサウンド制御データ生成処理のプログラムを実行する。また、サブＣＰＵ８１は、サウンドドライバとしての処理をサブＲＡＭ８３に転送したプログラムにしたがって実行する。
【０２６２】
このように、サブＣＰＵ８１は、サウンド関数ＲＡＭ転送処理において、相対的に処理負荷が高い所定のプログラムであるサウンドデータ生成処理を実行するためのプログラム及びサウンドドライバをサブＲＯＭ８２からサブＲＡＭ８３に転送して実行することによって、処理速度を向上させている。
【０２６３】
［サウンド制御タスク及びサウンド関数ＲＡＭ転送処理の具体例］
図２９を参照して、図２７及び図２８を参照してそれぞれ説明したサウンド制御タスク及びサウンド関数ＲＡＭ転送処理について具体的に説明する。なお、図２９において、サウンド制御タスク及びサウンド関数ＲＡＭ転送処理は、それぞれＣ言語で表されている。
【０２６４】
「SoundRamTrans()」は、図２８に示したサウンド関数ＲＡＭ転送処理を実行するためのプログラムである。サブＲＯＭ８２のアドレス「_SoundMakeData」で表される位置には、サウンド制御タスクのステップＳ５８４の処理を実行するためのプログラムが格納されている。
【０２６５】
サウンド関数ＲＡＭ転送処理のステップＳ５９１では、メモリコピー関数（memcpy）によって、サブＲＯＭ８２のアドレス「_SoundMakeData」からサブＲＡＭ８３のアドレス「__SoundMakeData」で表される位置に、プログラムサイズ（sizeof(_SoundMakeData)）分のメモリコピーが行われる。
【０２６６】
このように、サウンド制御タスクのステップＳ５８４の処理を実行するためのプログラムがサブＲＡＭ８３に転送される。なお、本実施形態において、サブＲＡＭ８３のアドレス「__SoundMakeData」で表される位置には、当該プログラムを格納するための領域がサウンド関数ＲＡＭ転送処理の実行前に確保されていることとする。
【０２６７】
サブＲＯＭ８２のアドレス「_SetSoundDriver」で表される位置には、サウンド制御タスクのステップＳ５８５の処理を実行するためのプログラムが格納されている。サウンド関数ＲＡＭ転送処理のステップＳ５９２では、メモリコピー関数（memcpy）によって、サブＲＯＭ８２のアドレス「_SetSoundDriver」からサブＲＡＭ８３のアドレス「__SetSoundDriver」で表される位置に、プログラムサイズ（sizeof(_SetSoundDriver)）分のメモリコピーが行われる。
【０２６８】
このように、サウンド制御タスクのステップＳ５８５の処理を実行するためのプログラ

ムがサブＲＡＭ８３に転送される。なお、本実施形態において、サブＲＡＭ８３のアドレス「__SetSoundDriver」で表される位置には、当該プログラムを格納するための領域がサウンド関数ＲＡＭ転送処理の実行前に確保されていることとする。

【０２６９】

「SoundTask()」は、図２７に示したサウンド制御タスクを実行するためのプログラムである。なお、図２９に示した「SoundTask()」は、ステップＳ５８３～Ｓ５８５の処理を表し、他の処理の記載は省略されている。

【０２７０】

「SoundTask()」において、「index」は、サウンドデータを識別するためのリクエスト番号を表す変数として用いられ、「SoundData」は、サブＲＯＭ８２のサウンドデータの格納先のアドレスを表すポインタ変数として用いられ、「SoundBuf」は、サブＲＡＭ８３のサウンド制御データの格納先のアドレスを表すポインタ変数として用いられる。

【０２７１】

「GetSoundData()」は、サウンド制御タスクのステップＳ５８３の処理を実行し、登録されたサウンドデータを取得する。具体的には、「GetSoundData()」は、「index」によって登録されたサウンドデータのリクエスト番号が指定されると、指定されたサウンドデータの格納先のアドレスを「SoundData」に格納する。

【０２７２】

「SoundMakeData()」は、図２８に示したサウンド関数ＲＡＭ転送処理のステップＳ５９１でサブＲＡＭ８３のアドレス「__SoundMakeData」に転送されたプログラム「__SoundMakeData()」のアクセス関数である。

【０２７３】

「__SoundMakeData()」は、引数「dat」が表すサウンドデータの格納先のアドレスが指定されると、アドレスが指定されたサウンドデータに基づいてサウンド制御データをサブＲＡＭ８３に生成し、生成したサウンド制御データの格納先のアドレスを引数「buf」に格納する。

【０２７４】

したがって、「SoundTask()」における「SoundMakeData()」は、サウンド制御タスクのステップＳ５８４の処理を実行し、サウンドデータの格納先のアドレスを表すポインタ変数「SoundData」が指定されることで、ポインタ変数「SoundBuf」にサウンド制御データの格納先であるサブＲＡＭ８３のアドレスを格納する。

【０２７５】

「SetSoundDriver()」は、図２８に示したサウンド関数ＲＡＭ転送処理のステップＳ５９１でサブＲＡＭ８３のアドレス「__SetSoundDriver」に転送されたプログラム「__SetSoundDriver()」のアクセス関数である。「__SetSoundDriver()」は、引数「buf」が表すサウンド制御データの格納先であるサブＲＡＭ８３のアドレスが指定されると、指定されたアドレスをサウンドドライバに登録する。

【０２７６】

したがって、「SoundTask()」における「SetSoundDriver()」は、サウンド制御タスクのステップＳ５８５の処理を実行し、サウンド制御データの格納先のアドレスを表す「SoundBuf」が指定されることで、サウンド制御データの格納先であるサブＲＡＭ８３のアドレスをサウンドドライバに登録する。

【０２７７】

［ランプ制御タスク］

次に、図３０を参照して、サブＣＰＵ８１により行われるランプ制御タスクについて説明する。

【０２７８】

まず、サブＣＰＵ８１は、サブＲＯＭ８２に記憶されている制御プログラムのうち所定のプログラムをサブＲＡＭ８３に転送するランプ関数ＲＡＭ転送処理を実行する（Ｓ６０１）。ランプ関数ＲＡＭ転送処理については、図３２を参照して後述する。

【０２７９】
　次に、サブＣＰＵ８１は、演出登録タスク（図２３のＳ５１６参照）等で登録されたランプデータをサブＲＡＭ８３に読み込むランプデータ読み込み処理を実行する（Ｓ６０２）。ランプデータ読み込み処理については、図３１を参照して後述する。
【０２８０】
　次に、サブＣＰＵ８１は、再生状態管理格納領域（図１４参照）のエントリパーツ番号格納領域の値に基づいて、ランプデータの再生中であるか否かを判断する（Ｓ６０３）。前述したように、ランプデータの再生中である場合には、再生状態管理格納領域のエントリパーツ番号格納領域に再生中のパーツデータを識別するためのパーツ番号が格納されている。一方、後述するように、ランプデータの再生中でない場合には、再生状態管理格納領域のエントリパーツ番号格納領域に特定値（例えば、「０」）が格納されている。
【０２８１】
　このため、本実施形態におけるサブＣＰＵ８１は、再生状態管理格納領域のエントリパーツ番号格納領域の値が特定値でなければ、ランプデータの再生中であると判断し、再生状態管理格納領域のエントリパーツ番号格納領域の値が特定値であれば、ランプデータの再生中でないと判断する。
【０２８２】
　Ｓ６０３において、サブＣＰＵ８１は、ランプデータの再生中であると判断した場合には（ＹＥＳ）、Ｓ６０４の処理を実行し、ランプデータの再生中でないと判断した場合には（ＮＯ）、Ｓ６０２の処理を実行する。
【０２８３】
　Ｓ６０４において、サブＣＰＵ８１は、ランプデータにおける再生中のパーツデータの属性データを取得する。次に、サブＣＰＵ８１は、再生中のパーツデータの再生位置における輝度パターンがエンドブロック（例えば、「－１」）であるか否かを判断する（Ｓ６０５）。
【０２８４】
　Ｓ６０５において、サブＣＰＵ８１は、再生中のパーツデータの再生位置における輝度パターンがエンドブロックであると判断した場合には（ＹＥＳ）、Ｓ６０８の処理を実行し、エンドブロックでないと判断した場合には（ＮＯ）、Ｓ６０６の処理を実行する。
【０２８５】
　Ｓ６０６において、サブＣＰＵ８１は、再生中のパーツデータの再生位置における輝度パターンから制御データ（図８～図１０参照）を生成する。前述したように、本実施形態において、制御データは、６ビット（０～６３の範囲）で輝度を表し、輝度パターンは、８ビット（０～２５５の範囲）で輝度を表す。
【０２８６】
　このため、サブＣＰＵ８１は、輝度パターンにおける輝度値をビットシフトにより６ビットの輝度値に変換（８ビットデータである輝度値をＬＳＢ方向に２ビットシフト）して、変換した輝度値に基づいてシリアルバス通信用の通信回路のデータフォーマットに応じた制御データを生成する。すなわち、サブＣＰＵ８１は、６ビットに変換した輝度値を制御データの該当する輝度データＬに登録する。
【０２８７】
　なお、本実施形態において、輝度パターンの輝度値を８ビットで表しているが、これは輝度値が８ビット又は７ビットに対応するドライバＩＣを使用した場合であっても、輝度値を変換する処理を変更するだけで、輝度パターンを転用可能とするためであり、ドライバＩＣの種類が異なってもランプデータを流用することが可能となる。
【０２８８】
　Ｓ６０６の処理を実行した後、Ｓ６０７の処理を実行する。Ｓ６０７において、サブＣＰＵ８１は、ランプ制御データ生成処理（Ｓ６０６）で生成した制御データをランプドライバ（ソフトウェア）に登録する。
【０２８９】

このように、制御データをランプドライバに登録することによって、ランプドライバの処理を実行するサブＣＰＵ８１は、制御データをドライバＩＣに送信する。Ｓ６０７の処理を実行した後、サブＣＰＵ８１は、Ｓ６０２の処理を実行する。

【０２９０】

Ｓ６０８において、サブＣＰＵ８１は、再生状態管理格納領域（図１４参照）のエントリパーツ番号格納領域とパーツ番号格納領域とを基に、再生中のパーツデータの次のパーツデータがあるか否かを判断する。Ｓ６０８において、サブＣＰＵ８１は、次のパーツデータがあると判断した場合には（ＹＥＳ）、Ｓ６０９の処理を実行し、次のパーツデータがないと判断した場合（例えば、図１６の場合、再生中パーツデータがパーツ番号：８、又はエンドコード格納領域（図１４参照）のエンドコードを検出）には（ＮＯ）、Ｓ６１０の処理を実行する。

【０２９１】

Ｓ６０９において、サブＣＰＵ８１は、再生状態管理格納領域のエントリパーツ番号格納領域を次のパーツ番号格納領域の値に更新し、再生位置を次のパーツデータに移動させる。Ｓ６１０において、サブＣＰＵ８１は、Ｓ６０４で取得した属性データがショット＋チェインであるか否かを判断する。

【０２９２】

Ｓ６１０において、サブＣＰＵ８１は、属性データがショット＋チェインであると判断した場合には（ＹＥＳ）、属性検索頭出処理を実行する（Ｓ６１１）。属性検索頭出処理において、サブＣＰＵ８１は、再生状態管理格納領域（図１４～図１８参照）に記憶された内容に基づいて、チェイン／ループ設定領域に「ＴＲＵＥ」が格納されていれば、ランプデータが表す識別情報の先頭（例えば、図１６の場合、パーツ番号：１）から属性データがショット＋チェインを表すパーツデータを検索（例えば、図１６の場合、パーツ番号：２）し、検出したパーツデータの先頭に再生位置を移動（例えば、図１４のエントリパーツ番号格納領域の値が「２」がセット）させる。Ｓ６１１の処理を実行した後、サブＣＰＵ８１は、Ｓ６０２の処理を実行する。

【０２９３】

Ｓ６１０において、属性データがショット＋チェインでないと判断した場合には（ＮＯ）、サブＣＰＵ８１は、Ｓ６０４で取得した属性データがループであるか否かを判断する（Ｓ６１２）。Ｓ６１２において、属性データがループでないと判断した場合には（ＮＯ）、サブＣＰＵ８１は、Ｓ６０２の処理を実行する。

【０２９４】

Ｓ６１２において、属性データがループであると判断した場合には（ＹＥＳ）、サブＣＰＵ８１は、パーツ内頭出処理を実行する（Ｓ６１３）。属性検索頭出処理において、サブＣＰＵ８１は、再生中のパーツデータの先頭に再生位置を移動させる。Ｓ６１３の処理を実行した後、サブＣＰＵ８１は、サブＣＰＵ８１は、Ｓ６０２の処理を実行する。

【０２９５】

なお、パーツ内頭出処理とは、属性データがループであり、パーツデータが「０」～「９」まで、昇順に変化するデータであれば、パーツデータの先頭である「０」に頭出しされ、「０」～「９」までを繰り返す、リピート機能である。

【０２９６】

［ランプデータ読み込み処理］

次に、図３１を参照して、ランプデータ読み込み処理について説明する。

【０２９７】

まず、サブＣＰＵ８１は、パチスロ１のエラーが検出されたか否かを判断する（Ｓ６３１）。パチスロ１のエラーには、ホッパー３３（図５参照）が空になったことを表すホッパーエンプティエラー、メダル補助庫３７（図５参照）に収容されたメダルが規定量に達したことを表す投入メダル補助収納庫満杯エラーなどがある。

【０２９８】

メインＣＰＵ５１は、これらのエラーを表すエラーコマンドを副制御回路４２にする。

サブＣＰＵ８１は、このエラーコマンドに基づきパチスロ１のエラーが検出されたか否かを判断する。

【０２９９】

Ｓ６３１において、サブＣＰＵ８１は、パチスロ１のエラーが検出されたと判断した場合には（ＹＥＳ）、Ｓ６３２の処理を実行し、パチスロ１のエラーが検出されなかったと判断した場合には（ＮＯ）、Ｓ６３４の処理を実行する。

【０３００】

Ｓ６３２において、サブＣＰＵ８１は、ランプデータの再生中であれば、再生中のランプデータのリクエスト番号をサブＲＡＭ８３に保存する。Ｓ６３２の処理を実行した後、サブＣＰＵ８１は、エラー用のランプデータを登録する（Ｓ６３３）。Ｓ６３３の処理を実行した後、サブＣＰＵ８１は、Ｓ６３８の処理を実行する。

【０３０１】

Ｓ６３４において、サブＣＰＵ８１は、パチスロ１のエラーが解除されたか否かを判断する（Ｓ６３４）。Ｓ６３４において、サブＣＰＵ８１は、パチスロ１のエラーが解除されたと判断した場合には（ＹＥＳ）、Ｓ６３５の処理を実行し、パチスロ１のエラーが解除されていないと判断した場合には（ＮＯ）、Ｓ６３７の処理を実行する。

【０３０２】

Ｓ６３５において、サブＣＰＵ８１は、Ｓ６３２で保存したリクエスト番号に対応する中断復帰用ランプデータのリクエスト番号を特定する。Ｓ６３５の処理を実行した後、サブＣＰＵ８１は、Ｓ６３５で特定したリクエスト番号のエラー用のランプデータを登録する（Ｓ６３６）。Ｓ６３６の処理を実行した後、サブＣＰＵ８１は、Ｓ６３８の処理を実行する。

【０３０３】

Ｓ６３７において、サブＣＰＵ８１は、演出登録タスクのＳ５１６（図２３参照）で登録されたランプデータが更新されたかた否かを判断する（Ｓ６３７）。Ｓ６３７において、サブＣＰＵ８１は、ランプデータが更新されたと判断した場合には（ＹＥＳ）、Ｓ６３８の処理を実行し、ランプデータが更新されていないと判断した場合には（ＮＯ）、ランプデータ読み込み処理を終了する。

【０３０４】

Ｓ６３８において、サブＣＰＵ８１は、Ｓ６３３、Ｓ６３６又は演出登録タスクのＳ５１６で登録されたランプデータを取得する。Ｓ６３８の処理を実行した後、サブＣＰＵ８１は、Ｓ６３８で取得したランプデータに基づきサブＲＡＭ８３に再生状態管理格納領域（図１４参照）を生成する再生データ生成処理を実行する（Ｓ６３９）。

【０３０５】

本実施形態におけるサブＣＰＵ８１は、ランプデータの再生を終了させる場合には、演出登録タスクのＳ５１６でブランクのランプデータを登録する。ブランクのランプデータを登録した場合には、サブＣＰＵ８１は、エントリパーツ番号格納領域に特定値（例えば、「０」）が設定され、パーツ番号格納領域を含まない再生状態管理格納領域をＳ６３９で生成する。Ｓ６３９の処理を実行した後、サブＣＰＵ８１は、ランプデータ読み込み処理を終了する。

【０３０６】

［ランプ関数ＲＡＭ転送処理］

次に、図３２を参照して、ランプ関数ＲＡＭ転送処理について説明する。

【０３０７】

まず、サブＣＰＵ８１は、ランプデータ読み込み処理における再生データ生成処理（図３１のＳ６３９参照）を実行するためのプログラムをサブＲＯＭ８２からサブＲＡＭ８３に転送する（Ｓ６５１）。

【０３０８】

次に、サブＣＰＵ８１は、ランプ制御タスクにおけるランプ制御データ生成処理（図３０のＳ６０６）を実行するためのプログラムをサブＲＯＭ８２からサブＲＡＭ８３に転送

する（Ｓ６５２）。

【０３０９】

次に、サブＣＰＵ８１は、ランプドライバをサブＲＯＭ８２からサブＲＡＭ８３に転送する（Ｓ６５３）。Ｓ６５３の処理を実行した後、サブＣＰＵ８１は、ランプ関数ＲＡＭ転送処理を終了する。

【０３１０】

サブＣＰＵ８１は、上位プログラムであるランプデータ読み込み処理のプログラムから再生データ生成処理のプログラムを実行し、上位プログラムであるランプ制御タスクのプログラムからランプ制御データ生成処理のプログラムを実行する。また、サブＣＰＵ８１は、ランプドライバとしての処理をサブＲＡＭ８３に転送したプログラムにしたがって実行する。

【０３１１】

このように、サブＣＰＵ８１は、ランプ関数ＲＡＭ転送処理において、相対的に処理負荷が高い所定のプログラムである再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラム並びにランプドライバをサブＲＯＭ８２からサブＲＡＭ８３に転送して実行することによって、処理速度を向上させている。

【０３１２】

［ランプ制御タスク、ランプデータ読み込み処理及びランプ関数ＲＡＭ転送処理の具体例］

図３３を参照して、図３０～図３２を参照してそれぞれ説明したランプ制御タスク、ランプデータ読み込み処理及びランプ関数ＲＡＭ転送処理について具体的に説明する。なお、図３３において、ランプ制御タスク及びランプ関数ＲＡＭ転送処理は、それぞれＣ言語で表されている。

【０３１３】

「LampRamTrans()」は、図３２に示したランプ関数ＲＡＭ転送処理を実行するためのプログラムである。サブＲＯＭ８２のアドレス「_LampPlayMake」で表される位置には、図３１に示したランプデータ読み込み処理のステップＳ６３９の処理を実行するためのプログラムが格納されている。

【０３１４】

ランプ関数ＲＡＭ転送処理のステップＳ６５１では、メモリコピー関数（memcpy）によって、サブＲＯＭ８２のアドレス「_LampPlayMake」からサブＲＡＭ８３のアドレス「__LampPlayMake」で表される位置に、プログラムサイズ（sizeof(_LampPlayMake)）分のメモリコピーが行われる。

【０３１５】

このように、図３１に示したランプデータ読み込み処理のステップＳ６３９の処理を実行するためのプログラムがサブＲＡＭ８３に転送される。なお、本実施形態において、サブＲＡＭ８３のアドレス「__LampPlayMake」で表される位置には、当該プログラムを格納するための領域がランプ関数ＲＡＭ転送処理の実行前に確保されていることとする。

【０３１６】

サブＲＯＭ８２のアドレス「_LampMakeData」で表される位置には、図３０に示したランプ制御タスクのステップＳ６０６の処理を実行するためのプログラムが格納されている。ランプ関数ＲＡＭ転送処理のステップＳ６５２では、メモリコピー関数（memcpy）によって、サブＲＯＭ８２のアドレス「_LampMakeData」からサブＲＡＭ８３のアドレス「__LampMakeData」で表される位置に、プログラムサイズ（sizeof(_LampMakeData)）分のメモリコピーが行われる。

【０３１７】

このように、図３０に示したランプ制御タスクのステップＳ６０６の処理を実行するためのプログラムがサブＲＡＭ８３に転送される。なお、本実施形態において、サブＲＡＭ８３のアドレス「__LampMakeData」で表される位置には、当該プログラムを格納するための領域がランプ関数ＲＡＭ転送処理の実行前に確保されていることとする。

【０３１８】

サブＲＯＭ８２のアドレス「_SetLampDriver」で表される位置には、図３０に示したランプ制御タスクのステップＳ６０７の処理を実行するためのプログラムが格納されている。ランプ関数ＲＡＭ転送処理のステップＳ５９３では、メモリコピー関数（memcpy）によって、サブＲＯＭ８２のアドレス「_SetLampDriver」からサブＲＡＭ８３のアドレス「__SetLampDriver」で表される位置に、プログラムサイズ（sizeof(_SetLampDriver)）分のメモリコピーが行われる。

【０３１９】

このように、図３０に示したランプ制御タスクのステップＳ６０７の処理を実行するためのプログラムがサブＲＡＭ８３に転送される。なお、本実施形態において、サブＲＡＭ８３のアドレス「__SetLampDriver」で表される位置には、当該プログラムを格納するための領域がランプ関数ＲＡＭ転送処理の実行前に確保されていることとする。

【０３２０】

「LampTask()」は、図３０に示したランプ制御タスクを実行するためのプログラムである。なお、図３３に示した「LampTask()」は、ステップＳ６０５～Ｓ６０７の処理を表し、他の処理の記載は省略されている。

【０３２１】

「LampTask()」において、「LampData」は、サブＲＯＭ８２の輝度データの格納先のアドレスを表すポインタ変数として用いられ、「LampBuf」は、サブＲＡＭ８３のランプ制御データの格納先のアドレスを表すポインタ変数として用いられる。「if(*LampData!=EndBlock)」の判定文は、ステップＳ６０５の処理を表す。

【０３２２】

「LampPlayMake()」は、図３２に示したランプ関数ＲＡＭ転送処理のステップＳ６５１でサブＲＡＭ８３のアドレス「__LampPlayMake」に転送されたプログラム「__LampPlayMake()」のアクセス関数である。

【０３２３】

「__LampPlayMake()」は、引数「dat」が表す輝度パターンの格納先のアドレスが指定されると、アドレスが指定された輝度パターンをサブＲＡＭ８３の再生状態管理格納領域（図１４参照）に読み込み、再生状態管理格納領域のアドレスを引数「buf」に格納する。

【０３２４】

「LampMakeData()」は、図３２に示したランプ関数ＲＡＭ転送処理のステップＳ６５２でサブＲＡＭ８３のアドレス「__LampMakeData」に転送されたプログラム「__LampMakeData()」のアクセス関数である。

【０３２５】

「__LampMakeData()」は、引数「dat」が表す輝度パターンの格納先のアドレスが指定されると、アドレスが指定された輝度パターンに基づいてランプ制御データをサブＲＡＭ８３に生成し、生成したランプ制御データの格納先のアドレスを引数「buf」に格納する。

【０３２６】

したがって、「LampTask()」における「LampMakeData()」は、図３０に示したランプ制御タスクのステップＳ６０６の処理を実行し、輝度パターンの格納先のアドレスを表すポインタ変数「LampData」が指定されることで、ポインタ変数「LampBuf」にランプ制御データの格納先であるサブＲＡＭ８３のアドレスを格納する。

【０３２７】

「SetLampDriver()」は、図３２に示したランプ関数ＲＡＭ転送処理のステップＳ６５３でサブＲＡＭ８３のアドレス「__SetLampDriver」に転送されたプログラム「__SetLampDriver()」のアクセス関数である。「__SetLampDriver()」は、「buf」が表すランプ制御データの格納先であるサブＲＡＭ８３のアドレスが指定されると、指定されたアドレスをランプドライバに登録する。

【０３２８】

したがって、「LampTask()」における「SetLampDriver()」は、図３０に示したランプ制御タスクのステップＳ６０７の処理を実行し、ランプ制御データの格納先のアドレスを表すポインタ変数「LampBuf」が指定されることで、ランプ制御データの格納先であるサブＲＡＭ８３のアドレスをランプドライバに登録する。

【０３２９】

［サブＣＰＵに実行させるプログラムの製造方法］

次に、図３４を参照して、副制御回路４２のサブＣＰＵに実行させるプログラムの製造方法について説明する。

【０３３０】

本実施形態において、サブＣＰＵ８１は、サブＲＯＭ８２に記憶されている制御プログラムを読み込みながら実行している。一方で、一般的に、ＲＯＭの読み込み速度よりＲＡＭの読み込み速度の方が、半導体の特性として速い。

【０３３１】

このため、サブＣＰＵ８１は、一部の処理負荷の高いプログラム（一般的に、「関数」又は「サブルーチン」などと呼称されている。）をサブＲＡＭ８３に転送し、サブＲＡＭ８３に転送したプログラムを読み込みながら実行することによって、一部の処理負荷の高いプログラムの実行速度を向上させている。

【０３３２】

ただし、サブＲＯＭ８２に記憶されているプログラムをサブＲＡＭ８３に転送しただけでは、転送したプログラムに記憶されているアドレスがサブＲＯＭ８２内のアドレスを指しているため正常に動作させることはできない。

【０３３３】

プログラムをサブＲＯＭ８２に記憶させる前に、サブＲＡＭ８３に転送するプログラム内のアドレスをサブＲＡＭ８３内のアドレスに変更する必要がある。本実施形態において、サブＲＡＭ８３に転送するプログラムは、図２９を参照して説明したサブＲＯＭ８２のアドレス「_SoundMakeData」及び「_SoundBuf」に格納されるプログラムと、図３３を参照して説明したサブＲＯＭ８２のアドレス「_LampPlayMake」、「_LampMakeData」及び「_LampBuf」に格納されるプログラムとを含む。

【０３３４】

また、サブＲＡＭ８３に転送するプログラムを使用する上位側プログラムにおいて、サブＲＡＭ８３に転送するプログラムの呼び出しアドレスをサブＲＡＭ８３内のアドレスに変更する必要がある。

【０３３５】

本実施形態において、サブＲＡＭ８３に転送するプログラムを使用する上位側プログラムは、図２９に示したアクセス関数である「SoundMakeData()」及び「SetSoundDriver()」と、図３３に示したアクセス関数である「LampPlayMake()」、「LampMakeData()」及び「SetLampDriver()」とを含む。

【０３３６】

したがって、本実施形態におけるプログラムの製造方法は、サブＲＡＭ８３に転送するプログラムと、サブＲＡＭ８３に転送するプログラムを使用する上位側プログラムと（以下、総称して「アドレス変換対象プログラム」ともいう）をアドレス変換対象プログラム以外のプログラムと異なる方法で製造する。

【０３３７】

図３４に示すように、本実施形態におけるプログラムの製造方法は、アドレス変換対象プログラムの対象Ｃ言語ソース群をコンパイルする工程Ｐ１を実行した後に、工程Ｐ１によって生成されたアセンブラソース群に対して上述したアドレス（具体低には、ラベル）の変更や命令（いわゆる、ニーモニック）の変更を行う工程Ｐ２を有する。

【０３３８】

変更内容として、例えば、ジャンプ先のラベルを変更し、変更前のラベルに絶対番地でジャンプする命令を変更後のラベルへの相対番地へジャンプする命令に書き換える。この

場合、元のＣ言語ソースから生成されたアセンブラソースのラベルはサブＲＯＭ８２を指しているため、サブＲＡＭ８３にジャンプできるように、ラベルを変更するとともに、相対位置（命令のある今のアドレスからプラス方向、又はマイナス方向への相対バイト数）へジャンプする命令に書き換える。

【０３３９】

また、本実施形態におけるプログラムの製造方法は、アドレス変換対象プログラム以外のプログラムの非対象Ｃ言語ソース群をコンパイルしてオブジェクトファイル（中間ファイル）を生成し、工程Ｐ２によってアドレスが変更されたＲＡＭ実行対応済みアセンブラソース群をアセンブルしてオブジェクトファイルを生成し、非対象Ｃ言語ソース群のオブジェクトファイルとＲＡＭ実行対応済みアセンブラソース群のオブジェクトファイルとを互いにリンクさせて最終的なプログラムを生成する工程Ｐ３と、工程Ｐ３によって生成されたプログラムをサブＲＯＭ８２に書き込む工程Ｐ４とを有する。

【０３４０】

［各種効果］

以上に説明したように、本発明の実施形態に係るパチスロ１は、実行する演出に応じたパターンデータをドライバＩＣとのシリアルバス通信用の通信回路のデータフォーマットに応じて変換し、変換したパターンデータに基づく制御データをドライバＩＣに送信することにより第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８及びドットマトリクス部１１９などの各種ランプ群を制御するため、ドライバＩＣの仕様に依存しないパターンデータによって各種ランプ群を制御することができる。

【０３４１】

特に、本発明の実施形態に係るパチスロ１は、実行する演出に応じたパターンデータに含まれるＬＥＤの輝度値をドライバＩＣに応じた輝度値に変換し、変換した輝度値に基づく制御データをドライバＩＣに送信することにより第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８及びドットマトリクス部１１９などの各種ランプ群を制御するため、ドライバＩＣの仕様に依存しないパターンデータによって各種ランプ群を制御することができる。

【０３４２】

また、本発明の実施形態に係るパチスロ１は、制御データにおいて、ＬＥＤの輝度を設定するためのデータに、エラー検出用データＥを同期させないことにより、データパターンの影響を受けることなく、サブＣＰＵ８１及びドライバＩＣ間のシリアルバス通信の異常を制御データの受信側で検出させることができる。

【０３４３】

また、本発明の実施形態に係るパチスロ１は、ドライバＩＣによって駆動される全てのＬＥＤの輝度を設定する第１モードと、ドライバＩＣに駆動させるＬＥＤを特定するチャネル番号及びチャネル番号によって特定されるＬＥＤの輝度を設定する第２モードと、のいずれのモードでも制御データを送信することができる。

【０３４４】

このため、マトリクス状に配置されたドットマトリクス部１１９を制御するときのように、ＬＥＤを個別に制御する場合には、特定のＬＥＤの輝度を設定する第２モードの方が、第１モードよりも制御データのデータ長が短くなる。

【０３４５】

一方、第１ランプ群１１１、第７ランプ群１１７及び第８ランプ群１１８を制御するときのように、ＬＥＤを全体的に制御する場合には、ＬＥＤを特定するチャネル番号を必要としない第１モードの方が、第２モードよりも制御データのデータ長が短くなる。

【０３４６】

また、全てのＬＥＤを消灯させる場合には、各ＬＥＤの輝度値の指定やＬＥＤを特定するチャネル番号を必要としない第３モードの方が、第１モード及び第２モードよりもデータ長が短くなる。

【０３４７】

このように、本発明の実施形態に係るパチスロ１は、ドライバＩＣが駆動するＬＥＤの用途に応じて、データ長が短くなる制御データのモードを選択することにより、サブＣＰＵ８１及びドライバＩＣ間の伝送負荷を低減させることができ、ＬＥＤの応答性を向上させることができる。

【０３４８】

また、本発明の実施形態に係るパチスロ１は、サブＣＰＵ８１に動作用のクロックをカウントさせることによって日時データを管理させる。したがって、サブＣＰＵ８１は、日時データを取得するたびに、ＲＴＣ８６とシリアル通信を行う必要がなくなる。このように、本発明の実施形態に係るパチスロ１は、サブＣＰＵ８１にかかる負荷を低減させることができる。

【０３４９】

また、本発明の実施形態に係るパチスロ１は、閏年であるときの日数と閏年でないときの日数とが月ごとに対応付けられている月別日テーブルを参照して日時データを更新するため、日時データを誤りなく更新することができる。

【０３５０】

また、本発明の実施形態に係るパチスロ１は、サブＲＡＭ８３の日時格納領域に記憶された日時データが表す日時に算出した経過時間が加算されるまで、サブＲＡＭ８３の日時格納領域に記憶された日時データに規定時間（１秒カウント）を加算することによって、処理負荷が高い除算処理を行わずに加算処理だけでサブＲＡＭ８３の日時格納領域に記憶された日時データを更新するため、サブＣＰＵ８１にかかる負荷を低減させることができる。

【０３５１】

また、本発明の実施形態に係るパチスロ１は、相対的に処理負荷が高いプログラムであるサウンドデータ生成処理を実行するためのプログラム、サウンドドライバ、再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラム並びにランプドライバをサブＲＡＭ８３に転送して実行することによって、サブＣＰＵ８１の処理速度を向上させることができる。

【０３５２】

また、本発明の実施形態に係るパチスロ１は、相対的に処理負荷が高いプログラム以外のプログラムをサブＲＯＭ８２に記憶された状態で実行することによって、サブＲＡＭ８３に必要な記憶容量を抑制するため、コストを抑制することができる。

【０３５３】

また、本発明の実施形態に係るパチスロ１は、パーツデータを組み合せることによって、第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８及びドットマトリクス部１１９などの各種ランプ群に多様な演出を実行させるため、パーツデータ単位で各種ランプ群の発光パターンの確認及び変更を行わせることができる。このように、本発明の実施形態に係るパチスロ１は、各種ランプ群の発光パターンの確認及び変更にかかる負担を低減することができる。

【０３５４】

また、本発明の実施形態に係るパチスロ１は、属性データが連続再生に設定されているパーツデータによって、第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８及びドットマトリクス部１１９などの各種ランプ群に演出を繰り返し実行させるため、各種ランプ群による演出を長時間にわたって実行させる場合であっても、各種ランプ群の発光パターンの確認及び変更にかかる負担を低減することができる。

【０３５５】

また、本発明の実施形態に係るパチスロ１は、エラー状態から復帰した場合、属性データがショット＋チェインを表していないパーツデータの識別情報が除かれた新たな演出データが表す識別情報の順序にしたがってパーツデータに含まれるパターンデータで第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８及びドットマトリクス部１１９などの各種ランプ群を制御するため、エラー復帰時に演出データを最初から再生すること

により遊技状態にそぐわない演出を実行してしまうことを防止することができる。

【０３５６】

［変形例］

本実施形態において、サブＣＰＵ８１は、電源投入時にＲＴＣ８６から取得した日時データをサブＲＡＭ８３に記憶し、サブＲＡＭ８３に記憶した日時データを動作用のクロックをカウントすることによって更新する例について説明した。

【０３５７】

これに加え、サブＣＰＵ８１は、所定の条件が成立するたびに、ＲＴＣ８６から日時データを取得し、サブＲＡＭ８３に記憶した日時データを調節（アジャスト）するようにしてもよい。所定の条件としては、カウントした値が所定値を超えたこと、スタートコマンドのような特定のコマンドを受信したこと、エラー状態から復帰したこと、又は、リセットボタン（不図示）が押下されたことなどが挙げられる。

【０３５８】

また、本実施形態において、相対的に処理負荷が高いプログラムであるサウンドデータ生成処理を実行するためのプログラム、サウンドドライバ、再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラム並びにランプドライバを起動時（本実施形態においては、サブＣＰＵ８１の電源投入時に起動されるサウンド制御タスク及びランプ制御タスクの初期化処理として、それぞれ実行されるサウンド関数ＲＡＭ転送処理及びランプ関数ＲＡＭ転送処理）にサブＲＡＭ８３に予め転送しておいて、実行する例について説明した。

【０３５９】

これに対し、本発明の実施形態に係るパチスロ１は、相対的に処理負荷が高い各プログラムを実行することを契機として各プログラムをサブＲＡＭ８３に転送して実行するようにしてもよい。

【０３６０】

また、本発明の実施形態に係るパチスロ１は、相対的に処理負荷が高い各プログラムを実行する場合、各プログラムがサブＲＡＭ８３に転送されていなければ、各プログラムをサブＲＡＭ８３に転送して実行するようにしてもよい。

【０３６１】

このように構成することにより、本発明の実施形態に係るパチスロ１は、相対的に処理負荷が高い各プログラムを実行する場合、各プログラムがサブＲＡＭ８３に転送されていれば、サブＲＡＭ８３に転送済みのプログラムを実行するため、各プログラムをサブＲＡＭ８３に転送するサブＣＰＵ８１の処理を省くことができる。

【０３６２】

また、本発明の実施形態に係るパチスロ１は、相対的に処理負荷が高い各プログラムを転送する領域がサブＲＡＭ８３に確保されていればサブＲＡＭ８３に転送し、各プログラムを転送する領域がサブＲＡＭ８３に確保されていなければサブＲＡＭ８３に各プログラムを転送する領域を確保し、確保した領域に各プログラムを転送して実行するようにしてもよい。

【０３６３】

このように構成することにより、本発明の実施形態に係るパチスロ１は、相対的に処理負荷が高い各プログラムを確実にサブＲＡＭ８３に転送して実行するため、サブＣＰＵ８１の処理速度を確実に向上させることができる。

【０３６４】

また、本発明の実施形態に係るパチスロ１は、相対的に処理負荷が高い各プログラムのうち、一部のプログラムを起動時にサブＲＡＭ８３に予め転送しておき、他のプログラムを実行時にサブＲＡＭ８３に転送するようにしてもよい。

【０３６５】

例えば、本発明の実施形態に係るパチスロ１は、サウンドデータ生成処理を実行するためのプログラム、サウンドドライバ、再生データ生成処理及びランプ制御データ生成処理

を実行するためのプログラム並びにランプドライバのうち、サウンドドライバ及びランプドライバを起動時にサブＲＡＭ８３に予め転送し、サウンドデータ生成処理、再生データ生成処理及びランプ制御データ生成処理を実行するためのプログラムを実行時にサブＲＡＭ８３に転送するようにしてもよい。

【０３６６】

また、本実施形態において、サブＣＰＵ８１は、第１発光体群のＬＥＤを駆動するドライバＩＣには、第２ポートから制御データを第２モードで送信し、第２発光体群のＬＥＤを駆動するドライバＩＣには、第１ポートから制御データを第１モードで送信する例について説明した。

【０３６７】

これに対し、サブＣＰＵ８１は、第１発光体群のＬＥＤを駆動するドライバＩＣには、第２ポートから制御データを第１モードで送信し、第２発光体群のＬＥＤを駆動するドライバＩＣには、第１ポートから制御データを第２モードで送信するようにしてもよい。

【０３６８】

また、サブＣＰＵ８１は、第１発光体群のＬＥＤを駆動するドライバＩＣには、第２ポートから制御データを第１モードで送信し、第２発光体群のＬＥＤを駆動するドライバＩＣには、第１ポートから制御データを第１モードで送信するようにしてもよい。

【０３６９】

また、サブＣＰＵ８１は、第１発光体群のＬＥＤを駆動するドライバＩＣには、第２ポートから制御データを第２モードで送信し、第２発光体群のＬＥＤを駆動するドライバＩＣには、第１ポートから制御データを第２モードで送信するようにしてもよい。

【０３７０】

また、サブＣＰＵ８１は、演出登録タスク（図２３のＳ５１６参照）等でランプデータを登録したときに、登録したランプデータから生成する制御データをどのモードで送信するかを決定するようにしてもよい。

【０３７１】

また、サブＲＯＭ８２に格納されたランプデータ又はパーツデータに予めモードを設定しておき、サブＣＰＵ８１は、ランプデータ又はパーツデータに予め設定されたモードで制御データを送信するようにしてもよい。

【０３７２】

［その他、本発明に係る遊技機の拡張性］

上記実施形態のパチスロ１では、遊技者のメダルの投入操作（すなわち、手持ちのメダルをメダル投入口１３に対して投入する操作、又は、クレジットされたメダルをＭＡＸベットボタン１４或いは１ベットボタン１５を操作して投入する操作）により遊技が開始され、遊技が終了したときにメダルの払い出しがある場合には、ホッパー３３を駆動してメダル払出口１８からメダルが払い出され、又は、クレジットされる形態について説明したが、本発明はこれに限定されるものではない。

【０３７３】

例えば、遊技者によって遊技に必要な遊技媒体が投入され、それに基づいて遊技が行われ、その遊技の結果に基づいて特典が付与される（例えば、メダルが払い出される）形態の全てに対して、本発明を適用することができる。すなわち、物理的な遊技者の動作によって遊技媒体が投入され（掛けられ）、遊技媒体が払い出される形態のみならず、主制御回路（主基板３１）自体が、遊技者が保有する遊技媒体を電磁的に管理し、メダルレスで遊技を可能にする形態であってもよい。なお、この場合、遊技者が保有する遊技媒体を電磁的に管理するのは、主制御回路（主基板３１）に装着され（接続され）且つ遊技媒体を管理する遊技媒体管理装置であってもよい。

【０３７４】

この場合、遊技媒体管理装置は、ＲＯＭ及びＲＷＭ（あるいは、ＲＡＭ）を有し、遊技機に設けられる装置であって、図示しない外部の遊技媒体取扱装置と所定のインターフェースを介して双方向通信可能に接続されるものであり、遊技媒体の貸出動作（すなわち、

遊技者が遊技媒体の投入操作を行う上で、必要な遊技媒体を提供する動作）或いは遊技媒体の払い出しに係る役に入賞（当該役が成立）した場合における遊技媒体の払出動作（すなわち、遊技者に対して遊技媒体の払い出しを行う上で、必要な遊技媒体を獲得させる動作）、又は、遊技の用に供する遊技媒体を電磁的に記録する動作を行い得るものとすればよい。また、遊技媒体管理装置は、実際の遊技媒体数の管理のみならず、例えば、その遊技媒体数の管理結果に基づいて、保有する遊技媒体数を表示する保有遊技媒体数表示装置（不図示）をパチスロ１の前面に設け、この保有遊技媒体数表示装置に表示される遊技媒体数を管理するものであってもよい。すなわち、遊技媒体管理装置は、遊技者が遊技の用に供することができる遊技媒体の総数を電磁的方法により記録し、表示することができるものとすればよい。

【０３７５】

また、この場合、遊技媒体管理装置は、遊技者が、記録された遊技媒体数を示す信号を、外部の遊技媒体取扱装置に対して自由に送信させることができる性能（機能）を有することが望ましい。また、遊技媒体管理装置は、遊技者が直接操作する場合以外の場合には、記録された遊技媒体数を減ずることができない性能を有することが望ましい。また、遊技媒体管理装置と外部の遊技媒体取扱装置との間に外部接続端子板（不図示）が設けられる場合には、遊技媒体管理装置は、その外部接続端子板を介してでなければ、遊技者が、記録された遊技媒体数を示す信号を送信できない性能を有することが望ましい。

【０３７６】

遊技機には、上記の他、遊技者が操作可能な貸出操作手段、返却（精算）操作手段、外部接続端子板が設けられ、遊技媒体取扱装置には、紙幣等の有価価値の投入口、記録媒体（例えばＩＣカード）の挿入口、携帯端末から電子マネー等の入金を行うための非接触通信アンテナ等、その他貸出操作手段、返却操作手段等の各種操作手段、遊技媒体取扱装置側外部接続端子板が設けられるようにしてもよい（いずれも不図示）。

【０３７７】

その際の遊技の流れとしては、例えば、遊技者が遊技媒体取扱装置に対し、上記いずれかの方法で有価価値を入金し、上記いずれかの貸出操作手段の操作に基づいて所定数の有価価値を減算し、遊技媒体取扱装置から遊技媒体管理装置に対し、減算した有価価値に対応する遊技媒体を増加させる。そして、遊技者は遊技を行い、さらに遊技媒体が必要な場合には上記操作を繰り返し行う。その後、遊技の結果、所定数の遊技媒体を獲得し、遊技を終了する際には、上記いずれかの返却操作手段を操作することにより遊技媒体管理装置から遊技媒体取扱装置に対し、遊技媒体数を送信し、遊技媒体取扱装置はその遊技媒体数を記録した記録媒体を排出する。また、遊技媒体管理装置は遊技媒体数を送信したときに、自身が記憶する遊技媒体数をクリアする。遊技者は排出された記録媒体を景品交換するために景品カウンター等に持って行くか、又は、記録された遊技媒体に基づいて他の遊技台で遊技を行うために遊技台を移動する。

【０３７８】

なお、上記例では、遊技媒体管理装置から全遊技媒体数を遊技媒体取扱装置に対して送信したが、遊技機又は遊技媒体取扱装置側で遊技者が所望する遊技媒体数のみを送信し、遊技者が所持する遊技媒体を分割して処理することとしてもよい。また、上記例では、遊技媒体取扱装置が記録媒体を排出することとしたが、現金又は現金等価物を排出するようにしてもよいし、携帯端末等に記憶させるようにしてもよい。また、遊技媒体取扱装置は遊技場の会員記録媒体を挿入可能とし、遊技媒体を会員記録媒体に貯留して、後日、該貯留された遊技媒体を用いて再遊技可能とするようにしてもよい。

【０３７９】

また、遊技機又は遊技媒体取扱装置において、図示しない所定の操作手段を操作することにより遊技媒体取扱装置又は遊技媒体管理装置に対し、遊技媒体又は有価価値のデータ通信をロックするロック操作を実行可能としてもよい。その際には、ワンタイムパスワード等の遊技者にしか知り得ない情報を設定することや遊技機又は遊技媒体取扱装置に設けられた撮像手段により遊技者を記憶するようにしてもよい。

【０３８０】
　なお、遊技媒体管理装置は、上述のように、メダルレスでのみ遊技を可能とするものであってもよいし、物理的な遊技者の動作によって遊技媒体が投入され（掛けられ）、遊技媒体が払い出される形態、及び、メダルレスで遊技を可能とする形態の両方の形態で遊技を可能とするものであってもよい。後者の場合には、遊技媒体管理装置が、上述のセレクタ３５やホッパー３３を直接的に制御する方式を採用することもできるし、これらが主制御回路（主基板３１）によって制御され、その制御結果が送信されることに基づいて、遊技者が遊技の用に供することができる遊技媒体の総数を電磁的方法により記録し且つ表示する制御を行い得る方式を採用することもできる。
【０３８１】
　また、上記例では、遊技媒体管理装置を、パチスロに適用する場合について説明しているが、例えば、遊技球を用いるスロットマシンや封入式遊技機においても同様に遊技媒体管理装置を設け、遊技者の遊技媒体が管理されるようにすることもできる。
【０３８２】
　上述した遊技媒体管理装置を設けた場合には、遊技媒体が物理的に遊技に供される場合に比べて、遊技機内部のセレクタ３５やホッパー３３などの装置を減らすことができ、遊技機の原価及び製造コストを削減できるのみならず、遊技者が直接遊技媒体に接触しないようにすることもでき、遊技環境が改善され、騒音も減らすことができるとともに、装置を減らしたことにより遊技機の消費電力を減らすことも可能になる。また、上述した遊技媒体管理装置を設けた場合には、遊技媒体や遊技媒体の投入口や払出口を介した不正行為を防止することができる。すなわち、上述した遊技媒体管理装置を設けた場合には、遊技機をとりまく種々の環境を改善可能な遊技機を提供することが可能になる。
【０３８３】
［発明の要旨］
＜要旨１＞
　ＬＥＤをドットマトリクス状に配置して表示ユニットを構成する遊技機が特開２００５－３２３７６８号公報に提案されている。
【０３８４】
　ＬＥＤをドットマトリクス状に配置したもののように、複数の発光体により構成された発光部の発光制御を行うためには、個々の発光体を直接に制御する方式と、制御データをＬＥＤドライバＩＣなどの発光駆動手段に送信することにより、発光駆動手段に個々の発光体を駆動させる方式とがある。
【０３８５】
　発光駆動手段の仕様は、発光駆動手段の製造メーカ、又は、同じ製造メーカにおけるシリーズによって異なることがある。このため、発光駆動手段を介して個々の発光体を制御する場合には、発光駆動手段に応じて、発光部の発光パターンを表すパターンデータを作成し直す必要がある。
【０３８６】
　本発明は、このような課題を解決するためになされたもので、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することを目的とする。
【０３８７】
　本発明に係る遊技機は、
　複数の発光体（ＬＥＤ）により構成された発光部（第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９）と、
　前記発光部を駆動するための発光駆動手段（ドライバＩＣ）と、
　前記発光部の発光パターンを制御するための制御部（サブＣＰＵ８１）と、
　前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブＲＯＭ８２）と、を備え、
　前記制御部は、

前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発光パターンに応じたパターンデータを決定し、
決定した前記パターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じた輝度値に変換し、
変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信する
構成を有している。
【０３８８】
この構成により、本発明に係る遊技機は、実行する発光パターンに応じたパターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。
【０３８９】
なお、本発明に係る遊技機において、
前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、６ビット周期のデータに対して９ビット周期）でエラー検出用データを挿入するようにしてもよい。
【０３９０】
この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。
【０３９１】
本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。
【０３９２】
＜要旨２＞
要旨１と同様な課題を解決するため、本発明に係る遊技機は、
複数の発光体（ＬＥＤ）により構成された発光部（第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９）と、
前記発光部を駆動するための発光駆動手段（ドライバＩＣ）と、
前記発光部の発光パターンを制御するための制御部（サブＣＰＵ８１）と、
前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブＲＯＭ８２）と、を備え、
前記制御部は、
前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発光パターンに応じたパターンデータを決定し、
決定した前記パターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じた輝度値に変換し、
変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信し、
前記制御データは、
前記発光駆動手段によって駆動される全ての発光体の輝度を設定する第１モードと、
前記発光駆動手段に駆動させる発光体を特定するチャネル番号及び前記チャネル番号によって特定される発光体の輝度を設定する第２モードと、のいずれのモードでも送信可能である
構成を有している。
【０３９３】
この構成により、本発明に係る遊技機は、実行する発光パターンに応じたパターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に

基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

【０３９４】

また、本発明に係る遊技機は、発光駆動手段によって駆動される全ての発光体の輝度を設定する第１モードと、発光駆動手段に駆動させる発光体を特定するチャネル番号及びチャネル番号によって特定される発光体の輝度を設定する第２モードと、のいずれのモードでも制御データを送信することができる。

【０３９５】

このため、発光体を個別に制御する場合には、特定の発光体の輝度を設定する第２モードの方が、第１モードよりも制御データのデータ長が短くなり、発光体を全体的に制御する場合には、発光体を特定するチャネル番号を必要としない第１モードの方が、第２モードよりも制御データのデータ長が短くなる。

【０３９６】

したがって、本発明に係る遊技機は、発光駆動手段が駆動する発光体の用途に応じて、データ長が短くなる制御データのモードを選択することにより、制御部と発光駆動手段との間の伝送負荷を低減させることができ、発光体の応答性を向上させることができる。

【０３９７】

なお、本発明に係る遊技機において、

前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、６ビット周期のデータに対して９ビット周期）でエラー検出用データを挿入するようにしてもよい。

【０３９８】

この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。

【０３９９】

本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。

【０４００】

＜要旨３＞

要旨１と同様な課題を解決するため、本発明に係る遊技機は、

複数の発光体（ＬＥＤ）により構成された発光部（第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９）と、

前記発光部を駆動するための発光駆動手段（ドライバＩＣ）と、

前記発光部の発光パターンを制御するための制御部（サブＣＰＵ８１）と、

前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブＲＯＭ８２）と、を備え、

前記制御部は、

前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発光パターンに応じたパターンデータを決定し、

決定した前記パターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じた輝度値に変換し、

変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信し、

前記制御データは、

前記発光駆動手段によって駆動される全ての発光体の輝度を設定する第１モードと、

前記発光駆動手段に駆動させる発光体を特定するチャネル番号及び前記チャネル番号によって特定される発光体の輝度を設定する第２モードと、のいずれのモードでも送信可能であり、

前記複数の発光体は、
マトリクス状に配置された第１発光体群と、
前記第１発光体群に含まれない第２発光体群と、を含み、
前記制御部は、
前記第１発光体群の発光体を駆動する発光駆動手段には、前記制御データを前記第２モードで送信し、
前記第２発光体群の発光体を駆動する発光駆動手段には、前記制御データを前記第１モードで送信する
構成を有している。

【０４０１】
この構成により、本発明に係る遊技機は、実行する発光パターンに応じたパターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

【０４０２】
また、本発明に係る遊技機は、発光駆動手段によって駆動される全ての発光体の輝度を設定する第１モードと、発光駆動手段に駆動させる発光体を特定するチャネル番号及びチャネル番号によって特定される発光体の輝度を設定する第２モードと、のいずれのモードでも制御データを送信することができる。

【０４０３】
このため、マトリクス状に配置された第１発光体群に対しては、発光体を個別に制御するため、特定の発光体の輝度を設定する第２モードの方が、第１モードよりも制御データのデータ長が短くなる。

【０４０４】
一方、第１発光体群に含まれない第２発光体群に対しては、発光体を全体的に制御するため、発光体を特定するチャネル番号を必要としない第１モードの方が、第２モードよりも制御データのデータ長が短くなる。

【０４０５】
したがって、本発明に係る遊技機は、発光駆動手段が駆動する発光体の用途に応じて、データ長が短くなる制御データのモードを選択することにより、制御部と発光駆動手段との間の伝送負荷を低減させることができ、発光体の応答性を向上させることができる。

【０４０６】
なお、本発明に係る遊技機において、
前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、６ビット周期のデータに対して９ビット周期）でエラー検出用データを挿入するようにしてもよい。

【０４０７】
この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。

【０４０８】
本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。

【０４０９】
＜要旨４＞
要旨１と同様な課題を解決するため、本発明に係る遊技機は、
複数の発光体（ＬＥＤ）により構成された発光部（第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９）と、
前記発光部を駆動するための発光駆動手段（ドライバＩＣ）と、

前記発光部の発光パターンを制御するための制御部（サブＣＰＵ８１）と、
前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブＲＯＭ８２）と、を備え、
前記制御部は、
前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発光パターンに応じたパターンデータを決定し、
決定した前記パターンデータを前記発光駆動手段に応じて変換し、
変換した前記パターンデータに基づく制御データを前記発光駆動手段にシリアルバス通信により送信する
構成を有している。

【０４１０】
この構成により、本発明に係る遊技機は、実行する前記発光パターンに応じたパターンデータを発光駆動手段に応じて変換し、変換したパターンデータに基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

【０４１１】
なお、本発明に係る遊技機において、
前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、６ビット周期のデータに対して９ビット周期）でエラー検出用データを挿入するようにしてもよい。

【０４１２】
この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。

【０４１３】
本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。

【０４１４】
＜要旨５＞
要旨１と同様な課題を解決するため、本発明に係る遊技機は、
複数の発光体（ＬＥＤ）により構成された発光部（第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９）と、
前記発光部を駆動するための発光駆動手段（ドライバＩＣ）と、
前記発光部の発光パターンを制御するための制御部（サブＣＰＵ８１）と、
前記発光部の前記発光パターンを表す複数のパターンデータが記憶されたパターンデータ記憶手段（サブＲＯＭ８２）と、を備え、
前記制御部は、
前記パターンデータ記憶手段に記憶された複数のパターンデータから、実行する前記発光パターンに応じたパターンデータを決定し、
決定した前記パターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じた輝度値に変換し、
変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信し、
前記制御データは、
前記発光駆動手段によって駆動される全ての発光体の輝度を設定する第１モードと、
前記発光駆動手段に駆動させる発光体を特定するチャネル番号及び前記チャネル番号によって特定される発光体の輝度を設定する第２モードと、
前記発光駆動手段によって駆動される全ての発光体を消灯させるコマンドを含む第３モードと、のいずれのモードでも送信可能である

構成を有している。

【０４１５】

この構成により、本発明に係る遊技機は、実行する発光パターンに応じたパターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

【０４１６】

また、本発明に係る遊技機は、発光駆動手段によって駆動される全ての発光体の輝度を設定する第１モードと、発光駆動手段に駆動させる発光体を特定するチャネル番号及びチャネル番号によって特定される発光体の輝度を設定する第２モードと、のいずれのモードでも制御データを送信することができる。

【０４１７】

このため、発光体を個別に制御する場合には、特定の発光体の輝度を設定する第２モードの方が、第１モードよりも制御データのデータ長が短くなり、発光体を全体的に制御する場合には、発光体を特定するチャネル番号を必要としない第１モードの方が、第２モードよりも制御データのデータ長が短くなる。

【０４１８】

また、全ての発光体を消灯させる場合には、各発光体の輝度値の指定や発光体を特定するチャネル番号を必要としない第３モードの方が、第１モード及び第２モードよりもデータ長が短くなる。

【０４１９】

したがって、本発明に係る遊技機は、発光駆動手段が駆動する発光体の用途に応じて、データ長が短くなる制御データのモードを選択することにより、制御部と発光駆動手段との間の伝送負荷を低減させることができ、発光体の応答性を向上させることができる。

【０４２０】

なお、本発明に係る遊技機において、

前記制御部は、前記制御データに対して、前記発光体の輝度を設定するためのデータと同期しない周期（例えば、６ビット周期のデータに対して９ビット周期）でエラー検出用データを挿入するようにしてもよい。

【０４２１】

この構成により、本発明に係る遊技機は、制御データにおいて、発光体の輝度を設定するためのデータに、エラー検出用データを同期させないことにより、データパターンの影響を受けることなく、シリアルバス通信の異常を制御データの受信側で検出させることができる。

【０４２２】

本発明によれば、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる遊技機を提供することができる。

【０４２３】

＜要旨６＞

演出制御ＣＰＵとＲＴＣ（Real Time Clock）とがＩ２Ｃ（Inter-Integrated Circuit）によるシリアルバス通信で接続された遊技機が特開２０１７－５１３８５号公報に提案されている。

【０４２４】

上述したような従来の遊技機は、ＲＴＣから日時データを取得するために、Ｉ２Ｃなどによるシリアル通信を行う必要があり、演出制御ＣＰＵなどの演算処理手段に負荷がかかることがある。

【０４２５】

本発明は、このような課題を解決するためになされたもので、演算処理手段にかかる負荷を低減させることができる遊技機を提供することを目的とする。

【０４２６】

本発明に係る遊技機は、
演算処理を行う演算処理手段（サブＣＰＵ８１）と、
前記演算処理手段を動作させるためのクロックを発生するクロック発生手段（発振回路８４）と、
前記演算処理手段が使用するためのデータを記憶できる演算記憶手段（サブＲＡＭ８３）と、
前記演算処理手段にシリアル通信で接続され日時を計時する計時手段（ＲＴＣ８６）と、を備え、
前記演算処理手段は、
前記計時手段から日時データを取得して前記演算記憶手段に記憶する日時取得手段と、
前記クロック発生手段によって発生されたクロックをカウントするカウント手段（クロックカウンタ）と、
前記カウント手段によってカウントされた値に基づいて経過時間を算出する経過時間算出手段と、
前記経過時間算出手段によって算出された経過時間が規定の経過時間に達した場合に、前記経過時間算出手段によって算出された経過時間に応じた時間分、前記演算記憶手段に記憶された前記日時データを更新する日時データ更新手段と、を有する
構成を有している。

【０４２７】
この構成により、本発明に係る遊技機は、演算処理手段に動作用のクロックをカウントさせることによって日時データを管理させる。したがって、演算処理手段は、日時データを取得するたびに、計時手段とシリアル通信を行う必要がなくなる。このように、本発明に係る遊技機は、演算処理手段にかかる負荷を低減させることができる。

【０４２８】
なお、本発明に係る遊技機において、前記日時データ更新手段は、前記経過時間から規定の経過時間を減算し、減算結果が規定の経過時間未満となるまで前記日時データを更新する処理を繰り返すようにしてもよい。

【０４２９】
本発明によれば、演算処理手段にかかる負荷を低減させることができる遊技機を提供することができる。

【０４３０】
＜要旨７＞
要旨６と同様な課題を解決するため、本発明に係る遊技機は、
演算処理を行う演算処理手段（サブＣＰＵ８１）と、
前記演算処理手段を動作させるためのクロックを発生するクロック発生手段（発振回路８４）と、
前記演算処理手段が使用するためのデータを記憶できる演算記憶手段（サブＲＡＭ８３）と、
前記演算処理手段にシリアル通信で接続され日時を計時する計時手段（ＲＴＣ８６）と、
閏年であるときの日数と閏年でないときの日数とが月ごとに対応付けられている月別日テーブルを記憶する月別日テーブル記憶手段（サブＲＯＭ８２）と、を備え、
前記演算処理手段は、
前記計時手段から日時データを取得して前記演算記憶手段に記憶する日時取得手段と、
前記クロック発生手段によって発生されたクロックをカウントするカウント手段と、
前記カウント手段によってカウントされた値に基づいて経過時間を算出する経過時間算出手段と、
前記経過時間算出手段によって算出された経過時間が規定の経過時間に達した場合に、前記経過時間算出手段によって算出された経過時間に応じた時間分、前記演算記憶手段に記憶された前記日時データを更新する日時データ更新手段と、を有し、
前記日時データ更新手段は、前記月別日テーブルを参照して前記日時データを更新する

構成を有している。

【０４３１】

この構成により、本発明に係る遊技機は、演算処理手段に動作用のクロックをカウントさせることによって日時データを管理させる。したがって、演算処理手段は、日時データを取得するたびに、計時手段とシリアル通信を行う必要がなくなる。このように、本発明に係る遊技機は、演算処理手段にかかる負荷を低減させることができる。

【０４３２】

また、本発明に係る遊技機は、閏年であるときの日数と閏年でないときの日数とが月ごとに対応付けられている月別日テーブルを参照して日時データを更新するため、日時データを誤りなく更新することができる。

【０４３３】

なお、本発明に係る遊技機において、前記日時データ更新手段は、前記経過時間から規定の経過時間を減算し、減算結果が規定の経過時間未満となるまで前記日時データを更新する処理を繰り返すようにしてもよい。

【０４３４】

本発明によれば、演算処理手段にかかる負荷を低減させることができる遊技機を提供することができる。

【０４３５】

＜要旨８＞

要旨６と同様な課題を解決するため、本発明に係る遊技機は、

演算処理を行う演算処理手段（サブＣＰＵ８１）と、

前記演算処理手段を動作させるためのクロックを発生するクロック発生手段（発振回路８４）と、

前記演算処理手段が使用するためのデータを記憶できる演算記憶手段（サブＲＡＭ８３）と、

前記演算処理手段にシリアル通信で接続され日時を計時する計時手段（ＲＴＣ８６）と、を備え、

前記演算処理手段は、

前記計時手段から日時データを取得して前記演算記憶手段に記憶する日時記憶手段と、

前記クロック発生手段によって発生されたクロックをカウントするカウント手段と、

前記カウント手段によってカウントされた値に基づいて経過時間を算出する経過時間算出手段と、

前記経過時間算出手段によって算出された経過時間が規定の経過時間に達した場合に、前記経過時間算出手段によって算出された経過時間に応じた時間分、前記演算記憶手段に記憶された前記日時データを更新する日時データ更新手段と、を有し、

前記日時データ更新手段は、前記演算記憶手段に記憶された前記日時データが表す日時に前記経過時間算出手段によって算出された経過時間が加算されるまで、前記演算記憶手段に記憶された前記日時データに所定時間を加算する

構成を有している。

【０４３６】

この構成により、本発明に係る遊技機は、演算処理手段に動作用のクロックをカウントさせることによって日時データを管理させる。したがって、演算処理手段は、日時データを取得するたびに、計時手段とシリアル通信を行う必要がなくなる。このように、本発明に係る遊技機は、演算処理手段にかかる負荷を低減させることができる。

【０４３７】

また、本発明に係る遊技機は、演算記憶手段に記憶された日時データが表す日時に経過時間算出手段によって算出された経過時間が加算されるまで、演算記憶手段に記憶された日時データに所定時間を加算することによって、処理負荷が高い除算処理を行わずに加算処理だけで演算記憶手段に記憶された日時データを更新するため、演算処理手段にかかる負荷を低減させることができる。

【０４３８】
　なお、本発明に係る遊技機において、前記日時データ更新手段は、前記経過時間から規定の経過時間を減算し、減算結果が規定の経過時間未満となるまで前記日時データを更新する処理を繰り返すようにしてもよい。
【０４３９】
　本発明によれば、演算処理手段にかかる負荷を低減させることができる遊技機を提供することができる。
【０４４０】
＜要旨９＞
　ＲＯＭに格納されたプログラムをＣＰＵに実行させる遊技機が特開２０００－２９６２２３号公報に提案されている。また、ＲＯＭに格納されたプログラムを起動時のブート処理によりＲＡＭに転送して、ＲＡＭに転送したプログラムをＣＰＵに実行させる遊技機が特開２００８－１４８８９１号公報に提案されている。
【０４４１】
　ＲＯＭに格納されたプログラムをＣＰＵなどの演算処理手段に実行させる従来の遊技機は、プログラムを実行するたびに演算処理手段がＲＯＭにアクセスするため、ＲＯＭのアクセス速度がボトルネックとなり、演算処理手段の処理速度を低下させていた。一方、ＲＡＭに転送したプログラムをＣＰＵなどの演算処理手段に実行させる従来の遊技機は、必要なＲＡＭの容量が増加するため、コストがかかってしまう。
【０４４２】
　本発明は、このような課題を解決するためになされたもので、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することを目的とする。
【０４４３】
　本発明に係る遊技機は、
　演算処理を行う演算処理手段（サブＣＰＵ８１）と、
　前記演算処理手段を動作させるためのプログラム及びデータが記憶された第１記憶手段（サブＲＯＭ８２）と、
　前記第１記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第２記憶手段（サブＲＡＭ８３）と、を備え、
　前記演算処理手段は、
　前記第１記憶手段に記憶されたプログラムのうちの所定のプログラムを前記第２記憶手段に転送し、
　前記所定のプログラムに対する上位プログラムから前記第２記憶手段に転送した前記所定のプログラムを実行する
　構成を有している。
【０４４４】
　この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラムを第２記憶手段に転送して実行するため、演算処理手段の処理速度を向上させることができる。
【０４４５】
　また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第１記憶手段に記憶された状態で実行することによって、第２記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。
【０４４６】
　なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第２記憶手段に転送する際には、第１の引数を前記第１記憶手段に記憶された所定のプログラムの先頭アドレスとし、第２の引数を前記第２記憶手段の転送先の先頭アドレスとし、第３の引数を前記第１記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。
【０４４７】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

【０４４８】

＜要旨１０＞

要旨９と同様な課題を解決するため、本発明に係る遊技機は、

演算処理を行う演算処理手段（サブＣＰＵ８１）と、

前記演算処理手段を動作させるためのプログラム及びデータが記憶された第１記憶手段（サブＲＯＭ８２）と、

前記第１記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第２記憶手段（サブＲＡＭ８３）と、を備え、

前記演算処理手段は、

前記第１記憶手段に記憶されたプログラムのうちの所定のプログラムを実行することを契機として前記所定のプログラムを前記第２記憶手段に転送し、

前記所定のプログラムに対する上位プログラムから前記第２記憶手段に転送した前記所定のプログラムを実行する

構成を有している。

【０４４９】

この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラムを実行することを契機として所定のプログラムを第２記憶手段に転送して実行することによって、演算処理手段の処理速度を向上させることができる。

【０４５０】

また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第１記憶手段に記憶された状態で実行することによって、第２記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。

【０４５１】

なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第２記憶手段に転送する際には、第１の引数を前記第１記憶手段に記憶された所定のプログラムの先頭アドレスとし、第２の引数を前記第２記憶手段の転送先の先頭アドレスとし、第３の引数を前記第１記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。

【０４５２】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

【０４５３】

＜要旨１１＞

要旨９と同様な課題を解決するため、本発明に係る遊技機は、

演算処理を行う演算処理手段（サブＣＰＵ８１）と、

前記演算処理手段を動作させるためのプログラム及びデータが記憶された第１記憶手段（サブＲＯＭ８２）と、

前記第１記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第２記憶手段（サブＲＡＭ８３）と、を備え、

前記演算処理手段は、

前記第１記憶手段に記憶されたプログラムのうちの所定のプログラムを前記第２記憶手段に予め転送しておき、

前記所定のプログラムに対する上位プログラムから前記第２記憶手段に転送した前記所定のプログラムを実行する

構成を有している。

【０４５４】

この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラムを第２記憶手段に予め転送しておいて実行することによって、演算処理手

段の処理速度を向上させることができる。
【０４５５】
　また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第１記憶手段に記憶された状態で実行することによって、第２記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。
【０４５６】
　なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第２記憶手段に転送する際には、第１の引数を前記第１記憶手段に記憶された所定のプログラムの先頭アドレスとし、第２の引数を前記第２記憶手段の転送先の先頭アドレスとし、第３の引数を前記第１記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。
【０４５７】
　本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。
【０４５８】
＜要旨１２＞
　要旨９と同様な課題を解決するため、本発明に係る遊技機は、
　演算処理を行う演算処理手段（サブＣＰＵ８１）と、
　前記演算処理手段を動作させるためのプログラム及びデータが記憶された第１記憶手段（サブＲＯＭ８２）と、
　前記第１記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第２記憶手段（サブＲＡＭ８３）と、を備え、
　前記演算処理手段は、
　前記第１記憶手段に記憶されたプログラムのうちの所定のプログラムを実行する場合、前記所定のプログラムが前記第２記憶手段に転送されていなければ、前記所定のプログラムを前記第２記憶手段に転送し、
　前記所定のプログラムに対する上位プログラムから前記第２記憶手段に転送した前記所定のプログラムを実行する
　構成を有している。
【０４５９】
　この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラム実行する場合、所定のプログラムが第２記憶手段に転送されていなければ、所定のプログラムを第２記憶手段に転送して実行することによって、演算処理手段の処理速度を向上させることができる。
【０４６０】
　また、本発明に係る遊技機は、所定のプログラムを実行する場合、所定のプログラムが第２記憶手段に転送されていれば、第２記憶手段に転送済みの所定のプログラムを実行するため、所定のプログラムを第２記憶手段に転送する演算処理手段の処理を省くことができる。
【０４６１】
　また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第１記憶手段に記憶された状態で実行することによって、第２記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。
【０４６２】
　なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第２記憶手段に転送する際には、第１の引数を前記第１記憶手段に記憶された所定のプログラムの先頭アドレスとし、第２の引数を前記第２記憶手段の転送先の先頭アドレスとし、第３の引数を前記第１記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。
【０４６３】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

【０４６４】

＜要旨１３＞

要旨９と同様な課題を解決するため、本発明に係る遊技機は、

演算処理を行う演算処理手段（サブＣＰＵ８１）と、

前記演算処理手段を動作させるためのプログラム及びデータが記憶された第１記憶手段（サブＲＯＭ８２）と、

前記第１記憶手段よりもアクセス速度が高く、かつ、書き換え可能な第２記憶手段（サブＲＡＭ８３）と、を備え、

前記演算処理手段は、

前記第１記憶手段に記憶されたプログラムのうちの所定のプログラムを転送する領域が前記第２記憶手段に確保されていれば前記所定のプログラムを前記第２記憶手段に転送し、

前記所定のプログラムを転送する領域が前記第２記憶手段に確保されていなければ前記第２記憶手段に前記所定のプログラムを転送する領域を確保し、確保した領域に前記所定のプログラムを転送し、

前記所定のプログラムに対する上位プログラムから前記第２記憶手段に転送した前記所定のプログラムを実行する

構成を有している。

【０４６５】

この構成により、本発明に係る遊技機は、所定のプログラムとして相対的に処理負荷が高いプログラムを第２記憶手段に転送して実行することによって、演算処理手段の処理速度を向上させることができる。

【０４６６】

また、本発明に係る遊技機は、所定のプログラムを転送する領域が第２記憶手段に確保されていなければ第２記憶手段に所定のプログラムを転送する領域を確保し、確保した領域に所定のプログラムを転送して実行するため、演算処理手段の処理速度を確実に向上させることができる。

【０４６７】

また、本発明に係る遊技機は、所定のプログラム以外のプログラムを第１記憶手段に記憶された状態で実行することによって、第２記憶手段に必要な記憶容量を抑制するため、コストを抑制することができる。

【０４６８】

なお、本発明に係る遊技機において、前記演算処理手段は、所定のプログラムを前記第２記憶手段に転送する際には、第１の引数を前記第１記憶手段に記憶された所定のプログラムの先頭アドレスとし、第２の引数を前記第２記憶手段の転送先の先頭アドレスとし、第３の引数を前記第１記憶手段に記憶された所定のプログラムのプログラム容量としてもよい。

【０４６９】

本発明によれば、コストをかけずに演算処理手段の処理速度を向上させることができる遊技機を提供することができる。

【０４７０】

＜要旨１４＞

ＬＥＤをドットマトリクス状に配置して表示ユニットを構成する遊技機が特開２００５－３２３７６８号公報に提案されている。

【０４７１】

ＬＥＤをドットマトリクス状に配置したもののように、複数の発光体により構成された発光部の発光制御を行うためには、全ての発光パターンをＲＯＭ等に記憶しておき、発光パターンを適宜更新する。このため、演出内容に応じた大量の発光パターンを用意しておく必要があり、その確認及び変更が開発者並びに確認者の負担となる。

【０４７２】
本発明は、このような課題を解決するためになされたもので、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することを目的とする。
【０４７３】
本発明に係る遊技機は、
複数の発光体（ＬＥＤ）により構成された発光部（第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９）と、
前記発光部の発光パターンを制御するための制御部（サブＣＰＵ８１）と、
前記発光部の前記発光パターンを制御するための複数の発光データ及び複数のパーツデータが記憶されたパターンデータ記憶手段（サブＲＯＭ８２）と、を備え、
前記パーツデータは、
前記パーツデータを識別するための識別情報と、
前記発光部の発光パターンを表すパターンデータと、
前記パーツデータの属性を示す属性データと、を含み、
前記発光データは、前記パーツデータの識別情報の順序を表し、
前記制御部は、
前記パターンデータ記憶手段に記憶された複数の発光データから、発光データを決定し、決定した発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、
前記発光データにおける最後の識別情報に対応するパーツデータに含まれるパターンデータで前記発光部を制御した後、前記発光データにおける最後の識別情報に対応するパーツデータに含まれる前記属性データが連続再生を表すことを条件として、前記発光データが表す識別情報の先頭から前記属性データが連続再生を表すパーツデータを検索し、検出したパーツデータから前記発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御する
構成を有している。
【０４７４】
この構成により、本発明に係る遊技機は、パーツデータを組み合せることによって、発光部に多様な演出を実行させるため、パーツデータ単位で発光部の発光パターンの確認及び変更を行わせることができる。このように、本発明に係る遊技機は、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。
【０４７５】
また、本発明に係る遊技機は、属性データが連続再生に設定されているパーツデータによって発光部に演出を繰り返し実行させるため、発光部による演出を長時間にわたって実行させる場合であっても、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。
【０４７６】
本発明によれば、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することができる。
【０４７７】
＜要旨１５＞
要旨１４と同様な課題を解決するため、本発明に係る遊技機は、
複数の発光体（ＬＥＤ）により構成された発光部（第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９）と、
前記発光部の発光パターンを制御するための制御部（サブＣＰＵ８１）と
前記発光部の発光パターンを制御するための複数の発光データ及び複数のパーツデータが記憶された発光データ記憶手段（サブＲＯＭ８２）と、を備えた遊技機であって、
前記パーツデータは、
前記パーツデータを識別するための識別情報と、
前記発光部の発光パターンを表すパターンデータと、

前記パーツデータの属性を示す属性データと、を含み、
前記発光データは、前記パーツデータの識別情報の順序を表し、
前記制御部は、
前記発光データ記憶手段に記憶された複数の発光データから、発光データを決定し、決定した発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、
前記発光データにおける最後の識別情報に対応するパーツデータに含まれるパターンデータで前記発光部を制御した後、前記発光データにおける最後の識別情報に対応するパーツデータに含まれる前記属性データが連続再生を表すことを条件として、前記発光データが表す識別情報の先頭から前記属性データが連続再生を表すパーツデータを検索し、検出したパーツデータから前記発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御し、
前記遊技機がエラー状態から復帰した場合、前記遊技機が前記エラー状態となったときに実行していたエラー状態データが表すパーツデータの識別情報の順序から前記属性データが前記連続再生を表していないパーツデータの識別情報が除かれた発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御する
構成を有している。

【０４７８】
この構成により、本発明に係る遊技機は、パーツデータを組み合せることによって、発光部に多様な演出を実行させるため、パーツデータ単位で発光部の発光パターンの確認及び変更を行わせることができる。このように、本発明に係る遊技機は、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【０４７９】
また、本発明に係る遊技機は、属性データが連続再生に設定されているパーツデータによって発光部に演出を繰り返し実行させるため、発光部による演出を長時間にわたって実行させる場合であっても、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【０４８０】
また、本発明に係る遊技機は、エラー状態から復帰した場合、属性データが連続再生を表していないパーツデータの識別情報が除かれた新たな演出データが表す識別情報の順序にしたがってパーツデータに含まれるパターンデータで発光部を制御するため、エラー復帰時に演出データを最初から再生することにより遊技状態にそぐわない演出を実行してしまうことを防止することができる。

【０４８１】
本発明によれば、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することができる。

【０４８２】
＜要旨１６＞
要旨１４と同様な課題を解決するため、本発明に係る遊技機は、
複数の発光体（ＬＥＤ）により構成された発光部（第１ランプ群１１１、第７ランプ群１１７、第８ランプ群１１８、ドットマトリクス部１１９）と、
前記発光部を駆動するための発光駆動手段（ＬＥＤドライバＩＣ）と、
前記発光部の発光パターンを制御するための制御部（サブＣＰＵ８１）と、
前記発光部の前記発光パターンを制御するための複数の発光データ及び複数のパーツデータが記憶された発光データ記憶手段（サブＲＯＭ８２）と、を備え、
前記パーツデータは、
前記パーツデータを識別するための識別情報と、
前記発光部の発光パターンを表すパターンデータと、
前記パーツデータの属性を示す属性データと、を含み、

前記発光データは、前記パーツデータの識別情報の順序を表し、
前記制御部は、
前記発光データ記憶手段に記憶された複数の発光データから、発光データを決定し、決定した発光データが表す識別情報の順序にしたがって前記パーツデータのパターンデータに含まれる前記発光体の輝度値を前記発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを前記発光駆動手段にシリアルバス通信により送信することによって前記発光部を制御し、
前記発光データにおける最後の識別情報に対応するパーツデータに含まれるパターンデータで前記発光部を制御した後、前記発光データにおける最後の識別情報に対応するパーツデータに含まれる前記属性データが連続再生を表すことを条件として、前記発光データが表す識別情報の先頭から前記属性データが連続再生を表すパーツデータを検索し、検出したパーツデータから前記発光データが表す識別情報の順序にしたがって前記パーツデータに含まれるパターンデータで前記発光部を制御する
構成を有している。

【０４８３】
この構成により、本発明に係る遊技機は、パーツデータを組み合せることによって、発光部に多様な演出を実行させるため、パーツデータ単位で発光部の発光パターンの確認及び変更を行わせることができる。このように、本発明に係る遊技機は、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【０４８４】
また、本発明に係る遊技機は、属性データが連続再生に設定されているパーツデータによって発光部に演出を繰り返し実行させるため、発光部による演出を長時間にわたって実行させる場合であっても、発光部の発光パターンの確認及び変更にかかる負担を低減することができる。

【０４８５】
また、本発明に係る遊技機は、パターンデータに含まれる発光体の輝度値を発光駆動手段に応じた輝度値に変換し、変換した輝度値に基づく制御データを発光駆動手段に送信することにより発光部を制御するため、発光駆動手段の仕様に依存しないパターンデータによって発光部を制御することができる。

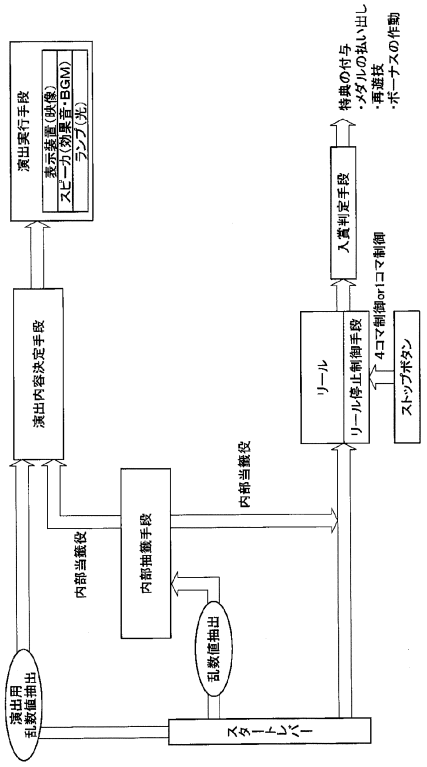
【０４８６】
本発明によれば、発光部の発光パターンの確認及び変更にかかる負担を低減することができる遊技機を提供することができる。

【符号の説明】

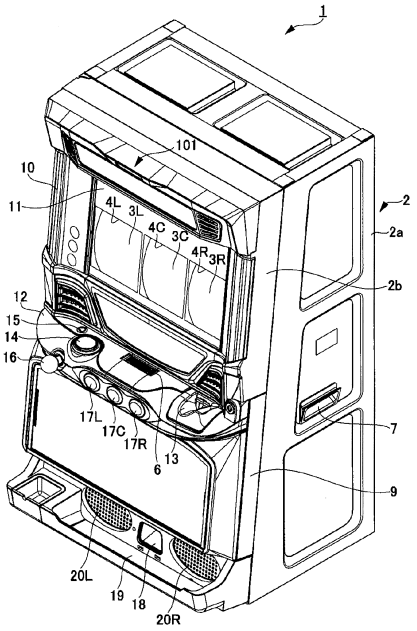
【０４８７】
１　パチスロ（遊技機）
８１　サブＣＰＵ（制御部、演算処理手段、日時取得手段、カウント手段、経過時間算出手段、日時データ更新手段）
８２　サブＲＯＭ（パターンデータ記憶手段、月別日テーブル記憶手段、第１記憶手段、演出データ記憶手段）
８３　サブＲＡＭ８３（演算記憶手段、第２記憶手段）
８４　発振回路（クロック発生手段）
８６　ＲＴＣ（計時手段）
１１１　第１ランプ群（発光部、第２発光体群）
１１７　第７ランプ群（発光部、第２発光体群）
１１８　第８ランプ群（発光部、第２発光体群）
１１９　ドットマトリクス部（発光部、第１発光体群）
１２１ａ～１２１ｂ、１２７ａ～１２７ｄ、１２８ａ～１２８ｃ、１２９ａ～１２９ｇ　ドライバＩＣ（発光駆動手段）

【図面】

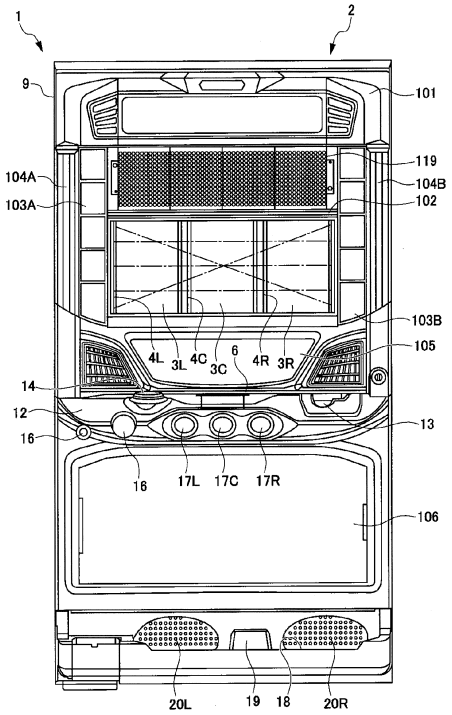
【図１】


演出実行手段
表示装置（映像）
スピーカ（効果音・BGM）
ランプ（光）
特典の付与
・メダルの払い出し
・再遊技
・ボーナスの作動
入賞判定手段
演出内容決定手段
リール
リール停止制御手段
4コマ制御or1コマ制御
ストップボタン
内部当籤役
内部抽籤手段
内部当籤役
乱数値抽出
演出用
乱数値抽出
スタートレバー

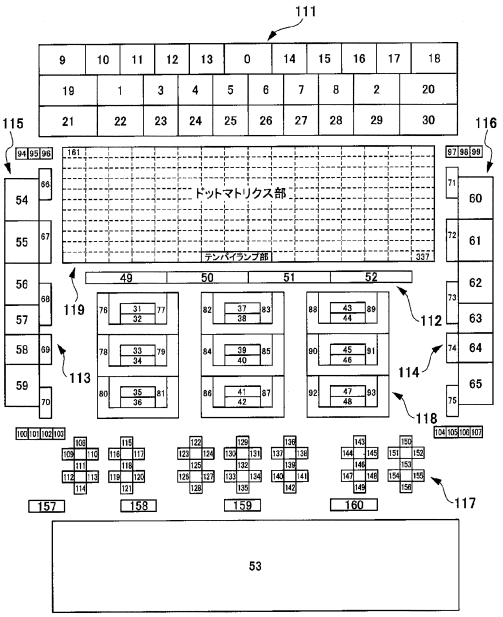
【図２】


1
101
10
11
4L
3L
4C
3C
4R
3R
2
2a
2b
12
15
14
16
17L
17C
17R
13
6
7
9
20L
19
18
20R

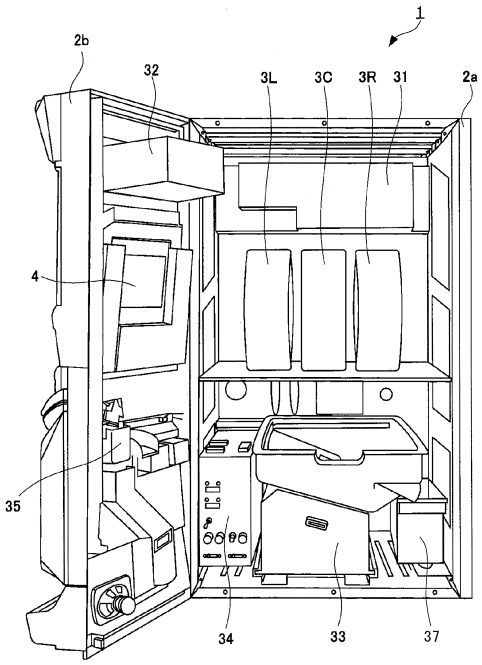
【図３】


1
2
9
101
119
104A
104B
103A
102
103B
105
4L
3L
4C
3C
6
4R
3R
14
12
16
13
16
17L
17C
17R
106
20L
19
18
20R

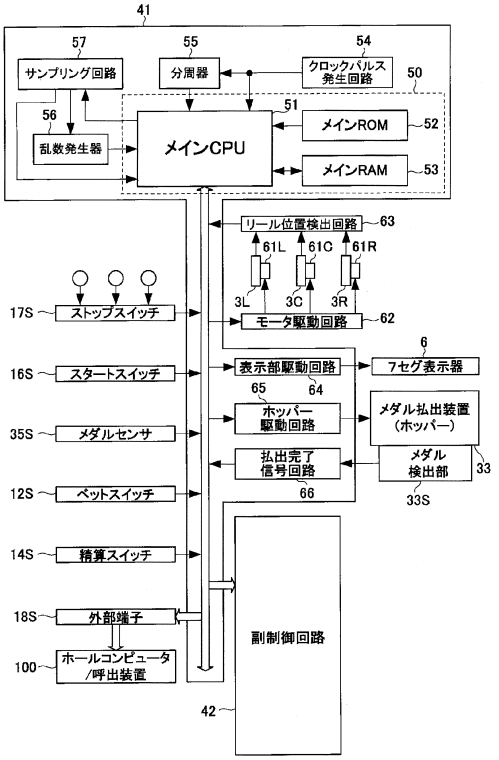
【図４】


111
ドットマトリクス部
テンパイランプ部
119
112
113
114
115
116
118
117
53

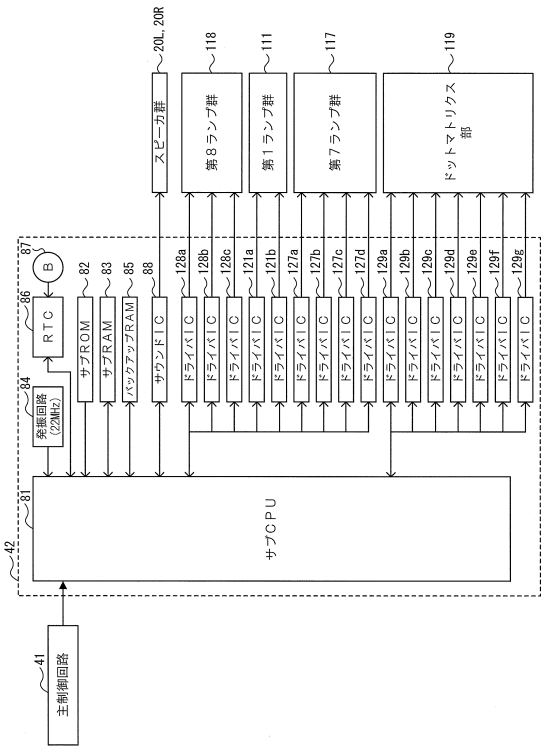
【図５】


1
2b
32
3L
3C
3R
31
2a
4
35
34
33
37

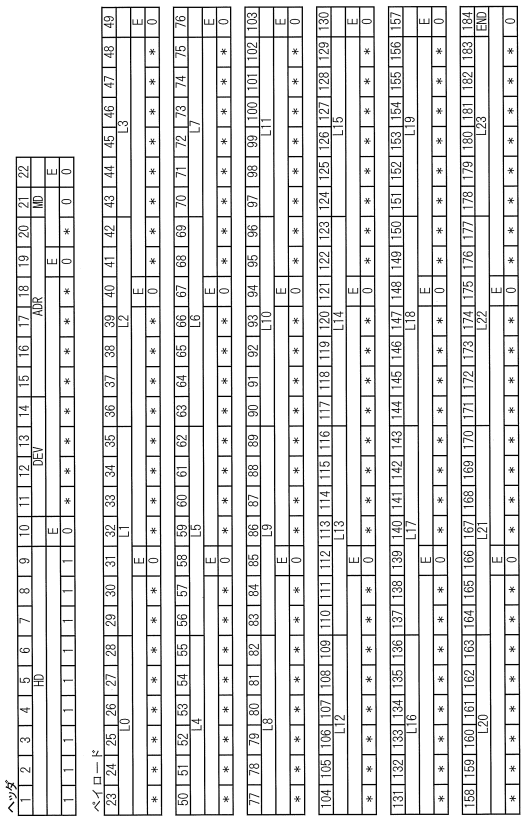
【図６】


41
57
サンプリング回路
55
分周器
54
クロックパルス発生回路
50
56
乱数発生器
51
メインCPU
メインROM
52
メインRAM
53
リール位置検出回路
63
61L
61C
61R
3L
3C
3R
17S
ストップスイッチ
モータ駆動回路
62
6
16S
スタートスイッチ
表示部駆動回路
7セグ表示器
65
64
ホッパー駆動回路
メダル払出装置（ホッパー）
35S
メダルセンサ
払出完了信号回路
メダル検出部
33
12S
ベットスイッチ
66
33S
14S
精算スイッチ
副制御回路
18S
外部端子
100
ホールコンピュータ/呼出装置
42

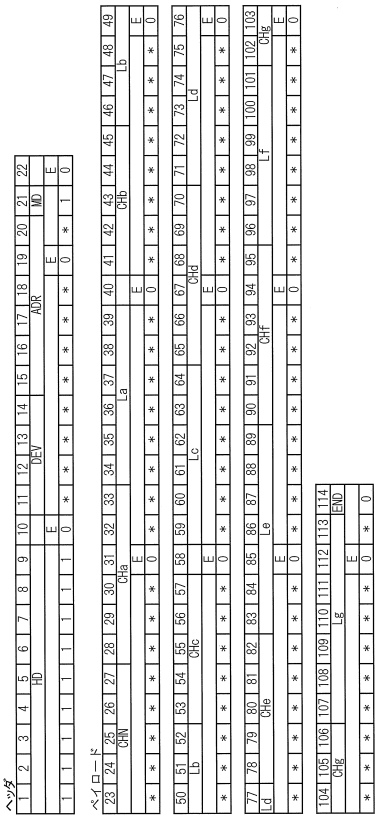
【図７】


20L, 20R
スピーカ群
118
第8ランプ群
111
第1ランプ群
117
第7ランプ群
119
ドットマトリクス部
87
B
86
RTC
84
発振回路（22MHz）
82
サブROM
83
サブRAM
85
バックアップRAM
88
サウンドIC
128a
128b
128c
121a
121b
127a
127b
127c
127d
129a
129b
129c
129d
129e
129f
129g
ドライバIC
81
サブCPU
42
41
主制御回路

【図８】


ヘッダ
ペイロード
ADR
DEV
MD
HD
END

【図９】



【図１０】

【図１１】

図柄配置表					
左リール		中リール		右リール	
図柄位置	図柄	図柄位置	図柄	図柄位置	図柄
20	ベル２	20	赤７	20	氷
19	氷	19	リプレイ	19	リプレイ
18	赤７	18	ベル１	18	チェリー
17	ベル２	17	ＢＡＲ	17	赤７
16	リプレイ	16	ブランク	16	ベル２
15	チェリー	15	リプレイ	15	氷
14	ＢＡＲ	14	ベル１	14	リプレイ
13	ベル１	13	チェリー	13	チェリー
12	氷	12	ドン１	12	ベル１
11	リプレイ	11	リプレイ	11	ＢＡＲ
10	ドン２	10	ベル１	10	リプレイ
9	ドン１	9	チェリー	9	ドン１
8	ドン１	8	氷	8	ベル２
7	ベル２	7	チェリー	7	氷
6	リプレイ	6	リプレイ	6	リプレイ
5	ブランク	5	ベル２	5	チェリー
4	チェリー	4	チェリー	4	ベル１
3	ドン１	3	ブランク	3	ＢＡＲ
2	ベル２	2	ベル１	2	リプレイ
1	氷	1	チェリー	1	チェリー
0	リプレイ	0	氷	0	ベル１

【図１２】

月別日テーブル

月	非閏年	閏年
１月	31	31
２月	28	29
３月	31	31
４月	30	30
５月	31	31
６月	30	30
７月	31	31
８月	31	31
９月	30	30
１０月	31	31
１１月	30	30
１２月	31	31

【図１３】

日時格納領域

格納領域種別	範囲	備考
年	2000～	
月	1～12	
日	1～31	
曜日	0～6	0=日曜日
時	0～23	
分	0～59	
秒	0～59	

【図１４】

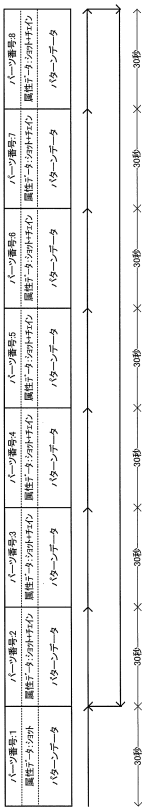
再生状態管理格納領域

格納領域	内容
エントリパーツ番号格納領域	再生中のパーツ番号
パーツ番号格納領域	パーツ番号
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
パーツ番号格納領域	
エンドコード格納領域	エンドコード
チェイン／ループ設定領域	チェイン又はループの有無

【図１５】

格納領域	値
エントリパーツ番号格納領域	1
パーツ番号格納領域	1
パーツ番号格納領域	2
パーツ番号格納領域	3
パーツ番号格納領域	4
パーツ番号格納領域	5
パーツ番号格納領域	6
パーツ番号格納領域	7
パーツ番号格納領域	8
エンドコード格納領域	エンドコード
チェイン／ループ設定領域	TRUE

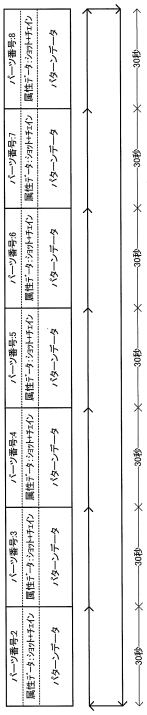
【図１６】



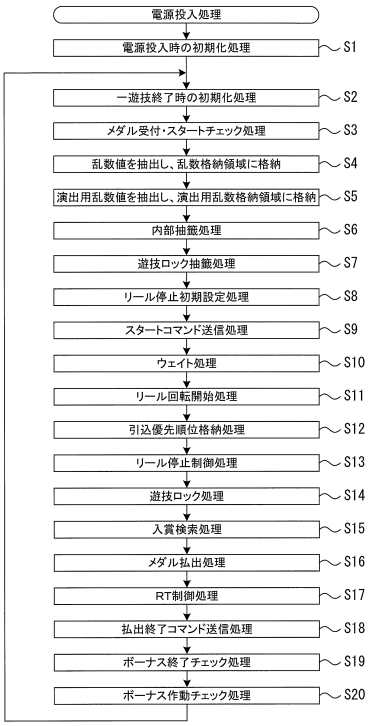
【図１７】

格納領域	値
エントリパーツ番号格納領域	2
パーツ番号格納領域	2
パーツ番号格納領域	3
パーツ番号格納領域	4
パーツ番号格納領域	5
パーツ番号格納領域	6
パーツ番号格納領域	7
パーツ番号格納領域	8
エントコード格納領域	エンドコード
チェイン／ループ設定領域	TRUE

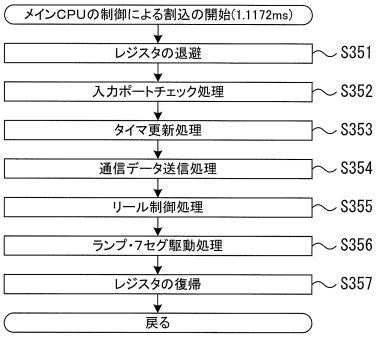
【図１８】



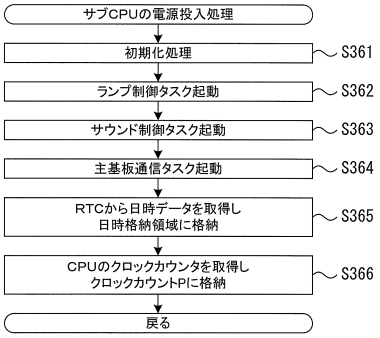
【図１９】



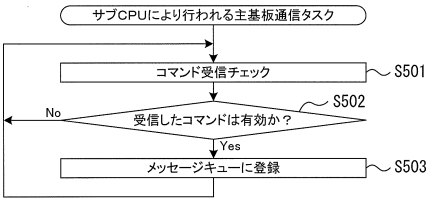
【図２０】



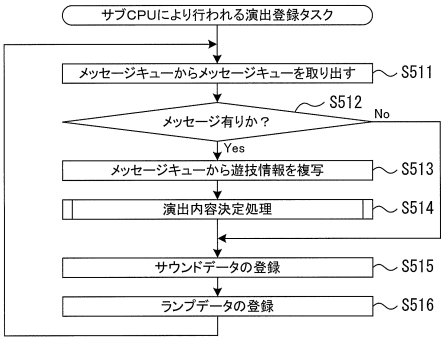
【図２１】



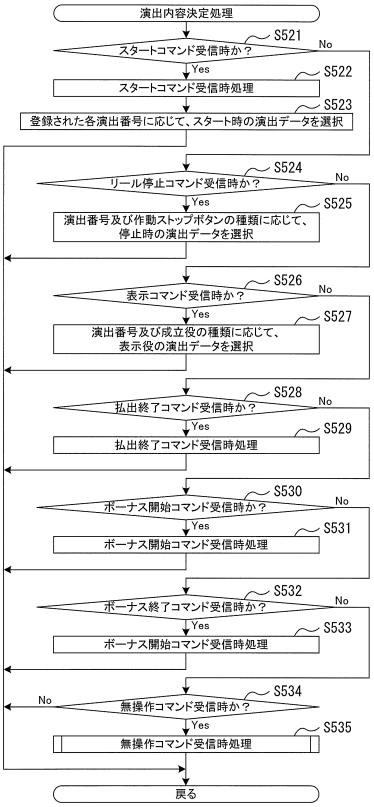
【図２２】



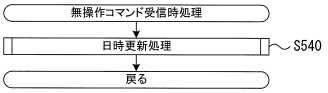
【図２３】



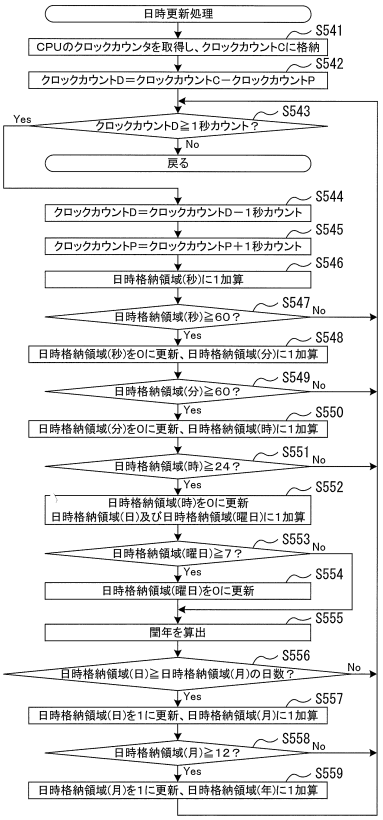
【図２４】



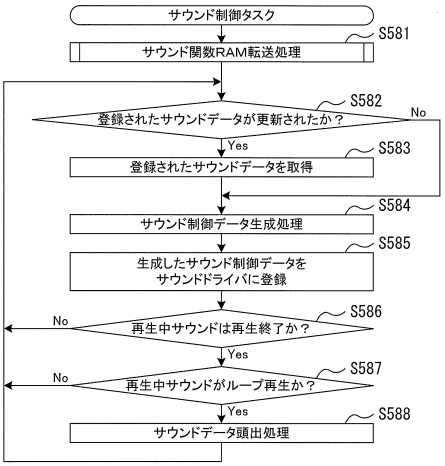
【図２５】



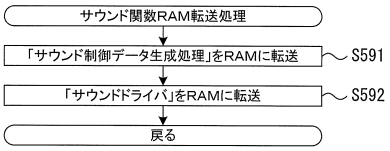
【図２６】



【図２７】



【図２８】

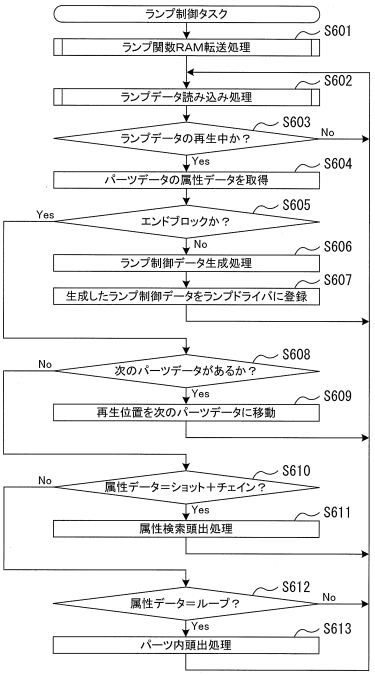


【図２９】

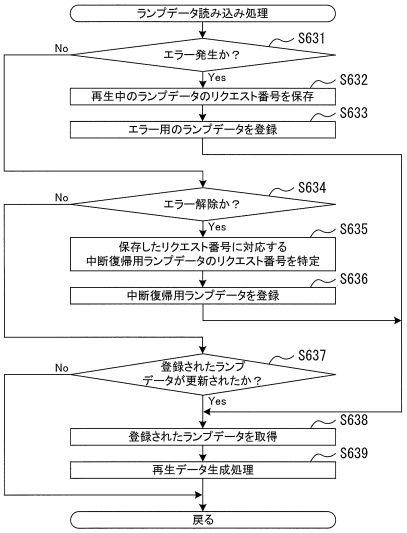
```
//サウンド関数RAM転送処理
void SoundRamTrans(){
    //'_SoundMakeData'をRAMの'_SoundMakeData'領域にコピー
    memcpy(_SoundMakeData,_SoundMakeData,sizeof(_SoundMakeData));      //(S591)
    //'_SetSoundDriver'をRAMの'_SetSoundDriver'領域にコピー
    memcpy(_SetSoundDriver,_ SetSoundDriver,sizeof(_SetSoundDriver));      //(S592)
}
//サウンド制御タスク
void SoundTask(){
    unsigned int index=0;            //サウンドデータのリクエスト番号
    unsigned char *SoundData;    //サウンドデータの保存領域
    unsigned char *SoundBuf;      //サウンド制御データの保存領域

    省略
    GetSoundData(index,SoundData);          //登録されたサウンドデータを取得(S583)
    SoundMakeData(SoundData,SoundBuf);  //サウンド制御データ生成処理(S584)
    SetSoundDriver(SoundBuf);                //サウンド制御データをサウンドドライバに登録(S585)
    省略
}
//RAMへコピーされたサウンド制御データ生成処理へのアクセス関数
void SoundMakeData(unsigned char *dat,unsigned char *buf){
    _SoundMakeData(dat,buf);            //RAMへコピーされたサウンド制御データ生成処理
}
//RAMへコピーされたサウンドドライバへのアクセス関数
void SetSoundDriver(unsigned char *buf){
    _SetSoundDriver(buf);                //サウンドドライバへサウンド制御データを登録
}
```

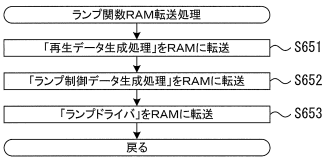
【図３０】



【図３１】



【図３２】

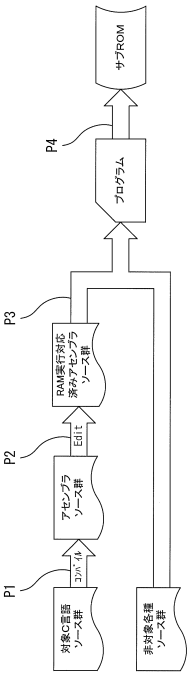


【図３３】

```
//ランプ関数RAM転送処理
void LampRamTrans(){
    //'_LampPlayMake' をRAMの'__LampPlayMake' 領域にコピー
    memcpy(_LampPlayMake,__LampPlayMake,sizeof(_LampPlayMake));      //(S651)
    //'_LampMakeData' をRAMの'__LampMakeData' 領域にコピー
    memcpy(_LampMakeData,__LampMakeData,sizeof(_LampMakeData));      //(S652)
    //'_SetLampDriver' をRAMの'__SetLampDriver' 領域にコピー
    memcpy(_SetLampDriver,__SetLampDriver,sizeof(_SetLampDriver));      //(S653)
}
//ランプ制御タスク
void LampTask(){
    unsigned char *LampData;    //輝度パターンの保存領域
    unsigned char *LampBuf;     //ランプ制御データの保存領域

    省略
    if(*LampData!=EndBlock){        //(S605)
        LampMakeData(LampData, LampBuf);  //ランプ制御データ生成処理(S606)
        SetLampDriver(LampBuf);                  //ランプ制御データをランプドライバに登録(S607)
    省略
}
//RAMへコピーされた再生データ生成処理へのアクセス関数
void LampPlayMake(unsigned char *dat,unsigned char *buf){
    __LampPlayMake(dat, buf);                //RAMへコピーされた再生データ生成処理
}
//RAMへコピーされたランプ制御データ生成処理へのアクセス関数
void LampMakeData(unsigned char *dat,unsigned char *buf){
    __LampMakeData(dat,buf);                //RAMへコピーされたランプ制御データ生成処理
}
//RAMへコピーされたランプドライバへのアクセス関数
void SetLampDriver(unsigned char *buf){
    __SetLampDriver(buf);                    //ランプドライバへランプ制御データを登録
}
```

【図３４】



フロントページの続き

(72)発明者　安間 由香里
東京都江東区有明３丁目７番２６号

審査官　鶴岡 直樹

(56)参考文献　特開２００８－１４８８９１（ＪＰ，Ａ）
特開２０１８－０１１９９１（ＪＰ，Ａ）
特開２０１８－０８６４２９（ＪＰ，Ａ）
特開２０１７－２０５４３９（ＪＰ，Ａ）
特開２０１２－１２０５５１（ＪＰ，Ａ）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ａ６３Ｆ　５／０４
Ａ６３Ｆ　７／０２