(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6258147号
(P6258147)

(45)発行日　平成30年1月10日(2018.1.10)　(24)登録日　平成29年12月15日(2017.12.15)

(51)Int.Cl.　F I
A63F 7/02 (2006.01)　A63F 7/02 312Z
A63F 7/02 334
A63F 7/02 320

請求項の数 3　(全 115 頁)

(21)出願番号　特願2014-151543(P2014-151543)
(22)出願日　平成26年7月25日(2014.7.25)
(65)公開番号　特開2016-28624(P2016-28624A)
(43)公開日　平成28年3月3日(2016.3.3)
審査請求日　平成28年2月19日(2016.2.19)

(73)特許権者　000144153
株式会社三共
東京都渋谷区渋谷三丁目２９番１４号
(74)代理人　100095407
弁理士　木村　満
(74)代理人　100148633
弁理士　桜田　圭
(74)代理人　100134599
弁理士　杉本　和之
(74)代理人　100166442
弁理士　鈴木　洋雅
(72)発明者　小倉　敏男
東京都渋谷区渋谷三丁目２９番１４号　株式会社三共内

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】
遊技媒体を用いて所定の遊技を行うことが可能であり、遊技媒体が進入可能な遊技領域が設けられ、始動領域を遊技媒体が通過したことにもとづいて、可変表示を行う遊技機であって、
始動領域として第１始動領域および第２始動領域が設けられ、
遊技媒体を振り分けるための振分装置が前記遊技領域に設けられ、
前記振分装置は、
遊技媒体が当該振分装置に流入可能な流入口と、
遊技媒体が通過可能な複数の通路と、
遊技媒体を前記複数の通路のうちのいずれかに振り分ける振分手段とを含み、
前記振分手段は、前記流入口から遊技媒体が流入したことにもとづいて、前記複数の通路のうちの第１通路に遊技媒体を振り分けやすい第１状態と第２通路に遊技媒体を振り分けやすい第２状態とに所定の順序に従って切り替わり、
前記第１始動領域は、前記第１通路に振り分けられた遊技媒体が通過しやすい態様で設けられ、
前記第２始動領域は、前記第２通路に振り分けられた遊技媒体が通過しやすい態様で設けられ、
始動領域に対応する記憶領域に格納された値に所定値を加算する加算手段と、
前記記憶領域に格納された値をクリアするクリア手段と、

前記記憶領域に格納された値にもとづいて異常と判定する異常判定手段と、
遊技媒体が前記第１始動領域または前記第２始動領域を通過したことにもとづいて遊技媒体が通過した始動領域に対応する前記記憶領域を前記加算手段による加算対象として設定し、遊技媒体が通過した始動領域とは異なる始動領域に対応する前記記憶領域を前記クリア手段によるクリア対象として設定する設定手段と、を備え、
前記設定手段は、
前記第１始動領域に対応する前記記憶領域を前記加算手段による加算対象とし、前記第２始動領域に対応する前記記憶領域を前記クリア手段によるクリア対象として仮設定する仮設定手段と、
遊技媒体が前記第２始動領域を通過したことにもとづいて、仮設定された前記加算手段による加算対象と前記クリア手段によるクリア対象とを入れ替える入れ替え手段とを含み、
前記加算手段は、前記設定手段によって加算対象とされた前記記憶領域に格納された値に前記所定値を加算し、
前記クリア手段は、前記設定手段によってクリア対象とされた前記記憶領域に格納された値をクリアし、
遊技の進行を制御する遊技制御用マイクロコンピュータと、
所定事象が発生したことにもとづいて第１リセットを発生させるか第２リセットを発生させるかを設定可能なリセット設定手段とをさらに備え、
前記第１リセットの発生後にはセキュリティチェックを実行する一方、前記第２リセットの発生後にはセキュリティチェックを実行せず、
前記遊技制御用マイクロコンピュータは、
遊技者にとって有利な有利状態に制御するか否かを決定する決定手段と、
前記決定手段による決定結果にもとづいて、可変表示の制御を行う可変表示制御手段と、
を含み、
前記遊技制御用マイクロコンピュータは、
遊技機への電力供給が停止していても、遊技に関する制御を行う際に発生する情報を所定期間保持可能な記憶手段と、
所定の処理を実行可能な所定処理実行手段と、
前記所定の処理の実行中に前記所定事象が発生したときに前記記憶手段の記憶内容を初期化する初期化手段とをさらに含み、
遊技媒体が通過しやすい第１状態と、遊技媒体が通過しにくい第２状態とに変化可能な可変入賞装置を備え、
遊技媒体は、前記可変入賞装置の状態にかかわらず、前記第２始動領域を通過可能である
ことを特徴とする遊技機。
【請求項２】
遊技媒体を用いて所定の遊技を行うことが可能であり、遊技媒体が進入可能な遊技領域が設けられ、始動領域を遊技媒体が通過したことにもとづいて、可変表示を行う遊技機であって、
始動領域として第１始動領域および第２始動領域が設けられ、
遊技媒体を振り分けるための振分装置が前記遊技領域に設けられ、
前記振分装置は、
遊技媒体が当該振分装置に流入可能な流入口と、
遊技媒体が通過可能な複数の通路と、
遊技媒体を前記複数の通路のうちのいずれかに振り分ける振分手段とを含み、
前記振分手段は、前記流入口から遊技媒体が流入したことにもとづいて、前記複数の通路のうちの第１通路に遊技媒体を振り分けやすい第１状態と第２通路に遊技媒体を振り分けやすい第２状態とに所定の順序に従って切り替わり、

前記第１始動領域は、前記第１通路に振り分けられた遊技媒体が通過しやすい態様で設けられ、
前記第２始動領域は、前記第２通路に振り分けられた遊技媒体が通過しやすい態様で設けられ、
始動領域に対応する記憶領域に格納された値から所定値を減算する減算手段と、
前記記憶領域に格納された値を特定値に設定する特定値設定手段と、
前記記憶領域に格納された値にもとづいて異常と判定する異常判定手段と、
遊技媒体が前記第１始動領域または前記第２始動領域を通過したことにもとづいて遊技媒体が通過した始動領域に対応する前記記憶領域を前記減算手段による減算対象として設定し、遊技媒体が通過した始動領域とは異なる始動領域に対応する前記記憶領域を前記特定値設定手段による特定値設定対象として設定する設定手段と、を備え、
前記設定手段は、
前記第１始動領域に対応する前記記憶領域を前記減算手段による減算対象とし、前記第２始動領域に対応する前記記憶領域を前記特定値設定手段による特定値設定対象として仮設定する仮設定手段と、
遊技媒体が前記第２始動領域を通過したことにもとづいて、仮設定された前記減算手段による減算対象と前記特定値設定手段による特定値設定対象とを入れ替える入れ替え手段とを含み、
前記減算手段は、前記設定手段によって減算対象とされた前記記憶領域に格納された値から前記所定値を減算し、
前記特定値設定手段は、前記設定手段によって特定値設定対象とされた前記記憶領域に格納された値を前記特定値に設定し、
遊技の進行を制御する遊技制御用マイクロコンピュータと、
所定事象が発生したことにもとづいて第１リセットを発生させるか第２リセットを発生させるかを設定可能なリセット設定手段とをさらに備え、
前記第１リセットの発生後にはセキュリティチェックを実行する一方、前記第２リセットの発生後にはセキュリティチェックを実行せず、
前記遊技制御用マイクロコンピュータは、
遊技者にとって有利な有利状態に制御するか否かを決定する決定手段と、
前記決定手段による決定結果にもとづいて、可変表示の制御を行う可変表示制御手段と、
を含み、
前記遊技制御用マイクロコンピュータは、
遊技機への電力供給が停止していても、遊技に関する制御を行う際に発生する情報を所定期間保持可能な記憶手段と、
所定の処理を実行可能な所定処理実行手段と、
前記所定の処理の実行中に前記所定事象が発生したときに前記記憶手段の記憶内容を初期化する初期化手段とをさらに含み、
遊技媒体が通過しやすい第１状態と、遊技媒体が通過しにくい第２状態とに変化可能な可変入賞装置を備え、
遊技媒体は、前記可変入賞装置の状態にかかわらず、前記第２始動領域を通過可能である
ことを特徴とする遊技機。
【請求項３】
低確率状態と、前記有利状態に制御されやすい特別状態とに制御可能な状態制御手段と、
前記低確率状態であるときと前記特別状態であるときとにおいて共通演出を実行可能な共通演出実行手段と、
をさらに備え、
前記共通演出実行手段は、前記有利状態と、前記有利状態とは異なる所定状態とのいず

れかに制御されたことにもとづいて、前記共通演出を実行可能である
ことを特徴とする請求項１又は請求項２に記載の遊技機。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、パチンコ遊技機等の遊技機に関する。
【背景技術】
【０００２】
遊技機として、遊技媒体である遊技球を発射装置によって遊技領域に発射し、遊技領域に設けられている入賞口などの入賞領域に遊技球が入賞すると、所定個の賞球が遊技者に払い出されるものがある。さらに、識別情報を可変表示（「変動」ともいう。）可能な可変表示部が設けられ、可変表示部において識別情報の可変表示の表示結果が特定表示結果となった場合に、所定の遊技価値を遊技者に与えるように構成されたものがある。
【０００３】
なお、遊技価値とは、遊技機の遊技領域に設けられた可変入賞球装置の状態が打球が入賞しやすい遊技者にとって有利な状態になることや、遊技者にとって有利な状態になるための権利を発生させたりすることや、賞球払出の条件が成立しやすくなる状態になることである。
【０００４】
パチンコ遊技機では、始動入賞口に遊技球が入賞したことにもとづいて可変表示部において開始される特別図柄（識別情報）の可変表示の表示結果として、あらかじめ定められた特定の表示態様が導出表示された場合に、「大当り」が発生する。なお、導出表示とは、図柄を停止表示させることである（いわゆる再変動の前の停止を除く。）。大当りが発生すると、例えば、大入賞口が所定回数開放して打球が入賞しやすい大当り遊技状態に移行する。そして、各開放期間において、所定個（例えば１０個）の大入賞口への入賞があると大入賞口は閉成する。そして、大入賞口の開放回数は、所定回数（例えば１５ラウンド）に固定されている。なお、各開放について開放時間（例えば２９秒）が決められ、入賞数が所定個に達しなくても開放時間が経過すると大入賞口は閉成する。以下、各々の大入賞口の開放期間をラウンドということがある。
【０００５】
そのような遊技機において、遊技媒体（遊技球）を振り分けるための振分装置が遊技領域に設けられるように構成されたものがある。例えば、特許文献１には、遊技領域に振分装置（振分ユニット）を設け、振分装置に流入した遊技媒体の自重により振分装置内に設けられた回転体が回転することによって、振分装置に流入した遊技媒体が第１領域と第２領域とに交互に振り分けられることが記載されている。
【先行技術文献】
【特許文献】
【０００６】
【特許文献１】特開２０１０－１０４５６４号公報（段落００４４－００５６、図３８）
【発明の概要】
【発明が解決しようとする課題】
【０００７】
しかし、特許文献１に記載された遊技機では、振分装置に関して何らかの異常が発生すると、正常に遊技媒体（遊技球）の振り分けを行うことができなくなり、遊技を正常に行うことができなくなるおそれがある。
【０００８】
そこで、本発明は、振分装置の異常により遊技を正常に行えなくなることを防止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】
【０００９】

（手段１）本発明による遊技機は、遊技媒体（例えば、遊技球）を用いて所定の遊技を行うことが可能であり、遊技媒体が進入可能な遊技領域（例えば、遊技領域７）が設けられ、始動領域（例えば、第１始動入賞口１３、第２始動入賞口１４、第３始動入賞口１７）を遊技媒体が通過したことにもとづいて、可変表示を行う遊技機であって、始動領域として第１始動領域（例えば、第１始動入賞口１３）および第２始動領域（例えば、第２始動入賞口１４）が設けられ、遊技媒体を振り分けるための振分装置（例えば、振分装置２００）が遊技領域に設けられ、振分装置は、遊技媒体が当該振分装置に流入可能な流入口（例えば、流入口２０１）と、遊技媒体が通過可能な複数の通路（例えば、左側通路２０３、右側通路２０４）と、遊技媒体を複数の通路のうちのいずれかに振り分ける振分手段（例えば、振分部材２０２）とを含み、振分手段は、流入口から遊技媒体が流入したことにもとづいて、複数の通路のうちの第１通路（例えば、左側通路２０３）に遊技媒体を振り分けやすい第１状態（例えば、図２（ａ），（ｄ）に示すように振分部材２０２が右側に倒れた状態）と第２通路（例えば、右側通路２０４）に遊技媒体を振り分けやすい第２状態（例えば、図２（ｂ），（ｃ）に示すように振分部材２０２が左側に倒れた状態）とに所定の順序に従って切り替わり（例えば、図２に示すように交互に切り替わる）、第１始動領域は、第１通路に振り分けられた遊技媒体が通過しやすい態様で設けられ（例えば、図２および図３に示すように、第１始動入賞口１３は左側流出口２０５の下方に設けられている）、第２始動領域は、第２通路に振り分けられた遊技媒体が通過しやすい態様で設けられ（例えば、図２および図３に示すように、第２始動入賞口１４は右側流出口２０６の下方に設けられている）、始動領域に対応する記憶領域（例えば、第１始動口監視用バッファおよび第２始動口監視用バッファ）に格納された値に所定値を加算する加算手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５４を実行する部分）と、記憶領域に格納された値をクリアするクリア手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５５を実行する部分）と、記憶領域に格納された値にもとづいて異常と判定する異常判定手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５９を実行する部分）と、遊技媒体が第１始動領域または第２始動領域を通過したことにもとづいて遊技媒体が通過した始動領域に対応する記憶領域を加算手段による加算対象として設定し、遊技媒体が通過した始動領域とは異なる始動領域に対応する記憶領域をクリア手段によるクリア対象として設定する設定手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５０～Ｓ２５３を実行する部分）と、を備え、設定手段は、第１始動領域に対応する記憶領域を加算手段による加算対象とし、第２始動領域に対応する記憶領域をクリア手段によるクリア対象として仮設定する仮設定手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５０～Ｓ２５１を実行する部分）と、遊技媒体が第２始動領域を通過したことにもとづいて、仮設定された加算手段による加算対象とクリア手段によるクリア対象とを入れ替える入れ替え手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５２～Ｓ２５３を実行する部分）とを含み、加算手段は、設定手段によって加算対象とされた記憶領域に格納された値に所定値を加算し（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５４を実行する部分）、クリア手段は、設定手段によってクリア対象とされた記憶領域に格納された値をクリアし（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５５を実行する部分）、遊技の進行を制御する遊技制御用マイクロコンピュータ（例えば、遊技制御用マイクロコンピュータ５６０）と、所定事象が発生（例えば、ＩＡＴ回路５０６ａからのＩＡＴ信号の入力、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号の入力）したことにもとづいて第１リセット（例えば、システムリセット）を発生させるか第２リセット（例えば、ユーザリセット）を発生させるかを設定可能なリセット設定手段（例えば、遊技制御用マイクロコンピュータ５６０におけるプログラム管理エリアの図１４に示すリセット設定（ＫＲＥＳ）のビット７の設定値に従ってステップＳ１００１，Ｓ１０１１を実行する部分）とをさらに備え、第１リセットの発生後にはセキュリティチェックを実行する一方、第２リセットの発生後にはセキュリティチェックを実行せず（例えば、遊技制御用マイクロコンピュータ５６０は、図４８（Ａ）に示すように、ステップＳ

１００４の後にはステップＳ１００６でセキュリティチェックを実行し、図４８（Ｂ）に示すように、ステップＳ１０１４の後にはセキュリティチェックを実行しない）、遊技制御用マイクロコンピュータは、遊技者にとって有利な有利状態に制御するか否かを決定する決定手段（例えば、遊技制御用マイクロコンピュータ５６０のＳ３００の処理を実行する部分）と、決定手段による決定結果にもとづいて、可変表示の制御を行う可変表示制御手段（例えば、遊技制御用マイクロコンピュータ５６０のＳ３０１の処理を実行する部分）と、を含み、遊技制御用マイクロコンピュータは、遊技機への電力供給が停止していても、遊技に関する制御を行う際に発生する情報を所定期間保持可能な記憶手段（例えば、ＲＡＭ５５（バックアップＲＡＭ））と、所定の処理（例えば、メイン処理におけるループ処理、タイマ割込処理）を実行可能な所定処理実行手段（例えば、遊技制御用マイクロコンピュータ５６０におけるステップＳ１６～Ｓ１９，Ｓ２０～Ｓ３４を実行する部分）と、所定の処理の実行中に所定事象が発生したときに記憶手段の記憶内容を初期化する初期化手段（例えば、遊技制御用マイクロコンピュータ５６０において、ステップＳ１６～Ｓ１９，Ｓ２０～Ｓ３４の実行中にＩＡＴ回路５０６ａからのＩＡＴ信号やウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号を入力したときには、図５２および図５３に示す電源断処理を実行することなく、リセット後にメイン処理が開始されたときにステップＳ１０～Ｓ１３を実行する部分）とをさらに含み、遊技媒体が通過しやすい第１状態と、遊技媒体が通過しにくい第２状態とに変化可能な可変入賞装置を備え、遊技媒体は、可変入賞装置の状態にかかわらず、第２始動領域を通過可能であることを特徴とする。

そのような構成により、振分装置の異常判定を行うことができるので、振分装置の異常により遊技を正常に行えなくなることを防止することができる。

【００１０】

（手段２）本発明による遊技機は、遊技媒体（例えば、遊技球）を用いて所定の遊技を行うことが可能であり、遊技媒体が進入可能な遊技領域（例えば、遊技領域７）が設けられ、始動領域（例えば、第１始動入賞口１３、第２始動入賞口１４、第３始動入賞口１７）を遊技媒体が通過したことにもとづいて、可変表示を行う遊技機であって、始動領域として第１始動領域（例えば、第１始動入賞口１３）および第２始動領域（例えば、第２始動入賞口１４）が設けられ、遊技媒体を振り分けるための振分装置（例えば、振分装置２００）が遊技領域に設けられ、振分装置は、遊技媒体が当該振分装置に流入可能な流入口（例えば、流入口２０１）と、遊技媒体が通過可能な複数の通路（例えば、左側通路２０３、右側通路２０４）と、遊技媒体を複数の通路のうちのいずれかに振り分ける振分手段（例えば、振分部材２０２）とを含み、振分手段は、流入口から遊技媒体が流入したことにもとづいて、複数の通路のうちの第１通路（例えば、左側通路２０３）に遊技媒体を振り分けやすい第１状態（例えば、図２（ａ），（ｄ）に示すように振分部材２０２が右側に倒れた状態）と第２通路（例えば、右側通路２０４）に遊技媒体を振り分けやすい第２状態（例えば、図２（ｂ），（ｃ）に示すように振分部材２０２が左側に倒れた状態）とに所定の順序に従って切り替わり（例えば、図２に示すように交互に切り替わる）、第１始動領域は、第１通路に振り分けられた遊技媒体が通過しやすい態様で設けられ（例えば、図２および図３に示すように、第１始動入賞口１３は左側流出口２０５の下方に設けられている）、第２始動領域は、第２通路に振り分けられた遊技媒体が通過しやすい態様で設けられ（例えば、図２および図３に示すように、第２始動入賞口１４は右側流出口２０６の下方に設けられている）、始動領域に対応する記憶領域（例えば、第１始動口監視用バッファおよび第２始動口監視用バッファ）に格納された値から所定値を減算する減算手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５４ｂを実行する部分）と、記憶領域に格納された値を特定値に設定する特定値設定手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５５ｂを実行する部分）と、記憶領域に格納された値にもとづいて異常と判定する異常判定手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５９ｂを実行する部分）と、遊技媒体が第１始動領域または第２始動領域を通過したことにもとづいて遊技媒体が通過した始動領域に対応する記憶領

域を減算手段による減算対象として設定し、遊技媒体が通過した始動領域とは異なる始動領域に対応する記憶領域を特定値設定手段による特定値設定対象として設定する設定手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５０ｂ～Ｓ２５３ｂを実行する部分）と、を備え、設定手段は、第１始動領域に対応する記憶領域を減算手段による減算対象とし、第２始動領域に対応する記憶領域を特定値設定手段による特定値設定対象として仮設定する仮設定手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５０ｂ～Ｓ２５１ｂを実行する部分）と、遊技媒体が第２始動領域を通過したことにもとづいて、仮設定された減算手段による減算対象と特定値設定手段による特定値設定対象とを入れ替える入れ替え手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５２ｂ～Ｓ２５３ｂを実行する部分）とを含み、減算手段は、設定手段によって減算対象とされた記憶領域に格納された値から所定値を減算し（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５４ｂを実行する部分）、特定値設定手段は、設定手段によって特定値設定対象とされた記憶領域に格納された値を特定値に設定し（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ２５５ｂを実行する部分）、遊技の進行を制御する遊技制御用マイクロコンピュータ（例えば、遊技制御用マイクロコンピュータ５６０）と、所定事象が発生（例えば、ＩＡＴ回路５０６ａからのＩＡＴ信号の入力、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号の入力）したことにもとづいて第１リセット（例えば、システムリセット）を発生させるか第２リセット（例えば、ユーザリセット）を発生させるかを設定可能なリセット設定手段（例えば、遊技制御用マイクロコンピュータ５６０におけるプログラム管理エリアの図１４に示すリセット設定（ＫＲＥＳ）のビット７の設定値に従ってステップＳ１００１，Ｓ１０１１を実行する部分）とをさらに備え、第１リセットの発生後にはセキュリティチェックを実行する一方、第２リセットの発生後にはセキュリティチェックを実行せず（例えば、遊技制御用マイクロコンピュータ５６０は、図４８（Ａ）に示すように、ステップＳ１００４の後にはステップＳ１００６でセキュリティチェックを実行し、図４８（Ｂ）に示すように、ステップＳ１０１４の後にはセキュリティチェックを実行しない）、遊技制御用マイクロコンピュータは、遊技者にとって有利な有利状態に制御するか否かを決定する決定手段（例えば、遊技制御用マイクロコンピュータ５６０のＳ３００の処理を実行する部分）と、決定手段による決定結果にもとづいて、可変表示の制御を行う可変表示制御手段（例えば、遊技制御用マイクロコンピュータ５６０のＳ３０１の処理を実行する部分）と、を含み、遊技制御用マイクロコンピュータは、遊技機への電力供給が停止していても、遊技に関する制御を行う際に発生する情報を所定期間保持可能な記憶手段（例えば、ＲＡＭ５５（バックアップＲＡＭ））と、所定の処理（例えば、メイン処理におけるループ処理、タイマ割込処理）を実行可能な所定処理実行手段（例えば、遊技制御用マイクロコンピュータ５６０におけるステップＳ１６～Ｓ１９，Ｓ２０～Ｓ３４を実行する部分）と、所定の処理の実行中に所定事象が発生したときに記憶手段の記憶内容を初期化する初期化手段（例えば、遊技制御用マイクロコンピュータ５６０において、ステップＳ１６～Ｓ１９，Ｓ２０～Ｓ３４の実行中にＩＡＴ回路５０６ａからのＩＡＴ信号やウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号を入力したときには、図５２および図５３に示す電源断処理を実行することなく、リセット後にメイン処理が開始されたときにステップＳ１０～Ｓ１３を実行する部分）とをさらに含み、遊技媒体が通過しやすい第１状態と、遊技媒体が通過しにくい第２状態とに変化可能な可変入賞装置を備え、遊技媒体は、可変入賞装置の状態にかかわらず、第２始動領域を通過可能であることを特徴とする。

そのような構成により、振分装置の異常判定を行うことができるので、振分装置の異常により遊技を正常に行えなくなることを防止することができる。

【００１１】

（手段３）手段１または手段２において、始動領域を遊技媒体が通過しやすい高頻度状態（例えば、高ベース状態）に制御する高頻度状態制御手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ１６６を実行し、高ベース状態に制御する部分）を備え、異常判定手段は、高頻度状態に制御されているときには異常と判定しない（例えば、遊

技制御用マイクロコンピュータ５６０がステップＳ２５７，Ｓ２５８を実行する部分）ように構成されていてもよい。

そのような構成によれば、不要な異常判定が行われてしまう無駄を防止することができる。

【００１２】

（手段４）手段１から手段３のうちのいずれかにおいて、未だ開始されていない可変表示に関する情報を保留記憶として記憶可能な保留記憶手段（例えば、第１保留記憶バッファ、第２保留記憶バッファ）と、保留記憶手段に記憶された保留記憶にもとづいて、有利状態に制御されるか否かを、当該保留記憶に対応する可変表示の表示結果を導出表示するよりも前に判定する特定判定手段（例えば、遊技制御用マイクロコンピュータ５６０がＳ２０６１Ａ，Ｓ２０６１Ｂを実行する部分）と、特定判定手段の判定にもとづいて、有利状態に制御されることを予告する特定演出を実行可能な特定演出実行手段（例えば、演出制御用マイクロコンピュータ１００がＳ６３１，Ｓ８０１，Ｓ８０２を実行する部分）と、始動領域を遊技媒体が通過しやすい高頻度状態に制御する高頻度状態制御手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ１６６を実行し、高ベース状態に制御する部分）とを備え、特定演出実行手段は、高頻度状態に制御されていないときには所定回の可変表示に亘る第１特定演出を実行可能であり、高頻度状態に制御されているときには所定回よりも少ない特定回の可変表示に亘る第２特定演出を実行可能である（例えば、演出制御用マイクロコンピュータ１００がＳ６３１を実行する部分。図７２参照）ように構成されていてもよい。

そのような構成によれば、遊技状態に応じた特定演出を実行することができる。

【００１３】

（手段５）手段１から手段４のうちのいずれかにおいて、低確率状態と、有利状態に制御されやすい特別状態とに制御可能な状態制御手段（例えば、遊技制御用マイクロコンピュータ５６０がステップＳ１６６，Ｓ１６７，Ｓ１７０の処理を実行する部分）と、低確率状態であるときと特別状態であるときとにおいて共通演出を実行可能な共通演出実行手段（例えば、演出制御用マイクロコンピュータ１００がステップＳ７０５の処理を実行する部分）と、をさらに備え、共通演出実行手段は、有利状態と、有利状態とは異なる所定状態とのいずれかに制御されたことにもとづいて、共通演出を実行可能に構成されていてもよい。

そのような構成によれば、所定状態に制御された後と、所定状態以外の状態に制御された後とにおいて、特定の可変表示パターンの選択割合が変わるため、所定状態に制御された後に行われる共通演出の実行中において、特定の可変表示パターンでの演出が増加するため、共通演出についての興趣を向上できる。

【図面の簡単な説明】

【００１４】

【図１】パチンコ遊技機を正面からみた正面図である。

【図２】振分装置を説明するための説明図である。

【図３】可変入賞球装置が開放状態に制御されているときに第２始動入賞口に遊技球が入賞する形態を説明するための説明図である。

【図４】遊技制御基板（主基板）の回路構成例を示すブロック図である。

【図５】演出制御基板、ランプドライバ基板および音声出力基板の回路構成例を示すブロック図である。

【図６】遊技制御用マイクロコンピュータの構成例を示すブロック図である。

【図７】遊技制御用マイクロコンピュータにおけるアドレスマップの一例を示す図である。

【図８】プログラム管理エリアの主要部分を例示する図である。

【図９】内蔵レジスタの主要部分を例示する図である。

【図１０】内蔵レジスタの主要部分を例示する図である。

【図１１】内蔵レジスタの主要部分を例示する図である。

【図１２】ヘッダ（ＫＨＤＲ）における設定データと動作との対応関係を例示する図である。
【図１３】プログラムコードエンドアドレス（ＫＰＣＥ）、プログラムコードスタートアドレス２（ＫＰＣＳ２）、およびプログラムコードエンドアドレス２（ＫＰＣＥ２）における設定内容の一例を示す図である。
【図１４】リセット設定（ＫＲＥＳ）における設定内容の一例を示す図である。
【図１５】１６ビット乱数初期設定１（ＫＲＬ１）のにおける設定内容の一例を示す図である。
【図１６】１６ビット乱数初期設定２（ＫＲＬ２）における設定内容の一例を示す図である。
【図１７】１６ビット乱数初期設定３（ＫＲＬ３）のにおける設定内容の一例を示す図である。
【図１８】８ビット乱数初期設定１（ＫＲＳ１）のにおける設定内容の一例を示す図である。
【図１９】８ビット乱数初期設定２（ＫＲＳ２）のにおける設定内容の一例を示す図である。
【図２０】セキュリティ時間設定（ＫＳＥＳ）における設定内容の一例を示す図である。
【図２１】乱数クロック監視設定（ＫＲＣＳ）における設定内容の一例を示す図である。
【図２２】内部情報レジスタの構成例等を示す図である。
【図２３】８ビット乱数回路の一構成例を示すブロック図である。
【図２４】１６ビット乱数回路の一構成例を示すブロック図である。
【図２５】ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）の構成例および設定内容の一例を示す説明図である。
【図２６】ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）の構成例および設定内容の一例を示す説明図である。
【図２７】ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）の構成例および設定内容の一例を示す説明図である。
【図２８】ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）の構成例および設定内容の一例を示す説明図である。
【図２９】ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）の構成例および設定内容の一例を示す説明図である。
【図３０】ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）の構成例および設定内容の一例を示す説明図である。
【図３１】ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）の構成例および設定内容の一例を示す説明図である。
【図３２】ＲＳ割り込み制御レジスタ（ＲＳＩＣ）の構成例および設定内容の一例を示す説明図である。
【図３３】ＲＬｎ最大値設定レジスタ（ＲＬｎＭＸ）の構成例および設定内容の一例を示す説明図である。
【図３４】ＲＳｎ最大値設定レジスタ（ＲＳｎＭＸ）の構成例および設定内容の一例を示す説明図である。
【図３５】乱数列変更レジスタ（ＲＤＳＣ）の構成例および設定内容の一例を示す説明図である。
【図３６】乱数ソフトラッチレジスタ（ＲＤＳＬ）の構成例および設定内容の一例を示す説明図である。
【図３７】乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）の構成例および設定内容の一例を示す説明図である。
【図３８】ＲＬｎソフトラッチ乱数値レジスタ（ＲＬｎＳＶ）の構成例および設定内容の一例を示す説明図である。
【図３９】ＲＳｎソフトラッチ乱数値レジスタ（ＲＳｎＳＶ）の構成例および設定内容の

一例を示す説明図である。
【図４０】ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）の構成例および設定内容の一例を示す説明図である。
【図４１】ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）の構成例および設定内容の一例を示す説明図である。
【図４２】ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）の構成例および設定内容の一例を示す説明図である。
【図４３】ＲＬ０ハードラッチ乱数値レジスタｍ（ＲＬ０ｍＨＶ）の構成例および設定内容の一例を示す説明図である。
【図４４】ＲＬ１ハードラッチ乱数値レジスタｍ（ＲＬ１ｍＨＶ）の構成例および設定内容の一例を示す説明図である。
【図４５】ＲＬ２ハードラッチ乱数値レジスタｍ（ＲＬ２ｍＨＶ）の構成例および設定内容の一例を示す説明図である。
【図４６】ＲＬ３ハードラッチ乱数値レジスタｍ（ＲＬ３ｍＨＶ）の構成例および設定内容の一例を示す説明図である。
【図４７】ＲＳｎハードラッチ乱数値レジスタ（ＲＳｎＨＶ）の構成例および設定内容の一例を示す説明図である。
【図４８】リセット設定（ＫＲＥＳ）での設定内容によるリセット動作の違いを説明するための説明図である。
【図４９】内蔵ＲＡＭ領域に格納されているデータの読み出し方の例を示す説明図である。
【図５０】主基板におけるＣＰＵが実行するメイン処理を示すフローチャートである。
【図５１】４ｍｓタイマ割込処理を示すフローチャートである。
【図５２】電源断処理の一例を示すフローチャートである。
【図５３】電源断処理の一例を示すフローチャートである。
【図５４】各乱数を示す説明図である。
【図５５】大当り判定テーブル、小当り判定テーブルおよび大当り種別判定テーブルを示す説明図である。
【図５６】特別図柄プロセス処理のプログラムの一例を示すフローチャートである。
【図５７】特別図柄プロセス処理のプログラムの一例を示すフローチャートである。
【図５８】始動口スイッチ通過処理を示すフローチャートである。
【図５９】保留記憶バッファの構成例を示す説明図である。
【図６０】特別図柄停止処理を示すフローチャートである。
【図６１】大当り終了処理を示すフローチャートである。
【図６２】普通図柄プロセス処理を示すフローチャートである。
【図６３】入賞順異常報知処理を示すフローチャートである。
【図６４】演出制御用ＣＰＵが実行する演出制御メイン処理を示すフローチャートである。
【図６５】コマンド受信バッファの構成例を示す説明図である。
【図６６】コマンド解析処理を示すフローチャートである。
【図６７】演出制御プロセス処理を示すフローチャートである。
【図６８】入賞順異常報知処理の変形例を示すフローチャートである。
【図６９】主基板におけるＣＰＵが実行するメイン処理の変形例を示すフローチャートである。
【図７０】始動口スイッチ通過処理の変形例を示すフローチャートである。
【図７１】コマンド解析処理の変形例を示すフローチャートである。
【図７２】先読み演出の演出態様を示す説明図である。
【発明を実施するための形態】
【００１５】
以下、本発明の実施の形態を、図面を参照して説明する。まず、遊技機の一例であるパ

チンコ遊技機１の全体の構成について説明する。図１はパチンコ遊技機１を正面からみた正面図である。

【００１６】

パチンコ遊技機１は、縦長の方形状に形成された外枠（図示せず）と、外枠の内側に開閉可能に取り付けられた遊技枠とで構成される。また、パチンコ遊技機１は、遊技枠に開閉可能に設けられている額縁状に形成されたガラス扉枠２を有する。遊技枠は、外枠に対して開閉自在に設置される前面枠（図示せず）と、機構部品等が取り付けられる機構板（図示せず）と、それらに取り付けられる種々の部品（後述する遊技盤６を除く）とを含む構造体である。

【００１７】

ガラス扉枠２の下部表面には打球供給皿（上皿）３がある。打球供給皿３の下部には、打球供給皿３に収容しきれない遊技球を貯留する余剰球受皿４や、打球を発射する打球操作ハンドル（操作ノブ）５が設けられている。また、ガラス扉枠２の背面には、遊技盤６が着脱可能に取り付けられている。なお、遊技盤６は、それを構成する板状体と、その板状体に取り付けられた種々の部品とを含む構造体である。また、遊技盤６の前面には、打ち込まれた遊技球が流下可能な遊技領域７が形成されている。

【００１８】

余剰球受皿（下皿）４を形成する部材には、例えば下皿本体の上面における手前側の所定位置（例えば下皿の中央部分）などに、スティック形状（棒形状）に構成され、遊技者が把持して複数方向（前後左右）に傾倒操作が可能なスティックコントローラ１２２が取り付けられている。なお、スティックコントローラ１２２には、遊技者がスティックコントローラ１２２の操作桿を操作手（例えば左手など）で把持した状態において、所定の操作指（例えば人差し指など）で押引操作することなどにより所定の指示操作が可能なトリガボタン１２１（図５を参照）が設けられ、スティックコントローラ１２２の操作桿の内部には、トリガボタン１２１に対する押引操作などによる所定の指示操作を検知するトリガセンサ１２５（図５を参照）が内蔵されている。また、スティックコントローラ１２２の下部における下皿の本体内部などには、操作桿に対する傾倒操作を検知する傾倒方向センサユニット１２３（図５を参照）が設けられている。また、スティックコントローラ１２２には、スティックコントローラ１２２を振動動作させるためのバイブレータ用モータ１２６（図５を参照）が内蔵されている。

【００１９】

打球供給皿（上皿）３を形成する部材には、例えば上皿本体の上面における手前側の所定位置（例えばスティックコントローラ１２２の上方）などに、遊技者が押下操作などにより所定の指示操作を可能なプッシュボタン１２０が設けられている。プッシュボタン１２０は、遊技者からの押下操作などによる所定の指示操作を、機械的、電気的、あるいは、電磁的に、検出できるように構成されていればよい。プッシュボタン１２０の設置位置における上皿の本体内部などには、プッシュボタン１２０に対してなされた遊技者の操作行為を検知するプッシュセンサ１２４（図５を参照）が設けられていればよい。図１に示す構成例では、プッシュボタン１２０とスティックコントローラ１２２の取付位置が、上皿及び下皿の中央部分において上下の位置関係にある。これに対して、上下の位置関係を保ったまま、プッシュボタン１２０及びスティックコントローラ１２２の取付位置を、上皿及び下皿において左右のいずれかに寄せた位置としてもよい。あるいは、プッシュボタン１２０とスティックコントローラ１２２の取付位置が上下の位置関係にはなく、例えば左右の位置関係にあるものとしてもよい。

【００２０】

遊技領域７の中央付近には、液晶表示装置（ＬＣＤ）で構成された演出表示装置９が設けられている。演出表示装置９の表示画面には、第１特別図柄または第２特別図柄の可変表示に同期した演出図柄の可変表示を行う演出図柄表示領域がある。よって、演出表示装置９は、演出図柄の可変表示を行う可変表示装置に相当する。演出図柄表示領域には、例えば「左」、「中」、「右」の３つの装飾用（演出用）の演出図柄を可変表示する図柄表

示エリアがある。図柄表示エリアには「左」、「中」、「右」の各図柄表示エリアがあるが、図柄表示エリアの位置は、演出表示装置９の表示画面において固定的でなくてもよいし、図柄表示エリアの３つ領域が離れてもよい。演出表示装置９は、演出制御基板に搭載されている演出制御用マイクロコンピュータによって制御される。演出制御用マイクロコンピュータが、第１特別図柄表示器８ａで第１特別図柄の可変表示が実行されているときに、その可変表示に伴って演出表示装置９で演出表示を実行させ、第２特別図柄表示器８ｂで第２特別図柄の可変表示が実行されているときに、その可変表示に伴って演出表示装置９で演出表示を実行させるので、遊技の進行状況を把握しやすくすることができる。

【００２１】

また、演出表示装置９において、最終停止図柄（例えば左右中図柄のうち中図柄）となる図柄以外の図柄が、所定時間継続して、大当り図柄（例えば左中右の図柄が同じ図柄で揃った図柄の組み合わせ）と一致している状態で停止、揺動、拡大縮小もしくは変形している状態、または、複数の図柄が同一図柄で同期して変動したり、表示図柄の位置が入れ替わっていたりして、最終結果が表示される前で大当り発生の可能性が継続している状態（以下、これらの状態をリーチ状態という。）において行われる演出をリーチ演出という。また、リーチ状態やその様子をリーチ態様という。さらに、リーチ演出を含む可変表示をリーチ可変表示という。そして、演出表示装置９に変動表示される図柄の表示結果が大当り図柄でない場合には「はずれ」となり、変動表示状態は終了する。遊技者は、大当りをいかにして発生させるかを楽しみつつ遊技を行う。

【００２２】

演出表示装置９の表示画面の右上方部には、演出図柄と後述する特別図柄および普通図柄とに次ぐ第４図柄を表示する第４図柄表示領域９ｃ，９ｄが設けられている。この実施の形態では、後述する第１特別図柄の変動表示に同期して第１特別図柄用の第４図柄の変動表示が行われる第１特別図柄用の第４図柄表示領域９ｃと、第２特別図柄の変動表示に同期して第２特別図柄用の第４図柄の変動表示が行われる第２特別図柄用の第４図柄表示領域９ｄとが設けられている。

【００２３】

この実施の形態では、特別図柄の変動表示に同期して演出図柄の変動表示が実行されるのであるが（ただし、正確には、演出図柄の変動表示は、演出制御用マイクロコンピュータ１００側で変動パターンコマンドにもとづいて認識した変動時間を計測することによって行われる。）、演出表示装置９を用いた演出を行う場合、例えば、演出図柄の変動表示を含む演出内容が画面上から一瞬消えるような演出が行われたり、可動物が画面上の全部または一部を遮蔽するような演出が行われたりするなど、演出態様が多様化してきている。そのため、演出表示装置９上の表示画面を見ていても、現在変動表示中の状態であるのか否か認識しにくい場合も生じている。そこで、この実施の形態では、演出表示装置９の表示画面の一部でさらに第４図柄の変動表示を行うことによって、第４図柄の状態を確認することにより現在変動表示中の状態であるのか否かを確実に認識可能としている。なお、第４図柄は、常に一定の動作で変動表示され、画面上から消えたり遮蔽物で遮蔽されたりすることはないため、常に視認することができる。

【００２４】

なお、第１特別図柄用の第４図柄と第２特別図柄用の第４図柄とを、第４図柄と総称することがあり、第１特別図柄用の第４図柄表示領域９ｃと第２特別図柄用の第４図柄表示領域９ｄを、第４図柄表示領域と総称することがある。

【００２５】

第４図柄の変動（可変表示）は、第４図柄表示領域９ｃ，９ｄを所定の表示色（例えば、青色）で一定の時間間隔で点灯と消灯とを繰り返す状態を継続することによって実現される。第１特別図柄表示器８ａにおける第１特別図柄の可変表示と、第１特別図柄用の第４図柄表示領域９ｃにおける第１特別図柄用の第４図柄の可変表示とは同期している。第２特別図柄表示器８ｂにおける第２特別図柄の可変表示と、第２特別図柄用の第４図柄表示領域９ｄにおける第２特別図柄用の第４図柄の可変表示とは同期している。同期とは、

可変表示の開始時点および終了時点が同じであって、可変表示の期間が同じであることをいう。

【００２６】

また、第１特別図柄表示器８ａにおいて大当り図柄が停止表示されるときには、第１特別図柄用の第４図柄表示領域９ｃにおいて大当りを想起させる表示色（はずれとは異なる表示色。例えば、はずれのときには青色で表示されるのに対して、大当りのときには赤色で表示される。また、第２特別図柄表示器８ｂにおいて大当り図柄が停止表示されるときには、第２特別図柄用の第４図柄表示領域９ｄにおいて大当りを想起させる表示色（はずれとは異なる表示色。例えば、はずれのときには青色で表示されるのに対して、大当りのときには赤色で表示される。

【００２７】

なお、この実施の形態では、第４図柄表示領域を演出表示装置９の表示画面の一部に設ける場合を示しているが、演出表示装置９とは別に、ランプやＬＥＤなどの発光体を用いて第４図柄表示領域を実現するようにしてもよい。この場合、例えば、第４図柄の変動（可変表示）を、２つのＬＥＤが交互に点灯する状態を継続することによって実現されるようにしてもよく、２つのＬＥＤのうちのいずれのＬＥＤが停止表示されたかによって大当り図柄が停止表示されたか否かを表すようにしてもよい。

【００２８】

演出表示装置９の右方には、識別情報としての第１特別図柄を可変表示する第１特別図柄表示器（第１可変表示部）８ａが設けられている。この実施の形態では、第１特別図柄表示器８ａは、０～９の数字を可変表示可能な簡易で小型の表示器（例えば７セグメントＬＥＤ）で実現されている。すなわち、第１特別図柄表示器８ａは、０～９の数字（または、記号）を可変表示するように構成されている。また、演出表示装置９の右方（第１特別図柄表示器８ａの右隣）には、識別情報としての第２特別図柄を可変表示する第２特別図柄表示器（第２可変表示部）８ｂも設けられている。第２特別図柄表示器８ｂは、０～９の数字を可変表示可能な簡易で小型の表示器（例えば７セグメントＬＥＤ）で実現されている。すなわち、第２特別図柄表示器８ｂは、０～９の数字（または、記号）を可変表示するように構成されている。

【００２９】

小型の表示器は、例えば方形状に形成されている。また、この実施の形態では、第１特別図柄の種類と第２特別図柄の種類とは同じ（例えば、ともに０～９の数字）であるが、種類が異なっていてもよい。また、第１特別図柄表示器８ａおよび第２特別図柄表示器８ｂは、それぞれ、例えば、００～９９の数字（または、２桁の記号）を可変表示するように構成されていてもよい。

【００３０】

以下、第１特別図柄と第２特別図柄とを特別図柄と総称することがあり、第１特別図柄表示器８ａと第２特別図柄表示器８ｂとを特別図柄表示器（可変表示部）と総称することがある。

【００３１】

なお、この実施の形態では、２つの特別図柄表示器８ａ，８ｂを備える場合を示しているが、遊技機は、特別図柄表示器を１つのみ備えるものであってもよい。

【００３２】

第１特別図柄の可変表示は、可変表示の実行条件である第１始動条件が成立（この実施の形態では、技球が第１始動入賞口１３を通過（入賞を含む）したこと）した後、可変表示の開始条件（例えば、第１保留記憶数が０でない場合であって、第１特別図柄および第２特別図柄の可変表示が実行されていない状態であり、かつ、大当り遊技が実行されていない状態）が成立したことにもとづいて開始され、可変表示時間（変動時間）が経過すると表示結果（停止図柄）を導出表示する。また、第２特別図柄の可変表示は、可変表示の実行条件である第２始動条件が成立（この実施の形態では、遊技球が第２始動入賞口１４を通過（入賞を含む）したこと）した後、可変表示の開始条件（例えば、第２保留記憶数

が０でない場合であって、第１特別図柄および第２特別図柄の可変表示が実行されていない状態であり、かつ、大当り遊技が実行されていない状態）が成立したことにもとづいて開始され、可変表示時間（変動時間）が経過すると表示結果（停止図柄）を導出表示する。なお、遊技球が通過するとは、入賞口やゲートなどのあらかじめ入賞領域として定められている領域を遊技球が通過したことであり、入賞口に遊技球が入った（入賞した）ことを含む概念である。また、表示結果を導出表示するとは、図柄（識別情報の例）を最終的に停止表示させることである。

【００３３】

演出表示装置９の下方には、流入した遊技球を振り分けるための振分装置２００が設けられている。また、演出表示装置９の下方には、左右に並ぶように第１始動入賞口１３および第２始動入賞口１４が設けられている。第１始動入賞口１３に入賞した遊技球は、遊技盤６の背面に導かれ、第１始動口スイッチ１３ａによって検出される。また、第２始動入賞口（第２始動口）１４に入賞した遊技球は、遊技盤６の背面に導かれ、第２始動口スイッチ１４ａによって検出される。

【００３４】

また、第２始動入賞口１４には、開放状態と閉鎖状態とに変化可能な可変入賞球装置１５が設けられている。可変入賞球装置１５は、ソレノイド１６によって開放状態とされる。可変入賞球装置１５が開放状態になることによって、遊技球が第２始動入賞口１４に始動入賞し易くなり、遊技者にとって有利な状態になる。なお、この実施の形態では、可変入賞球装置１５が閉鎖状態の場合であっても、開放状態と比較すると入賞しにくいものの振分装置２００を経由して第２始動入賞口１４に入賞可能であるように構成する場合を示しているが、可変入賞球装置１５が開放状態である場合にのみ第２始動入賞口１４に始動入賞可能に構成してもよい。

【００３５】

また、この実施の形態では、後述するように、遊技状態が確変状態（この実施の形態では、高確率状態に制御されるとともに時短状態（高ベース状態）にも制御される）である場合に、可変入賞球装置１５が開放状態となる頻度が高められ第２始動入賞口１４に始動入賞しやすくなる。そのため、この実施の形態では、遊技状態が確変状態に制御されている場合には、遊技者は遊技領域７の右方を狙って遊技球を発射するように発射操作を行う（いわゆる右打ち）方が有利になる。なお、遊技状態が確変状態に制御された場合には、例えば、演出表示装置９に「右を狙え！」などと表示して右打ちを示唆する表示を行うようにすることが望ましい。

【００３６】

以下、第１始動入賞口１３および第２始動入賞口１４を総称して始動入賞口または始動口ということがある。

【００３７】

なお、この実施の形態では、図１に示すように、第２始動入賞口１４に対してのみ開閉動作を行う可変入賞球装置１５が設けられているが、第１始動入賞口１３および第２始動入賞口１４のいずれについても開閉動作を行う可変入賞球装置が設けられている構成であってもよい。

【００３８】

また、遊技状態が確変状態や時短状態（高ベース状態）であっても一貫して遊技領域７の左方を狙って遊技球を発射させる仕様の遊技機である場合には、逆に左側の第１始動入賞口１３にのみ可変入賞球装置１５を設けるようにしてもよい。

【００３９】

第２特別図柄表示器８ｂの上方には、第２始動入賞口１４に入った有効入賞球数すなわち第２保留記憶数を表示する４つの表示器からなる第２特別図柄保留記憶表示器１８ｂが設けられている。第２特別図柄保留記憶表示器１８ｂは、有効始動入賞がある毎に、点灯する表示器の数を１増やす。そして、第２特別図柄表示器８ｂでの可変表示が開始される毎に、点灯する表示器の数を１減らす。

【００４０】
　また、第２特別図柄保留記憶表示器１８ｂのさらに上方には、第１始動入賞口１３に入った有効入賞球数すなわち第１保留記憶数（保留記憶を、始動記憶または始動入賞記憶ともいう。）を表示する４つの表示器からなる第１特別図柄保留記憶表示器１８ａが設けられている。第１特別図柄保留記憶表示器１８ａは、有効始動入賞がある毎に、点灯する表示器の数を１増やす。そして、第１特別図柄表示器８ａでの可変表示が開始される毎に、点灯する表示器の数を１減らす。
【００４１】
　また、演出表示装置９の表示画面の下部には、第１保留記憶数と第２保留記憶数との合計である合計数（合算保留記憶数）を表示する合算保留記憶表示部１８ｃが設けられている。そのように、この実施の形態では、合計数を表示する合算保留記憶表示部１８ｃが設けられていることによって、可変表示の開始条件が成立していない実行条件の成立数の合計を把握しやすくすることができる。なお、この実施の形態では、合算保留記憶表示部１８ｃにおいて、第１保留記憶と第２保留記憶とが第１始動入賞口１３および第２始動入賞口１４への入賞順に並べて表示されるとともに、第１保留記憶であるか第２保留記憶であるかを認識可能な態様で表示される（例えば、第１保留記憶は赤色で表示され、第２保留記憶は青色で表示される）。なお、合算保留記憶表示部１８ｃに代えて、第１保留記憶数を表示する第１保留記憶表示部と第２保留記憶数を表示する第２保留記憶表示部とを設けるように構成してもよい。
【００４２】
　演出表示装置９は、第１特別図柄表示器８ａによる第１特別図柄の可変表示時間中、および第２特別図柄表示器８ｂによる第２特別図柄の可変表示時間中に、装飾用（演出用）の図柄としての演出図柄の可変表示を行う。第１特別図柄表示器８ａにおける第１特別図柄の可変表示と、演出表示装置９における演出図柄の可変表示とは同期している。また、第２特別図柄表示器８ｂにおける第２特別図柄の可変表示と、演出表示装置９における演出図柄の可変表示とは同期している。また、第１特別図柄表示器８ａにおいて大当り図柄が停止表示されるときと、第２特別図柄表示器８ｂにおいて大当り図柄が停止表示されるときには、演出表示装置９において大当りを想起させるような演出図柄の組み合わせが停止表示される。
【００４３】
　また、図１に示すように、第１始動入賞口１３および第２始動入賞口１４の下方には、大入賞口を形成する特別可変入賞球装置２０が設けられている。特別可変入賞球装置２０は開閉板を備え、第１特別図柄表示器８ａに特定表示結果（大当り図柄）が導出表示されたときと、第２特別図柄表示器８ｂに特定表示結果（大当り図柄）が導出表示されたときに生起する特定遊技状態（大当り遊技状態）においてソレノイド２１によって開閉板が開放状態に制御されることによって、入賞領域となる大入賞口が開放状態になる。大入賞口に入賞した遊技球はカウントスイッチ２３で検出される。
【００４４】
　演出表示装置９の左方には、普通図柄を可変表示する普通図柄表示器１０が設けられている。この実施の形態では、普通図柄表示器１０は、０～９の数字を可変表示可能な簡易で小型の表示器（例えば７セグメントＬＥＤ）で実現されている。すなわち、普通図柄表示器１０は、０～９の数字（または、記号）を可変表示するように構成されている。また、小型の表示器は、例えば方形状に形成されている。なお、普通図柄表示器１０は、例えば、００～９９の数字（または、２桁の記号）を可変表示するように構成されていてもよい。また、普通図柄表示器１０は、７セグメントＬＥＤなどにかぎらず、例えば、所定の記号表示を点灯表示可能な表示器（例えば、「○」や「×」を交互に点灯表示可能な装飾ランプ）で構成されていてもよい。
【００４５】
　遊技領域７の左右にそれぞれ設けられたゲート３２を遊技球が通過しゲートスイッチ３２ａで検出されると、普通図柄表示器１０の表示の可変表示が開始される。そして、普通

図柄表示器１０における停止図柄が所定の図柄（当り図柄。例えば、図柄「７」。）である場合に、可変入賞球装置１５が所定回数、所定時間だけ開放状態になる。すなわち、可変入賞球装置１５の状態は、普通図柄の停止図柄が当り図柄である場合に、遊技者にとって不利な状態から有利な状態（第２始動入賞口１４に遊技球が入賞しやすい状態）に変化する。普通図柄表示器１０の近傍には、ゲート３２を通過した入賞球数を表示する４つのＬＥＤによる表示部を有する普通図柄保留記憶表示器４１が設けられている。ゲート３２への遊技球の通過がある毎に、すなわちゲートスイッチ３２ａによって遊技球が検出される毎に、普通図柄保留記憶表示器４１は点灯するＬＥＤを１増やす。そして、普通図柄表示器１０の可変表示が開始される毎に、点灯するＬＥＤを１減らす。さらに、通常状態に比べて大当りとすることに決定される確率が高い状態である確変状態（通常状態と比較して、特別図柄の変動表示結果として大当りと判定される確率が高められた状態）では、普通図柄表示器１０における停止図柄が当り図柄になる確率が高められるとともに、可変入賞球装置１５の開放時間と開放回数が高められる。なお、この実施の形態では、確変状態に制御されるときには、図柄の変動時間が短縮されている時短状態（特別図柄の可変表示時間が短縮される遊技状態）にも制御される。

【００４６】

遊技盤６の下部には、入賞しなかった打球（アウト球）が取り込まれるアウト口２６がある。また、遊技領域７の外側の左右上部および左右下部には、所定の音声出力として効果音や音声を発声する４つのスピーカ２７が設けられている。遊技領域７の外周には、前面枠に設けられた枠ＬＥＤ２８が設けられている。

【００４７】

遊技機には、遊技者が打球操作ハンドル５を操作することに応じて駆動モータを駆動し、駆動モータの回転力を利用して遊技球を遊技領域７に発射する打球発射装置（図示せず）が設けられている。打球発射装置から発射された遊技球は、遊技領域７を囲むように円形状に形成された打球レール（図示せず）を通って遊技領域７に入り、その後、遊技領域７を下りてくる。遊技球が第１始動入賞口１３に入り第１始動口スイッチ１３ａで検出されると、第１特別図柄の可変表示を開始できる状態であれば（例えば、特別図柄の可変表示が終了し、第１の開始条件が成立したこと）、第１特別図柄表示器８ａにおいて第１特別図柄の可変表示（変動）が開始されるとともに、演出表示装置９において演出図柄の可変表示が開始される。すなわち、第１特別図柄および演出図柄の可変表示は、第１始動入賞口１３への入賞に対応する。第１特別図柄の可変表示を開始できる状態でなければ、第１保留記憶数が上限値に達していないことを条件として、第１保留記憶数を１増やす。

【００４８】

遊技球が第２始動入賞口１４に入り第２始動口スイッチ１４ａで検出されると、第２特別図柄の可変表示を開始できる状態であれば（例えば、特別図柄の可変表示が終了し、第２の開始条件が成立したこと）、第２特別図柄表示器８ｂにおいて第２特別図柄の可変表示（変動）が開始されるとともに、演出表示装置９において演出図柄の可変表示が開始される。すなわち、第２特別図柄および演出図柄の可変表示は、第２始動入賞口１４への入賞に対応する。第２特別図柄の可変表示を開始できる状態でなければ、第２保留記憶数が上限値に達していないことを条件として、第２保留記憶数を１増やす。

【００４９】

この実施の形態では、大当り（１５Ｒ確変大当り、７Ｒ確変大当り、突然確変大当り）となった場合には、遊技状態を高確率状態（確変状態）に移行するとともに、遊技球が始動入賞しやすくなる（すなわち、特別図柄表示器８ａ，８ｂや演出表示装置９における可変表示の実行条件が成立しやすくなる）ように制御された遊技状態である高ベース状態に移行（この実施の形態では、時短状態に移行）する。高ベース状態である場合には、例えば、高ベース状態でない場合と比較して、可変入賞球装置１５が開放状態となる頻度が高められたり、可変入賞球装置１５が開放状態となる時間が延長されたりして、始動入賞しやすくなる。

【００５０】

なお、可変入賞球装置１５が開放状態となる時間を延長する（開放延長状態ともいう）のでなく、普通図柄表示器１０における停止図柄が当り図柄になる確率が高められる普通図柄確変状態に移行することによって、高ベース状態に移行してもよい。普通図柄表示器１０における停止図柄が所定の図柄（当り図柄）となると、可変入賞球装置１５が所定回数、所定時間だけ開放状態になる。この場合、普通図柄確変状態に移行制御することによって、普通図柄表示器１０における停止図柄が当り図柄になる確率が高められ、可変入賞球装置１５が開放状態となる頻度が高まる。従って、普通図柄確変状態に移行すれば、可変入賞球装置１５の開放時間と開放回数が高められ、始動入賞しやすい状態（高ベース状態）となる。すなわち、可変入賞球装置１５の開放時間と開放回数は、普通図柄の停止図柄が当り図柄であったり、特別図柄の停止図柄が確変図柄である場合等に高められ、遊技者にとって不利な状態から有利な状態（始動入賞しやすい状態）に変化する。なお、開放回数が高められることは、閉鎖状態から開放状態になることも含む概念である。

【００５１】

また、普通図柄表示器１０における普通図柄の変動時間（可変表示期間）が短縮される普通図柄時短状態に移行することによって、高ベース状態に移行してもよい。普通図柄時短状態では、普通図柄の変動時間が短縮されるので、普通図柄の変動が開始される頻度が高くなり、結果として普通図柄が当りとなる頻度が高くなる。従って、普通図柄が当たりとなる頻度が高くなることによって、可変入賞球装置１５が開放状態となる頻度が高くなり、始動入賞しやすい状態（高ベース状態）となる。

【００５２】

また、特別図柄や演出図柄の変動時間（可変表示期間）が短縮される時短状態に移行することによって、特別図柄や演出図柄の変動時間が短縮されるので、特別図柄や演出図柄の変動が開始される頻度が高くなり（換言すれば、保留記憶の消化が速くなる。）、無効な始動入賞が生じてしまう事態を低減することができる。従って、有効な始動入賞が発生しやすくなり、結果として、大当り遊技が行われる可能性が高まる。

【００５３】

さらに、上記に示した全ての状態（開放延長状態、普通図柄確変状態、普通図柄時短状態および特別図柄時短状態）に移行させることによって、始動入賞しやすくなる（高ベース状態に移行する）ようにしてもよい。また、上記に示した各状態（開放延長状態、普通図柄確変状態、普通図柄時短状態および特別図柄時短状態）のうちのいずれか複数の状態に移行させることによって、始動入賞しやすくなる（高ベース状態に移行する）ようにしてもよい。また、上記に示した各状態（開放延長状態、普通図柄確変状態、普通図柄時短状態および特別図柄時短状態）のうちのいずれか１つの状態にのみ移行させることによって、始動入賞しやすくなる（高ベース状態に移行する）ようにしてもよい。

【００５４】

次に、振分装置２００について説明する。図２は、振分装置２００を説明するための説明図である。図２に示すように、振分装置２００上部には、遊技球が流入可能な流入口２０１が設けられている。また、振分装置２００内部には、流入口２０１から振分装置２００内に流入した遊技球を左側通路２０３と右側通路２０４とのいずれかに振り分けるための振分部材２０２が設けられている。また、振分装置２００の下部には、左側通路２０３を通過した遊技球が振分装置２００から流出可能な左側流出口２０５と、右側通路２０４を通過した遊技球が振分装置２００から流出可能な右側流出口２０６とが設けられている。

【００５５】

図２（ａ）に示す例では、振分部材２０２によって振分装置２００内の右側通路２０４が遮蔽され、遊技球が左側通路２０３を通過可能な状態が示されている。図２（ａ）に示す状態において遊技球が流入口２０１から振分装置２００内に流入すると、図２（ｂ）に示すように、流入口２０１から流入した遊技球は振分部材２００によって左側通路２０３に振り分けられ、左側通路２０３を通過して左側流出口２０５から流出する。そして、左側流出口２０５の下方には第１始動入賞口１３が位置しているので、左側流出口２０５か

ら流出した遊技球は、第１始動入賞口１３に入賞する。

【００５６】

また、遊技球が左側通路２０３を通過するときに、振分部材２０２の回転軸部分に設けられている羽根部２０２ａに遊技球があたり、遊技球の自重によって羽根部２０２ａが押されることによって、図２（ａ）および図２（ｂ）に示すように、振分部材２０２が右側に倒れている状態から左側に倒れている状態に変化する。そのように変化することにより、図２（ｂ）に示すように、振分部材２０２によって振分装置２００内の左側通路２０３が遮蔽され、遊技球が右側通路２０４を通過可能な状態となる。

【００５７】

次いで、そのような状態において、図２（ｃ）に示すように、遊技球が流入口２０１から振分装置２００内に流入すると、図２（ｄ）に示すように、流入口２０１から流入した遊技球は振分部材２００によって右側通路２０４に振り分けられ、右側通路２０４を通過して右側流出口２０６から流出する。そして、右側流出口２０６の下方には第２始動入賞口１４が位置しているので、右側流出口２０６から流出した遊技球は、第２始動入賞口１４に入賞する。

【００５８】

また、遊技球が右側通路２０４を通過するときに、振分部材２０４の回転軸部分に設けられている羽根部２０２ａに遊技球があたり、遊技球の自重によって羽根部２０２ａが押されることによって、図２（ｃ）および図２（ｄ）に示すように、振分部材２０２が左側に倒れている状態から右側に倒れている状態に変化する。そのように変化することにより、図２（ｄ）に示すように、振分部材２０２によって振分装置２００内の右側通路２０４が遮蔽され、遊技球が左側通路２０３を通過可能な状態となる。

【００５９】

図２に示すような動作をすることにより、この実施の形態では、振分部材２０２によって振分装置２００に流入した遊技球が左側通路２０３と右側通路２０４とに交互に振り分けられ、第１始動入賞口１３と第２始動入賞口１４とに交互に入賞可能となる。

【００６０】

一方、この実施の形態では、振分装置２００の振分部材２０２の状態にかかわらず、可変入賞球装置１５が開放状態に制御されているときにも第２始動入賞口１４に遊技球が入賞可能である。図３は、可変入賞球装置１５が開放状態に制御されているときに第２始動入賞口１４に遊技球が入賞する形態を説明するための説明図である。図３に示すように、振分部材２０２によって振分装置２００内の右側通路２０４が遮蔽され、遊技球が左側通路２０３を通過可能な状態（右側通路２０４は通過不能または通過困難な状態）であっても、可変入賞球装置１５が開放状態に制御されていれば、振分装置２００の右方から遊技球が進入し第２始動入賞口１４に入賞することが可能である。

【００６１】

なお、この実施の形態では、図３に示すように、第２始動入賞口１４の左右両側に可変入賞球装置１５の羽根部品が設けられている場合を示しているが、可変入賞球装置１５が開放状態の場合に振分装置２００を経由せずに第２始動入賞口１４に入賞する場合には、図３に示すように振分装置２００の右方から遊技球が進入して入賞する場合が殆どであることから、可変入賞球装置１５の左側の羽根部品はなくてもよい。ただし、そのようにしてしまうと、振分装置２００の右方から進入した遊技球が第２始動入賞口１４の上を素通りして入賞できない場合も考えられるので、振分装置２００の右方から進入した遊技球の勢いを止めるための遮蔽部（例えば、合成樹脂製の突起状部品や釘）を第２始動入賞口の開口部の左側に設けるようにして、可変入賞球装置１５が開放中の第２始動入賞口１４に遊技球が入賞するように構成してもよい。

【００６２】

また、この実施の形態では、図２および図３に示したように、振分装置２００の下方に第１始動入賞口１３および第２始動入賞口１４を設けるように構成することによって、左側通路２０３に振り分けられた遊技球は略１００％で第１始動入賞口１３に入賞し、右側

通路２０４に振り分けられた遊技球は略１００％で第２始動入賞口１４に入賞する場合を示したが、左側通路２０３や右側通路２０４に振り分けられた遊技球が振分装置２００の外にこぼれる場合があるように構成し、左側通路２０３や右側通路２０４に振り分けられても必ずしも第１始動入賞口１３や第２始動入賞口１４に入賞しない場合があるようにしてもよい。例えば、図２および図３では左側流出口２０５および右側流出口２０６からダイレクトに真下に遊技球が落下するように構成したが、左側流出口２０５および右側流出口２０６に底面部材を設けて遊技者から見て奥側に一旦誘導されてから第１始動入賞口１３や第２始動入賞口１４に遊技球が流れるように構成するとともに、左側通路２０３や右側通路２０４の外側側面に開口部を設けて、左側通路２０３や右側通路２０４に振り分けられた遊技球の一部がその開口部から振分装置２００の外にこぼれて第１始動入賞口１３や第２始動入賞口１４に入賞しない場合があるように構成してもよい。

【００６３】

また、この実施の形態では、第１始動入賞口１３および第２始動入賞口１４が振分装置２００の外部に設けられている（具体的には、図２および図３に示すように、振分装置２００の下方に設けられている）場合を示したが、第１始動入賞口１３および第２始動入賞口１４が振分装置２００の内部に含まれているように構成してもよい。

【００６４】

また、この実施の形態では、振分装置２００に２つの通路２０３，２０４が設けられている場合を示したが、２つである場合にかぎらず、３以上の通路が設けられていてもよい。この場合、例えば、第１始動入賞口１３または第２始動入賞口１４のいずれか一方または両方に入賞可能な通路が複数存在するように構成してもよい。

【００６５】

また、この実施の形態では、振分装置２００の振分部材２０２が遊技球の自重によって物理的に左右に切り替わる場合を示したが、例えば、振分部材２０２を駆動するためのソレノイドやモータを設け、遊技制御用マイクロコンピュータ５６０からの制御によって振分部材２０２を交互に切り替えるように構成してもよい。

【００６６】

また、逆に、この実施の形態では、第２始動入賞口１４に設けられた可変入賞球装置１５についてはソレノイド１６を駆動制御することにより開閉動作を行う場合を示しているが、可変入賞球装置１５に振分部材２０２と類似の機構を適用して遊技球の自重により開閉動作を行うように構成するなど、可変入賞球装置１５を遊技制御用マイクロコンピュータ５６０による制御によらずに物理的に開閉動作するように構成してもよい。この場合、例えば、可変入賞球装置１５の開放状態を検出するためのセンサ（例えば、光センサ）を設けるようにし、そのセンサからの入力にもとづいて可変入賞球装置１５が開放状態であるか否かを判定するようにすればよい。

【００６７】

図４は、主基板（遊技制御基板）３１における回路構成の一例を示すブロック図である。なお、図４は、払出制御基板３７および演出制御基板８０等も示されている。主基板３１には、プログラムに従ってパチンコ遊技機１を制御する遊技制御用マイクロコンピュータ（遊技制御手段に相当）５６０、制御用クロック生成回路１１１、および乱数用クロック生成回路１１２が搭載されている。遊技制御用マイクロコンピュータ５６０は、ゲーム制御（遊技進行制御）用のプログラム等を記憶するＲＯＭ５４、ワークメモリとして使用される記憶手段としてのＲＡＭ５５、プログラムに従って制御動作を行うＣＰＵ５６を含む。また、この実施の形態では、ＲＯＭ５４およびＲＡＭ５５は遊技制御用マイクロコンピュータ５６０に内蔵されている。すなわち、遊技制御用マイクロコンピュータ５６０は、１チップマイクロコンピュータである。１チップマイクロコンピュータには、少なくともＣＰＵ５６のほかＲＡＭ５５が内蔵されていればよく、ＲＯＭ５４は外付けであっても内蔵されていてもよい。遊技制御用マイクロコンピュータ５６０には、さらに、ハードウェア乱数（ハードウェア回路が発生する乱数）を発生する乱数回路５０８ａ，５０８ｂが内蔵されている。

【００６８】
ここで、制御用クロック生成回路１１１は、遊技制御用マイクロコンピュータ５６０の外部にて、所定周波数の発振信号となる制御用クロックＣＣＬＫを生成する。制御用クロック生成回路１１１により生成された制御用クロックＣＣＬＫは、例えば、後述する図６に示すような遊技制御用マイクロコンピュータ５６０の制御用外部クロック端子を介してクロック回路５０２に供給される。乱数用クロック生成回路１１２は、遊技制御用マイクロコンピュータ５６０の外部にて、制御用クロックＣＣＬＫの発振周波数とは異なる所定周波数の発振信号となる乱数用クロックＲＣＬＫを生成する。乱数用クロック生成回路１１２により生成された乱数用クロックＲＣＬＫは、例えば、後述する図６に示すような遊技制御用マイクロコンピュータ５６０の乱数用外部クロック端子（ＲＣＫ端子）を介して乱数回路５０８ａ，５０８ｂに供給される。一例として、乱数用クロック生成回路１１２により生成される乱数用クロックＲＣＬＫの発振周波数は、制御用クロック生成回路１１１により生成される制御用クロックＣＣＬＫの発振周波数以下となるようにすればよい。あるいは、乱数用クロック生成回路１１２により生成される乱数用クロックＲＣＬＫの発振周波数は、制御用クロック生成回路１１１により生成される制御用クロックＣＣＬＫの発振周波数よりも高周波となるようにしてもよい。

【００６９】
なお、この実施の形態では、乱数用クロック生成回路１１２から専用の乱数用クロックＲＣＬＫを乱数回路５０８ａ，５０８ｂに入力する場合を示しているが、そのような態様にかぎられない。例えば、専用のクロックを用いるのではなく、制御用クロック生成回路１１１からの制御用クロックＣＣＬＫを遊技制御用マイクロコンピュータ５６０内部で乱数回路５０８ａ，５０８ｂに入力させるように構成してもよい。この場合、例えば、制御用クロックＣＣＬＫを分周した信号を用いて乱数回路５０８ａ，５０８ｂ内蔵の乱数カウンタ（後述する乱数生成回路５２５ａ，５２５ｂ）を更新させるようにしてもよい。また、この場合、乱数用クロック生成回路１１２は主基板３１上に設けなくてもよい。

【００７０】
また、ＲＡＭ５５は、その一部または全部が電源基板９１０において作成されるバックアップ電源によってバックアップされている不揮発性記憶手段としてのバックアップＲＡＭである。すなわち、遊技機に対する電力供給が停止しても、所定期間（バックアップ電源としてのコンデンサが放電してバックアップ電源が電力供給不能になるまで）は、ＲＡＭ５５の一部または全部の内容は保存される。特に、少なくとも、遊技状態すなわち遊技制御手段の制御状態に応じたデータ（特別図柄プロセスフラグなど）と未払出賞球数を示すデータは、バックアップＲＡＭに保存される。遊技制御手段の制御状態に応じたデータとは、停電等が生じた後に復旧した場合に、そのデータにもとづいて、制御状態を停電等の発生前に復旧させるために必要なデータである。また、制御状態に応じたデータと未払出賞球数を示すデータとを遊技の進行状態を示すデータと定義する。なお、この実施の形態では、ＲＡＭ５５の全部が、電源バックアップされているとする。

【００７１】
なお、遊技制御用マイクロコンピュータ５６０においてＣＰＵ５６がＲＯＭ５４に格納されているプログラムに従って制御を実行するので、以下、遊技制御用マイクロコンピュータ５６０（またはＣＰＵ５６）が実行する（または、処理を行う）ということは、具体的には、ＣＰＵ５６がプログラムに従って制御を実行することである。このことは、主基板３１以外の他の基板に搭載されているマイクロコンピュータについても同様である。ただし、後述するように、遊技機への電源投入時やシステムリセット発生時には、遊技制御用マイクロコンピュータ５６０は、プログラム管理エリアの設定内容に従って、内部リセット動作の設定や乱数回路５０８ａ，５０８ｂの設定、レジスタの設定などを行うのであるが、この設定動作については、遊技制御用マイクロコンピュータ５６０は、プログラムによらず、ハードウェア的に実行する。

【００７２】
また、ゲートスイッチ３２ａ、第１始動口スイッチ１３ａ、第２始動口スイッチ１４ａ

およびカウントスイッチ２３からの検出信号を遊技制御用マイクロコンピュータ５６０に与える入力ドライバ回路５８も主基板３１に搭載されている。また、可変入賞球装置１５を開閉するソレノイド１６、および大入賞口を形成する特別可変入賞球装置２０を開閉するソレノイド２１を遊技制御用マイクロコンピュータ５６０からの指令に従って駆動する出力回路５９も主基板３１に搭載されている。

【００７３】

また、遊技制御用マイクロコンピュータ５６０は、特別図柄を可変表示する第１特別図柄表示器８ａ、第２特別図柄表示器８ｂ、普通図柄を可変表示する普通図柄表示器１０、第１特別図柄保留記憶表示器１８ａ、第２特別図柄保留記憶表示器１８ｂおよび普通図柄保留記憶表示器４１の表示制御を行う。

【００７４】

なお、大当り遊技状態の発生を示す大当り情報等の情報出力信号を、ターミナル基板１６０を介して、ホールコンピュータ等の外部装置に対して出力する情報出力回路６４も主基板３１に搭載されている。

【００７５】

主基板３１と演出制御基板８０との間では、例えば主基板３１から中継基板７７を介して演出制御基板８０へと向かう単一方向のみでシリアル通信などを行うことにより、各種の演出制御コマンドが伝送される。この実施の形態では、演出制御基板８０に搭載されている演出制御手段（演出制御用マイクロコンピュータで構成される。）が、中継基板７７を介して遊技制御用マイクロコンピュータ５６０から演出内容を指示する演出制御コマンドを受信し、演出図柄を可変表示する演出表示装置９の表示制御を行う。

【００７６】

また、演出制御基板８０に搭載されている演出制御手段が、ランプドライバ基板３５を介して、遊技盤に設けられている装飾ＬＥＤ２５、および枠側に設けられている枠ＬＥＤ２８の表示制御を行うとともに、音声出力基板７０を介してスピーカ２７からの音出力の制御を行う。

【００７７】

図５は、中継基板７７、演出制御基板８０、ランプドライバ基板３５および音声出力基板７０の回路構成例を示すブロック図である。なお、図５に示す例では、ランプドライバ基板３５および音声出力基板７０には、マイクロコンピュータは搭載されていないが、マイクロコンピュータを搭載してもよい。また、ランプドライバ基板３５および音声出力基板７０を設けずに、演出制御に関して演出制御基板８０のみを設けてもよい。

【００７８】

演出制御基板８０は、演出制御用ＣＰＵ１０１、および演出図柄プロセスフラグ等の演出に関する情報を記憶するＲＡＭを含む演出制御用マイクロコンピュータ１００を搭載している。なお、ＲＡＭは外付けであってもよい。この実施の形態では、演出制御用マイクロコンピュータ１００におけるＲＡＭは電源バックアップされていない。演出制御基板８０において、演出制御用ＣＰＵ１０１は、内蔵または外付けのＲＯＭ（図示せず）に格納されたプログラムに従って動作し、中継基板７７を介して演出制御コマンドを受信する。また、演出制御用ＣＰＵ１０１は、演出制御コマンドにもとづいて、ＶＤＰ（ビデオディスプレイプロセッサ）１０９に演出表示装置９の表示制御を行わせる。

【００７９】

この実施の形態では、演出制御用マイクロコンピュータ１００と共動して演出表示装置９の表示制御を行うＶＤＰ１０９が演出制御基板８０に搭載されている。ＶＤＰ１０９は、演出制御用マイクロコンピュータ１００とは独立したアドレス空間を有し、そこにＶＲＡＭをマッピングする。ＶＲＡＭは、画像データを展開するためのバッファメモリである。そして、ＶＤＰ１０９は、ＶＲＡＭ内の画像データをフレームメモリを介して演出表示装置９に出力する。

【００８０】

演出制御用ＣＰＵ１０１は、受信した演出制御コマンドに従ってＣＧＲＯＭ（図示せず

）から必要なデータを読み出すための指令をＶＤＰ１０９に出力する。ＣＧＲＯＭは、演出表示装置９に表示されるキャラクタ画像データや動画像データ、具体的には、人物、文字、図形や記号等（演出図柄を含む）、および背景画像のデータをあらかじめ格納しておくためのＲＯＭである。ＶＤＰ１０９は、演出制御用ＣＰＵ１０１の指令に応じて、ＣＧＲＯＭから画像データを読み出す。そして、ＶＤＰ１０９は、読み出した画像データにもとづいて表示制御を実行する。

【００８１】

また、演出制御用ＣＰＵ１０１は、スティックコントローラ１２２のトリガボタン１２１に対する遊技者の操作行為を検出したことを示す情報信号としての操作検出信号を、トリガセンサ１２５から、入力ポート１０６を介して入力する。また、演出制御用ＣＰＵ１０１は、プッシュボタン１２０に対する遊技者の操作行為を検出したことを示す情報信号としての操作検出信号を、プッシュセンサ１２４から、入力ポート１０６を介して入力する。また、演出制御用ＣＰＵ１０１は、スティックコントローラ１２２の操作桿に対する遊技者の操作行為を検出したことを示す情報信号としての操作検出信号を、傾倒方向センサユニット１２３から、入力ポート１０６を介して入力する。また、演出制御用ＣＰＵ１０１は、出力ポート１０５を介してバイブレータ用モータ１２６に駆動信号を出力することにより、スティックコントローラ１２２を振動動作させる。

【００８２】

さらに、演出制御用ＣＰＵ１０１は、出力ポート１０５を介してランプドライバ基板３５に対してＬＥＤを駆動する信号を出力する。また、演出制御用ＣＰＵ１０１は、出力ポート１０４を介して音声出力基板７０に対して音番号データを出力する。

【００８３】

ランプドライバ基板３５において、ＬＥＤを駆動する信号は、入力ドライバ３５１を介してＬＥＤドライバ３５２に入力される。ＬＥＤドライバ３５２は、ＬＥＤを駆動する信号にもとづいて枠ＬＥＤ２８などの枠側に設けられている発光体に電流を供給する。また、遊技盤側に設けられている装飾ＬＥＤ２５に電流を供給する。

【００８４】

音声出力基板７０において、音番号データは、入力ドライバ７０２を介して音声合成用ＩＣ７０３に入力される。音声合成用ＩＣ７０３は、音番号データに応じた音声や効果音を発生し増幅回路７０５に出力する。増幅回路７０５は、音声合成用ＩＣ７０３の出力レベルを、ボリューム７０６で設定されている音量に応じたレベルに増幅した音声信号をスピーカ２７に出力する。音声データＲＯＭ７０４には、音番号データに応じた制御データが格納されている。音番号データに応じた制御データは、所定期間（例えば演出図柄の変動期間）における効果音または音声の出力態様を時系列的に示すデータの集まりである。

【００８５】

図６は、主基板３１に搭載された遊技制御用マイクロコンピュータ５６０の構成例を示している。図６に示す遊技制御用マイクロコンピュータ５６０は、例えば１チップマイクロコンピュータであり、外部バスインタフェース５０１と、クロック回路５０２と、照合用ブロック５０３と、固有情報記憶回路５０４と、演算回路５０５と、リセット／割込みコントローラ５０６と、ＣＰＵ（Central Processing Unit）５６と、ＲＯＭ（Read Only Memory）５４と、ＲＡＭ（Random Access Memory）５５と、フリーランカウンタ回路５０７と、乱数回路５０８ａ，５０８ｂと、タイマ回路５０９と、割り込みコントローラ５１０と、パラレル入力ポート５１１と、シリアル通信回路５１２と、パラレル出力ポート５１３と、アドレスデコード回路５１４とを備えて構成される。

【００８６】

また、遊技制御用マイクロコンピュータ５６０が搭載する乱数回路には、８ビット乱数を発生させる８ビット乱数回路５０８ａと、１６ビット乱数を発生させる１６ビット乱数回路５０８ｂとがある。なお、図６に示す例では、８ビット乱数回路５０８ａと、１６ビット乱数を発生させる１６ビット乱数回路５０８ｂとが１つずつ図示されているが、遊技制御用マイクロコンピュータ５６０は、８ビット乱数回路５０８ａと、１６ビット乱数を

発生させる１６ビット乱数回路５０８ｂとを、それぞれ４回路（４チャネル）ずつ搭載している。なお、この実施の形態では、８ビット乱数回路５０８ａの４つのチャネルをそれぞれＲＳ０～ＲＳ３と表現する場合があり、１６ビット乱数回路５０８ｂの４つのチャネルをそれぞれＲＬ０～ＲＬ３と表現する場合がある。

【００８７】

また、リセット／割り込みコントローラ５０６は、指定エリア外走行禁止（ＩＡＴ）回路５０６ａとウオッチドッグタイマ（ＷＤＴ）５０６ｂとを備える。ＩＡＴ回路５０６ａは、ユーザプログラムが指定エリア内で正しく実行されているか否かを監視する回路であり、指定エリア外でユーザプログラムが実行されたことを検出するとＩＡＴ発生信号を出力する機能を備える。また、ウオッチドッグタイマ５０６ｂは、設定期間ごとにタイムアウト信号を発生させる機能を備える。

【００８８】

図７は、遊技制御用マイクロコンピュータ５６０におけるアドレスマップの一例を示している。図７に示すように、アドレス０００００Ｈ～アドレス２ＦＦＦＨの領域は、遊技制御用マイクロコンピュータ５６０のＲＯＭ５４に割り当てられ、プログラムコード／データエリア（ユーザプログラムやデータを格納するエリア）とプログラム管理エリアとを含んでいる。図８は、ＲＯＭ５４におけるプログラム管理エリアの主要部分について、用途や内容の一例を示している。アドレスＦ０００Ｈ～アドレスＦ３ＦＦＨの領域は、遊技制御用マイクロコンピュータ５６０のＲＡＭ５５に割り当てられている。アドレスＦＥ００Ｈ～アドレスＦＥＢＦＨの領域は、遊技制御用マイクロコンピュータ５６０の内蔵レジスタに割り当てられる内蔵レジスタエリアである。図９～図１１は、内蔵レジスタエリアの主要部分について、用途や内容の一例を示している。アドレスＦＥＤ０Ｈ～アドレスＦＥＦＤＨの領域は、アドレスデコード回路５１４に割り当てられるＸＣＳ，ＸＣＳＥデコードエリアである。

【００８９】

プログラム管理エリアは、遊技制御用マイクロコンピュータ５６０がシステムリセット時に内部リセット動作の設定や乱数回路５０８ａ，５０８ｂの設定など各種設定を行うために必要な情報を格納する記憶領域である。図８に示すように、プログラム管理エリアには、ヘッダ（ＫＨＤＲ）、プログラムコードエンドアドレス（ＫＰＣＥ）、プログラムコードスタートアドレス２（ＫＰＣＳ２）、プログラムコードエンドアドレス２（ＫＰＣＥ２）、リセット設定（ＫＲＥＳ）、１６ビット乱数初期設定１（ＫＲＬ１）、１６ビット乱数初期設定２（ＫＲＬ２）、１６ビット乱数初期設定３（ＫＲＬ３）、８ビット乱数初期設定１（ＫＲＳ１）、８ビット乱数初期設定２（ＫＲＳ２）、セキュリティ時間設定（ＫＳＥＳ）、乱数クロック監視設定（ＫＲＣＳ）などが、含まれている。また、図９～図１１に示すように、内蔵レジスタエリアには、内部情報レジスタ（ＣＩＦ）や、乱数回路５０８ａ，５０８ｂで用いる各種レジスタなどが、含まれている。

【００９０】

プログラム管理エリアに記憶されるヘッダ（ＫＨＤＲ）は、プログラム管理エリアのスタートを示す８バイトのコード列の設定、および遊技制御用マイクロコンピュータ５６０における内部データの読出設定を示す。図１２は、ヘッダ（ＫＨＤＲ）における設定データと動作との対応関係を例示している。ここで、遊技制御用マイクロコンピュータ５６０では、ＲＯＭ読出防止機能と、バス出力マスク機能とを設定可能である。ＲＯＭ読出防止機能は、遊技制御用マイクロコンピュータ５６０が備えるＲＯＭ５４の記憶データについて、読出動作を許可または禁止する機能であり、読出禁止に設定された状態では、ＲＯＭ５４の記憶データを読み出すことができない。バス出力マスク機能は、外部バスインタフェース５０１に接続された外部装置から遊技制御用マイクロコンピュータ５６０の内部データに対する読出要求があった場合に、外部バスインタフェース５０１におけるアドレスバス出力、データバス出力および制御信号出力にマスクをかけることにより、外部装置から内部データの読み出しを不能にする機能である。図１２に示すように、プログラム管理エリアのスタートを示す８バイトのコード列として設定する設定データに対応して、ＲＯ

M読出防止機能やバス出力マスク機能の動作組合せが異なるように設定される。図１２に示す設定データのうち、ＲＯＭ読出が許可されるとともに、バス出力マスクが有効となる設定データは、バス出力マスク有効データともいう。また、ＲＯＭ読出が禁止されるとともに、バス出力マスクが有効となる設定データ（全て「００Ｈ」）は、ＲＯＭ読出禁止データともいう。ＲＯＭ読出が許可されるとともに、バス出力マスクが無効となる設定データは、バス出力マスク無効データともいう。

【００９１】

プログラム管理エリアに記憶されるプログラムコードエンドアドレス（ＫＰＣＥ）は、ユーザプログラムの００００Ｈから続くプログラムコードエリアの最終アドレスの設定を示す。図１３（Ａ）は、プログラムコードエンドアドレス（ＫＰＣＥ）における設定内容の一例を示している。

【００９２】

なお、この実施の形態では、アドレス００００Ｈ～アドレス２ＦＢＦＨまでのプログラムコード／データエリア内に２つのプログラムコードエリアを設定可能である。具体的には、１つ目のプログラムコードエリアは、アドレス００００Ｈからプログラムコードエンドアドレス（ＫＰＣＥ）で設定されるアドレスまでのエリアとして設定可能であり、２つ目のプログラムコードエリアは、プログラムコードスタートアドレス２（ＫＰＣＳ２）で設定されるアドレスからプログラムコードエンドアドレス２（ＫＰＣＥ２）で設定されるアドレスまでのエリアとして設定可能である。以下、１つ目のプログラムコードエリアに格納されるプログラムコードをプログラムコード１ともいい、２つ目のプログラムコードエリアに格納されるプログラムコードをプログラムコード２ともいう。

【００９３】

図１３（Ａ）に示すように、プログラムコードエンドアドレス（ＫＰＣＥ）のアドレス２ＦＤ３Ｈには、プログラムコード１の最終アドレスの下位アドレスが設定される。また、アドレス２ＦＤ４Ｈには、プログラムコード１の最終アドレスの上位アドレスが設定される。

【００９４】

プログラム管理エリアに記憶されるプログラムコードスタートアドレス２（ＫＰＣＳ２）は、ユーザプログラムが２つのブロックに分かれた場合の２つ目のプログラムコードエリアの先頭アドレスの設定を示す。図１３（Ｂ）は、プログラムコードスタートアドレス２（ＫＰＣＳ２）における設定内容の一例を示している。

【００９５】

図１３（Ｂ）に示すように、プログラムコードスタートアドレス２（ＫＰＣＳ２）のアドレス２ＦＤ５Ｈには、プログラムコード２の先頭アドレスの下位アドレスが設定される。また、アドレス２ＦＤ６Ｈには、プログラムコード２の先頭アドレスの上位アドレスが設定される。なお、プログラムコードエリアを２つに分けない場合には、プログラムコードスタートアドレス２（ＫＰＣＳ２）のアドレス２ＦＤ５ＨおよびアドレスＦＤ６Ｈにそれぞれ００００Ｈを設定するようにすればよい。

【００９６】

プログラム管理エリアに記憶されるプログラムコードエンドアドレス２（ＫＰＣＥ２）は、ユーザプログラムが２つのブロックに分かれた場合の２つ目のプログラムコードエリアの最終アドレスの設定を示す。図１３（Ｃ）は、プログラムコードエンドアドレス２（ＫＰＣＥ２）における設定内容の一例を示している。

【００９７】

図１３（Ｃ）に示すように、プログラムコードエンドアドレス２（ＫＰＣＥ２）のアドレス２ＦＤ７Ｈには、プログラムコード２の最終アドレスの下位アドレスが設定される。また、アドレス２ＦＤ８Ｈには、プログラムコード２の最終アドレスの上位アドレスが設定される。なお、プログラムコードエリアを２つに分けない場合には、プログラムコードエンドアドレス２（ＫＰＣＥ２）のアドレス２ＦＤ７ＨおよびアドレスＦＤ８Ｈにそれぞれ００００Ｈを設定するようにすればよい。

【００９８】
なお、図１３に示すプログラムコードエンドアドレス（ＫＰＣＥ）、プログラムコードスタートアドレス２（ＫＰＣＳ２）およびプログラムコードエンドアドレス２（ＫＰＣＥ２）の設定内容は、ＩＡＴ回路５０６ａによってユーザプログラムが指定エリア内で正しく実行されているか否かを監視する際に参照される。すなわち、ＩＡＴ回路５０６ａは、０００００Ｈからプログラムコードエンドアドレス（ＫＰＣＥ）で示されるアドレス、またはプログラムコードスタートアドレス２（ＫＰＣＳ２）で示されるアドレスからプログラムコードエンドアドレス２（ＫＰＣＥ２）で示されるアドレスまでの指定範囲でユーザプログラムが実行されているか否かを判定し、その指定範囲外でユーザプログラムが実行されていることを検出したことにもとづいてＩＡＴ信号を出力する。

【００９９】
プログラム管理エリアに記憶されるリセット設定（ＫＲＥＳ）は、内部リセット動作やウオッチドッグタイマ（ＷＤＴ）５０６ｂの動作許可／禁止の設定を示す。図１４は、リセット設定（ＫＲＥＳ）における設定内容の一例を示している。

【０１００】
リセット設定（ＫＲＥＳ）のビット［７］は、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号を入力したことや、ＩＡＴが発生したこと（ＩＡＴ回路５０６ａからのＩＡＴ信号を入力したとき）により内部リセットが発生したときの動作の設定を示している。図１４に示す例において、リセット設定（ＫＲＥＳ）のビット［７］におけるビット値が”０”であれば、タイムアウト信号やＩＡＴ信号を入力したときにユーザリセットが発生する。これに対して、リセット設定（ＫＲＥＳ）のビット［７］におけるビット値が”１”であれば、タイムアウト信号やＩＡＴ信号を入力したときにシステムリセットが発生する。

【０１０１】
リセット設定（ＫＲＥＳ）のビット［６］は、ウオッチドッグタイマ（ＷＤＴ）５０６ｂの起動方法の設定を示している。図１４に示す例において、リセット設定（ＫＲＥＳ）のビット［６］におけるビット値が”０”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にウオッチドッグタイマ（ＷＤＴ）５０６ｂが起動され時間計測が開始される。これに対して、リセット設定（ＫＲＥＳ）のビット［６］におけるビット値が”１”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）によりウオッチドッグタイマ（ＷＤＴ）５０６ｂが起動され時間計測が開始される。

【０１０２】
リセット設定（ＫＲＥＳ）のビット［５－４］は、ウオッチドッグタイマ（ＷＤＴ）５０６ｂの基準クロック信号の設定を示している。図１４に示す例では、リセット設定（ＫＲＥＳ）のビット［５－４］に”００”が設定された場合には、基準クロック信号として２１５×ＴＳＣＬＫが選択される。また、リセット設定（ＫＲＥＳ）のビット［５－４］に”０１”が設定された場合には、基準クロック信号として２１９×ＴＳＣＬＫが選択される。また、リセット設定（ＫＲＥＳ）のビット［５－４］に”１０”が設定された場合には、基準クロック信号として２２２×ＴＳＣＬＫが選択される。また、リセット設定（ＫＲＥＳ）のビット［５－４］に”１１”が設定された場合には、基準クロック信号として２２５×ＴＳＣＬＫが選択される。なお、ＳＣＬＫとは、遊技制御用マイクロコンピュータ５６０の内部システムクロックを示しており、ＴＳＣＬＫは、１／ＳＣＬＫを示している。

【０１０３】
リセット設定（ＫＲＥＳ）のビット［３－０］は、ウオッチドッグタイマ（ＷＤＴ）５０６ｂのタイムアウト時間の設定を示している。具体的には、ウオッチドッグタイマ（ＷＤＴ）５０６ｂのタイムアウト時間は、リセット設定（ＫＲＥＳ）のビット［５－４］で選択した基準クロックに、リセット設定（ＫＲＥＳ）のビット［３－０］で設定した設定値を乗算した値となる。例えば、リセット設定（ＫＲＥＳ）のビット［３－０］に”１０

００”を設定（すなわち、値「８」を設定）した場合、リセット設定（ＫＲＥＳ）のビット［５－４］に”００”を設定した場合には、タイムアウト時間は２１５×ＴＳＣＬＫ×８となり、リセット設定（ＫＲＥＳ）のビット［５－４］に”０１”を設定した場合には、タイムアウト時間は２１９×ＴＳＣＬＫ×８となり、リセット設定（ＫＲＥＳ）のビット［５－４］に”１０”を設定した場合には、タイムアウト時間は２２２×ＴＳＣＬＫ×８となり、リセット設定（ＫＲＥＳ）のビット［５－４］に”１１”を設定した場合には、タイムアウト時間は２２５×ＴＳＣＬＫ×８となる。また、リセット設定（ＫＲＥＳ）のビット［３－０］に”１１１１”を設定（すなわち、値「１５」を設定）した場合、リセット設定（ＫＲＥＳ）のビット［５－４］に”００”を設定した場合には、タイムアウト時間は２１５×ＴＳＣＬＫ×１５となり、リセット設定（ＫＲＥＳ）のビット［５－４］に”０１”を設定した場合には、タイムアウト時間は２１９×ＴＳＣＬＫ×１５となり、リセット設定（ＫＲＥＳ）のビット［５－４］に”１０”を設定した場合には、タイムアウト時間は２２２×ＴＳＣＬＫ×１５となり、リセット設定（ＫＲＥＳ）のビット［５－４］に”１１”を設定した場合には、タイムアウト時間は２２５×ＴＳＣＬＫ×１５となる。なお、図１４には、内部システムクロックが１０．０ＭＨｚと１２．０ＭＨｚである場合のタイムアウト時間の値の具体例もそれぞれ示されている。

【０１０４】

なお、ウオッチドッグタイマ（ＷＤＴ）５０６ｂを使用しないように設定する場合、図１４に示すように、リセット設定（ＫＲＥＳ）のビット［３－０］に”００００”を設定するようにすればよい。ただし、リセット設定（ＫＲＥＳ）のビット［３－０］に”００００”がセットされてウオッチドッグタイマ（ＷＤＴ）５０６ｂが使用禁止状態に設定された場合であっても、リセット設定（ＫＲＥＳ）のビット［７］の値を設定することにより、システムリセットするかユーザリセットとするかの設定を行うことは可能である。

【０１０５】

プログラム管理エリアに記憶される１６ビット乱数初期設定１（ＫＲＬ１）、１６ビット乱数初期設定２（ＫＲＬ２）および１６ビット乱数初期設定３（ＫＲＬ３）は、１６ビット乱数回路５０８ｂの設定を示す。図１５は、１６ビット乱数初期設定１（ＫＲＬ１）のにおける設定内容の一例を示している。また、図１６は、１６ビット乱数初期設定２（ＫＲＬ２）のにおける設定内容の一例を示している。さらに、図１７は、１６ビット乱数初期設定３（ＫＲＬ３）のにおける設定内容の一例を示している。

【０１０６】

まず、図１５を用いて、１６ビット乱数初期設定１（ＫＲＬ１）における設定内容を説明する。１６ビット乱数初期設定１（ＫＲＬ１）のビット「７」は、４チャネルの１６ビット乱数回路５０８ｂのうち、チャネル１の１６ビット乱数回路５０８ｂの起動方法の設定を示している。図１５に示す例において、１６ビット乱数初期設定１（ＫＲＬ１）のビット「７」におけるビット値が”０”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）により乱数の最大値設定が行われたことにより、チャネル１の１６ビット乱数回路５０８ｂが起動される。これに対して、１６ビット乱数初期設定１（ＫＲＬ１）のビット「７」におけるビット値が”１”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にチャネル１の１６ビット乱数回路５０８ｂが起動される。

【０１０７】

１６ビット乱数初期設定１（ＫＲＬ１）のビット「６」は、チャネル１の１６ビット乱数回路５０８ｂの更新クロックの設定を示している。図１５に示す例において、１６ビット乱数初期設定１（ＫＲＬ１）のビット「６」におけるビット値が”０”であれば、遊技制御用マイクロコンピュータ５６０の内部システムクロックを更新クロックとして用いる。これに対して、１６ビット乱数初期設定１（ＫＲＬ１）のビット「６」におけるビット値が”１”であれば、遊技制御用マイクロコンピュータ５６０の外部から入力された外部クロック信号を２分周した信号を更新クロックとして用いる。

【０１０８】

なお、この実施の形態では、既に説明した乱数用クロック生成回路１１２により生成された乱数用クロックＲＣＬＫを乱数用外部クロック端子（ＲＣＫ端子）を介して入力し、その乱数用クロックＲＣＬＫを２分周した信号を更新クロックとして用いるものとする。なお、このことは、他のチャネルの１６ビット乱数回路５０８ｂや８ビット乱数回路５０８ａについても同様である。

【０１０９】

１６ビット乱数初期設定１（ＫＲＬ１）のビット「５－４」は、チャネル１の１６ビット乱数回路５０８ｂが更新する乱数列を変更するか否かの設定を示している。図１５に示す例において、１６ビット乱数初期設定１（ＫＲＬ１）のビット「５－４」におけるビット値が”００”であれば、チャネル１の１６ビット乱数回路５０８ｂが更新する乱数列は変更されない。また、１６ビット乱数初期設定１（ＫＲＬ１）のビット「５－４」におけるビット値が”０１”であれば、チャネル１の１６ビット乱数回路５０８ｂが更新する乱数列をソフトウェア（ユーザプログラム）により変更できる。また、１６ビット乱数初期設定１（ＫＲＬ１）のビット「５－４」におけるビット値が”１０”であれば、チャネル１の１６ビット乱数回路５０８ｂが更新する乱数列が２周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。また、１６ビット乱数初期設定１（ＫＲＬ１）のビット「５－４」におけるビット値が”１１”であれば、チャネル１の１６ビット乱数回路５０８ｂが更新する乱数列が１周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。

【０１１０】

１６ビット乱数初期設定１（ＫＲＬ１）のビット「３」は、４チャネルの１６ビット乱数回路５０８ｂのうち、チャネル０の１６ビット乱数回路５０８ｂの起動方法の設定を示している。図１５に示す例において、１６ビット乱数初期設定１（ＫＲＬ１）のビット「３」におけるビット値が”０”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）により乱数の最大値設定が行われたことにより、チャネル０の１６ビット乱数回路５０８ｂが起動される。これに対して、１６ビット乱数初期設定１（ＫＲＬ１）のビット「３」におけるビット値が”１”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にチャネル０の１６ビット乱数回路５０８ｂが起動される。

【０１１１】

１６ビット乱数初期設定１（ＫＲＬ１）のビット「２」は、チャネル０の１６ビット乱数回路５０８ｂの更新クロックの設定を示している。図１５に示す例において、１６ビット乱数初期設定１（ＫＲＬ１）のビット「２」におけるビット値が”０”であれば、遊技制御用マイクロコンピュータ５６０の内部システムクロックを更新クロックとして用いる。これに対して、１６ビット乱数初期設定１（ＫＲＬ１）のビット「２」におけるビット値が”１”であれば、遊技制御用マイクロコンピュータ５６０の外部から入力された外部クロック信号を２分周した信号を更新クロックとして用いる。

【０１１２】

１６ビット乱数初期設定１（ＫＲＬ１）のビット「１－０」は、チャネル０の１６ビット乱数回路５０８ｂが更新する乱数列を変更するか否かの設定を示している。図１５に示す例において、１６ビット乱数初期設定１（ＫＲＬ１）のビット「１－０」におけるビット値が”００”であれば、チャネル０の１６ビット乱数回路５０８ｂが更新する乱数列は変更されない。また、１６ビット乱数初期設定１（ＫＲＬ１）のビット「１－０」におけるビット値が”０１”であれば、チャネル０の１６ビット乱数回路５０８ｂが更新する乱数列をソフトウェア（ユーザプログラム）により変更できる。また、１６ビット乱数初期設定１（ＫＲＬ１）のビット「１－０」におけるビット値が”１０”であれば、チャネル０の１６ビット乱数回路５０８ｂが更新する乱数列が２周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。また、１６ビット乱数初期設定１（ＫＲＬ１）のビット「１－０」におけるビット値が”１１”であれば、チャネル０の１６ビット乱数回路５０８ｂが更新する乱数列が１周目から自動的に変更され、以降、乱

数列が一巡するごとに自動的に乱数列が変更される。

【０１１３】

次に、図１６を用いて、１６ビット乱数初期設定２（ＫＲＬ２）における設定内容を説明する。１６ビット乱数初期設定２（ＫＲＬ２）のビット「７」は、４チャネルの１６ビット乱数回路５０８ｂのうち、チャネル３の１６ビット乱数回路５０８ｂの起動方法の設定を示している。図１６に示す例において、１６ビット乱数初期設定２（ＫＲＬ２）のビット「７」におけるビット値が”０”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）により乱数の最大値設定が行われたことにより、チャネル３の１６ビット乱数回路５０８ｂが起動される。これに対して、１６ビット乱数初期設定２（ＫＲＬ２）のビット「７」におけるビット値が”１”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にチャネル３の１６ビット乱数回路５０８ｂが起動される。

【０１１４】

１６ビット乱数初期設定２（ＫＲＬ２）のビット「６」は、チャネル３の１６ビット乱数回路５０８ｂの更新クロックの設定を示している。図１６に示す例において、１６ビット乱数初期設定２（ＫＲＬ２）のビット「６」におけるビット値が”０”であれば、遊技制御用マイクロコンピュータ５６０の内部システムクロックを更新クロックとして用いる。これに対して、１６ビット乱数初期設定２（ＫＲＬ２）のビット「６」におけるビット値が”１”であれば、遊技制御用マイクロコンピュータ５６０の外部から入力された外部クロック信号を２分周した信号を更新クロックとして用いる。

【０１１５】

１６ビット乱数初期設定２（ＫＲＬ２）のビット「５－４」は、チャネル３の１６ビット乱数回路５０８ｂが更新する乱数列を変更するか否かの設定を示している。図１６に示す例において、１６ビット乱数初期設定２（ＫＲＬ２）のビット「５－４」におけるビット値が”００”であれば、チャネル３の１６ビット乱数回路５０８ｂが更新する乱数列は変更されない。また、１６ビット乱数初期設定２（ＫＲＬ２）のビット「５－４」におけるビット値が”０１”であれば、チャネル３の１６ビット乱数回路５０８ｂが更新する乱数列をソフトウェア（ユーザプログラム）により変更できる。また、１６ビット乱数初期設定２（ＫＲＬ２）のビット「５－４」におけるビット値が”１０”であれば、チャネル３の１６ビット乱数回路５０８ｂが更新する乱数列が２周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。また、１６ビット乱数初期設定２（ＫＲＬ２）のビット「５－４」におけるビット値が”１１”であれば、チャネル３の１６ビット乱数回路５０８ｂが更新する乱数列が１周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。

【０１１６】

１６ビット乱数初期設定２（ＫＲＬ２）のビット「３」は、４チャネルの１６ビット乱数回路５０８ｂのうち、チャネル２の１６ビット乱数回路５０８ｂの起動方法の設定を示している。図１６に示す例において、１６ビット乱数初期設定２（ＫＲＬ２）のビット「３」におけるビット値が”０”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）により乱数の最大値設定が行われたことにより、チャネル２の１６ビット乱数回路５０８ｂが起動される。これに対して、１６ビット乱数初期設定２（ＫＲＬ２）のビット「３」におけるビット値が”１”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にチャネル２の１６ビット乱数回路５０８ｂが起動される。

【０１１７】

１６ビット乱数初期設定２（ＫＲＬ２）のビット「２」は、チャネル２の１６ビット乱数回路５０８ｂの更新クロックの設定を示している。図１６に示す例において、１６ビット乱数初期設定２（ＫＲＬ２）のビット「２」におけるビット値が”０”であれば、遊技制御用マイクロコンピュータ５６０の内部システムクロックを更新クロックとして用いる。これに対して、１６ビット乱数初期設定２（ＫＲＬ２）のビット「２」におけるビット

値が”１”であれば、遊技制御用マイクロコンピュータ５６０の外部から入力された外部クロック信号を２分周した信号を更新クロックとして用いる。

【０１１８】

１６ビット乱数初期設定２（ＫＲＬ２）のビット「１－０」は、チャネル２の１６ビット乱数回路５０８ｂが更新する乱数列を変更するか否かの設定を示している。図１６に示す例において、１６ビット乱数初期設定２（ＫＲＬ２）のビット「１－０」におけるビット値が”００”であれば、チャネル２の１６ビット乱数回路５０８ｂが更新する乱数列は変更されない。また、１６ビット乱数初期設定２（ＫＲＬ２）のビット「１－０」におけるビット値が”０１”であれば、チャネル２の１６ビット乱数回路５０８ｂが更新する乱数列をソフトウェア（ユーザプログラム）により変更できる。また、１６ビット乱数初期設定２（ＫＲＬ２）のビット「１－０」におけるビット値が”１０”であれば、チャネル２の１６ビット乱数回路５０８ｂが更新する乱数列が２周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。また、１６ビット乱数初期設定２（ＫＲＬ２）のビット「１－０」におけるビット値が”１１”であれば、チャネル２の１６ビット乱数回路５０８ｂが更新する乱数列が１周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。

【０１１９】

次に、図１７を用いて、１６ビット乱数初期設定３（ＫＲＬ３）のにおける設定内容を説明する。１６ビット乱数初期設定３（ＫＲＬ３）のビット「7」は、４チャネルの１６ビット乱数回路５０８ｂのうち、チャネル３の１６ビット乱数回路５０８ｂの１周目からのスタート値の設定を示している。図１７に示す例において、１６ビット乱数初期設定３（ＫＲＬ３）のビット「7」におけるビット値が”０”であれば、乱数更新の１周目のスタート値として０００１Ｈが用いられる。これに対して、１６ビット乱数初期設定３（ＫＲＬ３）のビット「7」におけるビット値が”１”であれば、乱数更新の１周目のスタート値として遊技制御用マイクロコンピュータ５６０のＩＤナンバをもとにした値が用いられる。遊技制御用マイクロコンピュータ５６０のＩＤナンバはチップごとに異なることから、スタート値としてＩＤナンバをもとにした値を用いることにより、乱数の更新タイミングを予測しにくくすることができ、乱数の更新タイミングを狙って不正に大当りを発生させるような行為を防止することができる。なお、ＩＤナンバをもとにした値として、ＩＤナンバそのものを用いてもよいし、ＩＤナンバに所定の演算（例えば、所定値を加算したり減算したりした値）を用いてもよい。

【０１２０】

１６ビット乱数初期設定３（ＫＲＬ３）のビット「6」は、チャネル３の１６ビット乱数回路５０８ｂのスタート値をシステムリセットごとに変更するか否かの設定を示している。図１７に示す例において、１６ビット乱数初期設定３（ＫＲＬ３）のビット「6」におけるビット値が”０”であれば、システムリセット時にスタート値の変更は行わない。これに対して、１６ビット乱数初期設定３（ＫＲＬ３）のビット「6」におけるビット値が”１”であれば、システムリセットごとにスタート値を変更する。

【０１２１】

１６ビット乱数初期設定３（ＫＲＬ３）のビット「5」は、４チャネルの１６ビット乱数回路５０８ｂのうち、チャネル２の１６ビット乱数回路５０８ｂの１周目からのスタート値の設定を示している。図１７に示す例において、１６ビット乱数初期設定３（ＫＲＬ３）のビット「5」におけるビット値が”０”であれば、乱数更新の１周目のスタート値として０００１Ｈが用いられる。これに対して、１６ビット乱数初期設定３（ＫＲＬ３）のビット「5」におけるビット値が”１”であれば、乱数更新の１周目のスタート値として遊技制御用マイクロコンピュータ５６０のＩＤナンバをもとにした値が用いられる。

【０１２２】

１６ビット乱数初期設定３（ＫＲＬ３）のビット「4」は、チャネル２の１６ビット乱数回路５０８ｂのスタート値をシステムリセットごとに変更するか否かの設定を示している。図１７に示す例において、１６ビット乱数初期設定３（ＫＲＬ３）のビット「4」に

おけるビット値が”０”であれば、システムリセット時にスタート値の変更は行わない。これに対して、１６ビット乱数初期設定３（ＫＲＬ３）のビット「６」におけるビット値が”１”であれば、システムリセットごとにスタート値を変更する。

【０１２３】

１６ビット乱数初期設定３（ＫＲＬ３）のビット「３」は、４チャネルの１６ビット乱数回路５０８ｂのうち、チャネル１の１６ビット乱数回路５０８ｂの１周目からのスタート値の設定を示している。図１７に示す例において、１６ビット乱数初期設定３（ＫＲＬ３）のビット「３」におけるビット値が”０”であれば、乱数更新の１周目のスタート値として０００１Ｈが用いられる。これに対して、１６ビット乱数初期設定３（ＫＲＬ３）のビット「３」におけるビット値が”１”であれば、乱数更新の１周目のスタート値として遊技制御用マイクロコンピュータ５６０のＩＤナンバをもとにした値が用いられる。

【０１２４】

１６ビット乱数初期設定３（ＫＲＬ３）のビット「２」は、チャネル１の１６ビット乱数回路５０８ｂのスタート値をシステムリセットごとに変更するか否かの設定を示している。図１７に示す例において、１６ビット乱数初期設定３（ＫＲＬ３）のビット「２」におけるビット値が”０”であれば、システムリセット時にスタート値の変更は行わない。これに対して、１６ビット乱数初期設定３（ＫＲＬ３）のビット「２」におけるビット値が”１”であれば、システムリセットごとにスタート値を変更する。

【０１２５】

１６ビット乱数初期設定３（ＫＲＬ３）のビット「１」は、４チャネルの１６ビット乱数回路５０８ｂのうち、チャネル０の１６ビット乱数回路５０８ｂの１周目からのスタート値の設定を示している。図１７に示す例において、１６ビット乱数初期設定３（ＫＲＬ３）のビット「１」におけるビット値が”０”であれば、乱数更新の１周目のスタート値として０００１Ｈが用いられる。これに対して、１６ビット乱数初期設定３（ＫＲＬ３）のビット「１」におけるビット値が”１”であれば、乱数更新の１周目のスタート値として遊技制御用マイクロコンピュータ５６０のＩＤナンバをもとにした値が用いられる。

【０１２６】

１６ビット乱数初期設定３（ＫＲＬ３）のビット「０」は、チャネル０の１６ビット乱数回路５０８ｂのスタート値をシステムリセットごとに変更するか否かの設定を示している。図１７に示す例において、１６ビット乱数初期設定３（ＫＲＬ３）のビット「０」におけるビット値が”０”であれば、システムリセット時にスタート値の変更は行わない。これに対して、１６ビット乱数初期設定３（ＫＲＬ３）のビット「０」におけるビット値が”１”であれば、システムリセットごとにスタート値を変更する。

【０１２７】

プログラム管理エリアに記憶される８ビット乱数初期設定１（ＫＲＳ１）および８ビット乱数初期設定２（ＫＲＳ２）は、８ビット乱数回路５０８ａの設定を示す。図１８は、８ビット乱数初期設定１（ＫＲＳ１）のにおける設定内容の一例を示している。また、図１９は、８ビット乱数初期設定２（ＫＲＳ２）のにおける設定内容の一例を示している。

【０１２８】

まず、図１８を用いて、８ビット乱数初期設定１（ＫＲＳ１）における設定内容を説明する。８ビット乱数初期設定１（ＫＲＳ１）のビット「７」は、４チャネルの８ビット乱数回路５０８ａのうち、チャネル１の８ビット乱数回路５０８ａの起動方法の設定を示している。図１８に示す例において、８ビット乱数初期設定１（ＫＲＳ１）のビット「７」におけるビット値が”０”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）により乱数の最大値設定が行われたことにより、チャネル１の８ビット乱数回路５０８ａが起動される。これに対して、８ビット乱数初期設定１（ＫＲＳ１）のビット「７」におけるビット値が”１”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にチャネル１の８ビット乱数回路５０８ａが起動される。

【０１２９】

８ビット乱数初期設定１（ＫＲＳ１）のビット「６」は、チャネル１の８ビット乱数回路５０８ａの更新クロックの設定を示している。図１８に示す例において、８ビット乱数初期設定１（ＫＲＳ１）のビット「６」におけるビット値が”０”であれば、遊技制御用マイクロコンピュータ５６０の内部システムクロックを更新クロックとして用いる。これに対して、８ビット乱数初期設定１（ＫＲＳ１）のビット「６」におけるビット値が”１”であれば、遊技制御用マイクロコンピュータ５６０の外部から入力された外部クロック信号を２分周した信号を更新クロックとして用いる。

【０１３０】

８ビット乱数初期設定１（ＫＲＳ１）のビット「５－４」は、チャネル１の８ビット乱数回路５０８ａが更新する乱数列を変更するか否かの設定を示している。図１８に示す例において、８ビット乱数初期設定１（ＫＲＳ１）のビット「５－４」におけるビット値が”００”であれば、チャネル１の８ビット乱数回路５０８ａが更新する乱数列は変更されない。また、８ビット乱数初期設定１（ＫＲＳ１）のビット「５－４」におけるビット値が”０１”であれば、チャネル１の８ビット乱数回路５０８ａが更新する乱数列をソフトウェア（ユーザプログラム）により変更できる。また、８ビット乱数初期設定１（ＫＲＳ１）のビット「５－４」におけるビット値が”１０”であれば、チャネル１の８ビット乱数回路５０８ａが更新する乱数列が２周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。また、８ビット乱数初期設定１（ＫＲＳ１）のビット「５－４」におけるビット値が”１１”であれば、チャネル１の８ビット乱数回路５０８ａが更新する乱数列が１周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。

【０１３１】

８ビット乱数初期設定１（ＫＲＳ１）のビット「３」は、４チャネルの８ビット乱数回路５０８ａのうち、チャネル０の８ビット乱数回路５０８ａの起動方法の設定を示している。図１８に示す例において、８ビット乱数初期設定１（ＫＲＳ１）のビット「３」におけるビット値が”０”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）により乱数の最大値設定が行われたことにより、チャネル０の８ビット乱数回路５０８ａが起動される。これに対して、８ビット乱数初期設定１（ＫＲＳ１）のビット「３」におけるビット値が”１”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にチャネル０の８ビット乱数回路５０８ａが起動される。

【０１３２】

８ビット乱数初期設定１（ＫＲＳ１）のビット「２」は、チャネル０の８ビット乱数回路５０８ａの更新クロックの設定を示している。図１８に示す例において、８ビット乱数初期設定１（ＫＲＳ１）のビット「２」におけるビット値が”０”であれば、遊技制御用マイクロコンピュータ５６０の内部システムクロックを更新クロックとして用いる。これに対して、８ビット乱数初期設定１（ＫＲＳ１）のビット「２」におけるビット値が”１”であれば、遊技制御用マイクロコンピュータ５６０の外部から入力された外部クロック信号を２分周した信号を更新クロックとして用いる。

【０１３３】

８ビット乱数初期設定１（ＫＲＳ１）のビット「１－０」は、チャネル０の８ビット乱数回路５０８ａが更新する乱数列を変更するか否かの設定を示している。図１８に示す例において、８ビット乱数初期設定１（ＫＲＳ１）のビット「１－０」におけるビット値が”００”であれば、チャネル０の８ビット乱数回路５０８ａが更新する乱数列は変更されない。また、８ビット乱数初期設定１（ＫＲＳ１）のビット「１－０」におけるビット値が”０１”であれば、チャネル０の８ビット乱数回路５０８ａが更新する乱数列をソフトウェア（ユーザプログラム）により変更できる。また、８ビット乱数初期設定１（ＫＲＳ１）のビット「１－０」におけるビット値が”１０”であれば、チャネル０の８ビット乱数回路５０８ａが更新する乱数列が２周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。また、８ビット乱数初期設定１（ＫＲＳ１）のビ

ット「１－０」におけるビット値が”１１”であれば、チャネル０の８ビット乱数回路５０８ａが更新する乱数列が１周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。

【０１３４】

次に、図１９を用いて、８ビット乱数初期設定２（ＫＲＳ２）における設定内容を説明する。８ビット乱数初期設定２（ＫＲＳ２）のビット「７」は、４チャネルの８ビット乱数回路５０８ａのうち、チャネル３の８ビット乱数回路５０８ａの起動方法の設定を示している。図１９に示す例において、８ビット乱数初期設定２（ＫＲＳ２）のビット「７」におけるビット値が”０”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）により乱数の最大値設定が行われたことにより、チャネル３の８ビット乱数回路５０８ａが起動される。これに対して、８ビット乱数初期設定２（ＫＲＳ２）のビット「７」におけるビット値が”１”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にチャネル３の８ビット乱数回路５０８ａが起動される。

【０１３５】

８ビット乱数初期設定２（ＫＲＳ２）のビット「６」は、チャネル３の８ビット乱数回路５０８ａの更新クロックの設定を示している。図１９に示す例において、８ビット乱数初期設定２（ＫＲＳ２）のビット「６」におけるビット値が”０”であれば、遊技制御用マイクロコンピュータ５６０の内部システムクロックを更新クロックとして用いる。これに対して、８ビット乱数初期設定２（ＫＲＳ２）のビット「６」におけるビット値が”１”であれば、遊技制御用マイクロコンピュータ５６０の外部から入力された外部クロック信号を２分周した信号を更新クロックとして用いる。

【０１３６】

８ビット乱数初期設定２（ＫＲＳ２）のビット「５－４」は、チャネル３の８ビット乱数回路５０８ａが更新する乱数列を変更するか否かの設定を示している。図１９に示す例において、８ビット乱数初期設定２（ＫＲＳ２）のビット「５－４」におけるビット値が”００”であれば、チャネル３の８ビット乱数回路５０８ａが更新する乱数列は変更されない。また、８ビット乱数初期設定２（ＫＲＳ２）のビット「５－４」におけるビット値が”０１”であれば、チャネル３の８ビット乱数回路５０８ａが更新する乱数列をソフトウェア（ユーザプログラム）により変更できる。また、８ビット乱数初期設定２（ＫＲＳ２）のビット「５－４」におけるビット値が”１０”であれば、チャネル３の８ビット乱数回路５０８ａが更新する乱数列が２周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。また、８ビット乱数初期設定２（ＫＲＳ２）のビット「５－４」におけるビット値が”１１”であれば、チャネル３の８ビット乱数回路５０８ａが更新する乱数列が１周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。

【０１３７】

８ビット乱数初期設定２（ＫＲＳ２）のビット「３」は、４チャネルの８ビット乱数回路５０８ａのうち、チャネル２の８ビット乱数回路５０８ａの起動方法の設定を示している。図１９に示す例において、８ビット乱数初期設定２（ＫＲＳ２）のビット「３」におけるビット値が”０”であれば、ユーザモードに移行した後、ソフトウェア（ユーザプログラム）により乱数の最大値設定が行われたことにより、チャネル２の８ビット乱数回路５０８ａが起動される。これに対して、８ビット乱数初期設定２（ＫＲＳ２）のビット「３」におけるビット値が”１”であれば、ユーザプログラムによらず、リセット発生時にユーザモードに移行したことにもとづいて自動的にチャネル２の８ビット乱数回路５０８ａが起動される。

【０１３８】

８ビット乱数初期設定２（ＫＲＳ２）のビット「２」は、チャネル２の８ビット乱数回路５０８ａの更新クロックの設定を示している。図１９に示す例において、８ビット乱数初期設定２（ＫＲＳ２）のビット「２」におけるビット値が”０”であれば、遊技制御用

マイクロコンピュータ５６０の内部システムクロックを更新クロックとして用いる。これに対して、８ビット乱数初期設定２（ＫＲＳ２）のビット「２」におけるビット値が”１”であれば、遊技制御用マイクロコンピュータ５６０の外部から入力された外部クロック信号を２分周した信号を更新クロックとして用いる。

【０１３９】

８ビット乱数初期設定２（ＫＲＳ２）のビット「１－０」は、チャネル２の８ビット乱数回路５０８ａが更新する乱数列を変更するか否かの設定を示している。図１９に示す例において、８ビット乱数初期設定２（ＫＲＳ２）のビット「１－０」におけるビット値が”００”であれば、チャネル２の８ビット乱数回路５０８ａが更新する乱数列は変更されない。また、８ビット乱数初期設定２（ＫＲＳ２）のビット「１－０」におけるビット値が”０１”であれば、チャネル２の８ビット乱数回路５０８ａが更新する乱数列をソフトウェア（ユーザプログラム）により変更できる。また、８ビット乱数初期設定２（ＫＲＳ２）のビット「１－０」におけるビット値が”１０”であれば、チャネル２の８ビット乱数回路５０８ａが更新する乱数列が２周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。また、８ビット乱数初期設定２（ＫＲＳ２）のビット「１－０」におけるビット値が”１１”であれば、チャネル２の８ビット乱数回路５０８ａが更新する乱数列が１周目から自動的に変更され、以降、乱数列が一巡するごとに自動的に乱数列が変更される。

【０１４０】

なお、８ビット乱数回路５０８ａに関しては、１６ビット乱数回路５０８ｂとは異なり、図１７に示したようなスタート値の設定を行う機能はない。

【０１４１】

プログラム管理エリアに記憶されるセキュリティ時間設定（ＫＳＥＳ）は、セキュリティモードを延長する時間の設定を示す。図２０は、セキュリティ時間設定（ＫＳＥＳ）における設定内容の一例を示している。

【０１４２】

セキュリティ時間設定（ＫＳＥＳ）のビット［７－６］は、セキュリティモード時間をランダムに延長する時間の設定を示している。図２０に示す例では、セキュリティ時間設定（ＫＳＥＳ）のビット［７－６］に”０１”が設定された場合には、セキュリティモード時間をランダムに延長するモードとして、ショートモードが設定され、具体的には、内部システムクロックが１０．０ＭＨｚである場合には０～０．８１６ｍｓの範囲の時間がランダムに延長され、内部システムクロックが１２．０ＭＨｚである場合には０～０．５１ｍｓの範囲の時間がランダムに延長される。また、セキュリティ時間設定（ＫＳＥＳ）のビット［７－６］に”１０”が設定された場合には、セキュリティモード時間をランダムに延長するモードとして、ミドルモードが設定され、具体的には、内部システムクロックが１０．０ＭＨｚである場合には０～２６．１１２ｍｓの範囲の時間がランダムに延長され、内部システムクロックが１２．０ＭＨｚである場合には０～１６．３２ｍｓの範囲の時間がランダムに延長される。また、セキュリティ時間設定（ＫＳＥＳ）のビット［７－６］に”１１”が設定された場合には、セキュリティモード時間をランダムに延長するモードとして、ロングモードが設定され、具体的には、内部システムクロックが１０．０ＭＨｚである場合には０～８３５．５８４ｍｓの範囲の時間がランダムに延長され、内部システムクロックが１２．０ＭＨｚである場合には０～５２２．２４ｍｓの範囲の時間がランダムに延長される。

【０１４３】

なお、セキュリティモード時間のランダム延長を行わないように設定する場合、図２０に示すように、セキュリティ時間設定（ＫＳＥＳ）のビット［７－６］に”００”を設定するようにすればよい。

【０１４４】

セキュリティ時間設定（ＫＳＥＳ）のビット［５］は、セキュリティモード時間を固定延長する時間の基準クロック信号の設定を示している。図２０に示す例では、セキュリテ

ィ時間設定（ＫＳＥＳ）のビット［５］に”０”が設定された場合には、基準クロック信号として２２２×ＴＳＣＬＫが選択される。また、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”１”が設定された場合には、基準クロック信号として２２４×ＴＳＣＬＫが選択される。

【０１４５】

セキュリティ時間設定（ＫＳＥＳ）のビット［４－０］は、セキュリティモード時間を固定で延長する時間の設定を示している。具体的には、セキュリティモード時間の固定延長時間は、セキュリティ時間設定（ＫＳＥＳ）のビット［５］で選択した基準クロックに、セキュリティ時間設定（ＫＳＥＳ）のビット［４－０］で設定した設定値を乗算した値となる。例えば、セキュリティ時間設定（ＫＳＥＳ）のビット［４－０］に”００００１”を設定（すなわち、値「１」を設定）した場合、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”０”を設定した場合には、固定延長時間は２２２×ＴＳＣＬＫ×１となり、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”１”を設定した場合には、固定延長時間は２２４×ＴＳＣＬＫ×１となる。また、セキュリティ時間設定（ＫＳＥＳ）のビット［４－０］に”０１０００”を設定（すなわち、値「８」を設定）した場合、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”０”を設定した場合には、固定延長時間は２２２×ＴＳＣＬＫ×８となり、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”１”を設定した場合には、固定延長時間は２２４×ＴＳＣＬＫ×８となる。また、セキュリティ時間設定（ＫＳＥＳ）のビット［４－０］に”１００００”を設定（すなわち、値「１６」を設定）した場合、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”０”を設定した場合には、固定延長時間は２２２×ＴＳＣＬＫ×１６となり、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”１”を設定した場合には、固定延長時間は２２４×ＴＳＣＬＫ×１６となる。また、セキュリティ時間設定（ＫＳＥＳ）のビット［４－０］に”１１１１１”を設定（すなわち、値「３１」を設定）した場合、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”０”を設定した場合には、固定延長時間は２２２×ＴＳＣＬＫ×３１となり、セキュリティ時間設定（ＫＳＥＳ）のビット［５］に”１”を設定した場合には、固定延長時間は２２４×ＴＳＣＬＫ×３１となる。なお、図２０には、内部システムクロックが１０．０ＭＨｚと１２．０ＭＨｚである場合の固定延長時間の値の具体例もそれぞれ示されている。

【０１４６】

なお、セキュリティモード時間の固定延長を行わないように設定する場合、図２０に示すように、セキュリティ時間設定（ＫＳＥＳ）のビット［４－０］に”０００００”を設定するようにすればよい。

【０１４７】

図２０に示すように、セキュリティモード時間は、セキュリティ時間設定（ＫＳＥＳ）のビット［７－６］の設定によるランダム延長と、セキュリティ時間設定（ＫＳＥＳ）のビット［５－０］の設定による固定延長との２種類の方法で延長設定が可能である。そして、これら２種類の方法で設定された時間の加算時間が最終的なセキュリティモード時間の延長時間となる。

【０１４８】

プログラム管理エリアに記憶される乱数クロック監視設定（ＫＲＣＳ）は、乱数用外部クロック端子（ＲＣＫ端子）から入力された外部クロック信号の監視周波数の設定を示す。図２１は、乱数クロック監視設定（ＫＲＣＳ）における設定内容の一例を示している。

【０１４９】

乱数クロック監視設定（ＫＲＣＳ）のビット［７－２］は、固定ビット（すなわち、特に設定に使用しないビット）であり、全ビット必ず”０”を設定するものとする。

【０１５０】

乱数クロック監視設定（ＫＲＣＳ）のビット［１－０］は、乱数を更新するためのクロックとして、乱数用外部クロック端子（ＲＣＫ端子）から入力された外部クロック信号を選択した場合に、その入力クロックの周波数異常の検出対象とする周波数の設定を示して

いる。図２１に示す例では、乱数クロック監視設定（ＫＲＣＳ）のビット［１－０］に”００”が設定された場合には、監視周波数としてＳＣＬＫ（内部システムクロック）の周波数未満を設定する。また、乱数クロック監視設定（ＫＲＣＳ）のビット［１－０］に”０１”が設定された場合には、監視周波数としてＳＣＬＫ（内部システムクロック）／２の周波数未満を設定する。また、乱数クロック監視設定（ＫＲＣＳ）のビット［１－０］に”１０”が設定された場合には、監視周波数としてＳＣＬＫ（内部システムクロック）／２２の周波数未満を設定する。また、乱数クロック監視設定（ＫＲＣＳ）のビット［１－０］に”１１”が設定された場合には、監視周波数としてＳＣＬＫ（内部システムクロック）／２３の周波数未満を設定する。

【０１５１】

なお、乱数用外部クロック端子（ＲＣＫ端子）から入力された外部クロック信号の監視周波数の異常を検出した場合には、後述する内部情報レジスタ（ＣＩＦ）のビット３に”１”がセットされる。

【０１５２】

なお、この実施の形態では、遊技制御用マイクロコンピュータ５６０は、８ビット乱数回路５０８ａと１６ビット乱数回路５０８ｂとのうち、１６ビット乱数回路５０８ｂの動作異常（外部クロック周波数異常および更新異常）を検出する機能を備えている。具体的には、遊技制御用マイクロコンピュータ５６０は、乱数更新用クロックとして乱数用外部クロック端子（ＲＣＫ端子）から入力された外部クロック信号が選択されている場合に、乱数クロック監視設定（ＫＲＣＳ）で設定されている監視周波数にもとづいて、外部クロック信号の周波数が低下したか否かを検出し、外部クロック信号の周波数が低下（外部クロック周波数異常）を検出した場合には、後述する内部情報レジスタ（ＣＩＦ）のビット３に”１”をセットする。

【０１５３】

また、遊技制御用マイクロコンピュータ５６０は、１６ビット乱数回路５０８ｂの乱数の更新状態を監視する機能を備え、更新状態に異常を検出すると（例えば、乱数値が同じ値のまま更新されなくなったり、通常は乱数値が１つずつカウントアップされていくのに乱数値のカウント値がいきなり２以上の値増加した状態を検出したりすると）、内部情報レジスタ（ＣＩＦ）のビット７～４のうちの対応するビットに”１”をセットする。

【０１５４】

なお、この実施の形態では、乱数クロック監視設定（ＫＲＣＳ）を用いて設定を行うことによって、１６ビット乱数回路５０８ｂの動作異常の検出に関して、監視対象の外部クロック信号の監視周波数を設定する場合を示しているが、外部クロック周波数異常の検出自体を行うか否かを設定可能に構成したり、更新異常の検出自体を行うか否かを設定可能に構成したりしてもよい。この場合、外部クロック周波数異常の検出自体を行うか否かの設定と、更新異常の検出自体を行うか否かの設定とをそれぞれ独立して行えるように構成してもよいし、両方の設定を一括して有効とするか無効とするかのみ行えるように構成してもよい。

【０１５５】

なお、外部クロック周波数異常の検出自体を行うか否かや更新異常の検出自体を行うか否かを設定可能とするためには、例えば、乱数回路自体を起動するか否かを設定するようにし、乱数回路を起動しないように設定した場合には、事実上、外部クロック周波数異常の検出や更新異常の検出を行えないので、外部クロック周波数異常の検出や更新異常の検出を行わないように設定したといえる。このように、外部クロック周波数異常の検出自体を行うか否かや更新異常の検出自体を行うか否かの設定は、乱数回路自体を起動するか否かを設定することによって実現することも含む概念である。

【０１５６】

また、この実施の形態では、乱数用クロック生成回路１１２から専用の乱数用クロックＲＣＬＫを乱数回路５０８ａ，５０８ｂに外部入力する場合を示しているが、例えば、制御用クロック生成回路１１１からの制御用クロックＣＣＬＫを外部入力する場合など専用

の乱数用クロックＲＣＬＫ以外のクロックを外部入力する場合であっても、外部クロック周波数異常の検出や更新異常の検出を行うことが可能である。なお、乱数回路の更新異常の検出に関しては、乱数用クロック生成回路１１２から専用の乱数用クロックＲＣＬＫ用いて乱数更新する場合と、制御用クロック生成回路１１１からの制御用クロックＣＣＬＫなど他のクロックを用いて乱数更新する場合とのいずれか一方の場合のみ設定可能に構成してもよい。

【０１５７】

また、この実施の形態では、外部クロック周波数の異常の検出を行い、遊技制御用マイクロコンピュータ５６０の内部システムクロックＳＣＬＫの周波数については特に異常の検出を行っていないが、それは次のような理由による。すなわち、乱数更新に内部システムクロックＳＣＬＫを用いる場合には、内部システムクロックＳＣＬＫに異常が発生しているような状況では、ＣＰＵ５６自体の動作が停止している筈であるので、ＣＰＵ５６が動作しているのに乱数の更新だけが停止しているような事態が生じる場合がなく、何らかの問題が生じるおそれがない。これに対して、乱数更新に外部クロック信号を用いる場合には、ＣＰＵ５６が動作しているのに乱数の更新だけが停止しているような事態が生じる可能性があり弊害が生じるおそれがあるためである。

【０１５８】

図６に示す遊技制御用マイクロコンピュータ５６０が備える外部バスインタフェース５０１は、遊技制御用マイクロコンピュータ５６０を構成するチップの外部バスと内部バスとのインタフェース機能や、アドレスバス、データバスおよび各制御信号の方向制御機能などを有するバスインタフェースである。例えば、外部バスインタフェース５０１は、遊技制御用マイクロコンピュータ５６０に外付けされた外部メモリや外部入出力装置などに接続され、これらの外部装置との間でアドレス信号やデータ信号、各種の制御信号などを送受信するものであればよい。

【０１５９】

遊技制御用マイクロコンピュータ５６０が備えるクロック回路５０２は、例えば制御用外部クロック端子ＥＸに入力される発振信号を２分周することなどにより、内部システムクロックＳＣＬＫを生成する回路である。なお、生成された内部システムクロックは、外部出力端子（ＣＬＫＯ端子）から外部に出力される。

【０１６０】

遊技制御用マイクロコンピュータ５６０が備える照合用ブロック５０３は、外部の照合機と接続し、チップの照合を行う機能を備える。

【０１６１】

遊技制御用マイクロコンピュータ５６０が備える固有情報記憶回路５０４は、例えば遊技制御用マイクロコンピュータ５６０の内部情報となる複数種類の固有情報を記憶する回路である。一例として、固有情報記憶回路５０４は、ＲＯＭコード、チップ個別ナンバー、ＩＤナンバーといった３種類の固有情報を記憶する。ＲＯＭ５４コードは、ＲＯＭ５４の所定領域における記憶データから生成される４バイトの数値であり、生成方法の異なる４つの数値が準備されればよい。チップ個別ナンバーは、遊技制御用マイクロコンピュータ５６０の製造時に付与される４バイトの番号であり、遊技制御用マイクロコンピュータ５６０を構成するチップ毎に異なる数値を示している。ＩＤナンバーは、遊技制御用マイクロコンピュータ５６０の製造時に付与される８バイトの番号であり、遊技制御用マイクロコンピュータ５６０を構成するチップ毎に異なる数値を示している。ここで、チップ個別ナンバーはユーザプログラムから読み取ることができる一方、ＩＤナンバーはユーザプログラムから読み取ることができないように設定されていればよい。なお、固有情報記憶回路５０４は、例えばＲＯＭ５４の所定領域を用いることなどにより、ＲＯＭ５４に含まれるようにしてもよい。あるいは、固有情報記憶回路５０４は、例えばＣＰＵ５６の内蔵レジスタを用いることなどにより、ＣＰＵ５６に含まれるようにしてもよい。

【０１６２】

遊技制御用マイクロコンピュータ５６０が備える演算回路５０５は、乗算および除算を

行う回路である。

【０１６３】

遊技制御用マイクロコンピュータ５６０が備えるリセット／割込みコントローラ５０６は、遊技制御用マイクロコンピュータ５６０の内部や外部にて発生する各種リセット、割込み要求を制御するためのものである。リセット／割込みコントローラ５０６が制御するリセットには、システムリセットとユーザリセットが含まれている。システムリセットは、外部システムリセット端子ＸＳＲＳＴに一定の期間にわたりローレベル信号が入力されたときに発生するリセットである。なお、この実施の形態では、リセット設定（ＫＲＥＳ）の設定により、ウォッチドッグタイマ（ＷＤＴ）のタイムアウト信号が発生したときや、指定エリア外走行禁止（ＩＡＴ）が発生したときにも、システムリセットが発生することがある。ユーザリセットは、ウォッチドッグタイマ（ＷＤＴ）のタイムアウト信号が発生したことや、指定エリア外走行禁止（ＩＡＴ）が発生したことなど、所定の要因により発生するリセットである。

【０１６４】

リセット／割込みコントローラ５０６が制御する割込みには、ノンマスカブル割込みＮＭＩとマスカブル割込みＩＮＴが含まれている。ノンマスカブル割込みＮＭＩは、ＣＰＵ５６の割込み禁止状態でも無条件に受け付けられる割込みであり、外部ノンマスカブル割込み端子ＸＮＭＩ（入力ポートＰＩ６と兼用）に一定の期間にわたりローレベル信号が入力されたときに発生する割込みである。マスカブル割込みＩＮＴは、ＣＰＵ５６の設定命令により、割込み要求の受け付けを許可／禁止できる割込みであり、優先順位設定による多重割込みの実行が可能である。マスカブル割込みＩＮＴの要因としては、外部マスカブル割込み端子ＸＩＮＴ（入力ポートＰＩ５と兼用）に一定の期間にわたりローレベル信号が入力されたこと、タイマ回路５０９にてタイムアウトが発生したこと、シリアル通信回路５１２にてデータ受信またはデータ送信による割込み要因が発生したこと、乱数回路５０８ａ，５０８ｂにて乱数値となる数値データの取込による割込み要因が発生したことなど、複数種類の割込み要因があらかじめ定められていればよい。

【０１６５】

リセット／割込みコントローラ５０６は、図９～図１１に示すような遊技制御用マイクロコンピュータ５６０が備える内蔵レジスタのうち、内部情報レジスタＣＩＦ（アドレスＦＥ２５Ｈ）などを用いて、割込みの制御やリセットの管理を行う。内部情報レジスタＣＩＦは、直前に発生したリセット要因を管理したり、乱数更新状態、乱数更新クロックを外部クロックとした場合の入力周波数の状態を読み取るためのレジスタである。

【０１６６】

図２２（Ａ）は、内部情報レジスタＣＩＦの構成例を示している。図２２（Ｂ）は、内部情報レジスタＣＩＦに格納される内部情報データの各ビットにおける設定内容の一例を示している。内部情報レジスタＣＩＦのビット番号［７］に格納される内部情報データＲＬ３ＥＲは、チャネル３の１６ビット乱数回路５０８ｂが更新する１６ビット乱数ＲＬ３の更新状態の異常を示す。図２２（Ｂ）に示す例では、１６ビット乱数ＲＬ３の更新異常が検知されないときに、内部情報データＲＬ３ＥＲのビット値が“０”となる一方、１６ビット乱数ＲＬ３の更新異常が検知されたときには、そのビット値が“１”となる。内部情報レジスタＣＩＦのビット番号［６］に格納される内部情報データＲＬ２ＥＲは、チャネル２の１６ビット乱数回路５０８ｂが更新する１６ビット乱数ＲＬ２の更新状態の異常を示す。図２２（Ｂ）に示す例では、１６ビット乱数ＲＬ２の更新異常が検知されないときに、内部情報データＲＬ２ＥＲのビット値が“０”となる一方、１６ビット乱数ＲＬ２の更新異常が検知されたときには、そのビット値が“１”となる。内部情報レジスタＣＩＦのビット番号［５］に格納される内部情報データＲＬ１ＥＲは、チャネル１の１６ビット乱数回路５０８ｂが更新する１６ビット乱数ＲＬ１の更新状態の異常を示す。図２２（Ｂ）に示す例では、１６ビット乱数ＲＬ１の更新異常が検知されないときに、内部情報データＲＬ１ＥＲのビット値が“０”となる一方、１６ビット乱数ＲＬ１の更新異常が検知されたときには、そのビット値が“１”となる。内部情報レジスタＣＩＦのビット番号［

４］に格納される内部情報データＲＬ０ＥＲは、チャネル０の１６ビット乱数回路５０８ｂが更新する１６ビット乱数ＲＬ０の更新状態の異常を示す。図２２（Ｂ）に示す例では、１６ビット乱数ＲＬ０の更新異常が検知されないときに、内部情報データＲＬ０ＥＲのビット値が“０”となる一方、１６ビット乱数ＲＬ０の更新異常が検知されたときには、そのビット値が“１”となる。なお、内部情報レジスタＣＩＦのビット番号［７－４」は、初期値として”０”が設定されている。

【０１６７】

内部情報レジスタＣＩＦのビット番号［３］に格納される内部情報データＲＣＥＲは、乱数更新用クロックとして乱数用外部クロック端子（ＲＣＫ端子）から入力された外部クロック信号が選択されている場合に、その外部クロック信号の周波数異常を示す。図２２（Ｂ）に示す例では、外部クロック信号の周波数異常が検知されないときに、内部情報データＲＣＥＲのビット値が“０”となる一方、外部クロック信号の周波数異常が検知されたときには、そのビット値が“１”となる。なお、内部情報レジスタＣＩＦのビット番号［３」は、初期値として”０”が設定されている。

【０１６８】

内部情報レジスタＣＩＦのビット番号［２］に格納される内部情報データＳＲＳＦは、直前に発生したリセット要因がシステムリセットであることを示す。図２２（Ｂ）に示す例では、直前のリセット要因がシステムリセットではないときに（システムリセット未発生）、内部情報データＳＲＳＦのビット値が“０”となる一方、システムリセットであるときには（システムリセット発生）、そのビット値が“１”となる。なお、内部情報レジスタＣＩＦのビット番号［２」は、初期値として”１”が設定されている。

【０１６９】

内部情報レジスタＣＩＦのビット番号［１］に格納される内部情報データＷＤＴＦは、直前に発生したリセット要因がウオッチドッグタイマ（ＷＤＴ）５０６ｂからタイムアウト信号を入力したことによるユーザリセットであることを示す。図２２（Ｂ）に示す例では、直前のリセット要因がタイムアウト信号によるユーザリセットではないときに（タイムアウト信号によるユーザリセット未発生）、内部情報データＷＤＴＦのビット値が“０”となる一方、タイムアウト信号によるユーザリセットであるときには（タイムアウト信号によるユーザリセット発生）、そのビット値が“１”となる。なお、内部情報レジスタＣＩＦのビット番号［１」は、初期値として”０”が設定されている。

【０１７０】

内部情報レジスタＣＩＦのビット番号［０］に格納される内部情報データＩＡＴＦは、直前に発生したリセット要因がＩＡＴ回路５０６ａからのＩＡＴ発生信号を入力したことによるユーザリセットであることを示す。図２２（Ｂ）に示す例では、直前のリセット要因がＩＡＴ発生信号によるユーザリセットではないときに（ＩＡＴ発生信号によるユーザリセット未発生）、内部情報データＩＡＴＦのビット値が“０”となる一方、ＩＡＴ発生信号によるユーザリセットであるときには（ＩＡＴ発生信号によるユーザリセット発生）、そのビット値が“１”となる。なお、内部情報レジスタＣＩＦのビット番号［０」は、初期値として”０”が設定されている。

【０１７１】

遊技制御用マイクロコンピュータ５６０が備えるＣＰＵ５６は、ＲＯＭ５４から読み出した制御コードにもとづいてユーザプログラム（ゲーム制御用の遊技制御処理プログラム）を実行することにより、パチンコ遊技機１における遊技制御を実行する制御用ＣＰＵである。こうした遊技制御が実行されるときには、ＣＰＵ５６がＲＯＭ５４から固定データを読み出す固定データ読出動作や、ＣＰＵ５６がＲＡＭ５５に各種の変動データを書き込んで一時記憶させる変動データ書込動作、ＣＰＵ５６がＲＡＭ５５に一時記憶されている各種の変動データを読み出す変動データ読出動作、ＣＰＵ５６が外部バスインタフェース５０１やパラレル入力ポート５１１、シリアル通信回路５１２などを介して遊技制御用マイクロコンピュータ５６０の外部から各種信号の入力を受け付ける受信動作、ＣＰＵ５６が外部バスインタフェース５０１やシリアル通信回路５１２、パラレル出力ポート５１３

などを介して遊技制御用マイクロコンピュータ５６０の外部へと各種信号を出力する送信動作等も行われる。

【０１７２】

遊技制御用マイクロコンピュータ５６０が備えるＲＯＭ５４には、ユーザプログラム（ゲーム制御用の遊技制御処理プログラム）を示す制御コードや固定データ等が記憶されている。

【０１７３】

遊技制御用マイクロコンピュータ５６０が備えるＲＡＭ５５は、ゲーム制御用のワークエリアを提供する。ここで、ＲＡＭ５５の少なくとも一部は、電源基板９１０において作成されるバックアップ電源によってバックアップされているバックアップＲＡＭであればよい。すなわち、パチンコ遊技機１への電力供給が停止しても、所定期間はＲＡＭ５５の少なくとも一部の内容が保存される。

【０１７４】

また、遊技制御用マイクロコンピュータ５６０は、フリーランカウンタ回路５０７として、８ビットのフリーランカウンタを４チャネル搭載している。

【０１７５】

遊技制御用マイクロコンピュータ５６０が備える乱数回路５０８ａ，５０８ｂは、８ビット乱数や１６ビット乱数といった、所定の更新範囲を有する乱数値となる数値データを生成する回路である。この実施の形態では、乱数回路５０８ａ，５０８ｂのうち１６ビット乱数回路５０８ｂが生成するハードウェア乱数は、大当りとするか否かを判定するための大当り判定用乱数（ランダムＲ）として用いられる。なお、ＣＰＵ５６は、乱数回路５０８ａ，５０８ｂから抽出した数値データにもとづき、乱数回路５０８ａ，５０８ｂとは異なるランダムカウンタを用いて、ソフトウェアによって各種の数値データを加工あるいは更新することで、遊技に用いられる乱数値の全部または一部を示す数値データをカウントするようにしてもよい。あるいは、ＣＰＵ５６は、乱数回路５０８ａ，５０８ｂを用いることなく、ソフトウェアによって大当り判定用乱数などの乱数値を示す数値データの一部をカウント（更新）するようにしてもよい。一例として、ハードウェアとなる乱数回路５０８ａ，５０８ｂからＣＰＵ５６により抽出された数値データを、ソフトウェアにより加工することで、大当り判定用乱数（ランダムＲ）を示す数値データが更新され、それ以外の乱数値（例えば、大当り種別判定用乱数や、変動パターン種別決定用乱数、変動パターン決定用乱数）を示す数値データは、ＣＰＵ５６がランダムカウンタなどを用いてソフトウェアにより更新すればよい。

【０１７６】

図２３は、８ビット乱数回路５０８ａの一構成例を示すブロック図である。また、図２４は、１６ビット乱数回路５０８ｂの一構成例を示すブロック図である。８ビット乱数回路５０８ａおよび１６ビット乱数回路５０８ｂは、図２３および図２４に示すように、乱数列変更選択回路５２３ａ，５２３ｂ、乱数生成回路５２５ａ，５２５ｂ、乱数列変更回路５２６ａ，５２６ｂ、および最大値比較回路５２７ａ，５２７ｂを備えて構成される。また、１６ビット乱数回路５０８ｂは、図２４に示すように、８ビット乱数回路５０８ａが備える構成要素に加えて、乱数スタート値選択回路５３５を備える。さらに、１６ビット乱数回路５０８ｂは、図２４に示すように、８ビット乱数回路５０８ａが備える構成要素に加えて、乱数生成回路５２５ｂが更新監視回路５３７を含む。

【０１７７】

また、図２４に示す例では、１６ビット乱数回路５０８ｂの回路部分の構成のみを示し、乱数列変更レジスタ５２２および最大値設定レジスタ５２４ｂ以外の１６ビット乱数回路５０８ｂが用いる各レジスタについては記載を省略している。なお、具体的には、１６ビット乱数回路５０８ｂは、図２３に示すＲＳハードラッチ選択レジスタ５２８ａ，５２８ｂに代えて図９に示すＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）～ＲＬ３ハードラッチ選択レジスタ（ＲＬ３ＬＳ）を用い、図２３に示すＲＳ０ハードラッチ乱数値レジスタ５２９ａ～ＲＳ３ハードラッチ乱数値レジスタ５２９ｄに代えて図１０および図

１１に示すＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）～ＲＬ３ハードラッチ乱数値レジスタ１（ＲＬ３ＨＶ１）を用い、図２３に示すＲＳハードラッチフラグレジスタ５３０に代えて図１０に示すＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）～ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）を用い、図２３に示すＲＳ割り込み制御レジスタ５３１に代えて図９に示すＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）～ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）を用い、図２３に示すＲＳ０ソフトラッチ乱数値レジスタ５３３ａ～ＲＳ３ソフトラッチ乱数値レジスタ５３３ｄに代えて図１０に示すＲＬ０ソフトラッチ乱数値レジスタ（ＲＬ０ＳＶ）～ＲＬ３ソフトラッチ乱数値レジスタ（ＲＬ３ＳＶ）を用いる。また、１６ビット乱数回路５０８ｂは、乱数値ソフトラッチレジスタ５３２および乱数ソフトラッチフラグレジスタ５３４については、８ビット乱数回路５０８ａと兼用で同じレジスタを用いる。

【０１７８】

また、８ビット乱数回路５０８ａは、既に説明したプログラム管理エリアに設けられた８ビット乱数初期設定５２１ａ（図８に示す８ビット乱数初期設定１（ＫＲＳ１）および８ビット乱数初期設定２（ＫＲＳ２））の設定内容に従って動作する。

【０１７９】

また、１６ビット乱数回路５０８ｂは、既に説明したプログラム管理エリアに設けられた１６ビット乱数初期設定５２１ｂ（図８に示す１６ビット乱数初期設定１（ＫＲＬ１）～１６ビット乱数初期設定２（ＫＲＬ２））の設定内容に従って動作する。また、１６ビット乱数回路５０８ｂは、８ビット乱数回路５０８ａの機能に加えて、乱数スタート値選択回路５３５が１６ビット乱数初期設定５３６（図８に示す１６ビット乱数初期設定３（ＫＲＬ３））の設定内容に従って動作することにより、１周目の乱数値のスタート値を変更する機能を備えている（図１７参照）。

【０１８０】

また、１６ビット乱数回路５０８ｂは、乱数生成回路５２５ｂが更新監視回路５３７を含んでおり、８ビット乱数回路５０８ａの機能に加えて、更新監視回路５３７が動作することにより外部クロック周波数異常および更新異常を検出する機能を備えている（図２１参照）。なお、この実施の形態では、１つの更新監視回路５３７により外部クロック周波数異常および更新異常の両方を検出する場合を示しているが、外部クロック周波数異常を検出する監視回路と更新異常を検出する監視回路とを別々に備えてもよい。

【０１８１】

なお、８ビット乱数回路５０８ａも更新監視回路を備えるように構成し、８ビット乱数回路５０８ａの外部クロック周波数異常および更新異常を検出可能に構成するようにしてもよい。

【０１８２】

また、乱数列変更レジスタ５２２は、図９に示すような遊技制御用マイクロコンピュータ５６０の内蔵レジスタに含まれる乱数列変更レジスタＲＤＳＣに対応している。なお、乱数列変更レジスタＲＤＳＣとして、８ビット乱数回路５０８ａおよび１６ビット乱数回路５０８ｂの各チャネルで共通のレジスタが用いられる。

【０１８３】

また、最大値設定レジスタ５２４ａ，５２４ｂは、図９に示すような遊技制御用マイクロコンピュータ５６０の内蔵レジスタに含まれるＲＳ０最大値設定レジスタ（ＲＳ０ＭＸ）～ＲＳ３最大値設定レジスタ（ＲＳ３ＭＸ）に対応している（１６ビット乱数回路５０８ｂの場合は、ＲＬ０最大値設定レジスタ（ＲＬ０ＭＸ）～ＲＬ３最大値設定レジスタ（ＲＬ３ＭＸ）に対応している）。

【０１８４】

また、ハードラッチ選択レジスタ５２８ａは、図９に示すような遊技制御用マイクロコンピュータ５６０の内蔵レジスタに含まれるＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）に対応している。また、ハードラッチ選択レジスタ５２８ｂは、図９に示すような遊技制御用マイクロコンピュータ５６０の内蔵レジスタに含まれるＲＳハードラッチ選択レ

ジスタ１（ＲＳＬＳ１）に対応している。なお、１６ビット乱数回路５０８ｂの場合は、図９に示すＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）～ＲＬ３ハードラッチ選択レジスタ３（ＲＬ３ＬＳ）に対応している。

【０１８５】

また、ＲＳ０ハードラッチ乱数値レジスタ５２９ａ～ＲＳ３ハードラッチ乱数値レジスタ５２９ｄは、図１１に示すような遊技制御用マイクロコンピュータ５６０の内蔵レジスタに含まれるＲＳ０ハードラッチ乱数値レジスタ（ＲＳ０ＨＶ）～ＲＳ３ハードラッチ乱数値レジスタ（ＲＳ３ＨＶ）に対応している。なお、１６ビット乱数回路５０８ｂの場合は、図１０に示すＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）～ＲＬ１ハードラッチ乱数値レジスタ１（ＲＬ１ＨＶ１）および図１１に示すＲＬ２ハードラッチ乱数値レジスタ０（ＲＬ２ＨＶ０）～ＲＬ３ハードラッチ乱数値レジスタ１（ＲＬ３ＨＶ１）に対応している。

【０１８６】

また、ＲＳハードラッチフラグレジスタ５３０は、図１０に示すＲＳハードラッチフラグレジスタ（ＲＳＨＦ）に対応している。なお、１６ビット乱数回路５０８ｂの場合は、図１０に示すＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）～ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）に対応している。

【０１８７】

また、ＲＳ割り込み制御レジスタ５３１は、図９に示すＲＳ割り込み制御レジスタ（ＲＳＩＣ）に対応している。なお、１６ビット乱数回路５０８ｂの場合は、図９に示すＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）～ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）に対応している。

【０１８８】

また、乱数ソフトラッチレジスタ５３２は、図９に示す乱数ソフトラッチレジスタ（ＲＤＳＬ）に対応している。なお、ソフトラッチレジスタＲＤＳＬとして、８ビット乱数回路５０８ａおよび１６ビット乱数回路５０８ｂの各チャネルで共通のレジスタが用いられる。

【０１８９】

また、ＲＳ０ソフトラッチ乱数値レジスタ５３３ａ～ＲＳ３ソフトラッチ乱数値レジスタ５３３ｄは、は、図１０に示すＲＳ０ソフトラッチ乱数値レジスタ（ＲＳ０ＳＶ）～ＲＳ３ソフトラッチ乱数値レジスタ（ＲＳ３ＳＶ）に対応している。なお、１６ビット乱数回路５０８ｂの場合は、図１０に示すＲＬ０ソフトラッチ乱数値（ＲＬ０ＳＶ）～ＲＬ３ソフトラッチ乱数値（ＲＬ３ＳＶ）に対応している。

【０１９０】

また、乱数ソフトラッチフラグレジスタ５３４は、図９に示す乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）に対応している。なお、乱数ソフトラッチフラグレジスタＲＤＳＦとして、８ビット乱数回路５０８ａおよび１６ビット乱数回路５０８ｂの各チャネルで共通のレジスタが用いられる。

【０１９１】

乱数列変更選択回路５２３ａ，５２３ｂは、図１８や図１９に示す８ビット乱数初期設定１（ＫＲＳ１）や８ビット乱数初期設定２（ＫＲＳ２）の設定内容に従って（１６ビット乱数回路５０８ｂの場合には、図１５や図１６に示す１６ビット乱数初期設定１（ＫＲＬ１）や１６ビット乱数初期設定２（ＫＲＬ２）の設定内容に従って）、乱数列の変更方法として、「変更しない」、「ソフトウェアで変更」、「２周目から自動で変更」または「１周目から自動で変更」のうちのいずれかを選択する。そして、「ソフトウェアで変更」、「２周目から自動で変更」または「１周目から自動で変更」のいずれかに選択した場合には、その選択方法に従って乱数列変更回路５２６ａ，５２６ｂに乱数列を変更させる。また、「変更しない」を選択した場合には乱数列を変更させる制御を行わない。

【０１９２】

乱数列変更回路５２６ａ，５２６ｂは、乱数生成回路５２５ａ，５２５ｂにより生成さ

れた数値データの順列を、乱数列変更選択回路５２３ａ，５２３ｂの指示に従って変更可能とする回路である。例えば、乱数列変更回路５２６ａ，５２６ｂは、「ソフトウェアで変更」が指示された場合には、乱数生成回路５２５ａ，５２５ｂが更新する乱数列をソフトウェア（ユーザプログラム）により変更する。また、例えば、乱数列変更回路５２６ａ，５２６ｂは、「２周目から自動で変更」が指示された場合には、乱数生成回路５２５ａ，５２５ｂが更新する乱数列を２周目から自動的に変更し、以降、乱数列が一巡するごとに自動的に乱数列を変更する。また、例えば、乱数列変更回路５２６ａ，５２６ｂは、「１周目から自動で変更」が指示された場合には、乱数生成回路５２５ａ，５２５ｂが更新する乱数列を１周目から自動的に変更し、以降、乱数列が一巡するごとに自動的に乱数列を変更する。

【０１９３】

乱数生成回路５２５ａ，５２５ｂは、例えば８ビットのカウンタ（１６ビット乱数回路５０８ｂの場合は１６ビットのカウンタ）などから構成され、乱数更新クロック信号などの入力にもとづき、数値データを更新可能な所定の範囲において所定の初期値から所定の最終値まで循環的に更新する回路である。例えば乱数生成回路５２５ａ，５２５ｂは、乱数更新クロック信号における立ち下がりエッジに応答して、「０」から「２５５」までの範囲内で設定された初期値から「２５５」まで１ずつ加算するように数値データをカウントアップして行く（１６ビット乱数回路５０８ｂの場合には、「０」から「６５５３５」までの範囲内で設定された初期値から「６５５３５」まで１ずつ加算するように数値データをカウントアップして行く）。そして、「２５５」までカウントアップした後には、「０」から初期値よりも１小さい最終値となる数値まで１ずつ加算するようにカウントアップすることで、数値データを循環的に更新する。

【０１９４】

最大値比較回路５２７ａ，５２７ｂは、図１８や図１９に示す８ビット乱数初期設定１（ＫＲＳ１）や８ビット乱数初期設定２（ＫＲＳ２）の設定内容に従って（１６ビット乱数回路５０８ｂの場合には、図１５や図１６に示す１６ビット乱数初期設定１（ＫＲＬ１）や１６ビット乱数初期設定２（ＫＲＬ２）の設定内容に従って）、乱数生成回路５２５ａ，５２５ｂが生成する乱数値の最大値を設定する。

【０１９５】

図２５（Ａ）は、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）の構成例を示している。図２５（Ｂ）は、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）に格納されるデータの各ビットにおける設定内容の一例を示している。ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット番号［７］に格納されるデータＲＬ０１ＲＦは、ＲＬ０ハードラッチ乱数値レジスタ１（ＲＬ０ＨＶ１）に、外部端子入力により、１６ビット乱数ＲＬ０の値を取り込む際の条件の設定を示している。図２５（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＬ０１ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＬ０１ＲＦは、初期値として”０”が設定されている。

【０１９６】

なお、この実施の形態では、プログラム管理エリアや内蔵レジスタのレジスタに関して、具体的には、プログラム管理エリアなどの対応するビットを”０”または”１”のいずれかの値としておくことにより、その対応するビットの値が読み込まれて、読み込まれた”０”または”１”の値が遊技制御用マイクロコンピュータ５６０の制御レジスタにハードウェア的に書き込まれることにより各種の設定が行われる。例えば、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット７については、そのビット７から読み込まれた値が”０”であれば、遊技制御用マイクロコンピュータ５６０の制御レジスタにハードウェア的に”０”が書き込まれることによりＲＬ０ハードラッチ乱数値レジスタ１（ＲＬ０ＨＶ１）から値を読み込まないと次の値をラッチしないように設定され、そのビット７から読み込まれた値が”１”であれば、遊技制御用マイクロコンピュータ５６０の制御レ

ジスタにハードウェア的に”１”が書き込まれることによりＲＬ０ハードラッチ乱数値レジスタ１（ＲＬ０ＨＶ１）から値を読み込まなくても次の値をラッチするように設定される。このことは、他のプログラム管理エリアの各設定項目や内蔵レジスタの各レジスタの各ビットに関しても同様である。

【０１９７】

ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット番号［６－４］に格納されるデータＲＬ０１ＬＳ０～ＲＬ０１ＬＳ２は、ＲＬ０ハードラッチ乱数値レジスタ１（ＲＬ０ＨＶ１）に、どの外部端子入力により、１６ビット乱数ＲＬ０の値を取り込むかの設定を示している。図２５（Ｂ）に示す例では、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット番号［６－４］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット番号［６－４］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＬ０１ＬＳ０～ＲＬ０１ＬＳ２は、初期値として”０００”が設定されている。

【０１９８】

ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット番号［３］に格納されるデータＲＬ００ＲＦは、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）に、外部端子入力により、１６ビット乱数ＲＬ０の値を取り込む際の条件の設定を示している。図２５（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＬ００ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＬ００ＲＦは、初期値として”０”が設定されている。

【０１９９】

ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット番号［２－０］に格納されるデータＲＬ００ＬＳ０～ＲＬ００ＬＳ２は、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）に、どの外部端子入力により、１６ビット乱数ＲＬ０の値を取り込むかの設定を示している。図２５（Ｂ）に示す例では、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット番号［２－０］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット番号［２－０］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＬ００ＬＳ０～ＲＬ００ＬＳ２は、初期値として”０００”が設定されている。

【０２００】

図２６（Ａ）は、ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）の構成例を示している。図２６（Ｂ）は、ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）に格納されるデータの各ビットにおける設定内容の一例を示している。ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）のビット番号［７］に格納されるデータＲＬ０３ＲＦは、ＲＬ０ハードラッチ乱数値レジスタ３（ＲＬ０ＨＶ３）に、外部端子入力により、１６ビット乱数ＲＬ０の値を取り込む際の条件の設定を示している。図２６（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＬ０３ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＬ０３ＲＦは、初期値として”０”が設定されている。

【０２０１】

ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）のビット番号［６－４］に格納されるデータＲＬ０３ＬＳ０～ＲＬ０３ＬＳ２は、ＲＬ０ハードラッチ乱数値レジスタ３（ＲＬ０ＨＶ３）に、どの外部端子入力により、１６ビット乱数ＲＬ０の値を取り込むかの設定を示している。図２６（Ｂ）に示す例では、ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）のビット番号［６－４］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）のビット番号［６－４］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＬ０３ＬＳ０～ＲＬ０３ＬＳ２は、初期値として”０００”が設定されている。

【０２０２】

ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）のビット番号［３］に格納されるデータＲＬ０２ＲＦは、ＲＬ０ハードラッチ乱数値レジスタ２（ＲＬ０ＨＶ２）に、外部端子入力により、１６ビット乱数ＲＬ０の値を取り込む際の条件の設定を示している。図２６（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＬ０２ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＬ０２ＲＦは、初期値として”０”が設定されている。

【０２０３】

ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）のビット番号［２－０］に格納されるデータＲＬ０２ＬＳ０～ＲＬ０２ＬＳ２は、ＲＬ０ハードラッチ乱数値レジスタ２（ＲＬ０ＨＶ２）に、どの外部端子入力により、１６ビット乱数ＲＬ０の値を取り込むかの設定を示している。図２６（Ｂ）に示す例では、ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）のビット番号［２－０］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）のビット番号［２－０］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＬ０２ＬＳ０～ＲＬ０２ＬＳ２は、初期値として”０００”が設定されている。

【０２０４】

図２７（Ａ）は、ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）の構成例を示している。図２７（Ｂ）は、ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、図２７において、ｎは０～３の値をとる。ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）のビット番号［７］に格納されるデータＲＬｎ１ＲＦは、ＲＬｎハードラッチ乱数値レジスタ１（ＲＬｎＨＶ１）に、外部端子入力により、１６ビット乱数ＲＬｎの値を取り込む際の条件の設定を示している。図２７（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＬｎ１ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＬｎ１ＲＦは、初期値として”０”が設定されている。

【０２０５】

ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）のビット番号［６－４］に格納されるデータＲＬｎ１ＬＳ０～ＲＬｎ１ＬＳ２は、ＲＬｎハードラッチ乱数値レジスタ１（ＲＬｎＨＶ１）に、どの外部端子入力により、１６ビット乱数ＲＬｎの値を取り込むかの設定を示している。図２７（Ｂ）に示す例では、ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）のビット番号［６－４］に”０００”が設定された場合にはＰＩ０端子が選択され、

”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）のビット番号［６－４］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＬｎ１ＬＳ０～ＲＬｎ１ＬＳ２は、初期値として”０００”が設定されている。

【０２０６】

ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）のビット番号［３］に格納されるデータＲＬｎ０ＲＦは、ＲＬｎハードラッチ乱数値レジスタ０（ＲＬｎＨＶ０）に、外部端子入力により、１６ビット乱数ＲＬｎの値を取り込む際の条件の設定を示している。図２７（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＬｎ０ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＬｎ０ＲＦは、初期値として”０”が設定されている。

【０２０７】

ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）のビット番号［２－０］に格納されるデータＲＬｎ０ＬＳ０～ＲＬｎ０ＬＳ２は、ＲＬｎハードラッチ乱数値レジスタ０（ＲＬｎＨＶ０）に、どの外部端子入力により、１６ビット乱数ＲＬｎの値を取り込むかの設定を示している。図２７（Ｂ）に示す例では、ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）のビット番号［２－０］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）のビット番号［２－０］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＬｎ０ＬＳ０～ＲＬｎ０ＬＳ２は、初期値として”０００”が設定されている。

【０２０８】

図２８（Ａ）は、ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）の構成例を示している。図２８（Ｂ）は、ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）に格納されるデータの各ビットにおける設定内容の一例を示している。ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）のビット番号［７］に格納されるデータＲＳ１ＲＦは、ＲＳ１ハードラッチ乱数値レジスタ（ＲＳ１ＨＶ）に、外部端子入力により、８ビット乱数ＲＳ１の値を取り込む際の条件の設定を示している。図２８（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＳ１ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＳ１ＲＦは、初期値として”０”が設定されている。

【０２０９】

ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）のビット番号［６－４］に格納されるデータＲＳ１ＬＳ０～ＲＳ１ＬＳ２は、ＲＳ１ハードラッチ乱数値レジスタ（ＲＳ１ＨＶ）に、どの外部端子入力により、８ビット乱数ＲＳ１の値を取り込むかの設定を示している。図２８（Ｂ）に示す例では、ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）のビット番号［６－４］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）のビット番号［６－４］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＳ１ＬＳ０～ＲＳ１ＬＳ２は、初期値として”０００”が設定されてい

る。
【０２１０】
ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）のビット番号［３］に格納されるデータＲＳ０ＲＦは、ＲＳ０ハードラッチ乱数値レジスタ（ＲＳ０ＨＶ）に、外部端子入力により、８ビット乱数ＲＳ０の値を取り込む際の条件の設定を示している。図２８（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＳ０ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＳ０ＲＦは、初期値として”０”が設定されている。
【０２１１】
ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）のビット番号［２－０］に格納されるデータＲＳ０ＬＳ０～ＲＳ０ＬＳ２は、ＲＳ０ハードラッチ乱数値レジスタ（ＲＳ０ＨＶ）に、どの外部端子入力により、８ビット乱数ＲＳ０の値を取り込むかの設定を示している。図２８（Ｂ）に示す例では、ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）のビット番号［２－０］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）のビット番号［２－０］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＳ０ＬＳ０～ＲＳ０ＬＳ２は、初期値として”０００”が設定されている。
【０２１２】
図２９（Ａ）は、ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）の構成例を示している。図２９（Ｂ）は、ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）に格納されるデータの各ビットにおける設定内容の一例を示している。ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）のビット番号［７］に格納されるデータＲＳ３ＲＦは、ＲＳ３ハードラッチ乱数値レジスタ（ＲＳ３ＨＶ）に、外部端子入力により、８ビット乱数ＲＳ３の値を取り込む際の条件の設定を示している。図２９（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データＲＳ３ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＳ３ＲＦは、初期値として”０”が設定されている。
【０２１３】
ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）のビット番号［６－４］に格納されるデータＲＳ３ＬＳ０～ＲＳ３ＬＳ２は、ＲＳ３ハードラッチ乱数値レジスタ（ＲＳ３ＨＶ）に、どの外部端子入力により、８ビット乱数ＲＳ３の値を取り込むかの設定を示している。図２９（Ｂ）に示す例では、ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）のビット番号［６－４］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ１）のビット番号［６－４］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＳ３ＬＳ０～ＲＳ３ＬＳ２は、初期値として”０００”が設定されている。
【０２１４】
ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）のビット番号［３］に格納されるデータＲＳ２ＲＦは、ＲＳ２ハードラッチ乱数値レジスタ（ＲＳ２ＨＶ）に、外部端子入力により、８ビット乱数ＲＳ２の値を取り込む際の条件の設定を示している。図２９（Ｂ）に示す例では、値を読み込まないと次の値をラッチしないように設定した場合には、データ

ＲＳ２ＲＦのビット値が“０”となる一方、値を読み込まなくても次の値をラッチするように設定した場合には、そのビット値が“１”となる。なお、データＲＳ２ＲＦは、初期値として”０”が設定されている。

【０２１５】

ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）のビット番号［２－０］に格納されるデータＲＳ２ＬＳ０～ＲＳ２ＬＳ２は、ＲＳ２ハードラッチ乱数値レジスタ（ＲＳ２ＨＶ）に、どの外部端子入力により、８ビット乱数ＲＳ２の値を取り込むかの設定を示している。図２９（Ｂ）に示す例では、ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）のビット番号［２－０］に”０００”が設定された場合にはＰＩ０端子が選択され、”００１”が設定された場合にはＰＩ１端子が選択され、”０１０”が設定された場合にはＰＩ２端子が選択され、”０１１”が設定された場合にはＰＩ３端子が選択され、”１００”が設定された場合にはＰＩ４端子が選択され、”１０１”が設定された場合にはＰＩ５／ＸＩＮＴ端子が選択される。なお、ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）のビット番号［２－０］に”１１０”や”１１１”が設定された場合には、その設定は無効である。また、データＲＳ２ＬＳ０～ＲＳ２ＬＳ２は、初期値として”０００”が設定されている。

【０２１６】

図３０（Ａ）は、ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）の構成例を示している。図３０（Ｂ）は、ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）のビット［７－６］のビット値は必ず”０”とされる。

【０２１７】

ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）のビット番号［５］に格納されるデータＲＬ１１ＩＥは、ＲＬ１ハードラッチ乱数値レジスタ１（ＲＬ１ＨＶ１）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３０（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ１１ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ１１ＩＥは、初期値として”０”が設定されている。

【０２１８】

ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）のビット番号［４］に格納されるデータＲＬ１０ＩＥは、ＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３０（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ１０ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ１０ＩＥは、初期値として”０”が設定されている。

【０２１９】

ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）のビット番号［３］に格納されるデータＲＬ０３ＩＥは、ＲＬ０ハードラッチ乱数値レジスタ３（ＲＬ０ＨＶ３）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３０（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ０３ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ０３ＩＥは、初期値として”０”が設定されている。

【０２２０】

ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）のビット番号［２］に格納されるデータＲＬ０２ＩＥは、ＲＬ０ハードラッチ乱数値レジスタ２（ＲＬ０ＨＶ２）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３０（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ０２ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ０２ＩＥは、初期値として”０”が設定されている。

【０２２１】

ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）のビット番号［１］に格納されるデータＲＬ０１ＩＥは、ＲＬ０ハードラッチ乱数値レジスタ１（ＲＬ０ＨＶ１）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３０（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ０１ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ０１ＩＥは、初期値として”０”が設定されている。

【０２２２】

ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）のビット番号［１］に格納されるデータＲＬ００ＩＥは、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３０（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ００ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ００ＩＥは、初期値として”０”が設定されている。

【０２２３】

図３１（Ａ）は、ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）の構成例を示している。図３１（Ｂ）は、ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）のビット［７－６］およびビット［３－２］のビット値は必ず”０”とされる。

【０２２４】

ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）のビット番号［５］に格納されるデータＲＬ３１ＩＥは、ＲＬ３ハードラッチ乱数値レジスタ１（ＲＬ３ＨＶ１）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３１（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ３１ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ３１ＩＥは、初期値として”０”が設定されている。

【０２２５】

ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）のビット番号［４］に格納されるデータＲＬ３０ＩＥは、ＲＬ３ハードラッチ乱数値レジスタ０（ＲＬ３ＨＶ０）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３１（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ３０ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ３０ＩＥは、初期値として”０”が設定されている。

【０２２６】

ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）のビット番号［１］に格納されるデータＲＬ２１ＩＥは、ＲＬ２ハードラッチ乱数値レジスタ１（ＲＬ２ＨＶ１）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３１（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ２１ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ２１ＩＥは、初期値として”０”が設定されている。

【０２２７】

ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）のビット番号［０］に格納されるデータＲＬ２０ＩＥは、ＲＬ２ハードラッチ乱数値レジスタ０（ＲＬ２ＨＶ０）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３１（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＬ２０ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＬ２０ＩＥは、初期値として”０”が設定されている。

【０２２８】

図３２（Ａ）は、ＲＳ割り込み制御レジスタ（ＲＳＩＣ）の構成例を示している。図３２（Ｂ）は、ＲＳ割り込み制御レジスタ（ＲＳＩＣ）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、ＲＳ割り込み制御レジスタ（ＲＳＩＣ）は、８

ビット乱数回路５０８ａとフリーランカウンタ回路５０７とで兼用で用いられるレジスタであり、ＲＳ割り込み制御レジスタ（ＲＳＩＣ）のビット［７－４］は、フリーランカウンタ５０７が用いるハードラッチレジスタ（ＦＲＣ０ハードラッチレジスタ（ＦＲ０ＨＶ）～ＦＲＣ３ハードラッチレジスタ（ＦＲ３ＨＶ））に関する設定を示している。

【０２２９】

ＲＳ割り込み制御レジスタ（ＲＳＩＣ）のビット番号［３］に格納されるデータＲＳ３ＩＥは、ＲＳ３ハードラッチ乱数値レジスタ（ＲＳ３ＨＶ）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３２（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＳ３ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＳ３ＩＥは、初期値として”０”が設定されている。

【０２３０】

ＲＳ割り込み制御レジスタ（ＲＳＩＣ）のビット番号［２］に格納されるデータＲＳ２ＩＥは、ＲＳ２ハードラッチ乱数値レジスタ（ＲＳ２ＨＶ）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３２（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＳ２ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＳ２ＩＥは、初期値として”０”が設定されている。

【０２３１】

ＲＳ割り込み制御レジスタ（ＲＳＩＣ）のビット番号［１］に格納されるデータＲＳ１ＩＥは、ＲＳ１ハードラッチ乱数値レジスタ（ＲＳ１ＨＶ）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３２（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＳ１ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＳ１ＩＥは、初期値として”０”が設定されている。

【０２３２】

ＲＳ割り込み制御レジスタ（ＲＳＩＣ）のビット番号［０］に格納されるデータＲＳ０ＩＥは、ＲＳ０ハードラッチ乱数値レジスタ（ＲＳ０ＨＶ）に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可の設定を示している。図３２（Ｂ）に示す例では、割り込み禁止に設定した場合には、データＲＳ０ＩＥのビット値が“０”となる一方、割り込み許可に設定した場合には、そのビット値が“１”となる。なお、データＲＳ０ＩＥは、初期値として”０”が設定されている。

【０２３３】

図３３（Ａ）は、ＲＬｎ最大値設定レジスタ（ＲＬｎＭＸ）の構成例を示している。図３３（Ｂ）は、ＲＬｎ最大値設定レジスタ（ＲＬｎＭＸ）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、図３３において、ｎは０～３の値をとる。図３３（Ｂ）に示すように、ＲＬｎ最大値設定レジスタ（ＲＬｎＭＸ）のビット番号［１５－０］に格納されるデータＲＬｎＭＸ１５～ＲＬｎＭＸ０は、１６ビット乱数ＲＬｎの最大値が設定される。

【０２３４】

図３４（Ａ）は、ＲＳｎ最大値設定レジスタ（ＲＳｎＭＸ）の構成例を示している。図３４（Ｂ）は、ＲＳｎ最大値設定レジスタ（ＲＳｎＭＸ）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、図３４において、ｎは０～３の値をとる。図３４（Ｂ）に示すように、ＲＳｎ最大値設定レジスタ（ＲＳｎＭＸ）のビット番号［７－０］に格納されるデータＲＳｎＭＸ７～ＲＳｎＭＸ０は、８ビット乱数ＲＳｎの最大値が設定される。

【０２３５】

図３５（Ａ）は、乱数列変更レジスタ（ＲＤＳＣ）の構成例を示している。図３５（Ｂ）は、乱数列変更レジスタ（ＲＤＳＣ）に格納されるデータの各ビットにおける設定内容の一例を示している。乱数列変更レジスタ（ＲＤＳＣ）のビット番号［７］に格納される

データＲＳ３ＳＣは、８ビット乱数ＲＳ３の乱数列変更要求ビットを示している。図３５（Ｂ）に示す例では、乱数列を変更しないに設定した場合には、データＲＳ３ＳＣのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＳ３ＳＣは、初期値として”０”が設定されている。

【０２３６】

乱数列変更レジスタ（ＲＤＳＣ）のビット番号［６］に格納されるデータＲＳ２ＳＣは、８ビット乱数ＲＳ２の乱数列変更要求ビットを示している。図３５（Ｂ）に示す例では、乱数列を変更しないに設定した場合には、データＲＳ２ＳＣのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＳ２ＳＣは、初期値として”０”が設定されている。

【０２３７】

乱数列変更レジスタ（ＲＤＳＣ）のビット番号［５］に格納されるデータＲＳ１ＳＣは、８ビット乱数ＲＳ１の乱数列変更要求ビットを示している。図３５（Ｂ）に示す例では、乱数列を変更しないに設定した場合には、データＲＳ１ＳＣのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＳ１ＳＣは、初期値として”０”が設定されている。

【０２３８】

乱数列変更レジスタ（ＲＤＳＣ）のビット番号［４］に格納されるデータＲＳ０ＳＣは、８ビット乱数ＲＳ０の乱数列変更要求ビットを示している。図３５（Ｂ）に示す例では、乱数列を変更しないに設定した場合には、データＲＳ０ＳＣのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＳ０ＳＣは、初期値として”０”が設定されている。

【０２３９】

乱数列変更レジスタ（ＲＤＳＣ）のビット番号［３］に格納されるデータＲＬ３ＳＣは、１６ビット乱数ＲＬ３の乱数列変更要求ビットを示している。図３５（Ｂ）に示す例では、乱数列を変更しないに設定した場合には、データＲＬ３ＳＣのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＬ３ＳＣは、初期値として”０”が設定されている。

【０２４０】

乱数列変更レジスタ（ＲＤＳＣ）のビット番号［２］に格納されるデータＲＬ２ＳＣは、１６ビット乱数ＲＬ２の乱数列変更要求ビットを示している。図３５（Ｂ）に示す例では、乱数列を変更しないに設定した場合には、データＲＬ２ＳＣのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＬ２ＳＣは、初期値として”０”が設定されている。

【０２４１】

乱数列変更レジスタ（ＲＤＳＣ）のビット番号［１］に格納されるデータＲＬ１ＳＣは、１６ビット乱数ＲＬ１の乱数列変更要求ビットを示している。図３５（Ｂ）に示す例では、乱数列を変更しないに設定した場合には、データＲＬ１ＳＣのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＬ１ＳＣは、初期値として”０”が設定されている。

【０２４２】

乱数列変更レジスタ（ＲＤＳＣ）のビット番号［０］に格納されるデータＲＬ０ＳＣは、１６ビット乱数ＲＬ０の乱数列変更要求ビットを示している。図３５（Ｂ）に示す例では、乱数列を変更しないに設定した場合には、データＲＬ０ＳＣのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＬ０ＳＣは、初期値として”０”が設定されている。

【０２４３】

図３６（Ａ）は、乱数ソフトラッチレジスタ（ＲＤＳＬ）の構成例を示している。図３６（Ｂ）は、乱数ソフトラッチレジスタ（ＲＤＳＬ）に格納されるデータの各ビットにおける設定内容の一例を示している。乱数ソフトラッチレジスタ（ＲＤＳＬ）のビット番号

［７］に格納されるデータＲＳ３ＳＬは、８ビット乱数ＲＳ３の乱数値を、ＲＳ３ソフトラッチ乱数値レジスタ（ＲＳ３ＳＶ）に取り込むためのビットを示している。図３６（Ｂ）に示す例では、乱数値を取り込まないに設定した場合には、データＲＳ３ＳＬのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＳ３ＳＬは、初期値として”０”が設定されている。

【０２４４】

乱数ソフトラッチレジスタ（ＲＤＳＬ）のビット番号［６］に格納されるデータＲＳ２ＳＬは、８ビット乱数ＲＳ２の乱数値を、ＲＳ２ソフトラッチ乱数値レジスタ（ＲＳ２ＳＶ）に取り込むためのビットを示している。図３６（Ｂ）に示す例では、乱数値を取り込まないに設定した場合には、データＲＳ２ＳＬのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＳ２ＳＬは、初期値として”０”が設定されている。

【０２４５】

乱数ソフトラッチレジスタ（ＲＤＳＬ）のビット番号［５］に格納されるデータＲＳ１ＳＬは、８ビット乱数ＲＳ１の乱数値を、ＲＳ１ソフトラッチ乱数値レジスタ（ＲＳ１ＳＶ）に取り込むためのビットを示している。図３６（Ｂ）に示す例では、乱数値を取り込まないに設定した場合には、データＲＳ１ＳＬのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＳ１ＳＬは、初期値として”０”が設定されている。

【０２４６】

乱数ソフトラッチレジスタ（ＲＤＳＬ）のビット番号［４］に格納されるデータＲＳ０ＳＬは、８ビット乱数ＲＳ０の乱数値を、ＲＳ０ソフトラッチ乱数値レジスタ（ＲＳ０ＳＶ）に取り込むためのビットを示している。図３６（Ｂ）に示す例では、乱数値を取り込まないに設定した場合には、データＲＳ０ＳＬのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＳ０ＳＬは、初期値として”０”が設定されている。

【０２４７】

乱数ソフトラッチレジスタ（ＲＤＳＬ）のビット番号［３］に格納されるデータＲＬ３ＳＬは、１６ビット乱数ＲＬ３の乱数値を、ＲＬ３ソフトラッチ乱数値レジスタ（ＲＬ３ＳＶ）に取り込むためのビットを示している。図３６（Ｂ）に示す例では、乱数値を取り込まないに設定した場合には、データＲＬ３ＳＬのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＬ３ＳＬは、初期値として”０”が設定されている。

【０２４８】

乱数ソフトラッチレジスタ（ＲＤＳＬ）のビット番号［２］に格納されるデータＲＬ２ＳＬは、１６ビット乱数ＲＬ２の乱数値を、ＲＬ２ソフトラッチ乱数値レジスタ（ＲＬ２ＳＶ）に取り込むためのビットを示している。図３６（Ｂ）に示す例では、乱数値を取り込まないに設定した場合には、データＲＬ２ＳＬのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＬ２ＳＬは、初期値として”０”が設定されている。

【０２４９】

乱数ソフトラッチレジスタ（ＲＤＳＬ）のビット番号［１］に格納されるデータＲＬ１ＳＬは、１６ビット乱数ＲＬ１の乱数値を、ＲＬ１ソフトラッチ乱数値レジスタ（ＲＬ１ＳＶ）に取り込むためのビットを示している。図３６（Ｂ）に示す例では、乱数値を取り込まないに設定した場合には、データＲＬ１ＳＬのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＬ１ＳＬは、初期値として”０”が設定されている。

【０２５０】

乱数ソフトラッチレジスタ（ＲＤＳＬ）のビット番号［０］に格納されるデータＲＬ０ＳＬは、１６ビット乱数ＲＬ０の乱数値を、ＲＬ０ソフトラッチ乱数値レジスタ（ＲＬ０

ＳＶ）に取り込むためのビットを示している。図３６（Ｂ）に示す例では、乱数値を取り込まないに設定した場合には、データＲＬ０ＳＬのビット値が“０”となる一方、乱数列を変更するに設定した場合には、そのビット値が“１”となる。なお、データＲＬ０ＳＬは、初期値として”０”が設定されている。

【０２５１】

図３７（Ａ）は、乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）の構成例を示している。図３７（Ｂ）は、乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）に格納されるデータの各ビットにおける設定内容の一例を示している。乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）のビット番号［７］に格納されるデータＲＳ３ＳＦは、ＲＳ３ソフトラッチ乱数値レジスタ（ＲＳ３ＳＶ）に、乱数値が取り込まれたことを示している。図３７（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＳ３ＳＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＳ３ＳＦは、初期値として”０”が設定されている。

【０２５２】

乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）のビット番号［６］に格納されるデータＲＳ２ＳＦは、ＲＳ２ソフトラッチ乱数値レジスタ（ＲＳ２ＳＶ）に、乱数値が取り込まれたことを示している。図３７（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＳ２ＳＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＳ２ＳＦは、初期値として”０”が設定されている。

【０２５３】

乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）のビット番号［５］に格納されるデータＲＳ１ＳＦは、ＲＳ１ソフトラッチ乱数値レジスタ（ＲＳ１ＳＶ）に、乱数値が取り込まれたことを示している。図３７（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＳ１ＳＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＳ１ＳＦは、初期値として”０”が設定されている。

【０２５４】

乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）のビット番号［４］に格納されるデータＲＳ０ＳＦは、ＲＳ０ソフトラッチ乱数値レジスタ（ＲＳ０ＳＶ）に、乱数値が取り込まれたことを示している。図３７（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＳ０ＳＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＳ０ＳＦは、初期値として”０”が設定されている。

【０２５５】

乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）のビット番号［３］に格納されるデータＲＬ３ＳＦは、ＲＬ３ソフトラッチ乱数値レジスタ（ＲＬ３ＳＶ）に、乱数値が取り込まれたことを示している。図３７（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ３ＳＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ３ＳＦは、初期値として”０”が設定されている。

【０２５６】

乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）のビット番号［２］に格納されるデータＲＬ２ＳＦは、ＲＬ２ソフトラッチ乱数値レジスタ（ＲＬ２ＳＶ）に、乱数値が取り込まれたことを示している。図３７（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ２ＳＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ２ＳＦは、初期値として”０”が設定されている。

【０２５７】

乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）のビット番号［１］に格納されるデータ

ＲＬ１ＳＦは、ＲＬ１ソフトラッチ乱数値レジスタ（ＲＬ１ＳＶ）に、乱数値が取り込まれたことを示している。図３７（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ１ＳＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ１ＳＦは、初期値として”０”が設定されている。

【０２５８】

乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）のビット番号［０］に格納されるデータＲＬ０ＳＦは、ＲＬ０ソフトラッチ乱数値レジスタ（ＲＬ０ＳＶ）に、乱数値が取り込まれたことを示している。図３７（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ０ＳＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ０ＳＦは、初期値として”０”が設定されている。

【０２５９】

図３８（Ａ）は、ＲＬｎソフトラッチ乱数値レジスタ（ＲＬｎＳＶ）の構成例を示している。図３８（Ｂ）は、ＲＬｎソフトラッチ乱数値レジスタ（ＲＬｎＳＶ）に格納されるデータの各ビットにおける格納内容の一例を示している。なお、図３８において、ｎは０～３の値をとる。図３８（Ｂ）に示すように、ＲＬｎソフトラッチ乱数値レジスタ（ＲＬｎＳＶ）のビット番号［１５－０］に格納されるデータＲＬｎＳＶ１５～ＲＬｎＳＶ０は、乱数ソフトラッチレジスタ（ＲＤＳＬ）により取り込まれた１６ビット乱数ＲＬｎの値が格納される。なお、乱数値が取り込まれると、乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）の該当するビットに”１”がセットされる。

【０２６０】

図３９（Ａ）は、ＲＳｎソフトラッチ乱数値レジスタ（ＲＳｎＳＶ）の構成例を示している。図３９（Ｂ）は、ＲＳｎソフトラッチ乱数値レジスタ（ＲＳｎＳＶ）に格納されるデータの各ビットにおける格納内容の一例を示している。なお、図３９において、ｎは０～３の値をとる。図３９（Ｂ）に示すように、ＲＳｎソフトラッチ乱数値レジスタ（ＲＳｎＳＶ）のビット番号［７－０］に格納されるデータＲＳｎＳＶ７～ＲＳｎＳＶ０は、乱数ソフトラッチレジスタ（ＲＤＳＬ）により取り込まれた８ビット乱数ＲＳｎの値が格納される。なお、乱数値が取り込まれると、乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）の該当するビットに”１”がセットされる。

【０２６１】

図４０（Ａ）は、ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）の構成例を示している。図４０（Ｂ）は、ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）のビット［７－６］のビット値は必ず”０”とされる。

【０２６２】

ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）のビット番号［５］に格納されるデータＲＬ１１ＨＦは、ＲＬ１ハードラッチ乱数値レジスタ１に、乱数値が取り込まれたことを示している。図４０（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ１１ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ１１ＨＦは、初期値として”０”が設定されている。

【０２６３】

ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）のビット番号［４］に格納されるデータＲＬ１０ＨＦは、ＲＬ１ハードラッチ乱数値レジスタ０に、乱数値が取り込まれたことを示している。図４０（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ１０ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ１０ＨＦは、初期値として”０”が設定されている。

【０２６４】

ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）のビット番号［３］に格納されるデータＲＬ０３ＨＦは、ＲＬ０ハードラッチ乱数値レジスタ３に、乱数値が取り込まれたことを示している。図４０（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ０３ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ０３ＨＦは、初期値として”０”が設定されている。

【０２６５】

ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）のビット番号［２］に格納されるデータＲＬ０２ＨＦは、ＲＬ０ハードラッチ乱数値レジスタ２に、乱数値が取り込まれたことを示している。図４０（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ０２ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ０２ＨＦは、初期値として”０”が設定されている。

【０２６６】

ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）のビット番号［１］に格納されるデータＲＬ０１ＨＦは、ＲＬ０ハードラッチ乱数値レジスタ１に、乱数値が取り込まれたことを示している。図４０（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ０１ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ０１ＨＦは、初期値として”０”が設定されている。

【０２６７】

ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）のビット番号［０］に格納されるデータＲＬ００ＨＦは、ＲＬ０ハードラッチ乱数値レジスタ０に、乱数値が取り込まれたことを示している。図４０（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ００ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ００ＨＦは、初期値として”０”が設定されている。

【０２６８】

図４１（Ａ）は、ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）の構成例を示している。図４１（Ｂ）は、ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）のビット［７－６］およびビット［３－２］のビット値は必ず”０”とされる。

【０２６９】

ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）のビット番号［５］に格納されるデータＲＬ３１ＨＦは、ＲＬ３ハードラッチ乱数値レジスタ１に、乱数値が取り込まれたことを示している。図４１（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ３１ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ３１ＨＦは、初期値として”０”が設定されている。

【０２７０】

ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）のビット番号［４］に格納されるデータＲＬ３０ＨＦは、ＲＬ３ハードラッチ乱数値レジスタ０に、乱数値が取り込まれたことを示している。図４１（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ３０ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ３０ＨＦは、初期値として”０”が設定されている。

【０２７１】

ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）のビット番号［１］に格納されるデータＲＬ２１ＨＦは、ＲＬ２ハードラッチ乱数値レジスタ１に、乱数値が取り込まれたこ

とを示している。図４１（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ２１ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ２１ＨＦは、初期値として”０”が設定されている。

【０２７２】

ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）のビット番号［１］に格納されるデータＲＬ２０ＨＦは、ＲＬ２ハードラッチ乱数値レジスタ０に、乱数値が取り込まれたことを示している。図４１（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＬ２０ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＬ２０ＨＦは、初期値として”０”が設定されている。

【０２７３】

図４２（Ａ）は、ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）の構成例を示している。図４２（Ｂ）は、ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）に格納されるデータの各ビットにおける設定内容の一例を示している。なお、ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）は、８ビット乱数回路５０８ａとフリーランカウンタ回路５０７とで兼用で用いられるレジスタであり、ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）のビット［７－４］は、フリーランカウンタ５０７が用いるハードラッチレジスタ（ＦＲＣ０ハードラッチレジスタ（ＦＲ０ＨＶ）～ＦＲＣ３ハードラッチレジスタ（ＦＲ３ＨＶ））に関する設定を示している。

【０２７４】

ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）のビット番号［３］に格納されるデータＲＳ３ＨＦは、ＲＳ３ハードラッチ乱数値レジスタ（ＲＳ３ＨＶ）に、乱数値が取り込まれたことを示している。図４２（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＳ３ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＳ３ＨＦは、初期値として”０”が設定されている。

【０２７５】

ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）のビット番号［２］に格納されるデータＲＳ２ＨＦは、ＲＳ２ハードラッチ乱数値レジスタ（ＲＳ２ＨＶ）に、乱数値が取り込まれたことを示している。図４２（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＳ２ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＳ２ＨＦは、初期値として”０”が設定されている。

【０２７６】

ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）のビット番号［１］に格納されるデータＲＳ１ＨＦは、ＲＳ１ハードラッチ乱数値レジスタ（ＲＳ１ＨＶ）に、乱数値が取り込まれたことを示している。図４２（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＳ１ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＳ１ＨＦは、初期値として”０”が設定されている。

【０２７７】

ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）のビット番号［０］に格納されるデータＲＳ０ＨＦは、ＲＳ０ハードラッチ乱数値レジスタ（ＲＳ０ＨＶ）に、乱数値が取り込まれたことを示している。図４２（Ｂ）に示す例では、乱数値が取り込まれていない場合には、データＲＳ０ＨＦのビット値が“０”となる一方、乱数値を取り込み済みである場合には、そのビット値が“１”となる。なお、データＲＳ０ＨＦは、初期値として”０”が設定されている。

【０２７８】

図４３（Ａ）は、ＲＬ０ハードラッチ乱数値レジスタｍ（ＲＬ０ｍＨＶ）の構成例を示

している。図４３（Ｂ）は、ＲＬ０ハードラッチ乱数値レジスタｍ（ＲＬ０ｍＨＶ）に格納されるデータの各ビットにおける格納内容の一例を示している。なお、図４３において、ｍは０～３の値をとる。図４３（Ｂ）に示すように、ＲＬ０ハードラッチ乱数値レジスタｍ（ＲＬ０ｍＨＶ）のビット番号［１５－０］に格納されるデータＲＬ０ｍＨＶ１５～ＲＬ０ｍＨＶ０は、外部端子入力により取り込まれた１６ビット乱数ＲＬ０の値が格納される。なお、乱数値が取り込まれると、ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）の該当するビットに”１”がセットされる。

【０２７９】

図４４（Ａ）は、ＲＬ１ハードラッチ乱数値レジスタｍ（ＲＬ１ｍＨＶ）の構成例を示している。図４４（Ｂ）は、ＲＬ１ハードラッチ乱数値レジスタｍ（ＲＬ１ｍＨＶ）に格納されるデータの各ビットにおける格納内容の一例を示している。なお、図４４において、ｍは０～３の値をとる。図４４（Ｂ）に示すように、ＲＬ１ハードラッチ乱数値レジスタｍ（ＲＬ１ｍＨＶ）のビット番号［１５－０］に格納されるデータＲＬ１ｍＨＶ１５～ＲＬ１ｍＨＶ０は、外部端子入力により取り込まれた１６ビット乱数ＲＬ１の値が格納される。なお、乱数値が取り込まれると、ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）の該当するビットに”１”がセットされる。

【０２８０】

図４５（Ａ）は、ＲＬ２ハードラッチ乱数値レジスタｍ（ＲＬ２ｍＨＶ）の構成例を示している。図４５（Ｂ）は、ＲＬ２ハードラッチ乱数値レジスタｍ（ＲＬ２ｍＨＶ）に格納されるデータの各ビットにおける格納内容の一例を示している。なお、図４５において、ｍは０～３の値をとる。図４５（Ｂ）に示すように、ＲＬ２ハードラッチ乱数値レジスタｍ（ＲＬ２ｍＨＶ）のビット番号［１５－０］に格納されるデータＲＬ２ｍＨＶ１５～ＲＬ２ｍＨＶ０は、外部端子入力により取り込まれた１６ビット乱数ＲＬ２の値が格納される。なお、乱数値が取り込まれると、ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）の該当するビットに”１”がセットされる。

【０２８１】

図４６（Ａ）は、ＲＬ３ハードラッチ乱数値レジスタｍ（ＲＬ３ｍＨＶ）の構成例を示している。図４６（Ｂ）は、ＲＬ３ハードラッチ乱数値レジスタｍ（ＲＬ３ｍＨＶ）に格納されるデータの各ビットにおける格納内容の一例を示している。なお、図４６において、ｍは０～３の値をとる。図４６（Ｂ）に示すように、ＲＬ３ハードラッチ乱数値レジスタｍ（ＲＬ３ｍＨＶ）のビット番号［１５－０］に格納されるデータＲＬ３ｍＨＶ１５～ＲＬ３ｍＨＶ０は、外部端子入力により取り込まれた１６ビット乱数ＲＬ３の値が格納される。なお、乱数値が取り込まれると、ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）の該当するビットに”１”がセットされる。

【０２８２】

図４７（Ａ）は、ＲＳｎハードラッチ乱数値レジスタ（ＲＳｎＨＶ）の構成例を示している。図４７（Ｂ）は、ＲＳｎハードラッチ乱数値レジスタ（ＲＳｎＨＶ）に格納されるデータの各ビットにおける格納内容の一例を示している。なお、図４７において、ｎは０～３の値をとる。図４７（Ｂ）に示すように、ＲＳｎハードラッチ乱数値レジスタ（ＲＬｎＨＶ）のビット番号［７０］に格納されるデータＲＳｎＨＶ７～ＲＳｎＨＶ０は、外部端子入力により取り込まれた８ビット乱数ＲＳｎの値が格納される。なお、乱数値が取り込まれると、ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）の該当するビットに”１”がセットされる。

【０２８３】

図６に示す遊技制御用マイクロコンピュータ５６０が備えるタイマ回路５０９は、８ビットプログラマブルタイマであり、遊技制御用マイクロコンピュータ５６０は、タイマ回路５０９として、８ビットのカウンタを３チャネル備える。この実施の形態では、タイマ回路５０９を用いてユーザプログラムによる設定により、リアルタイム割り込み要求や時間計測を行うことが可能である。

【０２８４】

図６に示す遊技制御用マイクロコンピュータ５６０が備える割り込みコントローラ５１０は、ＰＩ５／ＸＩＮＴ端子からの外部割り込み要求や、内蔵の周辺回路（例えば、シリアル通信回路５１２、乱数回路５０８ａ，５０８ｂ、タイマ回路５０９）からの割り込み要求を制御する回路である。

【０２８５】

図６に示す遊技制御用マイクロコンピュータ５６０が備えるパラレル入力ポート５１１は、８ビット幅の入力専用ポート（ＰＩＰ）を内蔵する。また、図６に示す遊技制御用マイクロコンピュータ５６０が備えるパラレル出力ポート５１３は、１１ビット幅の出力専用ポート（ＰＯＰ）を内蔵する。

【０２８６】

図６に示す遊技制御用マイクロコンピュータ５６０が備えるシリアル通信回路５１２は、外部に対する入出力において非同期シリアル通信を行う回路である。なお、遊技制御用マイクロコンピュータ５６０は、シリアル通信回路５１２として、送受信両用の１チャネルの回路と、送信用のみの３チャネルの回路とを備える。なお、例えば、送受信両用の回路については、例えば、双方向の通信が必要となる遊技制御用マイクロコンピュータ５６０と払出制御基板３７が搭載する払出制御用マイクロコンピュータとの間の通信に用いるようにし、送信用のみの回路については、例えば、一方向の通信でよい遊技制御用マイクロコンピュータ５６０から演出制御用マイクロコンピュータ１００に対する通信に用いるようにする。

【０２８７】

図６に示す遊技制御用マイクロコンピュータ５６０が備えるアドレスデコード回路５１４は、遊技制御用マイクロコンピュータ５６０の内部における各機能ブロックのデコードや、外部装置用のデコード信号であるチップセレクト信号のデコードを行うための回路である。チップセレクト信号により、遊技制御用マイクロコンピュータ５６０の内部回路、あるいは、周辺デバイスとなる外部装置を、選択的に有効動作させて、ＣＰＵ５６からのアクセスが可能となる。

【０２８８】

次に、遊技機の動作について説明する。まず、この実施の形態では、既に説明したように、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号やＩＡＴ回路５０６ａからのＩＡＴ信号が発生したときに、ユーザリセットを発生させるかシステムリセットを発生させるかを可能である（図１４参照）。図４８は、リセット設定（ＫＲＥＳ）での設定内容によるリセット動作の違いを説明するための説明図である。

【０２８９】

まず、図４８（Ａ）を用いてウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号やＩＡＴ回路５０６ａからのＩＡＴ信号が発生したときにシステムリセットを発生させるように設定した場合について説明する。この場合、図４８（Ａ）に示すように、遊技機に対して電源が投入され電力供給が開始されると、遊技制御用マイクロコンピュータ５６は、ＣＰＵコアを含む全ての内部回路を初期化するとともに、プログラム管理エリアの設定内容に従って、内部リセット動作の設定や乱数回路５０８ａ，５０８ｂの設定など遊技制御用マイクロコンピュータ５６０の各種設定をハードウェア的に行う（ステップＳ１００１）。具体的には、プログラム管理エリアの図１４に示すリセット設定（ＫＲＥＳ）の設定内容に従って内部リセットの動作の設定を行ったり、プログラム管理エリアの図１５～図１９に示す１６ビット乱数初期設定１（ＫＲＬ１）～８ビット乱数初期設定２（ＫＲＳ２）の設定内容に従って乱数回路５０８ａ，５０８ｂの設定を行ったりする。なお、図４８（Ａ）に示す例では、遊技制御用マイクロコンピュータ５６は、プログラム管理エリアの設定内容に従って、内部リセット動作の設定としてシステムリセットを設定する。また、プログラム管理エリアの設定内容は予め遊技機の製作時に遊技機製造メーカ（ユーザ）によって設定されているものとする。

【０２９０】

遊技制御用マイクロコンピュータ５６０の各種設定を完了すると、遊技制御用マイクロ

コンピュータ５６は、セキュリティモードに移行し、セキュリティチェックを実行する（ステップＳ１００２）。ステップＳ１００２で実行するセキュリティチェックでは、ユーザプログラムの認証を行う。具体的には、ユーザプログラムをもとに計算された認証コードが正しいか否か再計算を行う。そして、認証コードが正しければ、ステップＳ１００３に移行し、認証コードが正しくなければ、ＣＰＵ５６を停止する。なお、セキュリティモードに移行されるセキュリティモード時間は、既に説明したように、プログラム管理エリアの図２０に示すセキュリティ時間設定（ＫＳＥＳ）の設定内容に従って可変とされている。具体的には、プログラム管理エリアの図２０に示すセキュリティ時間設定（ＫＳＥＳ）の設定内容に従ってステップＳ１００１の設定が行われることによりセキュリティモード時間が設定される。なお、認証コードは、予め遊技機の製作時の内蔵ＲＯＭ５４への書き込み時に遊技機製造メーカ（ユーザ）によってユーザプログラムとともに書き込まれているものとする。

【０２９１】

そして、セキュリティチェックを終了すると、遊技制御用マイクロコンピュータ５６０は、ユーザモードに移行し、ユーザプログラムの実行を開始する。具体的には、後述する図５０のメイン処理の実行を開始する。

【０２９２】

次いで、ユーザプログラムが実行されているときに（具体的には、後述する図５０のメイン処理内のループ処理や図５１のタイマ割込処理の実行中に）、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号やＩＡＴ回路５０６ａからのＩＡＴ信号が発生したものとする。図４８（Ａ）に示す例では、ステップＳ１００１で内部リセット動作の設定としてシステムリセットが設定されていることから、タイムアウト信号やＩＡＴ信号の発生にもとづいてシステムリセットが発生する。

【０２９３】

そして、ステップＳ１００１と同様に、遊技制御用マイクロコンピュータ５６は、ＣＰＵコアを含む全ての内部回路を初期化するとともに、プログラム管理エリアの設定内容に従って、内部リセット動作の設定や乱数回路５０８ａ，５０８ｂの設定など遊技制御用マイクロコンピュータ５６０の各種設定をハードウェア的に行う（ステップＳ１００５）。また、遊技制御用マイクロコンピュータ５６０の各種設定を完了すると、ステップＳ１００２と同様に、遊技制御用マイクロコンピュータ５６は、セキュリティモードに移行し、セキュリティチェックを実行する（ステップＳ１００６）。

【０２９４】

そして、セキュリティチェックを終了すると、ステップＳ１００３と同様に、遊技制御用マイクロコンピュータ５６０は、ユーザモードに移行し、ユーザプログラムの実行を開始する。具体的には、後述する図５０のメイン処理の実行を再び開始する。

【０２９５】

以降、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号やＩＡＴ回路５０６ａからのＩＡＴ信号が発生するごとに、ステップＳ１００４～Ｓ１００７の動作が実行される。なお、図４８（Ａ）において、ステップＳ１００１，Ｓ１００２の具体的な処理内容とステップＳ１００５，Ｓ１００６の具体的な処理内容とは同じである。

【０２９６】

次に、図４８（Ｂ）を用いてウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号やＩＡＴ回路５０６ａからのＩＡＴ信号が発生したときにユーザリセットを発生させるように設定した場合について説明する。この場合、図４８（Ｂ）に示すように、遊技機に対して電源が投入され電力供給が開始されると、遊技制御用マイクロコンピュータ５６は、ＣＰＵコアを含む全ての内部回路を初期化するとともに、プログラム管理エリアの設定内容に従って、内部リセット動作の設定や乱数回路５０８ａ，５０８ｂの設定など遊技制御用マイクロコンピュータ５６０の各種設定をハードウェア的に行う（ステップＳ１０１１）。具体的には、プログラム管理エリアの図１４に示すリセット設定（ＫＲＥＳ）の設定内容に従って内部リセットの動作の設定を行ったり、プログラム管理エリアの図

１５～図１９に示す１６ビット乱数初期設定１（ＫＲＬ１）～８ビット乱数初期設定２（ＫＲＳ２）の設定内容に従って乱数回路５０８ａ，５０８ｂの設定を行ったりする。なお、図４８（Ｂ）に示す例では、遊技制御用マイクロコンピュータ５６は、プログラム管理エリアの設定内容に従って、内部リセット動作の設定としてユーザリセットを設定する。また、プログラム管理エリアの設定内容は予め遊技機の製作時に遊技機製造メーカ（ユーザ）によって設定されているものとする。

【０２９７】

遊技制御用マイクロコンピュータ５６０の各種設定を完了すると、遊技制御用マイクロコンピュータ５６は、セキュリティモードに移行し、セキュリティチェックを実行する（ステップＳ１０１２）。ステップＳ１０１２で実行するセキュリティチェックでは、ユーザプログラムの認証を行う。具体的には、ユーザプログラムをもとに計算された認証コードが正しいか否か再計算を行う。そして、認証コードが正しければ、ステップＳ１０１３に移行し、認証コードが正しくなければ、ＣＰＵ５６を停止する。なお、セキュリティモードに移行されるセキュリティモード時間は、既に説明したように、プログラム管理エリアの図２０に示すセキュリティ時間設定（ＫＳＥＳ）の設定内容に従って可変とされている。具体的には、プログラム管理エリアの図２０に示すセキュリティ時間設定（ＫＳＥＳ）の設定内容に従ってステップＳ１０１１の設定が行われることによりセキュリティモード時間が設定される。なお、認証コードは、予め遊技機の製作時の内蔵ＲＯＭ５４への書き込み時に遊技機製造メーカ（ユーザ）によってユーザプログラムとともに書き込まれているものとする。

【０２９８】

そして、セキュリティチェックを終了すると、遊技制御用マイクロコンピュータ５６０は、ユーザモードに移行し、ユーザプログラムの実行を開始する。具体的には、後述する図５０のメイン処理の実行を開始する。

【０２９９】

次いで、ユーザプログラムが実行されているときに（具体的には、後述する図５０のメイン処理内のループ処理や図５１のタイマ割込処理の実行中に）、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号やＩＡＴ回路５０６ａからのＩＡＴ信号が発生したものとする。図４８（Ｂ）に示す例では、ステップＳ１０１１で内部リセット動作の設定としてユーザリセットが設定されていることから、タイムアウト信号やＩＡＴ信号の発生にもとづいてユーザリセットが発生する。

【０３００】

ユーザリセットが発生した場合には、ステップＳ１０１１の遊技制御用マイクロコンピュータ５６０の各種設定やステップＳ１０１２のセキュリティチェックは実行されず、遊技制御用マイクロコンピュータ５６０の内部回路のうち、ＣＰＵコア、タイマ回路５０９、フリーランカウンタ回路５０７、演算回路５０５、パラレル入力ポート５１１、パラレル出力ポート５１３、シリアル通信回路５１２、および割り込みコントローラ５１０などを初期化する。そして、そのままユーザプログラムの先頭のアドレスに戻り、ユーザプログラムの実行が先頭のアドレスから再び開始される（ステップＳ１０１５）。具体的には、後述する図５０のメイン処理の実行を再び開始する。

【０３０１】

以降、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号やＩＡＴ回路５０６ａからのＩＡＴ信号が発生するごとに、ステップＳ１０１４～Ｓ１０１５の動作が実行される。

【０３０２】

また、この実施の形態では、遊技制御用マイクロコンピュータ５６０は、ユーザプログラムの実行中に内蔵ＲＡＭ領域に格納されているデータを読み出す場合、そのデータが格納されている内蔵ＲＡＭ領域の上位および下位全てのアドレスを指定するのではなく、アドレスの下位のみを指定してデータを読み出すことが可能である。図４９は、内蔵ＲＡＭ領域に格納されているデータの読み出し方の例を示す説明図である。この実施の形態では

、ユーザプログラムで参照されるデータは、内蔵ＲＡＭ領域のうちのＦ０００Ｈ～Ｆ０ＦＦＨ領域に格納されるものとし、データ格納領域のアドレスの上位が必ずＦ０Ｈとなるものとする。また、遊技制御用マイクロコンピュータ５６０は、データ格納領域の上位アドレスを固定値として格納しておくための専用のレジスタ（Ｑレジスタ）を備え、Ｑレジスタには固定値Ｆ０Ｈが設定されるものとする。

【０３０３】

図４９に示す例では、内蔵ＲＡＭ領域のアドレスＦ０２０Ｈに格納されているデータを読み出す場合が示されている。この場合、Ｑレジスタを用いてデータを読み出すためのコマンドＬＤＱを用いて、下位アドレス２０Ｈのみを指定して、データの読み出し動作を行う（具体的には、ＬＤＱ　Ａ，（２０Ｈ）を実行する）。すると、ＣＰＵ５６は、データ格納領域の上位アドレスをＱレジスタに設定されている固定値からＦ０Ｈと特定するとともに、ＬＤＱ命令で指定された下位アドレス２０Ｈを特定し、上位および下位を合わせたデータ格納領域のアドレスがＦ０２０Ｈであると特定する。そして、ＣＰＵ５６は、特定したＦ０２０Ｈに対応するデータ格納領域に格納されているデータａを読み出し、レジスタＡに格納する。

【０３０４】

なお、Ｑレジスタの値は、システムリセット時にハードウェア的に初期化されて初期値Ｆ０Ｈに自動設定されるものとする。例えば、遊技機に対して電源が投入され電力供給が開始されたときに、Ｑレジスタの下位４ビットは０に初期化されるとともに、上位４ビットは反転回路で反転されて全て値１となることによって、Ｑレジスタの初期値としてＦ０Ｈが自動設定される。なお、後述するように、この実施の形態では、ユーザプログラムの実行が開始されたときにも、ユーザプログラムによりＱレジスタに初期値Ｆ０Ｈを設定する処理が実行される（後述するステップＳ５Ａ参照）。

【０３０５】

なお、Ｑレジスタの初期値設定は、遊技機に対して電源が投入され電力供給が開始されたときに行うハードウェア的な自動設定のみでもよいし、ユーザプログラムの開始時に実行されるユーザプログラムによる設定のみでもよい。

【０３０６】

次に、システムチェックを実行した後、ユーザモードに移行した後にユーザプログラムに従って実行される処理を説明する。ユーザモードに移行すると、遊技制御用マイクロコンピュータ５６０は、メイン処理の実行を開始する。

【０３０７】

図５０は、主基板３１における遊技制御用マイクロコンピュータ５６０が実行するメイン処理を示すフローチャートである。メイン処理において、ＣＰＵ５６は、まず、必要な初期設定を行う。初期設定処理において、ＣＰＵ５６は、まず、割込禁止に設定する（ステップＳ１）。次に、割込モードの設定を行い（ステップＳ２）、スタックポインタにスタックポインタ指定アドレスを設定する（ステップＳ３）。そして、内蔵デバイスの初期化（内蔵デバイス（内蔵周辺回路）であるタイマ回路５０９、パラレル入力ポート５１１およびパラレル出力ポート５１３の初期化など）を行った後（ステップＳ４）、ＲＡＭをアクセス可能状態に設定する（ステップＳ５）。

【０３０８】

次いで、ＣＰＵ５６は、Ｑレジスタに初期値Ｆ０Ｈをセットする（ステップＳ５Ａ）。すなわち、ステップＳ５が実行されてＲＡＭ５５をアクセス可能状態に設定したタイミングで、Ｑレジスタに初期値Ｆ０Ｈがセットされる。

【０３０９】

次いで、ＣＰＵ５６は、入力ポートを介して入力されるクリアスイッチ（例えば、電源基板に搭載されている。）の出力信号（クリア信号）の状態を確認する（ステップＳ６）。その確認においてオンを検出した場合には、ＣＰＵ５６は、通常の初期化処理（ステップＳ１０～Ｓ１５）を実行する。

【０３１０】

クリアスイッチがオンの状態でない場合には、遊技機への電力供給が停止したときにバックアップＲＡＭ領域のデータ保護処理（例えばパリティデータの付加等の電力供給停止時処理（電源断処理））が行われたか否か確認する（ステップＳ７）。そのような保護処理が行われていないことを確認したら、ＣＰＵ５６は初期化処理を実行する。バックアップＲＡＭ領域にバックアップデータがあるか否かは、例えば、電源断処理においてバックアップＲＡＭ領域に設定されるバックアップ監視タイマの値が判定値（例えば２）と同じ値になっていることによって、電力供給停止時処理の処理結果が保存されていることを確認できる。なお、バックアップ監視タイマに代えて、例えば、電源断処理においてバックアップフラグをセットするようにし、ステップＳ７では、バックアップフラグがセットされているか否かを確認するようにしてもよい。

【０３１１】

電力供給停止時処理が行われたことを確認したら、ＣＰＵ５６は、バックアップＲＡＭ領域のデータチェックを行う（ステップＳ８）。この実施の形態では、データチェックとしてパリティチェックを行う。よって、ステップＳ８では、算出したチェックサムと、電力供給停止時処理で同一の処理によって算出され保存されているチェックサムとを比較する。不測の停電等の電力供給停止が生じた後に復旧した場合には、バックアップＲＡＭ領域のデータは保存されているはずであるから、チェック結果（比較結果）は正常（一致）になる。チェック結果が正常でないということは、バックアップＲＡＭ領域のデータが、電力供給停止時のデータとは異なっていることを意味する。そのような場合には、内部状態を電力供給停止時の状態に戻すことができないので、電力供給の停止からの復旧時でない電源投入時に実行される初期化処理を実行する。

【０３１２】

チェック結果が正常であれば、ＣＰＵ５６は、遊技制御手段の内部状態と演出制御手段等の電気部品制御手段の制御状態を電力供給停止時の状態に戻すための遊技状態復旧処理（ステップＳ４１～Ｓ４３の処理）を行う。具体的には、ＲＯＭ５４に格納されているバックアップ時設定テーブルの先頭アドレスをポインタに設定し（ステップＳ４１）、バックアップ時設定テーブルの内容を順次作業領域（ＲＡＭ５５内の領域）に設定する（ステップＳ４２）。作業領域はバックアップ電源によって電源バックアップされている。バックアップ時設定テーブルには、作業領域のうち初期化してもよい領域についての初期化データが設定されている。ステップＳ４１およびＳ４２の処理によって、作業領域のうち初期化してはならない部分については、保存されていた内容がそのまま残る。初期化してはならない部分とは、例えば、電力供給停止前の遊技状態を示すデータ（特別図柄プロセスフラグ、確変フラグ、時短フラグなど）、出力ポートの出力状態が保存されている領域（出力ポートバッファ）、未払出賞球数を示すデータが設定されている部分などである。

【０３１３】

また、ＣＰＵ５６は、電力供給復旧時の初期化コマンドとしての停電復旧指定コマンドを送信する（ステップＳ４３）。また、ＣＰＵ５６は、バックアップＲＡＭに保存されている表示結果（確変大当り、通常大当り、突然確変大当り、小当り、またははずれ）を指定した表示結果指定コマンドを演出制御基板８０に対して送信する（ステップＳ４４）。そして、ステップＳ１５に移行する。

【０３１４】

なお、この実施の形態では、バックアップＲＡＭ領域には、後述する変動時間タイマの値も保存される。従って、停電復旧した場合には、ステップＳ４４で表示結果指定コマンドが送信された後、保存していた変動時間タイマの値の計測を再開して特別図柄の変動表示が再開されるとともに、保存していた変動時間タイマの値がタイムアウトしたときに、さらに後述する図柄確定指定コマンドが送信される。また、この実施の形態では、バックアップＲＡＭ領域には、後述する特別図柄プロセスフラグの値も保存される。従って、停電復旧した場合には、保存されている特別図柄プロセスフラグの値に応じたプロセスから特別図柄プロセス処理が再開される。

【０３１５】

なお、停電復旧時に必ず表示結果指定コマンドを送信するのではなく、ＣＰＵ５６は、まず、バックアップＲＡＭ領域に保存している変動時間タイマの値が０であるか否かを確認するようにしてもよい。そして、変動時間タイマの値が０でなければ、変動中に停電した場合であると判断して、表示結果指定コマンドを送信するようにし、変動時間タイマが０であれば、停電時に変動中の状態ではなかったと判断して、表示結果指定コマンドを送信しないようにしてもよい。

【０３１６】

また、ＣＰＵ５６は、まず、バックアップＲＡＭ領域に保存している特別図柄プロセスフラグの値が３であるか否かを確認するようにしてもよい。そして、特別図柄プロセスフラグの値が３であれば、変動中に停電した場合であると判断して、表示結果指定コマンドを送信するようにし、特別図柄プロセスフラグが３でなければ、停電時に変動中ではなかったと判断して、表示結果指定コマンドを送信しないようにしてもよい。

【０３１７】

なお、この実施の形態では、バックアップ監視タイマ（または、バックアップフラグ）とチェックデータとの双方を用いてバックアップＲＡＭ領域のデータが保存されているか否か確認しているが、いずれか一方のみを用いてもよい。すなわち、バックアップ監視タイマ（または、バックアップフラグ）とチェックデータとのいずれかを、遊技状態復旧処理を実行するための契機としてもよい。

【０３１８】

初期化処理では、ＣＰＵ５６は、まず、ＲＡＭクリア処理を行い（ステップＳ１０）、加算バッファおよびクリアバッファをクリアする（ステップＳ１０ａ）。加算バッファおよびクリアバッファは、ＲＡＭ５５形成され（すなわちＲＡＭ内の領域であって）、後述する入賞順異常報知処理において始動口への入賞順異常が発生したことを検出するために用いられる。なお、ＲＡＭクリア処理によって、所定のデータ（例えば、普通図柄当り判定用乱数を生成するためのカウンタのカウント値のデータ）は０に初期化されるが、任意の値またはあらかじめ決められている値に初期化するようにしてもよい。また、ＲＡＭ５５の全領域を初期化せず、所定のデータ（例えば、普通図柄当り判定用乱数を生成するためのカウンタのカウント値のデータ）をそのままにしてもよい。また、この実施の形態では、遊技状態復旧処理（ステップＳ４１～Ｓ４３の処理）が実行される場合には、通常の初期化処理（ステップＳ１０～Ｓ１３）が実行されないが、遊技状態復旧処理（ステップＳ４１～Ｓ４３の処理）が実行される場合にも、加算バッファおよびクリアバッファのクリア（ステップＳ１０ａ）が実行されるようにしてもよい。このようにすることで、電力供給停止前の状態に復旧させて入賞順異常報知処理を行うのではなく、新たな状態で入賞順異常報知処理を行うことができる。

【０３１９】

次いで、ＣＰＵ５６は、ＲＯＭ５４に格納されている初期化時設定テーブルの先頭アドレスをポインタに設定し（ステップＳ１１）、初期化時設定テーブルの内容を順次作業領域に設定する（ステップＳ１２）。

【０３２０】

ステップＳ１１およびＳ１２の処理によって、例えば、普通図柄当り判定用乱数カウンタ、特別図柄バッファ、総賞球数格納バッファ、特別図柄プロセスフラグなど制御状態に応じて選択的に処理を行うためのフラグに初期値が設定される。

【０３２１】

また、ＣＰＵ５６は、サブ基板（主基板３１以外のマイクロコンピュータが搭載された基板。）を初期化するための初期化指定コマンド（遊技制御用マイクロコンピュータ５６０が初期化処理を実行したことを示すコマンドでもある。）をサブ基板に送信する（ステップＳ１３）。例えば、演出制御用マイクロコンピュータ１００は、初期化指定コマンドを受信すると、演出表示装置９において、遊技機の制御の初期化がなされたことを報知するための画面表示、すなわち初期化報知を行う。

【０３２２】

そして、ステップＳ１５において、ＣＰＵ５６は、所定時間（例えば４ｍｓ）毎に定期的にタイマ割込がかかるように遊技制御用マイクロコンピュータ５６０に内蔵されているタイマ回路５０９のレジスタの設定を行なう。すなわち、初期値として例えば４ｍｓに相当する値が所定のレジスタ（時間定数レジスタ）に設定される。この実施の形態では、４ｍｓ毎に定期的にタイマ割込がかかるとする。

【０３２３】

初期化処理の実行（ステップＳ１０～Ｓ１５）が完了すると、ＣＰＵ５６は、メイン処理で、表示用乱数更新処理（ステップＳ１７）および初期値用乱数更新処理（ステップＳ１８）を繰り返し実行する。表示用乱数更新処理および初期値用乱数更新処理を実行するときには割込禁止状態に設定し（ステップＳ１６）、表示用乱数更新処理および初期値用乱数更新処理の実行が終了すると割込許可状態に設定する（ステップＳ１９）。この実施の形態では、表示用乱数とは、大当りとしない場合の特別図柄の停止図柄を決定するための乱数や大当りとしない場合にリーチとするか否かを決定するための乱数であり、表示用乱数更新処理とは、表示用乱数を発生するためのカウンタのカウント値を更新する処理である。また、初期値用乱数更新処理とは、初期値用乱数を発生するためのカウンタのカウント値を更新する処理である。この実施の形態では、初期値用乱数とは、普通図柄に関して当りとするか否か決定するための乱数を発生するためのカウンタ（普通図柄当り判定用乱数発生カウンタ）のカウント値の初期値を決定するための乱数である。後述する遊技の進行を制御する遊技制御処理（遊技制御用マイクロコンピュータ５６０が、遊技機に設けられている演出表示装置、可変入賞球装置、球払出装置等の遊技用の装置を、自身で制御する処理、または他のマイクロコンピュータに制御させるために指令信号を送信する処理、遊技装置制御処理ともいう）において、普通図柄当り判定用乱数のカウント値が１周（普通図柄当り判定用乱数の取りうる値の最小値から最大値までの間の数値の個数分歩進したこと）すると、そのカウンタに初期値が設定される。

【０３２４】

なお、この実施の形態では、リーチ演出は、演出表示装置９において可変表示される演出図柄を用いて実行される。また、特別図柄の表示結果を大当り図柄にする場合には、リーチ演出は常に実行される。特別図柄の表示結果を大当り図柄にしない場合には、遊技制御用マイクロコンピュータ５６０は、乱数を用いた抽選によって、リーチ演出を実行するか否か決定する。ただし、実際にリーチ演出の制御を実行するのは、演出制御用マイクロコンピュータ１００である。

【０３２５】

タイマ割込が発生すると、ＣＰＵ５６は、図５１に示すステップＳ２０～Ｓ３４のタイマ割込処理を実行する。タイマ割込処理において、まず、電源断信号が出力されたか否か（オン状態になったか否か）を検出する電源断検出処理を実行する（ステップＳ２０）。電源断信号は、例えば電源基板に搭載されている電源監視回路が、遊技機に供給される電源の電圧の低下を検出した場合に出力する。そして、電源断検出処理において、ＣＰＵ５６は、電源断信号が出力されたことを検出したら、必要なデータをバックアップＲＡＭ領域に保存するための電力供給停止時処理を実行する。次いで、入力ドライバ回路５８を介して、ゲートスイッチ３２ａ、第１始動口スイッチ１３ａ、第２始動口スイッチ１４ａおよびカウントスイッチ２３の検出信号を入力し、それらの状態判定を行う（スイッチ処理：ステップＳ２１）。

【０３２６】

次に、ＣＰＵ５６は、第１始動入賞口１３と第２始動入賞口１４との入賞順異常を検出し入賞順異常を報知するための入賞順異常報知処理を実行する（ステップＳ２１ａ）。

【０３２７】

なお、「入賞順異常」とは、第１始動入賞口１３および第２始動入賞口１４に遊技球が入賞した順序（具体的には、第１始動口スイッチ１３ａおよび第２始動口スイッチ１４ａによって遊技球が検出された順序）が所定の順序と異なることである。この実施の形態では、既に説明したように、振分装置２００に流入した遊技球は振分部材２０２によって第

１始動入賞口１３と第２始動入賞口１４とに交互に入賞するように構成されている（ただし、確変状態（高ベース状態）には、振分装置２００の状態にかかわらず第２始動入賞口１４に入賞可能な場合もある）ので、第１始動入賞口１３または第２始動入賞口１４に遊技球が連続して入賞することはない（ただし、確変状態（高ベース状態）では第２始動入賞口１４に連続して入賞する場合がある）。そのため、この実施の形態では、第１始動入賞口１３または第２始動入賞口１４のいずれかに所定数以上（本例では４以上）連続して入賞した場合に入賞順異常が検出され、後述するような入賞順異常を報知する制御が行われる。

【０３２８】

次に、ＣＰＵ５６は、第１特別図柄表示器８ａ、第２特別図柄表示器８ｂ、普通図柄表示器１０、第１特別図柄保留記憶表示器１８ａ、第２特別図柄保留記憶表示器１８ｂ、普通図柄保留記憶表示器４１の表示制御を行う表示制御処理を実行する（ステップＳ２２）。第１特別図柄表示器８ａ、第２特別図柄表示器８ｂおよび普通図柄表示器１０については、ステップＳ３２，Ｓ３３で設定される出力バッファの内容に応じて各表示器に対して駆動信号を出力する制御を実行する。

【０３２９】

また、遊技制御に用いられる普通図柄当り判定用乱数等の各判定用乱数を生成するための各カウンタのカウント値を更新する処理を行う（判定用乱数更新処理：ステップＳ２３）。ＣＰＵ５６は、さらに、初期値用乱数および表示用乱数を生成するためのカウンタのカウント値を更新する処理を行う（初期値用乱数更新処理，表示用乱数更新処理：ステップＳ２４，Ｓ２５）。

【０３３０】

さらに、ＣＰＵ５６は、特別図柄プロセス処理を行う（ステップＳ２６）。特別図柄プロセス処理では、第１特別図柄表示器８ａ、第２特別図柄表示器８ｂおよび大入賞口を所定の順序で制御するための特別図柄プロセスフラグに従って該当する処理を実行する。ＣＰＵ５６は、特別図柄プロセスフラグの値を、遊技状態に応じて更新する。

【０３３１】

次いで、普通図柄プロセス処理を行う（ステップＳ２７）。普通図柄プロセス処理では、ＣＰＵ５６は、普通図柄表示器１０の表示状態を所定の順序で制御するための普通図柄プロセスフラグに従って該当する処理を実行する。ＣＰＵ５６は、普通図柄プロセスフラグの値を、遊技状態に応じて更新する。

【０３３２】

また、ＣＰＵ５６は、演出制御用マイクロコンピュータ１００に演出制御コマンドを送出する処理を行う（演出制御コマンド制御処理：ステップＳ２８）。

【０３３３】

さらに、ＣＰＵ５６は、例えばホール管理用コンピュータに供給される大当り情報、始動情報、確率変動情報などのデータを出力する情報出力処理を行う（ステップＳ２９）。

【０３３４】

また、ＣＰＵ５６は、第１始動口スイッチ１３ａ、第２始動口スイッチ１４ａおよびカウントスイッチ２３の検出信号にもとづく賞球個数の設定などを行う賞球処理を実行する（ステップＳ３０）。具体的には、第１始動口スイッチ１３ａ、第２始動口スイッチ１４ａおよびカウントスイッチ２３のいずれかがオンしたことにもとづく入賞検出に応じて、払出制御基板３７に搭載されている払出制御用マイクロコンピュータに賞球個数を示す払出制御コマンド（賞球個数信号）を出力する。払出制御用マイクロコンピュータは、賞球個数を示す払出制御コマンドに応じて球払出装置９７を駆動する。

【０３３５】

この実施の形態では、出力ポートの出力状態に対応したＲＡＭ領域（出力ポートバッファ）が設けられているのであるが、ＣＰＵ５６は、出力ポートの出力状態に対応したＲＡＭ領域におけるソレノイドのオン／オフに関する内容を出力ポートに出力する（ステップＳ３１：出力処理）。

【０３３６】
また、ＣＰＵ５６は、特別図柄プロセスフラグの値に応じて特別図柄の演出表示を行うための特別図柄表示制御データを特別図柄表示制御データ設定用の出力バッファに設定する特別図柄表示制御処理を行う（ステップＳ３２）。
【０３３７】
さらに、ＣＰＵ５６は、普通図柄プロセスフラグの値に応じて普通図柄の演出表示を行うための普通図柄表示制御データを普通図柄表示制御データ設定用の出力バッファに設定する普通図柄表示制御処理を行う（ステップＳ３３）。ＣＰＵ５６は、例えば、普通図柄の変動に関する開始フラグがセットされると終了フラグがセットされるまで、普通図柄の変動速度が０．２秒ごとに表示状態（「○」および「×」）を切り替えるような速度であれば、０．２秒が経過する毎に、出力バッファに設定される表示制御データの値（例えば、「○」を示す１と「×」を示す０）を切り替える。また、ＣＰＵ５６は、出力バッファに設定された表示制御データに応じて、ステップＳ２２において駆動信号を出力することによって、普通図柄表示器１０における普通図柄の演出表示を実行する。
【０３３８】
その後、割込許可状態に設定し（ステップＳ３４）、処理を終了する。
【０３３９】
以上の制御によって、この実施の形態では、遊技制御処理は４ｍｓ毎に起動されることになる。なお、遊技制御処理は、タイマ割込処理におけるステップＳ２１～Ｓ３３（ステップＳ２９を除く。）の処理に相当する。また、この実施の形態では、タイマ割込処理で遊技制御処理が実行されているが、タイマ割込処理では例えば割込が発生したことを示すフラグのセットのみがなされ、遊技制御処理はメイン処理において実行されるようにしてもよい。
【０３４０】
第１特別図柄表示器８ａまたは第２特別図柄表示器８ｂおよび演出表示装置９にはずれ図柄が停止表示される場合には、演出図柄の可変表示が開始されてから、演出図柄の可変表示状態がリーチ状態にならずに、リーチにならない所定の演出図柄の組み合わせが停止表示されることがある。このような演出図柄の可変表示態様を、可変表示結果がはずれ図柄になる場合における「非リーチ」（「通常はずれ」ともいう）の可変表示態様という。
【０３４１】
第１特別図柄表示器８ａまたは第２特別図柄表示器８ｂおよび演出表示装置９にはずれ図柄が停止表示される場合には、演出図柄の可変表示が開始されてから、演出図柄の可変表示状態がリーチ状態となった後にリーチ演出が実行され、最終的に大当り図柄とはならない所定の演出図柄の組み合わせが停止表示されることがある。このような演出図柄の可変表示結果を、可変表示結果が「はずれ」となる場合における「リーチ」（「リーチはずれ」ともいう）の可変表示態様という。
【０３４２】
この実施の形態では、第１特別図柄表示器８ａまたは第２特別図柄表示器８ｂに大当り図柄が停止表示される場合には、演出図柄の可変表示状態がリーチ状態になった後にリーチ演出が実行され、最終的に演出表示装置９における「左」、「中」、「右」の各図柄表示エリア９Ｌ、９Ｃ、９Ｒに、演出図柄が揃って停止表示される。
【０３４３】
第１特別図柄表示器８ａまたは第２特別図柄表示器８ｂに小当りである「５」が停止表示される場合には、演出表示装置９において、演出図柄の可変表示態様が「突然確変大当り」である場合と同様に演出図柄の可変表示が行われた後、所定の小当り図柄（突然確変大当り図柄と同じ図柄。例えば「１３５」）が停止表示されることがある。第１特別図柄表示器８ａまたは第２特別図柄表示器８ｂに小当り図柄である「５」が停止表示されることに対応する演出表示装置９における表示演出を「小当り」の可変表示態様という。
【０３４４】
ここで、小当りとは、大当りと比較して大入賞口の開放回数が少ない回数（この実施の

形態では０．１秒間の開放を２回）まで許容される当りである。なお、小当り遊技が終了した場合、遊技状態は変化しない。すなわち、確変状態から通常状態に移行したり通常状態から確変状態に移行したりすることはない。また、突然確変大当りとは、大当り遊技状態において大入賞口の開放回数が少ない回数（この実施の形態では０．１秒間の開放を２回）まで許容されるが大入賞口の開放時間が極めて短い大当りであり、かつ、大当り遊技後の遊技状態を確変状態に移行させるような大当りである（すなわち、そのようにすることにより、遊技者に対して突然に確変状態となったかのように見せるものである）。つまり、この実施の形態では、突然確変大当りと小当りとは、大入賞口の開放パターンが同じである。そのように制御することによって、大入賞口の０．１秒間の開放が２回行われると、突然確変大当りであるか小当りであるかまでは認識できないので、遊技者に対して高確率状態（確変状態）を期待させることができ、遊技の興趣を向上させることができる。

【０３４５】

図５２および図５３は、ステップＳ２０の電源断処理の一例を示すフローチャートである。電源断処理において、遊技制御用マイクロコンピュータ５６０は、まず、電源断信号が出力されているか否か（オン状態になっているか否か）確認する（ステップＳ４５０）。オン状態でなければ、ＲＡＭ５５に形成されているバックアップ監視タイマの値を０クリアする（ステップＳ４５１）。オン状態であれば、バックアップ監視タイマの値を１増やす（ステップＳ４５２）。そして、バックアップ監視タイマの値が判定値（例えば２）と一致すれば（ステップＳ４５３）、ステップＳ４５４以降の電力供給停止時処理すなわち電力の供給停止のための準備処理を実行する。つまり、遊技の進行を制御する状態から遊技状態を保存させるための電力供給停止時処理（電源断時制御処理）を実行する状態に移行する。なお、「ＲＡＭに形成されている」とは、ＲＡＭ内の領域であることを意味する。

【０３４６】

バックアップ監視タイマと判定値とを用いることによって、判定値に相当する時間だけ電源断信号のオン状態が継続したら、電力供給停止時処理が開始される。すなわち、ノイズ等で一瞬電源断信号のオン状態が発生しても、誤って電力供給停止時処理が開始されるようなことはない。なお、バックアップ監視タイマの値は、遊技機への電力供給が停止しても、所定期間はバックアップ電源によって保存される。従って、メイン処理におけるステップＳ８では、バックアップ監視タイマの値が判定値と同じ値になっていることによって、電力供給停止時処理の処理結果が保存されていることを確認できる。

【０３４７】

電力供給停止時処理において、遊技制御用マイクロコンピュータ５６０は、パリティデータを作成する（ステップＳ４５４～Ｓ４６３）。すなわち、まず、クリアデータ（００）をチェックサムデータエリアにセットし（ステップＳ４５４）、電力供給停止時でも内容が保存されるべきＲＡＭ領域の先頭アドレスに相当するチェックサム算出開始アドレスをポインタにセットする（ステップＳ４５５）。また、電力供給停止時でも内容が保存されるべきＲＡＭ領域の最終アドレスに相当するチェックサム算出回数をセットする（ステップＳ４５６）。

【０３４８】

次いで、チェックサムデータエリアの内容とポインタが指すＲＡＭ領域の内容との排他的論理和を演算する（ステップＳ４５７）。演算結果をチェックサムデータエリアにストアするとともに（ステップＳ４５８）、ポインタの値を１増やし（ステップＳ４５９）、チェックサム算出回数の値を１減算する（ステップＳ４６０）。そして、ステップＳ４５７～Ｓ４６０の処理を、チェックサム算出回数の値が０になるまで繰り返す（ステップＳ４６１）。

【０３４９】

チェックサム算出回数の値が０になったら、遊技制御用マイクロコンピュータ５６０は、チェックサムデータエリアの内容の各ビットの値を反転する（ステップＳ４６２）。そして、反転後のデータをチェックサムデータエリアにストアする（ステップＳ４６３）。

このデータが、電源投入時にチェックされるパリティデータになる。次いで、ＲＡＭアクセスレジスタにアクセス禁止値を設定する（ステップＳ４７１）。以後、内蔵ＲＡＭ５５のアクセスができなくなる。

【０３５０】

さらに、遊技制御用マイクロコンピュータ５６０は、ＲＯＭ５４に格納されているポートクリア設定テーブルの先頭アドレスをポインタにセットする（ステップＳ４７２）。ポートクリア設定テーブルにおいて、先頭アドレスには処理数（クリアすべき出力ポートの数）が設定され、次いで、出力ポートのアドレスおよび出力値データ（クリアデータ：出力ポートの各ビットのオフ状態の値）が、処理数分の出力ポートについて順次設定されている。

【０３５１】

遊技制御用マイクロコンピュータ５６０は、ポインタが指すアドレスのデータ（すなわち処理数）をロードする（ステップＳ４７３）。また、ポインタの値を１増やし（ステップＳ４７４）、ポインタが指すアドレスのデータ（すなわち出力ポートのアドレス）をロードする（ステップＳ４７５）。さらに、ポインタの値を１増やし（ステップＳ４７６）、ポインタが指すアドレスのデータ（すなわち出力値データ）をロードする（ステップＳ４７７）。そして、出力値データを出力ポートに出力する（ステップＳ４７８）。その後、処理数を１減らし（ステップＳ４７９）、処理数が０でなければステップＳ４７４に戻る。処理数が０であれば、すなわち、クリアすべき出力ポートを全てクリアしたら、タイマ割込を停止し（ステップＳ４８１）、ループ処理に入る。なお、出力ポートをクリアする処理をチェックサムデータを作成する処理の前に実行してもよい。例えば、ＣＰＵ５６は、ステップＳ４５３でＹと判定した後、直ちにステップＳ４７２～Ｓ４８０の出力ポートクリアの処理を実行するようにしてもよい。

【０３５２】

ループ処理では、電源断信号がオフ状態になったか否かを監視する（ステップＳ４８２）。そして、電源断信号がオン状態の間は（ステップＳ４８２のＹ）、ステップＳ４８２の処理を繰り返し実行して待機する。

【０３５３】

これに対して、ステップＳ４８２にて電源断信号がオフ状態となったときには（ステップＳ４８２のＮ）、所定の電断復旧時における設定を行った後（ステップＳ４８７）、図５０に示すメイン処理の先頭にリターンする。一例として、ステップＳ４８７の処理では、ＣＰＵ５６に内蔵されたスタックポインタに電源断復旧時ベクタテーブルの記憶アドレスを格納し、遊技制御用タイマ割込み処理から復帰（リターン）させる。ここで、電源断復旧時ベクタテーブルは、ＲＯＭ５４に記憶された制御コード（遊技制御プログラム）の先頭アドレスを指定するものであればよい。図５１に示すタイマ割込処理のような割込処理から復帰（リターン）するときには、スタックポインタで指定されるアドレスの記憶データが復帰アドレスとして読み出される。こうして、ステップＳ４８７の処理を実行した後には、ＣＰＵ５６により、ＲＯＭ５４に記憶されている制御コードの先頭から、遊技制御の実行を開始（再開）させることができる。

【０３５４】

以上の処理によって、電力供給が停止する場合には、ステップＳ４５４～Ｓ４８１の電力供給停止時処理が実行され、電力供給停止時処理が実行されたことを示すデータ（判定値になっているバックアップ監視タイマのおよびチェックサム）がバックアップＲＡＭへストアされ、ＲＡＭアクセスが禁止状態にされ、出力ポートがクリアされ、かつ、遊技制御処理を実行するためのタイマ割込が禁止状態に設定される。

【０３５５】

なお、この実施の形態では、電源断処理においてチェックデータの生成や出力ポートクリアなどの処理を終了した後、ステップＳ４８２で繰り返し電源断信号の入力を確認する電源断待ちループに移行するが、このような電源断信号の入力を確認する構成とはしないようにしてもよい。この場合、例えば、ウオッチドッグタイマ（ＷＴＤ）５０６ｂをユー

ザプログラムによって起動できるように設定した上で、電源断待ちループに入るときにウオッチドッグタイマ（ＷＴＤ）５０６ｂを起動させるようにし、遊技機が完全に電源断状態とならず電源の電圧値が完全に落ちきらないときにはウオッチドッグタイマ（ＷＴＤ）５０６ｂからのタイムアウト信号によるリセットが発生するようにしてもよい。

【０３５６】

また、例えば、遊技機への電源投入時に電源断信号が入力されているか否かを確認し、入力されていれば無限ループに移行したり、その無限ループにおいて電源断信号の入力を確認し、その入力がなくなるまで無限ループを継続したりするように構成されている場合、上記と同様の態様で無限ループに入るときにウオッチドッグタイマ（ＷＴＤ）５０６ｂを起動させ、同様の処理を行うように構成してもよい。

【０３５７】

なお、ウオッチドッグタイマ（ＷＴＤ）５０６ｂを起動させるように設定している場合には（ただし、上記のように電源断待ちループでのみウオッチドッグタイマ（ＷＴＤ）５０６ｂを起動させるものを除く）、正常にＣＰＵ５６が動作しているときにはタイムアウトしないように、ウオッチドッグタイマ（ＷＴＤ）５０６ｂをクリアする信号を出力するようにプログラムされる。具体的には、内蔵レジスタエリアに設けられているＷＤＴクリアレジスタ（図示せず）にクリアするための値を書き込むようにプログラムされる。

【０３５８】

この実施の形態では、ＲＡＭ５５がバックアップ電源によって電源バックアップ（遊技機への電力供給が停止しても所定期間はＲＡＭ５５の内容が保存されこと）されている。この例では、ステップＳ４５２～Ｓ４７９の処理によって、バックアップ監視タイマの値とともに、電源断信号が出力されたときのＲＡＭ５５の内容にもとづくチェックサムもＲＡＭ５５のバックアップ領域に保存される。遊技機への電力供給が停止した後、所定期間内に電力供給が復旧したら、遊技制御手段は、上述したステップＳ４１～Ｓ４４の処理によって、ＲＡＭ５５に保存されているデータ（電力供給が停止した直前の遊技制御手段による制御状態である遊技状態を示すデータ（例えば、プロセスフラグの状態、大当り中フラグの状態、確変フラグの状態、出力ポートの出力状態等）を含む）に従って、遊技状態を、電力供給が停止した直前の状態に戻すことができる。なお、電力供給停止の期間が所定期間を越えたらバックアップ監視タイマの値とチェックサムとが正規の値とは異なるはずであるから、その場合には、ステップＳ１０～Ｓ１３の初期化処理が実行される。

【０３５９】

以上のように、電力供給停止時処理（電力の供給停止のための準備処理）によって、遊技状態を電力供給が停止した直前の状態に戻すためのデータが確実に変動データ記憶手段（この例ではＲＡＭ５５の一部の領域）に保存される。よって、停電等による電源断が生じても、所定期間内に電源が復旧すれば、遊技状態を電力供給が停止した直前の状態に戻すことができる。

【０３６０】

また、電源断信号がオフ状態になった場合には、ステップＳ１に戻る。その場合、電力供給停止時処理が実行されたことを示すデータが設定されているので、ステップＳ４１～Ｓ４４の復旧処理が実行される。よって、電力供給停止時処理を実行した後に払出制御基板３７からの電源断信号がオフ状態になったときには、遊技の進行を制御する状態に戻る。従って、電源瞬断等が生じても、遊技制御処理が停止してしまうようなことはなく、自動的に、遊技制御処理が続行される。

【０３６１】

なお、図４８、図５０～図５３に示すような態様で遊技機が動作することによって、この実施の形態では、所定の処理（本例では、図５０に示すメイン処理のステップＳ１６～Ｓ１９のループ処理と、図５１のタイマ割込処理）の実行中に所定事象が発生（ＩＡＴ回路５０６ａからＩＡＴ信号を入力、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからタイムアウト信号を入力）したときにＲＡＭ５５（バックアップＲＡＭ）の記憶内容を初期化する初期化処理を実行する。具体的には、図５０に示すメイン処理のステップＳ１６～Ｓ１

９のループ処理や、図５１のタイマ割込処理の実行中に、ＩＡＴ回路５０６ａからＩＡＴ信号や、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからタイムアウト信号を入力したときには、図５２および図５３に示す電源断処理が実行されることなく、システムリセットまたはユーザリセット（図４８のステップＳ１００４，Ｓ１０１４参照）が発生することになる。そして、システムリセットが発生した後にはセキュリティチェックを実行した後にステップＳ１００７でユーザモードに移行されて図５０に示すメイン処理の実行が再び開始され、ユーザリセットが発生した後にはステップＳ１０１５でユーザプログラムの先頭に戻り図５０に示すメイン処理の実行が再び開始されるのであるが、バックアップ監視タイマ（または、バックアップフラグ）およびチェックデータのいずれもセットされていないことから、図５０に示すメイン処理でステップＳ１０に移行して初期化処理が実行されることになる。

【０３６２】

なお、「所定の処理」とは、遊技機への電源投入時の初期化処理や復旧処理が実行された後、遊技可能となった状態で実行されている処理であり、上記に説明したように、具体的には、図５０に示すメイン処理のステップＳ１６～Ｓ１９のループ処理と、図５１のタイマ割込処理とが該当する。ただし、電源電圧低下により電源断処理が実行されているときは所定の処理からは除かれる。

【０３６３】

なお、図４８、図５０～図５３に示すような態様で遊技機が動作することによって、電力供給停止時処理（本例では、図５２および図５３に示す電源断処理）を実行した後に所定事象が発生（ＩＡＴ回路５０６ａからＩＡＴ信号を入力、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからタイムアウト信号を入力）したときに、ＲＡＭ５５（バックアップＲＡＭ）の記憶内容にもとづいて制御状態を電力供給停止時処理を開始したときの状態に復旧させる復旧処理を実行する。具体的には、電源断信号を入力したことにもとづいて図５２および図５３に示す電源断処理を実行して図５３のループ処理を実行しているときに、ちょうどＩＡＴ回路５０６ａからＩＡＴ信号や、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからタイムアウト信号を入力したときには（ただし、ＩＡＴ回路５０６ａからＩＡＴ信号した場合には、何らかの原因で指定エリア外のプログラムを実行している状態となってしまった場合であるので、より正確には、電源断処理のループ処理に一度移行した後にＩＡＴ回路５０６ａからＩＡＴ信号した場合に相当する）、ＩＡＴ信号やタイムアウト信号を入力したことにもとづいてシステムリセットまたはユーザリセット（図４８のステップＳ１００４，Ｓ１０１４参照）が発生することになる。そして、システムリセットが発生した後にはセキュリティチェックを実行した後にステップＳ１００７でユーザモードに移行されて図５０に示すメイン処理の実行が再び開始され、ユーザリセットが発生した後にはステップＳ１０１５でユーザプログラムの先頭に戻り図５０に示すメイン処理の実行が再び開始されるのであるが、バックアップ監視タイマ（または、バックアップフラグ）およびチェックデータの両方がセットされていることから、図５０に示すメイン処理でステップＳ７およびステップＳ８でいずれもＹと判定されてステップＳ４１～Ｓ４４の復旧処理が実行されることになる。

【０３６４】

図５４は、この実施の形態で用いられる各ソフトウェア乱数を示す説明図である。各ソフトウェア乱数は、以下のように使用される。なお、前述したように、この実施の形態では、大当りとするか否かを判定するための大当り判定用乱数（ランダムＲ）については、１６ビット乱数回路５０８ｂが出力するハードウェア乱数が用いられる。

（１）ランダム１（ＭＲ１）：大当りの種類（後述する通常大当り、確変大当り、突然確変大当り）を決定する（大当り種別判定用）

（２）ランダム２（ＭＲ２）：変動パターンの種類（種別）を決定する（変動パターン種別判定用）

（３）ランダム３（ＭＲ３）：変動パターン（変動時間）を決定する（変動パターン判定用）

（４）ランダム４（ＭＲ４）：普通図柄にもとづく当りを発生させるか否か決定する（普通図柄当り判定用）
（５）ランダム５（ＭＲ５）：ランダム４の初期値を決定する（ランダム４初期値決定用）

【０３６５】
なお、この実施の形態では、大当り判定用乱数（ランダムＲ）についてのみ乱数回路から抽出したハードウェア乱数を用い、それ以外の乱数についてはソフトウェア乱数を用いる場合を示しているが、この実施の形態で示したものにかぎられない。例えば、大当り判定用乱数（ランダムＲ）に加えて図５４に示すランダム１～５の全てについて乱数回路から抽出したハードウェア乱数を用いるようにしてもよい。また、図５４に示すランダム１～５のうちの一部の乱数についてのみ乱数回路から抽出したハードウェア乱数を用い、それ以外についてはソフトウェア乱数を用いるように構成しても構わない。

【０３６６】
なお、この実施の形態では、変動パターンは、まず、変動パターン種別判定用乱数（ランダム２）を用いて変動パターン種別判定用テーブルを参照することにより変動パターン種別を決定し、変動パターン判定用乱数（ランダム３）を用いて変動パターン判定用テーブルを参照することにより、決定した変動パターン種別に含まれるいずれかの変動パターンに決定する。そのように、この実施の形態では、２段階の抽選処理によって変動パターンが決定される。

【０３６７】
なお、変動パターン種別とは、複数の変動パターンをその変動態様の特徴に従ってグループ化したものである。例えば、複数の変動パターンをリーチの種類でグループ化して、ノーマルリーチを伴う変動パターンを含む変動パターン種別と、スーパーリーチＡを伴う変動パターンを含む変動パターン種別と、スーパーリーチＢを伴う変動パターンを含む変動パターン種別とに分けてもよい。また、例えば、複数の変動パターンを擬似連の再変動の回数でグループ化して、擬似連を伴わない変動パターンを含む変動パターン種別と、再変動１回の変動パターンを含む変動パターン種別と、再変動２回の変動パターンを含む変動パターン種別と、再変動３回の変動パターンを含む変動パターン種別とに分けてもよい。また、例えば、複数の変動パターンを擬似連や滑り演出などの特定演出の有無でグループ化してもよい。

【０３６８】
ここで、本実施の形態では、擬似連の演出は、再変動回数（以下擬似連回数）が多い方が、可変表示結果が大当りとなる期待度が高い。また、ノーマルリーチを伴う変動パターンは、スーパーリーチを伴う変動パターンよりも、可変表示結果が「はずれ」のときに選択されやすい。逆に、スーパーリーチを伴う変動パターンは、ノーマルリーチを伴う変動パターンよりも、可変表示結果が「大当り」のときに選択されやすい。したがって、この実施の形態では、ノーマルリーチを伴う変動パターンよりもスーパーリーチを伴う変動パターンの方が、可変表示結果が大当りとなる期待度が高い。なお、「可変表示結果が大当りとなる期待度（信頼度）」とは、その特定の演出態様による可変表示（例えば、スーパーリーチを伴う可変表示）が実行された場合に大当りが出現する出現率（確率）を示している。例えば、スーパーリーチを伴う可変表示が実行される場合の大当りとなる期待度は、（大当りと決定されている場合にスーパーリーチが実行される割合）／（大当りと決定されている場合およびはずれと決定されている場合の両方にスーパーリーチが実行される割合）を計算することによって求められる。

【０３６９】
また、本実施の形態の低確低ベース中におけるはずれ変動での各変動パターンの選択割合は、例えば次の変動パターンテーブルに設定されている。変動結果がはずれのときには、非リーチはずれの変動パターンが５０％の割合で選択され、ノーマルはずれの変動パターンが３０％の割合で選択され、擬似連１回を伴うノーマルはずれの変動パターンが１０％の割合で選択され、スーパーはずれが５％の割合で選択され、変動パターンが擬似連２

回を伴うスーパーはずれが４％の割合で選択され、擬似連３回を伴うスーパーはずれが１％の割合で選択される。

【０３７０】

さらに、本実施の形態の低確低ベース中における当り変動での各変動パターンの選択割合は、例えば次のようになっている。ノーマル当りの変動パターンが５％の割合で選択され、擬似連１回を伴うノーマル当りの変動パターンが１０％の割合で選択され、スーパー当りが２０％の割合で選択され、擬似連２回を伴うスーパー当りの変動パターンが３０％の割合で選択され、擬似連３回を伴うスーパー当りの変動パターンが３５％の割合で選択される。また、突然確変当りの変動及び小当りの変動に対応する変動パターンが別途設けられている。なお、低確高ベース、高確低ベース及び高確高ベースについての変動パターンテーブルも別途設定されている（いずれも図示せず）。

【０３７１】

図５１に示された遊技制御処理におけるステップＳ２３では、遊技制御用マイクロコンピュータ５６０は、（１）の大当り種別判定用乱数、および（４）の普通図柄当り判定用乱数を生成するためのカウンタのカウントアップ（１加算）を行う。すなわち、それらが判定用乱数であり、それら以外の乱数が表示用乱数（ランダム２、ランダム３）または初期値用乱数（ランダム５）である。なお、遊技効果を高めるために、上記の乱数以外の乱数も用いてもよい。例えば、大当り種別判定用乱数（ランダム１）の初期値を決定するための初期値決定用乱数を設けるようにしてもよい。また、この実施の形態では、大当り判定用乱数として、遊技制御用マイクロコンピュータ５６０に内蔵されたハードウェア（遊技制御用マイクSロコンピュータ５６０の外部のハードウェアでもよい。）が生成する乱数を用いる。

【０３７２】

図５５（Ａ）は、大当り判定テーブルを示す説明図である。大当り判定テーブルとは、ＲＯＭ５４に記憶されているデータの集まりであって、ランダムＲと比較される大当り判定値が設定されているテーブルである。大当り判定テーブルには、通常状態（確変状態でない遊技状態）において用いられる通常時大当り判定テーブルと、確変状態において用いられる確変時大当り判定テーブルとがある。通常時大当り判定テーブルには、図５５（Ａ）の左欄に記載されている各数値が設定され、確変時大当り判定テーブルには、図５５（Ａ）の右欄に記載されている各数値が設定されている。図５５（Ａ）に記載されている数値が大当り判定値である。

【０３７３】

図５５（Ｂ），（Ｃ）は、小当り判定テーブルを示す説明図である。小当り判定テーブルとは、ＲＯＭ５４に記憶されているデータの集まりであって、ランダムＲと比較される小当り判定値が設定されているテーブルである。小当り判定テーブルには、第１特別図柄の変動表示を行うときに用いられる小当り判定テーブル（第１特別図柄用）と、第２特別図柄の変動表示を行うときに用いられる小当り判定テーブル（第２特別図柄用）とがある。小当り判定テーブル（第１特別図柄用）には、図５５（Ｂ）に記載されている各数値が設定され、小当り判定テーブル（第２特別図柄用）には、図５５（Ｃ）に記載されている各数値が設定されている。また、図５５（Ｂ），（Ｃ）に記載されている数値が小当り判定値である。

【０３７４】

なお、第１特別図柄の変動表示を行う場合にのみ小当りと決定するようにし、第２特別図柄の変動表示を行う場合には小当りを設けないようにしてもよい。この場合、図５５（Ｃ）に示す第２特別図柄用の小当り判定テーブルは設けなくてもよい。この実施の形態では、遊技状態が確変状態に移行されているときには主として第２特別図柄の変動表示が実行される。遊技状態が確変状態に移行されているときにも小当りが発生するようにし、確変となるか否かを煽る演出を行うように構成すると、現在の遊技状態が確変状態であるにもかかわらず却って遊技者に煩わしさを感じさせてしまう。そこで、第２特別図柄の変動表示中は小当りが発生しないように構成すれば、遊技状態が確変状態である場合には小当

りが発生しにくくし必要以上に確変に対する煽り演出を行わないようにすることができ、遊技者に煩わしさを感じさせる事態を防止することができる。

【０３７５】

ＣＰＵ５６は、所定の時期に、１６ビット乱数回路５０８ｂのカウント値を抽出して抽出値を大当り判定用乱数（ランダムＲ）の値とするのであるが、大当り判定用乱数値が図５５（Ａ）に示すいずれかの大当り判定値に一致すると、特別図柄に関して大当り（後述する１５Ｒ確変大当り、７Ｒ確変大当り、突然確変大当り）にすることに決定する。また、大当り判定用乱数値が図５５（Ｂ），（Ｃ）に示すいずれかの小当り判定値に一致すると、特別図柄に関して小当りにすることに決定する。なお、図５５（Ａ）に示す「確率」は、大当りになる確率（割合）を示す。また、図５５（Ｂ），（Ｃ）に示す「確率」は、小当りになる確率（割合）を示す。また、大当りにするか否か決定するということは、大当り遊技状態に移行させるか否か決定するということであるが、第１特別図柄表示器８ａまたは第２特別図柄表示器８ｂにおける停止図柄を大当り図柄にするか否か決定するということでもある。また、小当りにするか否か決定するということは、小当り遊技状態に移行させるか否か決定するということであるが、第１特別図柄表示器８ａまたは第２特別図柄表示器８ｂにおける停止図柄を小当り図柄にするか否か決定するということでもある。

【０３７６】

なお、この実施の形態では、図５５（Ｂ），（Ｃ）に示すように、小当り判定テーブル（第１特別図柄用）を用いる場合には３００分の１の割合で小当りと決定されるのに対して、小当り判定テーブル（第２特別図柄）を用いる場合には３０００分の１の割合で小当りと決定される場合を説明する。従って、この実施の形態では、第１始動入賞口１３に始動入賞して第１特別図柄の変動表示が実行される場合には、第２始動入賞口１４に始動入賞して第２特別図柄の変動表示が実行される場合と比較して、「小当り」と決定される割合が高い。

【０３７７】

図５５（Ｄ），（Ｅ）は、ＲＯＭ５４に記憶されている大当り種別判定テーブル１３１ａ，１３１ｂを示す説明図である。このうち、図５５（Ｄ）は、遊技球が第１始動入賞口１３に入賞したことにもとづく保留記憶を用いて（すなわち、第１特別図柄の変動表示が行われるとき）大当り種別を決定する場合の大当り種別判定テーブル（第１特別図柄用）１３１ａである。また、図５５（Ｅ）は、遊技球が第２始動入賞口１４に入賞したことにもとづく保留記憶を用いて（すなわち、第２特別図柄の変動表示が行われるとき）大当り種別を決定する場合の大当り種別判定テーブル（第２特別図柄用）１３１ｂである。

【０３７８】

大当り種別判定テーブル１３１ａ，１３１ｂは、可変表示結果を大当り図柄にする旨の判定がなされたときに、大当り種別判定用の乱数（ランダム１）にもとづいて、大当りの種別を「１５Ｒ確変大当り」、「７Ｒ確変大当り」、「突然確変大当り」のうちのいずれかに決定するために参照されるテーブルである。なお、この実施の形態では、図５５（Ｄ），（Ｅ）に示すように、大当り種別判定テーブル１３１ａには「突然確変大当り」に対して５個の判定値が割り当てられている（４０分の５の割合で突然確変大当りと決定される）のに対して、大当り種別判定テーブル１３１ｂには「突然確変大当り」に対して１個の判定値が割り当てられている（４０分の１の割合で突然確変大当りと決定される）場合を説明する。従って、この実施の形態では、第１始動入賞口１３に始動入賞して第１特別図柄の変動表示が実行される場合には、第２始動入賞口１４に始動入賞して第２特別図柄の変動表示が実行される場合と比較して、「突然確変大当り」と決定される割合が高い。なお、第１特別図柄用の大当り種別判定テーブル１３１ａにのみ「突然確変大当り」を振り分けるようにし、第２特別図柄用の大当り種別判定テーブル１３１ｂには「突然確変大当り」の振り分けを行わない（すなわち、第１特別図柄の変動表示を行う場合にのみ、「突然確変大当り」と決定される場合がある）ようにしてもよい。

【０３７９】

また、図５５（Ｄ），（Ｅ）に示すように、この実施の形態では、大当り種別判定テー

ブル１３１ｂの方が、大当り種別判定テーブル１３１ａと比較して、「１５Ｒ確変大当り」に対して多くの判定値が割り振られている。従って、この実施の形態では、第２始動入賞口１４や第３始動入賞口１７に始動入賞して第２特別図柄の変動表示が実行される場合には、第１始動入賞口１３に始動入賞して第１特別図柄の変動表示が実行される場合と比較して、有利な「１５Ｒ確変大当り」となる割合が高い。

【０３８０】

なお、この実施の形態では、第２特別図柄の変動表示が実行される場合の方が第１特別図柄の変動表示が実行される場合と比較し有利となる場合を示したが、例えば、大当り種別判定テーブル１３１ａと大当り種別判定テーブル１３１ｂとの割り振りを同じにして、第１特別図柄の変動表示を行う場合と第２特別図柄の変動表示を行う場合とで有利度が同じになるように構成してもよい。

【０３８１】

この実施の形態では、図５５（Ｄ），（Ｅ）に示すように、大当り種別として、「１５Ｒ確変大当り」、「７Ｒ確変大当り」および「突然確変大当り」がある。なお、この実施の形態では、大当り遊技において実行されるラウンド数が１５ラウンド、７ラウンドおよび２ラウンドの３種類である場合を示しているが、大当り遊技において実行されるラウンド数は、この実施の形態で示したものに限られない。例えば、１０ラウンドの大当り遊技に制御する１０Ｒ確変大当りや、５ラウンドの大当り遊技に制御する５Ｒ確変大当りが設けられていてもよい。また、この実施の形態では、大当り種別が「１５Ｒ確変大当り」、「７Ｒ確変大当り」および「突然確変大当り」の３種類である場合を示しているが、３種類にかぎらず、例えば、４種類以上の大当り種別を設けるようにしてもよい。また、逆に、大当り種別が３種類よりも少なくてもよく、例えば、大当り種別として２種類のみ設けられていてもよい。また、この実施の形態では、大当り種別として、大当り遊技後に確変状態に移行される確変大当りのみが設けられているが、大当り遊技後に確変状態に移行されずに通常状態に移行される通常大当りを設けるようにしてもよい。

【０３８２】

「１５Ｒ確変大当り」とは、１５ラウンドの大当り遊技状態に制御し、その大当り遊技状態の終了後に確変状態に移行させる大当りである（この実施の形態では、確変状態に移行されるとともに時短状態にも移行される。後述するステップＳ１６６参照）。そして、確変状態に移行した後、変動表示を所定回数（この実施の形態では７０回）終了すると確変状態（および時短状態）が終了する（ステップＳ１６７，Ｓ１４１～Ｓ１４４参照）。なお、変動表示を所定回数終了する前であっても、次の大当りが発生した場合にも、確変状態（および時短状態）を終了する（ステップＳ１３２参照）。

【０３８３】

また、「７Ｒ確変大当り」とは、７ラウンドの大当り遊技状態に制御し、その大当り遊技状態の終了後に確変状態に移行させる大当りである（この実施の形態では、確変状態に移行されるとともに時短状態にも移行される。後述するステップＳ１６６参照）。そして、確変状態に移行した後、変動表示を所定回数（この実施の形態では７０回）終了すると確変状態（および時短状態）が終了する（ステップＳ１６７，Ｓ１４１～Ｓ１４４参照）。なお、変動表示を所定回数終了する前であっても、次の大当りが発生した場合にも、確変状態（および時短状態）を終了する（ステップＳ１３２参照）。

【０３８４】

また、「突然確変大当り」とは、「１５Ｒ確変大当り」や「７Ｒ確変大当り」と比較して大入賞口の開放回数が少ない回数（この実施の形態では０．１秒間の開放を２回）まで許容される大当りである。すなわち、「突然確変大当り」となった場合には、２ラウンドの大当り遊技状態に制御される。また、「１５Ｒ確変大当り」や「７Ｒ確変大当り」では、１ラウンドあたりの大入賞口の開放時間が２９秒と長いのに対して、「突然確変大当り」では１ラウンドあたりの大入賞口の開放時間が０．１秒と極めて短く、大当り遊技中に大入賞口に遊技球が入賞することは殆ど期待できない。そして、この実施の形態では、その突然確変大当り遊技状態の終了後に確変状態に移行される（この実施の形態では、確変

状態に移行されるとともに時短状態にも移行される。後述するステップＳ１６６参照）。そして、確変状態に移行した後、変動表示を所定回数（この実施の形態では７０回）終了すると確変状態（および時短状態）が終了する（ステップＳ１６７，Ｓ１４１～Ｓ１４４参照）。なお、変動表示を所定回数終了する前であっても、次の大当りが発生した場合にも、確変状態（および時短状態）を終了する（ステップＳ１３２参照）。

【０３８５】

なお、前述したように、この実施の形態では、「小当り」となった場合にも、大入賞口の開放が０．１秒間ずつ２回行われ、「突然確変大当り」による大当り遊技状態と同様の制御が行われる。そして、「小当り」となった場合には、大入賞口の２回の開放が終了した後、遊技状態は変化せず、「小当り」となる前の遊技状態が維持される。そのようにすることによって、「突然確変大当り」であるか「小当り」であるかを認識できないようにし、遊技の興趣を向上させている。なお、この実施の形態のように大当り種別が全て確変大当りであるように構成する場合、小当りを設けなくてもよい。また、この実施の形態のように大当り種別が全て確変大当りである場合に小当りを設けるように構成する場合には、確変状態（高確率状態）に移行されるのみで時短状態（高ベース状態）を伴わない突然確変大当りを設けるようにすること（大入賞口の開放パターンも突然確変大当りと小当りの場合とで同じにすること）が好ましい。なお、突然確変大当りでの大入賞口の開放態様と小当りでの大入賞口の開放態様とが類似の開放態様や全く異なる開放態様であってもよく、少なくとも後述の第２特定遊技状態として１５ラウンドの大当りでの大入賞口の開放態様での遊技者にとっての有利度よりも低ければよい（例えば、開放時間が短い及び／又は開放回数が少ない開放態様とすればよい）。

【０３８６】

大当り種別判定テーブル１３１ａ，１３１ｂには、ランダム１の値と比較される数値であって、「１５Ｒ確変大当り」、「７Ｒ確変大当り」、「突然確変大当り」のそれぞれに対応した判定値（大当り種別判定値）が設定されている。ＣＰＵ５６は、ランダム１の値が大当り種別判定値のいずれかに一致した場合に、大当りの種別を、一致した大当り種別判定値に対応する種別に決定する。

【０３８７】

図５６および図５７は、主基板３１に搭載される遊技制御用マイクロコンピュータ５６０（具体的には、ＣＰＵ５６）が実行する特別図柄プロセス処理（ステップＳ２６）のプログラムの一例を示すフローチャートである。上述したように、特別図柄プロセス処理では第１特別図柄表示器８ａまたは第２特別図柄表示器８ｂおよび大入賞口を制御するための処理が実行される。特別図柄プロセス処理において、ＣＰＵ５６は、第１始動入賞口１３に遊技球が入賞したことを検出するための第１始動口スイッチ１３ａがオンしていたら、すなわち、第１始動入賞口１３への始動入賞が発生していたら、第１始動口スイッチ通過処理を実行する（ステップＳ３１１，Ｓ３１２）。また、ＣＰＵ５６は、第２始動入賞口１４に遊技球が入賞したことを検出するための第２始動口スイッチ１４ａがオンしていたら、すなわち第２始動入賞口１４への始動入賞が発生していたら、第２始動口スイッチ通過処理を実行する（ステップＳ３１３，Ｓ３１４）。そして、ステップＳ３００～Ｓ３１０のうちのいずれかの処理を行う。第１始動入賞口スイッチ１３ａまたは第２始動口スイッチ１４ａがオンしていなければ、内部状態に応じて、ステップＳ３００～Ｓ３１０のうちのいずれかの処理を行う。

【０３８８】

ステップＳ３００～Ｓ３１０の処理は、以下のような処理である。

【０３８９】

特別図柄通常処理（ステップＳ３００）：特別図柄プロセスフラグの値が０であるときに実行される。遊技制御用マイクロコンピュータ５６０は、特別図柄の可変表示が開始できる状態になると、保留記憶数バッファに記憶される数値データの記憶数（合算保留記憶数）を確認する。保留記憶数バッファに記憶される数値データの記憶数は合算保留記憶数カウンタのカウント値により確認できる。また、合算保留記憶数カウンタのカウント値が

０でなければ、大当り判定用乱数（ランダムＲ）を用いた抽選処理を実行することにより、第１特別図柄または第２特別図柄の可変表示の表示結果を大当り（または小当り）とするか否かを決定する。大当りとする場合には大当りフラグをセットする。また小当りとする場合には小当りフラグをセットする。そして、内部状態（特別図柄プロセスフラグ）をステップＳ３０１に応じた値（この例では１）に更新する。なお、大当りフラグは、大当り遊技が終了するときにリセットされる。また、小当りフラグは、小当り遊技が終了するときにリセットされる。

【０３９０】

変動パターン設定処理（ステップＳ３０１）：特別図柄プロセスフラグの値が１であるときに実行される。また、変動パターンを決定し、その変動パターンにおける変動時間（可変表示時間：可変表示を開始してから表示結果を導出表示（停止表示）するまでの時間）を特別図柄の可変表示の変動時間とすることに決定する。また、特別図柄の変動時間を計測する変動時間タイマをスタートさせる。そして、内部状態（特別図柄プロセスフラグ）をステップＳ３０２に対応した値（この例では２）に更新する。

【０３９１】

表示結果指定コマンド送信処理（ステップＳ３０２）：特別図柄プロセスフラグの値が２であるときに実行される。演出制御用マイクロコンピュータ１００に、表示結果指定コマンドを送信する制御を行う。そして、内部状態（特別図柄プロセスフラグ）をステップＳ３０３に対応した値（この例では３）に更新する。

【０３９２】

特別図柄変動中処理（ステップＳ３０３）：特別図柄プロセスフラグの値が３であるときに実行される。変動パターン設定処理で選択された変動パターンの変動時間が経過（ステップＳ３０１でセットされる変動時間タイマがタイムアウトすなわち変動時間タイマの値が０になる）すると、演出制御用マイクロコンピュータ１００に、図柄確定指定コマンドを送信する制御を行い、内部状態（特別図柄プロセスフラグ）をステップＳ３０４に対応した値（この例では４）に更新する。なお、演出制御用マイクロコンピュータ１００は、遊技制御用マイクロコンピュータ５６０が送信する図柄確定指定コマンドを受信すると演出表示装置９において第４図柄が停止されるように制御する。

【０３９３】

特別図柄停止処理（ステップＳ３０４）：特別図柄プロセスフラグの値が４であるときに実行される。大当りフラグがセットされている場合に、内部状態（特別図柄プロセスフラグ）をステップＳ３０５に対応した値（この例では５）に更新する。また、小当りフラグがセットされている場合には、内部状態（特別図柄プロセスフラグ）をステップＳ３０８に対応した値（この例では８）に更新する。大当りフラグおよび小当りフラグのいずれもセットされていない場合には、内部状態（特別図柄プロセスフラグ）をステップＳ３００に対応した値（この例では０）に更新する。なお、この実施の形態では、特別図柄プロセスフラグの値が４となったことにもとづいて、後述するように、特別図柄表示制御処理において特別図柄の停止図柄を停止表示するための特別図柄表示制御データが特別図柄表示制御データ設定用の出力バッファに設定され、ステップＳ２２の表示制御処理において出力バッファの設定内容に応じて実際に特別図柄の停止図柄が停止表示される。

【０３９４】

大入賞口開放前処理（ステップＳ３０５）：特別図柄プロセスフラグの値が５であるときに実行される。大入賞口開放前処理では、大入賞口を開放する制御を行う。具体的には、カウンタ（例えば、大入賞口に入った遊技球数をカウントするカウンタ）などを初期化するとともに、ソレノイド２１を駆動して大入賞口を開放状態にする。また、タイマによって大入賞口開放中処理の実行時間を設定し、内部状態（特別図柄プロセスフラグ）をステップＳ３０６に対応した値（この例では６）に更新する。なお、大入賞口開放前処理は各ラウンド毎に実行されるが、第１ラウンドを開始する場合には、大入賞口開放前処理は大当り遊技を開始する処理でもある。

【０３９５】

大入賞口開放中処理（ステップＳ３０６）：特別図柄プロセスフラグの値が６であるときに実行される。大当り遊技状態中のラウンド表示の演出制御コマンドを演出制御用マイクロコンピュータ１００に送信する制御や大入賞口の閉成条件の成立を確認する処理等を行う。大入賞口の閉成条件が成立し、かつ、まだ残りラウンドがある場合には、内部状態（特別図柄プロセスフラグ）をステップＳ３０５に対応した値（この例では５）に更新する。また、全てのラウンドを終えた場合には、内部状態（特別図柄プロセスフラグ）をステップＳ３０７に対応した値（この例では７）に更新する。

【０３９６】

大当り終了処理（ステップＳ３０７）：特別図柄プロセスフラグの値が７であるときに実行される。大当り遊技状態が終了したことを遊技者に報知する表示制御を演出制御用マイクロコンピュータ１００に行わせるための制御を行う。また、遊技状態を示すフラグ（例えば、確変フラグや時短フラグ）をセットする処理を行う。そして、内部状態（特別図柄プロセスフラグ）をステップＳ３００に対応した値（この例では０）に更新する。

【０３９７】

小当り開放前処理（ステップＳ３０８）：特別図柄プロセスフラグの値が８であるときに実行される。小当り開放前処理では、大入賞口を開放する制御を行う。具体的には、カウンタ（例えば、大入賞口に入った遊技球数をカウントするカウンタ）などを初期化するとともに、ソレノイド２１を駆動して大入賞口を開放状態にする。また、タイマによって大入賞口開放中処理の実行時間を設定し、内部状態（特別図柄プロセスフラグ）をステップＳ３０９に対応した値（この例では９）に更新する。なお、小当り開放前処理は各ラウンド毎に実行されるが、第１ラウンドを開始する場合には、小当り開放前処理は小当り遊技を開始する処理でもある。

【０３９８】

小当り開放中処理（ステップＳ３０９）：特別図柄プロセスフラグの値が９であるときに実行される。大入賞口の閉成条件の成立を確認する処理等を行う。大入賞口の閉成条件が成立し、かつ、まだ残りラウンドがある場合には、内部状態（特別図柄プロセスフラグ）をステップＳ３０８に対応した値（この例では８）に更新する。また、全てのラウンドを終えた場合には、内部状態（特別図柄プロセスフラグ）をステップＳ３１０に対応した値（この例では１０（１０進数））に更新する。

【０３９９】

小当り終了処理（ステップＳ３１０）：特別図柄プロセスフラグの値が１０であるときに実行される。小当り遊技状態が終了したことを遊技者に報知する表示制御を演出制御用マイクロコンピュータ１００に行わせるための制御を行う。そして、内部状態（特別図柄プロセスフラグ）をステップＳ３００に対応した値（この例では０）に更新する。

【０４００】

図５８は、ステップＳ３１２，Ｓ３１４の始動口スイッチ通過処理を示すフローチャートである。このうち、図５８（Ａ）は、ステップＳ３１２の第１始動口スイッチ通過処理を示すフローチャートである。また、図５８（Ｂ）は、ステップＳ３１４の第２始動口スイッチ通過処理を示すフローチャートである。

【０４０１】

まず、図５８（Ａ）を参照して第１始動口スイッチ通過処理について説明する。第１始動口スイッチ１３ａがオン状態の場合に実行される第１始動口スイッチ通過処理において、ＣＰＵ５６は、第１保留記憶数が上限値に達しているか否か（具体的には、第１保留記憶数をカウントするための第１保留記憶数カウンタの値が４でるか否か）を確認する（ステップＳ２０１Ａ）。

【０４０２】

第１保留記憶数が上限値に達していなければ、ＣＰＵ５６は、第１保留記憶数カウンタの値を１増やす（ステップＳ２０２Ａ）とともに、合算保留記憶数をカウントするための合算保留記憶数カウンタの値を１増やす（ステップＳ２０３Ａ）。

【０４０３】

次いで、ＣＰＵ５６は、ソフトウェア乱数（大当り種別判定用乱数（ランダム１）、変動パターン種別判定用乱数（ランダム２）および変動パターン判定用乱数（ランダム３））を生成するための各カウンタから値を抽出する（ステップＳ２０４Ａ）。また、ＣＰＵ５６は、チャネル０の１６ビット乱数回路５０８ｂが用いるＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）から、大当り判定用乱数（ランダムＲ）としての数値データを抽出する（ステップＳ２０５Ａ）。

【０４０４】

なお、既に、図２５で説明したように、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット２－０の設定内容によりいずれの端子からの信号（ラッチ信号）にもとづいて、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）に乱数値をラッチさせるかが設定されている。また、この実施の形態では、その設定された端子には、第１始動口スイッチ１３ａからの検出信号がラッチ信号として入力されているものとし、第１始動入賞口１３への始動入賞が発生したタイミングでＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）に乱数値をラッチできるように構成されている。

【０４０５】

そして、ＣＰＵ５６は、抽出したそれらのソフトウェア乱数および大当り判定用乱数（ランダムＲ）を、第１保留記憶バッファ（図５９参照）における保存領域に格納する処理を実行する（ステップＳ２０６Ａ）。なお、変動パターン判定用乱数（ランダム３）を第１始動口スイッチ通過処理（始動入賞時）において抽出して保存領域にあらかじめ格納しておくのではなく、第１特別図柄の変動開始時に抽出するようにしてもよい。例えば、遊技制御用マイクロコンピュータ５６０は、後述する変動パターン設定処理において、変動パターン判定用乱数（ランダム３）を生成するための変動パターン判定用乱数カウンタから値を直接抽出するようにしてもよい。

【０４０６】

図５９は、保留記憶に対応する乱数等を保存する領域（保留記憶バッファ）の構成例を示す説明図である。図５９に示すように、第１保留記憶バッファには、第１保留記憶数の上限値（この例では４）に対応した保存領域が確保されている。また、第２保留記憶バッファには、第２保留記憶数の上限値（この例では４）に対応した保存領域が確保されている。この実施の形態では、第１保留記憶バッファおよび第２保留記憶バッファには、ハードウェア乱数であるランダムＲ（大当り判定用乱数）や、ソフトウェア乱数である大当り種別判定用乱数（ランダム１）、変動パターン種別判定用乱数（ランダム２）および変動パターン判定用乱数（ランダム３）が記憶される。なお、第１保留記憶バッファおよび第２保留記憶バッファは、ＲＡＭ５５に形成されている。

【０４０７】

そして、ＣＰＵ５６は、第１保留記憶数が１増加したことを指定する第１保留記憶数加算指定コマンドを演出制御用マイクロコンピュータ１００に送信する制御を行う（ステップＳ２０７Ａ）。

【０４０８】

第１保留記憶数が上限値に達していれば（ステップＳ２０１ＡのＹ）、ＣＰＵ５６は、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）から数値データを抽出し（ステップＳ２０８Ａ）、抽出した数値データ（乱数）を格納することなく、第１始動口スイッチ通過処理を終了する。すなわち、この実施の形態では、図２５に示すプログラム管理エリアにおけるＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット３が”０”に設定され、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）から値を読み込まないと次の値をラッチできないように設定されているものとする。そのため、第１保留記憶数が上限値に達している場合であっても、ＣＰＵ５６は、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）から数値データを抽出する処理のみを行い（値の格納までは行わない）、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）が次の値をラッチできるようにしている。

【０４０９】

なお、ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）のビット３を”１”に設定し、ＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）から値を読み込まなくても次の値をラッチできるように設定してもよい。そのようにすれば、ステップＳ２０８Ａの処理は不要となる。

【０４１０】

次に、図５８（Ｂ）を参照して第２始動口スイッチ通過処理について説明する。第２始動口スイッチ１４ａがオン状態の場合に実行される第２始動口スイッチ通過処理において、ＣＰＵ５６は、第２保留記憶数が上限値に達しているか否か（具体的には、第２保留記憶数をカウントするための第２保留記憶数カウンタの値が４でるか否か）を確認する（ステップＳ２０１Ｂ）。

【０４１１】

第２保留記憶数が上限値に達していなければ、ＣＰＵ５６は、第２保留記憶数カウンタの値を１増やす（ステップＳ２０２Ｂ）とともに、合算保留記憶数をカウントするための合算保留記憶数カウンタの値を１増やす（ステップＳ２０３Ｂ）。

【０４１２】

次いで、ＣＰＵ５６は、ソフトウェア乱数（大当り種別判定用乱数（ランダム１）、変動パターン種別判定用乱数（ランダム２）および変動パターン判定用乱数（ランダム３））を生成するための各カウンタから値を抽出する（ステップＳ２０４Ｂ）。また、ＣＰＵ５６は、チャネル１の１６ビット乱数回路５０８ｂが用いるＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）から、大当り判定用乱数（ランダムＲ）としての数値データを抽出する（ステップＳ２０５Ｂ）。

【０４１３】

なお、既に、図２７で説明したように、ＲＬ１ハードラッチ選択レジスタ（ＲＬ１ＬＳ）のビット２－０の設定内容によりいずれの端子からの信号（ラッチ信号）にもとづいて、ＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）に乱数値をラッチさせるかが設定されている。また、この実施の形態では、その設定された端子には、第２始動口スイッチ１４ａからの検出信号がラッチ信号として入力されているものとし、第２始動入賞口１４への始動入賞が発生したタイミングでＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）に乱数値をラッチできるように構成されている。

【０４１４】

そして、ＣＰＵ５６は、抽出したそれらのソフトウェア乱数および大当り判定用乱数（ランダムＲ）を、第２保留記憶バッファ（図５９参照）における保存領域に格納する処理を実行する（ステップＳ２０６Ｂ）。なお、変動パターン判定用乱数（ランダム３）を第２始動口スイッチ通過処理（始動入賞時）において抽出して保存領域にあらかじめ格納しておくのではなく、第２特別図柄の変動開始時に抽出するようにしてもよい。例えば、遊技制御用マイクロコンピュータ５６０は、後述する変動パターン設定処理において、変動パターン判定用乱数（ランダム３）を生成するための変動パターン判定用乱数カウンタから値を直接抽出するようにしてもよい。

【０４１５】

そして、ＣＰＵ５６は、第２保留記憶数が１増加したことを指定する第２保留記憶数加算指定コマンドを演出制御用マイクロコンピュータ１００に送信する制御を行う（ステップＳ２０７Ｂ）。

【０４１６】

第２保留記憶数が上限値に達していれば（ステップＳ２０１ＢのＹ）、ＣＰＵ５６は、ＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）から数値データを抽出し（ステップＳ２０８Ｂ）、抽出した数値データ（乱数）を格納することなく、第２始動口スイッチ通過処理を終了する。すなわち、この実施の形態では、プログラム管理エリアにおけるＲＬ１ハードラッチ選択レジスタ（ＲＬ１ＬＳ）のビット３が”０”に設定され（図２７でｎ＝１とした場合に相当する）、ＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）から値を読み込まないと次の値をラッチできないように設定されているものとする。その

ため、第２保留記憶数が上限値に達している場合であっても、ＣＰＵ５６は、ＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）から数値データを抽出する処理のみを行い（値の格納までは行わない）、ＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）が次の値をラッチできるようにしている。

【０４１７】

なお、ＲＬ１ハードラッチ選択レジスタ０（ＲＬ１ＬＳ）のビット３を”１”に設定し、ＲＬ１ハードラッチ乱数値レジスタ０（ＲＬ１ＨＶ０）から値を読み込まなくても次の値をラッチできるように設定してもよい。そのようにすれば、ステップＳ２０８Ｂの処理は不要となる。

【０４１８】

また、この実施の形態では、ステップＳ２０５Ａ，Ｓ２０５Ｂの処理が実行されることによって、第１特別図柄の変動表示を実行する場合と第２特別図柄の変動表示を実行する場合とで異なる乱数値レジスタから乱数値を抽出して格納するようにしている。そのようにすることにより、例えば、乱数更新のスタート値を異ならせたり、乱数列の変更の設定を異ならせたり、乱数最大値の設定を異ならせることによって、第１特別図柄の変動表示を実行する場合と第２特別図柄の変動表示を実行する場合とで乱数値が同期しにくくすることができ、所定の乱数更新タイミングを狙って不正に大当りを発生させるなどの行為をしにくくしている。

【０４１９】

なお、この実施の形態では、１６ビット乱数回路５０８ｂの異なるチャネル（本例では、チャネル０とチャネル１）から乱数値を抽出することにより、第１特別図柄の変動表示を実行する場合と第２特別図柄の変動表示を実行する場合とで乱数を抽出する乱数値レジスタを異ならせる場合を示したが、この実施の形態で示した態様にかぎられない。例えば、１６ビット乱数回路５０８ｂの同じチャネルであっても、その同じチャネルで用いる異なるハードラッチ乱数値レジスタから（例えば、同じチャネル０のＲＬ０ハードラッチ乱数値レジスタ０（ＲＬ０ＨＶ０）とＲＬ０ハードラッチ乱数値レジスタ１（ＲＬ０ＨＶ１）とから）乱数値を抽出することにより、第１特別図柄の変動表示を実行する場合と第２特別図柄の変動表示を実行する場合とで乱数を抽出する乱数値レジスタを異ならせてもよい。

【０４２０】

また、この実施の形態では、ハードラッチ乱数値レジスタから乱数値を抽出する場合を示したが、例えば、ソフトラッチ乱数値レジスタから乱数値を抽出するようにしてもよい。この場合であっても、１６ビット乱数回路５０８ｂの異なるチャネルから乱数値を抽出することにより、または同じチャネルであっても異なるソフトラッチ乱数値レジスタから乱数値を抽出することにより、第１特別図柄の変動表示を実行する場合と第２特別図柄の変動表示を実行する場合とで乱数を抽出する乱数値レジスタを異ならせるようにすればよい。

【０４２１】

また、この実施の形態では、１６ビット乱数回路５０８ｂから乱数値を抽出する場合を示したが、例えば、８ビット乱数回路５０８ａから乱数値を抽出するようにしてもよい。この場合であっても、８ビット乱数回路５０８ａの異なるチャネルから乱数値を抽出することにより、または同じチャネルであっても異なるハードラッチ乱数値レジスタやソフトラッチ乱数値レジスタから乱数値を抽出することにより、第１特別図柄の変動表示を実行する場合と第２特別図柄の変動表示を実行する場合とで乱数を抽出する乱数値レジスタを異ならせるようにすればよい。

【０４２２】

なお、この実施の形態では、既に説明したように、遊技制御用マイクロコンピュータ５６０は、遊技機への電源投入時にステップＳ１００１，Ｓ１０１１を実行して乱数回路５０８ａ，５０８ｂに関する設定をハードウェア的に行い、その後に、ユーザプログラムの実行中に第１始動口スイッチ通過処理（ステップＳ３１２参照）や第２始動口スイッチ通

過処理（ステップＳ３１４参照）においてステップＳ２０５Ａ，Ｓ２０５Ｂの乱数抽出の処理を実行する。従って、この実施の形態では、乱数回路から数値データ（乱数値）を抽出するタイミングよりも前に、乱数回路の監視に関する設定が行われるように構成されている。

【０４２３】

次に、特別図柄プロセス処理における特別図柄停止処理について説明する。図６０は、特別図柄プロセス処理における特別図柄停止処理（ステップＳ３０４）を示すフローチャートである。特別図柄停止処理において、ＣＰＵ５６は、大当りフラグがセットされているか否かを確認する（ステップＳ１３１）。なお、大当りフラグは、特別図柄通常処理において大当りと判定されるとセットされる。大当りフラグがセットされている場合には、ＣＰＵ５６は、セットされていれば、確変状態であることを示す確変フラグ、および確変状態における特別図柄の変動可能回数を示す確変回数カウンタをリセットする（ステップＳ１３２）。また、ＣＰＵ５６は、演出制御用マイクロコンピュータ１００に大当り開始指定コマンドを送信する制御を行う（ステップＳ１３４）。具体的には、大当りの種別が「１５Ｒ確変大当り」または「７Ｒ確変大当り」である場合には、大当り開始指定コマンドを送信する。また、大当りの種別が突然確変大当りである場合には、小当り／突然確変大当り開始指定コマンドを送信する。なお、大当りの種別が「１５Ｒ確変大当り」、「７Ｒ確変大当り」または「突然確変大当り」のいずれであるかは、ＲＡＭ５５に記憶されている大当り種別を示すデータ（大当り種別バッファに記憶されているデータ）にもとづいて判定される。

【０４２４】

また、大当り表示時間タイマに大当り表示時間（大当りが発生したことを、例えば、演出表示装置９において報知する時間）に相当する値を設定する（ステップＳ１３５）。また、大入賞口開放回数カウンタに開放回数（例えば、「１５Ｒ確変大当り」の場合には１５回。「７Ｒ確変大当り」の場合には７回。「突然確変大当り」の場合には２回。）をセットする（ステップＳ１３６）。また、大当り遊技における１ラウンドあたりのラウンド時間もセットされる。具体的には、突然確変大当りの場合には、ラウンド時間として０．１秒がセットされ、１５Ｒ確変大当りや７Ｒ確変大当りの場合には、ラウンド時間として２９秒がセットされる。そして、特別図柄プロセスフラグの値を大入賞口開放前処理（ステップＳ３０５）に対応した値に更新する（ステップＳ１３７）。

【０４２５】

また、ステップＳ１３１で大当りフラグがセットされていなければ、ＣＰＵ５６は、確変状態における特別図柄の変動可能回数を示す確変回数カウンタの値が０となっているか否かを確認する（ステップＳ１４１）。確変回数カウンタの値が０でなければ、ＣＰＵ５６は、確変回数カウンタの値を－１する（ステップＳ１４２）。そして、ＣＰＵ５６は、減算後の確変回数カウンタの値が０になった場合には（ステップＳ１４３）、確変フラグをリセットする（ステップＳ１４４）。

【０４２６】

次いで、ＣＰＵ５６は、小当りフラグがセットされているか否かを確認する（ステップＳ１４５）。なお、小当りフラグは、特別図柄通常処理において小当りと判定されるとセットされる。小当りフラグがセットされていれば、ＣＰＵ５６は、演出制御用マイクロコンピュータ１００に小当り／突然確変大当り開始指定コマンドを送信する（ステップＳ１４６）。また、小当り表示時間タイマに小当り表示時間（小当りが発生したことを、例えば、演出表示装置９において報知する時間）に相当する値を設定する（ステップＳ１４７）。また、大入賞口開放回数カウンタに開放回数（例えば２回）をセットする（ステップＳ１４８）。また、小当り遊技における大入賞口の１回あたりの開放時間もセットされる。具体的には、突然確変大当りのラウンド時間と同じ０．１秒が、小当り遊技における大入賞口の１回あたりの開放時間としてセットされる。そして、特別図柄プロセスフラグの値を小当り開始前処理（ステップＳ３０８）に対応した値に更新する（ステップＳ１４９）。

【０４２７】
小当りフラグもセットされていなければ（ステップＳ１４５のＮ）、ＣＰＵ５６は、特別図柄プロセスフラグの値を特別図柄通常処理（ステップＳ３００）に対応した値に更新する（ステップＳ１５０）。
【０４２８】
次に、特別図柄プロセス処理における大当り終了処理について説明する。図６１は、特別図柄プロセス処理における大当り終了処理（ステップＳ３０７）を示すフローチャートである。大当り終了処理において、ＣＰＵ５６は、大当り終了表示タイマが設定されているか否か確認し（ステップＳ１６０）、大当り終了表示タイマが設定されている場合には、ステップＳ１６４に移行する。大当り終了表示タイマが設定されていない場合には、大当りフラグをリセットし（ステップＳ１６１）、大当り終了指定コマンドを送信する制御を行う（ステップＳ１６２）。ここで、「１５Ｒ確変大当り」または「７Ｒ確変大当り」であった場合には大当り終了指定コマンドを送信し、「突然確変大当り」であった場合には小当り／突然確変大当り終了指定コマンドを送信する。そして、大当り終了表示タイマに、演出表示装置９において大当り終了表示が行われている時間（大当り終了表示時間）に対応する表示時間に相当する値を設定し（ステップＳ１６３）、処理を終了する。
【０４２９】
ステップＳ１６４では、大当り終了表示タイマの値を１減算する（ステップＳ１６４）。そして、ＣＰＵ５６は、大当り終了表示タイマの値が０になっているか否か、すなわち大当り終了表示時間が経過したか否か確認する（ステップＳ１６５）。経過していなければ処理を終了する。
【０４３０】
大当り終了表示時間を経過していれば（ステップＳ１６５のＹ）、ＣＰＵ５６は、確変フラグをセットして確変状態に移行させる（ステップＳ１６６）。なお、この実施の形態では、確変フラグがセットされることによって、確変状態に制御されるとともに時短状態（高ベース状態）にも制御される。また、ＣＰＵ５６は、確変回数カウンタに所定回数（本例では、７０回）をセットする（ステップＳ１６７）。
【０４３１】
なお、時短状態であることを示す時短フラグもセットするようにし、確変状態の管理と時短状態の管理とを別々に行うように構成してもよい。また、この実施の形態では、大当り遊技を終了して確変状態および時短状態に移行した後、同じ回数（本例では、７０回）の変動表示を終了すると確変状態および時短状態の両方を終了する場合を示したが、確変状態の終了回数と時短状態の終了回数とを異ならせてもよい。
【０４３２】
そして、ＣＰＵ５６は、特別図柄プロセスフラグの値を特別図柄通常処理（ステップＳ３００）に対応した値に更新する（ステップＳ１７０）。
【０４３３】
次に、遊技制御用マイクロコンピュータ５６０（具体的には、ＣＰＵ５６）が実行する普通図柄プロセス処理（ステップＳ２８）について説明する。図６２は、普通図柄プロセス処理の一例を示すフローチャートである。普通図柄プロセス処理では、ＣＰＵ５６は、ゲート３２を遊技球が通過してゲートスイッチ３２ａがオン状態となったことを検出すると（ステップＳ５１１）、ゲートスイッチ通過処理（ステップＳ５１２）を実行する。そして、ＣＰＵ５６は、普通図柄プロセスフラグの値に応じてステップＳ５００～Ｓ５０３に示された処理のうちのいずれかの処理を実行する。
【０４３４】
ゲートスイッチ通過処理（ステップＳ５１２）：ＣＰＵ５６は、ゲート通過記憶カウンタのカウント値（ゲート通過記憶数）が最大値（この例では「４」）に達しているか否か確認する。最大値に達していなければ、ゲート通過記憶カウンタのカウント値を＋１する。なお、ゲート通過記憶カウンタの値に応じて普通図柄保留記憶表示器４１のＬＥＤが点灯される。そして、ＣＰＵ５６は、普通図柄当り判定用乱数（ランダム４）の値を抽出し

、ゲート通過記憶数の値に対応した保存領域（普通図柄判定用バッファ）に格納する処理を行う。

【０４３５】

普通図柄通常処理（ステップＳ５００）：ＣＰＵ５６は、普通図柄の変動を開始することができる状態（例えば普通図柄プロセスフラグの値がステップＳ５００を示す値となっている場合、具体的には、普通図柄表示器１０において普通図柄の変動表示がなされておらず、かつ、普通図柄表示器１０に当り図柄が導出表示されたことにもとづく可変入賞球装置１５の開閉動作中でもない場合）には、ゲート通過記憶数の値を確認する。具体的には、ゲート通過記憶数カウンタのカウント値を確認する。ゲート通過記憶数が０でなければ、当りとするか否か（普通図柄の停止図柄を当り図柄とするか否か）を決定する。そして、普通図柄プロセスタイマに普通図柄の変動時間をセットし、タイマをスタートさせる。そして、普通図柄プロセスフラグの値を普通図柄変動処理（ステップＳ５０１）を示す値（具体的には「１」）に更新する。

【０４３６】

普通図柄変動処理（ステップＳ５０１）：ＣＰＵ５６は、普通図柄プロセスタイマがタイムアウトしたか否か確認し、タイムアウトしていたら、普通図柄表示器１０における普通図柄の変動を停止し、普通図柄プロセスタイマに普通図柄停止図柄表示時間をセットし、タイマをスタートさせる。そして、普通図柄プロセスフラグの値を普通図柄停止処理（ステップＳ５０２）を示す値（具体的には「２」）に更新する。

【０４３７】

普通図柄停止処理（ステップＳ５０２）：ＣＰＵ５６は、普通図柄プロセスタイマがタイムアウトしたか否かを確認し、タイムアウトしていたら、普通図柄の停止図柄が当り図柄であるかどうかを確認する。当り図柄でなければ（はずれ図柄であれば）、普通図柄プロセスフラグの値を普通図柄通常処理（ステップＳ５００）を示す値（具体的には「０」）に更新する。一方、普通図柄の停止図柄が当り図柄であれば、普通図柄プロセスタイマに普通電動役物作動時間をセットし、タイマをスタートさせる。また、現在の遊技状態が確変状態であるか否かを確認し、確変状態（すなわち、高ベース状態）であれば、確変状態（高ベース状態）のときの普通電動役物（可変入賞球装置１５）の開放パターンを選択し、確変状態でなければ（すなわち、低ベース状態であれば）、低ベース状態のときの普通電動役物（可変入賞球装置１５）の開放パターンを選択し、選択した開放パターンを設定する。そして、普通図柄プロセスフラグの値を普通電動役物作動処理（ステップＳ５０３）を示す値（具体的には「３」）に更新する。

【０４３８】

普通電動役物作動処理（ステップＳ５０３）：ＣＰＵ５６は、普通図柄プロセスタイマがタイムアウトしていないことを条件に、設定された開放パターンで普通電動役物の開放を行う（可変入賞球装置１５の開閉動作を実行する）普通電動役物開放パターン処理を実行する。なお、この実施の形態では、確変状態（すなわち、高ベース状態）では、低ベース状態である場合と比較して可変入賞球装置１５の開放期間が長いものとする。そして、普通図柄プロセスタイマがタイムアウトすると、普通図柄プロセスフラグの値を普通図柄通常処理（ステップＳ５００）を示す値（具体的には「０」）に更新する。

【０４３９】

次に、主基板３１に搭載される遊技制御用マイクロコンピュータ５６０（具体的には、ＣＰＵ５６）が実行する特別図柄表示制御処理（ステップＳ３３）について説明する（フローチャート等は図示せず）。特別図柄表示制御処理では、ＣＰＵ５６は、特別図柄プロセスフラグの値が３であるか否かを確認する（ステップＳ３２０１）。特別図柄プロセスフラグの値が３であれば（すなわち、特別図柄変動中処理の実行中であれば）、ＣＰＵ５６は、特別図柄変動表示用の特別図柄表示制御データを特別図柄表示制御データ設定用の出力バッファに設定または更新する処理を行う（ステップＳ３２０２）。この場合、ＣＰＵ５６は、特別図柄ポインタが示す方の特別図柄（第１特別図柄または第２特別図柄）の変動表示を行うための特別図柄表示制御データを設定または更新する。例えば、変動速度

が１コマ／０．２秒であれば、０．２秒が経過する毎に、出力バッファに設定される特別図柄表示制御データの値を＋１する。そして、その後、表示制御処理（ステップＳ２３参照）が実行され、特別図柄表示制御データ設定用の出力バッファの内容に応じて特別図柄表示器８ａ，８ｂに対して駆動信号が出力されることによって、特別図柄表示器８ａ，８ｂにおける特別図柄の変動表示が実行される。

【０４４０】

特別図柄プロセスフラグの値が３でなければ、ＣＰＵ５６は、特別図柄プロセスフラグの値が４であるか否かを確認する（ステップＳ３２０３）。特別図柄プロセスフラグの値が４であれば（すなわち、特別図柄停止処理に移行した場合には）、ＣＰＵ５６は、特別図柄通常処理で設定された特別図柄の停止図柄を停止表示するための特別図柄表示制御データを特別図柄表示制御データ設定用の出力バッファに設定する処理を行う（ステップＳ３２０４）。この場合、ＣＰＵ５６は、特別図柄ポインタが示す方の特別図柄（第１特別図柄または第２特別図柄）の停止図柄を停止表示するための特別図柄表示制御データを設定する。そして、その後、表示制御処理（ステップＳ２３参照）が実行され、特別図柄表示制御データ設定用の出力バッファの内容に応じて特別図柄表示器８ａ，８ｂに対して駆動信号が出力されることによって、特別図柄表示器８ａ，８ｂにおいて特別図柄の停止図柄が停止表示される。なお、ステップＳ３２０４の処理が実行され停止図柄表示用の特別図柄表示制御データが設定された後には、設定データの変更が行われないので、ステップＳ２３の表示制御処理では最新の特別図柄表示制御データにもとづいて最新の停止図柄を次の変動表示が開始されるまで停止表示し続けることになる。また、ステップＳ３２０１において特別図柄プロセスフラグの値が２または３のいずれかであれば（すなわち、表示結果指定コマンド送信処理または特別図柄変動中処理のいずれかであれば）、特別図柄変動表示用の特別図柄表示制御データを更新するようにしてもよい。この場合、遊技制御用マイクロコンピュータ５６０側で認識する変動時間と演出制御用マイクロコンピュータ１００側で認識する変動時間との間にズレが生じないようにするため、表示結果指定コマンド送信処理においても変動時間タイマを１減算するように構成すればよい。

【０４４１】

なお、この実施の形態では、特別図柄プロセスフラグの値に応じて特別図柄表示制御データを出力バッファに設定する場合を示したが、特別図柄プロセス処理において、特別図柄の変動開始時に開始フラグをセットするとともに、特別図柄の変動終了時に終了フラグをセットするようにしてもよい。そして、特別図柄表示制御処理（ステップＳ３３）において、ＣＰＵ５６は、開始フラグがセットされたことにもとづいて特別図柄表示制御データの値の更新を開始するようにし、終了フラグがセットされたことにもとづいて停止図柄を停止表示させるための特別図柄表示制御データをセットするようにしてもよい。

【０４４２】

次に、遊技制御用マイクロコンピュータ５６０（具体的には、ＣＰＵ５６）が実行する入賞順異常報知処理（ステップＳ２１ａ）について説明する。図６３は、ステップＳ２１ａの入賞順異常報知処理を示すフローチャートである。入賞順異常報知処理において、遊技制御用マイクロコンピュータ５６０（具体的には、ＣＰＵ５６）は、第２始動入賞口１４に対応する第２始動口監視用バッファを加算バッファにセットし（ステップＳ２５０）、第１始動入賞口１３に対応する第１始動口監視用バッファをクリアバッファにセットする（ステップＳ２５１）。

【０４４３】

第１始動口監視用バッファおよび第２始動口監視用バッファは、加算バッファおよびクリアバッファと同様に、ＲＡＭ５５に形成され、対応する始動口に入賞があると、原則的に（例外については後述する）格納している値に所定値（本例では１）が加算される。加算バッファは、セットされた始動口監視用バッファを所定値（本例では１）の加算対象とするために用いられる。また、クリアバッファは、セットされた始動口監視用バッファをクリア（本例では格納している値を０にする）対象とするために用いられる。なお、この実施の形態では、始動口監視用バッファを加算バッファやクリアバッファにセットすると

の表現を用いるが、例えば、加算バッファまたはクリアバッファに、第１始動口監視用バッファまたは第２始動口監視用バッファを特定するためのポインタがセットされることによって実現される。

【０４４４】

次いで、ＣＰＵ５６は、第１始動口スイッチがオン状態であるか否かを確認し（ステップＳ２５２）、第１始動口スイッチがオン状態である場合には、加算バッファとクリアバッファのセット内容を入れ替える（ステップＳ２５３）。すなわち、ステップＳ２５３では、ＣＰＵ５６は、第１始動口監視用バッファを加算バッファにセットし、第２始動口監視用バッファをクリアバッファにセットする。ステップＳ２５３では、例えば、値を交換する命令（例えば、ＥＸ命令）１つを実行させることで入れ替えを行うことが可能である。

【０４４５】

この実施の形態では、ＣＰＵ５６は、ステップＳ２５０，Ｓ２５１で第２始動口監視用バッファを加算バッファに、第１始動口監視用バッファをクリアバッファにセットし（以下、仮セットともいう）、第１始動口スイッチがオン状態であれば（すなわち遊技球が入賞した始動口に対応した始動口監視用バッファが加算バッファにセットされていなければ（ステップＳ２５２のＹ））、ステップＳ２５３で加算バッファとクリアバッファのセット内容を入れ替える（遊技球が入賞した始動口に対応した始動口監視用バッファが加算バッファにセットされていれば、セット内容を入れ替えない）。このように、第１始動口監視用バッファおよび第２始動口監視用バッファを加算バッファとクリアバッファとに仮セットし、遊技球が入賞した始動口に対応した始動口監視用バッファが加算バッファにセットされていなければ、加算バッファとクリアバッファのセット内容を入れ替えることによって、遊技球が入賞した始動口（例えば、第１始動入賞口１３）に対応した始動口監視用バッファを加算対象とし、他方の始動口（例えば、第２始動入賞口１４）に対応した始動口監視用バッファをクリア対象としてセットすることができる。したがって、例えば、同じ始動口に連続して入賞が発生した場合には、対応する始動口監視用バッファが連続して加算対象としてセットされ、他方の始動口に対応する始動口監視用バッファが連続してクリア対象としてセットされる。また、前回入賞した始動口とは異なる始動口に入賞した場合には、前回加算対象としてセットされた始動口監視用バッファがクリア対象としてセットされ、前回クリア対象としてセットされた始動口監視用バッファが加算対象としてセットされる。なお、加算バッファとクリアバッファのセット内容を入れ替えるとは、最終的に（ステップＳ２５４を実行する前に）セット内容（例えば、ポインタ）が入れ替わっていればよく、処理の途中で一旦ポインタを示すデータを他の領域に保存してもよい。つまり、加算バッファのセット内容とクリアバッファのセット内容とを直接入れ替えるものに限らず、加算バッファのセット内容とクリアバッフのセット内容と（またはいずれか一方のみ）を他の領域に一旦格納した後に、入れ替えるものであってもよい。

【０４４６】

なお、この実施の形態では、ステップＳ２５０，Ｓ２５１で第２始動口監視用バッファを加算バッファに、第１始動口監視用バッファをクリアバッファに仮セットしているが、これに限らず、逆に仮セットするようにしてもよい。この場合には、ステップＳ２５２において、第２始動口スイッチがオン状態であるか否かを確認し、オン状態であればステップＳ２５３に移行するようにすればよい。

【０４４７】

ステップＳ２５３で加算バッファとクリアバッファのセット内容を入れ替えると、ＣＰＵ５６は、加算バッファを１加算し、加算バッファをロードする（ステップＳ２５４）。また、ＣＰＵ５６は、クリアデータ（本例では０）をセットし、クリアバッファにストアする（ステップＳ２５５）。すなわち、ステップＳ２５４において、加算バッファにセットされた始動口監視用バッファに格納された値に所定値（本例では１）を加算し、ステップＳ２５５において、クリアバッファにセットされた始動口監視用バッファに格納された値をクリア（本例では０に）する。このようにすることによって、同じ始動口に連続して

入賞が発生した場合には、対応する始動口監視用バッファに格納された値が連続して１加算され、他方の始動口に対応する始動口監視用バッファに格納された値がクリアされる。また、前回入賞した始動口とは異なる始動口に入賞した場合には、前回加算対象として所定値（本例では１）が加算された始動口監視用バッファに格納された値がクリア（本例では０に）され、前回クリア対象としてクリアされた始動口監視用バッファに格納された値に所定値（本例では１）が加算される。以下、「加算バッファの値」という表現を用いることがあるが、具体的には、加算バッファにセットされた始動口監視用バッファに格納された値のことを示す。また、「加算バッファを１加算する」という表現を用いることがあるが、具体的には、加算バッファにセットされた始動口監視用バッファに格納された値に所定値（本例では１）加算することである。また、「加算バッファをクリアする」という表現を用いることがあるが、具体的には、加算バッファにセットされた始動口監視用バッファに格納された値をクリア（本例では０に）することである。

【０４４８】

次いで、ＣＰＵ５６は、加算バッファの値が５以上であるか否かを判定する（ステップＳ２５９）。加算バッファの値が５以上となっていれば、ＣＰＵ５６は、加算バッファをクリアし（ステップＳ２６０）、ステップＳ２６１に移行する。加算バッファの値が５以上でなければ、そのまま処理を終了する。

【０４４９】

ステップＳ２６１では、ＣＰＵ５６は、入賞順異常報知指定コマンドを演出制御用マイクロコンピュータ１００に対して送信する制御を行う（ステップＳ２６１）。また、ＣＰＵ５６は、セキュリティ信号情報タイマに所定時間（本例では、３０秒）をセットする（ステップＳ２６２）。この実施の形態では、ステップＳ２６２でセキュリティ信号情報タイマに所定時間がセットされたことにもとづいて、情報出力処理（Ｓ３０参照）が実行されることによって、セキュリティ信号が所定時間（本例では、３０秒）外部出力される。すなわち、加算バッファの値が５以上である場合には（ステップＳ２５９のＹ）、第１始動入賞口１３または第２始動入賞口１４に連続して５以上の入賞が発生したことにより入賞順異常が発生したと判定して、入賞順異常報知を行うために入賞順異常報知指定コマンドを送信する制御を行うとともに、セキュリティ信号を所定期間（本例では、３０秒）外部出力するための処理を行う。

【０４５０】

また、ステップＳ２５２で第１始動口スイッチがオン状態ではない場合には、ＣＰＵ５６は、第２始動口スイッチがオン状態であるか否かを確認する（ステップＳ２５６）。第２始動口スイッチがオン状態でなければ（すなわち、いずれの始動口にも入賞していなければ）、そのまま処理を終了する。

【０４５１】

一方、第２始動口スイッチ１４ａがオン状態であれば、ＣＰＵ５６は、確変フラグがセットされているか否かを確認する（ステップＳ２５７）。確変フラグがセットされていれば（すなわち、確変状態であるとともに時短状態（高ベース状態）であれば）、ステップＳ２５５に移行する。確変フラグがセットされていなければ、ＣＰＵ５６は、普通図柄プロセスフラグの値が３であるか否かを確認する（ステップＳ２５８）。この実施の形態では、普通図柄プロセスフラグの値が３であれば（すなわち普通図柄プロセス処理の普通電動役物作動処理が実行されている場合には）、ステップＳ２５５に移行する。

【０４５２】

ステップＳ２５７で確変フラグがセットされていた場合、またはステップＳ２５８で普通図柄プロセスフラグの値が３である場合には、ステップＳ２５５で、ＣＰＵ５６は、クリアデータ（本例では０）をセットし、クリアバッファにセットする（ステップＳ２５５）。すなわち、確変フラグがセットされているということは、時短状態（高ベース状態）に制御され、可変入賞球装置１５が開放状態となる頻度が高められているのであるから、振分装置２００の状態にかかわらず第２始動入賞口１４に連続して入賞しやすい。そのため、この場合には、ステップＳ２５４を実行して加算バッファを１加算することなく、ス

テップＳ２５５のみを実行することでクリアバッファをクリアして、入賞順異常と判定しないように制御している。また、普通図柄プロセスフラグの値が３であり、普通図柄プロセス処理の普通電動役物作動処理が実行されているということは、可変入賞球装置１５が開放状態に制御されるのであるから第２始動入賞口１４に連続して入賞しやすい。そのため、この場合にも、ステップＳ２５４を実行して加算バッファを１加算することなく、ステップＳ２５５のみを実行することでクリアバッファをクリアして、入賞順異常と判定しないように制御している。そして、ステップＳ２５９に移行する。

【０４５３】

なお、図６３に示す例に限らず、例えば、ステップＳ２５７，Ｓ２５８に相当する処理を入賞順異常報知処理の開始時に実行し、確変フラグがセットされていた場合または普通図柄プロセスフラグの値が３である場合には、そのまま処理を終了し、以降の処理を実行しないように構成してもよい。また、例えば、確変フラグがセットされていた場合または普通図柄プロセスフラグの値が３である場合（すなわちステップＳ２５７でＹ、またはステップＳ２５８でＹとなった場合）、ステップＳ２５５（クリアバッファをクリアする処理）を実行しないようにしてもよい。つまり、確変フラグがセットされていた場合または普通図柄プロセスフラグの値が３である場合には、加算バッファを１加算する処理とクリアバッファをクリアする処理とのいずれも実行しないようにしてもよい。

【０４５４】

また、この実施の形態では、確変状態（すなわち、高ベース状態）であるか否かの判定と可変入賞球装置１５が開放中であるか否かの判定との両方を実行する場合を示しているが、いずれか一方のみを実行するようにしてもよい。具体的には、ステップＳ２５７の判定処理とステップＳ２５８の判定処理とのいずれか一方のみを実行するようにしてもよい。

【０４５５】

また、第２始動口スイッチ１４ａがオン状態であることにもとづいて直ちにステップＳ２５７以降の処理を実行して第２始動口監視用バッファに格納された値に所定値（本例では１）を加算するのではなく、さらに第２保留記憶数が上限値（本例では４）に達していないことを条件に第２始動口監視用バッファに格納された値に所定値（本例では１）を加算するようにしてもよい（すなわち、有効始動入賞のみをカウントするようにしてもよい）。また、逆に、第２保留記憶数が上限値（本例では４）に達している場合のみ第２始動口監視用バッファに格納された値に所定値（本例では１）を加算するようにしてもよい（すなわち、無効始動入賞のみをカウントするようにしてもよい）。

【０４５６】

また、この実施の形態では、第１始動入賞口１３への入賞については確変状態（すなわち、高ベース状態）中であるかいなかや可変入賞球装置１５が開放中であるか否かにかかわらずカウント（すなわち加算バッファに１加算すること）を継続し、確変状態（すなわち、高ベース状態）中や可変入賞球装置１５の開放中の期間も通して第１始動入賞口１３への遊技球の連続入賞を検出したことにもとづいて入賞順異常と判定する場合を示したが、確変状態（すなわち、高ベース状態）中や可変入賞球装置１５の開放中の期間はカウントを中断するようにしてもよい。そして、確変状態（すなわち、高ベース状態）中や可変入賞球装置１５の開放中の期間をまたがって第１始動入賞口１３への遊技球の連続入賞を検出したときに入賞順異常と判定するようにしてもよい。具体的には、例えば、第１始動入賞口１３に連続して２球入賞した後に確変状態（すなわち、高ベース状態）や可変入賞球装置１５が開放状態となった場合に一旦カウントを中断し、その確変状態（すなわち、高ベース状態）や可変入賞球装置１５の開放状態を終了した後にさらに第１始動入賞口１３に連続して２球入賞した場合に（その確変状態（すなわち、高ベース状態）中や可変入賞球装置１５の開放中に第２始動入賞口１４への入賞もなく）、合計で５球連続して第１始動入賞口１３に入賞したと判定して入賞順異常と判定するようにしてもよい。

【０４５７】

また、この実施の形態では、第１始動入賞口１３に連続して入賞した場合と第２始動入

賞口１４に連続して入賞した場合との両方について入賞順異常と判定する場合を示したが、第１始動入賞口１３に連続して入賞した場合と第２始動入賞口１４に連続して入賞した場合とのいずれか一方の場合についてのみ入賞順異常と判定するように構成してもよい。

【０４５８】

また、この実施の形態では、第１始動入賞口１３に連続して入賞して入賞順異常となった場合と第２始動入賞口１４に連続して入賞して入賞順異常となった場合とで、入賞順異常報知指定コマンドとして共通のコマンドを送信する場合を示したが、第１始動入賞口１３に連続して入賞して入賞順異常となった場合と第２始動入賞口１４に連続して入賞して入賞順異常となった場合とを区別可能に入賞順異常報知指定コマンドを送信するようにしてもよい。具体的には、第１始動入賞口１３に連続して入賞して入賞順異常となった場合には第１入賞順異常報知指定コマンドを送信するようにし、第２始動入賞口１４に連続して入賞して入賞順異常となった場合には第２入賞順異常報知指定コマンドを送信するようにしてもよい。

【０４５９】

また、この実施の形態では、加算バッファの値と比較する判定値が５である場合を示したが、この実施の形態で示した値にかぎらず、例えば、ステップＳ２５９において加算バッファの値が４以上であるかを判定するなど、判定値として他の値を用いてもよい。また、例えば、ステップＳ２５９において加算バッファの値が２以上であるか否かを判定するようにして、第１始動入賞口１３または第２始動入賞口１４に１球でも連続して入賞したら直ちに入賞順異常と判定するようにしてもよい。ただし、この実施の形態では、第１始動入賞口１３または第２始動入賞口１４に４球連続して入賞したことを条件に入賞順異常と判定することによって、判定基準に多少のマージンをもたせることにより誤判定を防止するようにしている。

【０４６０】

また、この実施の形態では、加算バッファの値と判定値（本例では５）とを比較して入賞順異常が発生したか否かを判定しているが、これに限らず、例えば、加算バッファの値とクリアバッファの値とを比較し、その差が所定値（例えば、５）以上であれば入賞順異常と判定するようにしてもよい。

【０４６１】

また、この実施の形態では、入賞順異常の判定を遊技制御用マイクロコンピュータ５６０側で実行する場合を示しているが、演出制御用マイクロコンピュータ１００側で実行するようにしてもよい。この場合、例えば、遊技制御用マイクロコンピュータ５６０は、第１始動入賞口１３に新たな始動入賞があったことを示すコマンドや第２始動入賞口１４に新たな始動入賞があったことを示すコマンドを送信するようにし、演出制御用マイクロコンピュータ１００は、それらのコマンドを受信したことにもとづいて、図６３に示す入賞順異常報知処理と同様の処理を行って入賞順異常を判定するようにすればよい。

【０４６２】

なお、入賞順異常報知処理は、始動口への入賞時の処理（具体的には特別図柄プロセス処理の第１始動口スイッチ通過処理および第２始動口スイッチ通過処理）において実行される（具体的には図６３の入賞順異常報知処理に相当する処理が実行される）ようにしてもよい。

【０４６３】

次に、演出制御手段の動作を説明する。図６４は、演出制御基板８０に搭載されている演出制御手段としての演出制御用マイクロコンピュータ１００（具体的には、演出制御用ＣＰＵ１０１）が実行するメイン処理を示すフローチャートである。演出制御用ＣＰＵ１０１は、電源が投入されると、メイン処理の実行を開始する。メイン処理では、まず、ＲＡＭ領域のクリアや各種初期値の設定、また演出制御の起動間隔（例えば、４ｍｓ）を決めるためのタイマの初期設定等を行うための初期化処理を行う（ステップＳ７０１）。その後、演出制御用ＣＰＵ１０１は、タイマ割込フラグの監視（ステップＳ７０２）を行うループ処理に移行する。タイマ割込が発生すると、演出制御用ＣＰＵ１０１は、タイマ割

込処理においてタイマ割込フラグをセットする。メイン処理において、タイマ割込フラグがセットされていたら、演出制御用ＣＰＵ１０１は、そのフラグをクリアし（ステップＳ７０３）、以下の演出制御処理を実行する。

【０４６４】

演出制御処理において、演出制御用ＣＰＵ１０１は、まず、受信した演出制御コマンドを解析し、受信した演出制御コマンドに応じたフラグをセットする処理等を行う（コマンド解析処理：ステップＳ７０４）。

【０４６５】

次いで、演出制御用ＣＰＵ１０１は、演出制御プロセス処理を行う（ステップＳ７０５）。演出制御プロセス処理では、制御状態に応じた各プロセスのうち、現在の制御状態（演出制御プロセスフラグ）に対応した処理を選択して演出表示装置９の表示制御を実行する。

【０４６６】

次いで、演出制御用ＣＰＵ１０１は、第４図柄プロセス処理を行う（ステップＳ７０６）。第４図柄プロセス処理では、制御状態に応じた各プロセスのうち、現在の制御状態（第４図柄プロセスフラグ）に対応した処理を選択して演出表示装置９の第４図柄表示領域９ｃ，９ｄにおいて第４図柄の表示制御を実行する。

【０４６７】

次いで、大当り図柄決定用乱数などの乱数を生成するためのカウンタのカウント値を更新する乱数更新処理を実行する（ステップＳ７０７）。その後、ステップＳ７０２に移行する。

【０４６８】

図６５は、主基板３１の遊技制御用マイクロコンピュータ５６０から受信した演出制御コマンドを格納するためのコマンド受信バッファの一構成例を示す説明図である。この例では、２バイト構成の演出制御コマンドを６個格納可能なリングバッファ形式のコマンド受信バッファが用いられる。従って、コマンド受信バッファは、受信コマンドバッファ１～１２の１２バイトの領域で構成される。そして、受信したコマンドをどの領域に格納するのかを示すコマンド受信個数カウンタが用いられる。コマンド受信個数カウンタは、０～１１の値をとる。なお、必ずしもリングバッファ形式でなくてもよい。

【０４６９】

なお、遊技制御用マイクロコンピュータ５６０から送信された演出制御コマンドは、演出制御ＩＮＴ信号にもとづく割込処理で受信され、ＲＡＭに形成されているバッファ領域に保存されている。コマンド解析処理では、バッファ領域に保存されている演出制御コマンドがどのコマンドであるのか解析する。なお、演出制御ＩＮＴ信号にもとづく割込処理は、４ｍｓごとに実行されるタイマ割込処理に優先して実行される。

【０４７０】

図６６は、コマンド解析処理（ステップＳ７０４）の具体例を示すフローチャートである。主基板３１から受信された演出制御コマンドは受信コマンドバッファに格納されるが、コマンド解析処理では、演出制御用ＣＰＵ１０１は、コマンド受信バッファに格納されているコマンドの内容を確認する。

【０４７１】

コマンド解析処理において、演出制御用ＣＰＵ１０１は、まず、コマンド受信バッファに受信コマンドが格納されているか否か確認する（ステップＳ６１１）。格納されているか否かは、コマンド受信個数カウンタの値と読出ポインタとを比較することによって判定される。両者が一致している場合が、受信コマンドが格納されていない場合である。コマンド受信バッファに受信コマンドが格納されている場合には、演出制御用ＣＰＵ１０１は、コマンド受信バッファから受信コマンドを読み出す（ステップＳ６１２）。なお、読み出したら読出ポインタの値を＋２しておく（ステップＳ６１３）。＋２するのは２バイト（１コマンド）ずつ読み出すからである。

【０４７２】

受信した演出制御コマンドが変動パターンコマンドであれば（ステップＳ６１４）、演出制御用ＣＰＵ１０１は、受信した変動パターンコマンドを、ＲＡＭに形成されている変動パターンコマンド格納領域に格納する（ステップＳ６１５）。そして、変動パターンコマンド受信フラグをセットする（ステップＳ６１６）。

【０４７３】

受信した演出制御コマンドが表示結果指定コマンドであれば（ステップＳ６１７）、演出制御用ＣＰＵ１０１は、受信した表示結果指定コマンド（表示結果１指定コマンド～表示結果５指定コマンド）を、ＲＡＭに形成されている表示結果指定コマンド格納領域に格納する（ステップＳ６１８）。

【０４７４】

受信した演出制御コマンドが図柄確定指定コマンドであれば（ステップＳ６１９）、演出制御用ＣＰＵ１０１は、確定コマンド受信フラグをセットする（ステップＳ６２０）。

【０４７５】

受信した演出制御コマンドが大当り開始指定コマンドであれば（ステップＳ６２１）、演出制御用ＣＰＵ１０１は、大当り開始指定コマンド受信フラグをセットする（ステップＳ６２２）。

【０４７６】

受信した演出制御コマンドが小当り／突然確変大当り開始指定コマンドであれば（ステップＳ６２３）、演出制御用ＣＰＵ１０１は、小当り／突然確変大当り開始指定コマンド受信フラグをセットする（ステップＳ６２４）。

【０４７７】

受信した演出制御コマンドが入賞順異常報知指定コマンドであれば（ステップＳ６２５）、演出制御用ＣＰＵ１０１は、ランプドライバ基板３５に対して制御信号（ランプ制御実行データ）を出力することによって、所定の異常報知パターンで枠ＬＥＤ２８の点灯制御を行う（ステップＳ６２６）。また、演出制御用ＣＰＵ１０１は、音声出力基板７０に対して制御信号（音番号データ）を出力することによって、所定の異常報知音をスピーカ２７から出力させる（ステップＳ６２７）。

【０４７８】

なお、この実施の形態では、枠ＬＥＤ２８を所定の異常報知パターンで点灯させるとともに所定の異常報知音をスピーカ２７から出力させることによって入賞順異常報知を行う場合を示しているが、入賞順異常報知の態様は、この実施の形態で示したものにかぎられない。例えば、枠ＬＥＤ２８の所定の異常報知パターンでの点灯と所定の異常報知音のスピーカ２７からの出力とのいずれか一方のみを行うことにより入賞順異常報知を行ってもよい。また、例えば、演出表示装置９において「入賞順異常です」などの文字列を含む入賞順異常報知画面を表示することにより入賞順異常報知を行ってもよい。

【０４７９】

受信した演出制御コマンドがその他のコマンドであれば、演出制御用ＣＰＵ１０１は、受信した演出制御コマンドに応じたフラグをセットする（ステップＳ６２８）。そして、ステップＳ６１１に移行する。

【０４８０】

図６７は、図６４に示されたメイン処理における演出制御プロセス処理（ステップＳ７０５）を示すフローチャートである。演出制御プロセス処理では、演出制御用ＣＰＵ１０１は、演出制御プロセスフラグの値に応じてステップＳ８００～Ｓ８０７のうちのいずれかの処理を行う。各処理において、以下のような処理を実行する。なお、演出制御プロセス処理では、演出表示装置９の表示状態が制御され、演出図柄の可変表示が実現されるが、第１特別図柄の変動に同期した演出図柄の可変表示に関する制御も、第２特別図柄の変動に同期した演出図柄の可変表示に関する制御も、一つの演出制御プロセス処理において実行される。なお、第１特別図柄の変動に同期した演出図柄の可変表示と、第２特別図柄の変動に同期した演出図柄の可変表示とを、別の演出制御プロセス処理により実行するように構成してもよい。また、この場合、いずれの演出制御プロセス処理により演出図柄の

変動表示が実行されているかによって、いずれの特別図柄の変動表示が実行されているかを判断するようにしてもよい。

【０４８１】

変動パターンコマンド受信待ち処理（ステップＳ８００）：遊技制御用マイクロコンピュータ５６０から変動パターンコマンドを受信しているか否か確認する。具体的には、コマンド解析処理でセットされる変動パターンコマンド受信フラグがセットされているか否か確認する。変動パターンコマンドを受信していれば、演出制御プロセスフラグの値を演出図柄変動開始処理（ステップＳ８０１）に対応した値に変更する。

【０４８２】

演出図柄変動開始処理（ステップＳ８０１）：演出図柄の変動が開始されるように制御する。そして、演出制御プロセスフラグの値を演出図柄変動中処理（ステップＳ８０２）に対応した値に更新する。

【０４８３】

演出図柄変動中処理（ステップＳ８０２）：変動パターンを構成する各変動状態（変動速度）の切替タイミング等を制御するとともに、変動時間の終了を監視する。そして、変動時間が終了したら、演出制御プロセスフラグの値を演出図柄変動停止処理（ステップＳ８０３）に対応した値に更新する。

【０４８４】

演出図柄変動停止処理（ステップＳ８０３）：演出図柄の変動を停止し表示結果（停止図柄）を導出表示する制御を行う。そして、演出制御プロセスフラグの値を大当り表示処理（ステップＳ８０４）または変動パターンコマンド受信待ち処理（ステップＳ８００）に対応した値に更新する。

【０４８５】

大当り表示処理（ステップＳ８０４）：変動時間の終了後、演出表示装置９に大当りの発生を報知するための画面を表示する制御を行う。そして、演出制御プロセスフラグの値をラウンド中処理（ステップＳ８０５）に対応した値に更新する。

【０４８６】

ラウンド中処理（ステップＳ８０５）：ラウンド中の表示制御を行う。そして、ラウンド終了条件が成立したら、最終ラウンドが終了していなければ、演出制御プロセスフラグの値をラウンド後処理（ステップＳ８０６）に対応した値に更新する。最終ラウンドが終了していれば、演出制御プロセスフラグの値を大当り終了処理（ステップＳ８０７）に対応した値に更新する。

【０４８７】

ラウンド後処理（ステップＳ８０６）：ラウンド間の表示制御を行う。そして、ラウンド開始条件が成立したら、演出制御プロセスフラグの値をラウンド中処理（ステップＳ８０５）に対応した値に更新する。

【０４８８】

大当り終了演出処理（ステップＳ８０７）：演出表示装置９において、大当り遊技状態が終了したことを遊技者に報知する表示制御を行う。そして、演出制御プロセスフラグの値を変動パターンコマンド受信待ち処理（ステップＳ８００）に対応した値に更新する。

【０４８９】

なお、この実施の形態では、始動入賞口への入賞が発生すると、対応する始動口監視バッファに格納された値に所定値が加算され、加算結果にもとづいて入賞順異常が発生したことを検出するように構成されているが、これに限らず、例えば、始動入賞口への入賞が発生すると、対応する始動口監視バッファに格納された値から所定値が減算され、減算結果にもとづいて入賞順異常が発生したことを検出するように構成してもよい。

【０４９０】

図６８は、入賞順異常報知処理の変形例を示すフローチャートである。図６８に示す入賞順異常報知処理の変形例では、ステップＳ２５０，２５１に代えて、第２始動口監視バッファを減算バッファにセットし（ステップＳ２５０ｂ）、第１始動口監視バッファを初

期値設置バッファにセットする（ステップＳ２５１ｂ）処理が実行される。減算バッファは、セットされた始動口監視用バッファを所定値（本例では１）の減算対象とするために用いられる。また、初期値設定バッファは、セットされた始動口監視用バッファを初期値設定（本例では初期値（５）をセットする）対象とするために用いられる。

【０４９１】

また、図６８に示す入賞順異常報知処理の変形例では、ステップＳ２５４，Ｓ２５５に代えて、減算バッファを１減算し、減算バッファをロードし（ステップＳ２５４ｂ）、初期値データ（本例では５）をセットし、初期値設定バッファにストアする（ステップＳ２５５ｂ）処理が実行される。すなわち、ステップＳ２５４ｂにおいて、減算バッファにセットされた始動口監視用バッファに格納された値から所定値（本例では１）を減算し、ステップＳ２５５ｂにおいて、初期値設定バッファにセットされた始動口監視用バッファに格納された値を初期値（本例では５）に設定する。このようにすることによって、同じ始動口に連続して入賞が発生した場合には、対応する始動口監視用バッファに格納された値が連続して１減算され、他方の始動口に対応する始動口監視用バッファに格納された値が初期値（本例では５）に設定される。また、前回入賞した始動口とは異なる始動口に入賞した場合には、前回減算対象として所定値（本例では１）が減算された始動口監視用バッファに格納された値が初期値（本例では５）に設定され、前回初期値設定対象として初期値（本例では５）が設置された始動口監視用バッファに格納された値から所定値（本例では１）が減算される。以下、「減算バッファの値」という表現を用いることがあるが、具体的には、減算バッファにセットされた始動口監視用バッファに格納された値のことを示す。また、「減算バッファを１減算する」という表現を用いることがあるが、具体的には、減算バッファにセットされた始動口監視用バッファに格納された値から所定値（本例では１）を減算することである。また、「減算バッファに初期値をセットする」という表現を用いることがあるが、具体的には、減算バッファにセットされた始動口監視用バッファに格納された値を初期値（本例では５）に設定することである。

【０４９２】

また、図６８に示す入賞順異常報知処理の変形例では、ステップＳ２５９，Ｓ２６０に代えて、減算バッファの値が０であるか否かを判定し（ステップＳ２５９ｂ）、減算バッファの値が０となっていれば、減算バッファに初期値をセットする（ステップＳ２６０ｂ）処理が実行される。なお、減算バッファの値と判定値（本例では０）とを比較して入賞順異常が発生したか否かを判定しているが、これに限らず、例えば、減算バッファの値と初期値設定バッファの値とを比較し、その差が所定値（例えば、５）以上であれば入賞順異常と判定するようにしてもよい。

【０４９３】

以上のように処理を実行することによって、始動入賞口への入賞が発生すると、対応する始動口監視バッファに格納された値から所定値が減算され、減算結果にもとづいて入賞順異常が発生したことを検出するように構成することができる。また、図６８に示すように構成する場合には、図６９に示すメイン処理の変形例のように、Ｓ１０ａに代えて、減算バッファに初期値を設定する処理（ステップＳ１０ｂ）を実行する。このようにすることで、電源投入後に減算バッファの値が初期化されることで、誤って入賞順異常を検出してしまうことを防止することができる。

【０４９４】

また、この実施の形態において、予告演出の対象となる変動表示が開始されるよりも前に実行される先読み予告演出を実行するように構成してもよい。先読み予告演出を実行するように構成する場合、例えば、図７０に示すように、遊技制御用マイクロコンピュータ５６０は、第１始動入賞口１３や第２始動入賞口１４への始動入賞が発生したタイミングで第１始動口スイッチ通過処理（ステップＳ３１２参照）や第２始動口スイッチ通過処理（ステップＳ３１４参照）にて始動入賞時の判定を行い、その判定結果を示す入賞時判定結果コマンドを送信する制御を行う（図７０のステップＳ２０６１Ａ，Ｓ２０６１Ｂ）。この場合、例えば、入賞時判定結果指定コマンドとして、大当りとなるか否かや、小当り

となるか否か、大当りの種別の判定結果を示す図柄指定コマンド、および変動パターン種別判定用乱数の値がいずれの判定値の範囲となるかの判定結果（変動パターン種別の判定結果）を示す変動カテゴリコマンドを送信する。従って、この場合、新たな始動入賞が発生したタイミングで１タイマ割り込み内で始動入賞時コマンドとして図柄指定コマンド、変動カテゴリコマンド、および保留記憶数加算指定コマンド（第１保留記憶数加算指定コマンド、第２保留記憶数加算指定コマンド）が送信されることになる。

【０４９５】

また、図７１に示すように、コマンド解析処理において、入賞時判定結果指定コマンドを受信すると（ステップＳ６２９）、演出制御用マイクロコンピュータ１００は、入賞時判定結果指定コマンドに応じた入賞時判定結果を入賞時判定結果記憶バッファに保存し（ステップＳ６３０）、入賞時判定結果記憶バッファに保存された入賞時判定結果にもとづいて先読み演出決定処理を行う（ステップＳ６３１）。その後、演出制御用マイクロコンピュータ１００は、先読み演出決定処理の決定結果にしたがって、先読み演出を実行する（例えば、演出図柄変動中処理（ステップＳ８０２）において実行される）。先読み演出は、例えば、予告対象となる当該変動表示よりも前に開始される変動表示中に、演出表示装置９の背景表示を特殊背景に切り替えてを表示することや、カウントダウン表示、当該変動表示に対応する保留表示の表示態様を変化させることによって実現される。また、ステップＳ６３１では、演出制御用マイクロコンピュータ１００は、例えば、図７２に示すように、高ベース状態であるか否かと、合算保留記憶数とに応じて、先読み演出の演出態様を決定する。具体的には、図７２に示すように、高ベース状態であるときには、当該変動表示よりも前に開始される１回から最高４回の変動表示に亘る先読み演出が実行され、高ベース状態でないときには、当該変動表示よりも前に開始される１回から最高８回の変動表示に亘る先読み演出が実行される。このようにすることで、高ベース状態ではないとき（すなわち通常状態であるとき）には、振分装置によって第１始動入賞口１３と第２始動入賞口１４とに交互に入賞するため、最大８回（第１保留記憶数と第２保留記憶数とを足した上限）で完結する連続演出（例えば、特殊背景に切り替える）を実行可能とし、高ベース状態であるときには、振分装置を介さずに可変入賞球装置１５によって第２始動入賞口１４のみに入賞しやすいため、最大４回（第２保留記憶数の上限）で完結する連続演出（例えば、カウントダウン表示）を実行可能とすることができる。したがって、遊技状態に応じた先読み演出を実行することができる。

【０４９６】

また、先読み演出の演出態様は、当該変動表示の表示結果が大当りか否かに応じて、決定されるようにしてもよい。例えば、当該変動表示に対応する保留表示が８個目の保留表示である場合（すなわち、先に実行される変動表示が８回ある場合）、当該変動表示の表示結果が大当りであるか否かに応じて、始動入賞後に直ちに先読み演出を開始する（つまり、８回の変動表示に亘って実行される）か、始動入賞後に数回の変動表示が行われた後に先読み演出を開始する（例えば、４回の変動表示に亘って実行される）かを決定するようにしてもよい。このように、当該変動表示の表示結果が大当りであるか否かに応じて、先読み演出の演出態様（何回の変動表示に亘って実行されるか）を決定することによって、例えば、多くの変動表示に亘って先読み演出が実行されるほど大当りの期待度が高くなるようにすることができ、遊技興趣を高めることができる。

【０４９７】

以上に説明したように、この実施の形態によれば、遊技球を振り分けるための振分装置２００が遊技領域７に設けられている。振分装置２００は、遊技領域７に進入した遊技球が当該振分装置２００に流入可能な流入口２０１と、流入口から流入した遊技球が通過可能な複数の通路（本例では、左側通路２０３、右側通路２０４）と、流入口から流入した遊技球を複数の通路のうちのいずれかに振り分ける振分手段（本例では、振分部材２０２）とを含む。また、振分手段は、流入口２０１から遊技球が流入したことにもとづいて、複数の通路のうちの第１通路（本例では、左側通路２０３）に遊技球を振り分けやすい第１状態（本例では、図２（ａ），（ｄ）に示すように振分部材２０２が右側に倒れた状態

）と第２通路（本例では、右側通路２０４）に遊技球を振り分けやすい第２状態（本例では、図２（ｂ），（ｃ）に示すように振分部材２０２が左側に倒れた状態）とに所定の順序に従って切り替わる（本例では、遊技球の自重によって交互に切り替わる）。また、第１始動領域（本例では、第１始動入賞口１３）は、第１通路に振り分けられた遊技球が通過しやすい態様で設けられている（本例では、図２に示すように、左側流出口２０５の下方に第１始動入賞口１３が設けられている）とともに、第２始動領域（本例では、第２始動入賞口１４）は、第２通路に振り分けられた遊技球が通過しやすい態様で設けられている（本例では、図２に示すように、右側流出口２０６の下方に第２始動入賞口１４が設けられている）。また、遊技制御用マイクロコンピュータ５６０は、遊技媒体が第１始動領域または第２始動領域を通過したことにもとづいて遊技媒体が通過した始動領域に対応する記憶領域（本例では、第１始動口監視用バッファまたは第２始動口監視用バッファ）を加算対象として当該記憶領域に格納された値に所定値（本例では１）を加算し、加算結果が所定条件（例えば、加算対象とされた記憶領域に格納された値が５以上である、または第１始動口監視用バッファに格納された値と第２始動口監視用バッファに格納された値との差（加算バッファの値とクリアバッファの値との差）が所定値（例えば５）以上である）を満たしたことにもとづいて異常と判定する。そのため、振分装置２００の異常判定を行うことができるので、振分装置２００の異常により遊技を正常に行えなくなることを防止することができる。

【０４９８】

また、この実施の形態によれば、所定事象が発生（本例では、ＩＡＴ５０６ａからのＩＡＴ信号の入力、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号の入力）したことにもとづいて第１リセット（本例では、システムリセット）を発生させるか第２リセット（本例では、ユーザリセット）を発生させるかを設定可能である（図１６に示すリセット設定（ＫＲＥＳ）のビット７参照）。そして、第１リセットの発生後にはセキュリティチェックを実行する一方、第２リセットの発生後にはセキュリティチェックを実行しない。そのため、遊技機や遊技店の状況などに応じて所定事象が発生したときに行うリセットの種類を最適なものに設定できるので、遊技制御用マイクロコンピュータ５６０に関するセキュリティ性を向上させることができる。

【０４９９】

なお、この実施の形態では、所定事象の発生として、ＩＡＴ５０６ａからのＩＡＴ信号を入力した場合と、ウオッチドッグタイマ（ＷＤＴ）５０６ｂからのタイムアウト信号を入力した場合とを示しているが、この実施の形態で示したものにかぎらず、遊技制御用マイクロコンピュータ５６０をリセットすべき何らかのエラーなどの状況が発生したことにもとづいて、所定事象が発生したとしてリセットするものであってもよい。

【０５００】

また、この実施の形態によれば、所定事象の発生には、ウオッチドッグタイマ（ＷＤＴ）５０６ｂのタイムアウトが含まれ、ウオッチドッグタイマ（ＷＤＴ）５０６ｂを起動させるか否かを設定可能である（本例では、図１４に示すリセット設定（ＫＲＥＳ）のビット３－０に”００００”を設定する）。そして、ウオッチドッグタイマ（ＷＤＴ）５０６ｂを起動させないと設定した場合であっても、所定事象が発生したことにもとづいて第１リセットを発生させるか第２リセットを発生させるかを設定可能である。具体的には、図１４に示すリセット設定（ＫＲＥＳ）において、ビット３－０に”００００”を設定していても、ビット７の設定を行うことによってリセットの種類を設定可能である。そのため、ウオッチドッグタイマ（ＷＤＴ）５０６ｂの設定にかかわらず、所定事象が発生したことにもとづいて発生させるリセットの種類の設定を共通化することができる。

【０５０１】

また、この実施の形態によれば、所定事象の発生には、指定された領域以外の領域に格納されたプログラムを実行する指定領域外実行（本例では、指定エリア外走行禁止（ＩＡＴ）が含まれる。そして、遊技制御用マイクロコンピュータ５６０は、所定の処理として所定時間（本例では４ｍｓ）毎に発生するタイマ割込に応じて実行されるタイマ割込処理

（図５１に示すタイマ割込処理）の実行中に指定領域外実行が発生（本例では、ＩＡＴ回路５０６ａからＩＡＴ信号を入力）した場合に、ＲＡＭ５５（バックアップＲＡＭ）の記憶内容を初期化する（本例では、リセットの後、図５０に示すステップＳ１０が実行される）。そのため、意図しないプログラムが実行された場合のセキュリティ性を向上させることができる。

【０５０２】

また、この実施の形態の変形例によれば、遊技制御用マイクロコンピュータ５６０は、遊技媒体が第１始動領域または第２始動領域を通過したことにもとづいて遊技媒体が通過した始動領域に対応する記憶領域（本例では、第１始動口監視用バッファまたは第２始動口監視用バッファ）を減算対象として当該記憶領域に格納された値から所定値（本例では１）を減算し、減算結果が所定条件（例えば、減算対象とされた記憶領域に格納された値が０である、または第１始動口監視用バッファに格納された値と第２始動口監視用バッファに格納された値との差（減算バッファの値と初期値設定バッファの値との差）が所定値（例えば５）以上である）を満たしたことにもとづいて異常と判定する。そのため、振分装置２００の異常判定を行うことができるので、振分装置２００の異常により遊技を正常に行えなくなることを防止することができる。

【０５０３】

また、この実施の形態によれば、ＲＡＭ５５（バックアップＲＡＭ）の記憶内容を初期化する（本例では、リセットの後、図５０に示すステップＳ１０が実行される）とともに、ステップＳ１０ａにおいて加算バッファおよびクリアバッファがクリアされる（またはステップＳ１０ｂにおいて減算バッファに初期値（本例では５）が設定される）。そのため、停電復旧時にも、振分装置２００の異常判定を正しく行うことができるので、振分装置２００の異常により遊技を正常に行えなくなることを防止することができる。

【０５０４】

また、この実施の形態によれば、遊技制御用マイクロコンピュータ５６０は、第１始動領域に対応する記憶領域（本例では、第１始動口監視用バッファまたは第２始動口監視用バッファ）を加算対象とし、第２始動領域に対応する記憶領域（本例では、第１始動口監視用バッファまたは第２始動口監視用バッファ（第１始動領域に対応する記憶領域とは異なる））をクリア対象として仮設定し、遊技媒体が第２始動領域を通過したことにもとづいて、仮設定された加算対象とクリア対象とを入れ替え、加算対象とされた記憶領域に格納された値に所定値（本例では１）を加算し、クリア対象とされた記憶領域に格納された値をクリアする。そして、記憶領域に格納された値にもとづいて（例えば、加算対象とされた記憶領域に格納された値が５以上であるとき、または第１始動口監視用バッファに格納された値と第２始動口監視用バッファに格納された値との差（加算バッファの値とクリアバッファの値との差）が所定値（例えば５）以上であるとき）異常と判定する。そのため、振分装置２００の異常判定を行うことができるので、振分装置２００の異常により遊技を正常に行えなくなることを防止することができる。

【０５０５】

また、この実施の形態の変形例によれば、また、遊技制御用マイクロコンピュータ５６０は、第１始動領域に対応する記憶領域（本例では、第１始動口監視用バッファまたは第２始動口監視用バッファ）を減算対象とし、第２始動領域に対応する記憶領域（本例では、第１始動口監視用バッファまたは第２始動口監視用バッファ（第１始動領域に対応する記憶領域とは異なる））を初期値設定対象として仮設定し、遊技媒体が第２始動領域を通過したことにもとづいて、仮設定された減算対象と初期値設定対象とを入れ替え、減算対象とされた記憶領域に格納された値から所定値（本例では１）を減算し、初期値設定対象とされた記憶領域に格納された値を初期値（本例では５）に設定する。そして、記憶領域に格納された値にもとづいて（例えば、減算対象とされた記憶領域に格納された値が０であるとき、または第１始動口監視用バッファに格納された値と第２始動口監視用バッファに格納された値との差（減算バッファの値と初期値設定バッファの値との差）が所定値（例えば５）以上であるとき）異常と判定する。そのため、振分装置２００の異常判定を行

うことができるので、振分装置２００の異常により遊技を正常に行えなくなることを防止することができる。

【０５０６】

（変形例２）

次に、この実施の形態における変形例２について説明する。

上述した実施の形態では、図５７に示す始動入賞時に実行される変動パターン設定処理（Ｓ３０１）において、小当り前後での変動では低確低ベース用の変動パターンテーブルである通常用変動パターン判定テーブルを共通に用いていた。これに対して、この変形例２では、小当り前の変動では、通常用変動パターン判定テーブルを用いて、小当り後の変動では小当り用変動パターン判定テーブルを用いてもよい。このような小当り用変動パターン判定テーブルの一例としては、特定の変動パターン（特定の可変表示パターン）としてのノーマルはずれの変動パターンの選択割合が通常用変動パターン判定テーブルにおける選択割合よりも高い値に（例えば３０％から４０％に）、非リーチはずれの選択割合が通常用変動パターン判定テーブルにおける選択割合よりも低い値に（例えば５０％から４０％に）、それぞれ予め設定されたテーブルである。このような小当り用変動パターン判定テーブルについては、小当り後、後述の共通演出の実行回数に対応する回数としての所定回数（例えば、２０回）の可変表示が行われるまで、又は次回大当りまで用いてもよい。また、高ベース状態において小当りに当選した場合には、高ベース状態終了時まで、小当り用変動パターン判定テーブルを用いてもよい。

【０５０７】

また、この変形例２では、突然確変大当り終了後・小当り終了後に、主基板３１側から演出制御基板８０側に突然確変大当り終了後・小当り終了後の遊技状態であることを示す演出制御コマンドを送ることによって、演出モードとして共通演出（潜伏演出）が実行される。この共通演出では、遊技者の見た目上、突然確変大当りであるか小当りであるかまでは認識できない又は認識困難な演出である。従って、遊技者は突然確変大当りの発生によって高確率状態（確変状態や確変時短状態）に制御されたか、小当りの発生によって低確率状態（通常状態や時短状態）に制御されたか、について認識することができないので、突然確変大当りの遊技や小当りの遊技が終了した後の遊技状態を、確変状態が潜伏しているかもしれないとの期待を持たせることができる。また、この共通演出は、小当り後については所定回数の可変表示により終了し、突然確変大当り後については次回大当りまで継続する（突然確変大当り後も小当り後と同様に所定回数の可変表示により終了してもよい）。

【０５０８】

ここで、この変形例２においては、小当り後に小当り用変動パターン判定テーブルが用いられるため、小当り後の共通演出の実行中において、リーチはずれの変動パターンの選択割合が小当り前に比べて高くなっている。これにより、共通演出の実行中にリーチ演出の発生頻度が小当り前よりも高くなるため、共通演出についての興趣を向上できる。これに加えて、遊技者の遊技意欲を向上させ、遊技機の稼動を向上させることができる。

【０５０９】

なお、この変形例２に対して、以下のような更なる変更を加えてもよい。

（１）小当り後の可変表示のみならず、突然確変大当り後の可変表示についても、小当り後用変動パターン判定テーブルと同様の変動パターン判定テーブルを用いてもよい。

（２）はずれとなる可変表示のみならず、大当り又は小当りとなる可変表示についても、変動パターンの選択割合を変更してもよい。例えば通常用変動パターン判定テーブルで規定されたノーマル当りの選択割合を５％から２０％に高めるように小当り用変動パターン判定テーブルを設定してもよい。

（３）小当りについて複数の種別を設定し、当選した小当り種別によって、通常用変動パターン判定テーブルを継続使用したり、通常用変動パターン判定テーブルから小当り用変動パターン判定テーブルに切り替えたりしてもよい。

（４）小当り用変動パターン判定テーブルを複数種類予め設定し、小当り後の可変表示回

数に応じて、小当り用変動パターン判定テーブルを切り替えてもよい。
（５）小当り種別と変動パターンの選択割合が異なる小当り用変動パターン判定テーブルとをそれぞれ複数設定して、小当り種別に対応する小当り用変動パターン判定テーブルを選択するようにしてもよい。
（６）小当りについて種別を複数設定し、当選した小当り種別によって、通常用変動パターン判定テーブルを継続使用したり、通常用変動パターン判定テーブルから小当り用変動パターン判定テーブルに切り替えたりしてもよい。
（７）小当り用変動パターン判定テーブルについて、リーチはずれの変動パターンの選択割合を通常用変動パターン判定テーブルにおけるリーチはずれの変動パターンの選択割合よりも高くしたが、この例に限らず、非リーチはずれの変動パターンを特定の変動パターンとし、非リーチはずれの変動パターンの選択割合を通常用変動パターン判定テーブルにおける非リーチはずれの変動パターンの選択割合よりも高めてもよい。この場合、可変表示結果がはずれとなるリーチ演出用の変動パターンの選択割合が相対的に低下するため、リーチ演出についての可変表示結果が大当りとなる期待度が相対的に上昇する。これにより、リーチ演出の発生に対して、より一層注目させることができ、興趣を向上できる。
（８）小当り用変動パターン判定テーブルでは、リーチはずれの変動パターンの選択割合を通常用変動パターン判定テーブルにおけるリーチはずれの変動パターンの選択割合よりも高くしたが、この例に限らず、特定の変動パターンとして、スーパーはずれの変動パターンや、擬似連１ノーマルはずれなどの変動パターンとし、これらの変動パターンの選択割合を、通常用変動パターン判定テーブルにおける選択割合から変更してもよい。

【０５１０】
　また、この変形例２において、通常状態では実行されない又は実行されにくい特殊演出（例えば大当り期待度が他のリーチ演出に比べて高いリーチ演出や、大当り確定のリーチ演出）が実行可能又は実行されやすいよう、変動パターン及び小当り用変動パターン判定テーブルを設定し、演出制御用マイクロコンピュータ１００が特殊演出を実行してもよい。この場合には、小当り後の可変表示において特殊演出が実行されるかどうかについて遊技者に期待させることができ、興趣を向上できる。

【０５１１】
　なお、上記の実施の形態においては、変動時間およびリーチ演出の種類や擬似連の有無等の変動態様を示す変動パターンを演出制御用マイクロコンピュータ１００に通知するために、変動を開始するときに１つの変動パターンコマンドを送信する例を示したが、２つ乃至それ以上のコマンドにより変動パターンを演出制御用マイクロコンピュータ１００に通知するようにしてもよい。具体的には、２つのコマンドにより通知する場合、遊技制御用マイクロコンピュータ５６０は、１つ目のコマンドでは擬似連の有無、滑り演出の有無など、リーチとなる以前（リーチとならない場合には所謂第２停止の前）の変動時間や変動態様を示すコマンドを送信し、２つ目のコマンドではリーチの種類や再抽選演出の有無など、リーチとなった以降（リーチとならない場合には所謂第２停止の後）の変動時間や変動態様を示すコマンドを送信するようにしてもよい。この場合、演出制御用マイクロコンピュータ１００は２つのコマンドの組合せから導かれる変動時間にもとづいて変動表示における演出制御を行うようにすればよい。なお、遊技制御用マイクロコンピュータ５６０の方では２つのコマンドのそれぞれにより変動時間を通知し、それぞれのタイミングで実行される具体的な変動態様については演出制御用マイクロコンピュータ１００の方で選択を行うようにしてもよい。２つのコマンドを送る場合、同一のタイマ割込内で２つのコマンドを送信する様にしてもよく、１つ目のコマンドを送信した後、所定期間が経過してから（例えば次のタイマ割込において）２つ目のコマンドを送信するようにしてもよい。なお、それぞれのコマンドで示される変動態様はこの例に限定されるわけではなく、送信する順序についても適宜変更可能である。このように２つ乃至それ以上のコマンドにより変動パターンを通知するようにすることで、変動パターンコマンドとして記憶しておかなければならないデータ量を削減することができる。

【０５１２】

また、上記の実施の形態では、演出装置を制御する回路が搭載された基板として、演出制御基板８０、音声出力基板７０およびランプドライバ基板３５が設けられているが、演出装置を制御する回路を１つの基板に搭載してもよい。さらに、演出表示装置９等を制御する回路が搭載された第１の演出制御基板（表示制御基板）と、その他の演出装置（ランプ、ＬＥＤ、スピーカ２７など）を制御する回路が搭載された第２の演出制御基板との２つの基板を設けるようにしてもよい。

【０５１３】

また、上記の実施の形態では、遊技制御用マイクロコンピュータ５６０は、演出制御用マイクロコンピュータ１００に対して直接コマンドを送信していたが、遊技制御用マイクロコンピュータ５６０が他の基板（例えば、図５に示す音声出力基板７０やランプドライバ基板３５など、または音声出力基板７０に搭載されている回路による機能とランプドライバ基板３５に搭載されている回路による機能とを備えた音／ランプ基板）に演出制御コマンドを送信し、他の基板を経由して演出制御基板８０における演出制御用マイクロコンピュータ１００に送信されるようにしてもよい。その場合、他の基板においてコマンドが単に通過するようにしてもよいし、音声出力基板７０、ランプドライバ基板３５、音／ランプ基板にマイクロコンピュータ等の制御手段を搭載し、制御手段がコマンドを受信したことに応じて音声制御やランプ制御に関わる制御を実行し、さらに、受信したコマンドを、そのまま、または例えば簡略化したコマンドに変更して、演出表示装置９を制御する演出制御用マイクロコンピュータ１００に送信するようにしてもよい。その場合でも、演出制御用マイクロコンピュータ１００は、上記の実施の形態における遊技制御用マイクロコンピュータ５６０から直接受信した演出制御コマンドに応じて表示制御を行うのと同様に、音声出力基板７０、ランプドライバ基板３５または音／ランプ基板から受信したコマンドに応じて表示制御を行うことができる。

【０５１４】

また、上記の実施の形態では、遊技機としてパチンコ機を例にしたが、本発明を、メダルが投入されて所定の賭け数が設定され、遊技者による操作レバーの操作に応じて複数種類の図柄を回転させ、遊技者によるストップボタンの操作に応じて図柄を停止させたときに停止図柄の組合せが特定の図柄の組み合わせになると、所定数のメダルが遊技者に払い出されるスロット機に適用することも可能である。

【０５１５】

また、上記の実施の形態では、遊技機として遊技媒体を使用するものを例にしたが本発明による遊技機は、所定数の景品としての遊技媒体を払い出す遊技機に限定されず、遊技球等の遊技媒体を封入し景品の付与条件が成立した場合に得点を付与する封入式の遊技機に適用することもできる。

【０５１６】

なお、上記の実施の形態において、「割合が異なる」とは、Ａ：Ｂ＝７０％：３０％やＡ：Ｂ＝４０％：６０％のような関係で割合が異なるものだけにかぎらず、Ａ：Ｂ＝１００％：０％のような関係で割合が異なるもの（すなわち、一方が１００％の割り振りで他方が０％の割り振りとなるようなもの）も含む概念である。

【産業上の利用可能性】

【０５１７】

本発明は、所定の遊技を行うことが可能なパチンコ遊技機等の遊技機に好適に適用される。

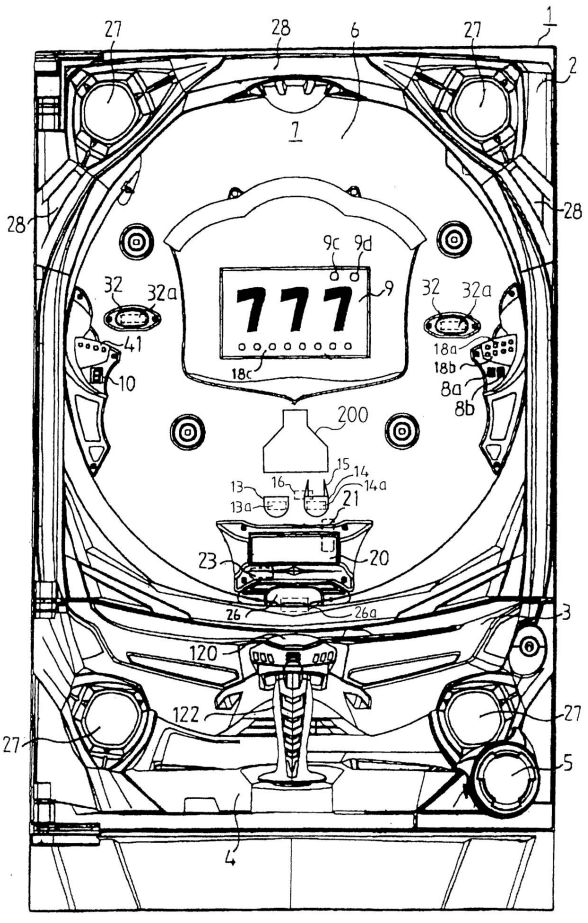
【符号の説明】

【０５１８】

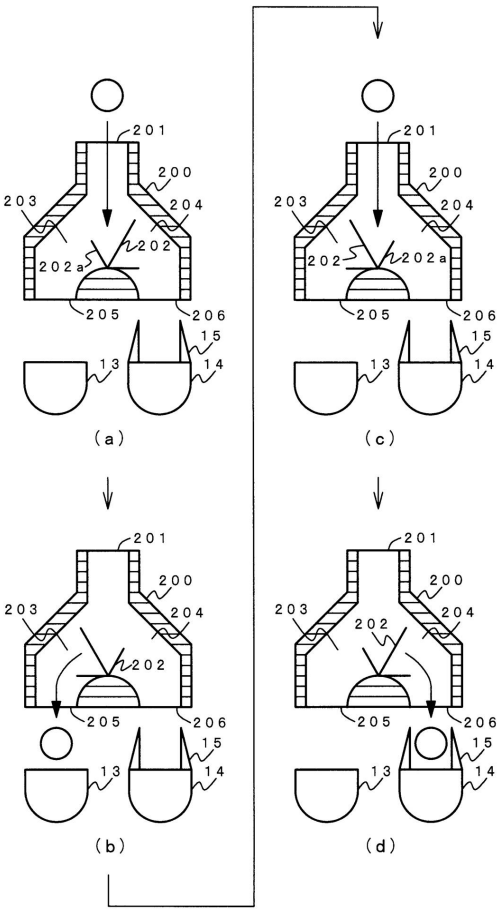
１　パチンコ遊技機
８ａ　第１特別図柄表示器
８ｂ　第２特別図柄表示器
９　演出表示装置
１３　第１始動入賞口

１４　　第２始動入賞口
２０　　特別可変入賞球装置
３１　　遊技制御基板（主基板）
５６　　ＣＰＵ
５０２　クロック回路
５０６　リセット／割り込みコントローラ
５０６ａ　ＩＡＴ回路
５０６ｂ　ウオッチドッグタイマ（ＷＤＴ）
５０７　フリーランカウンタ回路
５０８ａ　８ビット乱数回路
５０８ｂ　１６ビット乱数回路
５２５ａ，５２５ｂ　乱数生成回路
５３７　更新監視回路
５６０　遊技制御用マイクロコンピュータ
８０　　演出制御基板
１００　演出制御用マイクロコンピュータ
１０１　演出制御用ＣＰＵ
１０９　ＶＤＰ

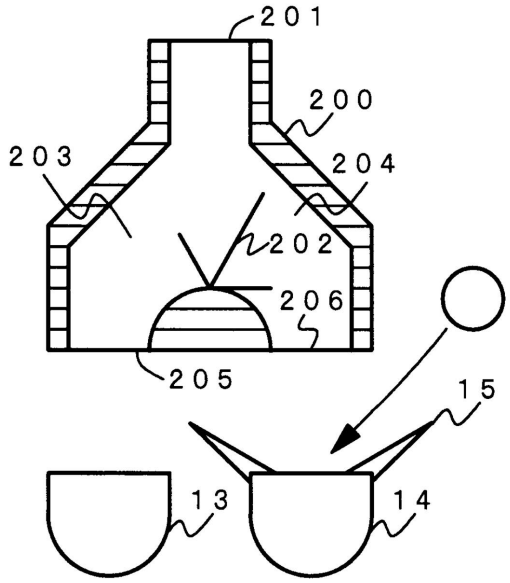
【図１】



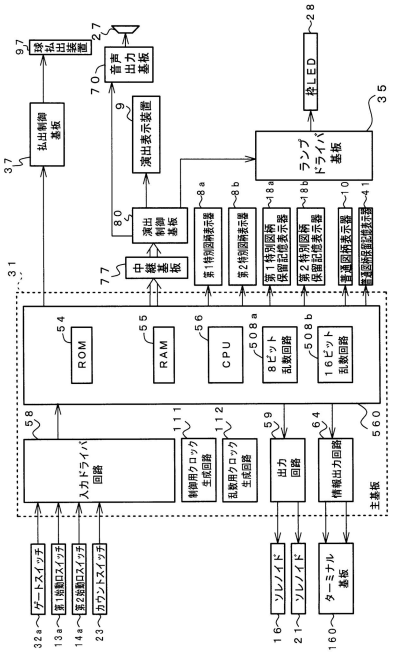
【図２】



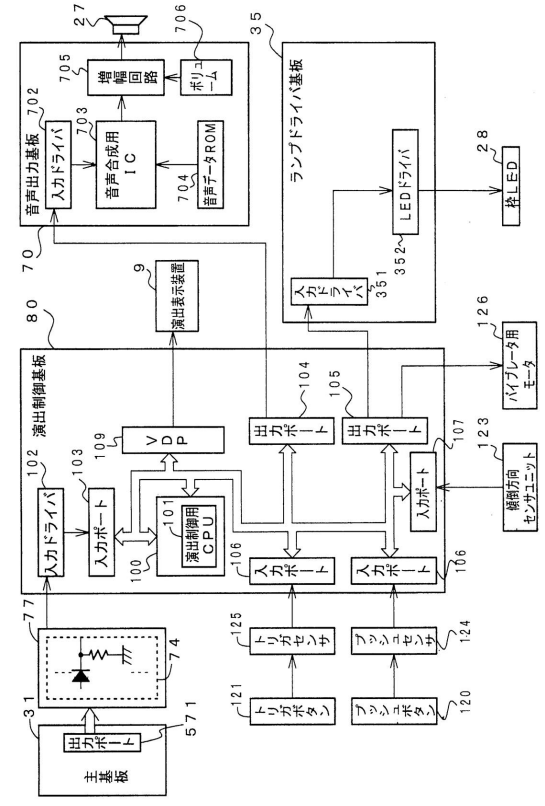
【図３】


201
200
203
204
202
206
205
15
13
14

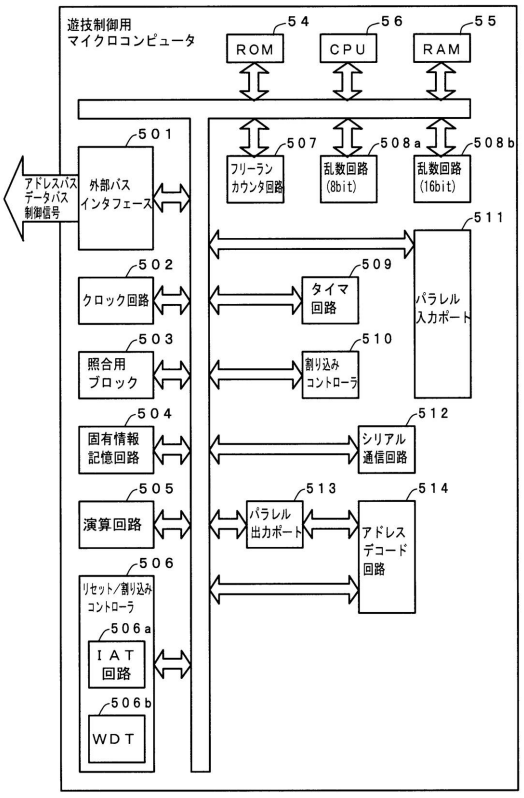
【図４】


ROM
RAM
CPU
8ビット乱数回路
16ビット乱数回路
入力ドライバ回路
制御用クロック生成回路
乱数用クロック生成回路
出力回路
情報出力回路
主基板
ランプドライバ基板
演出制御基板
中継基板
演出表示装置
音声出力基板
払出制御基板
賞球LED
ソレノイド
ターミナル基板
ゲートスイッチ
第1始動口スイッチ
第2始動口スイッチ
カウントスイッチ

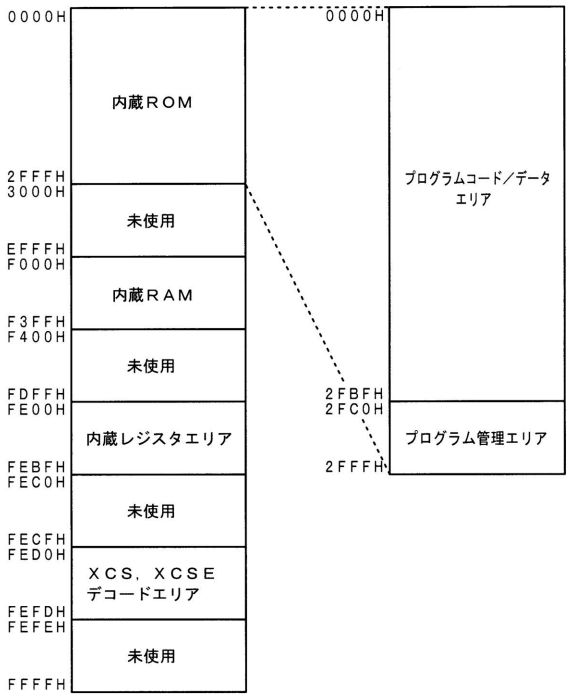
【図５】


音声出力基板
入力ドライバ
音声合成用IC
音声データROM
増幅回路
ボリューム
ランプドライバ基板
LEDドライバ
枠LED
演出表示装置
演出制御基板
VDP
演出制御用CPU
入力ポート
出力ポート
バイブレータ用モータ
傾斜方向センサユニット
トリガセンサ
プッシュセンサ
トリガボタン
プッシュボタン
主基板

【図６】


遊技制御用マイクロコンピュータ
ROM
CPU
RAM
外部バスインタフェース
アドレスバス
データバス
制御信号
フリーランカウンタ回路
乱数回路(8bit)
乱数回路(16bit)
クロック回路
タイマ回路
照合用ブロック
割り込みコントローラ
固有情報記憶回路
シリアル通信回路
演算回路
パラレル出力ポート
アドレスデコード回路
パラレル入力ポート
リセット/割り込みコントローラ
IAT回路
WDT

【図７】



【図８】

プログラム管理エリア（主要部分）

アドレス（H）	内容
2FC0～ 2FC7	ヘッダ（KHDR）
2FD3～ 2FD4	プログラムコードエンドアドレス （KPCE）
2FD5～ 2FD6	プログラムコードスタートアドレス2 （KPCS2）
2FD7～ 2FD8	プログラムコードエンドアドレス2 （KPCE2）
2FD9	リセット設定（KRES）
2FE0	16ビット乱数初期設定1（KRL1）
2FE1	16ビット乱数初期設定2（KRL2）
2FE2	16ビット乱数初期設定3（KRL3）
2FE3	8ビット乱数初期設定1（KRS1）
2FE4	8ビット乱数初期設定2（KRS2）
2FEA	セキュリティ時間設定（KSES）
2FEB	乱数クロック監視設定（KRCS）

【図９】

内蔵レジスタエリア（主要部分）

アドレス（H）	レジスタ名称
FE25	内部情報レジスタ（CIF）
FE5B	RL0ハードラッチ選択レジスタ0（RL0LS0）
FE5C	RL0ハードラッチ選択レジスタ1（RL0LS1）
FE5D	RL1ハードラッチ選択レジスタ（RL1LS）
FE5E	RL2ハードラッチ選択レジスタ（RL2LS）
FE5F	RL3ハードラッチ選択レジスタ（RL3LS）
FE60	RSハードラッチ選択レジスタ0（RSLS0）
FE61	RSハードラッチ選択レジスタ1（RSLS1）
FE64	RL割り込み制御レジスタ0（RLIC0）
FE65	RL割り込み制御レジスタ1（RLIC1）
FE66	RS割り込み制御レジスタ（RSIC）
FE67	RL0最大値設定レジスタ（RL0MX）の下位バイト
FE68	RL0最大値設定レジスタ（RL0MX）の上位バイト
FE69	RL1最大値設定レジスタ（RL1MX）の下位バイト
FE6A	RL1最大値設定レジスタ（RL1MX）の上位バイト
FE6B	RL2最大値設定レジスタ（RL2MX）の下位バイト
FE6C	RL2最大値設定レジスタ（RL2MX）の上位バイト
FE6D	RL3最大値設定レジスタ（RL3MX）の下位バイト
FE6E	RL3最大値設定レジスタ（RL3MX）の上位バイト
FE6F	RS0最大値設定レジスタ（RS0MX）
FE70	RS1最大値設定レジスタ（RS1MX）
FE71	RS2最大値設定レジスタ（RS2MX）
FE72	RS3最大値設定レジスタ（RS3MX）
FE73	乱数列変更レジスタ（RDSC）
FE74	乱数ソフトラッチレジスタ（RDSL）
FE75	乱数ソフトラッチフラグレジスタ（RDSF）

【図１０】

内蔵レジスタエリア（主要部分）

アドレス（H）	レジスタ名称
FE76	RL0ソフトラッチ乱数値レジスタ（RL0SV）の下位バイト
FE77	RL0ソフトラッチ乱数値レジスタ（RL0SV）の上位バイト
FE78	RL1ソフトラッチ乱数値レジスタ（RL1SV）の下位バイト
FE79	RL1ソフトラッチ乱数値レジスタ（RL1SV）の上位バイト
FE7A	RL2ソフトラッチ乱数値レジスタ（RL2SV）の下位バイト
FE7B	RL2ソフトラッチ乱数値レジスタ（RL2SV）の上位バイト
FE7C	RL3ソフトラッチ乱数値レジスタ（RL3SV）の下位バイト
FE7D	RL3ソフトラッチ乱数値レジスタ（RL3SV）の上位バイト
FE7E	RS0ソフトラッチ乱数値レジスタ（RS0SV）
FE7F	RS1ソフトラッチ乱数値レジスタ（RS1SV）
FE80	RS2ソフトラッチ乱数値レジスタ（RS2SV）
FE81	RS3ソフトラッチ乱数値レジスタ（RS3SV）
FE82	RLハードラッチフラグレジスタ0（RLHF0）
FE83	RLハードラッチフラグレジスタ1（RLHF1）
FE84	RSハードラッチフラグレジスタ（RSHF）
FE85	RL0ハードラッチ乱数値レジスタ0（RL0HV0）の下位バイト
FE86	RL0ハードラッチ乱数値レジスタ0（RL0HV0）の上位バイト
FE87	RL0ハードラッチ乱数値レジスタ1（RL0HV1）の下位バイト
FE88	RL0ハードラッチ乱数値レジスタ1（RL0HV1）の上位バイト
FE89	RL0ハードラッチ乱数値レジスタ2（RL0HV2）の下位バイト
FE8A	RL0ハードラッチ乱数値レジスタ2（RL0HV2）の上位バイト
FE8B	RL0ハードラッチ乱数値レジスタ3（RL0HV3）の下位バイト
FE8C	RL0ハードラッチ乱数値レジスタ3（RL0HV3）の上位バイト
FE8D	RL1ハードラッチ乱数値レジスタ0（RL1HV0）の下位バイト
FE8E	RL1ハードラッチ乱数値レジスタ0（RL1HV0）の上位バイト
FE8F	RL1ハードラッチ乱数値レジスタ1（RL1HV1）の下位バイト
FE90	RL1ハードラッチ乱数値レジスタ1（RL1HV1）の上位バイト

【図１１】

内蔵レジスタエリア（主要部分）

アドレス（H）	レジスタ名称
FE91	RL2ハードラッチ乱数値レジスタ0（RL2HV0）の下位バイト
FE92	RL2ハードラッチ乱数値レジスタ0（RL2HV0）の上位バイト
FE93	RL2ハードラッチ乱数値レジスタ1（RL2HV1）の下位バイト
FE94	RL2ハードラッチ乱数値レジスタ1（RL2HV1）の上位バイト
FE95	RL3ハードラッチ乱数値レジスタ0（RL3HV0）の下位バイト
FE96	RL3ハードラッチ乱数値レジスタ0（RL3HV0）の上位バイト
FE97	RL3ハードラッチ乱数値レジスタ1（RL3HV1）の下位バイト
FE98	RL3ハードラッチ乱数値レジスタ1（RL3HV1）の上位バイト
FE99	RS0ハードラッチ乱数値レジスタ（RS0HV）
FE9A	RS1ハードラッチ乱数値レジスタ（RS1HV）
FE9B	RS2ハードラッチ乱数値レジスタ（RS2HV）
FE9C	RS3ハードラッチ乱数値レジスタ（RS3HV）

【図１２】

ヘッダ（KHDR）

アドレス（H）	設定データ（H）		
2FC0	4C	00	左記以外
2FC1	45	00	
2FC2	4D	00	
2FC3	35	00	
2FC4	30	00	
2FC5	41	00	
2FC6	56	00	
2FC7	35	00	
ROM読出防止機能	無効	有効	無効
バス出力マスク機能	有効	有効	無効

【図１３】

（A）プログラムコードエンドアドレス（KPCE）

アドレス	設定データ
2FD3	プログラムコード1の最終アドレスの下位アドレス
2FD4	プログラムコード1の最終アドレスの上位アドレス

（B）プログラムコードスタートアドレス2（KPCS2）

アドレス	設定データ
2FD5	プログラムコード2の先頭アドレスの下位アドレス
2FD6	プログラムコード2の先頭アドレスの上位アドレス

（C）プログラムコードエンドアドレス（KPCE2）

アドレス	設定データ
2FD7	プログラムコード2の最終アドレスの下位アドレス
2FD8	プログラムコード2の最終アドレスの上位アドレス

【図１４】

リセット設定（KRES）

ビット番号	設定内容
7	ウオッチドッグタイマ（WDT）のタイムアウト信号・IAT発生により内部リセットが発生したときの動作を設定 0：ユーザリセット 1：システムリセット
6	ウオッチドッグタイマ（WDT）の起動方法を設定 0：ユーザモード移行で自動起動 1：ソフトウェアで起動
5－4	ウオッチドッグタイマ（WDT）の基準クロックを設定 設定値 00：$2^{15}\times T_{SCLK}$ を選択 設定値 01：$2^{19}\times T_{SCLK}$ を選択 設定値 10：$2^{22}\times T_{SCLK}$ を選択 設定値 11：$2^{25}\times T_{SCLK}$ を選択
3－0	ウオッチドッグタイマ（WDT）のタイムアウト時間を設定 タイムアウト時間＝（ビット5－4で選択した基準クロック）×（設定値） ウオッチドッグタイマ（WDT）を使用しない場合は0000Hを設定

設定値とタイムアウト時間の一例

設定値	ビット5－4	設定周期	タイムアウト時間 SCLK＝10.0MHz	タイムアウト時間 SCLK＝12.0MHz
0000	—	ウオッチドッグタイマを禁止（使用しない）		
1000	00	$2^{15}\times T_{SCLK}\times 8$	約52．4ms	約43．7ms
	01	$2^{19}\times T_{SCLK}\times 8$	約419．4ms	約349．5ms
	10	$2^{22}\times T_{SCLK}\times 8$	約3．36s	約2．80s
	11	$2^{25}\times T_{SCLK}\times 8$	約23．49s	約22．37s
1111	00	$2^{15}\times T_{SCLK}\times 15$	約98．3ms	約81．9ms
	01	$2^{19}\times T_{SCLK}\times 15$	約786．4ms	約655．4ms
	10	$2^{22}\times T_{SCLK}\times 15$	約6．29s	約5．24s
	11	$2^{25}\times T_{SCLK}\times 15$	約50．33s	約41．94s

【図１５】

１６ビット乱数初期設定１（ＫＲＬ１）

ビット番号	設定内容
7	１６ビット乱数ｃｈ１の起動方法を選択 ０：最大値設定（ソフトウェア）で起動 １：ユーザモード移行で自動起動
6	１６ビット乱数ｃｈ１の更新クロックを設定 ０：内部システムクロック １：外部（ＲＣＫ端子に入力された）クロックの２分周
5－4	１６ビット乱数ｃｈ１の乱数列変更方法を設定 設定値：設定内容 ００：変更しない ０１：ソフトウェアで変更 １０：２周目から自動で変更 １１：１周目から自動で変更
3	１６ビット乱数ｃｈ０の起動方法を選択 ０：最大値設定（ソフトウェア）で起動 １：ユーザモード移行で自動起動
2	１６ビット乱数ｃｈ０の更新クロックを設定 ０：内部システムクロック １：外部（ＲＣＫ端子に入力された）クロックの２分周
1－0	１６ビット乱数ｃｈ０の乱数列変更方法を設定 設定値：設定内容 ００：変更しない ０１：ソフトウェアで変更 １０：２周目から自動で変更 １１：１周目から自動で変更

【図１６】

１６ビット乱数初期設定２（ＫＲＬ２）

ビット番号	設定内容
7	１６ビット乱数ｃｈ３の起動方法を選択 ０：最大値設定（ソフトウェア）で起動 １：ユーザモード移行で自動起動
6	１６ビット乱数ｃｈ３の更新クロックを設定 ０：内部システムクロック １：外部（ＲＣＫ端子に入力された）クロックの２分周
5－4	１６ビット乱数ｃｈ３の乱数列変更方法を設定 設定値：設定内容 ００：変更しない ０１：ソフトウェアで変更 １０：２周目から自動で変更 １１：１周目から自動で変更
3	１６ビット乱数ｃｈ２の起動方法を選択 ０：最大値設定（ソフトウェア）で起動 １：ユーザモード移行で自動起動
2	１６ビット乱数ｃｈ２の更新クロックを設定 ０：内部システムクロック １：外部（ＲＣＫ端子に入力された）クロックの２分周
1－0	１６ビット乱数ｃｈ２の乱数列変更方法を設定 設定値：設定内容 ００：変更しない ０１：ソフトウェアで変更 １０：２周目から自動で変更 １１：１周目から自動で変更

【図１７】

１６ビット乱数初期設定３（ＫＲＬ３）

ビット番号	設定内容
7	１６ビット乱数ｃｈ３のスタート値を設定 ０：０００１Ｈ １：ＩＤナンバを基にした値
6	１６ビット乱数ｃｈ３のスタート値を、システムリセット毎に変更するかを設定 ０：変更しない １：変更する
5	１６ビット乱数ｃｈ２のスタート値を設定 ０：０００１Ｈ １：ＩＤナンバを基にした値
4	１６ビット乱数ｃｈ２のスタート値を、システムリセット毎に変更するかを設定 ０：変更しない １：変更する
3	１６ビット乱数ｃｈ１のスタート値を設定 ０：０００１Ｈ １：ＩＤナンバを基にした値
2	１６ビット乱数ｃｈ１のスタート値を、システムリセット毎に変更するかを設定 ０：変更しない １：変更する
1	１６ビット乱数ｃｈ０のスタート値を設定 ０：０００１Ｈ １：ＩＤナンバを基にした値
0	１６ビット乱数ｃｈ０のスタート値を、システムリセット毎に変更するかを設定 ０：変更しない １：変更する

【図１８】

８ビット乱数初期設定１（ＫＲＳ１）

ビット番号	設定内容
7	８ビット乱数ｃｈ１の起動方法を選択 ０：最大値設定（ソフトウェア）で起動 １：ユーザモード移行で自動起動
6	８ビット乱数ｃｈ１の更新クロックを設定 ０：内部システムクロック １：外部（ＲＣＫ端子に入力された）クロックの２分周
5－4	８ビット乱数ｃｈ１の乱数列変更方法を設定 設定値：設定内容 ００：変更しない ０１：ソフトウェアで変更 １０：２周目から自動で変更 １１：１周目から自動で変更
3	８ビット乱数ｃｈ０の起動方法を選択 ０：最大値設定（ソフトウェア）で起動 １：ユーザモード移行で自動起動
2	８ビット乱数ｃｈ０の更新クロックを設定 ０：内部システムクロック １：外部（ＲＣＫ端子に入力された）クロックの２分周
1－0	８ビット乱数ｃｈ０の乱数列変更方法を設定 設定値：設定内容 ００：変更しない ０１：ソフトウェアで変更 １０：２周目から自動で変更 １１：１周目から自動で変更

【図１９】

８ビット乱数初期設定２（ＫＲＳ２）

ビット番号	設定内容
7	８ビット乱数ｃｈ３の起動方法を選択 ０：最大値設定（ソフトウェア）で起動 １：ユーザモード移行で自動起動
6	８ビット乱数ｃｈ３の更新クロックを設定 ０：内部システムクロック １：外部（ＲＣＫ端子に入力された）クロックの２分周
5－4	８ビット乱数ｃｈ３の乱数列変更方法を設定 設定値：設定内容 ００：変更しない ０１：ソフトウェアで変更 １０：２周目から自動で変更 １１：１周目から自動で変更
3	８ビット乱数ｃｈ２の起動方法を選択 ０：最大値設定（ソフトウェア）で起動 １：ユーザモード移行で自動起動
2	８ビット乱数ｃｈ２の更新クロックを設定 ０：内部システムクロック １：外部（ＲＣＫ端子に入力された）クロックの２分周
1－0	８ビット乱数ｃｈ２の乱数列変更方法を設定 設定値：設定内容 ００：変更しない ０１：ソフトウェアで変更 １０：２周目から自動で変更 １１：１周目から自動で変更

【図２０】

セキュリティ時間設定（ＫＳＥＳ）

ビット番号	設定内容
7－6	セキュリティモード時間をランダムに延長する時間を設定 設定値／設定周期／延長される時間の範囲（ＳＣＬＫ＝10.0MHz／ＳＣＬＫ＝12.0MHz） ００／延長しない ０１／ショートモード／0～0.816ms／0～0.51ms １０／ミドルモード／0～26.112ms／0～16.32ms １１／ロングモード／0～835.584ms／0～522.24ms
5	セキュリティモード時間を固定延長する時間の基準クロックを設定 設定値／選択基準クロック 0／2^{22} × T_{SCLK} を選択 1／2^{24} × T_{SCLK} を選択
4－0	セキュリティモード時間を固定で延長する時間を設定 固定延長時間＝（ビット５で選択された基準クロック）×（設定値） 固定延長をしない場合は００００００Ｈを設定 設定値と延長時間の一例 設定値／ビット5／設定内容／固定延長時間（ＳＣＬＫ＝10.0MHz／ＳＣＬＫ＝12.0MHz） 00001／0／2^{22} × T_{SCLK} ×１／約０．４２ｓ／約０．３５ｓ 1／2^{24} × T_{SCLK} ×１／約１．６８ｓ／約１．４０ｓ 01000／0／2^{22} × T_{SCLK} ×８／約３．３６ｓ／約２．８０ｓ 1／2^{24} × T_{SCLK} ×８／約１３．４２ｓ／約１１．１８ｓ 10000／0／2^{22} × T_{SCLK} ×１６／約６．７１ｓ／約５．５９ｓ 1／2^{24} × T_{SCLK} ×１６／約２６．８４ｓ／約２２．３７ｓ 11111／0／2^{22} × T_{SCLK} ×３１／約１３．００ｓ／約１０．８４ｓ 1／2^{24} × T_{SCLK} ×３１／約５２．０１ｓ／約４３．３４ｓ

【図２１】

乱数クロック監視設定（ＫＲＣＳ）

ビット番号	設定内容
7－2	固定ビット（必ず全ビット０を設定）
1－0	乱数を更新するためのクロックを、外部（ＲＣＫ端子に入力された）クロックを選択した場合に、その入力クロックの異常を検知する周波数を設定。異常を検知した場合は、内部情報レジスタ（ＣＩＦ）のビット３にセットされる。 設定値：設定内容（異常検知周波数） ００：ＲＣＫ＜ＳＣＬＫ ０１：ＲＣＫ＜ＳＣＬＫ／2^2 １０：ＲＣＫ＜ＳＣＬＫ／2^2 １１：ＲＣＫ＜ＳＣＬＫ／2^3

【図２２】

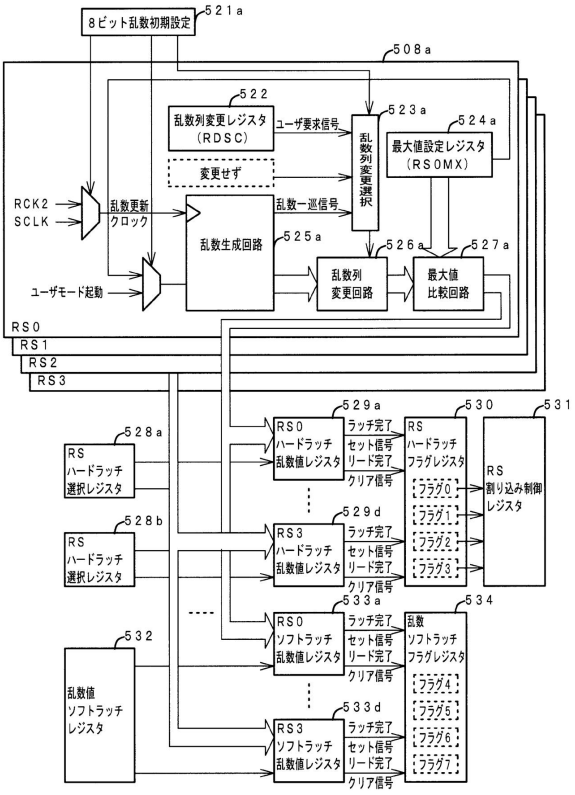
（Ａ）

ビット番号	7	6	5	4	3	2	1	0
ＣＩＦ	RL3ER	RL2ER	RL1ER	RL0ER	RCER	SRSF	WDTF	IATF

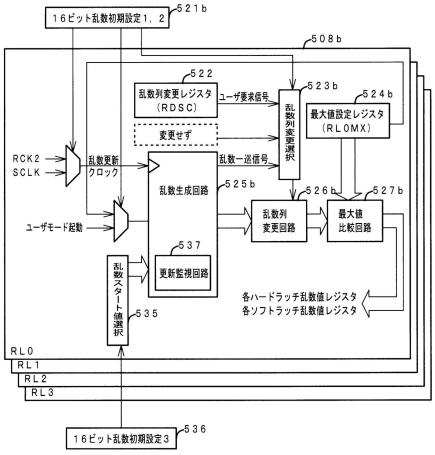
（Ｂ）内部情報レジスタ（ＣＩＦ）

ビット番号	ビット名	意味
7	ＲＬ３ＥＲ	１６ビット乱数ＲＬ３の更新状態の異常を示すビット ０：更新異常なし（初期値） １：更新異常あり
6	ＲＬ２ＥＲ	１６ビット乱数ＲＬ２の更新状態の異常を示すビット ０：更新異常なし（初期値） １：更新異常あり
5	ＲＬ１ＥＲ	１６ビット乱数ＲＬ１の更新状態の異常を示すビット ０：更新異常なし（初期値） １：更新異常あり
4	ＲＬ０ＥＲ	１６ビット乱数ＲＬ０の更新状態の異常を示すビット ０：更新異常なし（初期値） １：更新異常あり
3	ＲＣＥＲ	乱数更新用の外部クロック入力の周波数異常を示すビット ０：周波数異常なし（初期値） １：周波数異常あり
2	ＳＲＳＦ	直前に発生したリセット要因がシステムリセットであることを示すビット ０：システムリセット未発生 １：システムリセット発生（初期値）
1	ＷＤＴＦ	直前に発生したリセット要因がＷＤＴタイムアウト信号によるユーザリセットであることを示すビット ０：ＷＤＴタイムアウト信号によるユーザリセット未発生（初期値） １：ＷＤＴタイムアウト信号によるユーザリセット発生
0	ＩＡＴＦ	直前に発生したリセット要因がＩＡＴ発生信号によるユーザリセットであることを示すビット ０：ＩＡＴ発生信号によるユーザリセット未発生（初期値） １：ＩＡＴ発生信号によるユーザリセット発生

【図２３】



【図２４】



【図２５】

（A）

ビット番号	7	6	5	4	3	2	1	0
RL0LS0	RL01RF	RL01LS2	RL01LS1	RL01LS0	RL00RF	RL00LS2	RL00LS1	RL00LS0

（B）　ＲＬ０ハードラッチ選択レジスタ０（ＲＬ０ＬＳ０）

ビット番号	ビット名	意味
7	RL01RF	ＲＬ０ハードラッチ乱数値レジスタ１(RL0HV1)に、外部端子入力により、１６ビット乱数ＲＬ０の値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
６－４	RL01LS0～RL01LS2	ＲＬ０ハードラッチ乱数値レジスタ１(RL0HV1)に、どの外部端子入力により、１６ビット乱数ＲＬ０の値を取り込むかを設定するためのビット 設定値／選択される外部端子／設定値／選択される外部端子 ０００／ＰＩ０端子（初期値）／１００／ＰＩ４端子 ００１／ＰＩ１端子／１０１／ＰＩ５／ＸＩＮＴ端子 ０１０／ＰＩ２端子／１１０／無効 ０１１／ＰＩ３端子／１１１
３	RL00RF	ＲＬ０ハードラッチ乱数値レジスタ０(RL0HV0)に、外部端子入力により、１６ビット乱数ＲＬ０の値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
２－０	RL00LS0～RL00LS2	ＲＬ０ハードラッチ乱数値レジスタ０(RL0HV0)に、どの外部端子入力により、１６ビット乱数ＲＬ０の値を取り込むかを設定するためのビット 設定値／選択される外部端子／設定値／選択される外部端子 ０００／ＰＩ０端子（初期値）／１００／ＰＩ４端子 ００１／ＰＩ１端子／１０１／ＰＩ５／ＸＩＮＴ端子 ０１０／ＰＩ２端子／１１０／無効 ０１１／ＰＩ３端子／１１１

【図２６】

（A）

ビット番号	7	6	5	4	3	2	1	0
RL0LS1	RL03RF	RL03LS2	RL03LS1	RL03LS0	RL02RF	RL02LS2	RL02LS1	RL02LS0

（B）　ＲＬ０ハードラッチ選択レジスタ１（ＲＬ０ＬＳ１）

ビット番号	ビット名	意味
7	RL03RF	ＲＬ０ハードラッチ乱数値レジスタ３(RL0HV3)に、外部端子入力により、１６ビット乱数ＲＬ０の値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
６－４	RL03LS0～RL03LS2	ＲＬ０ハードラッチ乱数値レジスタ３(RL0HV3)に、どの外部端子入力により、１６ビット乱数ＲＬ０の値を取り込むかを設定するためのビット 設定値／選択される外部端子／設定値／選択される外部端子 ０００／ＰＩ０端子（初期値）／１００／ＰＩ４端子 ００１／ＰＩ１端子／１０１／ＰＩ５／ＸＩＮＴ端子 ０１０／ＰＩ２端子／１１０／無効 ０１１／ＰＩ３端子／１１１
３	RL02RF	ＲＬ０ハードラッチ乱数値レジスタ２(RL0HV2)に、外部端子入力により、１６ビット乱数ＲＬ０の値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
２－０	RL02LS0～RL02LS2	ＲＬ０ハードラッチ乱数値レジスタ２(RL0HV2)に、どの外部端子入力により、１６ビット乱数ＲＬ０の値を取り込むかを設定するためのビット 設定値／選択される外部端子／設定値／選択される外部端子 ０００／ＰＩ０端子（初期値）／１００／ＰＩ４端子 ００１／ＰＩ１端子／１０１／ＰＩ５／ＸＩＮＴ端子 ０１０／ＰＩ２端子／１１０／無効 ０１１／ＰＩ３端子／１１１

【図２７】

（A）

ビット番号	7	6	5	4	3	2	1	0
RLnLS	RLn1RF	RLn1LS2	RLn1LS1	RLn1LS0	RLn0RF	RLn0LS2	RLn0LS1	RLn0LS0

（B）　ＲＬｎハードラッチ選択レジスタ（ＲＬｎＬＳ）（ｎ＝１～３）

ビット番号	ビット名	意味
7	RLn1RF	ＲＬｎハードラッチ乱数値レジスタ１(RLnHV1)に、外部端子入力により、１６ビット乱数ＲＬｎの値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
６－４	RLn1LS0～ RLn1LS2	ＲＬｎハードラッチ乱数値レジスタ１(RLnHV1)に、どの外部端子入力により、１６ビット乱数ＲＬｎの値を取り込むかを設定するためのビット 設定値：選択される外部端子：設定値：選択される外部端子 ０００：ＰＩ０端子（初期値）：１００：ＰＩ４端子 ００１：ＰＩ１端子：１０１：ＰＩ５／ＸＩＮＴ端子 ０１０：ＰＩ２端子：１１０：無効 ０１１：ＰＩ３端子：１１１：
３	RLn0RF	ＲＬｎハードラッチ乱数値レジスタ０(RLnHV0)に、外部端子入力により、１６ビット乱数ＲＬｎの値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
２－０	RLn0LS0～ RLn0LS2	ＲＬｎハードラッチ乱数値レジスタ０(RLnHV0)に、どの外部端子入力により、１６ビット乱数ＲＬｎの値を取り込むかを設定するためのビット 設定値：選択される外部端子：設定値：選択される外部端子 ０００：ＰＩ０端子（初期値）：１００：ＰＩ４端子 ００１：ＰＩ１端子：１０１：ＰＩ５／ＸＩＮＴ端子 ０１０：ＰＩ２端子：１１０：無効 ０１１：ＰＩ３端子：１１１：

【図２８】

（A）

ビット番号	7	6	5	4	3	2	1	0
RSLS0	RS1RF	RS1LS2	RS1LS1	RS1LS0	RS0RF	RS0LS2	RS0LS1	RS0LS0

（B）　ＲＳハードラッチ選択レジスタ０（ＲＳＬＳ０）

ビット番号	ビット名	意味
7	RS1RF	ＲＳ１ハードラッチ乱数値レジスタ(RS1HV)に、外部端子入力により、８ビット乱数ＲＳ１の値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
６－４	RS1LS0～ RS1LS2	ＲＳ１ハードラッチ乱数値レジスタ(RS1HV)に、どの外部端子入力により、８ビット乱数ＲＳ１の値を取り込むかを設定するためのビット 設定値：選択される外部端子：設定値：選択される外部端子 ０００：ＰＩ０端子（初期値）：１００：ＰＩ４端子 ００１：ＰＩ１端子：１０１：ＰＩ５／ＸＩＮＴ端子 ０１０：ＰＩ２端子：１１０：無効 ０１１：ＰＩ３端子：１１１：
３	RS0RF	ＲＳ０ハードラッチ乱数値レジスタ(RS0HV)に、外部端子入力により、８ビット乱数ＲＳ０の値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
２－０	RS0LS0～ RS0LS2	ＲＳ０ハードラッチ乱数値レジスタ(RS0HV)に、どの外部端子入力により、８ビット乱数ＲＳ０の値を取り込むかを設定するためのビット 設定値：選択される外部端子：設定値：選択される外部端子 ０００：ＰＩ０端子（初期値）：１００：ＰＩ４端子 ００１：ＰＩ１端子：１０１：ＰＩ５／ＸＩＮＴ端子 ０１０：ＰＩ２端子：１１０：無効 ０１１：ＰＩ３端子：１１１：

【図２９】

（A）

ビット番号	7	6	5	4	3	2	1	0
RSLS1	RS3RF	RS3LS2	RS3LS1	RS3LS0	RS2RF	RS2LS2	RS2LS1	RS2LS0

（B）　ＲＳハードラッチ選択レジスタ１（ＲＳＬＳ１）

ビット番号	ビット名	意味
7	RS3RF	ＲＳ３ハードラッチ乱数値レジスタ(RS3HV)に、外部端子入力により、８ビット乱数ＲＳ３の値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
６－４	RS3LS0～ RS3LS2	ＲＳ３ハードラッチ乱数値レジスタ(RS3HV)に、どの外部端子入力により、８ビット乱数ＲＳ３の値を取り込むかを設定するためのビット 設定値：選択される外部端子：設定値：選択される外部端子 ０００：ＰＩ０端子（初期値）：１００：ＰＩ４端子 ００１：ＰＩ１端子：１０１：ＰＩ５／ＸＩＮＴ端子 ０１０：ＰＩ２端子：１１０：無効 ０１１：ＰＩ３端子：１１１：
３	RS2RF	ＲＳ２ハードラッチ乱数値レジスタ(RS2HV)に、外部端子入力により、８ビット乱数ＲＳ２の値を取り込む際の条件を設定するためのビット ０：値を読み込まないと、次の値をラッチしない（初期値） １：値を読み込まなくても、次の値をラッチする
２－０	RS2LS0～ RS2LS2	ＲＳ２ハードラッチ乱数値レジスタ(RS2HV)に、どの外部端子入力により、８ビット乱数ＲＳ２の値を取り込むかを設定するためのビット 設定値：選択される外部端子：設定値：選択される外部端子 ０００：ＰＩ０端子（初期値）：１００：ＰＩ４端子 ００１：ＰＩ１端子：１０１：ＰＩ５／ＸＩＮＴ端子 ０１０：ＰＩ２端子：１１０：無効 ０１１：ＰＩ３端子：１１１：

【図３０】

（A）

ビット番号	7	6	5	4	3	2	1	0
RLIC0	0	0	RL11IE	RL10IE	RL03IE	RL02IE	RL01IE	RL00IE

（B）　ＲＬ割り込み制御レジスタ０（ＲＬＩＣ０）

ビット番号	ビット名	意味
5	RL11IE	ＲＬ１ハードラッチ乱数値レジスタ１(RL1HV1)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 ０：割り込み禁止（初期値） １：割り込み許可
4	RL10IE	ＲＬ１ハードラッチ乱数値レジスタ０(RL1HV0)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 ０：割り込み禁止（初期値） １：割り込み許可
3	RL03IE	ＲＬ０ハードラッチ乱数値レジスタ３(RL0HV3)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 ０：割り込み禁止（初期値） １：割り込み許可
2	RL02IE	ＲＬ０ハードラッチ乱数値レジスタ２(RL0HV2)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 ０：割り込み禁止（初期値） １：割り込み許可
1	RL01IE	ＲＬ０ハードラッチ乱数値レジスタ１(RL0HV1)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 ０：割り込み禁止（初期値） １：割り込み許可
0	RL00IE	ＲＬ０ハードラッチ乱数値レジスタ０(RL0HV0)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 ０：割り込み禁止（初期値） １：割り込み許可

【図３１】

(A)

ビット番号	7	6	5	4	3	2	1	0
RLIC1	0	0	RL31IE	RL30IE	0	0	RL21IE	RL20IE

(B) ＲＬ割り込み制御レジスタ１（ＲＬＩＣ１）

ビット番号	ビット名	意味
5	ＲＬ３１ＩＥ	ＲＬ３ハードラッチ乱数値レジスタ１(RL3HV1)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
4	ＲＬ３０ＩＥ	ＲＬ３ハードラッチ乱数値レジスタ０(RL3HV0)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
1	ＲＬ２１ＩＥ	ＲＬ２ハードラッチ乱数値レジスタ１(RL2HV1)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
0	ＲＬ２０ＩＥ	ＲＬ２ハードラッチ乱数値レジスタ０(RL2HV0)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可

【図３２】

(A)

ビット番号	7	6	5	4	3	2	1	0
RSIC	FR3IE	FR2IE	FR1IE	FR0IE	RS3IE	RS2IE	RS1IE	RS0IE

(B) ＲＳ割り込み制御レジスタ（ＲＳＩＣ）

ビット番号	ビット名	意味
7	ＦＲ３ＩＥ	ＦＲＣ３ハードラッチレジスタ(FR3HV)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
6	ＦＲ２ＩＥ	ＦＲＣ２ハードラッチレジスタ(FR2HV)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
5	ＦＲ１ＩＥ	ＦＲＣ１ハードラッチレジスタ(FR1HV)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
4	ＦＲ０ＩＥ	ＦＲＣ０ハードラッチレジスタ(FR0HV)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
3	ＲＳ３ＩＥ	ＲＳ３ハードラッチ乱数値レジスタ(RS3HV)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
2	ＲＳ２ＩＥ	ＲＳ２ハードラッチ乱数値レジスタ(RS2HV)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
1	ＲＳ１ＩＥ	ＲＳ１ハードラッチ乱数値レジスタ(RS1HV)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可
0	ＲＳ０ＩＥ	ＲＳ０ハードラッチ乱数値レジスタ(RS0HV)に、乱数値が取り込まれたことを要因とする割り込みの禁止／許可を設定 0：割り込み禁止（初期値） 1：割り込み許可

【図３３】

(A)

ビット番号	7	6	5	4	3	2	1	0
RLnMX(L)	RLnMX7	RLnMX6	RLnMX5	RLnMX4	RLnMX3	RLnMX2	RLnMX1	RLnMX0
RLnMX(H)	RLnMX15	RLnMX14	RLnMX13	RLnMX12	RLnMX11	RLnMX10	RLnMX9	RLnMX8

(B) ＲＬｎ最大値設定レジスタ（ＲＬｎＭＸ）（ｎ＝０～３）

ビット番号	ビット名	意味
１５－０	RLnMX15－RLnMX0	１６ビット乱数ＲＬｎの最大値設定

【図３４】

(A)

ビット番号	7	6	5	4	3	2	1	0
RSnMX	RSnMX7	RSnMX6	RSnMX5	RSnMX4	RSnMX3	RSnMX2	RSnMX1	RSnMX0

(B) ＲＳｎ最大値設定レジスタ（ＲＳｎＭＸ）（ｎ＝０～３）

ビット番号	ビット名	意味
７－０	RSnMX7－RSnMX0	８ビット乱数ＲＳｎの最大値設定

【図３５】

(A)

ビット番号	7	6	5	4	3	2	1	0
RDSC	RS3SC	RS2SC	RS1SC	RS0SC	RL3SC	RL2SC	RL1SC	RL0SC

(B) 乱数列変更レジスタ（ＲＤＳＣ）

ビット番号	ビット名	意味
7	RS3SC	８ビット乱数ＲＳ３の乱数列変更要求ビット 0：乱数列を変更しない（初期値） 1：乱数列を変更する
6	RS2SC	８ビット乱数ＲＳ２の乱数列変更要求ビット 0：乱数列を変更しない（初期値） 1：乱数列を変更する
5	RS1SC	８ビット乱数ＲＳ１の乱数列変更要求ビット 0：乱数列を変更しない（初期値） 1：乱数列を変更する
4	RS0SC	８ビット乱数ＲＳ０の乱数列変更要求ビット 0：乱数列を変更しない（初期値） 1：乱数列を変更する
3	RL3SC	１６ビット乱数ＲＬ３の乱数列変更要求ビット 0：乱数列を変更しない（初期値） 1：乱数列を変更する
2	RL2SC	１６ビット乱数ＲＬ２の乱数列変更要求ビット 0：乱数列を変更しない（初期値） 1：乱数列を変更する
1	RL1SC	１６ビット乱数ＲＬ１の乱数列変更要求ビット 0：乱数列を変更しない（初期値） 1：乱数列を変更する
0	RL0SC	１６ビット乱数ＲＬ０の乱数列変更要求ビット 0：乱数列を変更しない（初期値） 1：乱数列を変更する

【図３６】

(A)

ビット番号	7	6	5	4	3	2	1	0
RDSL	RS3SL	RS2SL	RS1SL	RS0SL	RL3SL	RL2SL	RL1SL	RL0SL

(B) 乱数ソフトラッチレジスタ（ＲＤＳＬ）

ビット番号	ビット名	意味
7	ＲＳ３ＳＬ	８ビット乱数ＲＳ３の乱数値を、ＲＳ３ソフトラッチ乱数値レジスタ(RS3SV)に取り込むためのビット 0：乱数値を取り込まない（初期値） 1：乱数値取り込み
6	ＲＳ２ＳＬ	８ビット乱数ＲＳ２の乱数値を、ＲＳ２ソフトラッチ乱数値レジスタ(RS2SV)に取り込むためのビット 0：乱数値を取り込まない（初期値） 1：乱数値取り込み
5	ＲＳ１ＳＬ	８ビット乱数ＲＳ１の乱数値を、ＲＳ１ソフトラッチ乱数値レジスタ(RS1SV)に取り込むためのビット 0：乱数値を取り込まない（初期値） 1：乱数値取り込み
4	ＲＳ０ＳＬ	８ビット乱数ＲＳ０の乱数値を、ＲＳ０ソフトラッチ乱数値レジスタ(RS0SV)に取り込むためのビット 0：乱数値を取り込まない（初期値） 1：乱数値取り込み
3	ＲＬ３ＳＬ	１６ビット乱数ＲＬ３の乱数値を、ＲＬ３ソフトラッチ乱数値レジスタ(RL3SV)に取り込むためのビット 0：乱数値を取り込まない（初期値） 1：乱数値取り込み
2	ＲＬ２ＳＬ	１６ビット乱数ＲＬ２の乱数値を、ＲＬ２ソフトラッチ乱数値レジスタ(RL2SV)に取り込むためのビット 0：乱数値を取り込まない（初期値） 1：乱数値取り込み
1	ＲＬ１ＳＬ	１６ビット乱数ＲＬ１の乱数値を、ＲＬ１ソフトラッチ乱数値レジスタ(RL1SV)に取り込むためのビット 0：乱数値を取り込まない（初期値） 1：乱数値取り込み
0	ＲＬ０ＳＬ	１６ビット乱数ＲＬ０の乱数値を、ＲＬ０ソフトラッチ乱数値レジスタ(RL0SV)に取り込むためのビット 0：乱数値を取り込まない（初期値） 1：乱数値取り込み

【図３７】

(A)

ビット番号	7	6	5	4	3	2	1	0
RDSF	RS3SF	RS2SF	RS1SF	RS0SF	RL3SF	RL2SF	RL1SF	RL0SF

(B) 乱数ソフトラッチフラグレジスタ（ＲＤＳＦ）

ビット番号	ビット名	意味
7	ＲＳ３ＳＦ	ＲＳ３ソフトラッチ乱数値レジスタ(RS3SV)に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
6	ＲＳ２ＳＦ	ＲＳ２ソフトラッチ乱数値レジスタ(RS2SV)に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
5	ＲＳ１ＳＦ	ＲＳ１ソフトラッチ乱数値レジスタ(RS1SV)に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
4	ＲＳ０ＳＦ	ＲＳ０ソフトラッチ乱数値レジスタ(RS0SV)に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
3	ＲＬ３ＳＦ	ＲＬ３ソフトラッチ乱数値レジスタ(RL3SV)に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
2	ＲＬ２ＳＦ	ＲＬ２ソフトラッチ乱数値レジスタ(RL2SV)に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
1	ＲＬ１ＳＦ	ＲＬ１ソフトラッチ乱数値レジスタ(RL1SV)に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
0	ＲＬ０ＳＦ	ＲＬ０ソフトラッチ乱数値レジスタ(RL0SV)に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み

【図３８】

(A)

ビット番号	7	6	5	4	3	2	1	0
RLnSV(L)	RLnSV7	RLnSV6	RLnSV5	RLnSV4	RLnSV3	RLnSV2	RLnSV1	RLnSV0
RLnSV(H)	RLnSV15	RLnSV14	RLnSV13	RLnSV12	RLnSV11	RLnSV10	RLnSV9	RLnSV8

(B) ＲＬｎソフトラッチ乱数値レジスタ（ＲＬｎＳＶ）（ｎ＝０～３）

ビット番号	ビット名	意味
１５－０	RLnSV15－RLnSV0	乱数ソフトラッチレジスタ(RDSL)により取り込まれた１６ビット乱数ＲＬｎの値 乱数値が取り込まれると、乱数ソフトラッチフラグレジスタ(RDSF)の当該ビットがセット（１）される

【図３９】

(A)

ビット番号	7	6	5	4	3	2	1	0
RSnSV	RSnSV7	RSnSV6	RSnSV5	RSnSV4	RSnSV3	RSnSV2	RSnSV1	RSnSV0

(B) ＲＳｎソフトラッチ乱数値レジスタ（ＲＳｎＳＶ）（ｎ＝０～３）

ビット番号	ビット名	意味
７－０	RSnSV7－RSnSV0	乱数ソフトラッチレジスタ(RDSL)により取り込まれた８ビット乱数ＲＳｎの値 乱数値が取り込まれると、乱数ソフトラッチフラグレジスタ(RDSF)の当該ビットがセット（１）される

【図４０】

(A)

ビット番号	7	6	5	4	3	2	1	0
RLHF0	0	0	RL11HF	RL10HF	RL03HF	RL02HF	RL01HF	RL00HF

(B) ＲＬハードラッチフラグレジスタ０（ＲＬＨＦ０）

ビット番号	ビット名	意味
5	ＲＬ１１ＨＦ	ＲＬ１ハードラッチ乱数値レジスタ１に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
4	ＲＬ１０ＨＦ	ＲＬ１ハードラッチ乱数値レジスタ０に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
3	ＲＬ０３ＨＦ	ＲＬ０ハードラッチ乱数値レジスタ３に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
2	ＲＬ０２ＨＦ	ＲＬ０ハードラッチ乱数値レジスタ２に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
1	ＲＬ０１ＨＦ	ＲＬ０ハードラッチ乱数値レジスタ１に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み
0	ＲＬ００ＨＦ	ＲＬ０ハードラッチ乱数値レジスタ０に、乱数値が取り込まれたことを示すビット 0：乱数値取り込まれていない（初期値） 1：乱数値取り込み済み

【図４１】

（A）

ビット番号	7	6	5	4	3	2	1	0
RLHF1	0	0	RL31HF	RL30HF	0	0	RL21HF	RL20HF

（B） ＲＬハードラッチフラグレジスタ１（ＲＬＨＦ１）

ビット番号	ビット名	意味
5	ＲＬ３１ＨＦ	ＲＬ３ハードラッチ乱数値レジスタ１に、乱数値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
4	ＲＬ３０ＨＦ	ＲＬ３ハードラッチ乱数値レジスタ０に、乱数値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
1	ＲＬ２１ＨＦ	ＲＬ２ハードラッチ乱数値レジスタ１に、乱数値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
0	ＲＬ２０ＨＦ	ＲＬ２ハードラッチ乱数値レジスタ０に、乱数値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み

【図４２】

（A）

ビット番号	7	6	5	4	3	2	1	0
RSHF	FR3HF	FR2HF	FR1HF	FR0HF	RS3HF	RS2HF	RS1HF	RS0HF

（B） ＲＳハードラッチフラグレジスタ（ＲＳＨＦ）

ビット番号	ビット名	意味
7	ＦＲ３ＨＦ	ＦＲＣ３ハードラッチレジスタ(FR3HV)に、カウンタ値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
6	ＦＲ２ＨＦ	ＦＲＣ２ハードラッチレジスタ(FR2HV)に、カウンタ値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
5	ＦＲ１ＨＦ	ＦＲＣ１ハードラッチレジスタ(FR1HV)に、カウンタ値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
4	ＦＲ０ＨＦ	ＦＲＣ０ハードラッチレジスタ(FR0HV)に、カウンタ値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
3	ＲＳ３ＨＦ	ＲＳ３ハードラッチレジスタ(RS3HV)に、カウンタ値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
2	ＲＳ２ＨＦ	ＲＳ２ハードラッチレジスタ(RS2HV)に、カウンタ値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
1	ＲＳ１ＨＦ	ＲＳ１ハードラッチレジスタ(RS1HV)に、カウンタ値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み
0	ＲＳ０ＨＦ	ＲＳ０ハードラッチレジスタ(RS0HV)に、カウンタ値が取り込まれたことを示すビット ０：乱数値取り込まれていない（初期値） １：乱数値取り込み済み

【図４３】

（A）

ビット番号	7	6	5	4	3	2	1	0
RL0mHV (L)	RL0mHV7	RL0mHV6	RL0mHV5	RL0mHV4	RL0mHV3	RL0mHV2	RL0mHV1	RL0mHV0
RL0mHV (H)	RL0mHV15	RL0mHV14	RL0mHV13	RL0mHV12	RL0mHV11	RL0mHV10	RL0mHV9	RL0mHV8

（B） ＲＬ０ハードラッチ乱数値レジスタｍ（ＲＬ０ｍＨＶ）（ｍ＝０～３）

ビット番号	ビット名	意味
１５－０	RL0mHV15-RL0mHV0	外部端子入力により取り込まれた１６ビット乱数ＲＬ０の値乱数値が取り込まれると、ＲＬハードラッチフラグレジスタ０(RLHF0)の当該ビットがセット（１）される

【図４４】

（A）

ビット番号	7	6	5	4	3	2	1	0
RL1mHV (L)	RL1mHV7	RL1mHV6	RL1mHV5	RL1mHV4	RL1mHV3	RL1mHV2	RL1mHV1	RL1mHV0
RL1mHV (H)	RL1mHV15	RL1mHV14	RL1mHV13	RL1mHV12	RL1mHV11	RL1mHV10	RL1mHV9	RL1mHV8

（B） ＲＬ１ハードラッチ乱数値レジスタｍ（ＲＬ１ｍＨＶ）（ｍ＝０～１）

ビット番号	ビット名	意味
１５－０	RL1mHV15-RL1mHV0	外部端子入力により取り込まれた１６ビット乱数ＲＬ１の値乱数値が取り込まれると、ＲＬハードラッチフラグレジスタ０(RLHF0)の当該ビットがセット（１）される

【図４５】

（A）

ビット番号	7	6	5	4	3	2	1	0
RL2mHV (L)	RL2mHV7	RL2mHV6	RL2mHV5	RL2mHV4	RL2mHV3	RL2mHV2	RL2mHV1	RL2mHV0
RL2mHV (H)	RL2mHV15	RL2mHV14	RL2mHV13	RL2mHV12	RL2mHV11	RL2mHV10	RL2mHV9	RL2mHV8

（B） ＲＬ２ハードラッチ乱数値レジスタｍ（ＲＬ２ｍＨＶ）（ｍ＝０～１）

ビット番号	ビット名	意味
１５－０	RL2mHV15-RL2mHV0	外部端子入力により取り込まれた１６ビット乱数ＲＬ２の値乱数値が取り込まれると、ＲＬハードラッチフラグレジスタ１(RLHF1)の当該ビットがセット（１）される

【図４６】

（A）

ビット番号	7	6	5	4	3	2	1	0
RL3mHV (L)	RL3mHV7	RL3mHV6	RL3mHV5	RL3mHV4	RL3mHV3	RL3mHV2	RL3mHV1	RL3mHV0
RL3mHV (H)	RL3mHV15	RL3mHV14	RL3mHV13	RL3mHV12	RL3mHV11	RL3mHV10	RL3mHV9	RL3mHV8

（B） ＲＬ３ハードラッチ乱数値レジスタｍ（ＲＬ３ｍＨＶ）（ｍ＝０～１）

ビット番号	ビット名	意味
１５－０	RL3mHV15-RL3mHV0	外部端子入力により取り込まれた１６ビット乱数ＲＬ３の値乱数値が取り込まれると、ＲＬハードラッチフラグレジスタ１(RLHF1)の当該ビットがセット（１）される

【図４７】

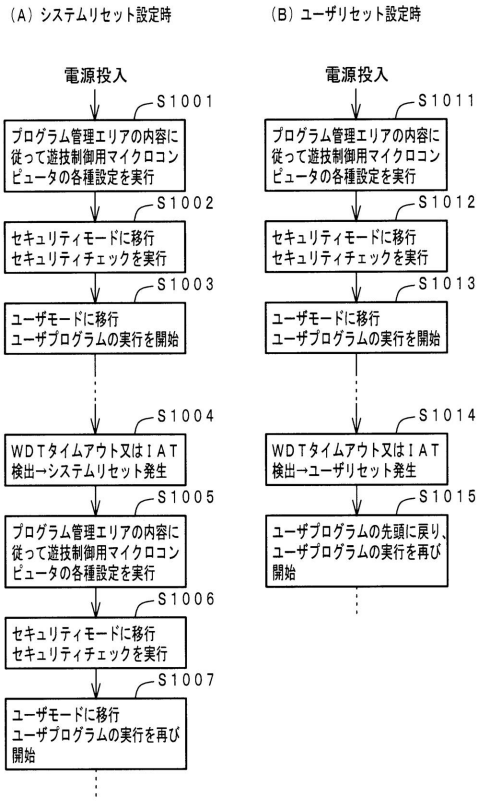
（A）

ビット番号	7	6	5	4	3	2	1	0
RSnHV	RSnHV7	RSnHV6	RSnHV5	RSnHV4	RSnHV3	RSnHV2	RSnHV1	RSnHV0

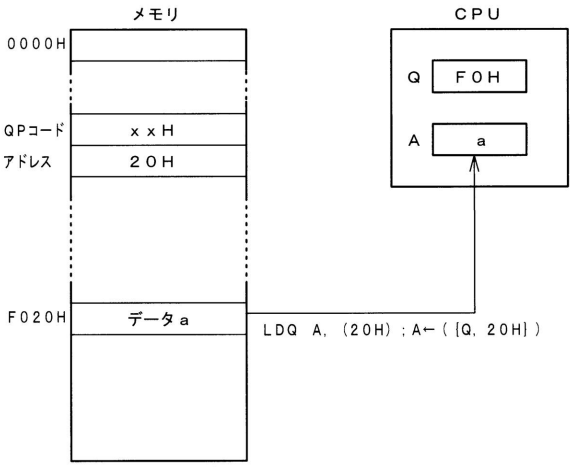
（B） ＲＳｎハードラッチ乱数値レジスタ（ＲＳｎＨＶ）（ｎ＝０～３）

ビット番号	ビット名	意味
７－０	RSnHV7－RSnHV0	外部端子入力により取り込まれた８ビット乱数ＲＳｎの値乱数値が取り込まれると、ＲＳハードラッチフラグレジス(RSHF)の当該ビットがセット（１）される

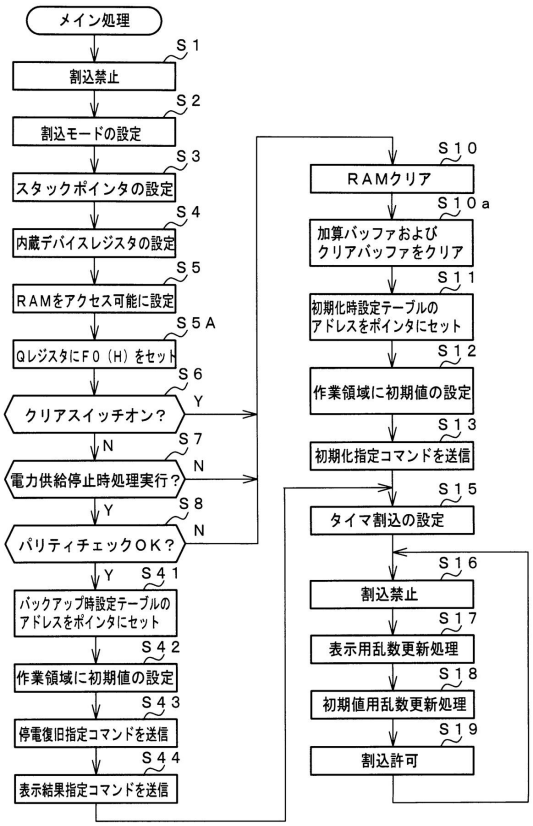
【図４８】



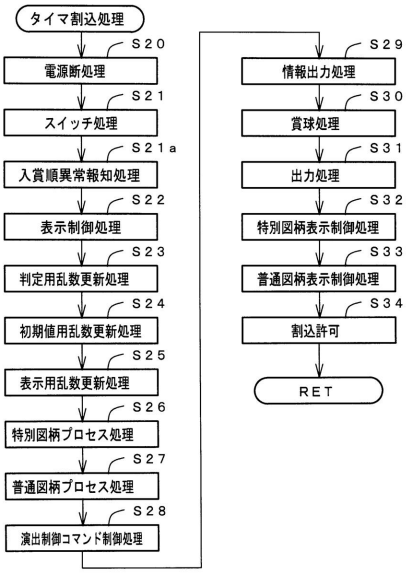
【図４９】



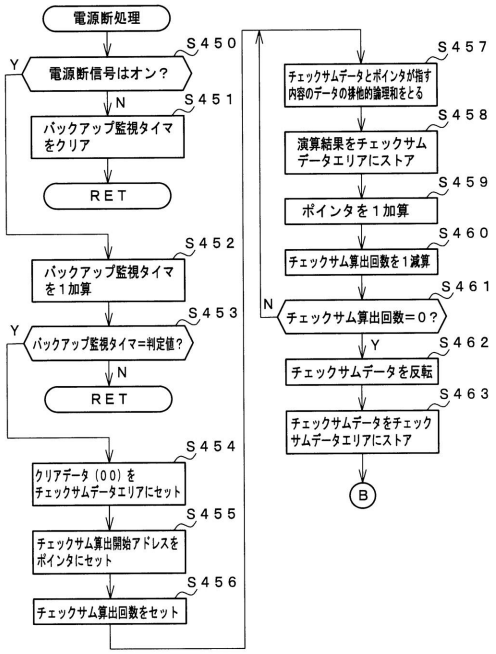
【図５０】



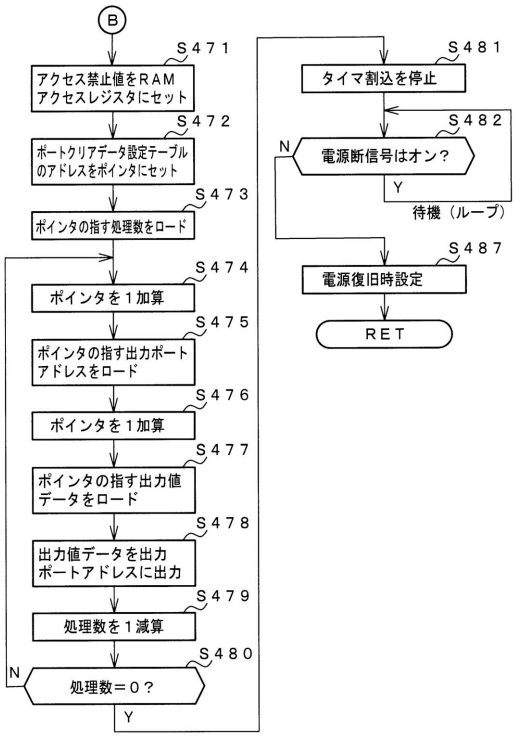
【図５１】



【図５２】



【図５３】



【図５４】

乱数	範囲	用途	加算
ランダム１	0～39	大当り種別判定用	0.002秒毎に１ずつ加算
ランダム２	1～251	変動パターン種別判定用	0.002秒毎および割り込み処理余り時間に１ずつ加算
ランダム３	1～997	変動パターン判定用	0.002秒毎および割り込み処理余り時間に１ずつ加算
ランダム４	3～13	普通図柄当り判定用	0.002秒毎に１ずつ加算
ランダム５	3～13	ランダム４初期値決定用	0.002秒毎および割り込み処理余り時間に１ずつ加算

【図５５】

大当り判定テーブル

大当り判定値　（ランダムR［0～65535］と比較される）	
通常時（非確変時）	確変時
1020～1079，13320～13477（確率：1/300）	1020～1519，13320～15004（確率：1/30）

（A）

小当り判定テーブル（第１特別図柄用）

小当り判定値（ランダムR［0～65535］と比較される）
54000～54217（確率：1／300）

（B）

小当り判定テーブル（第２特別図柄用）

小当り判定値（ランダムR［0～65535］と比較される）
54000～54022（確率：1／3000）

（C）

131a

大当り種別判定テーブル（第１特別図柄用）

大当り種別判定値（ランダム１と比較される）		
１５R確変大当り	７R確変大当り	突然確変大当り
０～１４	１５～３４	３５～３９

（D）

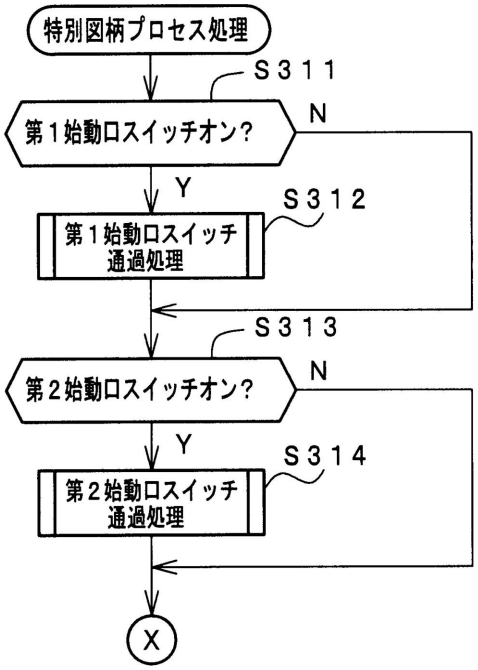
131b

大当り種別判定テーブル（第２特別図柄用）

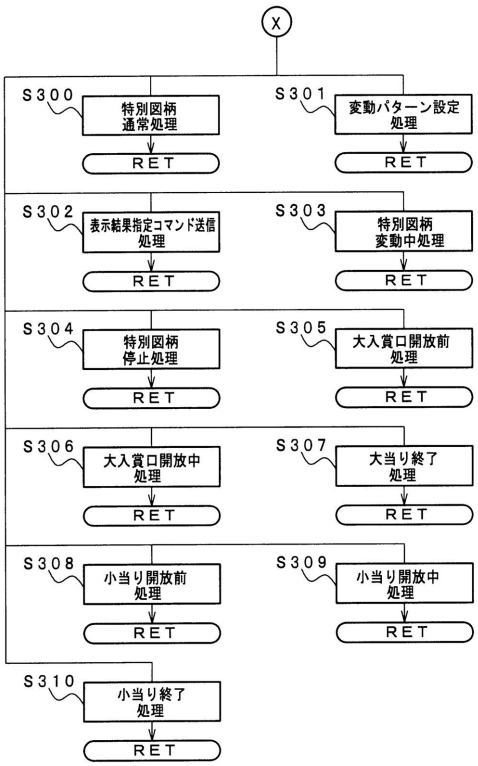
大当り種別判定値（ランダム１と比較される）		
１５R確変大当り	７R確変大当り	突然確変大当り
０～２９	３０～３８	３９

（E）

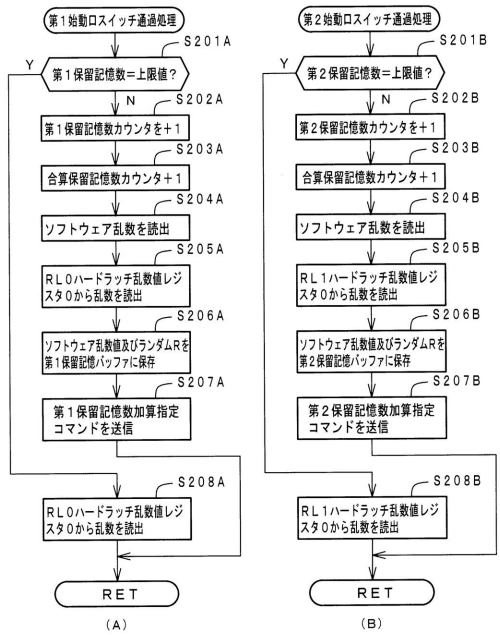
【図５６】



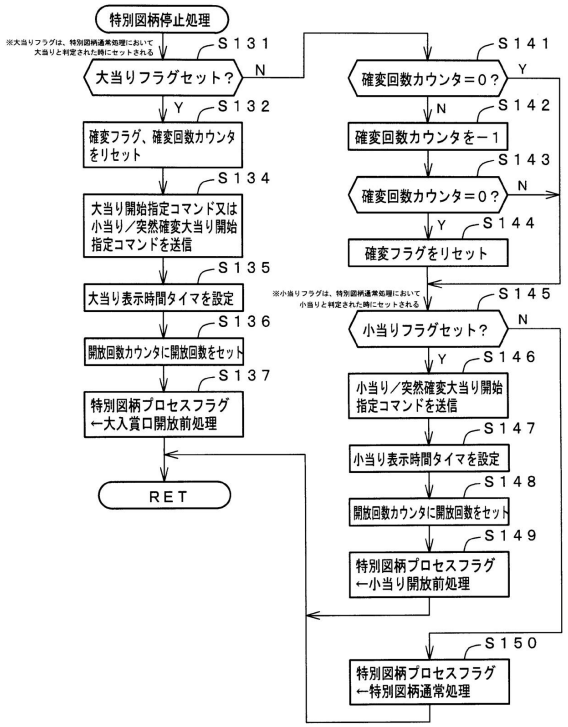
【図５７】



【図５８】



【図６０】

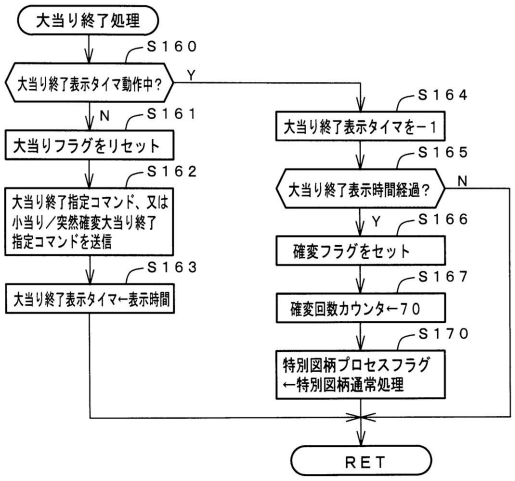


【図５９】

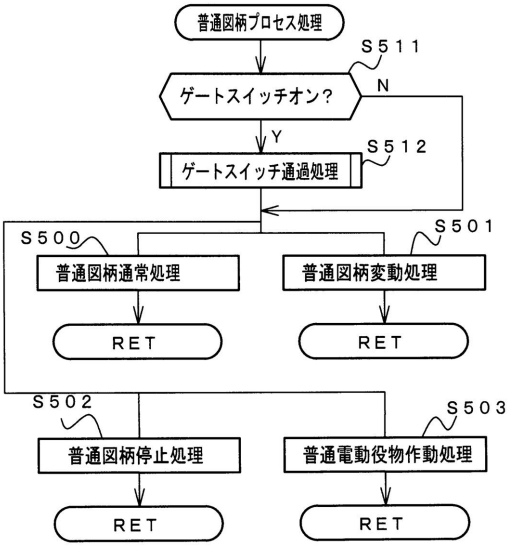
第１保留記憶バッファ	第１保留記憶数＝１に応じた保存領域
	第１保留記憶数＝２に応じた保存領域
	第１保留記憶数＝３に応じた保存領域
	第１保留記憶数＝４に応じた保存領域

第２保留記憶バッファ	第２保留記憶数＝１に応じた保存領域
	第２保留記憶数＝２に応じた保存領域
	第２保留記憶数＝３に応じた保存領域
	第２保留記憶数＝４に応じた保存領域

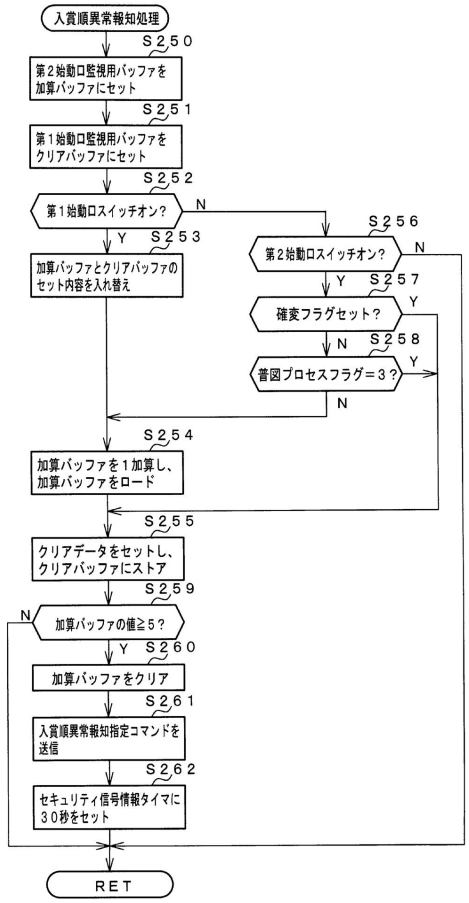
【図６１】



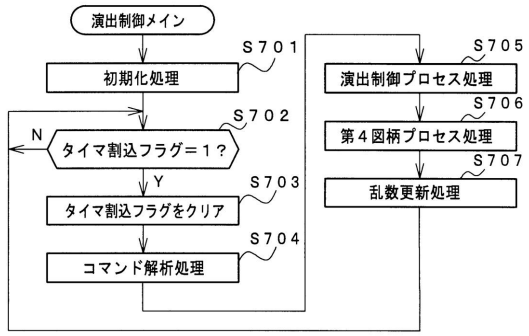
【図６２】



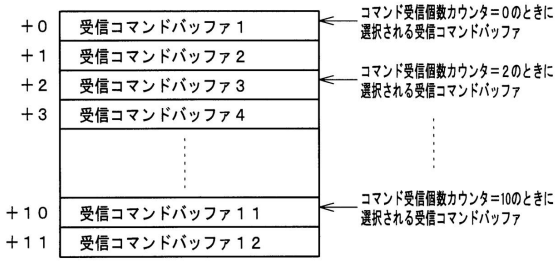
【図６３】



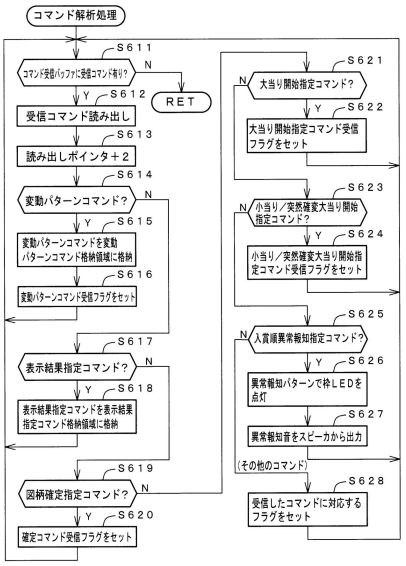
【図６４】



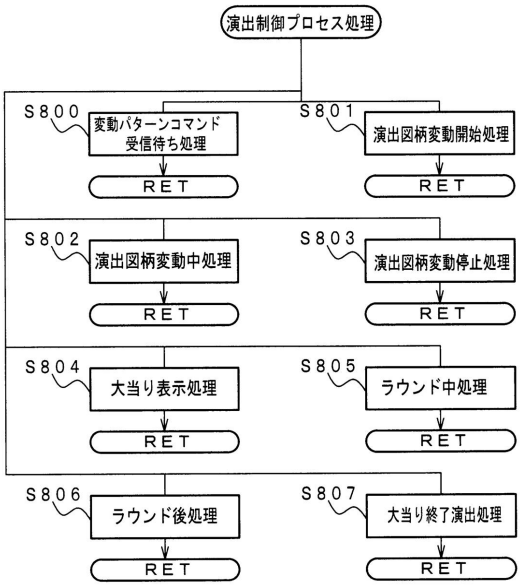
【図６５】



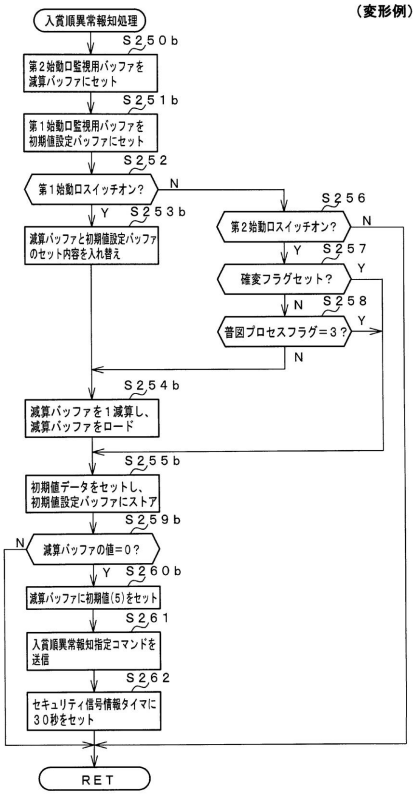
【図６６】



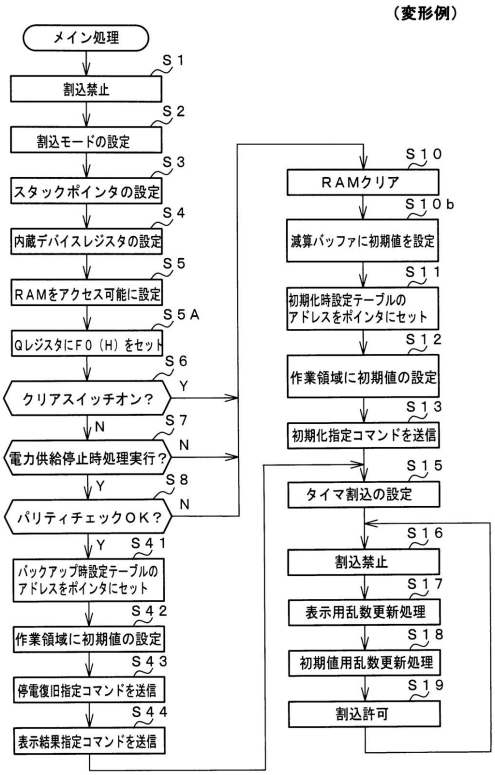
【図６７】



【図６８】

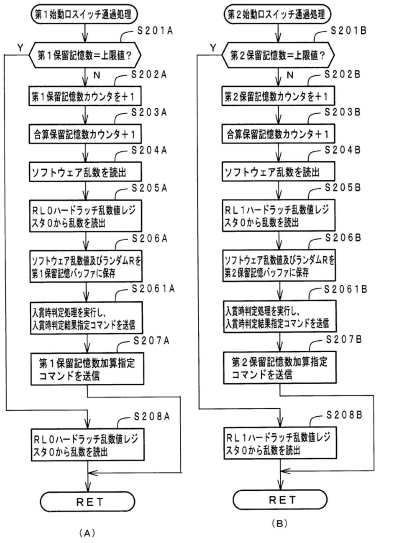


【図６９】



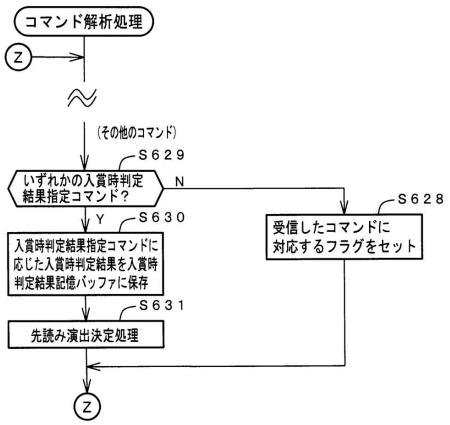
【図７０】

（変形例）



【図７１】

（変形例）



【図７２】

（A）先読み演出の演出態様（非高ベース時）　（変形例）

合算保留記憶数	演出態様
1	当該変動表示前の変動表示において実行される
2	当該変動表示前の2回の変動表示に亘って実行される
3	当該変動表示前の3回の変動表示に亘って実行される
4	当該変動表示前の4回の変動表示に亘って実行される
5	当該変動表示前の5回の変動表示に亘って実行される
6	当該変動表示前の6回の変動表示に亘って実行される
7	当該変動表示前の7回の変動表示に亘って実行される
8	当該変動表示前の8回の変動表示に亘って実行される

（B）先読み演出の演出態様（高ベース時）

合算保留記憶数	演出態様
1	当該変動表示前の変動表示において実行される
2	当該変動表示前の2回の変動表示に亘って実行される
3	当該変動表示前の3回の変動表示に亘って実行される
4	当該変動表示前の4回の変動表示に亘って実行される

フロントページの続き

審査官　廣瀬　貴理

(56)参考文献　特開２０１０－００５０９９（ＪＰ，Ａ）
特開２００７－１５１６０１（ＪＰ，Ａ）
特開２０１４－１００４０２（ＪＰ，Ａ）
特開２０１４－０８３３１７（ＪＰ，Ａ）
特開平１０－１２７８８９（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ａ６３Ｆ　　７／０２