(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7520064号
(P7520064)

(45)発行日　令和6年7月22日(2024.7.22)

(24)登録日　令和6年7月11日(2024.7.11)

(51)国際特許分類
Ａ６３Ｆ　7/02 (2006.01)

ＦＩ
Ａ６３Ｆ　7/02　３２６Ｚ

請求項の数　1　（全77頁）

(21)出願番号　特願2022-32873(P2022-32873)
(22)出願日　令和4年3月3日(2022.3.3)
(65)公開番号　特開2023-128498(P2023-128498A)
(43)公開日　令和5年9月14日(2023.9.14)
審査請求日　令和5年5月24日(2023.5.24)

(73)特許権者　000154679
株式会社平和
東京都台東区東上野一丁目１６番１号
(74)代理人　100120592
弁理士　山崎 崇裕
(74)代理人　100184712
弁理士　扇原 梢伸
(74)代理人　100192223
弁理士　加久田 典子
(72)発明者　吉岡 将吾
東京都台東区東上野一丁目１６番１号
株式会社平和内
(72)発明者　杉山 純也
東京都台東区東上野一丁目１６番１号
株式会社平和内

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】
遊技の実行を制御する遊技制御手段と、
遊技に関する情報を記憶可能な記憶領域を有する記憶手段とを備え、
前記記憶手段は、
前記記憶領域に互いにビット数の異なる第１データ及び第２データが組として設定されるとともに、前記第１データと対応付けた結果データが設定されたテーブル領域を有し、
前記遊技制御手段は、
前記テーブル領域から第１レジスタと第２レジスタのペアに前記第１データと前記第２データの組を読み込んだ後、前記第１レジスタの値を所定値で除算したときの商を前記第１レジスタに格納し、その余りを第３レジスタに格納する命令を実行することで、前記第３レジスタと前記第２レジスタのペアによって前記第２データを生成し、取得した乱数値と生成した前記第２データとを比較した結果に基づいて、前記第２データと組になる前記第１データと対応付けて前記テーブル領域に設定された前記結果データを選択結果として第４レジスタに格納し、前記第１レジスタと前記第４レジスタに格納された２つのデータを用いて、前記第１レジスタに格納した内容と前記第４レジスタに格納した前記選択結果とを返す処理をさらに実行することを特徴とする遊技機。
【発明の詳細な説明】
【技術分野】
【０００１】

本発明は、遊技を実行する遊技機に関する。
【背景技術】
【０００２】
特許文献１には、封入式の遊技機（管理遊技機）が記載されている。
【先行技術文献】
【特許文献】
【０００３】
【文献】特開２０１９－１２２８７４号公報
【発明の概要】
【発明が解決しようとする課題】
【０００４】
近年では、いずれも似通った遊技機ばかりが提案されており、斬新な遊技機が望まれている。
【０００５】
そこで、本発明は、斬新な遊技機を提供することを課題とする。
【課題を解決するための手段】
【０００６】
本発明は、上記の課題を解決するため以下の解決手段を採用する。なお、以下の解決手段及び括弧書中の文言はあくまで例示であり、本発明はこれに限定されるものではない。また、本発明は、以下の解決手段に示す各発明特定事項を少なくとも１つ含む発明とすることができる。さらに、以下の解決手段に示す各発明特定事項には、発明特定事項を限定する要素を追加して下位概念化することができ、発明特定事項を限定する要素を削除して上位概念化することもできる。
【０００７】
〔第１の技術的特徴：エラーに関するコマンドを使用領域とは別の領域外で生成〕
解決手段Ａ１：本解決手段の遊技機は、遊技の実行を制御する遊技制御手段と、記憶領域に設けられた使用領域内に遊技の実行に必要となる情報を格納し、前記使用領域とは別に設けられた領域外に遊技の実行に必要とならない情報を格納することが可能な記憶手段とを備え、前記遊技制御手段は、遊技の実行中に前記領域外の情報を用いてエラーに関する判定を実行し、当該判定の結果を示す結果情報を生成して前記領域外に保存する一方、前記使用領域の情報を用いて前記領域外の前記結果情報を参照することで、前記判定の結果を外部に通知するための通知情報（例えば、サブコマンド）を生成して前記使用領域（例えば、サブコマンドバッファ）に保存することを特徴とする遊技機である。
【０００８】
本解決手段の遊技機は、以下の構成を備えている。
（１）遊技制御手段（主制御基板）が設けられている。遊技制御手段には、例えば主制御ＣＰＵが搭載されている。
【０００９】
（２）記憶手段が設けられている。記憶手段には、ＲＯＭ、ＲＷＭ等の記憶領域が含まれ、記憶領域には使用領域の他に領域外（使用領域とは別の領域）が設けられている。使用領域には、遊技の実行に必要となる情報（例えば、プログラムコード、データテーブル、フラグ、コマンド等）が格納される。領域外には、遊技の実行に必要とならない情報（例えば、プログラムコード、データテーブル、フラグ、コマンド等）が格納される。
【００１０】
（３）遊技制御手段は、エラーに関する判定（例えば、エラーの発生又は解除）を領域外で実行する。エラーに関して何らかの判定をした場合、結果情報（例えば、使用領域でサブコマンドバッファに設定される情報）を生成して領域外に保存しておく。
【００１１】
（４）一方、遊技制御手段は、使用領域では領域外の結果情報を参照して通知情報を生成するだけとし、エラーに関する判定を使用領域では実行しない。

【００１２】
本解決手段によれば、エラーに関する判定に必要となる情報量（プログラムコード量）を使用領域に格納する必要がなくなり、それだけ使用領域の情報量を削減することができる。その結果、斬新な遊技機を提供することができる。
【００１３】
解決手段Ａ２：本解決手段の遊技機は、上述したいずれかの解決手段において、前記遊技制御手段は、前記領域外の情報を用いて複数種類のエラーに関する判定を実行し、前記結果情報を種類毎に生成して前記領域外に保存する際に前記結果情報の数を合わせて保存しておき、前記使用領域の情報を用いて前記領域外を参照する際は、保存されている前記結果情報の数が０である場合は前記通知情報を生成する処理を実行しないことを特徴とする遊技機である。
【００１４】
本解決手段では、以下の特徴が追加される。
（１）遊技制御手段は、領域外の情報を用いて複数種類のエラー（例えば、スイッチ異常エラー、扉開放エラー、磁気検出エラー、電波検出エラー等）に関する判定を実行する。いずれかの種類のエラーに関して何らかの判定をした場合、結果情報をエラーの種類毎に生成して領域外に保存するが、このとき結果情報の数も合わせて保存しておく。逆に、全ての種類のエラーに関して何も判定しなければ（例えば、全ての種類のエラーの発生又は解除があったと判定しなければ）、結果情報を生成しないので、結果情報の数は０が保存される。
【００１５】
（２）そして、遊技制御手段は、使用領域の情報を用いて領域外を参照する際、領域外の結果情報の数が０である場合は通知情報を生成する処理を実行しないこととする。領域外の結果情報の数が０であるということは、エラーに関する判定の結果を外部に通知しないということなので、通知情報を生成する必要がないからである。
【００１６】
本解決手段によれば、使用領域から領域外を参照する際のロジックをより簡素化することができ、それによってさらに使用領域の情報量（プログラムコード使用量）を削減することができる。その結果、より斬新な遊技機を提供することができる。
【００１７】
〔第２の技術的特徴：割込み処理のコードを削減した構成〕
解決手段Ｂ１：本解決手段の遊技機は、遊技の実行を制御する遊技制御手段を備えた遊技機であって、前記遊技制御手段は、電源投入時の初期化処理が完了した後のメインループ処理に移行すると、割込み禁止状態にして前記メインループ処理内の全ての処理を実行し、終了後に前記メインループ処理の末尾から先頭に戻る期間のみを割込み許可状態とすることを特徴とする遊技機である。
【００１８】
本解決手段の遊技機は、以下の構成を備えている。
（１）遊技制御手段（主制御基板）が設けられている。遊技制御手段には、例えば主制御ＣＰＵが搭載されている。
【００１９】
（２）遊技制御手段は、遊技機の電源が投入されると初期化処理（例えば、ＣＰＵ初期化処理）を実行する。初期化処理が完了すると、メインループ処理に移行し、処理内に含まれる処理をループして実行する状態となる。
（３）このとき遊技制御手段は、メインループ処理全体を割込み禁止命令と割込み許可命令で囲んだロジックを実行する。すなわち、遊技制御手段は、メインループ処理に移行すると、割込み禁止状態にして全ての処理を実行する。割込み許可状態にするのは、全ての処理の実行終了後にメインループ処理の末尾から先頭に戻る期間だけとする。
【００２０】
本解決手段によれば、メインループ処理内で全ての処理の実行中に別の処理（例えば、

割込み処理）を実行する余地がないので、プログラム上でメインループ処理中に別の処理が発生したり、そこから復帰したりする際の手当て（例えば、一時記憶中の内容の保全）について考慮する必要がなく、それだけプログラム全体としてコード量を削減することができる。また、逆にいえば、メインループ処理の中に割込み許可状態で実行されるような処理を設けないこととしているので、メインループ処理のコード量も削減することができる。その結果、より斬新な遊技機を提供することができる。

【００２１】

解決手段Ｂ２：本解決手段の遊技機は、上述したいずれかの解決手段において、前記遊技制御手段は、前記メインループ処理の実行中に発生した割込み処理の実行時、前記メインループ処理に関するレジスタの退避及び復帰の処理を非実行とする遊技機である。

【００２２】

本解決手段では、以下の特徴が追加される。

（１）メインループ処理の実行中に、割込み処理（例えば、タイマ割込み処理、電断予告時割込み処理）が発生する。

（２）割込み処理の実行時は、メインループ処理に関するレジスタの退避及び復帰の処理を実行しない。

【００２３】

本解決手段によれば、メインループ処理での割込み処理の発生時にレジスタを退避させたり、あるいはメインループ処理への復帰時にレジスタを復帰させたりする処理をいずれも省略することができ、それだけプログラムコード量を削減することができる。その結果、より斬新な遊技機を提供することができる。

【００２４】

〔第３の技術的特徴：特定命令を使用したチェックサム算定処理の構成〕

解決手段Ｃ１：本解決手段の遊技機は、遊技の実行を制御する遊技制御手段と、遊技に関する情報を記憶可能な記憶領域を有する記憶手段とを備え、前記遊技制御手段は、前記記憶領域で指定した範囲の先頭アドレス及び領域数に基づき、単一の命令を実行することで前記指定した範囲のチェックサムの算定処理を実行することを特徴とする遊技機である。

【００２５】

本解決手段の遊技機は、以下の構成を備えている。

（１）遊技制御手段（主制御基板）が設けられている。遊技制御手段には、例えば主制御ＣＰＵが搭載されている。

（２）記憶手段が設けられている。記憶手段には、ＲＯＭ、ＲＷＭ等の記憶領域が含まれる。

【００２６】

（３）遊技制御手段は、チェックサムの算定処理を実行する。チェックサムの算定処理は、例えば、電源断時及び電源投入時（電断復帰時）に実行することができる。

【００２７】

（４）その上で、遊技制御手段は、チェックサムを算定する対象の記憶領域の範囲をその先頭アドレス及び領域数（例えば、先頭アドレスから連続したアドレス数）で指定しておき、単一の命令（例えば、ＡＤＤＩＲ命令、ＡＤＤＩＲＳ命令）を実行することでチェックサムの算定処理を実行する。

【００２８】

本解決手段によれば、チェックサムの算定処理の実行に用いる命令数を最小（１つ）にすることができるので、複数命令を実行しない分、処理の実行に必要な時間を最短にすることができる。その結果、より斬新な遊技機を提供することができる。

【００２９】

解決手段Ｃ２：本解決手段の遊技機は、上述したいずれかの解決手段において、前記遊技制御手段は、前記記憶領域で指定した第１範囲の第１先頭アドレス及び第１領域数に基づき、単一の命令を実行することで前記第１範囲のチェックサムの算定処理を実行した後、別の第２範囲の第２先頭アドレス及び第２領域数に基づき、単一の命令を実行すること

で前記第２範囲のチェックサムの算定処理を実行することを特徴とする遊技機である。
【００３０】
本解決手段では、以下の特徴が追加される。
（１）記憶領域で第１範囲のチェックサム算定処理を実行する場合、第１範囲についての第１先頭アドレス及び第１領域数を指定し、単一の命令を実行する。
（２）他に、記憶領域で別の第２範囲のチェックサム算定処理を実行する場合は、第２範囲についての第２先頭アドレス及び第２領域数を指定し、単一の命令を実行する。
【００３１】
本解決手段によれば、記憶領域の中でチェックサムの算定を行う範囲（例えば、使用領域と領域外）が複数ある場合は、それら範囲毎に先頭アドレス及び領域数を指定し、それぞれについて単一の命令を実行するだけで算定処理を実行することができる。したがって、複数の範囲（使用領域、領域外）について個別にチェックサムを算定する場合であっても、全体の処理時間を最短にすることができる。その結果、より斬新な遊技機を提供することができる。
【００３２】
解決手段Ｃ３：本解決手段の遊技機は、上述したいずれかの解決手段（例えば解決手段Ｃ２）において、前記遊技制御手段は、前記第１範囲の処理と前記第２範囲の処理とでは、異なる命令を実行可能であることを特徴とする遊技機である。
【００３３】
本解決手段では、以下の特徴が追加される。
すなわち、第１の範囲のチェックサム算定処理では、特定の命令を実行可能である。また、第２の範囲のチェックサム算定処理では、特殊な命令を実行可能である。
【００３４】
本解決手段によれば、チェックサムを算定する対象の第１の範囲と第２の範囲とで異なる命令を実行しつつも、いずれも単一命令であるため、やはり処理時間を最短化することができる。
【００３５】
解決手段Ｃ４：本解決手段の遊技機は、上述したいずれかの解決手段において、前記遊技制御手段は、電源投入時に実行する初期化処理では、チェックサムの算定結果を前記記憶領域に保存し、電源断時に実行する割込み処理では、チェックサムの算定結果に対して２の補数を取る処理を実行した結果を前記記憶領域に保存することを特徴とする遊技機である。
【００３６】
本解決手段では、以下の特徴が追加される。
（１）チェックサムの算定処理は、電源投入時の初期化処理及び電源断時の割込み処理においてそれぞれ実行する。
（２）電源投入時の初期化処理では、チェックサムの算定結果を記憶領域（例えば、ＲＷＭ）に保存する。
（３）電源断時の割込み処理では、チェックサムの算定結果の符号を反転させて記憶領域（例えば、ＲＷＭ）に保存する。
【００３７】
本解決手段によれば、電源投入時に比較して電源断時の処理が２の補数を取る処理を実行している分、長いものとなるが、それでも電源投入時の初期化処理及び電源断時の割込み処理のいずれにおいても、必要な処理時間を最短にすることができる。その結果、より斬新な遊技機を提供することができる。
【００３８】
〔第４の技術的特徴：変動パターン選択２処理におけるコード削減の構成〕
解決手段Ｄ１：本解決手段の遊技機は、遊技の実行を制御する遊技制御手段と、遊技に関する情報を記憶可能な記憶領域を有する記憶手段とを備え、前記記憶手段は、前記記憶領域に互いにビット数が異なる第１データ及び第２データが組として設定されたテーブル

領域を有し、前記遊技制御手段は、前記テーブル領域から第１レジスタと第２レジスタのペアに前記第１データと前記第２データの組を読み込んだ後、前記第１レジスタの値を所定値で除算したときの商を前記第１レジスタに格納し、その余りを第３レジスタに格納する命令を実行することで、前記第３レジスタと前記第２レジスタのペアによって前記第２データを生成することを特徴とする遊技機である。

【００３９】

本解決手段の遊技機は、以下の構成を備えている。

（１）遊技制御手段（主制御基板）が設けられている。遊技制御手段には、例えば主制御ＣＰＵが搭載されている。

（２）記憶手段が設けられている。記憶手段には、ＲＯＭ、ＲＷＭ等の記憶領域が含まれる。

【００４０】

（３）記憶手段の記憶領域には、第１データ（例えば、図柄の変動パターンを決定する要素となるモード番号）と第２データ（例えば、乱数値と比較される比較値）が組になったデータ構成のテーブル領域が設定されている。例えば、第１データは１バイト未満であるが、第２データは１バイトより大きく、これらを組にすると、全体で２バイトとなるデータ構造である。なお、テーブル領域内には、第１データと第２データの組が複数設けられている。

【００４１】

（４）遊技制御手段は、所定の命令（例えば、ＬＤＩＮ命令）を実行し、テーブル領域から第１レジスタと第２レジスタのペア（例えば、レジスタ・ペアＡＥ）に第１データと第２データの組（例えば、ＨＬレジスタの内容で指定されるアドレスのテーブル領域のデータ）を読み込む。このとき、第１データと第２データの組は、ちょうど２バイトのレジスタのペアに収まる。

【００４２】

（５）遊技制御手段は、特定の命令（例えば、ＤＩＶ　Ｄ，Ａ，１６）を実行し、第１レジスタの値を所定値（例えば、１６）で除算するとともに、その時の商を第１レジスタ（例えば、Ａレジスタ）に格納し、余りを第３レジスタ（例えば、Ｄレジスタ）に格納する。

【００４３】

（６）上記（５）の結果から、第３レジスタと第２レジスタのペア（例えば、レジスタ・ペアＤＥ）には第２データ（例えば、乱数値と比較される比較値）が生成されることになる。また、同時に除算の商である第１データ（例えば、モード番号データ）が第１レジスタ（例えば、Ａレジスタ）内で右シフトされ、そのまま第１レジスタに残った状態で設定される。

【００４４】

本解決手段によれば、テーブル領域から第１レジスタと第２レジスタのペアに読み出した第１データと第２データの組に対して、そこに含まれる「第１データと第２データの分離」及び「第１データの右シフト」を１命令で実行することができ、それに伴って第２データを別の第３レジスタと第２レジスタのペアによって生成することができる。これにより、テーブル領域のデータの組を用いた選択処理を実行するためのプログラムコード量を最小化することができる。その結果、より斬新な遊技機を提供することができる。

【００４５】

解決手段Ｄ２：本解決手段の遊技機は、上述したいずれかの解決手段において、前記遊技制御手段は、取得した乱数値と生成した前記第２データとを比較した結果に基づいて、前記第１レジスタに格納した内容と対応付けられた値を選択結果として返す処理をさらに実行することを特徴とする遊技機である。

【００４６】

本解決手段では、以下の特徴が追加される。

遊技制御手段は、生成した第２データを取得した乱数値と比較し、その結果に基づいて選択結果を返す。このとき選択結果は、第１レジスタに格納した内容、つまり、第１デー

タと対応付けられた値（例えば、テーブル領域のアドレスで指し示される値）となる。
【００４７】
本解決手段によれば、選択結果を返す際に、改めて第１データをテーブル領域から読み出してくる処理を実行する必要がなく、それだけプログラムコード量を削減することができる。その結果、より斬新な遊技機を提供することができる。
【００４８】
〔第５の技術的特徴：ベース、役物比率及び連役比率の算定タイミングの構成〕
解決手段Ｅ１：本解決手段の遊技機は、遊技の実行を制御する遊技制御手段を備え、遊技に関して算定可能な複数種類の性能を有した遊技機であって、前記遊技制御手段は、制御に際してタイマ割込み処理を繰り返し実行しつつ、前記複数種類の性能を算定する複数の算定処理のうち、１回のタイマ割込み処理の中ではいずれか１つの算定処理のみを実行可能とすることを特徴とする遊技機である。
【００４９】
本解決手段の遊技機は、以下の構成を備えている。
（１）遊技制御手段（主制御基板）が設けられている。遊技制御手段には、例えば主制御ＣＰＵが搭載されている。
（２）遊技に関して算定可能な複数種類の性能を有している。複数種類の性能は、例えば、ベース値、役物比率、連続役物比率等である。
【００５０】
（３）遊技制御手段は、制御に際してタイマ割込み処理を繰り返し実行する。このとき、複数種類の性能を算定する複数の算定処理があるが、１回のタイマ割込み処理の中では、いずれか１つの算定処理のみを実行可能としている。
【００５１】
本解決手段によれば、遊技機の性能として複数種類の性能があり、それぞれを算定する複数の算定処理があるとしても、１回のタイマ割込み処理の中では複数を実行しないことにより、遊技機の性能を算定する処理に要する時間を最短にすることができる。その結果、より斬新な遊技機を提供することができる。
【００５２】
解決手段Ｅ２：本解決手段の遊技機は、上述したいずれかの解決手段において、前記遊技制御手段は、１回のタイマ割込み処理の実行中、前記複数の算定処理に対して設定された優先度に基づいて、今回のタイマ割込み処理内で実行可能ないずれか１つの算定処理を決定することを特徴とする遊技機である。
【００５３】
本解決手段では、以下の特徴が追加される。
（１）複数の算定処理には、予め優先度が設定されている。例えば、優先度の高い順にベース値の算定処理→役物比率の算定処理→連続役物比率の算定処理である。
（２）今回のタイマ割込み処理の中で、複数の算定処理を実行する条件が満たされているとした場合でも、複数を実行することとせず、優先度の高い算定処理だけを実行する。
【００５４】
本解決手段によれば、条件的に複数種類の性能を算定する状況になったとしても、優先度に基づいて、いずれか１つの算定処理だけを実行することにより、処理時間を最短にすることができる。その結果、より斬新な遊技機を提供することができる。
【発明の効果】
【００５５】
本発明によれば、斬新な遊技機を提供することができる。
【図面の簡単な説明】
【００５６】
【図１】管理遊技機のブロック図である。
【図２】主制御基板３０（主制御ＣＰＵ３１）が用いるメモリ領域のメモリマップを示す図である。

【図３】ＣＰＵ初期化処理の手順例を示すフローチャートである（１／２）。
【図４】ＣＰＵ初期化処理の手順例を示すフローチャートである（２／２）。
【図５】割込み待ちメインループ処理（１）のプログラムを示す図である。
【図６】割込み待ちメインループ処理（２）の手順例を示すフローチャートである。
【図７】割込み待ちメインループ処理（２）のプログラムを示す図である。
【図８】比較例の割込み待ちメインループ処理の手順例を示すフローチャートである。
【図９】比較例の割込み待ちメインループ処理のプログラムを示す図である。
【図１０】比較例のその他乱数更新処理のプログラムを示す図である。
【図１１】タイマ割込み処理のプログラムを示す図である。
【図１２】電源断時の退避処理のプログラムを示す図である。
【図１３】タイマ割込み処理の手順例を示すフローチャートである。
【図１４】電源断時の退避処理の手順例を示すフローチャートである（１／２）。
【図１５】電源断時の退避処理の手順例を示すフローチャートである（２／２）。
【図１６】比較例のチェックサム算定処理のプログラムを示す図である。
【図１７】比較例のチェックサム確認処理のプログラムを示す図である。
【図１８】本実施形態のチェックサム算定処理のプログラムを示す図である。
【図１９】本実施形態のチェックサム確認処理のプログラムを示す図である。
【図２０】本実施形態による処理時間の短縮化の検証結果を示す図である。
【図２１】使用領域の状態管理処理のプログラムを示す図である。
【図２２】状態管理処理の手順例を示すフローチャートである。
【図２３】領域外に格納されるエラー確認処理のプログラムを示す図である（１／５）。
【図２４】領域外に格納されるエラー確認処理のプログラムを示す図である（２／５）。
【図２５】領域外に格納されるエラー確認処理のプログラムを示す図である（３／５）。
【図２６】領域外に格納されるエラー確認処理のプログラムを示す図である（４／５）。
【図２７】領域外に格納されるエラー確認処理のプログラムを示す図である（５／５）。
【図２８】エラー管理処理テーブル１の構成例を示す図である（１／２）。
【図２９】エラー管理処理テーブル１の構成例を示す図である（２／２）。
【図３０】エラー管理処理テーブル２の構成例を示す図である。
【図３１】エラー確認処理の手順例を示すフローチャートである（１／７）。
【図３２】エラー確認処理の手順例を示すフローチャートである（２／７）。
【図３３】エラー確認処理の手順例を示すフローチャートである（３／７）。
【図３４】エラー確認処理の手順例を示すフローチャートである（４／７）。
【図３５】エラー確認処理の手順例を示すフローチャートである（５／７）。
【図３６】エラー確認処理の手順例を示すフローチャートである（６／７）。
【図３７】エラー確認処理の手順例を示すフローチャートである（７／７）。
【図３８】変動パターン決定処理の手順例を示すフローチャートである。
【図３９】比較例の変動パターン選択２処理のプログラムを示す図である。
【図４０】比較例の変動パターン選択２処理の手順例を示すフローチャートである。
【図４１】本実施形態の変動パターン選択２処理のプログラム関係を示す図である。
【図４２】テーブル「Ｄ＿ＭＯＤ＿ＳＥＬ＿０７」を使用した振り分けを示す図である。
【図４３】本実施形態の変動パターン選択２処理の手順例を示すフローチャートである。
【図４４】性能表示モニタ表示処理の手順例を示すフローチャートである（１／３）。
【図４５】性能表示モニタ表示処理の手順例を示すフローチャートである（２／３）。
【図４６】性能表示モニタ表示処理の手順例を示すフローチャートである（３／３）。
【図４７】未定賞球カウンタ加算処理のプログラムを示す図である。
【図４８】特別遊技管理フェーズ等の確認を実行するプログラムを示す図である。
【図４９】連続役物賞球カウンタ等更新処理のプログラムを示す図である。
【図５０】ベース値算定後の算定管理バッファ設定処理のプログラムを示す図である。
【図５１】算定管理バッファに基づく処理の振り分けのプログラムを示す図である。
【図５２】役物比率算定後の算定管理バッファ設定処理のプログラムを示す図である。

【図５３】連続役物比率算定後の算定管理バッファ設定処理のプログラムを示す図である。
【図５４】ベース値の算定部分のプログラムを示す図である（１／２）。
【図５５】ベース値の算定部分のプログラムを示す図である（２／２）。
【図５６】各種性能値の算定に必要となる時間を示す図である。
【図５７】別案で算定フラグと実行する処理の対応関係を示す図である。
【発明を実施するための形態】
【００５７】
以下、本発明の実施形態について、図面を参照しながら説明する。
図１は、管理遊技機のブロック図である。
管理遊技機１００は、管理遊技機遊技盤１０と、管理遊技機枠２０とを備えており、機械単体内で一定数の遊技球を循環させ、遊技に用いる遊技球や遊技者に遊技球を直接払い出すことを不要とした封入式の遊技機である。管理遊技機１００は、管理遊技機専用の専用ユニット（ＩＣカードユニット）と接続することで遊技が可能になる。
【００５８】
管理遊技機遊技盤１０は、遊技板（アクリル板）、図柄表示装置（特別図柄表示装置、普通図柄表示装置）、入賞・球通過検知機構（各種スイッチ）、風車、遊技くぎ、その他の演出用装置（演出用可動体、液晶表示器、情報表示装置、スピーカ、ランプ）、入賞表示装置、落下の方向に変化を与える装置（可動部材、蓋部材）等を備えている。
【００５９】
管理遊技機枠２０は、遊技球、発射装置、電源装置、球磨き装置、遊技球循環装置、遊技球数等表示装置、ガラス板、夜間監視装置、その他演出用装置、鉄球検出装置、前飾り、計数スイッチ等を備えている。
【００６０】
管理遊技機１００では、管理遊技機枠２０の前飾りを開放状態にすることが可能となっている。前飾りを開放状態にすると、前飾り及びガラス板が開放し、ガラス板の後方に配置されている管理遊技機遊技盤１０が露出する。
【００６１】
〔管理遊技機の概要〕
管理遊技機の概要は、以下の通りである。
（１）管理遊技機１００は、パチンコ機と同一の分類に属し、遊技者は、遊技球を管理遊技機遊技盤１０の盤面に打ち出すことにより、遊技を行う。
（２）管理遊技機１００は、構造的に上皿・下皿が無く、遊技者が直接遊技球に触れることができない。
（３）管理遊技機１００は、発射に必要な最小数の遊技球を管理遊技機１００内で循環させ、循環途中には、発射装置、球磨き装置、及び、遊技球循環装置が設けられている。
（４）発射装置は、現行遊技機の発射性能を有しており、下部発射であっても上部発射であってもよい。
（５）球磨き装置は、遊技球の汚れを落とすための研磨布がカセット内に収納されており、一定期間毎に交換する。
（６）遊技球数や獲得遊技球数は、数値データとして管理遊技機１００（枠制御基板５０）で管理して、表示する。
（７）遊技終了時は、計数スイッチ５４（計数ボタン）を押下することで管理遊技機１００（枠制御基板５０）が管理していた遊技球数を専用ユニットに移行して管理することができる。
【００６２】
管理遊技機遊技盤１０は、主制御基板３０（遊技制御手段）及び演出制御基板４０（演出制御手段）を備えている。
主制御基板３０は、主制御基板３０は、主制御ＣＰＵ３１を有しており、遊技の進行に関する内容（又は遊技の実行）を制御する。演出制御基板４０は、演出制御ＣＰＵ４１を有しており、遊技の演出に関する内容を制御する。主制御基板３０は、演出制御基板４０

及び枠制御基板５０と通信可能である。演出制御基板４０は、主制御基板３０と通信可能であるが、この通信は、主制御基板３０から演出制御基板４０への片方向のみで行われ、双方向の通信は行われない。

【００６３】

主制御基板３０には、入賞口スイッチ３２、アウトスイッチ３３、盤面磁気検出第１センサ３４、盤面磁気検出第２センサ３５、電波検出センサ３６及び性能表示モニタ３７が接続されている。また、主制御基板３０には、扉開放スイッチ３８が接続されている。入賞口スイッチ３２は、普通図柄や特別図柄に対応する入賞口、大入賞口、普通入賞口（一般入賞口）等の各スイッチを含んでいる。アウトスイッチ３３は、アウト通路に配置されている。遊技領域（盤面）に打ち出された遊技球は、遊技領域を流下し、入賞口又はアウト口に入球するが、いずれの場合であっても、アウト通路に導かれ、アウトスイッチ３３により検出される。扉開放スイッチ３８は、例えば、ガラス扉が開放している場合には、ガラス扉の開放を検出し、内枠が開放している場合には、内枠の開放を検出し、扉開放信号（エラー信号）を主制御基板３０に出力する。なお、扉開放スイッチ３８は、ガラス扉用と内枠用の２つのスイッチで構成してもよく、ガラス扉及び内枠を兼用した１つのスイッチで構成してもよい。

【００６４】

盤面磁気検出第１センサ３４、盤面磁気検出第２センサ３５及び電波検出センサ３６は、エラー（不正）検出センサである。盤面磁気検出第１センサ３４及び盤面磁気検出第２センサ３５は、例えば管理遊技機遊技盤１０の異なる位置に設置され、それぞれの位置近傍で所定強度以上の磁気を検出すると、磁気検出信号を主制御基板３０に出力する。電波検出センサ３６は、例えば管理遊技機遊技盤１０に設置され、その位置近傍で所定強度以上の電波を検出すると、電波検出信号を主制御基板３０に出力する。

【００６５】

性能表示モニタ３７は、管理遊技機１００の性能等を表示する表示装置であり、例えば複数の７セグメントＬＥＤを有する。性能表示モニタ３７は、性能としてベース値、役物比率、連続役物比率を所定の条件で表示する。また、性能表示モニタ３７は、管理遊技機１００が設定を有する場合、設定値を表示する。その他に性能表示モニタ３７は、管理遊技機１００で発生したエラーのコード情報を表示することもできる。

【００６６】

演出制御基板４０には、液晶表示器４２、スピーカ４３及びランプ４４が接続されている。演出制御基板４０（演出制御ＣＰＵ４１）は、主制御基板３０から送信されたコマンド（サブコマンド）を受信し、受信したコマンドに基づいて演出の内容を決定し、液晶表示器４２、スピーカ４３及びランプ４４を用いて演出を実行する。

【００６７】

管理遊技機枠２０は、枠制御基板５０を備えている。枠制御基板５０は、枠制御ＣＰＵ５１を有しており、遊技球数に関する内容を制御する。枠制御基板５０は、主制御基板３０と通信可能である。枠制御基板５０には、減算センサ５２、ファール球センサ５３、計数スイッチ５４、遊技球数等表示装置５５、タッチセンサ５６及び前飾り開放スイッチ５７が接続されている。

【００６８】

枠制御基板５０（枠制御ＣＰＵ５１）は、主制御基板３０から賞球数コマンド（獲得遊技球数の要求）を受信した場合、賞球数コマンドに含まれる賞球数を遊技球数へ加算し、遊技球数等表示装置５５に表示する。枠制御基板５０は、減算センサ５２が遊技球の発射を検出した場合、遊技球数を１減算し、遊技球数等表示装置５５に表示する。枠制御基板５０は、ファール球センサ５３が遊技球の通過を検出した場合、遊技球数を１加算し、遊技球数等表示装置５５に表示する。枠制御基板５０は、計数スイッチ５４が押下された場合、計数球数を図示しない専用ユニットへ送信し、遊技球数から計数球数を減算し、減算後の遊技球数を遊技球数等表示装置５５に表示する。枠制御基板５０は、図示しない専用ユニットから貸球数の個数情報を受信した場合、その個数情報と遊技球数とを加算した個

数を遊技球数として遊技球数等表示装置５５に表示する。なお、ファール球（戻り球）とは、発射装置により遊技球を発射したものの遊技領域に到達せずに戻ってきた遊技球である。

【００６９】

なお、タッチセンサ５６は、静電容量の変化から遊技者の身体が図示しないグリップユニット（発射ハンドル、球発射装置）に触れていることを検出し、その検出信号を枠制御基板５０に出力する。また、前飾り開放スイッチ５７は、管理遊技機枠２０の前飾りが開放状態になると、扉開放信号（エラー信号）を枠制御基板５０に出力する。

【００７０】

図２は、主制御基板３０（主制御ＣＰＵ３１）が用いるメモリ領域のメモリマップを示す図である。

主制御基板３０（主制御ＣＰＵ３１）が用いるメモリ領域は、ＲＯＭに割り当てられたメモリ領域（００００Ｈ～２ＦＦＦＨ）と、ＲＡＭ（ＲＷＭ）に割り当てられたメモリ領域（Ｆ０００Ｈ～Ｆ３ＦＦＨ）とを含んでいる。

【００７１】

アドレス０００００Ｈ～０ＢＦＦＦＨの領域（第１領域）は、使用領域のプログラムコード（プログラム）を格納可能な領域である。使用領域のプログラムコードは、遊技の実行に必要となる（遊技の進行を制御するための）第１情報である。

【００７２】

アドレス０００００Ｈ～０ＢＦＦＦＨの領域（第１領域）には、使用領域のプログラムコード（第１情報）、及び、遊技の実行に必要とならないプログラムコード（第２情報）のうちの一部の情報である特別プログラムコード（特別情報）が格納されている。

【００７３】

アドレス０Ｃ００Ｈ～０ＦＦＦＨの領域は、未使用の領域である。

【００７４】

アドレス１０００Ｈ～１ＢＦＦＨの領域（第１領域）は、使用領域のプログラムデータ（データ）を格納可能な領域である。使用領域のプログラムデータは、遊技の実行に必要となる第１情報である。

アドレス１０００Ｈ～１ＢＦＦＨの領域（第１領域）には、使用領域のプログラムデータ（第１情報）、及び、遊技の実行に必要とならないプログラムデータ（第２情報）のうちの一部の情報である特別プログラムデータ（特別情報）が格納されている。

【００７５】

アドレス１Ｃ００Ｈ～１ＦＦＦＨの領域は、未使用の領域である。

【００７６】

アドレス２０００Ｈ～２ＦＦＦＨの領域（第２領域）は、領域外のプログラムコード・プログラムデータを格納可能な領域であり、使用領域とは別の領域である。領域外のプログラムコード・プログラムデータは、遊技の実行に必要とならない（例えば、遊技機規則で定める試験を行うための処理や、性能表示モニタに関連する処理に関する）第２情報である。さらに本実施形態では、エラーの発生、解除の判定を行う処理を領域外で実行することを１つの技術的特徴としている。この技術的特徴についてはさらに後述する。

【００７７】

アドレス２０００Ｈ～２ＦＦＦＨの領域（第２領域）には、遊技の実行に必要とならないプログラムコード・プログラムデータ（第２情報）のうちの残りの情報である非特別プログラムコード・非特別プログラムデータ（非特別情報）が格納されている。

【００７８】

アドレス３０００Ｈ～ＥＦＦＦＨの領域は、未使用の領域である。

【００７９】

アドレスＦ０００Ｈ～Ｆ１ＦＦＨの領域は、使用領域のワーク領域及びスタックＲＷＭとして用いられる。

アドレスＦ２００Ｈ～Ｆ３ＦＦＨの領域は、領域外のワーク領域及びスタックＲＷＭと

して用いられる。
　これらの領域は、使用領域のプログラムが実行されている際や領域外のプログラムが実行されている際に、一時的に用いられる領域やデータを一時的に退避させるスタック領域として用いられる。
【００８０】
　アドレスＦ４００Ｈ～ＦＦＦＦＨの領域は、未使用の領域である。
【００８１】
　ＲＯＭのメモリ領域には、使用領域及び領域外以外に、プログラムのタイトル、バージョン等の任意のデータが格納される領域や、主制御基板３０（主制御ＣＰＵ３１）がプログラムを実行するために必要な情報が格納されるプログラム管理領域を設けてもよい。
【００８２】
　このように、本実施形態のＲＯＭ及びＲＡＭ（記憶領域を有した記憶手段）は、遊技の実行に必要となるプログラムコード・プログラムデータ（第１情報）を格納可能な使用領域（第１領域）、及び、遊技の実行に必要とならないプログラムコード・プログラムデータ（第２情報）を格納可能な領域外（第２領域）を有する。
【００８３】
　ここで、使用領域とは、遊技の進行を制御するために使用可能な容量が遊技機規則で定められている領域であり、領域外とは、使用可能な容量の計算に含めない領域である。
【００８４】
　主制御基板３０（主制御ＣＰＵ３１）は、遊技の実行に必要となるプログラムコード・プログラムデータ（第１情報）及び遊技の実行に必要とならないプログラムコード・プログラムデータ（第２情報）に基づいて管理遊技機１００を制御する（遊技制御手段）。
【００８５】
　使用領域（第１領域）には、遊技の実行に必要となるプログラムコード・プログラムデータ（第１情報）、及び、遊技の実行に必要とならないプログラムコード・プログラムデータ（第２情報）のうちの一部の情報である特別プログラムコード・特別プログラムデータ（特別情報）が格納されている。
【００８６】
　領域外（第２領域）には、遊技の実行に必要とならないプログラムコード・プログラムデータ（第２情報）のうちの残りの情報である非特別プログラムコード・非特別プログラムデータ（非特別情報）が格納されている。
【００８７】
　特別プログラムコード・特別プログラムデータ（特別情報）は、遊技の内容が異なる複数種類の機種における各機種で異なる情報であり、非特別プログラムコード・非特別プログラムデータ（非特別情報）は、遊技の内容が異なる複数種類の機種における各機種で共通する情報である。
【００８８】
　このため、遊技機の仕様を変更する場合には、特別プログラムコード・特別プログラムデータを変更する必要がある。一方、非特別プログラムコード・非特別プログラムデータは、遊技機の仕様を変更する場合であっても、変更する必要がない。
【００８９】
　特別プログラムコード・特別プログラムデータ（特別情報）には、ベースを算出する処理のうち、遊技状態に基づいてベースを算出するか否かを判定する判定処理、及び、賞球数の値を確認する確認処理のうち少なくとも一方の処理に関する情報が含まれている。判定処理又は確認処理のいずれかは、非特別プログラムコード・非特別プログラムデータ（非特別情報）としてもよい。
【００９０】
　本実施形態では、主制御基板３０（主制御ＣＰＵ３１）の性能上、同一の命令（プログラムコード）を使用する場合であっても、使用領域で使用する場合と、領域外で使用する場合とで処理速度が異なることがある。例えば、使用領域ではＬＤＱ命令（特殊ロード命

令）を使用する場合、処理速度は速くなるが、領域外でＬＤＱ命令を使用する場合、処理速度は速くならない。このため、領域外では処理速度が遅いＬＤ命令（通常ロード命令）を使わざるを得ない。この理由は、ＬＤＱ命令を領域外で使用する場合、スタックポインタ、Ｑレジスタの退避、再設定、復帰処理を追加する必要があるため、処理速度がかえって遅くなってしまうからである。

このため、本実施形態では、領域外に関する処理を実行する場合よりも、使用領域に関する処理を実行する場合の方が、処理速度が速い。

【００９１】

主制御基板３０（主制御ＣＰＵ３１）は、使用領域（第１領域）に格納されているプログラムコード・プログラムデータを用いることにより（実行することにより）、遊技の実行に必要となる使用領域の処理（第１処理）を実行することができる（第１処理実行手段）。

【００９２】

また、主制御基板３０（主制御ＣＰＵ３１）は、領域外（第２領域）に格納されているプログラムコード・プログラムデータを用いることにより（実行することにより）、遊技の実行に必要とならない領域外の処理（第２処理）を実行することができる（第２処理実行手段）。

以上が管理遊技機１００の制御に関する構成例である。

【００９３】

続いて、主制御基板３０（主制御ＣＰＵ３１）により実行される制御上の処理について説明する。なお、以下では主制御基板３０が実行するものとして説明する。

【００９４】

〔主制御装置におけるＣＰＵ初期化（メイン）処理〕

管理遊技機１００に電源が投入されると、主制御基板３０がＣＰＵ初期化処理を開始する。ＣＰＵ初期化処理は、前回の電源遮断時に保存されたバックアップ情報を元に遊技状態を復旧（いわゆる復電）したり、逆にバックアップ情報をクリアしたりすることで、管理遊技機１００の初期状態を整えるための処理である。また、ＣＰＵ初期化処理は、初期状態の調整後に管理遊技機１００の安定した遊技動作を保証するためのメイン処理（メイン制御プログラム）として位置付けられる。

【００９５】

図３及び図４は、ＣＰＵ初期化処理の手順例を示すフローチャートである。

ステップＳ２３０：主制御基板３０は、スタックエリアベースアドレスをセットする処理を実行する。

ステップＳ２３１：主制御基板３０は、内蔵レジスタ設定テーブルのアドレスをセットする処理を実行する。

ステップＳ２３２：主制御基板３０は、内蔵レジスタアドレスの上位バイト及び設定データ数をロードする処理を実行する。

ステップＳ２３３：主制御基板３０は、内蔵レジスタ（ＦＦｘｘＨ）を設定する処理を実行する。

ステップＳ２３４：主制御基板３０は、内蔵レジスタアドレスの上位バイト及び設定データ数をロードする処理を実行する。

ステップＳ２３５：主制御基板３０は、内蔵レジスタ（ＦＦｘｘＨ）及び発射許可信号の設定する処理を実行する。

ステップＳ２３６：主制御基板３０は、Ｑレジスタに作業領域先頭アドレスの上位バイトをセットする処理を実行する。

ステップＳ２３７：主制御基板３０は、電源投入時の入力サポートの状態を取得する処理を実行する。

ステップＳ２３８：主制御基板３０は、サブ制御起動待ち（演出制御基板の起動待ち）、かつ、主制御基板の電源が安定することを確認する処理を実行する。

【００９６】

ステップＳ２３９：主制御基板３０は、ＲＷＭアクセスプロテクトレジスタにＲＡＭアクセス許可値を出力する処理を実行する。
ステップＳ２４０：主制御基板３０は、設定確認及び不正検出状態をクリアする処理を実行する。
ステップＳ２４１：主制御基板３０は、チェックサム確認処理を呼び出す処理を実行する。

【００９７】
ステップＳ２４２：主制御基板３０は、チェックサムが正常であるか否かを確認する処理を実行する。
その結果、チェックサムが正常であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ２４３を実行する。一方、チェックサムが正常であることを確認できない場合（Ｎｏ）、主制御基板３０は、外部結合子（２）に移行してステップＳ２４６（図４）を実行する。

【００９８】
ステップＳ２４３：主制御基板３０は、ＲＷＭクリアスイッチが押下されていたか否かを確認する処理を実行する。図示しないＲＷＭクリアスイッチは、主制御基板３０に接続されている。
その結果、ＲＷＭクリアスイッチが押下されていたことを確認した場合（Ｙｅｓ）、主制御基板３０は、外部結合子（３）に移行してステップＳ２４７（図４）を実行する。一方、ＲＷＭクリアスイッチが押下されていたことを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ２４４を実行する。

【００９９】
ステップＳ２４４：主制御基板３０は、電源復帰時の初期設定を実行する。
ステップＳ２４５：主制御基板３０は、設定確認処理への移行確認を実行する。
ステップＳ２４５の処理を終えると、主制御基板３０は、外部結合子（１）に移行してステップＳ２５３（図４）を実行する。

【０１００】
ステップＳ２４６：主制御基板３０は、領域外チェック処理を呼び出す処理を実行する。
ステップＳ２４７：主制御基板３０は、作業領域初期化処理を実行する。

【０１０１】
ステップＳ２４８：主制御基板３０は、ＲＷＭ異常であるか否かを確認する処理を実行する。
その結果、ＲＷＭ異常であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ２４９を実行する。一方、ＲＷＭ異常であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ２５０を実行する。

【０１０２】
ステップＳ２４９：主制御基板３０は、ＲＷＭ異常状態指定値をセットする処理を実行する。次に、主制御基板３０は、ステップＳ２５２を実行する。

【０１０３】
ステップＳ２５０：主制御基板３０は、設定変更条件を満たしたか否かを確認する処理を実行する。
その結果、設定変更条件を満たしたことを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ２５１を実行する。一方、設定変更条件を満たしたことを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ２５２を実行する。

【０１０４】
ステップＳ２５１：主制御基板３０は、設定切り替え状態指定をセットする処理を実行する。
ステップＳ２５２：主制御基板３０は、遊技機状態フラグを更新する処理を実行する。
ステップＳ２５３：主制御基板３０は、電源投入時設定処理を呼び出す処理を実行する。
ステップＳ２５４：主制御基板３０は、枠コマンド待機状態をセットする処理を実行す

る。
ステップＳ２５５：主制御基板３０は、デイジーチェーンをリセットする処理を実行する。
ステップＳ２５６：主制御基板３０は、割込みを禁止する処理（割込み禁止命令）を実行する。
ステップＳ２５７：主制御基板３０は、初期値乱数更新処理を呼び出す処理を実行する。
ステップＳ２５９：主制御基板３０は、サブコマンド送信処理を呼び出す処理を実行する。
ステップＳ２６０：主制御基板３０は、割込みを許可する処理（割込み許可命令）を実行する。
ステップＳ２５９の処理を終えると、主制御基板３０は、ステップＳ２５６の処理に戻って処理を継続する。これにより、メインループ処理が開始する。
【０１０５】
〔第２の技術的特徴：割込み処理のコードを削減した構成〕
ここで、本実施形態の第２の技術的特徴について、複数の例を挙げて説明する。なお、本実施形態の技術的特徴は、第１から第５までが存在するが、説明の都合上、順番が入れ替わっている。したがって、第１の技術的特徴及び第３以降の技術的特徴については、第２の技術的特徴より後で説明することとする。
【０１０６】
〔第２の技術的特徴の課題〕
使用領域のプログラム容量（コード量）の削減を実現する。本課題は、管理遊技機１００以外のパチンコ機であっても、記憶手段に使用領域及び領域外を有する場合には該当する。
【０１０７】
〔第２の技術的特徴による解決手段〕
本実施形態では、メインプログラム容量を削減するため、割込み処理内のプログラムコードを削減できる構成とした。図４の割込み待ちメインループ処理（１）は、本解決手段の１つを構成するものである。
【０１０８】
〔割込み待ちメインループ処理（１）の詳細〕
すなわち、割込み待ちメインループ処理（１）は、主制御基板３０が図４の点線枠で囲った部分（符号ＭＬ）をループする処理である。本実施形態では、割込み待ちメインループ処理（１）の中で実行する全ての処理、つまり、初期値乱数更新処理（ステップＳ２５７）及びサブコマンド送信処理（ステップＳ２５９）を割込み禁止命令（ステップＳ２５６）と割込み許可命令（ステップＳ２６０）で囲んでいる。このため、割込み待ちメインループ処理（１）の中で割込み許可状態となるのは、割込み許可命令（ステップＳ２６０）の実行後、プログラム末尾から先頭に戻る命令を実行してから割込み禁止命令（ステップＳ２５６）を実行するまでの期間となる。
【０１０９】
図５は、割込み待ちメインループ処理（１）のプログラムを示す図である。このプログラムは、図４のフローチャートのメインループ部分（点線枠の部分）に対応する。割込み待ちメインループ処理（１）のプログラム（コード）は、以下の通りである。
【０１１０】
「ＣＰＵＩＮＩＴ＿８０：」は、割込み待ちメインループ処理（１）の開始（先頭アドレス）を示すラベルである。
【０１１１】
〔割込み待ちメインループ処理〕
「ＤＩ」は、割込み禁止状態にする命令である。これにより、割込み待ちメインループ処理（１）の開始時に割込み禁止状態となる。
【０１１２】

〔初期値乱数更新処理〕
「ＣＡＬＬＦ　ＩＮＩ＿ＲＮＤ」は、初期値乱数更新処理を呼び出す処理である。本処理では、各種のソフトウェア乱数の初期値をランダムに更新する処理を実行する。
【０１１３】
〔サブコマンド送信処理〕
「ＣＡＬＬＦ　ＳＢＣ＿ＯＵＴ」は、サブコマンド送信処理を呼び出す処理である。本処理では、サブコマンドバッファに格納されているサブコマンドをポート出力する処理を実行する。
【０１１４】
「ＥＩ」は、割込み許可状態にする命令である。これにより、割込み許可状態となる。
「ＪＲ　ＣＰＵＩＮＩＴ＿８０」は、割込み待ちメインループ処理（１）の先頭アドレスにジャンプする命令である。
【０１１５】
〔割込み許可状態となる期間〕
ＥＩ命令はその仕様上、実行した直後は割込みが発生せずに次の命令が実行される。このため、割込み待ちメインループ処理（１）の中で割込みが発生する可能性があるのは、「ＪＲ　ＣＰＵＩＮＩＴ＿８０」命令を実行した直後だけに限定されることになる。
【０１１６】
一方、割込み待ちメインループ処理（１）において、「ＪＲ　ＣＰＵＩＮＩＴ＿８０」を実行する前後でレジスタは一切使用していない。よって、許可状態で発生した割込み処理の中でレジスタの値が書き換わったとしても、割込み待ちメインループ処理（１）側としては何ら問題がなく、割込み処理側でレジスタの退避及び復帰の手当てを考慮する必要がなくなることになる。
【０１１７】
〔割込み待ちメインループ処理（２）〕
図６は、割込み待ちメインループ処理（２）の手順例を示すフローチャートである。本実施形態では、このような割込み待ちメインループ処理（２）を採用することもできる。以下、図４の割込み待ちメインループ処理（１）と異なる点を説明する。
【０１１８】
ステップＳ２５８：主制御基板３０は、主コマンド解析処理を呼び出す処理を実行する。
【０１１９】
次に、図７は、割込み待ちメインループ処理（２）のプログラムを示す図である。このプログラムは、図６のフローチャートに対応する。以下、図５のプログラムと異なる点を説明する。割込み待ちメインループ処理（２）のプログラム（コード）は、以下の通りである。
【０１２０】
〔主コマンド解析処理〕
「ＣＡＬＬＦ　ＰＹ＿ＣＭＤＡ」は、主コマンド解析処理を呼び出す処理である。本処理では、枠制御基板５０から送信されているコマンド（主コマンド）を解析する処理を実行する。
【０１２１】
割込み待ちメインループ処理（２）においても同様に、全ての処理を割込み禁止状態で実行するため、割込み許可状態ではレジスタを使用していない。
【０１２２】
〔比較例〕
図８は、比較例の割込み待ちメインループ処理の手順例を示すフローチャートである。ここでは、本実施形態との比較のため、割込み待ちメインループ処理の中に割込み許可状態で実行される処理が存在している例を挙げるものとする。
【０１２３】
ステップＳ１４１：本実施形態と同様に、割込み禁止命令を実行し、割込み禁止状態と

する。
ステップＳ１４２：本実施形態と同様の初期値乱数更新処理を実行する。
ステップＳ１４３：本実施形態と同様の主コマンド解析処理を実行する。
ステップＳ１４４：本実施形態と同様のサブコマンド送信処理を実行する。
ステップＳ１４６：本実施形態と同様に、割込み許可命令を実行し、次の処理実行時から割込み許可状態とする。
【０１２４】
ステップＳ１４７：比較例において、その他乱数更新処理が実行される。この処理では、ソフトウェア乱数のうち当選図柄の決定に関わらない乱数（リーチグループ決定乱数、リーチモード決定乱数、変動パターン決定乱数等）を更新する。
【０１２５】
図９は、比較例の割込み待ちメインループ処理のプログラムを示す図である。このプログラムは、図８のフローチャートに対応している。以下、本実施形態の割込み待ちメインループ処理（２）のプログラム（図７）と異なる点を説明する。比較例の割込み待ちメインループ処理のプログラム（コード）は、以下の通りである。
【０１２６】
図９中の「ＣＡＬＬＦ　ＳＢＣ＿ＯＵＴ」では、サブコマンド送信処理を呼び出す処理を実行する。そして、次にＥＩ命令を実行して割込み許可状態とする。ここまでのプログラムコードは本実施形態と同様であるが、以下の処理が異なる。
【０１２７】
「ＣＡＬＬＦ　ＥＴＣ＿ＲＮＤ」は、その他乱数更新処理を呼び出す処理である。この処理は、割込み許可状態で実行される。その他乱数更新処理を割込み禁止状態にしていないのは、割込み禁止期間を極力短くするためである。すなわち、その他乱数更新処理まで割込み禁止状態にしてしまうと、メインループ処理の中で割込み禁止期間が長くなり、タイマ割込み処理を十分に実行する猶予が少なくなるからである。
【０１２８】
図１０は、比較例のその他乱数更新処理のプログラムを示す図である。比較例では、割込み許可状態で以下のプログラムが実行されることになる。比較例のその他乱数更新処理のプログラム（コード）は、以下の通りである。
【０１２９】
「ＥＴＣＲＮＤ：」は、その他乱数更新処理の開始を示すラベルである。
【０１３０】
〔特別図柄リーチモード決定乱数更新処理〕
「ＬＤ　Ａ，Ｒ」は、Ｒレジスタ値をセットする命令である。
「ＬＤＱ　ＨＬ，（ＬＯＷ　Ｒ＿ＲＮＤ＿ＴＺ＿＿ＭＯＤ）」は、特別図柄リーチモード決定乱数をロードする命令である。
「ＡＤＤＷＢ　ＨＬ，Ａ」、「ＩＮＣ　ＨＬ」は、特別図柄リーチモード決定乱数を更新する命令である。
「ＬＤ　ＢＣ，＠ＲＮＤ＿ＭＯＤ＿ＭＡＸ＋１」は、特別図柄リーチモード決定乱数上限値をセットする命令である。
【０１３１】
〔割込み発生の想定ポイント〕
プログラムコードは続いているが、以下の〔１〕と〔２〕の間にタイマ割込みが発生した場合を想定する。
〔１〕「ＤＩＶ　ＨＬ，ＢＣ」は、更新結果を特別図柄リーチモード決定乱数上限値で除算する命令である。
〔２〕「ＬＤＱ　（ＬＯＷ　Ｒ＿ＲＮＤ＿ＴＺ＿＿ＭＯＤ），ＢＣ」は、特別図柄リーチモード決定乱数に剰余をセーブする命令である。
【０１３２】
〔割込み処理側のレジスタの退避及び復帰〕

上記〔１〕と〔２〕の間に割込み処理が発生しても、復帰後はその他乱数更新処理が続行されるが、処理の中でＡレジスタの他にＢＣレジスタを使用していることが分かる。このため、タイマ割込み処理中にＢＣレジスタの値が変化した状態で戻ってくると、〔２〕で特別図柄リーチモード決定乱数にセーブするＢＣレジスタの値と、〔１〕で演算した結果設定されたＢＣレジスタは全く無関係な値となる。

したがって、タイマ割込み処理側では、処理の開始時にＢＣレジスタ等を退避し、タイマ割込み処理の終了時にＢＣレジスタ等を復帰する必要がある。

【０１３３】

プログラムコードが以下に続く。

〔特別図柄変動パターン乱数更新処理〕

「ＬＤ　ＤＥ，@ＲＮＤ＿ＰＡＴ＿ＭＡＸ＊２５６＋ＬＯＷ　Ｒ＿ＲＮＤ＿ＴＺ＿＿ＰＡＴ」は、特別図柄変動パターン乱数上限値及び特別図柄変動パターン乱数のアドレスの下位バイトをセットする命令である。

「ＲＳＴ　ＣＯＵＮＴＵＰ」は、カウントアップ処理を呼び出す処理である。

「ＲＥＴ」は、呼び出し元の割込み待ちメインループ処理に復帰する命令である。

【０１３４】

〔本実施形態と比較例との対比〕

以下に、本実施形態と比較例とをプログラム上で対比して説明する。

図１１は、タイマ割込み処理のプログラムを示す図である。ここでは、対比の結果が明白となる部分を取り上げて説明する。タイマ割込み処理のプログラム（コード）は、以下の通りである。

【０１３５】

「ＴＭＲ＿ＩＰＴ：」は、タイマ割込み処理の開始を示すラベルである。

【０１３６】

〔レジスタ退避処理〕

「ＰＵＳＨ　ＧＰＲ」は、ＡＦ，ＢＣ，ＤＥ，ＨＬレジスタを退避する命令である。

ここで、図中に「比較例：必要／本実施形態：不要」と示したように、比較例のプログラムでは本処理が必要となるが、本実施形態では本処理が全く不要となる点で異なる。

【０１３７】

〔ダイナミックポート出力処理〕

「ＣＡＬＬＦ　ＤＹＭ＿ＯＵＴ」は、ダイナミックポート出力処理を呼び出す処理である。

（中略）

【０１３８】

〔安全装置発動管理フェーズ設定処理〕

「ＣＡＬＬＥＸ　Ｅ＿ＳＦＳＥＴ」は、安全装置発動管理フェーズ設定処理を呼び出す処理である。

【０１３９】

〔レジスタ復帰処理〕

「ＰＯＰ　ＧＰＲ」は、ＨＬ，ＤＥ，ＢＣ，ＡＦレジスタを復帰する命令である。

ここで、図中に「比較例：必要／本実施形態：不要」と示したように、比較例のプログラムでは本処理が必要となるが、本実施形態では本処理が全く不要となる点で異なる。

【０１４０】

〔割込許可処理〕

「ＴＭＲ＿ＩＰＴ＿３０」は、割込み許可処理の開始を示すラベルである。

「ＥＩ」は、割込み許可状態にする命令である。

「ＲＥＴＩ」命令により、割込み発生前に復帰する。

【０１４１】

以上の対比から明らかなように、比較例では、割込み処理の開始時にレジスタを退避しておき、終了時にレジスタを復帰させた上で割込み発生前に復帰する処理を実行する必要

があるのに対し、本実施形態では、それらの処理を実行する必要がない。したがって、その分のプログラムコード量を削減することができる。より具体的には、「ＰＵＳＨ　ＧＰＲ」命令と「ＰＯＰ　ＧＰＲ」命令が不要となる。これらはいずれも２バイトであるため、本実施形態では比較例の場合よりも、タイマ割込み処理のコード使用量を合計４バイト削減することが可能となる。

【０１４２】

図１２は、電源断時の退避処理のプログラムを示す図である。ここでも同様に、対比の結果が明白となる部分を取り上げて説明する。電源断時の退避処理のプログラム（コード）は、以下の通りである。

【０１４３】

「ＰＲＦＰＲＯＣ：」は、電源断時の退避処理の開始を示すラベルである。

【０１４４】

〔電源断予告信号検出確認処理〕

「ＰＵＳＨ　ＡＦ」は、ＡＦレジスタを退避する命令である。ここで、図中に「比較例：必要／本実施形態：不要」と示したように、比較例のプログラムでは本命例の実行が必要となるが、本実施形態では不要となる。

「ＪＩＢＩＴ　Ｚ，@ＩＮ３＿ＰＷＦ＿ＰＯＳ，（ＬＯＷ　@ＩＮ３＿ＰＲＴ），ＰＲＦＰＲＯＣ＿１０」は、入力ポート３の電源断予告信号ビットが０であれば、「ＰＲＦＰＲＯＣ＿１０」に分岐する命令である。

「ＰＯＰ　ＡＦ」は、ＡＦレジスタを復帰する命令であるが、ここでも図中に「比較例：必要／本実施形態：不要」と示したように、比較例のプログラムでは本命例の実行が必要となるが、本実施形態では不要となる。

「ＥＩ」命令で割込み許可状態とする。

「ＲＥＴＩ」命令で電源断時の退避処理発生前に復帰する。

【０１４５】

〔出力ポートクリア処理〕

「ＰＲＦＰＲＯＣ＿１０：」は、出力ポートクリア処理の開始を示すラベルである。

「ＯＵＴ　（ＬＯＷ　@ＯＴ１＿ＰＲＴ），@ＯＴ１＿ＬＣＥ＿ＢＩＴ」は、出力ポート１に発射許可信号を出力する命令である。

「ＬＤＦ　ＨＬ，Ｄ＿ＯＵＴ＿ＰＲＴ」は、出力ポートアドレステーブルのアドレスをセットする命令である。

「ＬＤ　Ｂ，@ＰＲＴ＿ＣＬＲ＿ＬＯＰ」は、出力ポート初期化ループカウンタをセットする命令である。

「ＸＯＲ　Ａ，Ａ」は、［００Ｈ］をセットする命令である。

（以下略）

【０１４６】

以上の対比から明らかなように、比較例では、電源断時の退避処理の開始時にレジスタを退避しておき、入力ポート３の電源断予告信号ビットが０でなくなった（電源断状態ではなくなった）場合、電断割込み発生前のプログラムアドレスに戻る準備として、レジスタを復帰させる処理を実行する必要があるのに対し、本実施形態では、それらの処理を実行する必要がない。したがって、その分のプログラムコード量を削減することができる。より具体的には、「ＰＵＳＨ　ＡＦ」命令と「ＰＯＰ　ＡＦ」命令が不要となる。これらはいずれも１バイトであるため、本実施形態では比較例の場合よりも、電源断時の退避処理のコード使用量を合計２バイト削減することが可能となる。

【０１４７】

〔フローチャートでの対比〕

図１３は、タイマ割込み処理の手順例を示すフローチャートである。図１３のフローチャートは、図１１のプログラムに一部対応している。図１３では、比較例では実行する必要があるが、本実施形態では実行する必要がない手順を点線で示している。

【０１４８】

〔本実施形態で不要のステップ〕
上記のように、本実施形態ではレジスタ退避処理（ステップＳ３００）が不要となる。したがって、ダイナミックポート出力処理（ステップＳ３０２）から開始する。
【０１４９】
ステップＳ３０２：主制御基板３０は、ダイナミックポート出力処理を実行する。この処理では、性能表示モニタ３７や図示しない統合表示基板等に実装された各ランプの点灯をダイナミック点灯方式で制御するために、コモン単位でのポート出力を行う。具体的には、主制御基板３０は、出力ポートをクリアした後、選択されたコモンに対応するコモン用のポート出力バッファに、生成されたコモン出力用データを出力する。出力される内容は、前回のタイマ割込みで設定された内容である。
【０１５０】
ここで、各コモン用のポート出力バッファに出力されるデータは、タイマ割込み処理が発生する毎に、このダイナミックポート出力処理において１コモンずつ順繰りにポート出力される。例えば、次回に実行されるタイマ割込み処理ではコモン１用として格納されたデータがポート出力され、次々回に実行されるタイマ割込み処理ではコモン２用として格納されたデータがポート出力される、という具合に各コモン用のポート出力バッファに格納されたデータが１つずつ順番に処理されていく。これにより、所定の表示態様（図柄の変動表示や停止表示、作動記憶数表示、遊技状態表示等を行う態様）や性能表示モニタ３７を構成する各ランプがコモン単位で順繰りに駆動され、ダイナミック点灯方式により点灯制御される。
【０１５１】
ステップＳ３０３：主制御基板３０は、ポート入力処理を実行する。この処理では、入力ポート情報に基づき最新のスイッチ状態を正確に取得するために、主制御基板３０は、パラレルＩ／Ｏポートから各種スイッチ信号の入力値と前回入力値の反転結果値との論理積を入力ポートオン検出フラグに格納する。この結果、入力ポートオン検出フラグの値（ＯＮ／ＯＦＦ）により、各種スイッチ信号の前回からの変化を踏まえた正確な入力状態を把握することが可能となる。各種スイッチ信号には、入賞口スイッチ３２からの入賞検出信号、不正検出スイッチ（盤面磁気検出第１センサ３４、盤面磁気検出第２センサ３５、電波検出センサ３６等）からの検出信号等が含まれる。
【０１５２】
ステップＳ３０３ａ：主制御基板３０は、遊技機状態フラグが「００Ｈ（遊技可能状態）」であるか否かを確認する。
【０１５３】
その結果、遊技機状態フラグが「００Ｈ（遊技可能状態）」であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ３０４を実行する。一方、遊技機状態フラグが「００Ｈ（遊技可能状態）」であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ３０３ｂを実行する。
【０１５４】
このような判断処理を実行している理由は、設定変更中や設定確認中等である場合、通常の遊技に係る処理を実行しないようにするためである。そして、設定変更中や設定確認中等において、タイマ割込み処理内の通常の遊技に係る処理を実行しないことで、主制御基板３０のプログラムの負荷を軽減することができる。
【０１５５】
ステップＳ３０３ｂ：主制御基板３０は、遊技機状態フラグが「０３Ｈ以上（致命的なエラー状態）」であるか否かを確認する。
【０１５６】
その結果、遊技機状態フラグが「０３Ｈ以上（致命的なエラー状態）」であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ３０３ｃを実行せずに、ステップＳ３１８を実行する。一方、遊技機状態フラグが「０３Ｈ以上（致命的なエラー状態）」であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ３０３ｃを実行

する。
【０１５７】
　このような判断処理を実行している理由は、致命的なエラー状態が発生している場合には、設定変更中や設定確認中に移行させないためである。
【０１５８】
　ステップＳ３０３ｃ：主制御基板３０は、設定変更処理（設定関連処理）を実行する。この処理を実行することにより、主制御基板３０は、設定の値を変更する際に実行される設定変更処理又は設定の値を確認する際に実行される設定確認処理の少なくとも一方を含む設定関連処理を実行することができる（設定関連処理実行手段）。設定関連処理は、設定変更中又は設定確認中に実行される処理が含まれる。そして、設定変更処理を終えると、主制御基板３０は、ステップＳ３１８を実行する。ステップＳ３１８に移行する理由は、外部情報管理処理において外部信号（セキュリティ信号）を出力するためである。
【０１５９】
　このような処理を実行することにより、主制御基板３０は、設定関連処理及びベース関連処理のうち、一方の処理の実行中には、他方の処理の少なくとも一部の処理の実行を制限することができる（制限手段）。つまり、主制御基板３０は、設定変更中又は設定確認中である場合（ステップＳ３０３ａ：Ｙｅｓ）、ベース関連処理の少なくとも一部の処理（ステップＳ３０４～ステップＳ３１７、特に、ステップＳ３０８）を実行しないようにすることができる。
【０１６０】
　ステップＳ３０４：主制御基板３０は、タイマ更新処理を実行する（特別情報継続送信処理実行手段）。この処理では、主制御基板３０は、遊技に用いる各種タイマ（図柄の変動時間・停止時間や電動役物の開放時間・閉鎖時間等を管理するタイマ）の他、外部情報用の各種タイマ、セキュリティ信号出力用タイマ等のカウンタを更新する処理（減算処理等）を実行する。
【０１６１】
　ステップＳ３０５：主制御基板３０は、初期値乱数更新処理を実行する。処理の内容は、割込み待ちメインループ処理の初期値乱数更新処理（図４のステップＳ２５７）と同様である。
【０１６２】
　ステップＳ３０６：主制御基板３０は、当り図柄乱数更新処理を実行する。この処理では、主制御基板３０は特別図柄及び普通図柄の抽選用の各種乱数を発生させるためのカウンタの値を更新する。各カウンタの値は、ＲＡＭのカウンタ領域にてインクリメントされ、それぞれ規定の範囲内でループする。各種乱数には、例えば大当り図柄乱数等が含まれる。
【０１６３】
　ステップＳ３０７：主制御基板３０は、スイッチ管理処理を実行する。この処理では、先のポート入力処理（ステップＳ３０３）で入力したスイッチ信号のうち、入賞口スイッチ３２からの入賞検出信号に基づいて遊技中に発生した事象の判定を行い、それぞれ発生した事象に応じた処理を実行する。
【０１６４】
　ステップＳ３０８：主制御基板３０は、加算数算定処理を実行する（ベース関連処理実行手段）。加算数算定処理は、ベース関連処理である。本処理は、使用領域の処理である。この処理において、主制御基板３０は、遊技状態と、入賞口スイッチ３２、アウトスイッチ３３の状態を確認し、ベース計算用に「領域外のＲＡＭに加算すべき値（４ｍｓの間に発生した賞球数、アウト数等の加算数）」を算出する。具体的には、主制御基板３０は、発射球数Ａ（通常アウト加算数）、発射球数Ｂ（総アウト加算数）、獲得球数（通常賞球加算数）の値を決定する処理を実行する。
【０１６５】
　ステップＳ３０９，ステップＳ３１０：主制御基板３０は、特別遊技管理処理及び普通

遊技管理処理を実行する。これらの処理は、管理遊技機１００における遊技を具体的に進行させるためのものである。

【０１６６】

特別遊技管理処理（ステップＳ３０９）では、主制御基板３０は、第１特別図柄又は第２特別図柄に対応する内部抽選の実行を制御したり、図示しない第１特別図柄表示装置及び第２特別図柄表示装置による変動表示や停止表示を決定したり、その表示結果に応じて図示しない第１可変入賞装置及び第２可変入賞装置の作動を制御したりする。

【０１６７】

また、普通遊技管理処理（ステップＳ３１０）では、主制御基板３０は、図示しない普通図柄表示装置による変動表示や停止表示を決定したり、その表示結果に応じて図示しない可変始動入賞装置の作動を制御したりする。

【０１６８】

ステップＳ３１１：主制御基板３０は、状態管理処理を実行する。この処理では、主制御基板３０は、入賞頻度の異常やベース異常等のエラー状態が発生していないか否かのチェックを行う。エラー状態を検知した場合、主制御基板３０は、遊技場のホールコンピュータに対してはセキュリティ信号の出力により、また、演出制御基板４０に対しては所定のエラーコマンドの送信により、異常が発生したことを通知する。なお、この処理において、主制御基板３０は、扉開放や不正検出（磁石、電波、振動等）に関するエラー処理を実行してもよい。

【０１６９】

ステップＳ３１２：主制御基板３０は、入賞口スイッチ処理を実行する。この処理では、先のポート入力処理（ステップＳ３０３）において入賞口スイッチ３２から入力された入賞検出信号に基づき格納した各入力ポートオン検出フラグがＯＮの場合に、それぞれの対象となる賞球制御カウンタを１加算して更新する。

【０１７０】

ステップＳ３１３：主制御基板３０は、払出制御管理処理を実行する。この処理では、主制御基板３０はまず払出コマンドバッファが空でないか（送信すべき払出コマンドがセットされているか）否かを確認し、空でない（払出コマンドがセットされている）場合は、払出コマンドバッファに出力された各種払出コマンドを枠制御基板５０に対して送信する。例えば、電源投入時の起動モードを示す払出コマンドは、ＣＰＵ初期化処理の過程でセットされ、払出コマンドバッファに出力され、この起動モードを示す払出コマンドがこの処理で送信される。一方、払出コマンドバッファが空である場合は、賞球の払い出しを指示するための処理に進む。主制御基板３０は賞球制御カウンタが０でないか否かを確認し、賞球制御カウンタが０でない場合は、このカウンタに対応する賞球個数を指示する賞球指定の払出コマンドを枠制御基板５０に対して送信する。

【０１７１】

また、特に電源投入時においては、ＣＰＵ初期化処理の過程でセットされた払出コマンドが正常に送信された場合、主制御基板３０はこれを契機として発射許可信号をオンにする。具体的には、主制御基板３０は電源投入時に出力した払出コマンドバッファをクリアするとともに、出力ポートの特定ビットをセットすることで発射許可信号をオンにする。これにより、電源投入後の正常動作を確認した上で遊技球の発射が許可され、この発射許可信号が枠制御基板５０に送られることにより、遊技球の発射が可能な状態となる。

【０１７２】

また、主制御基板３０は、払出制御管理処理において、演出制御基板４０に対して賞球個数の内容を伝達する賞球内容コマンドを出力する。図示しない第１可変入賞装置又は第２可変入賞装置に対応する入賞口スイッチ３２から入賞検出信号が入力された場合、第１利益（例えば、遊技球１５個分）に対応する賞球内容コマンドを生成する。なお、コマンドを生成するとは、コマンドをサブコマンド送信用バッファに格納することを意味する（以下、同様）。また、図示しない普通入賞口に対応する入賞口スイッチ３２から入賞検出信号が入力された場合、第２利益（例えば、遊技球１０個分）に対応する賞球内容コマン

ドを生成する。賞球内容コマンドは、割込み待ちメインループ処理のサブコマンド送信処理（図４中のステップＳ２５９）において演出制御基板４０に送信される。

【０１７３】

ステップＳ３１５：主制御基板３０は、試験信号管理処理を実行する。この処理は、領域外の処理とすることができる。この処理では、主制御基板３０が自己の内部状態（例えば、普通図柄遊技管理状態、特別図柄遊技管理状態、発射位置指定状態、大当り中、小当り中、確率変動機能作動中、時間短縮機能作動中等）を表す各種の試験信号を生成し、これらをポート出力要求バッファに格納する。この試験信号により、例えば主制御基板３０の外部でその内部状態を試験することができる。

【０１７４】

ステップＳ３１６：主制御基板３０は、ＬＥＤ表示設定処理を実行する。この処理では、主制御基板３０は、図示しない普通図柄表示装置、普通図柄作動記憶ランプ、第１特別図柄表示装置、第２特別図柄表示装置、第１特別図柄作動記憶ランプ、第２特別図柄作動記憶ランプ、遊技状態表示装置等に含まれるＬＥＤを点灯制御するためのコモン出力用データを生成する処理を実行する。なお、性能表示モニタ３７に関連する処理は、性能表示モニタ制御処理において実行する。

【０１７５】

より詳細には、主制御基板３０は、特別遊技管理処理（ステップＳ３０９）や普通遊技管理処理（ステップＳ３１０）において決定された図柄の変動表示や停止表示、作動記憶数表示、遊技状態表示等に基づいて、対応する態様で各ランプを点灯させるための駆動信号を、コモン出力用データとして生成する。

【０１７６】

ステップＳ３１７：主制御基板３０は、ソレノイドデータ設定処理を実行する。この処理では、主制御基板３０は、特別遊技管理処理（ステップＳ３０９）や普通遊技管理処理（ステップＳ３１０）等において生成された図示しない普通電動役物ソレノイド、第１大入賞口ソレノイド及び第２大入賞口ソレノイドの各駆動信号、試験信号等を合わせて（合成して）ポート出力バッファに格納する。この後、主制御基板３０は、ソレノイドデータポート出力処理を実行する。この処理では、主制御基板３０は各出力バッファ（ポート出力バッファ）に値が格納されているかを確認し、値が格納されている場合はポート出力する。例えば、ポート出力バッファに格納された各ソレノイドの各駆動信号をポート出力する。この場合、各駆動信号が対応する各ソレノイドに送信され、各ソレノイドは駆動信号に応じた動作を実行する。

【０１７７】

ステップＳ３１８：次に主制御基板３０は、外部情報管理処理を実行する。この処理では、主制御基板３０は図示しない専用ユニットを通じて遊技場のホールコンピュータに外部出力する外部情報信号（例えば賞球情報、扉開放情報、図柄確定回数情報、大当り情報、始動口情報、セキュリティ信号等）をポート出力要求バッファに格納する。

外部情報管理処理では、「設定変更中又は設定確認中」等である場合に、外部信号（セキュリティ信号）を出力する。

【０１７８】

ここで、セキュリティ信号とは、何かしらの異常状態が発生した場合、設定変更状態、又は設定確認状態に移行した場合に出力される信号である。

【０１７９】

主制御基板３０は、遊技機状態フラグ（遊技機状態情報）に格納されている値を参照してセキュリティ信号を送信するか否かを決定する（特別情報送信処理実行手段）。例えば、遊技機状態フラグに、遊技可能状態に対応する値（００Ｈ）が格納されている場合、セキュリティ信号を送信しないと判断し、遊技可能状態に対応する値（００Ｈ）以外の値（０１Ｈ～０５Ｈ）が格納されている場合、セキュリティ信号を送信すると判断する。

【０１８０】

ステップＳ３１９：主制御基板３０は、性能表示モニタ制御処理を実行する（ベース関

連処理実行手段）。性能表示モニタ制御処理は、ベース関連処理である。本処理は、領域外の処理である。この処理において、主制御基板３０は、性能表示モニタ３７の表示内容の切替等の処理を実行したり、性能としてベース値、役物比率、連続役物比率を算定したり、算定したベース値、役物比率、連続役物比率を性能表示モニタ３７に表示するための駆動信号を、コモン出力用データとして生成したりする処理を実行する。なお、処理の詳細は、後述する。

【０１８１】

〔本実施形態で不要のステップ〕

ここで上記のように、本実施形態ではレジスタ復帰処理（ステップＳ３２０）が不要となる。

【０１８２】

ステップＳ３２１：主制御基板３０は、安全装置発動管理フェーズ設定処理を実行する。この処理では、安全装置発動管理フェーズを設定する。安全装置は、１日の払出数が規定数に達した場合に発動し、管理遊技機１００を遊技停止（打止）にする。

【０１８３】

ステップＳ３２２：主制御基板３０は、割込み許可処理を実行する。ここで割込みが許可されることにより、タイマ割込み処理の次ステップ以降を実行している間に他の割込みが発生することが可能となる。このように、タイマ割込み処理は多重割込みが許可されている。

【０１８４】

以上の手順を実行すると、主制御基板３０はタイマ割込み処理を終了して割込み待ちメインループ処理に復帰する。上記のように、割込み待ちメインループ処理ではレジスタを使用中に割込み許可状態とならないので、復帰した先では、特にレジスタに対する手当てを考慮する必要がない。

【０１８５】

図１４及び図１５は、電源断時の退避処理の手順例を示すフローチャートである。このフローチャートは、図１２のプログラムに一部対応している。本処理は、主制御基板３０（主制御ＣＰＵ３１）が実行する。

【０１８６】

〔本実施形態で不要のステップ〕

上記のように、本実施形態では、図中に点線で示したレジスタ退避処理（ステップＳ２７０）が不要となる。したがって、電源断予告信号検出確認処理（ステップＳ２８０）から開始する。

【０１８７】

〔電源断予告信号検出確認処理〕

ステップＳ２８０：主制御基板３０は、電源断予告信号がＯＮであるか否かを確認（再確認）する処理を実行する。

その結果、電源断予告信号がＯＮであることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ２８２を実行する。一方、電源断予告信号がＯＮであることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ２８１を実行する。

【０１８８】

ステップＳ２８１：主制御基板３０は、割込みを許可する処理を実行する。

ステップＳ２８１の処理を終えると、主制御基板３０は、割込み前のアドレスに戻る。

【０１８９】

〔本実施形態で不要のステップ〕

ここで、上記のように本実施形態では、電源断状態ではなくなったために割込み前に復帰する場合でも、図中に点線で示したレジスタ復帰処理（ステップＳ２７２）が不要となる。

【０１９０】

ステップＳ２８２：主制御基板３０は、出力ポート１に発射許可信号を出力する処理を

実行する。
　ステップＳ２８３：主制御基板３０は、出力ポートアドレステーブルのアドレスをセットする処理を実行する。
　ステップＳ２８４：主制御基板３０は、出力ポート初期化ループカウンタをセットする処理を実行する。
　ステップＳ２８５：主制御基板３０は、Ａレジスタに［００Ｈ］をセットする処理を実行する。
　ステップＳ２８６：主制御基板３０は、対象アドレスの内容をロードする処理を実行する。
　ステップＳ２８７：主制御基板３０は、出力ポートに［００Ｈ］を出力する処理を実行する。
　ステップＳ２８８：主制御基板３０は、ＨＬレジスタをインクリメントする処理を実行する。
【０１９１】
　ステップＳ２８９：主制御基板３０は、ループ継続であるか（すなわち、ステップＳ２８６からステップＳ２８８までの処理を再び実行するか）否かを確認する処理を実行する。
　その結果、ループ継続であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ２８６を実行する。一方、ループ継続であることを確認できない場合（Ｎｏ）、主制御基板３０は、外部結合子（１）に移行してステップＳ２９０（図１５）を実行する。
【０１９２】
　ステップＳ２９０：主制御基板３０は、チェックサム算定処理を呼び出す処理を実行する。
　ステップＳ２９１：主制御基板３０は、ＲＷＭアクセスプロテクトレジスタにＲＡＭアクセス禁止値を出力する処理を実行する。
　ステップＳ２９２：主制御基板３０は、電源断予告信号確認設定値をセットする処理を実行する。
【０１９３】
　ステップＳ２９３：主制御基板３０は、電源断予告信号がオンであるか否かを確認する処理を実行する。
　その結果、電源断予告信号がオンであることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ２９２を実行する。一方、電源断予告信号がオンであることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ２９４を実行する。
【０１９４】
　ステップＳ２９４：主制御基板３０は、ＨＬレジスタをデクリメントする処理を実行する。
【０１９５】
　ステップＳ２９５：主制御基板３０は、第２ゼロフラグが０であるか否かを確認する処理を実行する。
　その結果、第２ゼロフラグが０であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ２９３を実行する。一方、第２ゼロフラグが０であることを確認できない場合（Ｎｏ）、主制御基板３０は、［０００００Ｈ］番地にジャンプする処理を実行する。
【０１９６】
〔第３の技術的特徴：特定命令を使用したチェックサム算定処理の構成〕
　ここで、本実施形態の第３の技術的特徴について説明する。第３の技術的特徴は、電源断時の退避処理の中で実行されるチェックサム算定処理（図１５のステップＳ２９０）、及びＣＰＵ初期化処理の中で実行されるチェックサム確認処理（図３のステップＳ２４１）に特定の命令を使用した構成である。
【０１９７】
〔第３の技術的特徴の課題〕
　電源断時のチェックサムの算定、電源再投入時のチェックサムの確認にかかる時間を短

縮することを課題とする。
　すなわち、管理遊技機１００のような遊技機では、電源投入時にバックアップチェックデータ及びチェックサムの確認を行い、異常であった場合（初回起動時を含む）は全てのＲＷＭをクリアしている。
　このとき、バックアップチェックデータ及びチェックサムの設定は、電源断時の退避処理で作成、保存を行うことになるが、既に電源供給は絶たれているため、チェックサムの算定及び確認については、いずれも可能な限り短い時間で処理を終わらせる必要がある。
【０１９８】
〔チェックサムの算定方法〕
　チェックサムの算定方法は、「スタック領域の２バイトを除く使用領域の全領域と領域外の使用ＲＷＭを減算した結果の下位１バイト」とする。例えば、算定対象のＲＷＭの値が１，２，３，４である場合は、０－１－２－３－４＝－１０となる。
　チェックサムは符号なしの１バイトの整数（０～２５５）であるため、ＲＷＭに保存される値は－１０＋２５６＝２４６となる。
【０１９９】
　本発明の発明者等は鋭意研究を重ねた結果、特定の命令をチェックサム算定処理に使用することで、処理を極めて短時間内に終了させることができるとの知見を得た。特定の命令は、主制御基板３０（主制御ＣＰＵ３１）が実行可能なＡＤＤＩＲ命令、ＡＤＤＩＲＳ命令である。これらの命令をチェックサムの算定処理に使用することで、処理時間（ステート数）を最短にすることができる。以下、第３の技術的特徴について比較例と対比しつつ説明する。
【０２００】
〔比較例のチェックサム算定処理〕
　図１６は、比較例のチェックサム算定処理のプログラムを示す図である。比較例では、本実施形態で使用する特定の命令を使用しない構成である。なお、図１６のプログラムコード中、各行に付記している括弧書きの数値は、主制御ＣＰＵ３１が命令を実行するために必要とする内部クロック数（ステート数）を示している。比較例のチェックサム算定処理のプログラム（コード）は、以下の通りである。
【０２０１】
　「Ｅ＿ＣＬＳＵＭ＿０１：」は、チェックサム算定処理の開始を示すラベルである。
（中略）
【０２０２】
〔使用領域チェックサム算定処理〕
（１）「ＸＯＲ　Ａ，Ａ」は、Ａレジスタに［００Ｈ］をセットする命令である。ステート数は１である。
（３）「ＬＤ　ＨＬ，＠ＲＭ＿ＲＡＭＡＤＲ」は、ＨＬレジスタに作業領域先頭アドレスをセットする命令である。ステート数は３である。
（３）「ＬＤ　ＢＣ，＠ＣＨＫ＿ＡＲＥ」は、ＢＣレジスタにチェックサム算定領域数をセットする命令である。ステート数は３である。
【０２０３】
〔ループ処理〕
　「Ｅ＿ＣＬＳＵＭ＿１０：」は、ループ処理の先頭アドレスを示すラベルである。
（２）「ＳＵＢ　Ａ，（ＨＬ）」は、ＨＬレジスタに現状セットされているアドレスのＲＷＭチェックサムを算定する命令である。ステート数は２である。
（１）「ＩＮＣ　ＨＬ」は、ＨＬレジスタをインクリメントし、次のＲＷＭアドレスをセットする命令である。ステート数は１である。
（２）「ＤＥＣ　ＢＣ」は、チェックサム算定領域数を１減算する命令である。チェックサム算定領域を全て算定するとＢＣレジスタが０になる。ステート数は２である。
（３／２）「ＪＲ　ＮＴＺ，Ｅ＿ＣＬＳＵＭ＿１０」は、チェックサム算定領域を全て算定しておらず、ループ継続であれば先頭に分岐する命令である。ステート数は、分岐の場

合は３となり、非分岐の場合は２となる。
【０２０４】
〔領域外チェックサム算定処理〕
（２）「ＬＤＱ　ＨＬ，ＬＯＷ　＠ＲＭ＿ＥＸＴＡＤＲ」は、ＨＬレジスタに領域外作業領域先頭アドレスをセットする命令である。ステート数は２である。
（２）「ＬＤ　Ｂ，＠ＥＸＤ＿ＵＳＥ＿ＥＲＷ」は、Ｂレジスタに領域外ＲＷＭ使用領域数をセットする命令である。ステート数は２である。
【０２０５】
「Ｅ＿ＣＬＳＵＭ＿２０：」は、ループ処理の先頭アドレスを示すラベルである。
（２）「ＳＵＢ　Ａ，（ＨＬ）」は、ＨＬレジスタに現状セットされているアドレスのＲＷＭチェックサムを算定する命令である。ステート数は２である。
（１）「ＩＮＣ　ＨＬ」は、ＨＬレジスタをインクリメントし、次のＲＷＭアドレスをセットする命令である。ステート数は１である。
（３／２）「ＤＪＮＺ　Ｅ＿ＣＬＳＵＭ＿２０」は、領域外ＲＷＭ使用領域を全て算定しておらず（Ｂレジスタを減算して０でなく）、ループ継続であれば先頭に分岐する命令である。ステート数は、分岐の場合は３となり、非分岐の場合は２となる。
【０２０６】
〔サムチェックバッファ設定処理〕
（３）「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＳＵＭ），Ａ」は、サムチェックバッファに算定結果をセーブする命令である。ステート数は３である。
【０２０７】
〔比較例のチェックサム算定処理必要ステート数〕
以上より、例えばチェックサム対象となる使用領域のＲＷＭが５１０バイト、領域外のＲＷＭが１７０バイトであるとすると、チェックサムの算定に必要な総ステート数は、１＋３＋３＋（５１０－１）＊（２＋１＋２＋３）＋（２＋１＋２＋２）＋２＋２＋（１７０－１）＊（２＋１＋３）＋（２＋１＋２）＋２＝５１１２となる。したがって、主制御ＣＰＵ３１の内部システムクロックが１６ＭＨｚである場合、比較例のチェックサム算定処理の実行時間は０．３１９５ｍｓとなる。
【０２０８】
〔チェックサムの確認方法〕
チェックサムの確認方法は、「スタック領域の２バイトを除く使用領域の全領域と領域外の使用ＲＷＭを減算した結果の下位１バイトが０であるか」とする。例えば、ＲＷＭの値が１，２，３，４であり、チェックサムとして２４６が保存されている場合は、１＋２＋３＋４＋２４６＝２５６となり、下位１バイトは０となるので、この場合はチェックサムが正しく保存されていると確認することができる。
【０２０９】
〔比較例のチェックサム確認処理〕
図１７は、比較例のチェックサム確認処理のプログラムを示す図である。ここでも同様に、比較例では、本実施形態で使用する特定の命令を使用しない構成である。また、図１７のプログラムコード中、各行に付記している括弧書きの数値はステート数を示している。比較例のチェックサム確認処理のプログラム（コード）は、以下の通りである。
【０２１０】
「Ｅ＿ＣＫＳＵＭ＿０１：」は、チェックサム確認処理の開始を示すラベルである。
（中略）
【０２１１】
〔使用領域チェックサム算定処理〕
（１）「ＸＯＲ　Ａ，Ａ」は、Ａレジスタに［００Ｈ］をセットする命令である。ステート数は１である。
（３）「ＬＤ　ＨＬ，＠ＲＭ＿ＲＡＭＡＤＲ」は、ＨＬレジスタに作業領域先頭アドレスをセットする命令である。ステート数は３である。

（３）「ＬＤ　ＢＣ，＠ＣＨＫ＿ＡＲＥ」は、ＢＣレジスタにチェックサム算定領域数をセットする命令である。ステート数は３である。
【０２１２】
　「Ｅ＿ＣＫＳＵＭ＿１０：」は、ループ処理の先頭アドレスを示すラベルである。
（２）「ＡＤＤ　Ａ，（ＨＬ）」は、ＨＬレジスタに現状セットされているアドレスのＲＷＭチェックサムを算定する命令である。ステート数は２である。
（１）「ＩＮＣ　ＨＬ」は、ＨＬレジスタをインクリメントし、次のＲＷＭアドレスをセットする命令である。ステート数は１である。
（２）「ＤＥＣ　ＢＣ」は、チェックサム算定領域数を１減算する命令である。チェックサム算定領域を全て算定するＢＣレジスタが０になる。ステート数は２である。
（３／２）「ＪＲ　ＮＴＺ，Ｅ＿ＣＫＳＵＭ＿１０」は、チェックサム算定領域を全て算定しておらず、ループ継続であれば先頭に分岐する命令である。ステート数は、分岐の場合は３となり、非分岐の場合は２となる。
【０２１３】
〔領域外チェックサム算定処理〕
（２）「ＬＤＱ　ＨＬ，ＬＯＷ　＠ＲＭ＿ＥＸＴＡＤＲ」は、ＨＬレジスタに領域外作業領域先頭アドレスをセットする命令である。ステート数は２である。
（２）「ＬＤ　Ｂ，＠ＥＸＤ＿ＵＳＥ＿ＥＲＷ」は、Ｂレジスタに領域外ＲＷＭ使用領域数をセットする命令である。ステート数は２である。
【０２１４】
　「Ｅ＿ＣＫＳＵＭ＿２０：」は、ループ処理の先頭アドレスを示すラベルである。
（２）「ＡＤＤ　Ａ，（ＨＬ）」は、ＨＬレジスタに現状セットされているアドレスのＲＷＭチェックサムを算定する命令である。ステート数は２である。
（１）「ＩＮＣ　ＨＬ」は、ＨＬレジスタをインクリメントし、次のＲＷＭアドレスをセットする命令である。ステート数は１である。
（３／２）「ＤＪＮＺ　Ｅ＿ＣＫＳＵＭ＿２０」は、領域外ＲＷＭ使用領域を全て算定しておらず（Ｂレジスタを減算して０でなく）、ループ継続であれば先頭に分岐する命令である。ステート数は、分岐の場合は３となり、非分岐の場合は２となる。
【０２１５】
　「Ｅ＿ＣＫＳＵＭ＿３０」は、チェックサム結果フラグ設定処理の開始を示すラベルである。なお、チェックサムが異常であるかの判定処理は、使用領域からチェックサム結果フラグ（Ｒ＿ＥＲＷ＿ＢＲＳ）を参照して行うが、ここではプログラムを省略する。
【０２１６】
〔チェックサム結果フラグ設定処理〕
（３）「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＢＲＳ），Ａ」は、チェックサム結果フラグに値をセーブする命令である。ステート数は３である。
【０２１７】
〔比較例のチェックサム確認処理必要ステート数〕
　以上より、電源投入時のチェックサム確認処理でチェックサムの算定に必要な総ステート数は、電源断時と同じく５１１２ステートとなる。したがって、比較例のチェックサム確認処理の実行時間は同じく０．３１９５ｍｓとなる。
【０２１８】
〔本実施形態のチェックサム算定処理〕
　図１８は、本実施形態のチェックサム算定処理のプログラムを示す図である。本実施形態では、チェックサムの算定及び確認にＡＤＤＩＲ命令及びＡＤＤＩＲＳ命令を使用している。ＡＤＤＩＲ命令及びＡＤＤＩＲＳ命令は、ＨＬレジスタが指定した開始番地から、ＢＣレジスタ（ＡＤＤＩＲＳ命令はＢレジスタ）で指定したバイト数のＲＷＭの和を計算する命令である。なお、和ではなく差を計算する命令は存在しないため、電源断時の退避処理でサムチェックバッファの値を設定する際は、一旦、和を計算してから２の補数を取る（符号を反転する）こととする。以下、比較例と同じ命令については説明を省略し、本

実施形態に特有の命令を中心に説明する。本実施形態のチェックサム算定処理のプログラム（コード）は、以下の通りである。

【０２１９】

「Ｅ＿ＣＬＳＵＭ＿０１：」は、チェックサム算定処理の開始を示すラベルである。

（中略）

【０２２０】

〔使用領域チェックサム算定処理〕

（１）「ＸＯＲ　Ａ，Ａ」：比較例と同じ。

（３）「ＬＤ　ＨＬ，@ＲＭ＿ＲＡＭＡＤＲ」：比較例と同じ。

（３）「ＬＤ　ＢＣ，@ＣＨＫ＿ＡＲＥ」比較例と同じ。

（３）「ＡＤＤＩＲ」は、ブロックサーチ命令である。具体的には、開始アドレス（@ＲＭ＿ＲＡＭＡＤＲ）からチェックサム算定領域数（@ＣＨＫ＿ＡＲＥ）で指定したバイト数のＲＷＭの和を計算する。ここでは、加算１バイトあたりのステート数が３である。

【０２２１】

〔領域外チェックサム算定処理〕

（２）「ＬＤＱ　ＨＬ，ＬＯＷ　@ＲＭ＿ＥＸＴＡＤＲ」：比較例と同じ。

（２）「ＬＤ　Ｂ，@ＥＸＤ＿ＵＳＥ＿ＥＲＷ」：比較例と同じ。

（３）「ＡＤＤＩＲＳ」は、ブロックサーチ命令である。具体的には、開始アドレス（@ＲＭ＿ＥＸＴＡＤＲ）からチェックサム算定領域数（@ＥＸＤ＿ＵＳＥ＿ＥＲＷ）で指定したバイト数のＲＷＭの和を計算する。ここでは、加算１バイトあたりのステート数が３である。

【０２２２】

〔サムチェックバッファ設定処理〕

（２）「ＮＥＧ」は、２の補数をセットする命令である。ステート数は２である。

（３）「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＳＵＭ），Ａ」は、サムチェックバッファに値をセーブする命令である。ステート数は３である。

【０２２３】

〔本実施形態のチェックサム算定処理必要ステート数〕

以上より、比較例と同じくチェックサム対象となる使用領域のＲＷＭが５１０バイト、領域外のＲＷＭが１７０バイトであるとすると、本実施形態でチェックサムの算定に必要な総ステート数は、１＋３＋３＋５１０＊３＋２＋２＋１７０＊３＋２＋３＝２０５６となる。したがって、主制御ＣＰＵ３１の内部システムクロックが１６ＭＨｚである場合、本実施形態では処理時間が０．１２８５ｍｓに短縮されることになる。

【０２２４】

以下、同様にチェックサム確認処理についても説明する。

図１９は、本実施形態のチェックサム確認処理のプログラムを示す図である。本実施形態のチェックサム確認処理のプログラム（コード）は、以下の通りである。

【０２２５】

「Ｅ＿ＣＬＳＵＭ＿０１：」は、チェックサム確認処理の開始を示すラベルである。

（中略）

【０２２６】

〔使用領域チェックサム算定処理〕

（１）「ＸＯＲ　Ａ，Ａ」：比較例と同じ。

（３）「ＬＤ　ＨＬ，@ＲＭ＿ＲＡＭＡＤＲ」：比較例と同じ。

（３）「ＬＤ　ＢＣ，@ＣＨＫ＿ＡＲＥ」：比較例と同じ。

（３）「ＡＤＤＩＲ」は、ブロックサーチ命令である（電源投入時のチェックサム確認であるため、ブロックサーチと表現する）。具体的には、開始アドレス（@ＲＭ＿ＲＡＭＡＤＲ）からチェックサム算定領域数（@ＣＨＫ＿ＡＲＥ）で指定したバイト数のＲＷＭの和を計算する。そして、加算１バイトあたりのステート数は３である。

【０２２７】

〔領域外チェックサム算定処理〕
（２）「ＬＤＱ　ＨＬ，ＬＯＷ　＠ＲＭ＿ＥＸＴＡＤＲ」：比較例と同じ。
（２）「ＬＤ　Ｂ，＠ＥＸＤ＿ＵＳＥ＿ＥＲＷ」：比較例と同じ。
（３）「ＡＤＤＩＲＳ」は、ブロックサーチ命令である。具体的には、開始アドレス（＠ＲＭ＿ＥＸＴＡＤＲ）からチェックサム算定領域数（＠ＥＸＤ＿ＵＳＥ＿ＥＲＷ）で指定したバイト数のＲＷＭの和を計算する。そして、加算１バイトあたりのステート数は３である。
【０２２８】
〔チェックサム結果フラグ設定処理〕
（３）「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＢＲＳ），Ａ」は、チェックサム結果フラグに値をセーブする命令である。ステート数は３である。
【０２２９】
〔本実施形態のチェックサム確認処理必要ステート数〕
　以上より、本実施形態のチェックサム確認処理でチェックサムの算定に必要な総ステート数は、１＋３＋３＋５１０＊３＋２＋２＋１７０＊３＋３＝２０５４となる。ＮＥＧ命令を使用して符号の反転を行う必要がないため、電源断時よりステート数が２少なく、処理時間が０．１２８３７５ｍｓに短縮されることになる。
【０２３０】
〔第３の技術的特徴の検証結果〕
　図２０は、本実施形態による処理時間の短縮化の検証結果を示す図である。図２０の検証結果から、以下の優位性が明らかである。
（１）チェックサム算定処理の処理時間を比較例から約５９．７８％改善することができる。
（２）チェックサム確認処理の処理時間を比較例から約５９．８２％改善することができる。
（３）総合すると、チェックサムの算定に要する時間を概ね１／２．５に短縮することができる。
（４）これにより、電源投入時はより早期にメインループ処理に移行することができる。
（５）特に電源断時は、電源供給が完全に途絶えてしまう前に確実にチェックサム算定を完了させることができ、退避処理の確実性を向上させることができる。
（６）管理遊技機１００がスロット機である場合も同様の有用性を得ることができる。
【０２３１】
〔第１の技術的特徴：エラー発生時・解除時のコマンドを領域外で生成する構成〕
　次に、本実施形態の第１の技術的特徴について説明する。
【０２３２】
〔第２の技術的特徴の課題〕
　使用領域のプログラム容量（コード量）の削減を実現する。本課題は、管理遊技機１００以外のパチンコ機であっても、記憶手段に使用領域及び領域外を有する場合には該当する。
【０２３３】
〔第１の技術的特徴による解決手段〕
　本実施形態では、領域外に格納したプログラムを実行してエラーの発生、解除を判定する。領域外でエラーの発生又は解除と判定した場合、サブコマンドを生成して領域外のＲＷＭに保存する。使用領域に格納したプログラムは、領域外のＲＷＭにサブコマンドが保存された場合、それを読み込んで使用領域にサブコマンドバッファを設定する構成とした。
【０２３４】
　なお、領域外でエラー発生又は解除と判定しても、そのままサブコマンドバッファを設定しないのは次の理由による。すなわち、サブコマンドバッファは使用領域のＲＷＭ内にあるため、領域外のプログラムから書き込みを行うことはできない。また、ハンドル検出については、不正対策に該当するものではなく、遊技の実行に関わる内容であるため、使

用領域で検出を行う必要がある。
【０２３５】
　プログラム構成の概要は、使用領域側に状態管理処理を構成し、領域外にエラー確認処理のモジュールを配置して、呼び出し実行する。領域外のエラー確認処理では、各種エラーの発生又は解除を判定し、判定の結果を領域外のＲＷＭにサブコマンド（使用領域からの参照用）を保存しておき、呼び出し元に復帰する。使用領域側では、領域外でサブコマンドが保存されたことを確認した場合、改めてサブコマンドを使用領域のＲＷＭに設定することとする。
【０２３６】
〔使用領域の状態管理処理〕
　図２１は、使用領域の状態管理処理のプログラムを示す図である。状態管理処理のプログラム（コード）は、以下の通りである。なお、図２１のプログラムコード中、各行に付記している括弧書きの数値は、使用バイト数を示している。
【０２３７】
　「ＳＴＣ＿ＰＲＣ：」は、状態管理処理の開始を示すラベルである。
【０２３８】
〔エラー確認処理〕
（２）「ＣＡＬＬＥＸ　Ｅ＿ＣＫＥＲＲ」は、領域外のエラー確認処理を呼び出す処理である。ここでのコード量は２バイトである。
　なお、領域外のエラー確認処理では、エラーに係るサブコマンド等を領域外のＲＷＭに設定する。この処理の終了後、以下のプログラムで入賞異常系のエラーフラグ（Ｒ＿ＥＲＲ＿ＦＬＧ）をクリアする。このように、入賞異常系のエラーは、設定した直後の状態管理処理でフラグをクリアされるが、クリアされる前にサブコマンドの設定、遊技停止要求フラグの設定、不正検知１情報ビットタイマの設定を行うこととする。
【０２３９】
〔エラーフラグクリア処理〕
（２）「ＣＬＲＱ　（ＬＯＷ　Ｒ＿ＥＲＲ＿ＦＬＧ）」は、入賞異常系のエラーフラグをクリアする命令である。ここでのコード量は２バイトである。
【０２４０】
〔サブコマンド要求カウンタ確認処理〕
　以下のプログラムでは、エラー確認処理で領域外のＲＷＭに設定したサブコマンドを読み出し、サブコマンドセット処理（ＳＢＣＭＤＳＴ）を呼び出す。
（３）「ＬＤ　ＨＬ，Ｒ＿ＥＲＷ＿ＳＢＣ」は、ＨＬレジスタにサブコマンド要求カウンタのアドレスをセットする命令である。ここでのコード量は３バイトである。
（１）「ＬＤ　Ｂ，（ＨＬ）」は、ＨＬレジスタで指定される対象アドレスの内容をロードする命令である。具体的には、領域外で設定したサブコマンドの数をロードする。ここでのコード量は１バイトである。
（２）「ＪＲ　ＴＺ，ＳＴＣ＿ＰＲＣ＿２０」は、第２ゼロフラグが１であればラベル「ＳＴＣ＿ＰＲＣ＿２０」に分岐する命令である。具体的には、ロードしたサブコマンドの数が０の場合に分岐する。ここでのコード量は２バイトである。
【０２４１】
　「ＳＴＣ＿ＰＲＣ＿１０：」は、サブコマンドセット処理の開始を示すラベルである。
〔サブコマンドセット処理〕
（２）「ＩＮＬＤ　ＤＥ，（ＨＬ）」は、サブコマンドをロードする命令である。具体的には、サブコマンド要求バッファ（Ｒ＿ＥＲＷ＿ＳＣＤ）から、先行コマンドと後続コマンドをロードする。サブコマンド要求バッファは、サブコマンド要求カウンタの直後に配置されており、ＩＮＬＤ命令でＨＬレジスタを更新しながらＤＥレジスタにサブコマンド要求バッファの値をロードする。ここでのコード量は２バイトである。
（１）「ＲＳＴ　ＳＢＣＭＤＳＴ」は、サブコマンドセット処理を呼び出す処理である。ここでは、サブコマンドバッファにサブコマンドを設定する。なお、領域外で設定したサ

ブコマンドは、すべて加算不許可（Ｅレジスタのｂｉｔ０を「１」）としている。これは、使用領域でＡレジスタのクリア（「ＸＯＲ　Ａ，Ａ」等）を省略するためである。ここでのコード量は１バイトである。

【０２４２】

〔ループ回数判定処理〕

（２）「ＤＪＮＺ　ＳＴＣ＿ＰＲＣ＿１０」は、サブコマンドを全てセットしておらず（Ｂレジスタを減算して０でなく）、ループ継続であれば先頭「ＳＴＣ＿ＰＲＣ＿１０」に分岐する命令である。すなわち、領域外で設定したサブコマンドをすべて読み込んでいない場合、ループ先頭へ戻る。ここでのコード量は２バイトである。

【０２４３】

「ＳＴＣ＿ＰＲＣ＿２０：」は、遊技盤エラー状態フラグ更新処理の開始を示すラベルである。

〔遊技盤エラー状態フラグ更新処理〕

領域外のエラー確認処理において、遊技を停止しなければならない致命的な異常があると判定（エラーを判定）した場合、領域外の遊技停止要求フラグに遊技停止を表す値（＠ＹＵＧ＿ＨＮＦ，０６Ｈ）が設定されることになる。

以下の使用領域のプログラムでは、この値を使用領域の遊技盤エラー状態フラグ（Ｒ＿ＹＵＧ＿ＥＲＲ）と論理和する。なお、遊技盤エラー状態フラグの取り得る値は「００Ｈ」か「０６Ｈ」のいずれかである。

（３）「ＬＤ　Ａ，（Ｒ＿ＥＲＷ＿ＹＥＲ）」は、遊技停止要求フラグをロードする命令である。ここでのコード量は３バイトである。

（３）「ＯＲＱ　（ＬＯＷ　Ｒ＿ＹＵＧ＿ＥＲＲ），Ａ」は、遊技停止要求フラグを遊技盤エラー状態フラグと論理和する命令である。ここでのコード量は３バイトである。

【０２４４】

〔ハンドル検出信号確認処理〕

（２）「ＬＤＱ　Ａ，（ＬＯＷ　Ｒ＿ＩＮ１＿ＰＲＴ）」は、入力ポート１状態フラグをロードする命令である。ここでのコード量は２バイトである。

（２）「ＡＮＤ　Ａ，＠ＩＮ１＿ＨＤＬ＿ＢＩＴ」は、入力ポート１状態フラグとハンドル検出信号を論理積する命令である。ここでのコード量は２バイトである。

（３）「ＲＣＰＱ　Ｚ，Ａ，（ＬＯＷ　Ｒ＿ＨＤＬ＿ＳＴＳ）」は、ハンドルステータスと比較し、ゼロフラグが１であれば呼び出し元に復帰する命令である。ここでのコード量は３バイトである。

【０２４５】

〔ハンドル検出指定コマンドセット処理〕

ハンドル検出は不正防止対策ではないため、使用領域で確認する必要がある。

以下のプログラムでは、ハンドル検出信号が変化しない場合はそのまま呼び出し元に戻り、ハンドル検出信号が変化した場合はサブコマンドを送信する。なお、チャタリングなどにより検出・非検出が短時間で変化することが考えられるが、その場合でもすべてサブコマンドを送信し、対処を行うか否かは演出制御基板４０（演出制御ＣＰＵ４１）側のプログラムで判断することとしている。

（２）「ＬＤＱ　（ＬＯＷ　Ｒ＿ＨＤＬ＿ＳＴＳ），Ａ」は、ハンドルステータスにセーブ（ステータスを更新）する命令である。ここでのコード量は２バイトである。

（３）「ＬＤ　ＤＥ，＠ＣＭＤ＿ＳＴＳ＿ＨＤＬ」は、サブコマンドのハンドル検出指定（＠ＣＭＤ＿ＳＴＳ＿ＨＤＬ）をセットする命令である。ここでのコード量は３バイトである。

（１）「ＲＳＴ　ＳＢＣＭＤＳＴ」は、サブコマンドセット処理を呼び出す処理である。ここでのコード量は１バイトである。

（１）「ＲＥＴ」は、呼び出し元に復帰する命令である。コード量は１バイトである。

【０２４６】

図２２は、状態管理処理の手順例を示すフローチャートである。このフローチャートは

、図２１のプログラムに対応している。本処理は、主制御基板３０（主制御ＣＰＵ３１）が実行する。

【０２４７】

ステップＳ５００：主制御基板３０は、領域外のエラー確認処理を呼び出す処理を実行する。これにより、主制御基板３０は領域外でエラー確認処理を実行し、その実行後に本処理に復帰することになる。

【０２４８】

ステップＳ５０１：主制御基板３０は、エラーフラグをクリアする。

ステップＳ５０２：主制御基板３０は、サブコマンド要求カウンタをロードする。

ステップＳ５０３：主制御基板３０は、ロードしたサブコマンド要求カウンタの値が０であるか否かを確認する。

その結果、ロードした値が０であることを確認した場合（Ｙｅｓ）、主制御基板３０はステップＳ５０７を実行する。一方、ロードした値が０あることを確認できない場合（Ｎｏ）、主制御基板３０はステップＳ５０４を実行する。

【０２４９】

ステップＳ５０４：主制御基板３０は、サブコマンドをロードする。サブコマンドは、領域外のＲＷＭに保存されているものをロードする。

ステップＳ５０５：主制御基板３０は、サブコマンドセット処理を呼び出す処理を実行する。この処理において、主制御基板３０は使用領域のサブコマンドバッファを設定する。

ステップＳ５０６：主制御基板３０は、全てのサブコマンドをセット（ロードした領域外の全サブコマンドに対応するサブコマンドバッファを設定）したか否かを確認する。全サブコマンドをセットしたことを確認できた場合（Ｙｅｓ）、主制御基板３０はステップＳ５０７を実行する。一方、未だ全サブコマンドをセットしたことを確認できない場合（Ｎｏ）、主制御基板３０はステップＳ５０４を実行してループを継続する。

【０２５０】

ステップＳ５０７：主制御基板３０は、遊技停止要求フラグをロードする。

ステップＳ５０８：主制御基板３０は、遊技盤エラー状態フラグに遊技停止要求フラグを論理和する。

ステップＳ５０９：主制御基板３０は、入力ポート１状態フラグをロードする。

ステップＳ５１０：主制御基板３０は、入力ポート１状態フラグとハンドル検出信号を論理積する。

ステップＳ５１１：主制御基板３０は、ステップＳ５１０の値がハンドルステータスと一致するか否かを確認する。その結果、ハンドルステータスと一致する（ハンドル検出信号が変化しない）ことを確認した場合（Ｙｅｓ）、主制御基板３０は呼び出し元に復帰する。一方、ハンドルステータスと一致することを確認できない（ハンドル検出信号が変化した）場合（Ｎｏ）、主制御基板３０はステップＳ５１２を実行する。

ステップＳ５１２：主制御基板３０は、ハンドルステータスを更新する。

ステップＳ５１３：主制御基板３０は、サブコマンドの「ハンドル検出指定」をセットする。

ステップＳ５１４：主制御基板３０は、サブコマンドセット処理を呼び出す処理を実行する。この処理では、主制御基板３０は、サブコマンドバッファに「ハンドル検出指定」のサブコマンドを設定する。

以上の手順を終えると、主制御基板３０は、呼び出し元に戻る。

【０２５１】

〔状態管理処理のまとめ〕

以上のように、使用領域のプログラムコード量は（図２１）３５バイト（フローチャートで示す手順数１５ステップ）であり、プログラム容量の増大を抑えることができる。なお、ハンドル検出信号に関しては使用領域で処理を行っているが、それでも簡略化のために検出用タイマを持たず、信号のオンエッジ／オフエッジだけでサブコマンドを送信するロジックとし、コード量の増大を抑えることができている。

【０２５２】
〔領域外のエラー確認処理〕
図２３から図２７は、領域外に格納されるエラー確認処理のプログラムを示す図である。上記のように、使用領域ではエラー発生又は解除の判定を行わないこととしている分、領域外には処理に必要な相当量のプログラムを格納している。エラー確認処理のプログラム（コード）は、以下の通りである。なお、特に省略しない限り、プログラムは図２３～図２７で一連のものである。
【０２５３】
「Ｅ＿ＣＫＥＲＲ：」は、エラー確認処理の開始を示すラベルである。
「ＪＰ　Ｅ＿ＣＫＥＲＲ＿０１」は、ラベル「Ｅ＿ＣＫＥＲＲ＿０１」のアドレスへジャンプする命令である。
（中略）
【０２５４】
ここで、本実施形態では使用領域から領域外のモジュール（Ｅ＿ＣＫＥＲＲ：）を呼び出す際、ＣＡＬＬＥＸ命令を使用しているが、呼び出し先のアドレスが２０００Ｈ～２０ＦＦＨである場合は２バイト、それ以外である場合は４バイトのコード量が必要となる。このため、呼び出し先モジュールのラベルと、モジュール本体へのジャンプ命令を２０００Ｈ以降に並べることとし、各モジュールの本体部分のプログラムをその後に記述する構成とする。これにより、領域外のモジュールとして最大８６個までを２バイトのＣＡＬＬＥＸ命令によって呼び出すことができる。
上記の「ＪＰ　Ｅ＿ＣＫＥＲＲ＿０１」は、そのための２バイトのジャンプ命令である。ジャンプ先のアドレスは２０００Ｈ～２０ＦＦＨでなくてもよいので、ラベル「Ｅ＿ＣＫＥＲＲ＿０１」のアドレスはそれ以降のアドレスに記述されている。
【０２５５】
〔初期化処理〕
初期化処理は、「ＪＰ　Ｅ＿ＣＫＥＲＲ＿０１」命令で呼び出される。
「Ｅ＿ＣＫＥＲＲ＿０１：」は、初期化処理の開始を示すラベルである。
「ＬＤ　Ｑ，ＨＩＧＨ　＠ＲＭ＿ＥＸＴＡＤＲ」は、領域外作業領域先頭アドレスの上位バイトをセットする命令である。
ここで、本実施形態では、ＣＡＬＬＥＸ命令によりレジスタセットのバンクが「バンク０」から「バンク１」に変更され、また、ＲＥＴＥＸ命令によりバンクが「バンク１」から「バンク０」に変更される。バンク０のＱレジスタとバンク１のＱレジスタは、機能としては同等のものであるが、値は互いに独立している。例えば、バンク０のＱレジスタにＦ０Ｈという値を設定したとしても、バンク１のＱレジスタの値は変化しない。
上記のように記憶手段のＲＷＭには、Ｆ０００Ｈ～Ｆ３ＦＦＨまでの４キロバイトが実装されているが、使用領域のＲＷＭとしてＦ０００Ｈ～Ｆ１ＦＦＨ、領域外のＲＷＭとしてＦ３００Ｈ～Ｆ３ＦＦＨを使用している（図２のメモリマップ参照）。また、Ｆ２００Ｈ～Ｆ２ＦＦＨは未使用である。
このとき、領域外用（「バンク１」とする）のＱレジスタにはＦ３Ｈを設定する。これにより、Ｆ３００Ｈ～Ｆ３ＦＦＨのＲＷＭに対してＱレジスタを用いた命令を使用することができることになる。領域外のプログラムコードは、規則上の制限を受けないが、より長さを抑えたコードを使用した方が実行時間も長期化を抑えることができるため、領域外についてもコードを短縮化することが望ましい。
【０２５６】
〔サブコマンド要求カウンタクリア処理〕
「ＣＬＲＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＳＢＣ）」は、サブコマンド要求カウンタをクリアする命令である。本命令により、領域外でセットするサブコマンドの数が初期化される。
【０２５７】
〔ループ設定処理１〕
ループ設定処理１では、エラー確認用のループで使用するテーブルのアドレス及びルー

プ回数を設定する。ハンドル検出は使用領域で行うため、ここで判定するエラーは、「スイッチ異常」、「扉開放」、「盤面磁気エラー１」、「盤面磁気エラー２」及び「電波エラー」の５種類である。なお、エラーの種類は、機種（盤面の種類）によって異なる場合がある。

「ＬＤ　ＨＬ，Ｄ＿ＥＴＢ＿ＥＲ１」は、エラー管理処理テーブル１のアドレスをセットする命令である。なお、エラー管理処理テーブル１については後述する。

「ＬＤ　Ｂ，＠Ｄ＿ＥＴＢ＿ＥＲ１＿ＮＵＭ」は、状態項目数をセットする命令である。

【０２５８】

〔監視対象信号ビットデータ判定処理〕

「Ｅ＿ＣＫＥＲＲ＿１０：」は、監視対象信号ビットデータ判定処理の開始を示すラベルである。

「ＬＤＩＮ　ＤＥ，（ＨＬ）」は、入力ポート状態フラグのアドレスをロードする命令である。

「ＬＤ　Ａ，（ＤＥ）」は、ロードした対象アドレスの内容をＡレジスタにロードする命令である。

「ＬＤ　Ｃ，０」は、Ｃレジスタに０をセットする命令である。

「ＡＮＤ　Ａ，（ＨＬ）」は、監視対象ビットデータと論理積する命令である。つまり、上の命令で先ずＣレジスタに０を設定し、本命令では目的のビット以外をすべて０にする。

「ＪＲ　Ｚ，Ｅ＿ＣＫＥＲＲ＿２０」は、ゼロフラグが１であれば分岐する命令である。

「ＩＮＣ　Ｃ」は、Ｃレジスタをインクリメントする命令である。すなわち、目的のビットが０の場合は、上の命令で「Ｅ＿ＣＫＥＲＲ＿２０」に分岐し、そうでない場合は本命令でＣレジスタに１を設定することになる。これにより、エラー検出時はＣレジスタに１が設定され、エラー未検出時はＣレジスタに０が設定されることになる。

【０２５９】

〔監視対象ステータス今回値処理〕

監視対象ステータス今回値処理は、上記の「ＪＲ　Ｚ，Ｅ＿ＣＫＥＲＲ＿２０」命令でゼロフラグが１の場合に呼び出される。

「Ｅ＿ＣＫＥＲＲ＿２０：」は、監視対象ステータス今回値処理の開始を示すラベルである。監視対象ステータス今回値処理は、以下の監視対象ステータス今回値更新処理及び監視対象ステータス今回値判定処理を含む構成である。

【０２６０】

〔監視対象ステータス今回値更新処理〕

「ＩＮＬＤ　ＤＥ，（ＨＬ）」は、監視対象ステータス前回値のアドレスをロードする命令である。

「ＬＤ　Ａ，（ＤＥ）」は、対象アドレスの内容をロードする命令である。ここでは、上の命令で監視対象ステータス前回値のアドレスをロードし、本命令でＡレジスタに前回値をロードする。

「ＣＰ　Ａ，Ｃ」は、ＡレジスタとＣレジスタを比較する命令である。ここでは、監視対象ステータス前回値と監視対象ステータス今回値を設定し、ゼロフラグを更新する。

「ＬＤ　（ＤＥ），Ｃ」は、対象アドレスの内容にセーブする命令である。ここでは、監視対象ステータス今回値を監視対象ステータス前回値にセーブする。

「ＩＮＣ　ＤＥ」は、ＤＥレジスタをインクリメントする命令である。すなわち、監視対象ステータス前回値を指しているＤＥレジスタに１を加算し、当該エラーの監視タイマのアドレスを指した状態にする。

「ＩＮＣ　ＨＬ」は、ＨＬレジスタをインクリメントする命令である。ここでは、テーブルを指しているＨＬレジスタに１を加算し、当該エラーの監視タイマ値のアドレスを指した状態にする。

【０２６１】

〔監視対象ステータス今回値判定処理〕

「ＪＲ　Ｚ，Ｅ＿ＣＫＥＲＲ＿３０」は、ゼロフラグが１であれば分岐する命令である。ここでは、先の「ＣＰ　Ａ，Ｃ」命令で設定したゼロフラグを確認し、ゼロフラグがセット（Ｚ）である場合、すなわち監視対象ステータス前回値と監視対象ステータス今回値が一致するものであれば、「Ｅ＿ＣＫＥＲＲ＿３０」のアドレスに分岐する。一方、ゼロフラグがリセット（ＮＺ）である場合は以下となる。

「ＬＤ　Ａ，（ＨＬ）」は、対象アドレスの内容をロードする命令である。ここでは、当該エラーの監視タイマ値をロードする。

「ＬＤ　（ＤＥ），Ａ」は、対象アドレスの内容にセーブする命令である。ここでは、監視タイマをセーブする。

【０２６２】

〔監視タイマ値判定処理〕

監視タイマ値判定処理は、監視対象ステータス前回値と監視対象ステータス今回値が一致する場合に呼び出される。

「Ｅ＿ＣＫＥＲＲ＿３０：」は、監視タイマ値判定処理の開始を示すラベルである。

本実施形態では、エラー状態が一定時間以上継続した時点で正式にエラー発生と判定する。エラー解除も同様に、エラー解除状態が一定時間以上継続した時点で正式にエラー解除と判定する。このため、エラーの発生及び解除の判定には、タイマの設定及び減算を行う処理が必要となる。本処理では、タイマの設定及び減算を行うこととする。

【０２６３】

〔監視タイマ値判定処理１〕

「ＩＮＣ　ＨＬ」は、ＨＬレジスタをインクリメントする命令である。

「ＬＤ　Ａ，（ＤＥ）」は、対象アドレスの内容をロードする命令である。

「ＪＴ　Ｚ，Ａ，Ｅ＿ＣＫＥＲＲ＿４０」は、ゼロフラグが１であれば分岐する命令である。ここでは、監視タイマがゼロである場合、「Ｅ＿ＣＫＥＲＲ＿４０」へ分岐する。なお、監視タイマの減算は以下の処理で実行している。

「ＤＥＣ　Ａ」は、タイマ値を１減算する命令である。すなわち、上の命令で監視タイマがゼロでない場合、監視タイマから１を減算する。

「ＬＤ　（ＤＥ），Ａ」は、対象アドレスの内容にセーブする命令である。ここでは、監視タイマ値が更新される。

【０２６４】

〔監視タイマ値判定処理２〕

「ＪＴ　ＮＺ，Ａ，Ｅ＿ＣＫＥＲＲ＿４０」は、ゼロフラグが０であれば分岐する命令である。上記の「ＤＥＣ　Ａ」命令でタイマ値を１減算するが、その結果、監視タイマが０でない場合もやはり、「Ｅ＿ＣＫＥＲＲ＿４０」へ分岐することになる。なお、「Ｅ＿ＣＫＥＲＲ＿４０」の処理は後述する。

一方、監視タイマが１から０になった場合、以下のエラーフラグ判定処理１へ進む。

【０２６５】

〔エラーフラグ判定処理１〕

「ＩＮＣ　ＤＥ」は、ＤＥレジスタをインクリメントする命令である。

「ＬＤ　Ａ，（ＤＥ）」は、対象アドレスの内容をロードする命令である。

「ＣＰ　Ａ，Ｃ」は、ＡレジスタとＣレジスタを比較する命令である。

「ＪＲ　Ｚ，Ｅ＿ＣＫＥＲＲ＿４０」は、ゼロフラグが１であれば分岐する命令である。

「ＬＤ　Ａ，Ｃ」は、ステータス今回値をセットする命令である。

「ＬＤ　（ＤＥ），Ａ」は、対象アドレスの内容にセーブする命令である。

「ＥＸ　ＤＥ，ＨＬ」は、ＤＥレジスタとＨＬレジスタを交換する命令である。

このように、監視タイマが１から０になった場合、エラーフラグ判定処理１で改めてエラーフラグ（現在のエラー状態）と監視対象ステータス今回値を比較する。その結果、エラーフラグと監視対象ステータス今回値が一致した場合は、「Ｅ＿ＣＫＥＲＲ＿４０」に分岐する。

一方、エラーフラグと監視対象ステータス今回値が不一致である場合は、エラーフラグ

に監視対象ステータス今回値をセーブし、以下のサブコマンド設定処理１へ進む。
【０２６６】
〔サブコマンド設定処理１〕
サブコマンド設定処理１は、エラー発生又は解除と判定した場合、サブコマンドをセットする処理である。ただし、領域外の処理から使用領域のサブコマンドセット処理を呼び出したり、使用領域のサブコマンドバッファを書き換えたりすることは不可であるため、仮のサブコマンドを領域外のＲＷＭ（バッファ）に設定する。これにより、設定したサブコマンドを使用領域から参照可能とすることができる。
「ＬＤＱ　ＨＬ，ＬＯＷ　Ｒ＿ＥＲＷ＿ＳＢＣ」は、サブコマンド要求カウンタのアドレスをＨＬレジスタにセットする命令である。
「ＩＮＣ　（ＨＬ）」は、ＨＬレジスタのサブコマンド要求カウンタを１加算する命令である。
「ＬＤ　Ａ，（ＨＬ）」は、対象アドレスの内容をロードする命令である。ここでは、加算後のＨＬレジスタのサブコマンド要求カウンタ値をＡレジスタにロードする。
「ＡＤＤＷＢ　ＨＬ，Ａ」は、ＨＬレジスタにＡレジスタを加算する命令である。
「ＡＤＤＷＢ　ＨＬ，Ａ」は、ＨＬレジスタにＡレジスタを加算する命令である。これにより、ＨＬレジスタにカウンタ値を２回加算したことになる。
「ＬＤ　（ＨＬ），＠ＣＭＤ＿ＳＴＳ＿ＨＥＤ」は、対象アドレスの内容に状態指定先行コマンドをセーブする命令である。ここでは、先行コマンドとして、すべてのエラー発生・解除に共通のコマンド（８５Ｈ）をセーブする。
「ＤＥＣ　ＨＬ」は、ＨＬレジスタをデクリメントする命令である。ここでは、ＨＬレジスタを１減算することにより、後続コマンドのアドレスを指した状態にする。
「ＬＤ　Ａ，Ｃ」は、監視対象ステータス今回値をセットする命令である。
〔ステータス今回値判定処理１〕
「ＡＤＤ　Ａ，Ａ」は、Ａレジスタに値を加算する命令である。
「ＡＤＤ　Ａ，（ＤＥ）」は、監視対象ステータス今回値にサブコマンドを加算する命令である。すなわち、上記の「ＬＤ　Ａ，Ｃ」命令でセットした監視対象ステータス今回値の２倍を算定して後続コマンド値を加算する。
「ＬＤ　（ＨＬ），Ａ」は、対象アドレスの内容にセーブする命令である。ここで後続コマンドをＨＬレジスタにセーブする。
なお、監視対象ステータス今回値を２倍する理由は、サブコマンドセット処理（ＳＢＣＭＤＳＴ）で要求しているコマンドの形式に合わせるためである。例えば、後続コマンドは００Ｈ～７ＦＨのいずれかとなるが、サブコマンドセット処理を呼び出す際にＥレジスタに設定する値は、後続コマンドの２倍に加算許可フラグ（０：加算許可、１：加算不許可）を加えたものであるため、監視対象ステータス今回値の２倍を加算する必要がある。
【０２６７】
〔ＲＷＭ書き換え処理〕
以下のＲＷＭ書き換え処理は、エラー発生に伴うＲＷＭの書き換えを行う処理である。
「ＥＸ　ＤＥ，ＨＬ」は、ＤＥレジスタとＨＬレジスタを交換する命令である。
「ＪＴ　Ｚ，Ｃ，Ｅ＿ＣＫＥＲＲ＿４０」は、ゼロフラグが１であれば分岐する命令である。ここでは、エラー発生時であるかエラー解除時であるかを判定し、エラー解除時であれば「Ｅ＿ＣＫＥＲＲ＿４０」へ分岐する。
「ＬＤ　Ｅ，（ＨＬ＋１）」は、エラー発生時設定対象ラムアドレス下位バイトをロードする命令である。ここでは、エラー発生時に書き換えを行うＲＷＭの下位バイトをロードする。
「ＪＲ　ＴＺ，Ｅ＿ＣＫＥＲＲ＿４０」は、第２ゼロフラグが１であれば分岐する命令である。ここでは、ロードした値が０である場合、エラー発生時にＲＷＭの書き換えを行わないものとして「Ｅ＿ＣＫＥＲＲ＿４０」へ分岐する。
一方、ロードした値が０でなく、ＲＷＭの書き換えを行う場合は以下となる。
「ＬＤ　Ａ，（ＨＬ＋２）」は、エラー発生時設定データをロードする命令である。こ

こでは、書き換えを行うためのデータをロードしている。
「ＬＤ　（ＤＥ），Ａ」は、ロードしたデータを対象アドレスの内容にセーブする命令である。
【０２６８】
〔ループ終了処理判定１処理〕
以下のループ終了処理判定１処理では、次のエラーに係る処理を行うため、テーブルアドレスに３を加算し、ループ終了でなければ「Ｅ＿ＣＫＥＲＲ＿１０」に戻る。
「Ｅ＿ＣＫＥＲＲ＿４０：」は、ループ終了処理判定１処理の開始を示すラベルである。
「ＡＤＤＷＢ　ＨＬ，００３Ｈ」は、ＨＬレジスタに［００３Ｈ］を加算する命令である。
〔ループ終了判定１〕
「ＤＪＮＺ　Ｅ＿ＣＫＥＲＲ＿１０」は、ループ継続であれば分岐する命令である。ここでは、上記のようにループ終了でなければ「Ｅ＿ＣＫＥＲＲ＿１０」に戻ることになる。
【０２６９】
〔ループ設定処理２～ループ終了判定２処理までのプログラム〕
以下のループ設定処理２からループ終了判定２処理では、入賞異常系のエラーの確認、及びサブコマンドのセットを行う。ここでは、スイッチ異常等の確認を行うループ処理と比較して、「エラー発生のみでエラー解除の判定を行わない」、「タイマを持たず、エラーが発生した場合はただちにサブコマンド等をセットする」という違いがある。
【０２７０】
〔ループ設定処理２〕
「ＬＤ　ＨＬ，Ｄ＿ＥＴＢ＿ＥＲ２」は、エラー管理処理テーブル２のアドレスをセットする命令である。なお、エラー管理処理テーブル２の内容は後述する。
「ＬＤ　Ｂ，＠Ｄ＿ＥＴＢ＿ＥＲ２＿ＮＵＭ」は、状態項目数をセットする命令である。
【０２７１】
〔エラーフラグ判定処理２〕
「Ｅ＿ＣＫＥＲＲ＿５０：」は、エラーフラグ判定処理２の開始を示すラベルである。
「ＬＤ　Ａ，（Ｒ＿ＥＲＲ＿ＦＬＧ）」は、エラーフラグをロードする命令である。
「ＡＮＤ　Ａ，（ＨＬ）」は、エラーフラグと監視対象ビットデータを論理積する命令である。
「ＪＲ　Ｚ，Ｅ＿ＣＫＥＲＲ＿６０」は、ゼロフラグが１であれば「Ｅ＿ＣＫＥＲＲ＿６０」に分岐する命令である。
「ＬＤ　Ａ，０１Ｈ」は：Ａレジスタに［０１Ｈ］をセットする命令である。
【０２７２】
「Ｅ＿ＣＫＥＲＲ＿６０：」は、「ＪＲ　Ｚ，Ｅ＿ＣＫＥＲＲ＿６０」命令の分岐先アドレスを示すラベルである。
「ＩＮＣ　ＨＬ」は、ＨＬレジスタをインクリメントする命令である。
「ＬＤＩＮ　ＤＥ，（ＨＬ）」は、エラー発生フラグのアドレスをロードする命令である。
【０２７３】
〔エラー発生フラグ更新処理〕
「ＬＤ　（ＤＥ），Ａ」は、対象アドレスの内容にエラー発生フラグをセーブする命令である。
〔ステータス今回値判定処理〕
「ＪＴ　Ｚ，Ａ，Ｅ＿ＣＫＥＲＲ＿７０」は、ゼロフラグが１であれば「Ｅ＿ＣＫＥＲＲ＿７０」に分岐する命令である。
「ＥＸ　ＤＥ，ＨＬ」は、ＤＥレジスタとＨＬレジスタを交換する命令である。
【０２７４】
〔サブコマンド設定処理２〕
「ＬＤＱ　ＨＬ，ＬＯＷ　Ｒ＿ＥＲＷ＿ＳＢＣ」は、ＨＬレジスタにサブコマンド要求

カウンタのアドレスをセットする命令である。
「ＩＮＣ （ＨＬ）」は、サブコマンド要求カウンタを１加算する命令である。
「ＬＤ Ａ，（ＨＬ）」は、対象アドレスの内容をロードする命令である。ここでは、サブコマンド要求カウンタの値をＡレジスタにロードする。
「ＡＤＤＷＢ ＨＬ，Ａ」は、ＨＬレジスタにＡレジスタを加算する命令である。
「ＡＤＤＷＢ ＨＬ，Ａ」は、ＨＬレジスタにＡレジスタを加算する命令である（２回目）。
「ＬＤ （ＨＬ），＠ＣＭＤ＿ＳＴＳ＿ＨＥＤ」は、対象アドレスの内容に状態指定先行コマンド（＠ＣＭＤ＿ＳＴＳ＿ＨＥＤ）をセーブする命令である。
「ＤＥＣ ＨＬ」は、ＨＬレジスタをデクリメントする命令である。
「ＬＤ Ａ，（ＤＥ）」は、対象アドレスの内容をロードする命令である。ここでは、ＤＥレジスタの内容をＡレジスタにロードする。
「ＬＤ （ＨＬ），Ａ」は、対象アドレスの内容にセーブする命令である。ここでは、ＨＬレジスタにＡレジスタの内容をセーブする。
「ＥＸ ＤＥ，ＨＬ」は、ＤＥレジスタとＨＬレジスタを交換する命令である。
【０２７５】
〔エラー発生時設定対象ラムアドレス確認処理〕
「ＬＤ Ｅ，（ＨＬ＋１）」は、エラー発生時設定対象ラムアドレス下位バイトをロードする命令である。
「ＪＲ ＴＺ，Ｅ＿ＣＫＥＲＲ＿７０」は、第２ゼロフラグが１であれば「Ｅ＿ＣＫＥＲＲ＿７０」に分岐する命令である。
「ＬＤ Ａ，（ＨＬ＋２）」は、エラー発生時設定データをロードする命令である。
「ＬＤ （ＤＥ），Ａ」は、対象アドレスの内容にセーブする命令である。ここでは、ＤＥレジスタの内容にＡレジスタの内容をセーブする。
【０２７６】
「Ｅ＿ＣＫＥＲＲ＿７０：」は、「ＪＲ Ｚ，Ｅ＿ＣＫＥＲＲ＿７０」命令の分岐先アドレスを示すラベルである。
「ＡＤＤＷＢ ＨＬ，００３Ｈ」は、ＨＬレジスタに［００３Ｈ］を加算する命令である。
【０２７７】
〔ループ終了判定２〕
「ＤＪＮＺ Ｅ＿ＣＫＥＲＲ＿５０」は、ループ継続であれば「Ｅ＿ＣＫＥＲＲ＿５０」に分岐する命令である。
【０２７８】
〔エラーフラグ判定処理３〕
以下のエラーフラグ判定処理３では、不正検知１情報出力のためのタイマを設定する。具体的には、入賞異常系のエラーが発生したと判定するか、スイッチ異常、盤面磁気エラー１、盤面磁気エラー２、電波エラーのいずれかが発生したと判定した場合、不正検知１情報ビットタイマに１秒をセーブする。
【０２７９】
「ＬＤ Ａ，（Ｒ＿ＥＲＲ＿ＦＬＧ）」は、エラーフラグをＡレジスタにロードする命令である。
「ＯＲＱ Ａ，（ＬＯＷ Ｒ＿ＥＲＷ＿ＳＥＦ）」は、スイッチエラーフラグとＡレジスタを論理和する命令である。
「ＯＲＱ Ａ，（ＬＯＷ Ｒ＿ＥＲＷ＿Ｍ１Ｆ）」は、盤面磁気エラー１フラグとＡレジスタを論理和する命令である。
「ＯＲＱ Ａ，（ＬＯＷ Ｒ＿ＥＲＷ＿Ｍ２Ｆ）」は、盤面磁気エラー２フラグとＡレジスタを論理和する命令である。
「ＯＲＱ Ａ，（ＬＯＷ Ｒ＿ＥＲＷ＿Ｄ１Ｆ）」は、電波エラー１フラグとＡレジスタを論理和する命令である。

「ＪＴ　Ｚ，Ａ，Ｅ＿ＣＫＥＲＲ＿８０」は、ゼロフラグが１であれば「Ｅ＿ＣＫＥＲＲ＿８０」に分岐する命令である。

【０２８０】

〔不正検知１情報ビットタイマセット処理〕

「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿Ｉ１Ｔ），＠ＴＭＲ＿ＨＬＤ＿ＳＥＣ」は、不正検知１情報ビットタイマに不正検知情報１保持時間をセーブする命令である。

【０２８１】

「Ｅ＿ＣＫＥＲＲ＿８０：」は、上記の「ＪＴ　Ｚ，Ａ，Ｅ＿ＣＫＥＲＲ＿８０」命令でゼロフラグが１である場合の分岐先アドレスを示すラベルである。

「ＲＥＴＥＸ」は、呼び出し元に復帰する命令である。

【０２８２】

〔領域外のテーブル構成〕

図２８及び図２９は、エラー管理処理テーブル１の構成例を示す図である。

「Ｄ＿ＥＴＢ＿ＥＲ１：」は、エラー管理処理テーブル１の開始アドレスを示すラベルである。

「ＤＷ　Ｒ＿ＩＮ１＿ＰＲＴ」は、入力ポート１状態フラグのアドレスを示すラベルである。

「ＤＢ　＠ＩＮ１＿ＳＷＥ＿ＢＩＴ」は、スイッチ異常信号ビットデータである。

「ＤＷ　Ｒ＿ＥＲＷ＿ＳＥＯ」は、スイッチステータス前回値のアドレスである。

「ＤＢ　＠ＴＭＲ＿ＣＨＫ＿ＳＷＩ」は、スイッチ異常監視タイマ値である。

「ＤＢ　ＬＯＷ　＠ＣＭＤ＿ＥＲＲ＿ＳＷＩ」は、サブコマンド（スイッチエラー指定）である。

「ＤＢ　０」は、スイッチエラー指定時のエラー発生時設定対象ラムアドレスである。

「ＤＢ　０」は、スイッチエラー指定時のエラー発生時設定データである。したがって、スイッチエラー指定の場合、遊技は停止されない。

【０２８３】

「ＤＷ　Ｒ＿ＩＮ１＿ＰＲＴ」は、入力ポート１状態フラグのアドレスである。

「ＤＢ　＠ＩＮ１＿ＤＲＥ＿ＢＩＴ」は、扉開放スイッチ信号ビットデータである。

「ＤＷ　Ｒ＿ＥＲＷ＿ＤＥＯ」は、扉開放ステータス前回値のアドレスである。

「ＤＢ　＠ＴＭＲ＿ＣＨＫ＿ＤＯＲ」は、扉監視タイマ値である。

「ＤＢ　ＬＯＷ　＠ＣＭＤ＿ＥＲＲ＿ＢＤＲ」は、サブコマンド（扉開放エラー指定）である。

「ＤＢ　０」は、扉開放エラー指定時のエラー発生時設定対象ラムアドレスである。

「ＤＢ　０」は、扉開放エラー指定時のエラー発生時設定データである。したがって、扉開放エラー指定の場合、遊技は停止されない。

【０２８４】

「ＤＷ　Ｒ＿ＩＮ１＿ＰＲＴ」は、入力ポート１状態フラグのアドレスである。

「ＤＢ　＠ＩＮ１＿Ｍ１Ｅ＿ＢＩＴ」は、盤面磁気検出センサ１信号ビットデータである。

「ＤＷ　Ｒ＿ＥＲＷ＿Ｍ１Ｏ」は、盤面磁気検出センサ１ステータス前回値のアドレスである。

「ＤＢ　＠ＴＭＲ＿ＣＨＫ＿ＭＡＧ」は、磁気監視タイマ値である。

「ＤＢ　ＬＯＷ　＠ＣＭＤ＿ＥＲＲ＿ＭＧ２」は、サブコマンド（盤面磁気エラー２指定）である。

「ＤＢ　ＬＯＷ　Ｒ＿ＥＲＷ＿ＹＥＲ」は、盤面磁気エラー２指定時のエラー発生時設定対象ラムアドレスである。

「ＤＢ　＠ＹＵＧ＿ＨＮＦ」は、盤面磁気エラー２指定時のエラー発生時設定データである。盤面磁気エラー２指定の場合、遊技が停止される。

【０２８５】

「ＤＷ　Ｒ＿ＩＮ１＿ＰＲＴ」は、入力ポート１状態フラグのアドレスである。

「ＤＢ　＠ＩＮ１＿Ｍ２Ｅ＿ＢＩＴ」は、盤面磁気検出センサ２信号ビットデータである。
「ＤＷ　Ｒ＿ＥＲＷ＿Ｍ２Ｏ」は、盤面磁気検出センサ２ステータス前回値である。
「ＤＢ　＠ＴＭＲ＿ＣＨＫ＿ＭＡＧ」は、磁気監視タイマ値である。
「ＤＢ　ＬＯＷ　＠ＣＭＤ＿ＥＲＲ＿ＭＧ３」は、サブコマンド（盤面磁気エラー３指定）である。
「ＤＢ　ＬＯＷ　Ｒ＿ＥＲＷ＿ＹＥＲ」は、盤面磁気エラー３指定時のエラー発生時設定対象ラムアドレスである。
「ＤＢ　＠ＹＵＧ＿ＨＮＦ」は、盤面磁気エラー３指定時のエラー発生時設定データである。盤面磁気エラー３指定の場合、遊技が停止される。

【０２８６】
「ＤＷ　Ｒ＿ＩＮ０＿ＰＲＴ」は、入力ポート０状態フラグのアドレスである。
「ＤＢ　＠ＩＮ０＿Ｄ１Ｅ＿ＢＩＴ」は、電波検出センサ１ビットデータである。
「ＤＷ　Ｒ＿ＥＲＷ＿Ｄ１Ｏ」は、電波検出センサ１ステータス前回値である。
「ＤＢ　＠ＴＭＲ＿ＣＨＫ＿ＤＥＮ」は、電波監視タイマ値である。
「ＤＢＬＯＷ　＠ＣＭＤ＿ＥＲＲ＿ＤＥ１」は、サブコマンド（電波エラー１指定）である。
「ＤＢ　０」は、電波エラー１指定時のエラー発生時設定対象ラムアドレスである。
「ＤＢ　０」は、電波エラー１指定時のエラー発生時設定データである。したがって、電波エラー１指定の場合、遊技は停止されない。

【０２８７】
「＠Ｄ＿ＥＴＢ＿ＥＲ１＿ＮＵＭ　ＥＱＵ　（＄　－　Ｄ＿ＥＴＢ＿ＥＲ１）／９」は、状態項目数である。

【０２８８】
〔エラー管理処理テーブル２〕
図３０は、エラー管理処理テーブル２の構成例を示す図である。
「Ｄ＿ＥＴＢ＿ＥＲ２：」は、エラー管理処理テーブル２の開始アドレスを示すラベルである。
「ＤＢ　＠ＥＲＦ＿ＩＬ１＿ＢＩＴ」は、不正入賞１エラービットデータである。
「ＤＷ　Ｒ＿ＥＲＷ＿ＦＳ１」は、不正入賞１エラーフラグである。
「ＤＢ　ＬＯＷ　＠ＣＭＤ＿ＥＲＲ＿ＦＳ１」は、サブコマンド（不正入賞エラー１発生指定）である。
「ＤＢ　ＬＯＷ　Ｒ＿ＥＲＷ＿ＹＥＲ」は、不正入賞エラー１発生指定時のエラー発生時設定対象ラムアドレスである。
「ＤＢ　＠ＹＵＧ＿ＨＮＦ」は、不正入賞エラー１発生指定時のエラー発生時設定データである。不正入賞エラー１発生指定の場合、遊技が停止される。

【０２８９】
「ＤＢ　＠ＥＲＦ＿ＩＬ３＿ＢＩＴ」は、不正入賞３エラービットデータである。
「ＤＷ　Ｒ＿ＥＲＷ＿ＦＳ３」は、不正入賞３エラーフラグである。
「ＤＢ　ＬＯＷ　＠ＣＭＤ＿ＥＲＲ＿ＦＳ３」は、サブコマンド（不正入賞エラー３発生指定）である。
「ＤＢ　０」は、不正入賞エラー３発生指定時のエラー発生時設定対象ラムアドレスである。
「ＤＢ　０」は、不正入賞エラー３発生指定時のエラー発生時設定データである。したがって、不正入賞エラー３発生指定の場合、遊技は停止されない。

【０２９０】
「ＤＢ　＠ＥＲＦ＿ＦＤＮ＿ＢＩＴ」は、普電異常入賞エラービットデータである。
「ＤＷ　Ｒ＿ＥＲＷ＿ＦＤＮ」は、普電異常入賞エラーフラグである。
「ＤＢ　ＬＯＷ　＠ＣＭＤ＿ＥＲＲ＿ＦＤＮ」は、サブコマンド（普電異常入賞エラー発生指定）である。

「ＤＢ　０」は、普電異常入賞エラー発生指定時のエラー発生時設定対象ラムアドレスである。
「ＤＢ　０」は、普電異常入賞エラー発生指定時のエラー発生時設定データである。したがって、普電異常入賞エラー発生指定の場合、遊技は停止されない。
【０２９１】
「＠Ｄ＿ＥＴＢ＿ＥＲ２＿ＮＵＭ　ＥＱＵ　（＄　－　Ｄ＿ＥＴＢ＿ＥＲ２）／６」は、状態項目数である。
【０２９２】
図３１から図３７は、エラー確認処理の手順例を示すフローチャートである。図３１から図３７のフローチャートは、図２３から図２７のプログラムに対応している。エラー確認処理は、主制御基板３０（主制御ＣＰＵ３１）により領域外で実行される。
【０２９３】
ステップＳ４００：主制御基板３０は、Ｑレジスタに領域外作業領域の上位バイトをセットする。
ステップＳ４０１：主制御基板３０は、サブコマンド要求カウンタをクリアする。
ステップＳ４０２：主制御基板３０は、エラー管理処理テーブル１のアドレスをセットする。
ステップＳ４０３：主制御基板３０は、状態項目数をセットする。
ステップＳ４０４：主制御基板３０は、入力ポート状態フラグのアドレスをロードする。
ステップＳ４０５：主制御基板３０は、対象アドレスの内容をロードする。
ステップＳ４０６：主制御基板３０は、Ｃレジスタに０をセットする。
ステップＳ４０８：主制御基板３０は、対象アドレスの内容と監視対象ビットを論理積する。
ステップＳ４０９：主制御基板３０は、ゼロフラグが１か否か（異常未発生か？）を確認する。その結果、ゼロフラグが１であることを確認した場合（Ｙｅｓ）、主制御基板３０はステップＳ４１１を実行する。一方、ゼロフラグが１であることを確認できない場合（Ｎｏ）、主制御基板３０はステップＳ４１０を実行する。
【０２９４】
ステップＳ４１０：主制御基板３０は、Ｃレジスタをインクリメント（Ｃレジスタに１をセット）する。
ステップＳ４１１：主制御基板３０は、ステータス前回値のアドレスをロードする。
ステップＳ４１２：主制御基板３０は、対象アドレスの内容をロードする。
ステップＳ４１３：主制御基板３０は、ロードした内容とＣレジスタを比較してゼロフラグを更新する。
ステップＳ４１４：主制御基板３０は、対象アドレスにＣレジスタ（ステータス今回値）をセーブする。
ステップＳ４１５：主制御基板３０は、ＤＥレジスタをインクリメント（タイマのアドレスをセット）する。
ステップＳ４１６：主制御基板３０は、ＨＬレジスタをインクリメント（タイマ値のアドレスを指す）する。
【０２９５】
〔図３２：外部結合子（２）以降〕
ステップＳ４１７：主制御基板３０は、ゼロフラグが１か否かを確認する。その結果、ゼロフラグが１であることを確認した場合（Ｙｅｓ）、主制御基板３０はステップＳ４２０を実行する。一方、ゼロフラグが１であることを確認できない場合（Ｎｏ）、主制御基板３０はステップＳ４１８を実行する。
ステップＳ４１８：主制御基板３０は、タイマ値をロードする。
ステップＳ４１９：主制御基板３０は、タイマにタイマ値をセーブする。
ステップＳ４２０：主制御基板３０は、ＨＬレジスタをインクリメントする。
ステップＳ４２１：主制御基板３０は、タイマをロードする。

ステップＳ４２２：主制御基板３０は、タイマの値が０か否かを確認する。その結果、タイマの値が０であることを確認した場合（Ｙｅｓ）、主制御基板３０は、外部結合子（４）に移行してステップＳ４４６（図３４）を実行する。一方、ゼロフラグが１であることを確認できない場合（Ｎｏ）、主制御基板３０はステップＳ４２３を実行する。

【０２９６】

ステップＳ４２３：主制御基板３０は、タイマの値を１減算する。

ステップＳ４２４：主制御基板３０は、減算後のタイマの値が０でないか否かを確認する。その結果、タイマの値が０でないことを確認した場合（Ｙｅｓ）、主制御基板３０は、外部結合子（４）に移行してステップＳ４４６（図３４）を実行する。一方、タイマの値が０でないことを確認できない場合（Ｎｏ）、主制御基板３０はステップＳ４２５を実行する。

【０２９７】

ステップＳ４２５：主制御基板３０は、ＤＥレジスタをインクリメント（エラーフラグのアドレスをセット）する。

ステップＳ４２６：主制御基板３０は、エラーフラグをロードする。

ステップＳ４２７：主制御基板３０は、エラーフラグとステータス今回値が一致するか否かを確認する。その結果、一致することを確認した場合（Ｙｅｓ）、主制御基板３０は、外部結合子（４）に移行してステップＳ４４６（図３４）を実行する。一方、一致を確認できない場合（Ｎｏ）、主制御基板３０はステップＳ４２８を実行する。

【０２９８】

ステップＳ４２８：主制御基板３０は、エラーフラグにステータス今回値をセーブする。

ステップＳ４２９：主制御基板３０は、ＤＥレジスタとＨＬレジスタを交換する。

【０２９９】

〔図３３：外部結合子（３）以降〕

ステップＳ４３０：主制御基板３０は、ＨＬレジスタにサブコマンド要求カウンタのアドレスをセットする。

ステップＳ４３１：主制御基板３０は、サブコマンド要求カウンタを１加算する。

ステップＳ４３２：主制御基板３０は、サブコマンド要求カウンタをロードする。

ステップＳ４３３：主制御基板３０は、ＨＬレジスタにサブコマンド要求カウンタ値を２回加算する。

ステップＳ４３４：主制御基板３０は、サブコマンド要求バッファに状態指定先行コマンド（８５Ｈ）をセーブする。

ステップＳ４３５：主制御基板３０は、ＨＬレジスタを１減算する。

ステップＳ４３６：主制御基板３０は、ステータス今回値をセットする。

ステップＳ４３７：主制御基板３０は、ステータス今回値を２倍する。

ステップＳ４３８：主制御基板３０は、ステータス今回値の２倍に後続コマンド値を加算する。

ステップＳ４３９：主制御基板３０は、サブコマンド要求バッファに加算結果をセーブする。

ステップＳ４４０：主制御基板３０は、ＤＥレジスタとＨＬレジスタを交換する。

【０３００】

〔図３４：外部結合子（５）以降〕

ステップＳ４４１：主制御基板３０は、ステータス今回値が０か否かを確認する。その結果、ステータス今回値が０であることを確認した場合（Ｙｅｓ）、主制御基板３０はステップＳ４４６を実行する。一方、ステータス今回値が０であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ４４２を実行する。

ステップＳ４４２：主制御基板３０は、エラー発生時設定対象ラムアドレス下位バイトをロードする。

ステップＳ４４３：主制御基板３０は、ロードした値が０か否かを確認する。その結果、ロードした値が０であることを確認した場合（Ｙｅｓ）、主制御基板３０はステップＳ

４４６を実行する。一方、ロードした値が０であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ４４４を実行する。

【０３０１】

ステップＳ４４４：主制御基板３０は、エラー発生時設定データをロードする。

ステップＳ４４５：主制御基板３０は、対象ラムアドレスにセーブする。

ステップＳ４４６：主制御基板３０は、ＨＬレジスタに３を加算する。

ステップＳ４４７：主制御基板３０は、ループ継続か否かを確認する。その結果、ループ継続であることを確認した場合（Ｙｅｓ）、主制御基板３０は、外部結合子（１）に移行してステップＳ４０４（図３１）を実行する。一方、ループ継続であることを確認できない場合（Ｎｏ）、主制御基板３０は、外部結合子（６）に移行してステップＳ４４８（図３５）を実行する。

【０３０２】

〔図３５：外部結合子（６）以降〕

ステップＳ４４８：主制御基板３０は、エラー管理処理テーブル２のアドレスをセットする。

ステップＳ４４９：主制御基板３０は、状態項目数をセットする。

ステップＳ４５０：主制御基板３０は、エラーフラグをロードする。

ステップＳ４５１：主制御基板３０は、エラーフラグと監視対象ビットを論理積する。

ステップＳ４５２：主制御基板３０は、ゼロフラグが１か否か（異常未発生か？）を確認する。その結果、ゼロフラグが１であることを確認した場合（Ｙｅｓ）、主制御基板３０はステップＳ４５４を実行する。一方、ゼロフラグが１であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ４５３を実行する。

【０３０３】

ステップＳ４５３：主制御基板３０は、Ａレジスタに１をセットする。

ステップＳ４５４：主制御基板３０は、ＨＬレジスタをインクリメント（エラー発生フラグのアドレスを指す）する。

ステップＳ４５５：主制御基板３０は、エラー発生フラグのアドレスをロードする。

ステップＳ４５６：主制御基板３０は、エラー発生フラグにＡレジスタをセーブする。

ステップＳ４５７：主制御基板３０は、ゼロフラグが１か否か（異常未発生か？）を確認する。その結果、ゼロフラグが１であることを確認した場合（Ｙｅｓ）、主制御基板３０は外部結合子（９）に移行してステップＳ４７２（図３６）を実行する。一方、ゼロフラグが１であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ４５８を実行する。

【０３０４】

ステップＳ４５８：主制御基板３０は、ＤＥレジスタとＨＬレジスタを交換する。

ステップＳ４５９：主制御基板３０は、ＨＬレジスタにサブコマンド要求カウンタのアドレスをセットする。

ステップＳ４６０：主制御基板３０は、サブコマンド要求カウンタを１加算する。

【０３０５】

〔図３６：外部結合子（７）以降〕

ステップＳ４６１：主制御基板３０は、サブコマンド要求カウンタをロードする。

ステップＳ４６２：主制御基板３０は、ＨＬレジスタにサブコマンド要求カウンタ値を２回加算する。

ステップＳ４６３：主制御基板３０は、サブコマンド要求バッファに状態指定先行コマンド（８５Ｈ）をセーブする。

ステップＳ４６４：主制御基板３０は、ＨＬレジスタを１減算する。

ステップＳ４６５：主制御基板３０は、後続コマンドをロードする。

ステップＳ４６６：主制御基板３０は、サブコマンド要求バッファに後続コマンドをセーブする。

ステップＳ４６７：主制御基板３０は、ＤＥレジスタとＨＬレジスタを交換する。

ステップＳ４６８：主制御基板３０は、エラー発生時設定対象ラムアドレス下位バイトをロードする。
ステップＳ４６９：主制御基板３０は、ロードした値が０か否かを確認する。その結果、値が０であることを確認した場合（Ｙｅｓ）、主制御基板３０はステップＳ４７２を実行する。一方、値が０であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ４７０を実行する。
【０３０６】
ステップＳ４７０：主制御基板３０は、エラー発生時設定データをロードする。
ステップＳ４７１：主制御基板３０は、ロードした内容を対象ラムアドレスにセーブする。
ステップＳ４７２：主制御基板３０は、ＨＬレジスタに３を加算する。
ステップＳ４７３：主制御基板３０は、ループ継続か否かを確認する。その結果、ループ継続であることを確認した場合（Ｙｅｓ）、主制御基板３０は、外部結合子（８）に移行してステップＳ４５０（図３５）を実行する。一方、ループ継続であることを確認できない場合（Ｎｏ）、主制御基板３０は、外部結合子（１０）に移行してステップＳ４７４（図３７）を実行する。
【０３０７】
〔図３７：外部結合子（１０）以降〕
ステップＳ４７４：主制御基板３０は、エラーフラグをロードする。
ステップＳ４７５：主制御基板３０は、ロードしたエラーフラグをスイッチエラーフラグと論理和する。
ステップＳ４７６：主制御基板３０は、ロードしたエラーフラグを盤面磁気１エラーフラグと論理和する。
ステップＳ４７７：主制御基板３０は、ロードしたエラーフラグを盤面磁気２エラーフラグと論理和する。
ステップＳ４７８：主制御基板３０は、ロードしたエラーフラグを電波エラーフラグと論理和する。
【０３０８】
ステップＳ４７９：主制御基板３０は、ゼロフラグが１か否か（上記エラー未発生か？）を確認する。その結果、ゼロフラグが１であること（上記エラー未発生）を確認した場合（Ｙｅｓ）、主制御基板３０は、呼び出し元に復帰する。一方、ゼロフラグが１であること（上記エラー未発生）を確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ４８０を実行する。
ステップＳ４８０：主制御基板３０は、不正検知１情報ビットタイマにタイマ値をセットする。
以上の手順を終了すると、主制御基板３０は呼び出し元に戻る。
【０３０９】
〔第１の技術的特徴のまとめ〕
以上のような状態管理処理及びエラー確認処理のプログラム構成によれば、使用領域及びデータ領域のコード量の増大を抑えることができる。
【０３１０】
〔第４の技術的特徴：変動パターン選択２処理におけるコード削減の構成〕
次に、本実施形態の第４の技術的特徴について説明する。第４の技術的特徴は、変動パターン選択２処理にけるコード使用量の増大を抑える構成である。
【０３１１】
〔第４の技術的特徴の概要〕
変動パターン選択２処理は、特別図柄の変動パターンを決定する処理において呼び出されるモジュールであり、遊技の進行（実行）に関する内容を制御するため、使用領域にプログラムを配置する必要がある。先ず、変動パターン決定処理の概要を説明する。
【０３１２】

〔変動パターン決定処理の概要〕
図３８は、変動パターン決定処理の手順例を示すフローチャートである。このフローチャートは、主制御基板３０（主制御ＣＰＵ３１）により実行される。
ステップＳ６００：主制御基板３０は、状態別及び停止図柄別テーブルアドレスをセットする。ここでセットされるテーブルアドレスは、以後の手順において最終的に変動パターンを決定（乱数抽選により選択）するための基礎となるテーブルマスタの場所を指定するためのものである。本実施形態では、はずれ時又は大当り時のいずれについても処理が共通となっているが、変動パターンの決定に用いるテーブルの場所（テーブルアドレス）については、現在の遊技状態別及び停止図柄別のものがＲＯＭに記憶したテーブルで規定されている。
【０３１３】
ステップＳ６０１：主制御基板３０は、作動記憶数別オフセット値を取得する。このオフセット値は、例えば今回の変動対象図柄が第１特別図柄であれば第１特別図柄作動記憶数カウンタ値、第２特別図柄であれば第２特別図柄作動記憶数カウンタ値に基づいて、作動記憶数別オフセット値テーブルから取得することができる。なお、作動記憶数別オフセット値は、第１特別図柄の作動記憶数及び第２特別図柄の作動記憶数を合計したカウンタ値に基づいて取得されるテーブル構成としてもよい。
【０３１４】
ステップＳ６０２：主制御基板３０は、取得した作動記憶数別オフセット値から、第１段階選択用テーブルをセットする。ここでいう「第１段階選択用テーブル」は、変動パターンの決定に関する３つの乱数（変動グループ決定乱数、変動モード決定乱数及び変動パターン決定乱数）のうち、特に変動グループ決定乱数を用いて変動パターンの第１カテゴリである「変動グループ」の抽選を行うテーブルである。
【０３１５】
ステップＳ６０３：主制御基板３０は、ＲＷＭの乱数記憶領域から変動グループ決定乱数をロードする。
ステップＳ６０４：主制御基板３０は、第１段階変動グループ決定処理（変動パターン選択１処理）を実行する。この処理は、上記のように変動グループ決定乱数を用いて変動パターンの第１カテゴリ抽選（変動グループの抽選）を行い、取得した変動グループ指定オフセット値を戻り値としてセットするものである。
【０３１６】
ステップＳ６０６：主制御基板３０は、先の戻り値である変動モード選択オフセットを用いて「第２段階選択用テーブル」をセットする。ここでいう「第２段階選択用テーブル」は、特に変動モード決定乱数を用いて変動パターンの第２カテゴリである「変動モード」の抽選を行うテーブルである。
【０３１７】
ステップＳ６０７：主制御基板３０は、ＲＷＭ乱数記憶領域から変動モード決定乱数をロードする。
ステップＳ６０８：主制御基板３０は、第２段階変動モード決定処理（本実施形態の変動パターン選択２処理）を実行する。この処理は、上記のように変動モード決定乱数を用いて変動パターンの第２カテゴリ抽選（変動モードの抽選）を行い、取得した変動モード指定オフセット（パターン選択オフセット）値を戻り値としてセットするものである。ここで取得した変動モード指定オフセット（パターン選択オフセット）は、最終第３段階の変動パターン決定処理で用いられる変動パターン選択テーブルのアドレスを指定するものとなる。なお、具体的な処理の内容については、比較例との対比とともに後述する。
【０３１８】
ステップＳ６０９：主制御基板３０は、戻り値である変動モード指定オフセット（パターン選択オフセット）を用いて第３段階選択用テーブル（変動パターン選択テーブル）をセットする。具体的には、上記のように変動モード指定オフセットの値で示されるテーブルアドレスをＲＯＭ上で指定する。

【０３１９】
ステップＳ６１０：テーブルアドレスを指定すると、主制御基板３０は、変動パターン決定乱数をロードする。
ステップＳ６１１：主制御基板３０は、第３段階変動パターン決定処理を実行する。この処理では、主制御基板３０は変動パターン決定乱数に基づき、上記の「第３段階選択用テーブル（変動モード別変動パターン選択テーブル）」から対応する変動パターン番号を選択する。また主制御基板３０は、選択した変動パターン番号を戻り値にセットする。ここで選択した変動パターン番号は、最終的に当該変動で行われる一意の変動パターン（変動開始から予告の発生、テンパイまでの時間、リーチ態様、リーチ変動時間、最終停止表示に至るまでの変動態様）を表すものであり、この変動パターン番号から当該変動におけるトータルの変動時間（変動秒数）が一意に定まることになる。
【０３２０】
ステップＳ６１２：主制御基板３０は、戻り値として選択した変動パターン番号から変動パターンコマンドを生成する。生成した変動パターンコマンドは、サブコマンド送信処理で演出制御基板４０に送信される。そして、演出制御基板４０は、受け取った変動パターンコマンドから当該変動の変動パターン情報を把握したり、変動演出パターンを選択したりすることができる。
【０３２１】
〔第２の技術的特徴の課題〕
上記のように、変動パターン選択２処理において使用領域のプログラム使用量（コード量）の増大を抑えたい。
一般に、データテーブル構成においてデータＡとデータＢがペアになっており、データＡの範囲が０～１０、データＢの範囲が０～４０００であるとする場合を考える。この場合、普通にデータＡとデータＢをテーブルに記載した場合、データＡの記載に１バイト、データＢの記載に２バイト、合計３バイトが必要となる。
ここで、データＡ×４０９６＋データＢで構成した別の「データＣ」を考えた場合、データＣの範囲は０～５３９６０であるため、この構成であれば、２バイトのデータとして記載できる。
【０３２２】
例えば、本実施形態の変動パターン２選択処理（ＨＰＴ＿ＭＯＤ）で読み込むテーブルには、４ビットのモード番号データと１２ビットの比較値データからなる１６ビットのデータが記載されている。このようなテーブルデータを扱う場合、比較例として以下の構成が考えられる。
【０３２３】
〔比較例のデータ処理構成〕
比較例では、モード番号データと比較値データの組をテーブルデータから先ず１６ビットのデータとして読み込み、上位４ビットを０にすることで、比較値データを生成する。そして、比較値データと乱数値を比較し、条件を満たした場合、再度テーブルからデータを読み込み、今度は上位４ビットを有効にするという処理を行う。
【０３２４】
具体的には、比較例において変動パターン選択２処理を呼び出すにあたり、Ａレジスタにモード選択オフセット、ＢＣレジスタに乱数値をセットしておくこととする。本処理では、最終的にＡレジスタにモード番号データ、Ｂレジスタにパターン選択オフセットがセットされる。このため本処理では、比較値データ及びモード番号データをテーブルからロードした後、上位バイトのモード番号データ部分をマスクし、比較値データ部分を抽出する。そして、乱数値が比較値未満となった場合、改めて比較値（の上位４ビット）及びモード番号データをロードし、４ビット右シフトすることで、モード番号を抽出する。比較例の場合、使用コード量は２２バイトである。
【０３２５】
図３９は、比較例の変動パターン選択２処理のプログラムを示す図である。比較例のプ

ログラム（コード）は、以下の通りである。なお、図３９のプログラムコード中、各行に付記している括弧書きの数値は、使用バイト数を示している。

【０３２６】

「ＨＰＴ＿ＭＯＤ：」は、変動パターン選択２処理の開始を示すラベルである。

〔モード選択テーブルアドレス選択処理〕

（２）「ＬＤＦ　ＨＬ，Ｄ＿ＭＯＤ＿ＳＥＬ」は、モード選択テーブルのアドレスをセットする命令である。ここでのコード量は２バイトである。

（１）「ＲＳＴ　ＷＯＲＤＳＥＬ」は、ワードデータ選択処理を呼び出す処理である。ここでのコード量は１バイトである。

【０３２７】

「ＨＰＴ＿ＭＯＤ＿１０：」は、比較値取得処理の開始を示すラベルである。

〔比較値取得処理〕

（２）「ＬＤＩＮ　ＡＥ，（ＨＬ）」は、比較値及びモード番号データをロードしてテーブルアドレスを２加算する命令である。ここでのコード量は２バイトである。

（２）「ＡＮＤ　Ａ，００ＦＨ」は、Ａレジスタを［００ＦＨ］と論理積する命令である。ここでのコード量は２バイトである。

（１）「ＬＤ　Ｄ，Ａ」は、Ｄレジスタに抽出値（Ａレジスタ）をセットする命令である。ここでのコード量は１バイトである。

【０３２８】

〔モード判定処理〕

（２）「ＣＰ　ＢＣ，ＤＥ」は、ＢＣレジスタの乱数値とＤＥレジスタの比較値を比較する命令である。ここでのコード量は２バイトである。

（２）「ＪＲ　Ｃ，ＨＰＴ＿ＭＯＤ＿２０」は、キャリーフラグが１であれば「ＨＰＴ＿ＭＯＤ＿２０」に分岐する命令である。ここでのコード量は２バイトである。

【０３２９】

〔テーブル更新処理〕

（１）「ＩＮＣ　ＨＬ」は、ＨＬレジスタで示すテーブルアドレスを１加算して次のデータブロックへ更新する命令である。ここでのコード量は１バイトである。

（２）「ＪＲ　ＨＰＴ＿ＭＯＤ＿１０」は、「ＨＰＴ＿ＭＯＤ＿１０」にジャンプする命令である。ここでのコード量は２バイトである。

【０３３０】

「ＨＰＴ＿ＭＯＤ＿２０：」は、パターン選択オフセット設定処理の開始を示すラベルである。

〔パターン選択オフセット設定処理〕

（１）「ＬＤ　Ｂ，（ＨＬ）」は、ＨＬレジスタで示す対象アドレスの内容をＢレジスタにロードする命令である。ここでのコード量は１バイトである。

〔モード番号データ設定処理〕

（１）「ＤＥＣ　ＨＬ」は、ＨＬレジスタのテーブルアドレスを１減算する命令である。ここでのコード量は１バイトである。

（１）「ＬＤ　Ａ，（ＨＬ）」は、ＨＬレジスタで示す対象アドレスの内容をロードする命令である。ここでのコード量は１バイトである。

（３）「ＳＲＬ　Ａ，４」は、Ａレジスタを４ビット右シフトしてモード番号データを算出する命令である。ここでのコード量は３バイトである。

（１）「ＲＥＴ」は、呼び出し元に復帰する命令である。ここでのコード量は１バイトである。

【０３３１】

図４０は、比較例の変動パターン選択２処理の手順例を示すフローチャートである。このフローチャートは、図３９のプログラムに対応している。ここでは便宜上、比較例の主制御基板が処理を実行することとする。

【０３３２】

ステップＳ７００：主制御基板は、ＨＬレジスタにモード選択テーブルのアドレスをセットする。
ステップＳ７０１：主制御基板は、ワードデータ選択処理を呼び出す処理を実行する。
ステップＳ７０２：主制御基板は、比較値及びモード番号データをロードし、ＨＬレジスタを２加算する。
ステップＳ７０３：主制御基板は、Ａレジスタと［０ＦＨ］を論理積し、比較値の上位４ビットを抽出する。
ステップＳ７０４：主制御基板は、ＤレジスタにＡレジスタをセットする。
ステップＳ７０５：主制御基板は、乱数値（ＢＣレジスタ）と比較値（ＤＥレジスタ）を比較する。
ステップＳ７０６：主制御基板は、キャリーフラグが１か否か（乱数値が比較値未満か？）を確認する。その結果、キャリーフラグが１であることを確認した場合（Ｙｅｓ）、主制御基板はステップＳ７０８を実行する。一方、キャリーフラグが１であることを確認できない場合（Ｎｏ）、主制御基板は、ステップＳ７０７を実行する。
ステップＳ７０７：主制御基板は、ＨＬレジスタを１加算してステップＳ７０２を実行する。

【０３３３】

ステップＳ７０８：主制御基板は、Ｂレジスタにパターン選択オフセットをロードする。
ステップＳ７０９：主制御基板は、ＨＬレジスタを１減算する。
ステップＳ７１０：主制御基板は、比較値及びモード番号の上位バイトデータをロードする。
ステップＳ７１１：主制御基板は、Ａレジスタの内容を４ビット右シフトし、モード番号を抽出する。
以上の手順を終了すると、主制御基板は呼び出し元に戻る。

【０３３４】

〔本実施形態の変動パターン選択２処理の構成〕

次に、本実施形態の変動パターン選択２処理の構成について説明する。
本実施形態においても、比較例と同様に、変動パターン選択２処理を呼び出すにあたり、Ａレジスタにモード選択オフセット、ＢＣレジスタに乱数値をセットしておくこととする。そして本実施形態では、Ａレジスタにモード番号データ、Ｂレジスタにパターン選択オフセットがセットされることとなる。本実施形態の変動パターン選択２処理で使用するコード量は１６バイトである。

【０３３５】

図４１は、本実施形態の変動パターン選択２処理のプログラム関係を示す図である。図４１中（Ａ）がプログラム本体を示し、図４１中（Ｂ）がＨＬレジスタにより指定されるテーブルを示す。なお、図４１のプログラムコード中、各行頭に付記している数値は、使用バイト数を示している。

【０３３６】

「ＨＰＴ＿ＭＯＤ」は、変動パターン選択２処理の開始を示すラベルである。

〔モード選択テーブルアドレス選択処理〕

（２）「ＬＤＦ　ＨＬ，Ｄ＿ＭＯＤ＿ＳＥＬ」は、モード選択テーブルのアドレスをセットする命令である。ここでのコード量は２バイトである。
（１）「ＲＳＴ　ＷＯＲＤＳＥＬ」は、ワードデータ選択処理を呼び出す処理である。ここでのコード量は１バイトである。

【０３３７】

「ＨＰＴ＿ＭＯＤ＿１０：」は、比較値取得処理の開始を示すラベルである。

〔比較値取得処理〕

（２）「ＬＤＩＮ　ＡＥ，（ＨＬ）」は、比較値及びモード番号データをロードしてテーブルアドレスを２加算する命令である。ここでは、テーブルから比較値及びモード番号をＡＥレジスタにロードしつつ、テーブルアドレスに２を加算する。この結果、Ａレジスタ

の上位４ビットはモード番号、下位４ビットは比較値の上位バイト、Ｅレジスタは比較値の下位バイトが設定される。ここでのコード量は２バイトである。
（１）「ＩＮＣ　ＨＬ」は、ＨＬレジスタをインクリメントする命令である。テーブルアドレスはパターン選択オフセットを指し示しているため、ここではＨＬレジスタが示すテーブルアドレスに１を加算し、次の比較値及びモード番号を指し示す状態にする。ここでのコード量は１バイトである。
【０３３８】
〔モード番号データ設定処理〕
（３）「ＤＩＶ　Ｄ，Ａ，１６」は、モード番号を算定する命令である。「ＤＩＶ　Ｄ，Ａ，１６」命令は、「モード番号と比較値の分離」及び「モード番号の右シフト」を単一の命令で実行するものである。すなわち、ここではＡレジスタを１６で除算し、Ａレジスタに商（モード番号データ）、Ｄレジスタに余り（比較値の上位バイト）を設定する処理である。この結果、ＤＥレジスタには乱数との比較値が設定（比較値データが生成）される。ここでのコード量は３バイトである。
【０３３９】
〔モード判定処理〕
（２）「ＣＰ　ＢＣ，ＤＥ」は、ＢＣレジスタの乱数値とＤＥレジスタの比較値の内容を比較する命令である。ここでは、乱数値（ＢＣレジスタ）と比較値（ＤＥレジスタ）を比較し、キャリーフラグをセットする。例えば、ＤＥレジスタの値が３３６８である場合、ＢＣレジスタの値が０～３３６７であればキャリーフラグがセット（Ｃ）され、３３６８以上であればキャリーフラグがリセット（ＮＣ）される。ここでのコード量は２バイトである。
（２）「ＪＲ　ＮＣ，ＨＰＴ＿ＭＯＤ＿１０」は、キャリーフラグが０であれば「ＨＰＴ＿ＭＯＤ＿１０」に分岐する命令である。ここでは、上で設定したキャリーフラグを確認し、リセット（ＮＣ）であれば、「ＨＰＴ＿ＭＯＤ＿１０」に戻る。ＨＬレジスタは次の比較値及びモード番号を指しているため、戻った先のＬＤＩＮ命令で次の値をロードすることができる。ここでのコード量は２バイトである。
【０３４０】
〔パターン選択オフセット設定処理〕
（１）「ＤＥＣ　ＨＬ」は、ＨＬレジスタをデクリメントする命令である。すなわち、上記の「ＩＮＣ　ＨＬ」命令で進めたテーブルアドレスを１減算し、再びパターン選択オフセットを指す状態とする。ここでのコード量は１バイトである。
（１）「ＬＤ　Ｂ，（ＨＬ）」は、対象アドレスの内容をロードする命令である。ここでは、Ｂレジスタをパターン選択オフセットで上書きする。ここでのコード量は１バイトである。
（１）「ＲＥＴ」は、呼び出し元に復帰する命令である。ここでのコード量は１バイトである。
　モード番号データは、モード番号データ設定処理でＡレジスタに設定された状態を保持しており、比較例のように再びテーブルから値を読み出して加工する必要はない。
【０３４１】
〔テーブル構成例〕
　モード選択テーブルアドレス選択処理では、呼び出し時に設定されたモード選択オフセット（Ａレジスタ）に応じて、比較値、モード番号データ及びパターン選択オフセットが記載されたテーブルのアドレスをＨＬレジスタに設定する。図４１中（Ｂ）がテーブル構成例である。
　「Ｄ＿ＭＯＤ＿ＳＥＬ＿０７：」は、テーブルの先頭を示すラベルである。
　「ＤＷ　３３６８　＋　１０００Ｈ　＊　００４Ｈ」は、比較値とモード番号データの合成値である。
　「ＤＢ　＠Ｄ＿ＰＡＴ＿ＳＥＬ＿１０　　」は、パターン選択１０オフセットである。
　「ＤＷ　３８６８　＋　１０００Ｈ　＊　００５Ｈ」は、比較値とモード番号データの合

成値である。
「ＤＢ　＠Ｄ＿ＰＡＴ＿ＳＥＬ＿１１」は、パターン選択１１オフセットである。
「ＤＷ　＠ＳＴＰ　＋　１０００Ｈ　＊　００６Ｈ」は、比較値とモード番号データの合成値である。
「ＤＢ　＠Ｄ＿ＰＡＴ＿ＳＥＬ＿１２」は、パターン選択１２オフセットである。
【０３４２】
〔テーブルを使用した振分〕
図４２は、テーブル「Ｄ＿ＭＯＤ＿ＳＥＬ＿０７」を使用した振り分けを示す図である。すなわち、図４１中（Ｂ）のテーブルを選択した場合、ＢＣレジスタに設定された乱数値に応じて以下の振り分けを行うことになる。なお、処理の呼び出し時に設定される乱数は、０～４０９２までの４０９３通りであるとする。
取得した乱数値が０～３３６７の場合、モード番号データは０４Ｈとなり、パターン選択オフセットは「１０（＠Ｄ＿ＰＡＴ＿ＳＥＬ＿１０）」となる。
取得した乱数値が３３６８～３８６７の場合、モード番号データは０５Ｈとなり、パターン選択オフセットは「１１（＠Ｄ＿ＰＡＴ＿ＳＥＬ＿１１）」となる。
取得した乱数値が３８６８以上の場合、モード番号データは０６Ｈとなり、パターン選択オフセットは「１２（＠Ｄ＿ＰＡＴ＿ＳＥＬ＿１２）」となる。
【０３４３】
図４３は、本実施形態の変動パターン選択２処理の手順例を示すフローチャートである。このフローチャートは、図４１中（Ａ）のプログラムに対応している。変動パターン選択２処理は、主制御基板３０（主制御ＣＰＵ３１）が実行する。
ステップＳ８００：主制御基板３０は、ＨＬレジスタにモード選択テーブルのアドレスをセットする。
ステップＳ８０１：主制御基板３０は、ワードデータ選択処理を呼び出す処理を実行する。
ステップＳ８０２：主制御基板３０は、比較値及びモード番号データをロードし、ＨＬレジスタを２加算する。
ステップＳ８０３：主制御基板３０は、ＨＬレジスタを１加算する。
ステップＳ８０４：主制御基板３０は、Ａレジスタを１６で除算し、商（モード番号データ）をＡレジスタ、余り（比較値）をＤレジスタにセットする。
ステップＳ８０５：主制御基板３０は、乱数値（ＢＣレジスタ）と比較値（ＤＥレジスタ）を比較する。
ステップＳ８０６：主制御基板３０は、キャリーフラグが０であるか否か（乱数値が比較値以上か？）を確認する。その結果、キャリーフラグが０であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ８０２を実行する。一方、キャリーフラグが０であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ８０７を実行する。
【０３４４】
ステップＳ８０７：主制御基板３０は、ＨＬレジスタを１減算する。
ステップＳ８０８：主制御基板３０は、Ｂレジスタにパターン選択オフセットをロードする。
以上の手順を終了すると、主制御基板３０は、呼び出し元に戻る。
【０３４５】
〔第４の技術的特徴のまとめ〕
以上のように、本実施形態では、変動パターン選択２処理のプログラムコード量を比較例の場合よりも６バイト削減することができる。したがって、プログラム容量の増大を抑えることができる。
【０３４６】
〔第５の技術的特徴：ベース、役物比率及び連続役物比率の算定タイミングの構成〕
次に、本実施形態の第５の技術的特徴について説明する。第５の技術的特徴は、管理遊

技機１００の性能を算出する処理を実行するタイミングに関する構成である。
【０３４７】
〔第５の技術的特徴の課題〕
管理遊技機１００の性能として、ベース値、役物比率及び連続役物比率がある。これらの性能値を算出する処理は、タイマ割込み処理中の性能表示モニタ制御処理（Ｅ＿ＳＨＹＭＴ）で実行する。このとき、ベース、役物比率及び連続役物比率の算定時間が長くなることを抑えたい。
【０３４８】
〔第５の技術的特徴の概要〕
タイマ割込み処理を終了させる必要がある時間が限られるため、１回のタイマ割込み処理の中では、ベース値、役物比率及び連続役物比率をそれぞれ算定する処理について、２つ以上の処理を実行しない構成とする。なお、ベース値、役物比率及び連続役物比率は、それぞれ以下の条件を満たした場合に算定することとする。
【０３４９】
ベース値を算定するのは、以下の「ア」または「イ」を満たした場合である。
ア　区間内の総アウト数が２９９（初回のみ）又は６００００個に達した場合
イ　「現在のベース値」の表示開始タイミング
【０３５０】
役物比率を算定するのは、上記の「ア」及び「イ」を満たさず、かつ、以下の「ウ」～「オ」を満たした場合である。
ウ　前回のタイマ割込みでベース値を算定した場合
エ　「未定の賞球」を「連続役物賞球カウンタ」に加算した場合
オ　今回のタイマ割込みで賞球が発生した場合
【０３５１】
連続役物比率を算定するのは、上記の「ア」～「オ」を満たさず、かつ、以下の「カ」を満たした場合である。
カ　前回のタイマ割込みで役物比率を算定した場合
【０３５２】
〔性能モニタ表示処理〕
図４４から図４６は、性能表示モニタ表示処理の手順例を示すフローチャートである。性能モニタ表示処理は、タイマ割込み処理の中で呼び出し実行されるモジュール（図１３のステップＳ３１９）である。本実施形態の第５の技術的特徴となる構成は、性能モニタ表示処理の中に組み込まれているので、以下、性能モニタ表示処理について説明する。性能モニタ表示処理は、主制御基板３０（主制御ＣＰＵ３１）が実行する。
【０３５３】
ステップＳ９００：主制御基板３０は、遊技可能状態か否かを確認する。その結果、遊技許可状態であることが確認できない場合（Ｎｏ）、主制御基板３０は、外部結合子（２）に移行してステップＳ９１７（図４５）を実行する。一方、遊技許可状態であることが確認できた場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９０１を実行する。
ステップＳ９０１：主制御基板３０は、今回アウト数の算定処理を実行する。
ステップＳ９０２：主制御基板３０は、役物作動確認処理を実行する。
ステップＳ９０３：主制御基板３０は、今回賞球数の算定処理を実行する。
ステップＳ９０４：主制御基板３０は、総賞球カウンタ更新処理を実行する。
ステップＳ９０５：主制御基板３０は、役物賞球カウンタ更新処理を実行する。
ステップＳ９０６：主制御基板３０は、未定賞球カウンタ更新処理を実行する。
ステップＳ９０７：主制御基板３０は、大当り中又は小当り中であるか否かを確認する。その結果、大当り中又は小当り中であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９０８を実行する。一方、大当り中又は小当り中であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ９１１を実行する。
【０３５４】

ステップＳ９０８：主制御基板３０は、大当り中か否かを確認する。その結果、大当り中であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９０９を実行する。一方、大当り中であることを確認できない場合（Ｎｏ）、主制御基板３０は、外部結合子（１）に移行してステップＳ９１２（図４５）を実行する。

【０３５５】

ステップＳ９０９：主制御基板３０は、算定管理バッファに「次回役物比率算定指定値」をセーブする。

ステップＳ９１０：主制御基板３０は、連続役物賞球カウンタに未定賞球カウンタを加算する。

ステップＳ９１１：主制御基板３０は、未定賞球カウンタをクリアする。

【０３５６】

〔図４５：結合子（１）以降〕

ステップＳ９１２：主制御基板３０は、通常遊技時であるか否かを確認する。その結果、通常遊技時であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９１３を実行する。一方、通常遊技時であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ９１５を実行する。

【０３５７】

ステップＳ９１３：主制御基板３０は、通常賞球カウンタ更新処理を実行する。

ステップＳ９１４：主制御基板３０は、通常アウトカウンタ更新処理を実行する。

ステップＳ９１５：主制御基板３０は、総アウトカウンタ更新処理を実行する。

ステップＳ９１６：主制御基板３０は、総アウトカウンタが比較値以上であるか否かを確認する。比較値は、２９９個（初回のみ）又は６００００個である。その結果、総アウトカウンタが比較値以上であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９１９を実行する。一方、総アウトカウンタが比較値以上であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ９１７を実行する。

【０３５８】

ステップＳ９１７：主制御基板３０は、性能表示用タイマ更新処理を実行する。

ステップＳ９１８：主制御基板３０は、ベース値（現在）の表示開始タイミングか否かを確認する。なお、管理遊技機１００の性能表示モニタ３７には、「現在のベース値」、「前回区間のベース値」、「前々回区間のベース値」及び「３回前区間のベース値」を約５秒ごとに切り替えて表示する。このとき、「３回目区間のベース値」から「現在のベース値」に表示を切り替えるタイミングが表示開始タイミングとなるので、その確認を行う。その結果、現在のベース値の表示開始タイミングであることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９１９を実行する。一方、表示開始タイミングであることを確認できない場合（Ｎｏ）、主制御基板３０は、外部結合子（３）に移行してステップＳ９２２（図４６）を実行する。

【０３５９】

ステップＳ９１９：主制御基板３０は、ベース値の算定処理を実行する。

ステップＳ９２０：主制御基板３０は、ベース値の更新処理を実行する。

ステップＳ９２１：主制御基板３０は、算定管理バッファに「次回役物比率算定指定値」をセーブする。

【０３６０】

〔図４６：外部結合子（３）以降〕

ステップＳ９２２：主制御基板３０は、今回賞球数が０以外であるか否かを確認する。その結果、今回賞球数が０以外であること確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９２６を実行する。一方、今回賞球数が０以外であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ９２３を実行する。

【０３６１】

ステップＳ９２３：主制御基板３０は、算定管理バッファをロードする。

ステップＳ９２４：主制御基板３０は、ロードした算定管理バッファが「次回役物比率

算定指定値」であるか否かを確認する。その結果、「次回役物比率算定指定値」であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９２６を実行する。一方、「次回役物比率算定指定値」であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ９２５を実行する。

【０３６２】

ステップＳ９２５：主制御基板３０は、ロードした算定管理バッファが「次回連続役物比率算定指定値」であるか否かを確認する。その結果、「次回連続役物比率算定指定値」であることを確認した場合（Ｙｅｓ）、主制御基板３０は、ステップＳ９２８を実行する。一方、「次回連続役物比率算定指定値」であることを確認できない場合（Ｎｏ）、主制御基板３０は、ステップＳ９３０を実行する。

【０３６３】

ステップＳ９２６：主制御基板３０は、役物比率算定処理を実行する。

ステップＳ９２７：主制御基板３０は、算定管理バッファに「次回連続役物比率算定指定値」をセーブする。

ステップＳ９２８：主制御基板３０は、連続役物比率算定処理を実行する。

ステップＳ９２９：主制御基板３０は、算定管理バッファをクリアする。

ステップＳ９３０：主制御基板３０は、識別セグデータ設定処理を実行する。

ステップＳ９３１：主制御基板３０は、比率セグデータ設定処理を実行する。

以上の手順を終了すると、主制御基板３０は呼び出し元に戻る。

【０３６４】

〔第５の技術的特徴のまとめ〕

以上のように、性能表示モニタ表示処理において、性能値としてベース値の算定処理（ステップＳ９１９）、役物比率算定処理（ステップＳ９２６）及び連続役物比率算定処理（ステップＳ９２８）があるが、これらの処理は、１回あたりのタイマ割込み処理の中では２つ以上が実行されない。

【０３６５】

より詳細には、ベース値を算定する必要があるのは、「区間内の総アウト数が２９９個（初回のみ）又は６００００個に達した場合」、及び「現在のベース値」の表示開始タイミングである。また、役物比率を算定する必要があるのは、「賞球が発生した場合」であり、連続役物比率を算定する必要があるのは、「賞球が発生した場合」及び「未定の賞球を連続役物比率対象の賞球に加算した場合」である。

【０３６６】

しかし、本実施形態の性能表示モニタ表示処理においては、ベース値、役物比率及び連続役物比率について、同時に複数の算定が必要となる（条件を満たす）場合があったとしても、「ベース値＞役物比率＞連続役物比率」の優先度を設定している。このため、２つ以上の算定が必要となったとしても、優先度の順位に従って、最も優先度の高い１つのみを算定することとしている。このため、算定が必要となったものの残りは、次回以降のタイマ割込み処理の中で算定することになる。

【０３６７】

以下、本実施形態の性能表示モニタ表示処理のうち、第５の技術的特徴に関係する部分のプログラムについて説明する。

【０３６８】

〔未定賞球カウンタ加算処理〕

図４７は、未定賞球カウンタ加算処理（図４４のステップＳ９０６）のプログラムを示す図である。本実施形態では、未定賞球カウンタ（Ｒ＿ＥＲＷ＿ＭＴＳ）に今回連続役物賞球数（Ｒ＿ＥＲＷ＿ＰＲＮ）を加算することとしている。「今回連続役物賞球数」は、今回のタイマ割込み処理の中で、（連続役物作動時の）大入賞口入賞により獲得した賞球数の合計である。

【０３６９】

未定賞球カウンタ加算処理のプログラムは、以下の通りである。なお、図４７のプログ

ラムコード中、各行に付記している括弧書きの番号は、説明の便宜のために付している。
【０３７０】
「Ｅ＿ＳＨＹＭＴ＿３０：」は、未定賞球カウンタ加算処理の開始を示すラベルである。
（中略）
【０３７１】
〔未定賞球カウンタ更新処理〕
〔１〕「ＬＤＱ　Ａ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＰＲＮ）」は、今回連続役物賞球数をロードする命令である。ここでは、今回連続役物賞球数をロードする。
〔２〕「ＬＤＱ　ＨＬ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋０）」は、未定賞球カウンタ＋０をロードする命令である。ここでは、未定賞球カウンタの下位２バイトをロードする。本実施形態の管理遊技機１００は、いわゆる１種２種の遊技を混合した仕様であることから、大入賞口への入賞で得られる賞球は、小当り中であることもあり、賞球数を連続役物比率算定の数に入れるか否かは大当りとなるまで確定しない。そこで本実施形態では、大当り中か否かを問わず、大入賞口への入賞による賞球数は、必ず一度未定の賞球に加算することとしている。
〔３〕「ＡＤＤＷＢ　ＨＬ，Ａ」は、ＨＬレジスタにＡレジスタを加算する命令である。ここでは、未定賞球カウンタと今回連続役物賞球数の和を算定する。
〔４〕「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋０），ＨＬ」は、未定賞球カウンタ＋０にセーブする命令である。ここでは、算定結果を今回連続役物賞球数の下位２バイトにセーブする。なお、〔３〕においてオーバーフローが発生した場合、キャリーフラグがセットされ、発生していない場合はリセットされている。
〔５〕「ＬＤＱ　Ａ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋２）」は、未定賞球カウンタ＋２をロードする命令である。ここでは、未定賞球カウンタの上位１バイトをロードする。
〔６〕「ＡＤＣ　Ａ，０」は、未定賞球カウンタの３バイト目にキャリーフラグを加算する命令である。ここでは、「未定賞球カウンタの上位１バイト＋０＋キャリーフラグ」の和を算定する。すなわち、キャリーフラグがセットであれば、未定賞球カウンタの上位１バイトに１を加算することになる。
〔７〕「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋２），Ａ」は、未定賞球カウンタ＋２にセーブする命令である。ここでは、〔６〕で算定した結果を新たな未定賞球カウンタの上位１バイトとしてセーブする。
【０３７２】
〔特別遊技管理フェーズ等の確認〕
本実施形態の性能表示モニタ表示処理では、遊技機の遊技状態及び未定賞球カウンタの値を確認し、連続役物比率対象の賞球への振り替えを行うか否かを決定している。これは、図４４のステップＳ９０７～ステップＳ９０９等で行う内容に相当する。
【０３７３】
図４８は、特別遊技管理フェーズ等の確認を実行するプログラムを示す図である。以下、同様に説明する。
〔１〕「ＬＤ　Ａ，（Ｒ＿ＴＯＫ＿ＰＨＳ）」は、特別遊技管理フェーズをロードする命令である。
〔２〕「ＪＣＰ　Ｃ，Ａ，＠ＳＨＴ＿ＰＲＥ，Ｅ＿ＳＨＹＭＴ＿４２」は、小当り大入賞口開始ウエイト状態指定値と比較し、キャリーフラグが１であれば分岐する命令である。ここでは、特別遊技管理フェーズの値を確認し、小当りでも大当りでもなければ、「Ｅ＿ＳＨＹＭＴ＿４２」へ分岐する。
〔３〕「ＬＤ　Ａ，（Ｒ＿ＢＩＧ＿ＦＬＧ）」は、条件装置作動中フラグをロードする命令である。
〔４〕「ＪＣＰ　ＮＺ，Ａ，＠ＴＺ＿＿ＢＨＴ，Ｅ＿ＳＨＹＭＴ＿４３」は、特別図柄大当り情報と比較し、ゼロフラグが０であれば分岐する命令である。ここでは、条件装置作動中フラグの値を確認し、大当りでなければ「Ｅ＿ＳＨＹＭＴ＿４３」へ分岐する。
〔５〕「ＪＴＱ　ＮＺ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋２），Ｅ＿ＳＨＹＭＴ＿４１」は

、未定賞球カウンタ＋２が０以外であれば分岐する命令である。
〔５〕「ＪＴＷＱ　Ｚ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋０），Ｅ＿ＳＨＹＭＴ＿４３」は、未定賞球カウンタ＋０，＋１が０であれば分岐する命令である。
　上の２つの〔５〕では、遊技状態が大当り中である場合、未定賞球カウンタの値を確認する。そして、未定賞球カウンタの値が０でない場合は、「Ｅ＿ＳＨＹＭＴ＿４１」以降の処理を行い、０である場合は「Ｅ＿ＳＨＹＭＴ＿４３」へ分岐する。
【０３７４】
　「Ｅ＿ＳＨＹＭＴ＿４１：」は、上記〔５〕の分岐先アドレスを示すラベルである。
〔６〕「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＴＳＫ），＠ＥＸＤ＿ＴＳＫ＿ＹＫＨ」は、算定管理バッファに次回役物比率算定指定値をセーブする命令である。ここでは、算定管理バッファに「次回役物比率算定指定値」をセーブする。
　なお、ここで「次回連続役物比率算定指定値」ではなく、「次回役物比率算定指定値」をセーブしている理由は、このとき算定管理バッファには既に「次回役物比率算定指定値」が設定されている場合があることを考慮したものである。
【０３７５】
〔連続役物賞球カウンタ等更新処理〕
　本実施形態の性能表示モニタ表示処理では、未定賞球カウンタの賞球を連続役物賞球カウンタへ移し替える処理（図４４のステップＳ９０５，ステップＳ９１０，ステップＳ９１１等）を行っている。以下、これらを連続役物賞球カウンタ等更新処理として説明する。
【０３７６】
　図４９は、連続役物賞球カウンタ等更新処理のプログラムを示す図である。同様に説明する。
〔連続役物賞球カウンタ更新処理〕
〔１〕「ＬＤＱ　ＨＬ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＲＹＳ＋０）」は、連続役物賞球カウンタ＋０をロードする命令である。
〔２〕「ＬＤＱ　ＤＥ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋０）」は、未定賞球カウンタ＋０をロードする命令である。上記〔１〕と〔２〕では、連続役物賞球カウンタ及び未定賞球カウンタの下位２バイトをロードする。
〔３〕「ＡＤＤ　ＨＬ，ＤＥ」は、連続役物賞球カウンタに未定賞球カウンタを加算する命令である。
〔４〕「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＲＹＳ＋０），ＨＬ」は、連続役物賞球カウンタ＋０にセーブする命令である。上記〔３〕で和を求め、〔４〕で連続役物賞球カウンタの下位２バイトをセーブする。
〔５〕「ＬＤＱ　Ａ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＲＹＳ＋２）」は、連続役物賞球カウンタ＋２をロードする命令である。
〔６〕「ＬＤＱ　ＨＬ，ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋２」は、未定賞球カウンタ＋２のアドレスをセットする命令である。上記〔５〕では連続役物賞球カウンタの上位１バイトをロードし、〔６〕では未定賞球カウンタの上位１バイトのアドレスをセットする。
〔７〕「ＡＤＣ　Ａ，（ＨＬ）」は、連続役物賞球カウンタの３バイト目に未定賞球カウンタの３バイト目及びキャリーフラグを加算する命令である。
〔８〕「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＲＹＳ＋２），Ａ」は、連続役物賞球カウンタ＋２にセーブする命令である。上記〔７〕では連続役物賞球カウンタの上位１バイトと、未定賞球カウンタの上位１バイト及びキャリーフラグの和を算定し、〔８〕ではその結果をセーブする。
【０３７７】
　「Ｅ＿ＳＨＹＭＴ＿４２：」は、未定賞球カウンタクリア処理の開始を示すラベルである。
〔未定賞球カウンタクリア処理〕
〔９〕「ＣＬＲＷＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋０）」は、未定賞球カウンタ＋０をクリアする命令である。

〔１０〕「ＣＬＲＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＭＴＳ＋２）」は、未定賞球カウンタ＋２をクリアする命令である。上記〔９〕及び〔１０〕では未定賞球カウンタのクリアを行う。このクリアは、遊技状態が大当り及び小当りのどちらでもない場合にも行われる。

【０３７８】

〔ベース値算定後の算定管理バッファ設定処理〕

本実施形態の性能表示モニタ制御処理では、ベース値の算定を行った場合、算定管理バッファに次回役物比率算定指定値をセーブする処理（図４６のステップＳ９２６，ステップＳ９２７）を行っている。以下、これをベース値算定後の算定管理バッファ設定処理として説明する。

【０３７９】

図５０は、ベース値算定後の算定管理バッファ設定処理のプログラムを示す図である。同様に説明する。

〔算定管理バッファ更新処理１〕

「Ｅ＿ＳＨＹＭＴ＿５７：」は、算定管理バッファ更新処理１の開始を示すラベルである。

「ＬＤＱ　（ＬＯＷ　Ｒ＿ＥＲＷ＿ＴＳＫ），＠ＥＸＤ＿ＴＳＫ＿ＹＫＨ」は、算定管理バッファに次回役比算定指定値をセーブする命令である。

「ＪＰ　Ｅ＿ＳＨＹＭＴ＿９０」は、「Ｅ＿ＳＨＹＭＴ＿９０」へジャンプする命令である。

【０３８０】

〔算定管理バッファに基づく処理の振り分け〕

本実施形態の性能表示モニタ制御処理においてベース値の算定を行わなかった場合、算定管理バッファ（Ｒ＿ＥＲＷ＿ＴＳＫ）を確認し、処理の振り分けを行っている（図４６のステップＳ９２２，ステップＳＳ９２３，ステップＳ９２４，ステップＳ９２５，ステップＳ９２６等）。具体的には、役物比率の算定を行う場合は、「Ｅ＿ＳＨＹＭＴ＿７０」へ分岐し、連続役物比率の算定を行う場合は、「Ｅ＿ＳＨＹＭＴ＿８０」へ分岐する。いずれも行わない場合は、「Ｅ＿ＳＨＹＭＴ＿９０」へ分岐する。以下、この処理を算定管理バッファに基づく処理の振り分けとして説明する。

【０３８１】

図５１は、算定管理バッファに基づく処理の振り分けのプログラムを示す図である。同様に説明する。

「Ｅ＿ＳＨＹＭＴ＿６０：」は、処理の開始を示すラベルである。

〔ベースを算定しない場合は役物比率算定、連続役物比率算定を行うかを確認する〕

〔１〕「ＪＴＱ　ＮＺ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＰＡＹ），Ｅ＿ＳＨＹＭＴ＿７０」は、今回賞球数が０以外であれば分岐する命令である。ここでは、今回賞球数（Ｒ＿ＥＲＷ＿ＰＡＹ）が０でない場合にＥ＿ＳＨＹＭＴ＿７０へ分岐する。

〔２〕「ＬＤＱ　Ａ，（ＬＯＷ　Ｒ＿ＥＲＷ＿ＴＳＫ）」は、算定管理バッファをロードする命令である。

〔２〕「ＪＣＰ　Ｚ，Ａ，＠ＥＸＤ＿ＴＳＫ＿ＹＫＨ，Ｅ＿ＳＨＹＭＴ＿７０」は、次回役物比率算定指定値と比較し、ゼロフラグが１であれば分岐する命令である。

上の２つの〔２〕では、算定管理バッファが次回役比算定指定値（＠ＥＸＤ＿ＴＳＫ＿ＹＫＨ（１））である場合に「Ｅ＿ＳＨＹＭＴ＿７０」へ分岐する。

〔３〕「ＣＰ　Ａ，＠ＥＸＤ＿ＴＳＫ＿ＲＹＨ」は、次回連続役物比率算定指定値と比較する命令である。ここでは、算定管理バッファと次回連役算定指定値（＠ＥＸＤ＿ＴＳＫ＿ＲＹＨ（２））を比較する。

〔４〕「ＪＰ　Ｚ，Ｅ＿ＳＨＹＭＴ＿８０」は、ゼロフラグが１であれば分岐する命令である。ここでは、上記〔３〕で一致した場合は「Ｅ＿ＳＨＹＭＴ＿８０」へ分岐する。

「ＪＰ　Ｅ＿ＳＨＹＭＴ＿９０」は、「Ｅ＿ＳＨＹＭＴ＿９０」へ分岐する命令である。上記〔３〕で一致しなかった場合、ここで「Ｅ＿ＳＨＹＭＴ＿９０」へ分岐する。

【０３８２】

〔役物比率算定後の算定管理バッファ設定処理〕
図５２は、役物比率算定後の算定管理バッファ設定処理のプログラムを示す図である。この処理では、役物比率の算定を行った場合、算定管理バッファに次回連続役物比率算定指定値をセーブする。
「Ｅ＿ＳＨＹＭＴ＿７２：」は、処理の開始を示すラベルである。
「ＬＤＱ （ＬＯＷ Ｒ＿ＥＲＷ＿ＹＡＫ），Ａ」は、役物比率バッファにセーブする命令である。
「ＬＤＱ （ＬＯＷ Ｒ＿ＥＲＷ＿ＴＳＫ），＠ＥＸＤ＿ＴＳＫ＿ＲＹＨ」は、算定管理バッファに次回連役算定指定値をセーブする命令である。
「ＪＰ Ｅ＿ＳＨＹＭＴ＿９０」は、「Ｅ＿ＳＨＹＭＴ＿９０」にジャンプする命令である。

【０３８３】
〔連続役物比率算定後の算定管理バッファ設定処理〕
図５３は、連続役物比率算定後の算定管理バッファ設定処理のプログラムを示す図である。この処理では、役物比率の算定を行った場合、算定管理バッファをクリアする。
「Ｅ＿ＳＨＹＭＴ＿８２：」は、処理の開始を示すラベルである。
「ＬＤＱ （ＬＯＷ Ｒ＿ＥＲＷ＿ＲＥＮ），Ａ」は、連続役物比率バッファにセーブする命令である。
「ＣＬＲＱ （ＬＯＷ Ｒ＿ＥＲＷ＿ＴＳＫ）」は、算定管理バッファをクリアする命令である。

【０３８４】
〔第５の技術的特徴のまとめ〕
以上のように、タイマ割込み処理中における性能値（ベース値、役物比率及び連続役物比率）の算定時間が長期化することを抑え、かつ、算定の遅れも最小限にすることができる。

【０３８５】
〔ステート数による検証〕
以上で本実施形態の有用性は明らかとなっているが、以下にステート数による検証を追加する。

【０３８６】
〔ベース値の算定に必要なステート数〕
図５４及び図５５は、ベース値の算定部分のプログラムを示す図である。なお、プログラムコード中、命令の左の数値は、当該命令の実行に必要なステート数（クロック数）を示している。また、数値が２種類記載されている場合、左側が分岐を行う場合のステート数で、右側が分岐を行わない場合のステート数である。

【０３８７】
「Ｅ＿ＳＨＹＭＴ＿５０：」は、処理の開始を示すラベルである。
〔除数設定レジスタ更新処理〕
１ 「ＸＯＲ Ａ，Ａ」は、Ａレジスタに［００Ｈ］をセットする命令である。
５ 「ＬＤＱ ＢＣ，（ＬＯＷ Ｒ＿ＥＲＷ＿ＴＪＳ）」は、通常賞球カウンタをロードする命令である。
４ 「ＬＤＱ ＨＬ，（ＬＯＷ Ｒ＿ＥＲＷ＿ＴＪＯ）」は、通常アウトカウンタをロードする命令である。
３／２ 「ＪＲ ＴＺ，Ｅ＿ＳＨＹＭＴ＿５４」は、第２ゼロフラグが１であれば「Ｅ＿ＳＨＹＭＴ＿５」に分岐する命令である。
５ 「ＯＵＴ （ＬＯＷ ＠ＤＩＶＢ３２＿＋０），ＨＬ」は、除数設定レジスタの最下位アドレス＋０に出力する命令である。
２ 「ＣＬＲ ＨＬ」は、ＨＬレジスタに［０００００Ｈ］をセットする命令である。
５ 「ＯＵＴ （ＬＯＷ ＠ＤＩＶＢ３２＿＋２），ＨＬ」は：除数設定レジスタの最下位アドレス＋２に出力する命令である。

【０３８８】
〔被除数設定レジスタ更新処理〕
以下は、通常賞球数の２００倍を算定する命令である。
１「ＬＤ　Ａ，Ｃ」
４「ＭＵＬ　ＨＬ，Ａ，２００」
１「ＥＸ　ＤＥ，ＨＬ」
１「ＬＤ　Ａ，Ｂ」
４「ＭＵＬ　ＨＬ，Ａ，２００」
２「ＡＤＤＷＢ　ＨＬ，Ｄ」
１「ＬＤ　Ａ，Ｅ」
【０３８９】
３「ＯＵＴ　（ＬＯＷ　＠ＤＩＶＡ３２＿＋０），Ａ」は、被除数設定レジスタの最下位アドレス＋０に出力する命令である。
５「ＯＵＴ　（ＬＯＷ　＠ＤＩＶＡ３２＿＋１），ＨＬ」は、被除数設定レジスタの最下位アドレス＋１に出力する命令である。
４「ＯＵＴ　（ＬＯＷ　＠ＤＩＶＡ３２＿＋３），０」は、被除数設定レジスタの最下位アドレス＋３に０を出力する命令である。
【０３９０】
〔ベース値除算待機処理〕
２「ＬＤ　Ｂ，２」は、Ｂレジスタに２をセットする命令である。
「Ｅ＿ＳＨＹＭＴ＿５１：」は、処理の開始を示すラベルである。
３／２「ＤＪＮＺ　Ｅ＿ＳＨＹＭＴ＿５１」は、ループ継続であれば「Ｅ＿ＳＨＹＭＴ＿５１：」に分岐する命令である。
〔除算結果確認処理〕
２「ＬＤ　Ａ，１００」は、Ａレジスタに１００をセットする命令である。
５「ＩＮ　ＨＬ，（ＬＯＷ　＠ＤＩＶ３２＿＿＋２）」は、除算結果レジスタの最下位アドレス＋２を入力する命令である。
３／２「ＪＲ　ＮＴＺ，Ｅ＿ＳＨＹＭＴ＿５４」は、第２ゼロフラグが０であれば「Ｅ＿ＳＨＹＭＴ＿５４」に分岐する命令である。
５「ＩＮ　ＨＬ，（ＬＯＷ　＠ＤＩＶ３２＿＿）」は、除算結果レジスタの最下位アドレスを入力する命令である。
４「ＣＰ　ＨＬ，１００＊２」は、除算結果の１，２バイト目を確認する命令である。
３／２「ＪＲ　ＮＣ，Ｅ＿ＳＨＹＭＴ＿５４」は、キャリーフラグが０であれば「Ｅ＿ＳＨＹＭＴ＿５４」に分岐する命令である。
１「ＬＤ　Ａ，Ｌ」は、除算結果をセットする命令である。
２「ＳＲＬ　Ａ」は、１ビット右シフトして値を算出する命令である。
２「ＡＤＣ　Ａ，０」は、キャリーフラグを加算してベース算定値を算定する命令である。
【０３９１】
「Ｅ＿ＳＨＹＭＴ＿５４：」は、処理の開始を示すラベルである。
〔ベース算定値更新処理〕
２「ＬＤＱ　ＨＬ，ＬＯＷ　Ｒ＿ＥＲＷ＿ＢＡＳ」は、ベース算定値のアドレスをセットする命令である。
２「ＬＤ　（ＨＬ），Ａ」は、対象アドレスの内容にセーブする命令である。
【０３９２】
以上より、除数設定レジスタ更新処理では、１＋５＋４＋２＋５＋２＋５＝２４ステートを必要とする（通常アウトカウンタが０でなく、「ＪＲ　ＴＺ，Ｅ＿ＳＨＹＭＴ＿５４」で分岐を行わない場合）。
【０３９３】
被除数設定レジスタ更新処理では、１＋４＋１＋１＋４＋２＋１＋３＋５＋４＝２６ステートを必要とする。

【０３９４】
ベース値除算待機処理では、２＋３＋２＝７ステートを必要とする。
【０３９５】
除算結果確認処理では、２＋５＋２＋５＋４＋２＋１＋２＋２＝２５ステートを必要とする（ベース算定結果が０～９９．５％であり、２か所のＪＲ命令では分岐を行わないこととする）。
ベース算定値更新処理では、２＋２＝４ステートを必要とする。
よって、ベース値を算定し、結果を保存するまでのステート数の合計は、２４＋２６＋７＋２５＋４＝８６ステートである。
【０３９６】
〔役物比率の算定に必要なステート数〕
役物比率の算定はベース値同様に行うが、結果を保存するまでのステート数の合計は８１ステートである。
【０３９７】
〔連続役物比率の算定に必要なステート数〕
連続役物比率の算定はベース値同様に行うが、結果を保存するまでのステート数の合計は８１ステートである。
【０３９８】
〔処理時間による検証〕
図５６は、各種性能値の算定に必要となる時間を示す図である。
主制御ＣＰＵ３１の内部システムクロックが１６ＭＨｚであるとする。この場合、１ステート＝１６００万分の１秒となるため、ベース値等の算定に必要な時間は以下の通りとなる。
ベース値算定は、８６ステートで５．３７５μｓの処理時間である。
役物比率算定は、８１ステートで５．０６２５μｓの処理時間である。
連続役物比率算定は、８１ステートで５．０６２５μｓの処理時間である。
したがって、ベース値、役物比率及び連続役物比率をすべて算定した場合、２４８ステート（１５．５μｓ）が必要となるが、本実施形態のように１回のタイマ割込み処理中で１つしか算定しない場合、最大８６ステート（５．３７５μｓ）となる。
【０３９９】
〔別案〕
次に、算出処理時間を抑えるための別案について説明する。
別案では、ベース値算定フラグ、役物比率算定フラグ及び連続役物比率算定フラグを用意し、算定が必要となった場合にフラグをセット、算定を行った場合にフラグをリセットする。複数のフラグがセットされている場合、最も優先度の高いもの１つのみを算定する。この別案においても、本実施形態と同様の有用性が得られるが、若干プログラムが複雑になる。以下、より詳細に説明する。
【０４００】
〔各種フラグ〕
別案においては、以下のフラグを使用する。
以下のフラグはＲＷＭ初期化時にリセットし、電源投入時（電源復帰時）は現状維持とする。
・　ベース値算定フラグ
・　役物比率算定フラグ
・　連続役物比率算定フラグ
これらのフラグはそれぞれが別のＲＷＭでもよいし、同一ＲＷＭ内の別ｂｉｔでも良い。
【０４０１】
〔フラグセットの契機〕
以下の条件を満たした場合にそれぞれのフラグをセットする。なお、すでに当該フラグがセットされていた場合、セット状態を継続する。

「ベース値算定フラグ」
以下の場合にベース値算定フラグをセットする。
・　ベース値の表示内容切り替えによって、現在ベース値の表示を開始する場合
・　総アウト数があらかじめ定められた値（６００００個、初回のみ２９９個）に達した場合
「役物比率算定フラグ」
以下の場合に役物比率算定フラグをセットする。
・　賞球が発生した場合
「連続役物比率算定フラグ」
以下の場合に連続役物比率算定フラグをセットする。
・　賞球が発生した場合
・　未定の賞球を連続役物比率対象の賞球に加算した場合

【０４０２】
〔実行する処理〕
図５７は、別案で算定フラグと実行する処理の対応関係を示す図である。
ベース値算定フラグ等の状態により、タイマ割込み処理内で実行する処理は、図５７に示すように決定する。なお、算定を行った場合、対応するフラグはリセットする。

【０４０３】
本発明は上述した一実施形態に制約されることなく、種々に変形して実施することができる。例えば、上述した実施形態は、以下の変形が可能である。
（１）管理遊技機１００の場合で説明したが、管理遊技機１００の構成を有しない既存のパチンコ機にも適用可能である。
（２）上述したプログラム及びフローチャートはあくまで例示であり、本発明はこれらに限定されるものではない。

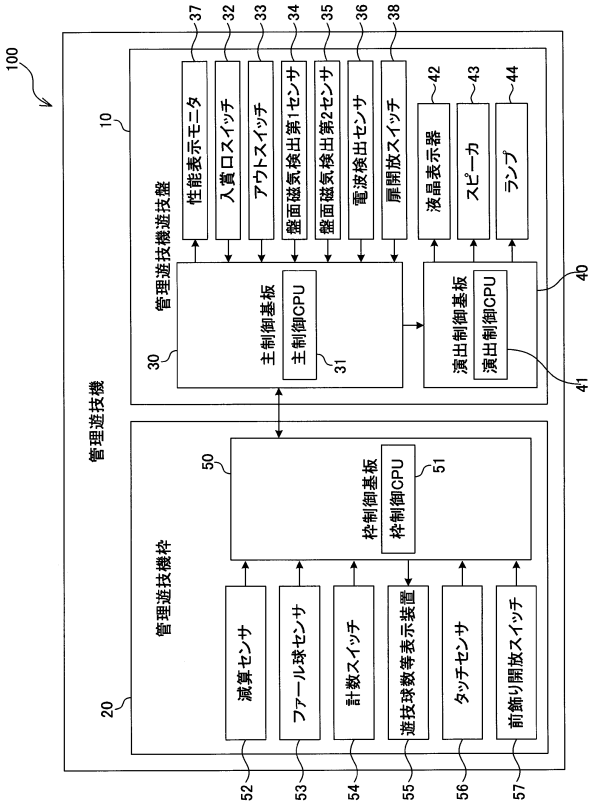
【０４０４】
（３）本発明は、スロットマシンにも適用可能である。すなわち、遊技の実行と遊技の停止を制御する主制御基板と、メダルの払出しを制御するメダル数制御基板とを備えるスロットマシンにおいても、本発明は適用可能である。
この場合、主制御基板が、遊技で消費したメダル枚数情報と、遊技で獲得したメダル枚数情報とをメダル数制御基板に送信し、メダル数制御基板が差枚数を算出する。そして、メダル数制御基板が算出した差枚数情報を主制御基板に送信し、主制御基板は受信した差枚数情報に基づいて遊技停止状態を設定してもよい。
また、主制御基板が、遊技で獲得したメダル枚数情報をメダル数制御基板に送信し、メダル数制御基板が、遊技で獲得したメダル枚数情報と遊技で消費したメダル枚数情報とに基づいて差枚数を算出する。そして、メダル数制御基板が算出した差枚数情報を主制御基板に送信し、主制御基板は受信した差枚数情報に基づいて遊技停止状態を設定してもよい。

【符号の説明】
【０４０５】
１０　管理遊技機遊技盤
２０　管理遊技機枠
３０　主制御基板
３２　入賞口スイッチ
３３　アウトスイッチ
３４　盤面磁気検出第１センサ
３５　盤面磁気検出第２センサ
３６　電波検出センサ
３７　性能表示モニタ
３８　扉開放スイッチ
４０　演出制御基板
５０　枠制御基板

１００　管理遊技機

【図面】

【図１】

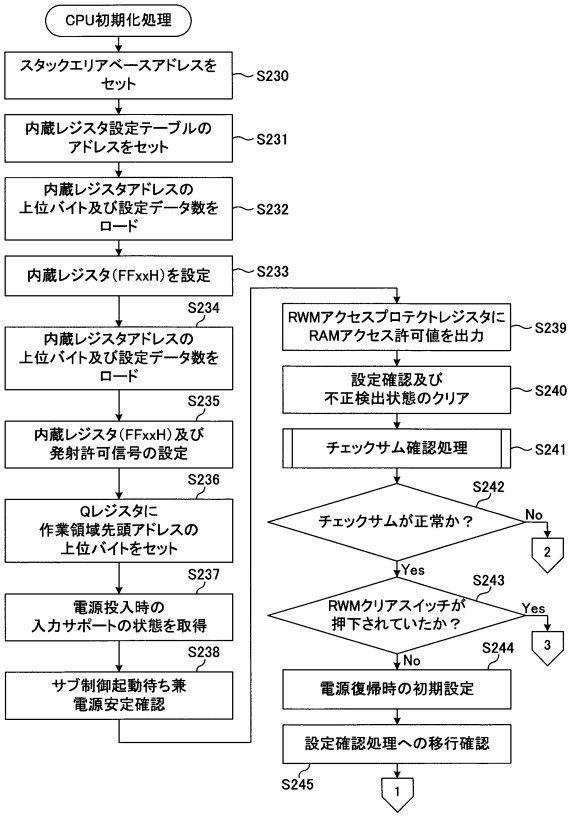


【図２】

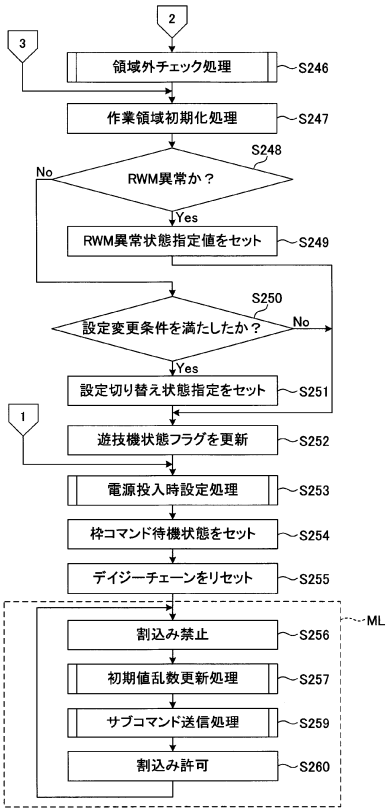
〔メモリマップ〕

アドレス	格納可能情報	メモリ
0000H ～ 0BFFH	使用領域のプログラムコード	ROM
0C00H ～ 0FFFH	未使用	
1000H ～ 1BFFH	使用領域のプログラムデータ	
1C00H ～ 1FFFH	未使用	
2000H ～ 2FFFH	領域外のプログラムコード 領域外のプログラムデータ	
3000H ～ EFFFH	未使用	なし
F000H ～ F1FFH	使用領域のワーク及びスタックRWM	RAM
F200H ～ F3FFH	領域外のワーク及びスタックRWM	
F400H ～ FFFFH	未使用	なし

【図３】



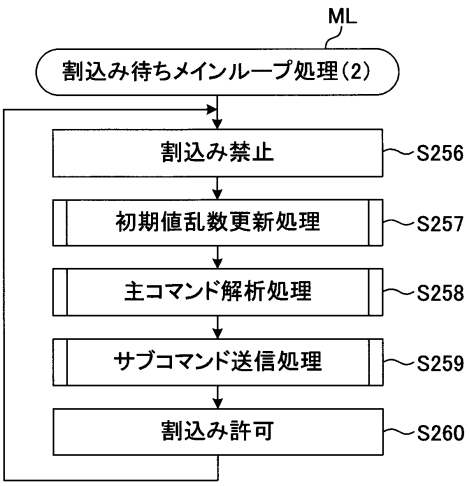
【図４】



【図５】

```
[割込み待ちメインループ処理(1)]
CPUINIT_80:
;!------< 割込み待ちﾒｲﾝﾙｰﾌﾟ処理 >------!
        DI              ;割込み禁止
;!------< 初期値乱数更新処理 >------!
        CALLF   INI_RND ;初期値乱数更新処理
;!------< ｻﾌﾞｺﾏﾝﾄﾞ送信処理 >------!
        CALLF   SBC_OUT ;ｻﾌﾞｺﾏﾝﾄﾞ送信処理
        EI              ;割込み許可
        JR      CPUINIT_80
```

【図６】



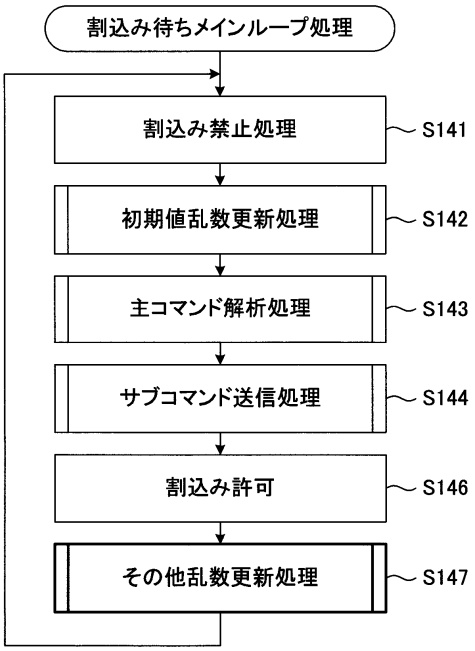
【図７】

〔割込み待ちメインループ処理(2)〕

```
CPUINIT_80:
        DI                  ;割込み禁止
        CALLF  INIRND       ;初期値乱数更新処理
        CALLF  PY_CMDA      ;主ｺﾏﾝﾄﾞ解析処理
        CALLF  SBC_OUT      ;ｻﾌﾞｺﾏﾝﾄﾞ送信処理
        EI                  ;割込み許可
        JR  CPUINIT_80      ;
```

【図８】

〔比較例〕



【図９】

〔比較例の割込み待ちメインループ処理〕

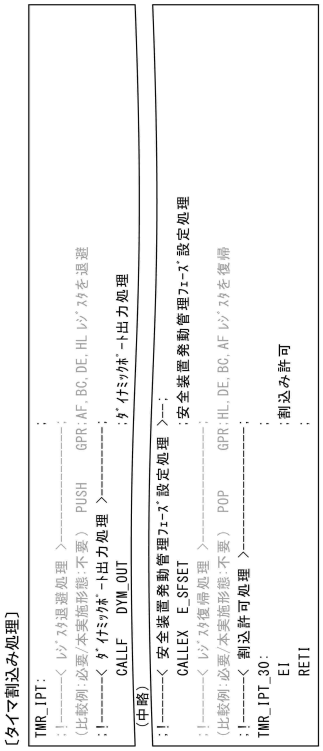
```
CPUINIT_80:
;!------< 割込み待ちﾒｲﾝﾙｰﾌﾟ処理 >-------;
        DI                  ;割込み禁止
;!------< 初期値乱数更新処理 >-----------;
        CALLF   INI_RND     ;初期値乱数更新処理
;!------< 主ｺﾏﾝﾄﾞ解析処理 >-------------;
        CALLF   PY_CMDA     ;主ｺﾏﾝﾄﾞ解析処理
;!------< ｻﾌﾞｺﾏﾝﾄﾞ送信処理 >-----------;
        CALLF   SBC_OUT     ;ｻﾌﾞｺﾏﾝﾄﾞ送信処理
        EI                  ;割込み許可
;!------< その他乱数更新処理 >---------;
        CALLF   ETC_RND     ;その他乱数更新処理
        JR      CPUINIT_80  ;
```

【図１０】

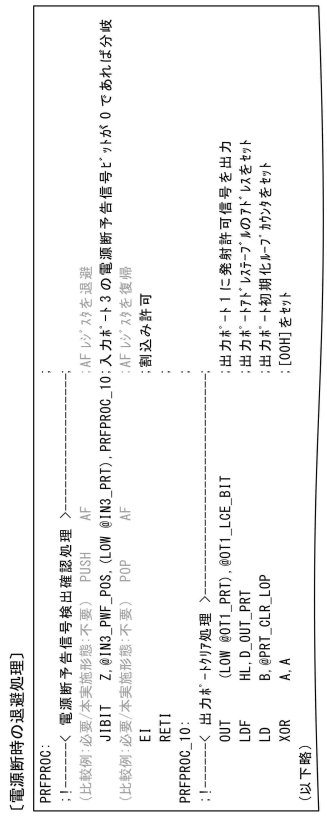
〔比較例のその他乱数更新処理〕

```
ETCRND:
;!------< 特別図柄ﾘｰﾁﾓｰﾄﾞ決定乱数更新処理 >---;
        LD      A,R             ;Rﾚｼﾞｽﾀ値をｾｯﾄ
        LDQ     HL,(LOW R_RND_TZ__MOD)  ;特別図柄ﾘｰﾁﾓｰﾄﾞ決定乱数をﾛｰﾄﾞ
        ADDWB   HL,A            ;特別図柄ﾘｰﾁﾓｰﾄﾞ決定乱数を更新
        INC     HL              ;
        LD      BC,&RND_MOD_MAX+1       ;特別図柄ﾘｰﾁﾓｰﾄﾞ決定乱数上限値をｾｯﾄ
[1]     DIV     HL,BC           ;更新結果を特別図柄ﾘｰﾁﾓｰﾄﾞ決定乱数上限値で除算
[2]     LDQ     (LOW R_RND_TZ__MOD),BC  ;特別図柄ﾘｰﾁﾓｰﾄﾞ決定乱数に剰余をｾｰﾌﾞ
;!------< 特別図柄変動ﾊﾟﾀｰﾝ乱数更新処理 >---;
        LD      DE,&RND_PAT_MAX*256+LOW R_RND_TZ__PAT   ;特別図柄変動ﾊﾟﾀｰﾝ乱数上限値及び特別図柄変動ﾊﾟﾀｰﾝ乱数のｱﾄﾞﾚｽの下位ﾊﾞｲﾄをｾｯﾄ
        RST     COUNTUP         ;ｶｳﾝﾄｱｯﾌﾟ処理
        RET                     ;
```

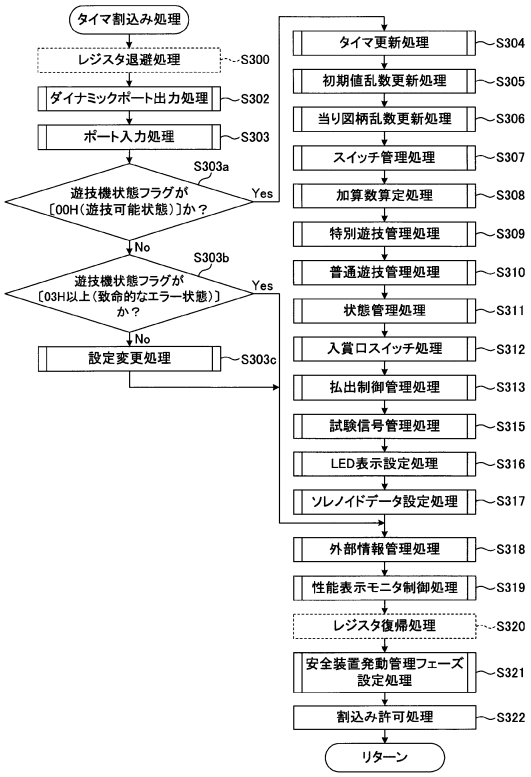
【図１１】



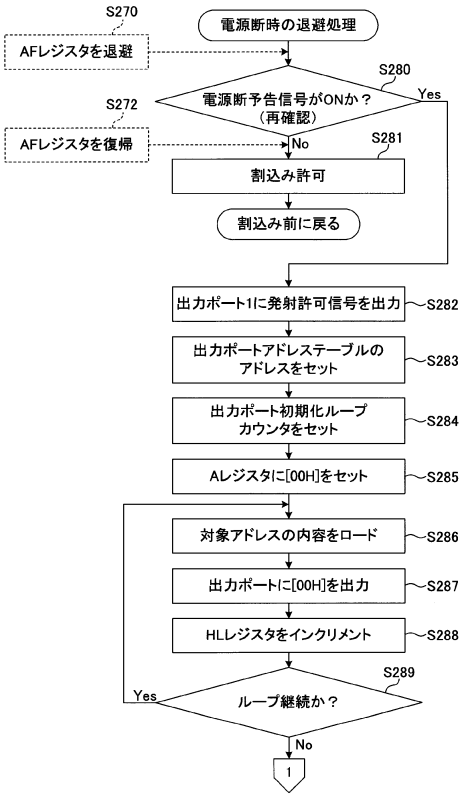
【図１２】



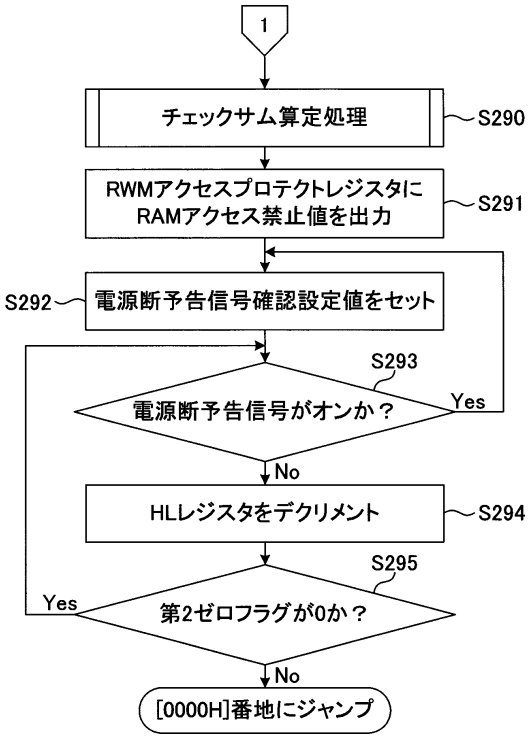
【図１３】



【図１４】



【図１５】



【図１６】

〔比較例のチェックサム算定処理〕

```
E_CLSUM_01:
 (中略)
 ;!-------< 使用領域ﾁｪｯｸｻﾑ算定処理 >----------;
 (1)     XOR     A,A             ;[00H]をｾｯﾄ
 (3)     LD      HL,@RM_RAMADR   ;作業領域先頭ｱﾄﾞﾚｽをｾｯﾄ
 (3)     LD      BC,@CHK_ARE     ;ﾁｪｯｸｻﾑ算定領域数をｾｯﾄ
 E_CKSUM_10:
 (2)     SUB     A,(HL)          ;RWMﾁｪｯｸｻﾑを算定
 (1)     INC     HL              ;次の RWMｱﾄﾞﾚｽをｾｯﾄ
 (2)     DEC     BC              ;ﾁｪｯｸｻﾑ算定領域数を 1 減算
 (3/2)   JR      NTZ,E_CLSUM_10  ;ﾙｰﾌﾟ継続であれば分岐
 ;!-------< 領域外ﾁｪｯｸｻﾑ算定処理 >----------;
 (2)     LDQ     HL,LOW @RM_EXTADR ;領域外作業領域先頭ｱﾄﾞﾚｽをｾｯﾄ
 (2)     LD      B,@EXD_USE_ERW  ;領域外 RWM使用領域数をｾｯﾄ
 E_CKSUM_20:
 (2)     ADD     A,(HL)          ;RWMﾁｪｯｸｻﾑを算定
 (1)     INC     HL              ;次の RWMｱﾄﾞﾚｽをｾｯﾄ
 (3/2)   DJNZ    E_CKSUM_20      ;ﾙｰﾌﾟ継続であれば分岐
 E_CKSUM_30:
 ;!-------< ﾁｪｯｸｻﾑ結果ﾌﾗｸﾞ設定処理 >----------;
 (3)     LDQ     (LOW R_ERW_BRS),A ;ﾁｪｯｸｻﾑ結果ﾌﾗｸﾞにｾｰﾌﾞ
```

【図１７】

〔比較例のチェックサム確認処理〕

```
E_CKSUM_01:
 (中略)
 ;!-------< 使用領域ﾁｪｯｸｻﾑ算定処理 >----------;
 (1)     XOR     A,A             ;[00H]をｾｯﾄ
 (3)     LD      HL,@RM_RAMADR   ;作業領域先頭ｱﾄﾞﾚｽをｾｯﾄ
 (3)     LD      BC,@CHK_ARE     ;ﾁｪｯｸｻﾑ算定領域数をｾｯﾄ
 E_CKSUM_10:
 (2)     ADD     A,(HL)          ;RWMﾁｪｯｸｻﾑを算定
 (1)     INC     HL              ;次の RWMｱﾄﾞﾚｽをｾｯﾄ
 (2)     DEC     BC              ;ﾁｪｯｸｻﾑ算定領域数を 1 減算
 (3/2)   JR      NTZ,E_CKSUM_10  ;ﾙｰﾌﾟ継続であれば分岐
 ;!-------< 領域外ﾁｪｯｸｻﾑ算定処理 >----------;
 (2)     LDQ     HL,LOW @RM_EXTADR ;領域外作業領域先頭ｱﾄﾞﾚｽをｾｯﾄ
 (2)     LD      B,@EXD_USE_ERW  ;領域外 RWM使用領域数をｾｯﾄ
 E_CKSUM_20:
 (2)     ADD     A,(HL)          ;RWMﾁｪｯｸｻﾑを算定
 (1)     INC     HL              ;次の RWMｱﾄﾞﾚｽをｾｯﾄ
 (3/2)   DJNZ    E_CKSUM_20      ;ﾙｰﾌﾟ継続であれば分岐
 E_CKSUM_30:
 ;!-------< ﾁｪｯｸｻﾑ結果ﾌﾗｸﾞ設定処理 >----------;
 (3)     LDQ     (LOW R_ERW_BRS),A ;ﾁｪｯｸｻﾑ結果ﾌﾗｸﾞにｾｰﾌﾞ
```

【図１８】

〔本実施形態のチェックサム算定処理〕

```
E_CLSUM_01:
 (中略)
 ;!-------< 使用領域ﾁｪｯｸｻﾑ算定処理 >----------;
 (1)     XOR     A,A             ;[00H]をｾｯﾄ
 (3)     LD      HL,@RM_RAMADR   ;作業領域先頭ｱﾄﾞﾚｽをｾｯﾄ
 (3)     LD      BC,@CHK_ARE     ;ﾁｪｯｸｻﾑ算定領域数をｾｯﾄ
 (3)     ADDIR                   ;ﾌﾞﾛｯｸｶｰﾁ
 ;!-------< 領域外ﾁｪｯｸｻﾑ算定処理 >----------;
 (2)     LDQ     HL,LOW @RM_EXTADR ;領域外作業領域先頭ｱﾄﾞﾚｽをｾｯﾄ
 (2)     LD      B,@EXD_USE_ERW  ;領域外 RWM使用領域数をｾｯﾄ
 (3)     ADDIRS                  ;ﾌﾞﾛｯｸｶｰﾁ
 ;!-------< ｻﾑﾁｪｯｸﾊﾞｯﾌｧ設定処理 >----------;
 (2)     NEG                     ;2 の補数をｾｯﾄ
 (3)     LDQ     (LOW R_ERW_SUM),A ;ｻﾑﾁｪｯｸﾊﾞｯﾌｧにｾｰﾌﾞ
```

【図１９】

〔本実施形態のチェックサム確認処理〕

```
E_CLSUM_01:
 (中略)
 ;!------< 使用領域ﾁｪｯｸｻﾑ算定処理 >------;
 (1)     XOR     A,A             ;[00H]をｾｯﾄ
 (3)     LD      HL,@RM_RAMADR   ;作業領域先頭ｱﾄﾞﾚｽをｾｯﾄ
 (3)     LD      BC,@CHK_ARE     ;ﾁｪｯｸｻﾑ算定領域数をｾｯﾄ
 (3)     ADDIR                   ;ﾌﾞﾛｯｸｻｰﾁ
 ;!------< 領域外ﾁｪｯｸｻﾑ算定処理 >------;
 (2)     LDQ     HL,LOW @RM_EXTADR ;領域外作業領域先頭ｱﾄﾞﾚｽをｾｯﾄ
 (2)     LD      B,@EXD_USE_ERW  ;領域外 RWW 使用領域数をｾｯﾄ
 (3)     ADDIRS                  ;ﾌﾞﾛｯｸｻｰﾁ
 ;!------< ﾁｪｯｸｻﾑ結果ﾌﾗｸﾞ設定処理 >------;
 (3)     LDQ     (LOW R_ERW_BRS),A ;ﾁｪｯｸｻﾑ結果ﾌﾗｸﾞにｾｰﾌﾞ
```

【図２０】

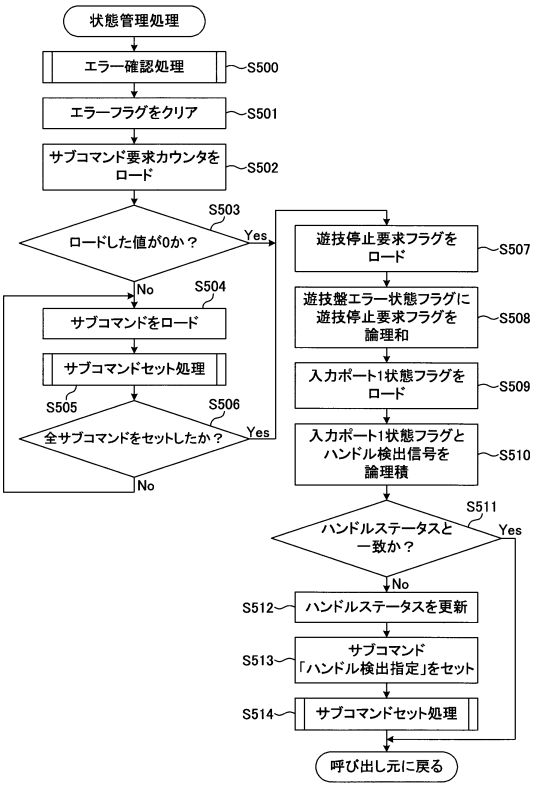
項目	比較例	本実施形態	改善率
チェックサム算定	5112ステート (0.3195ms)	2056ステート (0.1285ms)	約59.78% (約1/2.488)
チェックサム確認	5112ステート (0.3195ms)	2054ステート (0.128375ms)	約59.82% (約1/2.485)

【図２１】

〔使用領域側の状態管理処理〕

```
STC_PRC:
 ;!------< ｴﾗｰ確認処理 >------;
 2       CALLEX  E_CKERR         ;ｴﾗｰ確認処理
 ;!------< ｴﾗｰﾌﾗｸﾞｸﾘｱ処理 >------;
 2       CLRQ    (LOW R_ERR_FLG) ;ｴﾗｰﾌﾗｸﾞをｸﾘｱ
 ;!------< ｻﾌﾞｺﾏﾝﾄﾞ要求ｶｳﾝﾀ確認処理 >------;
 3       LD      HL,R_ERW_SBC    ;ｻﾌﾞｺﾏﾝﾄﾞ要求ｶｳﾝﾀのｱﾄﾞﾚｽをｾｯﾄ
 1       LD      B,(HL)          ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
 2       JR      TZ,STC_PRC_20   ;第 2 ﾋﾞｯﾄﾛｸﾌﾗｸﾞが 1 であれば分岐
STC_PRC_10:                     ;
 ;!------< ｻﾌﾞｺﾏﾝﾄﾞｾｯﾄ処理 >------;
 2       INCD    DE,(HL)         ;ｻﾌﾞｺﾏﾝﾄﾞをﾛｰﾄﾞ
 1       RST     SBCMDST         ;ｻﾌﾞｺﾏﾝﾄﾞｾｯﾄ処理
 ;!------< ﾙｰﾌﾟ回数判定処理 >------;
 2       DJNZ    STC_PRC_10      ;ﾙｰﾌﾟ継続であれば分岐
STC_PRC_20:                     ;
 ;!------< 遊技盤ｴﾗｰ状態ﾌﾗｸﾞ更新処理 >------;
 3       LD      A,(R_ERW_YER)   ;遊技停止要求ﾌﾗｸﾞをﾛｰﾄﾞ
 3       ORQ     (LOW R_YUG_ERR),A ;遊技盤ｴﾗｰ状態ﾌﾗｸﾞと論理和
 ;!------< ﾊﾝﾄﾞﾙ検出信号確認処理 >------;
 2       LDQ     A,(LOW R_IN1_PRT) ;入力ﾎﾟｰﾄ 1 状態ﾌﾗｸﾞをﾛｰﾄﾞ
 2       AND     A,@IN1_HDL_BIT  ;ﾊﾝﾄﾞﾙ検出信号と論理積
 3       ROPQ    Z,A,(LOW R_HDL_STS) ;ﾊﾝﾄﾞﾙｽﾃｰﾀｽと比較しﾋﾞｯﾄﾌﾗｸﾞが 1 であればﾘﾀｰﾝ
 ;!------< ﾊﾝﾄﾞﾙ検出指定ｺﾏﾝﾄﾞｾｯﾄ処理 >------;
 2       LDQ     (LOW R_HDL_STS),A ;ﾊﾝﾄﾞﾙｽﾃｰﾀｽにｾｰﾌﾞ
 3       LD      DE,@CMD_STS_HDL ;ﾊﾝﾄﾞﾙ検出指定をｾｯﾄ
 1       RST     SBCMDST         ;ｻﾌﾞｺﾏﾝﾄﾞｾｯﾄ処理
 1       RET                     ;
```

【図２２】



【図２３】

〔領域外のエラー確認処理(E_CKERR)(1/5)〕

```
E_CKERR:                                ;
        JP      E_CKERR_01              ;
```

（中略）

```
E_CKERR_01:                             ;
        LD      Q,HIGH @RM_EXTADR       ;領域外作業領域先頭ｱﾄﾞﾚｽの上位ﾊﾞｲﾄをｾｯﾄ
;!------< ｻﾌﾞｺﾏﾝﾄﾞ要求ｶｳﾝﾀｸﾘｱ処理 >------;
        CLRQ    (LOW R_ERW_SBC) ;ｻﾌﾞｺﾏﾝﾄﾞ要求ｶｳﾝﾀをｸﾘｱ
;!------< ﾙｰﾌﾟ設定処理 1 >----------------;
        LD      HL,D_ETB_ER1            ;ｴﾗｰ管理処理ﾃｰﾌﾞﾙ 1 のｱﾄﾞﾚｽをｾｯﾄ
        LD      B,@D_ETB_ER1_NUM        ;状態項目数をｾｯﾄ
E_CKERR_10:                             ;
;!------< 監視対象信号ﾋﾞｯﾄﾃﾞｰﾀ判定処理 >--;
        LDIN    DE,(HL)                 ;入力ﾎﾟｰﾄ状態ﾌﾗｸﾞのｱﾄﾞﾚｽをﾛｰﾄﾞ
        LD      A,(DE)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        LD      C,0                     ;C ﾚｼﾞｽﾀに 0 をｾｯﾄ
        AND     A,(HL)                  ;監視対象ﾋﾞｯﾄﾃﾞｰﾀと論理積
        JR      Z,E_CKERR_20            ;ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        INC     C                       ;C ﾚｼﾞｽﾀをｲﾝｸﾘﾒﾝﾄ
E_CKERR_20:                             ;
;!------< 監視対象ｽﾃｰﾀｽ今回値更新処理 >--;
        INLD    DE,(HL)                 ;ｽﾃｰﾀｽ前回値のｱﾄﾞﾚｽをﾛｰﾄﾞ
        LD      A,(DE)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        CP      A,C                     ;A ﾚｼﾞｽﾀと C ﾚｼﾞｽﾀを比較
        LD      (DE),C                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
        INC     DE                      ;DE ﾚｼﾞｽﾀをｲﾝｸﾘﾒﾝﾄ
        INC     HL                      ;HL ﾚｼﾞｽﾀをｲﾝｸﾘﾒﾝﾄ
```

【図２４】

〔領域外のエラー確認処理(E_CKERR)(2/5)〕

```
; !------<監視対象ｽﾃｰﾀｽ今回値判定処理>--;
        JR      Z,E_CKERR_30            ;ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        LD      A,(HL)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        LD      (DE),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
E_CKERR_30:                             ;
;!------< 監視ﾀｲﾏ値判定処理 1 >----------;
        INC     HL                      ;HL ﾚｼﾞｽﾀをｲﾝｸﾘﾒﾝﾄ
        LD      A,(DE)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        JT      Z,A,E_CKERR_40          ;ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        DEC     A                       ;ﾀｲﾏ値を 1 減算
        LD      (DE),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
;!------< 監視ﾀｲﾏ値判定処理 2 >----------;
        JT      NZ,A,E_CKERR_40         ;ｾﾞﾛﾌﾗｸﾞが 0 であれば分岐
;!------< ｴﾗｰﾌﾗｸﾞ判定処理 1 >-----------;
        INC     DE                      ;DE ﾚｼﾞｽﾀをｲﾝｸﾘﾒﾝﾄ
        LD      A,(DE)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        CP      A,C                     ;A ﾚｼﾞｽﾀと C ﾚｼﾞｽﾀを比較
        JR      Z,E_CKERR_40            ;ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        LD      A,C                     ;ｽﾃｰﾀｽ今回値をｾｯﾄ
        LD      (DE),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
        EX      DE,HL                   ;DE ﾚｼﾞｽﾀと HL ﾚｼﾞｽﾀを交換
;!------< ｻﾌﾞｺﾏﾝﾄﾞ設定処理 1 >-----------;
        LDQ     HL,LOW R_ERW_SBC        ;ｻﾌﾞｺﾏﾝﾄﾞ要求ｶｳﾝﾀのｱﾄﾞﾚｽをｾｯﾄ
        INC     (HL)                    ;ｻﾌﾞｺﾏﾝﾄﾞ要求ｶｳﾝﾀを 1 加算
        LD      A,(HL)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        ADDWB   HL,A                    ;HL ﾚｼﾞｽﾀに A ﾚｼﾞｽﾀを加算
        ADDWB   HL,A                    ;HL ﾚｼﾞｽﾀに A ﾚｼﾞｽﾀを加算
        LD      (HL),@CMD_STS_HED       ;対象ｱﾄﾞﾚｽの内容に状態指定先行ｺﾏﾝﾄﾞをｾｰﾌﾞ
        DEC     HL                      ;HL ﾚｼﾞｽﾀをﾃﾞｸﾘﾒﾝﾄ
        LD      A,(DE)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        LD      (HL),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
        EX      DE,HL                   ;DE ﾚｼﾞｽﾀと HL ﾚｼﾞｽﾀを交換
;!------< ｴﾗｰﾌﾗｸﾞ判定処理 2 >-----------;
        LD      A,(R_ERR_FLG)           ;ｴﾗｰﾌﾗｸﾞをﾛｰﾄﾞ
        AND     A,(HL)                  ;ｴﾗｰﾌﾗｸﾞと監視対象ﾋﾞｯﾄﾃﾞｰﾀを論理積
        JR      Z,E_CKERR_60            ;ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        LD      A,01H                   ;A ﾚｼﾞｽﾀに[01H]をｾｯﾄ
E_CKERR_60:                             ;
        INC     HL                      ;HL ﾚｼﾞｽﾀをｲﾝｸﾘﾒﾝﾄ
        LDIN    DE,(HL)                 ;ｴﾗｰ発生ﾌﾗｸﾞのｱﾄﾞﾚｽをﾛｰﾄﾞ
```

【図２５】

〔領域外のエラー確認処理(E_CKERR)(3/5)〕

```
;!------< ｽﾃｰﾀｽ今回値判定処理 1 >--------;
        ADD     A,A                     ;A ﾚｼﾞｽﾀに値を加算
        ADD     A,(DE)                  ;ｽﾃｰﾀｽ今回値にｻﾌﾞｺﾏﾝﾄﾞを加算
        LD      (HL),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
;!------< RWM書換え処理 >----------------;
        EX      DE,HL                   ;DE ﾚｼﾞｽﾀと HL ﾚｼﾞｽﾀを交換
        JT      Z,C,E_CKERR_40          ;ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        LD      E,(HL+1)                ;ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ下位ﾊﾞｲﾄをﾛｰﾄﾞ
        JR      TZ,E_CKERR_40           ;第 2 ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        LD      A,(HL+2)                ;ｴﾗｰ発生時設定ﾃﾞｰﾀをﾛｰﾄﾞ
        LD      (DE),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
E_CKERR_40:                             ;
        ADDWB   HL,003H                 ;HL ﾚｼﾞｽﾀに[003H]を加算
;!------< ﾙｰﾌﾟ終了判定 1 >----------------;
        DJNZ    E_CKERR_10              ;ﾙｰﾌﾟ継続であれば分岐
;!------< ﾙｰﾌﾟ設定処理 2 >----------------;
        LD      HL,D_ETB_ER2            ;ｴﾗｰ管理処理ﾃｰﾌﾞﾙ 2 のｱﾄﾞﾚｽをｾｯﾄ
        LD      B,@D_ETB_ER2_NUM;状態項目数をｾｯﾄ
E_CKERR_50:                             ;
;!------< ｴﾗｰﾌﾗｸﾞ判定処理 2 >------------;
        LD      A,(R_ERR_FLG)           ;ｴﾗｰﾌﾗｸﾞをﾛｰﾄﾞ
        AND     A,(HL)                  ;ｴﾗｰﾌﾗｸﾞと監視対象ﾋﾞｯﾄﾃﾞｰﾀを論理積
        JR      Z,E_CKERR_60            ;ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        LD      A,01H                   ;A ﾚｼﾞｽﾀに[01H]をｾｯﾄ
E_CKERR_60:                             ;
        INC     HL                      ;HL ﾚｼﾞｽﾀをｲﾝｸﾘﾒﾝﾄ
        LDIN    DE,(HL)                 ;ｴﾗｰ発生ﾌﾗｸﾞのｱﾄﾞﾚｽをﾛｰﾄﾞ
```

【図２６】

〔領域外のエラー確認処理(E_CKERR)(4/5)〕

```
;!------< ｴﾗｰ発生ﾌﾗｸﾞ更新処理 >----------;
        LD      (DE),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
;!------< ｽﾃｰﾀｽ今回値判定処理 >----------;
        JT      Z,A,E_CKERR_70          ;ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        EX      DE,HL                   ;DE ﾚｼﾞｽﾀと HL ﾚｼﾞｽﾀを交換
;!------< ｻﾌﾞｺﾏﾝﾄﾞ設定処理 2 >------------;
        LDQ     HL,LOW R_ERW_SBC;ｻﾌﾞｺﾏﾝﾄﾞ要求ｶｳﾝﾀのｱﾄﾞﾚｽをｾｯﾄ
        INC     (HL)                    ;ｻﾌﾞｺﾏﾝﾄﾞ要求ｶｳﾝﾀを 1 加算
        LD      A,(HL)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        ADDWB   HL,A                    ;HL ﾚｼﾞｽﾀに A ﾚｼﾞｽﾀを加算
        ADDWB   HL,A                    ;HL ﾚｼﾞｽﾀに A ﾚｼﾞｽﾀを加算
        LD      (HL),@CMD_STS_HED       ;対象ｱﾄﾞﾚｽの内容に状態指定先行ｺﾏﾝﾄﾞをｾｰﾌﾞ
        DEC     HL                      ;HL ﾚｼﾞｽﾀをﾃﾞｸﾘﾒﾝﾄ
        LD      A,(DE)                  ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
        LD      (HL),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
        EX      DE,HL                   ;DE ﾚｼﾞｽﾀと HL ﾚｼﾞｽﾀを交換
;!------< ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ確認処理 >--;
        LD      E,(HL+1)                ;ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ下位ﾊﾞｲﾄをﾛｰﾄﾞ
        JR      TZ,E_CKERR_70           ;第 2 ｾﾞﾛﾌﾗｸﾞが 1 であれば分岐
        LD      A,(HL+2)                ;ｴﾗｰ発生時設定ﾃﾞｰﾀをﾛｰﾄﾞ
        LD      (DE),A                  ;対象ｱﾄﾞﾚｽの内容にｾｰﾌﾞ
E_CKERR_70:                             ;
        ADDWB   HL,003H                 ;HL ﾚｼﾞｽﾀに[003H]を加算
;!------< ﾙｰﾌﾟ終了判定 2 >----------------;
        DJNZ    E_CKERR_50              ;ﾙｰﾌﾟ継続であれば分岐
```

10 20 30 40 50

【図２７】

〔領域外のエラー確認処理(E_CKERR)(5/5)〕

```
;!------< ｴﾗｰﾌﾗｸﾞ判定処理３ >------------;
        LD      A,(R_ERR_FLG)           ;ｴﾗｰﾌﾗｸﾞをﾛｰﾄﾞ
        ORQ     A,(LOW R_ERW_SEF)       ;ｽｲｯﾁｴﾗｰﾌﾗｸﾞと論理和
        ORQ     A,(LOW R_ERW_M1F)       ;磁気ｴﾗｰ１ﾌﾗｸﾞと論理和
        ORQ     A,(LOW R_ERW_M2F)       ;磁気ｴﾗｰ２ﾌﾗｸﾞと論理和
        ORQ     A,(LOW R_ERW_D1F)       ;電波ｴﾗｰ１ﾌﾗｸﾞと論理和
        JT      Z,A,E_CKERR_80          ;ｾﾞﾛﾌﾗｸﾞが１であれば分岐
;!------< 不正検知１情報ﾋﾞｯﾄﾀｲﾏｾｯﾄ処理 >--;
        LDQ     (LOW R_ERW_I1T),@TMR_HLD_SEC;不正検知１情報ﾋﾞｯﾄﾀｲﾏに不正検知情報１保持時間をｾｰﾌﾞ
E_CKERR_80:                             ;
        RETEX                           ;
```

【図２８】

〔エラー管理処理テーブル1(1/2)〕

```
D_ETB_ER1:
        DW      R_IN1_PRT               ;!入力ﾎﾟｰﾄ１状態ﾌﾗｸﾞのｱﾄﾞﾚｽ
        DB      @IN1_SWE_BIT            ;!ｽｲｯﾁ異常信号ﾋﾞｯﾄﾃﾞｰﾀ
        DW      R_ERW_SEO               ;!ｽｲｯﾁｽﾃｰﾀｽ前回値のｱﾄﾞﾚｽ
        DB      @TMR_CHK_SWI            ;!ｽｲｯﾁ異常監視ﾀｲﾏ値
        DB      LOW @CMD_ERR_SWI        ;!ｻﾌﾞｺﾏﾝﾄﾞ(ｽｲｯﾁｴﾗｰ指定)
        DB      0                       ;!ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ
        DB      0                       ;!ｴﾗｰ発生時設定ﾃﾞｰﾀ
                                        ;
        DW      R_IN1_PRT               ;!入力ﾎﾟｰﾄ１状態ﾌﾗｸﾞのｱﾄﾞﾚｽ
        DB      @IN1_DRE_BIT            ;!扉開放ｽｲｯﾁ信号ﾋﾞｯﾄﾃﾞｰﾀ
        DW      R_ERW_DEO               ;!扉開放ｽﾃｰﾀｽ前回値のｱﾄﾞﾚｽ
        DB      @TMR_CHK_DOR            ;!扉監視ﾀｲﾏ値
        DB      LOW @CMD_ERR_BDR        ;!ｻﾌﾞｺﾏﾝﾄﾞ(扉開放ｴﾗｰ指定)
        DB      0                       ;!ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ
        DB      0                       ;!ｴﾗｰ発生時設定ﾃﾞｰﾀ
                                        ;!
        DW      R_IN1_PRT               ;!入力ﾎﾟｰﾄ１状態ﾌﾗｸﾞのｱﾄﾞﾚｽ
        DB      @IN1_M1E_BIT            ;!盤面磁気検出ｾﾝｻ１信号ﾋﾞｯﾄﾃﾞｰﾀ
        DW      R_ERW_M1O               ;!盤面磁気検出ｾﾝｻ１ｽﾃｰﾀｽ前回値のｱﾄﾞﾚｽ
        DB      @TMR_CHK_MAG            ;!磁気監視ﾀｲﾏ値
        DB      LOW @CMD_ERR_MG2        ;!ｻﾌﾞｺﾏﾝﾄﾞ(磁気ｴﾗｰ２指定)
        DB      LOW R_ERW_YER           ;!ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ
        DB      @YUG_HNF                ;!ｴﾗｰ発生時設定ﾃﾞｰﾀ
                                        ;!
```

【図２９】

〔エラー管理処理テーブル1(2/2)〕

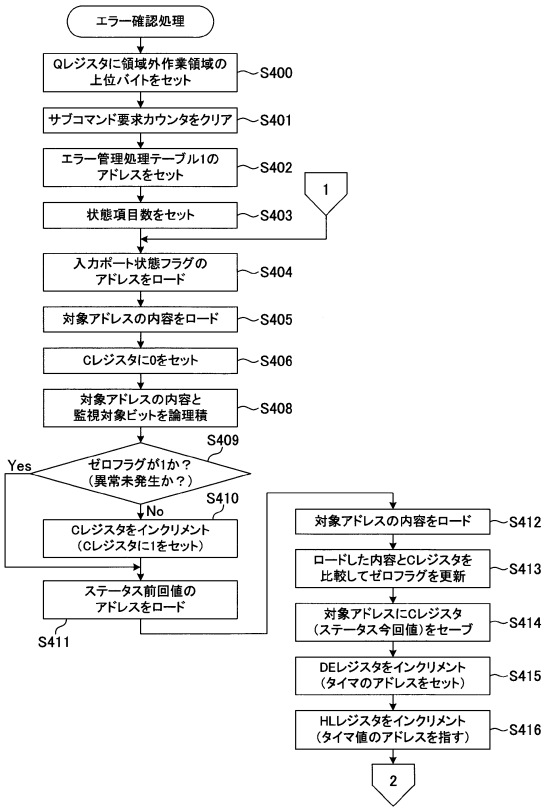
```
        DW      R_IN1_PRT               ;!入力ﾎﾟｰﾄ１状態ﾌﾗｸﾞのｱﾄﾞﾚｽ
        DB      @IN1_M2E_BIT            ;!盤面磁気検出ｾﾝｻ２信号ﾋﾞｯﾄﾃﾞｰﾀ
        DW      R_ERW_M2O               ;!盤面磁気検出ｾﾝｻ２ｽﾃｰﾀｽ前回値
        DB      @TMR_CHK_MAG            ;!磁気監視ﾀｲﾏ値
        DB      LOW @CMD_ERR_MG3        ;!ｻﾌﾞｺﾏﾝﾄﾞ(磁気ｴﾗｰ３指定)
        DB      LOW R_ERW_YER           ;!ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ
        DB      @YUG_HNF                ;!ｴﾗｰ発生時設定ﾃﾞｰﾀ
                                        ;!
        DW      R_INO_PRT               ;!入力ﾎﾟｰﾄ０状態ﾌﾗｸﾞのｱﾄﾞﾚｽ
        DB      @INO_D1E_BIT            ;!電波検出ｾﾝｻ１ﾋﾞｯﾄﾃﾞｰﾀ
        DW      R_ERW_D1O               ;!電波検出ｾﾝｻ１ｽﾃｰﾀｽ前回値
        DB      @TMR_CHK_DEN            ;!電波監視ﾀｲﾏ値
        DB      LOW @CMD_ERR_DE1        ;!ｻﾌﾞｺﾏﾝﾄﾞ(電波ｴﾗｰ１指定)
        DB      0                       ;!ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ
        DB      0                       ;!ｴﾗｰ発生時設定ﾃﾞｰﾀ
                                        ;!
@D_ETB_ER1_NUM  EQU     ($ - D_ETB_ER1)/9;状態項目数
```

【図３０】

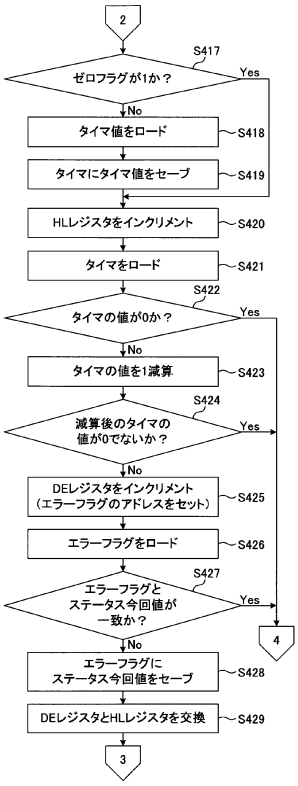
〔エラー管理処理テーブル2〕

```
D_ETB_ER2:
        DB      @ERF_IL1_BIT            ;!不正入賞１ｴﾗｰﾋﾞｯﾄﾃﾞｰﾀ
        DW      R_ERW_FS1               ;!不正入賞１ｴﾗｰﾌﾗｸﾞ
        DB      LOW @CMD_ERR_FS1        ;!ｻﾌﾞｺﾏﾝﾄﾞ(不正入賞ｴﾗｰ１発生指定)
        DB      LOW R_ERW_YER           ;!ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ
        DB      @YUG_HNF                ;!ｴﾗｰ発生時設定ﾃﾞｰﾀ
                                        ;!
        DB      @ERF_IL3_BIT            ;!不正入賞３ｴﾗｰﾋﾞｯﾄﾃﾞｰﾀ
        DW      R_ERW_FS3               ;!不正入賞３ｴﾗｰﾌﾗｸﾞ
        DB      LOW @CMD_ERR_FS3        ;!ｻﾌﾞｺﾏﾝﾄﾞ(不正入賞ｴﾗｰ３発生指定)
        DB      0                       ;!ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ
        DB      0                       ;!ｴﾗｰ発生時設定ﾃﾞｰﾀ
                                        ;!
        DB      @ERF_FDN_BIT            ;!普電異常入賞ｴﾗｰﾋﾞｯﾄﾃﾞｰﾀ
        DW      R_ERW_FDN               ;!普電異常入賞ｴﾗｰﾌﾗｸﾞ
        DB      LOW @CMD_ERR_FDN        ;!ｻﾌﾞｺﾏﾝﾄﾞ(普電異常入賞ｴﾗｰ発生指定)
        DB      0                       ;!ｴﾗｰ発生時設定対象ﾗﾑｱﾄﾞﾚｽ
        DB      0                       ;!ｴﾗｰ発生時設定ﾃﾞｰﾀ
                                        ;!
@D_ETB_ER2_NUM  EQU     ($ - D_ETB_ER2)/6;状態項目数
```

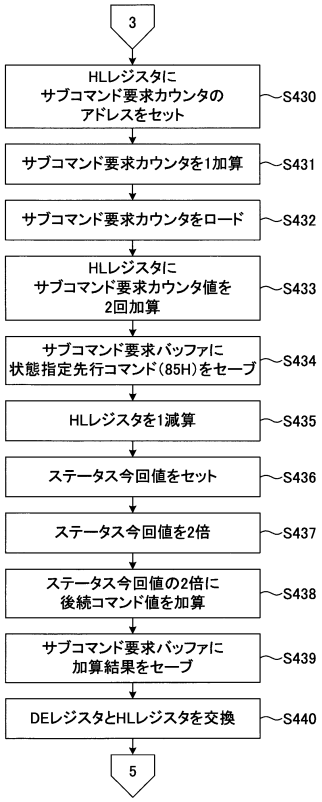
【図３１】



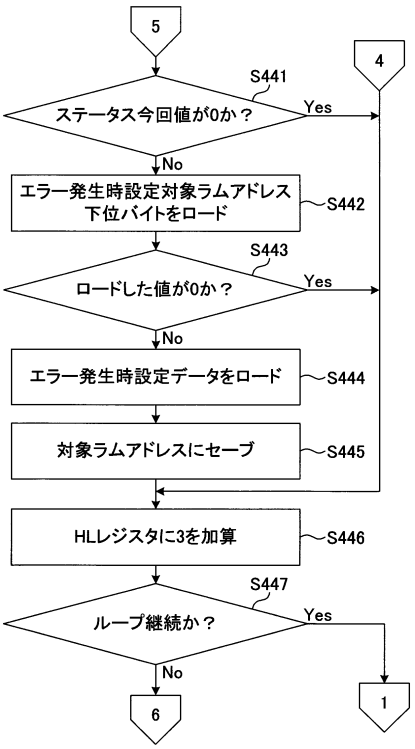
【図３２】



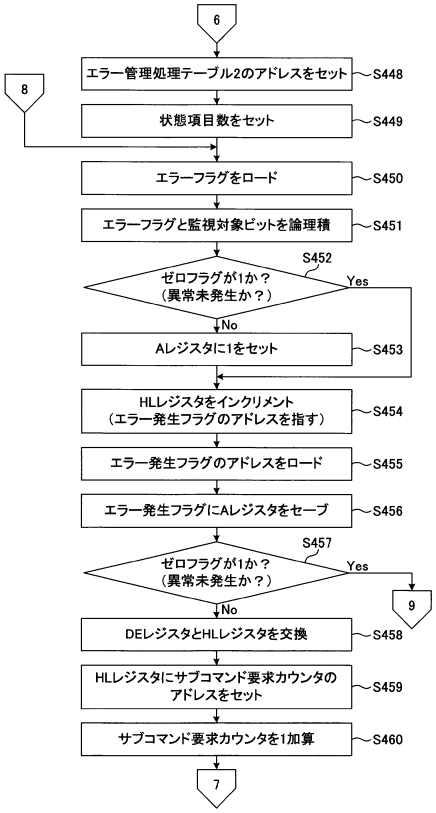
【図３３】



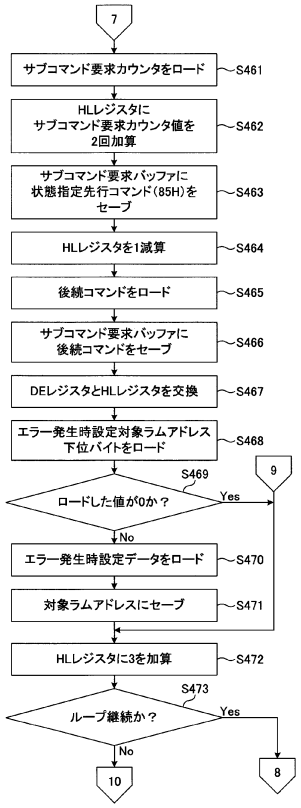
【図３４】



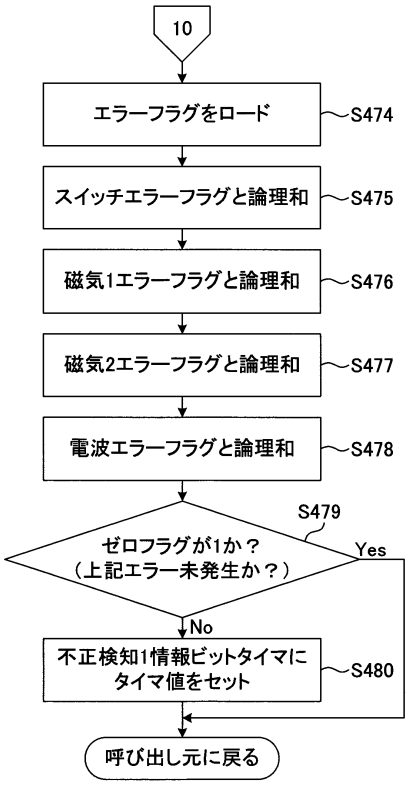
【図３５】



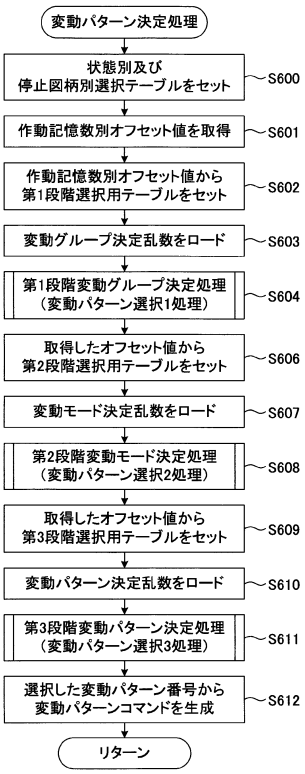
【図３６】



【図３７】



【図３８】

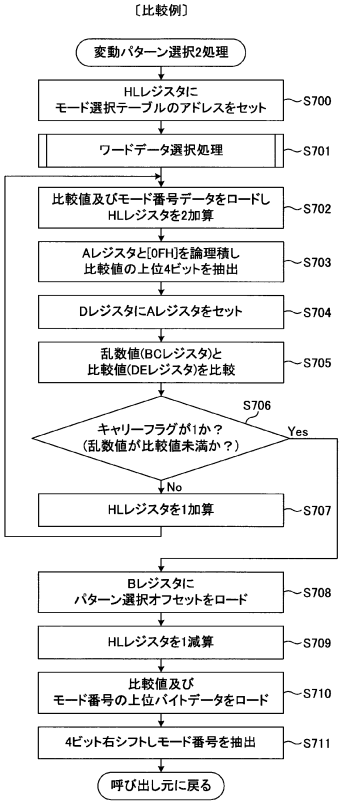


【図３９】

〔比較例の変動パターン選択2処理〕

```
HPT_MOD:
;!------< ﾓｰﾄﾞ選択ﾃｰﾌﾞﾙｱﾄﾞﾚｽ選択処理 >---;
2       LDF     HL,D_MOD_SEL        ;ﾓｰﾄﾞ選択ﾃｰﾌﾞﾙのｱﾄﾞﾚｽをｾｯﾄ
1       RST     WORDSEL             ;ﾜｰﾄﾞﾃﾞｰﾀ選択処理
HPT_MOD_10:
;!------< 比較値取得処理 >----------;
2       LDIN    AE,(HL)             ;比較値及びﾓｰﾄﾞ番号ﾃﾞｰﾀをﾛｰﾄﾞしﾃｰﾌﾞﾙｱﾄﾞﾚｽを 2 加算
2       AND     A,00FH              ;[00FH]と論理積
1       LD      D,A                 ;抽出値をｾｯﾄ
;!------< ﾓｰﾄﾞ判定処理 >------------;
2       CP      BC,DE               ;乱数値と比較
2       JR      C,HPT_MOD_20        ;ｷｬﾘｰﾌﾗｸﾞが 1 であれば分岐
;!------< ﾃｰﾌﾞﾙ更新処理 >----------;
1       INC     HL                  ;ﾃｰﾌﾞﾙｱﾄﾞﾚｽを 1 加算し次のﾃﾞｰﾀﾌﾞﾛｯｸへ更新
2       JR      HPT_MOD_10          ;
HPT_MOD_20:
;!ﾊﾟﾀｰﾝ選択ｵﾌｾｯﾄ設定処理
1       LD      B,(HL)              ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
;!ﾓｰﾄﾞ番号ﾃﾞｰﾀ設定処理
1       DEC     HL                  ;ﾃｰﾌﾞﾙｱﾄﾞﾚｽを 1 減算
1       LD      A,(HL)              ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
3       SRL     A,4                 ;4ﾋﾞｯﾄ右ｼﾌﾄし値を算出
1       RET                         ;
```

【図４０】



【図４１】

(A)〔本実施形態の変動パターン選択2処理〕

```
HPT_MOD:
;!------< ﾓｰﾄﾞ選択ﾃｰﾌﾞﾙｱﾄﾞﾚｽ選択処理 >---;
2       LDF     HL,D_MOD_SEL        ;ﾓｰﾄﾞ選択ﾃｰﾌﾞﾙのｱﾄﾞﾚｽをｾｯﾄ
1       RST     WORDSEL             ;ﾜｰﾄﾞﾃﾞｰﾀ選択処理
HPT_MOD_10:
;!------< 比較値取得処理 >----------;
2       LDIN    AE,(HL)             ;比較値及びﾓｰﾄﾞ番号ﾃﾞｰﾀをﾛｰﾄﾞしﾃｰﾌﾞﾙｱﾄﾞﾚｽを 2 加算
1       INC     HL                  ;HL ﾚｼﾞｽﾀをｲﾝｸﾘﾒﾝﾄ
;!------< ﾓｰﾄﾞ番号ﾃﾞｰﾀ設定処理 >----;
3       DIV     D,A,16              ;ﾓｰﾄﾞ番号を算定
;!------< ﾓｰﾄﾞ判定処理 >------------;
2       CP      BC,DE               ;乱数値と比較
2       JR      NC,HPT_MOD_10       ;ｷｬﾘｰﾌﾗｸﾞが 0 であれば分岐
;!------< ﾊﾟﾀｰﾝ選択ｵﾌｾｯﾄ設定処理 >----;
1       DEC     HL                  ;HL ﾚｼﾞｽﾀをﾃﾞｸﾘﾒﾝﾄ
1       LD      B,(HL)              ;対象ｱﾄﾞﾚｽの内容をﾛｰﾄﾞ
1       RET                         ;
```

(B)〔HLレジスタによる指定テーブル〕

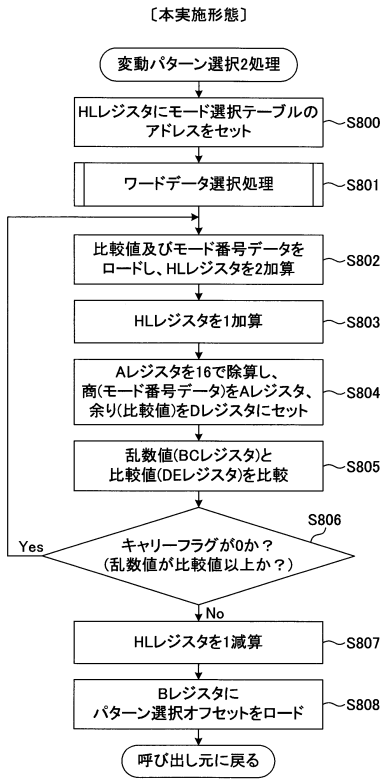
```
D_MOD_SEL_07:
        DW      3368 + 1000H * 004H     ;比較値とモード番号データの合成値
        DB      @D_PAT_SEL_10           ;パターン選択１０オフセット
        DW      3868 + 1000H * 005H     ;比較値とモード番号データの合成値
        DB      @D_PAT_SEL_11           ;パターン選択１１オフセット
        DW      @STP + 1000H * 006H     ;比較値とモード番号データの合成値
        DB      @D_PAT_SEL_12           ;パターン選択１２オフセット
```

【図４２】

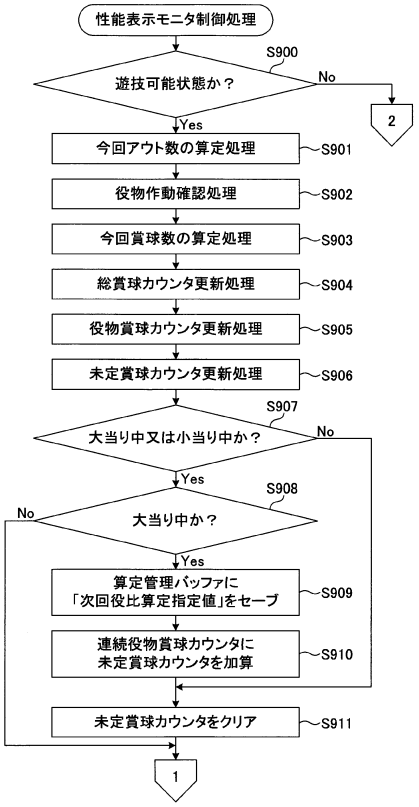
〔テーブルD_MOD_SEL_07を使用した振り分け〕

乱数値	モード番号データ	パターン選択オフセット
0～3367	04H	10(@D_PAT_SEL_10)
3368～3867	05H	11(@D_PAT_SEL_11)
3868以上	06H	12(@D_PAT_SEL_12)

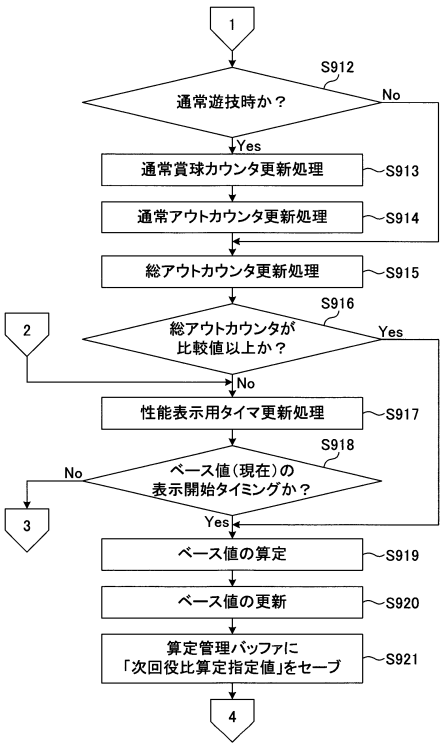
【図４３】



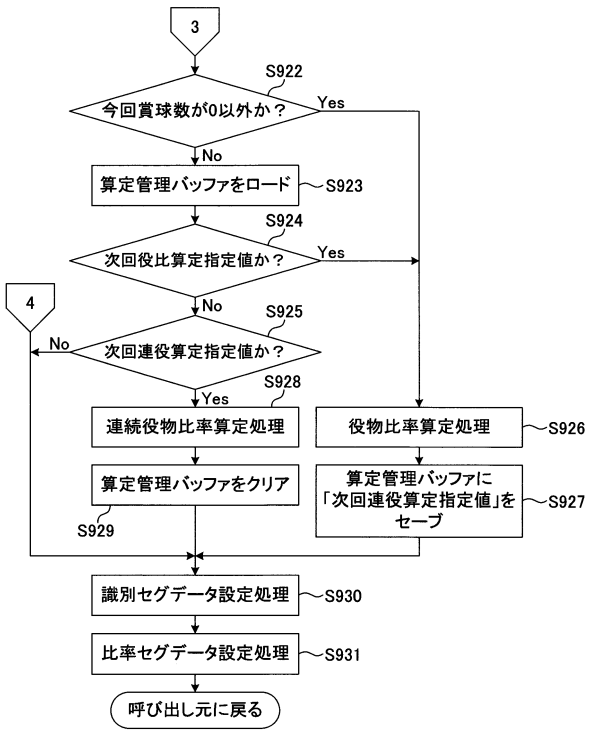
【図４４】



【図４５】



【図４６】



【図４７】

〔未定賞球カウンタ加算処理〕

```
E_SHYMT_30:
 (中略)
 ;!------< 未定賞球ｶｳﾝﾀ更新処理 >----------;
 [1]     LDQ     A,(LOW R_ERW_PRN)       ;今回連続役物賞球数をﾛｰﾄﾞ
 [2]     LDQ     HL,(LOW R_ERW_MTS+0)    ;未定賞球ｶｳﾝﾀ+0 をﾛｰﾄﾞ
 [3]     ADDWB   HL,A                    ;HL ﾚｼﾞｽﾀに A ﾚｼﾞｽﾀを加算
 [4]     LDQ     (LOW R_ERW_MTS+0),HL    ;未定賞球ｶｳﾝﾀ+0 にｾｰﾌﾞ
 [5]     LDQ     A,(LOW R_ERW_MTS+2)     ;未定賞球ｶｳﾝﾀ+2 をﾛｰﾄﾞ
 [6]     ADC     A,0                     ;未定賞球ｶｳﾝﾀの 3ﾊﾞｲﾄ目にｷｬﾘｰﾌﾗｸﾞを加算
 [7]     LDQ     (LOW R_ERW_MTS+2),A     ;未定賞球ｶｳﾝﾀ+2 にｾｰﾌﾞ
```

【図４８】

〔特別遊技管理フェーズ等の確認〕

```
[1]     LD      A,(R_TOK_PHS)           ;特別遊技管理ﾌｪｰｽﾞをﾛｰﾄﾞ
[2]     JCP     C,A,#SHT_PRE_E_SHYMT_42 ;小当りor入賞口開始ｳｴｲﾄ状態指定値と比較しｷｬﾘｰﾌﾗｸﾞが 1 であれば分岐
[3]     LD      A,(R_BIG_FLG)           ;条件装置作動中ﾌﾗｸﾞをﾛｰﾄﾞ
[4]     JCP     NZ,A,@TZ__BHT,E_SHYMT_43 ;特別図柄大当り情報と比較しﾋﾞｯｸﾘﾌﾗｸﾞが 0 であれば分岐
[5]     JTQ     NZ,(LOW R_ERW_MTS+2),E_SHYMT_41;未定賞球ｶｳﾝﾀ+2 が 0 以外であれば分岐
[5]     JTWQ    Z,(LOW R_ERW_MTS+0),E_SHYMT_43 ;未定賞球ｶｳﾝﾀ+0,+1 が 0 であれば分岐
E_SHYMT_41:                             ;
[6]     LDQ     (LOW R_ERW_TSK),@EXD_TSK_YKH ;算定管理ﾊﾞｯﾌｧに次回役比算定指定値をｾｰﾌﾞ
```

【図４９】

〔連続役物賞球カウンタ等更新処理〕

```
 ;!------< 連続役物賞球ｶｳﾝﾀ更新処理 >----------;
 [1]     LDQ     HL,(LOW R_ERW_RYS+0)    ;連続役物賞球ｶｳﾝﾀ+0 をﾛｰﾄﾞ
 [2]     LDQ     DE,(LOW R_ERW_MTS+0)    ;未定賞球ｶｳﾝﾀ+0 をﾛｰﾄﾞ
 [3]     ADD     HL,DE                   ;連続役物賞球ｶｳﾝﾀに未定賞球ｶｳﾝﾀを加算
 [4]     LDQ     (LOW R_ERW_RYS+0),HL    ;連続役物賞球ｶｳﾝﾀ+0 にｾｰﾌﾞ
 [5]     LDQ     A,(LOW R_ERW_RYS+2)     ;連続役物賞球ｶｳﾝﾀ+2 をﾛｰﾄﾞ
 [6]     LDQ     HL,LOW R_ERW_MTS+2      ;未定賞球ｶｳﾝﾀ+2 のｱﾄﾞﾚｽをｾｯﾄ
 [7]     ADC     A,(HL)                  ;連続役物賞球ｶｳﾝﾀの 3ﾊﾞｲﾄ目に未定賞球ｶｳﾝﾀの 3ﾊﾞｲﾄ目及びｷｬﾘｰﾌﾗｸﾞを加算
 [8]     LDQ     (LOW R_ERW_RYS+2),A     ;連続役物賞球ｶｳﾝﾀ+2 にｾｰﾌﾞ
 E_SHYMT_42:                             ;
 ;!------< 未定賞球ｶｳﾝﾀｸﾘｱ処理 >----------;
 [9]     CLRWQ   (LOW R_ERW_MTS+0)       ;未定賞球ｶｳﾝﾀ+0 をｸﾘｱ
 [10]    CLRQ    (LOW R_ERW_MTS+2)       ;未定賞球ｶｳﾝﾀ+2 をｸﾘｱ
```

【図５０】

〔ベース値算定後の算定管理バッファ設定処理〕

```
 ;!---MEMO        算定管理ﾊﾞｯﾌｧ更新処理 1          ;
 E_SHYMT_57:                                       ;
         LDQ     (LOW R_ERW_TSK),@EXD_TSK_YKH;算定管理ﾊﾞｯﾌｧに次回役比算定指定値をｾｰﾌﾞ
         JP      E_SHYMT_90                        ;
```

【図５１】

〔算定管理バッファに基づく処理の振分け〕

```
E_SHYMT_60:                                                    ;
;!----MEMO ベースを算定しない場合は役比算定、連役算定を行うかを確認する。
[1]     JTQ     NZ,(LOW R_ERW_PAY),E_SHYMT_70:今回賞球数が０以外であれば分岐
[2]     LDQ     A,(LOW R_ERW_TSK)       ;算定管理ﾊﾞｯﾌｧをﾛｰﾄﾞ
[2]     JCP     Z,A,@EXD_TSK_YKH,E_SHYMT_70  ;次回役比算定指定値と比較しｾﾞﾛﾌﾗｸﾞが１であれば分岐
[3]     CP      A,@EXD_TSK_RYH          ;次回連役算定指定値と比較
[4]     JP      Z,E_SHYMT_80            ;ｾﾞﾛﾌﾗｸﾞが１であれば分岐
        JP      E_SHYMT_90              ;
```

【図５２】

〔役物比率算定後の算定管理バッファ設定処理〕

```
E_SHYMT_72:                                                    ;
        LDQ     (LOW R_ERW_YAK),A       ;役物比率ﾊﾞｯﾌｧにｾｰﾌﾞ
        LDQ     (LOW R_ERW_TSK),@EXD_TSK_RYH:算定管理ﾊﾞｯﾌｧに次回連役算定指定値をｾｰﾌﾞ
        JP      E_SHYMT_90              ;
```

【図５３】

〔連続役物比率算定後の算定管理バッファ設定処理〕

```
E_SHYMT_82:                                                    ;
        LDQ     (LOW R_ERW_REN),A       ;連続役物比率ﾊﾞｯﾌｧにｾｰﾌﾞ
        CLRQ    (LOW R_ERW_TSK)         ;算定管理ﾊﾞｯﾌｧをｸﾘｱ
```

【図５４】

〔ベース値の算定に必要なステート数(1/2)〕

```
E_SHYMT_50:                                                    ;
;!-------< 除数設定ﾚｼﾞｽﾀ更新処理 >--------;                    ;
1       XOR     A,A                     ;[00H]をｾｯﾄ
5       LDQ     BC,(LOW R_ERW_TJS)      ;通常賞球ｶｳﾝﾀをﾛｰﾄﾞ
4       LDQ     HL,(LOW R_ERW_TJO)      ;通常ｱｳﾄｶｳﾝﾀをﾛｰﾄﾞ
3/2     JR      TZ,E_SHYMT_54           ;第２ｾﾞﾛﾌﾗｸﾞが１であれば分岐
5       OUT     (LOW @DIVB32_+0),HL     ;除数設定ﾚｼﾞｽﾀの最下位ｱﾄﾞﾚｽ+0に出力
2       CLR     HL                      ;[0000H]をｾｯﾄ
5       OUT     (LOW @DIVB32_+2),HL     ;除数設定ﾚｼﾞｽﾀの最下位ｱﾄﾞﾚｽ+2に出力
;!-------< 被除数設定ﾚｼﾞｽﾀ更新処理 >--------;                  ;
1       LD      A,C                     ;通常賞球数の 200 倍を算定
4       MUL     HL,A,200                ;!                      ;
1       EX      DE,HL                   ;!                      ;
1       LD      A,B                     ;!                      ;
4       MUL     HL,A,200                ;!                      ;
2       ADDWB   HL,D                    ;!                      ;
1       LD      A,E                     ;!                      ;
3       OUT     (LOW @DIVA32_+0),A      ;被除数設定ﾚｼﾞｽﾀの最下位ｱﾄﾞﾚｽ+0に出力
5       OUT     (LOW @DIVA32_+1),HL     ;被除数設定ﾚｼﾞｽﾀの最下位ｱﾄﾞﾚｽ+1に出力
4       OUT     (LOW @DIVA32_+3),0      ;被除数設定ﾚｼﾞｽﾀの最下位ｱﾄﾞﾚｽ+3に０を出力
```

【図５５】

〔ベース値の算定に必要なステート数(2/2)〕

```
;!------< ベース値除算待機処理 >------------;
2       LD      B,2             ;Bレジスタに 2 をセット
        :                       ;
E_SHYMT_51:                     ;
3/2     DJNZ    E_SHYMT_51      ;ループ継続であれば分岐
;!------< 除算結果確認処理 >------------------;
2       LD      A,100           ;Aレジスタに 100 をセット
5       IN      HL,(LOW @DIV32__+2) ;除算結果レジスタの最下位アドレス+2 を入力
3/2     JR      NTZ,E_SHYMT_54  ;第２ゼロフラグが０であれば分岐
5       IN      HL,(LOW @DIV32__)   ;除算結果レジスタの最下位アドレスを入力
4       CP      HL,100*2        ;除算結果の 1,2バイト目を確認
3/2     JR      NC,E_SHYMT_54   ;キャリーフラグが０であれば分岐
1       LD      A,L             ;除算結果をセット
2       SRL     A               ;１ビット右シフトし値を算出
2       ADC     A,0             ;キャリーフラグを加算してベース算定値を算定
E_SHYMT_54:                     ;
;!------< ベース算定値更新処理 >------------;
2       LDQ     HL,LOW R_ERW_BAS    ;ベース算定値のアドレスをセット
2       LD      (HL),A          ;対象アドレスの内容にセーブ
```

【図５６】

〔ベース値等の算定に必要な時間〕

算定対象	ステート数	処理時間
ベース値	86	5.375μs
役物比率	81	5.0625μs
連続役物比率	81	5.0625μs

【図５７】

〔算定フラグと実行する処理の対応〕

ベース値 算定フラグ	役物比率 算定フラグ	連続役物比率 算定フラグ	実行する処理
セット	※	※	ベース値を算定
リセット	セット	※	役物比率を算定
リセット	リセット	セット	連続役物比率を算定
リセット	リセット	リセット	何もしない

※はセット・リセットを問わないことを示す。

フロントページの続き

(72)発明者　久保田 智裕
東京都台東区東上野一丁目１６番１号 株式会社平和内
(72)発明者　田村 純一
東京都台東区東上野一丁目１６番１号 株式会社平和内
(72)発明者　山口 功太郎
東京都台東区東上野一丁目１６番１号 株式会社平和内
(72)発明者　西澤 暢
東京都台東区東上野一丁目１６番１号 株式会社平和内
審査官　廣瀬 貴理
(56)参考文献　特開２００３－０００８８９（ＪＰ，Ａ）
特開２０２１－１８０７７５（ＪＰ，Ａ）
特開２０１６－１８７６１５（ＪＰ，Ａ）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ａ６３Ｆ　７／０２
Ａ６３Ｆ　５／０４