

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5543229号  
(P5543229)

(45) 発行日 平成26年7月9日(2014.7.9)

(24) 登録日 平成26年5月16日(2014.5.16)

(51) Int.Cl.

A63F 5/04 (2006.01)

F 1

A 6 3 F 5/04 5 1 2 Z  
A 6 3 F 5/04 5 1 6 D

請求項の数 1 (全 79 頁)

(21) 出願番号 特願2010-12560 (P2010-12560)  
 (22) 出願日 平成22年1月22日 (2010.1.22)  
 (65) 公開番号 特開2011-147676 (P2011-147676A)  
 (43) 公開日 平成23年8月4日 (2011.8.4)  
 審査請求日 平成24年12月20日 (2012.12.20)

(73) 特許権者 000144153  
 株式会社三共  
 東京都渋谷区渋谷三丁目29番14号  
 (74) 代理人 100098729  
 弁理士 重信 和男  
 (74) 代理人 100116757  
 弁理士 清水 英雄  
 (74) 代理人 100123216  
 弁理士 高木 祐一  
 (74) 代理人 100089336  
 弁理士 中野 佳直  
 (74) 代理人 100163212  
 弁理士 溝淵 良一  
 (74) 代理人 100148161  
 弁理士 秋庭 英樹

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】

所定の遊技を行うことが可能な遊技機であって、  
遊技制御処理プログラムに基づき遊技機における遊技制御を実行する制御用CPUが内蔵された遊技制御用マイクロコンピュータと、  
前記遊技制御用マイクロコンピュータに内蔵又は外付けされ、乱数値となる数値データを生成する乱数回路と、

電力供給が停止しても格納されているデータが保持されるバックアップ領域を有するデータ記憶手段と、

を備え、

前記乱数回路は、

数値データを予め定められた手順により更新して出力する数値更新手段と、

前記数値更新手段から出力された数値データを乱数値として取り込んで格納する乱数値格納手段と、

を含み、

前記遊技制御用マイクロコンピュータは、

前記乱数回路によって生成された乱数値に基づいて、前記制御用CPUにより所定の決定を行う制御決定手段と、

所定信号の入力に基づいて前記数値更新手段から出力された数値データが前記乱数値格納手段に格納されたときにオン状態にされて新たな数値データの格納を制限する一方、前

記乱数値格納手段に格納された数値データが乱数値の読み出タイミングにて前記制御用CPUにより読み出されたときにオフ状態にされて新たな数値データの格納を許可する所定のフラグと、

電断条件が成立したときに前記バックアップ領域に保持されているデータに基づいて復帰可能とするための電断処理を実行する電断処理実行手段と、

前記電断処理の実行後、電力供給が停止せずに一定時間継続した場合に起動命令を行う起動命令手段と、

前記起動命令を契機に前記バックアップ領域に保持されているデータに基づいて制御状態を復帰させる制御状態復帰手段と、

前記電断処理の実行後、電力供給が停止するのを待機しているときに、前記所定のフラグをオフ状態にする電断待機時処理手段と、

を含む

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、弾球遊技機やスロットマシンなどの遊技機に関する。

【背景技術】

【0002】

この種の遊技機では、所定の契機で抽選を行い、その抽選に当選することで大当たりなど遊技者にとって有利な状態に制御するものが一般的である。抽選を行う方法としては、一定の範囲で数値を更新するカウンタと、抽選契機となる事象を設け、抽選契機となる事象が検出された際に、カウンタの数値を抽出し、抽出した値を乱数として用いる方法がある（例えば、特許文献1参照）。

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2005-152268号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

特許文献1に記載の遊技機のように、カウンタの数値を抽出して抽選を行う方法では、抽選契機となる事象が検出されてカウンタの数値を抽出した後、信号線にノイズがのることによって再度カウンタから数値が抽出された場合、抽選契機となる事象の検出によって抽出された数値とは異なる数値によって抽選が行われてしまうこととなる。

【0005】

このような問題を解決する方法として、カウンタから一度数値を抽出した後、抽出した数値が読み出されるまで保持しておく方法が考えられるが、このような方法を採用した場合、電源投入時など、電圧が不安定な状態でスイッチ類が誤って検出される可能性があり、この場合、本来の抽選契機とは異なる検出にも関わらず、カウンタから数値が抽出されて保持されたままとなり、本来の抽選契機とは異なるタイミングで抽出された数値によって抽選が行われてしまうという新たな問題が生じることとなる。

【0006】

本発明は、このような問題点に着目してなされたものであり、本来の抽選契機とは異なるタイミングで抽出された数値を用いて抽選が行われてしまうことを防止できる遊技機を提供することを目的とする。

【課題を解決するための手段】

【0007】

上記課題を解決するために、本発明の請求項1に記載の遊技機は、  
所定の遊技を行うことが可能な遊技機であって、

10

20

30

40

50

遊技制御処理プログラムに基づき遊技機における遊技制御を実行する制御用CPUが内蔵された遊技制御用マイクロコンピュータと、

前記遊技制御用マイクロコンピュータに内蔵又は外付けされ、乱数値となる数値データを生成する乱数回路と、

電力供給が停止しても格納されているデータが保持されるバックアップ領域を有するデータ記憶手段と、

を備え、

前記乱数回路は、

数値データを予め定められた手順により更新して出力する数値更新手段と、

前記数値更新手段から出力された数値データを乱数値として取り込んで格納する乱数値格納手段と、

を含み、

前記遊技制御用マイクロコンピュータは、

前記乱数回路によって生成された乱数値に基づいて、前記制御用CPUにより所定の決定を行う制御決定手段と、

所定信号の入力に基づいて前記数値更新手段から出力された数値データが前記乱数値格納手段に格納されたときにオン状態にされて新たな数値データの格納を制限する一方、前記乱数値格納手段に格納された数値データが乱数値の読み出タイミングにて前記制御用CPUにより読み出されたときにオフ状態にされて新たな数値データの格納を許可する所定のフラグと、

電断条件が成立したときに前記バックアップ領域に保持されているデータに基づいて復帰可能とするための電断処理を実行する電断処理実行手段と、

前記電断処理の実行後、電力供給が停止せずに一定時間継続した場合に起動命令を行う起動命令手段と、

前記起動命令を契機に前記バックアップ領域に保持されているデータに基づいて制御状態を復帰させる制御状態復帰手段と、

前記電断処理の実行後、電力供給が停止するのを待機しているときに、前記所定のフラグをオフ状態にする電断待機時処理手段と、

を含む

ことを特徴としている。

本発明の手段1に記載の遊技機は、

所定の遊技を行うことが可能な遊技機(スロットマシン1)であって、

数値データを更新する数値更新手段(乱数列変更回路555)と、

所定の信号(スタートスイッチ7のon)が入力されたことに基づいて前記数値更新手段(乱数列変更回路555)によって更新された数値データを乱数値として抽出し、数値データ格納領域(乱数値レジスタR1D)に格納する乱数抽出手段(乱数ラッチセレクタ558A)と、

前記所定の信号(スタートスイッチ7のon)が入力されたことに基づき前記数値データ格納領域(乱数値レジスタR1D)に格納されている数値データを読み出す数値データ読み出手段(ゲーム開始時の数値データの読み出し)と、

前記数値データ読み出手段が読み出した数値データを用いて遊技に関連する決定を行う遊技関連決定手段(内部抽選)と、

前記数値データ格納領域(乱数値レジスタR1D)に前記乱数抽出手段(乱数ラッチセレクタ558A)により数値データが格納された後、該格納されている数値データが読み出されるまでは、前記乱数抽出手段(乱数ラッチセレクタ558A)により新たな数値データが格納されることなく、該格納されている数値データを保持する数値データ保持手段(新たな数値データのラッチの禁止)と、

電力供給が停止しても電力供給停止前の制御状態に復帰させるためのバックアップデータ(RAM507の格納データ)を保持するバックアップデータ保持手段(バックアップ電源)と、

10

20

30

40

50

電断条件（電圧低下信号の検出）が成立したときに前記バックアップデータ（RAM507の格納データ）に基づいて復帰可能とするための電断処理（電断処理（メイン））を実行する電断処理実行手段と、

前記電断処理（電断処理（メイン））の実行後、電力供給が停止せずに一定時間継続した場合に起動命令（ユーザリセット）を行う起動命令手段と、

前記起動命令（ユーザリセット）を契機に前記バックアップデータ（RAM507の格納データ）に基づいて制御状態を復帰させる制御状態復帰手段と、

前記電断処理（電断処理（メイン））の実行後、電力供給が停止するのを待機している期間において、前記数値データ格納領域（乱数値レジスタR1D）に数値データが格納されているか否かを判定し、前記数値データが格納されていると判定した場合に、該格納されている数値データを読み出すことにより、前記数値データ保持手段による数値データの保持を解除する保持解除手段（数値データのダミー読み出し）と、  
10

を備える

ことを特徴としている。

この特徴によれば、所定の信号の入力により抽出された数値データが数値データ格納領域に格納された後、該格納された数値データが読み出されるまでは、格納されている数値データが保持され、新たな数値データによって上書きされてしまうことがないので、静電気などにより信号線にノイズがのっても数値データが変わってしまうことがなく、このような場合であっても所定の信号が入力されたタイミングで抽出された数値データと異なる数値データを用いて入賞の発生を許容するか否かの決定が行われてしまうことがない。  
20

また、瞬停などにより一時的に電圧が低下して電断処理が実行された場合には、一定時間が経過しても電力供給が停止しない場合に起動命令が行われ、もとの状態に復帰するとともに、電力供給の停止を待機している期間において数値データ格納領域に格納された場合には、その数値データが読み出され、数値データ保持手段による数値データの保持が解除されるようになっており、瞬停など、一時に電圧が不安定となり、信号線にノイズがのって数値データが数値データ格納領域に格納され、その状態が保持されたままの状態となつても、すぐに読み出されて数値データ格納領域に新たな数値データを格納可能な状態となるため、瞬停などの復帰後、その間にノイズなどによって保持されていた数値データを用いて遊技に関連する決定が行われしまうことがなく、本来の抽選契機とは異なるタイミングで抽出された数値を用いて抽選が行われてしまうことを防止できる。  
30

尚、数値データ保持手段は、前記数値データ格納領域に前記乱数抽出手段により数値データが格納された後、該格納されている数値データが読み出されるまで、前記乱数抽出手段による数値データの新たな抽出を禁止することにより、数値データ格納領域に格納されている数値データを保持するようにしても良いし、前記数値データ格納領域に前記乱数抽出手段により数値データが格納された後、該格納されている数値データが読み出されるまで、前記乱数抽出手段により数値データの抽出が行われても数値データ格納領域への格納を禁止することにより、数値データ格納領域に格納されている数値データを保持するようにしても良い。

#### 【0008】

本発明の手段2に記載の遊技機は、手段1に記載の遊技機であって、  
40

前記数値データ読出手段は、前記所定の信号（スタートスイッチ7のon）が所定期間（約2.24ms）継続して入力されたことを条件に、前記数値データを読み出すことを特徴としている。

この特徴によれば、静電気などのノイズによって所定の信号の入力が誤って検出されたにも関わらず、数値データが読み出されて遊技に関連する決定が行われてしまうことを防止できる。

#### 【0009】

本発明の手段3に記載の遊技機は、手段2に記載の遊技機であって、

前記数値データ格納領域（乱数値レジスタR1D）に前記乱数抽出手段（乱数ラッチセレクタ558A）により数値データが格納されたときに、数値データが新たに格納された  
50

旨を示す格納情報（乱数ラッチフラグ）を設定する格納情報設定手段（乱数ラッチフラグレジスタ R D F M ）を備え、

前記数値データ読出手段は、前記所定の信号（スタートスイッチ 7 の o n ）が所定期間（約 2 . 2 4 m s ）継続して入力され、かつ前記格納情報（乱数ラッチフラグ）が設定されていることを条件に、前記数値データを読み出す

ことを特徴としている。

この特徴によれば、乱数抽出手段により数値データが抽出されていない状態であるにも関わらず、数値データが読み出されて遊技に関連する決定が行われてしまうことがなく、本来の抽選契機のタイミングで抽出された数値データを用いて遊技に関連する決定を行うことができる。

10

#### 【 0 0 1 0 】

本発明の手段 4に記載の遊技機は、手段 1 ~ 3のいずれかに記載の遊技機であって、

前記遊技機毎に個別に割り当てられた識別符号（I D ナンバー）が記憶される不揮発性メモリ（R O M 5 0 6 ）を備え、

前記遊技機は、電力供給が開始したときに前記不揮発性メモリ（R O M 5 0 6 ）に記憶されている識別符号（I D ナンバー）に基づいて初期数値データ（数値データのスタート値）を生成する初期数値データ生成手段（乱数回路設定処理）をさらに備え、

前記数値更新手段（乱数列変更回路 5 5 5 ）は、電力供給が開始したときに前記初期数値データ生成手段（乱数回路設定処理）により生成された初期数値データ（数値データのスタート値）から前記数値データの更新を開始する

20

ことを特徴としている。

この特徴によれば、遊技機毎に個別に割り当てられた識別符号に基づいて生成された初期数値データから数値データの更新が開始されるので、遊技機毎に初期数値データが異なり、初期数値データを特定することが困難となるため、特定の数値データのタイミングを狙って開始操作手段の操作を検出させる不正を効果的に防止できる。

#### 【 0 0 1 1 】

本発明の手段 5に記載の遊技機は、手段 1 ~ 4のいずれかに記載の遊技機であって、遊技の制御を行う遊技制御手段（C P U 5 0 5 ）を備え、

前記数値データ更新手段（乱数列変更回路 5 5 5 ）は、前記遊技制御手段（C P U 5 0 5 ）を動作させるための動作クロック（制御用クロック）とは周期の異なる動作クロック（乱数用クロック）を入力して前記数値データを更新する

30

ことを特徴としている。

この特徴によれば、遊技制御手段の動作と数値データの更新周期とが同期することにより、遊技関連決定手段により用いられる乱数値に偏りが生じてしまうことを防止できるとともに、遊技制御手段に不正基板が接続されても遊技制御手段の動作から数値データの更新周期を特定することは不可能となるため、特定の数値データのタイミングを狙って所定の信号を入力させる不正を効果的に防止できる。

#### 【 0 0 1 2 】

本発明の手段 6に記載の遊技機は、手段 1 ~ 5のいずれかに記載の遊技機であって、発光ダイオードにて構成されるL E D 表示器と、

40

カウンタ値に基づいて計時を行う計時手段（ウォッチドッグタイマ 4 9 a ）と、

前記L E D 表示器をダイナミック点灯させる駆動信号の入力に基づいて前記計時手段（ウォッчドッグタイマ 4 9 a ）のカウンタ値を初期化（クリア）するカウンタ値初期化手段と、

を備え、

前記電断処理（電断処理（メイン））は、前記L E D 表示器をダイナミック点灯させる駆動信号の出力を停止させる処理（出力ポートの初期化）を含み、

前記起動命令手段は、前記計時手段（ウォッчドッグタイマ 4 9 a ）によるカウンタ値が初期化されずに予め定められた閾値に到達した場合（オーバーフローした場合）に前記起動命令（ユーザリセット）を行う

50

ことを特徴としている。

この特徴によれば、L E D 表示器の駆動信号を監視するのみで正常に動作しているか否かを監視することが可能となる。

#### 【0013】

本発明の手段7に記載の遊技機は、手段1～6のいずれかに記載の遊技機であって、

前記遊技機は、電力供給が開始したときに初期数値データ（乱数の初期値）を生成する初期数値データ生成手段をさらに備え、

前記数値更新手段は、電力供給が開始したときに前記初期数値データ生成手段により生成された初期数値データ（乱数の初期値）から前記数値データの更新を開始するが、前記電断処理の実行後、電力供給が停止するのを待機している期間においても前記数値データの更新を継続し、前記電断処理の実行後、電力供給が停止せずに前記起動命令（ユーザリセット）により前記制御状態復帰手段が制御状態を復帰させる場合には、前記初期数値データ（乱数の初期値）を用いずに前記数値データの更新を継続する

ことを特徴としている。

この特徴によれば、意図的に起動命令を行わせても数値データの更新は継続するため、数値データの更新が開始されるタイミングを特定することは困難となり、特定の数値データのタイミングを狙って所定の信号を入力させる不正を効果的に防止できる。

#### 【図面の簡単な説明】

#### 【0014】

【図1】本発明が適用された遊技機の一例であるスロットマシンの正面図である。

20

【図2】スロットマシンの内部構造図である。

【図3】リールの図柄配列を示す図である。

【図4】スロットマシンの構成を示すブロック図である。

【図5】メイン制御部の構成を示すブロック図である。

【図6】メイン制御部におけるアドレスマップの一例を示す図である。

【図7】プログラム管理エリア及び内蔵レジスタの主要部分を例示する図である。

【図8】ヘッダ及び機能設定における設定内容の一例を示す図である。

【図9】第1乱数初期設定、第2乱数初期設定及び割込み初期設定における設定内容の一例を示す図である。

【図10】セキュリティ時間設定における設定内容の一例を示す図である。

30

【図11】内部情報レジスタの構成例等を示す図である。

【図12】乱数回路の構成例を示すブロック図である。

【図13】乱数列変更レジスタの構成例等を示す図である。

【図14】乱数列変更回路による乱数更新規則の変更動作を示す説明図である。

【図15】乱数列変更回路による乱数更新規則の変更動作を示す説明図である。

【図16】乱数値取込レジスタの構成例等を示す図である。

【図17】乱数ラッチ選択レジスタの構成例等を示す図である。

【図18】乱数値レジスタの構成例を示す図である。

【図19】乱数ラッチフラグレジスタの構成例等を示す図である。

【図20】乱数割込み制御レジスタの構成例等を示す図である。

40

【図21】入力ポートレジスタの構成例等を示す図である。

【図22】入賞として定められた役の構成及び遊技状態別の内部抽選の対象役を示す図である。

【図23】メイン制御部が起動時に実行する起動処理（メイン）の制御内容を示すフローチャートである。

【図24】メイン制御部が実行するセキュリティチェック処理の制御内容を示すフローチャートである。

【図25】メイン制御部が実行する乱数回路設定処理の制御内容を示すフローチャートである。

【図26】メイン制御部が実行する乱数回路異常検査処理の制御内容を示すフローチャー

50

トである。

【図27】メイン制御部がエラー発生時に実行するエラー処理の制御内容を示すフローチャートである。

【図28】メイン制御部が実行する設定変更処理の制御内容を示すフローチャートである。

【図29】メイン制御部が設定変更処理後に実行するゲーム処理の制御内容を示すフローチャートである。

【図30】メイン制御部が実行するB E T処理の制御内容を示すフローチャートである。

【図31】メイン制御部が実行するB E T処理の制御内容を示すフローチャートである。

【図32】メイン制御部が実行するB E T処理の制御内容を示すフローチャートである。

【図33】メイン制御部が一定間隔毎に実行するタイマ割込処理（メイン）の制御内容を示すフローチャートである。

【図34】メイン制御部が一定間隔毎に実行するタイマ割込処理（メイン）の制御内容を示すフローチャートである。

【図35】メイン制御部がタイマ割込処理（メイン）において実行するスイッチ入力判定処理の制御内容を示すフローチャートである。

【図36】メイン制御部がタイマ割込処理（メイン）において電断を検出したことに応じて実行する電断処理（メイン）の制御内容を示すフローチャートである。

【図37】乱数回路における動作を説明するためのタイミングチャートである。

【図38】スタートスイッチの操作と乱数値レジスタの関連を示すタイミングチャートである。

【図39】スタートスイッチの操作と乱数値レジスタの関連を示すタイミングチャートである。

【図40】スタートスイッチの操作と乱数値レジスタの関連を示すタイミングチャートである。

【図41】スタートスイッチの操作と乱数値レジスタの関連を示すタイミングチャートである。

【図42】スタートスイッチの検出状況を示すタイミングチャートである。

【発明を実施するための形態】

【0015】

本発明の実施例を以下に説明する。

【実施例】

【0016】

本発明が適用された遊技機の一例であるスロットマシンの実施例を図面を用いて説明すると、本実施例のスロットマシン1は、前面が開口する筐体1aと、この筐体1aの側端に回動自在に枢支された前面扉1bと、から構成されている。

【0017】

本実施例のスロットマシン1の筐体1aの内部には、図2に示すように、外周に複数種の図柄が配列されたリール2L、2C、2R（以下、左リール、中リール、右リール）が水平方向に並設されており、図1に示すように、これらリール2L、2C、2Rに配列された図柄のうち連続する3つの図柄が前面扉1bに設けられた透視窓3から見えるように配置されている。

【0018】

リール2L、2C、2Rの外周部には、図3に示すように、それぞれ「黒7」、「網7（図中網掛け7）」、「白7」、「BAR」、「リプレイ」、「スイカ」、「黒チェリー」、「白チェリー」、「ベル」、「オレンジ」といった互いに識別可能な複数種類の図柄が所定の順序で、それぞれ21個ずつ描かれている。リール2L、2C、2Rの外周部に描かれた図柄は、透視窓3において各々上中下三段に表示される。

【0019】

各リール2L、2C、2Rは、各々対応して設けられリールモータ32L、32C、3

10

20

30

40

50

2 R (図4参照)によって回転させることで、各リール2 L、2 C、2 Rの図柄が透視窓3に連続的に変化しつつ表示されるとともに、各リール2 L、2 C、2 Rの回転を停止させることで、透視窓3に3つの連続する図柄が表示結果として導出表示されるようになっている。

#### 【0020】

リール2 L、2 C、2 Rの内側には、リール2 L、2 C、2 Rそれぞれに対して、基準位置を検出するリールセンサ33 L、33 C、33 Rと、リール2 L、2 C、2 Rを背面から照射するリールLED55と、が設けられている。また、リールLED55は、リール2 L、2 C、2 Rの連続する3つの図柄に対応する12のLEDからなり、各図柄をそれぞれ独立して照射可能とされている。10

#### 【0021】

前面扉1 bの各リール2 L、2 C、2 Rの手前側(遊技者側)の位置には、液晶表示器51(図1参照)の表示領域51 aが配置されている。液晶表示器51は、液晶素子に対して電圧が印加されていない状態で、透過性を有するノーマリー・ホワイトタイプの液晶パネルを有しており、表示領域51 aの透視窓3に対応する透過領域51 b及び透視窓3を介して遊技者側から各リール2 L、2 C、2 Rが視認できるようになっている。また、表示領域51 aの透過領域51 bを除く領域の裏面には、背後から表示領域51 aを照射するバックライト(図示略)が設けられるとともに、さらにその裏面には、内部を隠蔽する隠蔽部材(図示略)が設けられている。

#### 【0022】

前面扉1 bには、メダルを投入可能なメダル投入部4、メダルが払い出されるメダル払出口9、クレジット(遊技者所有の遊技用価値として記憶されているメダル数)を用いてメダル1枚分の賭数を設定する際に操作される1枚BETスイッチ5、クレジットを用いて、その範囲内において遊技状態に応じて定められた規定数の賭数のうち最大の賭数(本実施例では遊技状態がRB(BB)の場合には2、通常遊技状態では3)を設定する際に操作されるMAXBETスイッチ6、クレジットとして記憶されているメダル及び賭数の設定に用いたメダルを精算する(クレジット及び賭数の設定に用いた分のメダルを返却させる)際に操作される精算スイッチ10、ゲームを開始する際に操作されるスタートスイッチ7、リール2 L、2 C、2 Rの回転を各々停止する際に操作されるストップスイッチ8 L、8 C、8 R、が遊技者により操作可能にそれぞれ設けられている。30

#### 【0023】

また、前面扉1 bには、クレジットとして記憶されているメダル枚数が表示されるクレジット表示器11、後述するBB中のメダルの獲得枚数やエラー発生時にその内容を示すエラーコード等が表示される遊技補助表示器12、入賞の発生により払い出されたメダル枚数が表示されるペイアウト表示器13が設けられている。

#### 【0024】

また、前面扉1 bには、賭数が1設定されている旨を点灯により報知する1BETLE D14、賭数が2設定されている旨を点灯により報知する2BETLED15、賭数が3設定されている旨を点灯により報知する3BETLED16、メダルの投入が可能な状態を点灯により報知する投入要求LED17、スタートスイッチ7の操作によるゲームのスタート操作が有効である旨を点灯により報知するスタート有効LED18、ウェイト(前回のゲーム開始から一定期間経過していないためにリールの回転開始を待機している状態)中である旨を点灯により報知するウェイト中LED19、後述するリプレイゲーム中である旨を点灯により報知するリプレイ中LED20が設けられている。40

#### 【0025】

MAXBETスイッチ6の内部には、1枚BETスイッチ5及びMAXBETスイッチ6の操作による賭数の設定操作が有効である旨を点灯により報知するBETスイッチ有効LED21(図4参照)が設けられており、ストップスイッチ8 L、8 C、8 Rの内部には、該当するストップスイッチ8 L、8 C、8 Rによるリールの停止操作が有効である旨を点灯により報知する左、中、右停止有効LED22 L、22 C、22 R(図4参照)が50

それぞれ設けられている。

**【0026】**

前面扉1bの内側には、所定のキー操作により後述するエラー状態及び後述する打止状態を解除するためのリセット操作を検出するリセットスイッチ23、後述する設定値の変更中や設定値の確認中にその時点の設定値が表示される設定値表示器24、メダル投入部4から投入されたメダルの流路を、筐体1a内部に設けられた後述のホッパータンク34a(図2参照)側またはメダル払出口9側のいずれか一方に選択的に切り替えるための流路切替ソレノイド30、メダル投入部4から投入され、ホッパータンク34a側に流下したメダルを検出する投入メダルセンサ31を有するメダルセレクタ(図示略)、前面扉1bの開放状態を検出するドア開放検出スイッチ25(図4参照)、通常時においてはエラー状態や後述の打止状態を解除するためのリセットスイッチとして機能し、後述の設定変更状態においては後述する内部抽選の当選確率(出玉率)の設定値を変更するための設定スイッチとして機能するリセット/設定スイッチ38が設けられている。10

**【0027】**

筐体1a内部には、図2に示すように、前述したリール2L、2C、2R、リールモータ32L、32C、32R、各リール2L、2C、2Rのリール基準位置をそれぞれ検出可能なリールセンサ33L、33C、33R(図4参照)からなるリールユニット2、外部出力信号を出力するための外部出力基板1000、メダル投入部4から投入されたメダルを貯留するホッパータンク34a、ホッパータンク34aに貯留されたメダルをメダル払出口9より払い出すためのホッパーモータ34b、ホッパーモータ34bの駆動により払い出されたメダルを検出する払出センサ34cからなるホッパーユニット34、電源ボックス100が設けられている。20

**【0028】**

ホッパーユニット34の側部には、ホッパータンク34aから溢れたメダルが貯留されるオーバーフロータンク35が設けられている。オーバーフロータンク35の内部には、貯留された所定量のメダルを検出可能な高さに設けられた左右に離間する一対の導電部材からなる満タンセンサ35aが設けられており、導電部材がオーバーフロータンク35内に貯留されたメダルを介して接触することにより導電したときに内部に貯留されたメダル貯存量が所定量以上となったこと、すなわちオーバーフロータンクが満タン状態となったことを検出できるようになっている。30

**【0029】**

電源ボックス100の前面には、後述のBB終了時に打止状態(リセット操作がなされるまでゲームの進行が規制される状態)に制御する打止機能の有効/無効を選択するための打止スイッチ36a、後述のBB終了時に自動精算処理(クレジットとして記憶されているメダルを遊技者の操作によらず精算(返却)する処理)に制御する自動精算機能の有効/無効を選択するための自動精算スイッチ36b、設定変更状態または設定確認状態に切り替えるための設定キースイッチ37、電源をON/OFFする際に操作される電源スイッチ39が設けられている。

**【0030】**

本実施例のスロットマシン1においてゲームを行う場合には、まず、メダルをメダル投入部4から投入するか、或いはクレジットを使用して賭数を設定する。クレジットを使用するには1枚BETスイッチ5またはMAXBETスイッチ6を操作すれば良い。遊技状態に応じて定められた規定数の賭数が設定されると、入賞ラインL1~L5(図1参照)が有効となり、スタートスイッチ7の操作が有効な状態、すなわち、ゲームが開始可能な状態となる。本実施例では、規定数の賭数として遊技状態がRB(BB)では2枚、通常遊技状態では3枚が定められている。尚、遊技状態に対応する規定数のうち最大数を超えてメダルが投入された場合には、その分はクレジットに加算される。40

**【0031】**

入賞ラインとは、各リール2L、2C、2Rの透視窓3に表示された図柄の組合せが入賞図柄の組合せであるかを判定するために設定されるラインである。本実施例では、図150

に示すように、各リール 2 L、2 C、2 R の中段に並んだ図柄に跨って設定された入賞ライン L 1、各リール 2 L、2 C、2 R の上段に並んだ図柄に跨って設定された入賞ライン L 2、各リール 2 L、2 C、2 R の下段に並んだ図柄に跨って設定された入賞ライン L 3、リール 2 L の上段、リール 2 C の中段、リール 2 R の下段、すなわち右下がりに並んだ図柄に跨って設定された入賞ライン L 4、リール 2 L の下段、リール 2 C の中段、リール 2 R の上段、すなわち右上がりに並んだ図柄に跨って設定された入賞ライン L 5 の 5 種類が入賞ラインとして定められている。

#### 【0032】

ゲームが開始可能な状態でスタートスイッチ 7 を操作すると、各リール 2 L、2 C、2 R が回転し、各リール 2 L、2 C、2 R の図柄が連続的に変動する。この状態でいずれかのストップスイッチ 8 L、8 C、8 R を操作すると、対応するリール 2 L、2 C、2 R の回転が停止し、透視窓 3 に表示結果が導出表示される。10

#### 【0033】

そして全てのリール 2 L、2 C、2 R が停止されることで 1 ゲームが終了し、有効化されたいずれかの入賞ライン L 1 ~ L 5 上に予め定められた図柄の組合せ（以下、役とも呼ぶ）が各リール 2 L、2 C、2 R の表示結果として停止した場合には入賞が発生し、その入賞に応じて定められた枚数のメダルが遊技者に対して付与され、クレジットに加算される。また、クレジットが上限数（本実施例では 50）に達した場合には、メダルが直接メダル払出口 9（図 1 参照）から払い出されるようになっている。尚、有効化された複数の入賞ライン上にメダルの払出を伴う図柄の組合せが揃った場合には、有効化された入賞ラインに揃った図柄の組合せそれぞれに対して定められた払出枚数を合計し、合計した枚数のメダルが遊技者に対して付与されることとなる。ただし、1 ゲームで付与されるメダルの払出枚数には、上限（本実施例では 15 枚）が定められており、合計した払出枚数が上限を超える場合には、上限枚数のメダルが付与されることとなる。また、有効化されたいずれかの入賞ライン L 1 ~ L 5 上に、遊技状態の移行を伴う図柄の組合せが各リール 2 L、2 C、2 R の表示結果として停止した場合には図柄の組合せに応じた遊技状態に移行するようになっている。20

#### 【0034】

図 4 は、スロットマシン 1 の構成を示すブロック図である。スロットマシン 1 には、図 4 に示すように、遊技制御基板 4 0、演出制御基板 9 0、電源基板 1 0 1 が設けられており、遊技制御基板 4 0 によって遊技状態が制御され、演出制御基板 9 0 によって遊技状態に応じた演出が制御され、電源基板 1 0 1 によってスロットマシン 1 を構成する電気部品の駆動電源が生成され、各部に供給される。30

#### 【0035】

電源基板 1 0 1 には、外部から A C 1 0 0 V の電源が供給されるとともに、この A C 1 0 0 V の電源からスロットマシン 1 を構成する電気部品の駆動に必要な直流電圧が生成され、遊技制御基板 4 0 及び遊技制御基板 4 0 を介して接続された演出制御基板 9 0 に供給されるようになっている。

#### 【0036】

また、電源基板 1 0 1 には、前述したホッパーモータ 3 4 b、払出センサ 3 4 c、満タンセンサ 3 5 a、打止スイッチ 3 6 a、自動精算スイッチ 3 6 b、設定キースイッチ 3 7、電源スイッチ 3 9 が接続されている。40

#### 【0037】

遊技制御基板 4 0 には、前述した 1 枚 B E T スイッチ 5、M A X B E T スイッチ 6、スタートスイッチ 7、ストップスイッチ 8 L、8 C、8 R、精算スイッチ 1 0、リセットスイッチ 2 3、投入メダルセンサ 3 1、ドア開放検出スイッチ 2 5、リセット / 設定スイッチ 3 8、リールセンサ 3 3 L、3 3 C、3 3 R が接続されているとともに、電源基板 1 0 1 を介して前述した払出センサ 3 4 c、満タンセンサ 3 5 a、打止スイッチ 3 6 a、自動精算スイッチ 3 6 b、設定キースイッチ 3 7 が接続されており、これら接続されたスイッチ類の検出信号が入力されるようになっている。50

## 【0038】

また、遊技制御基板40には、前述したクレジット表示器11、遊技補助表示器12、ペイアウト表示器13、1~3BETLED14~16、投入要求LED17、スタート有効LED18、ウェイト中LED19、リプレイ中LED20、BETスイッチ有効LED21、左、中、右停止有効LED22L、22C、22R、設定値表示器24、流路切替ソレノイド30、リールモータ32L、32C、32Rが接続されているとともに、電源基板101を介して前述したホッパーモータ34bが接続されており、これら電気部品は、遊技制御基板40に搭載された後述のメイン制御部41の制御に基づいて駆動されるようになっている。

## 【0039】

遊技制御基板40には、メイン制御部41、制御用クロック生成回路42、乱数用クロック生成回路43、スイッチ検出回路44、モータ駆動回路45、ソレノイド駆動回路46、LED駆動回路47、電断検出回路48、リセット回路49が搭載されている。

## 【0040】

メイン制御部41は、1チップマイクロコンピュータにて構成され、後述するROM506に記憶された制御プログラムを実行して、遊技の進行に関する処理を行うとともに、遊技制御基板40に搭載された制御回路の各部を直接的または間接的に制御する。

## 【0041】

制御用クロック生成回路42は、メイン制御部41の外部にて、所定周波数の発振信号となる制御用クロックCCLKを生成する。制御用クロック生成回路42により生成された制御用クロックCCLKは、例えば図5(A)に示すようなメイン制御部41の制御用外部クロック端子EXCを介してクロック回路502に供給される。乱数用クロック生成回路43は、メイン制御部41の外部にて、制御用クロックCCLKの発振周波数とは異なる所定周波数の発振信号となる乱数用クロックRCLKを生成する。乱数用クロック生成回路43により生成された乱数用クロックRCLKは、例えば図5(A)に示すようなメイン制御部41の乱数用外部クロック端子ERCを介して乱数回路509に供給される。一例として、乱数用クロック生成回路43により生成される乱数用クロックRCLKの発振周波数は、制御用クロック生成回路42により生成される制御用クロックCCLKの発振周波数以下となるようにすれば良い。

## 【0042】

スイッチ検出回路44は、遊技制御基板40に直接または電源基板101を介して接続されたスイッチ類から入力された検出信号を取り込んでメイン制御部41に伝送する。モータ駆動回路45は、メイン制御部41から出力されたモータ駆動信号をリールモータ32L、32C、32Rに伝送する。ソレノイド駆動回路46は、メイン制御部41から出力されたソレノイド駆動信号を流路切替ソレノイド30に伝送する。LED駆動回路は、メイン制御部41から出力されたLED駆動信号を遊技制御基板40に接続された各種表示器やLEDに伝送する。電断検出回路48は、スロットマシン1に供給される電源電圧を監視し、電圧低下を検出したときに、その旨を示す電圧低下信号をメイン制御部41に対して出力する。リセット回路49は、電源投入時または電源遮断時などの電源が不安定な状態においてメイン制御部41にシステムリセット信号を与える。また、リセット回路49は、ウォッチドッグタイマ49a(図5(B)参照)を内蔵し、ウォッチドッグタイマ49aがタイムアップした場合、すなわちメイン制御部41のCPU505の動作が一定時間停止した場合においてメイン制御部41にユーザリセット信号を与える。

## 【0043】

図5(A)は、遊技制御基板40に搭載されたメイン制御部41の構成例を示している。図5に示すメイン制御部41は、1チップマイクロコンピュータであり、外部バスインターフェース501と、クロック回路502と、固有情報記憶回路503と、リセット/割込コントローラ504と、CPU505と、ROM506と、RAM507と、CTC(カウンタ/タイマサーキット)508と、乱数回路509と、PIP(パラレルインプットポート)510と、シリアル通信回路511と、アドレスデコード回路512とを備え

10

20

30

40

50

て構成される。

#### 【0044】

図6は、メイン制御部41におけるアドレスマップの一例を示している。図6に示すように、アドレス0000H～アドレス1FFFHの領域は、ROM506に割り当てられ、ユーザプログラムエリアとプログラム管理エリアとを含んでいる。図7(A)は、ROM506におけるプログラム管理エリアの主要部分について、用途や内容の一例を示している。アドレス2000H～アドレス20FFFHの領域は、メイン制御部41の内蔵レジスタに割り当てられる内蔵レジスタエリアである。図7(B)は、内蔵レジスタエリアの主要部分について、用途や内容の一例を示している。アドレス7E00H～アドレス7FFFHの領域は、RAM507に割り当てられたワークエリアであり、I/Oマップやメモリマップに割り付けることができる。アドレスFDD0H～アドレスFDFFBHの領域は、アドレスデコード回路512に割り当てられるXCSデコードエリアである。  
10

#### 【0045】

プログラム管理エリアは、CPU505がユーザプログラムを実行するために必要な情報を格納する記憶領域である。図7(A)に示すように、プログラム管理エリアには、ヘッダKHDR、機能設定KFC5、第1乱数初期設定KRS1、第2乱数初期設定KRS2、割込初期設定KVIS、セキュリティ時間設定KSESなどが、含まれている。

#### 【0046】

プログラム管理エリアに記憶されるヘッダKHDRは、メイン制御部41における内部データの読み出しが示す。図8(A)は、ヘッダKHDRにおける設定データと動作との対応関係を例示している。ここで、メイン制御部41では、ROM読み出しが防止機能と、バス出力マスク機能とを設定可能である。ROM読み出しが防止機能は、メイン制御部41が備えるROM506の記憶データについて、読み出しが許可又は禁止する機能であり、読み出しが禁止に設定された状態では、ROM506の記憶データを読み出すことができない。バス出力マスク機能は、外部バスインターフェース501に接続された外部装置からメイン制御部41の内部データに対する読み出しがあった場合に、外部バスインターフェース501におけるアドレスバス出力、データバス出力及び制御信号出力にマスクをかけることにより、外部装置から内部データの読み出しを不能にする機能である。図8(A)に示すように、ヘッダKHDRの設定データに対応して、ROM読み出しが防止機能やバス出力マスク機能の動作組合せが異なるように設定される。図8(A)に示す設定データのうち、ROM読み出しが許可されるとともに、バス出力マスクが有効となる設定データは、バス出力マスク有効データともいう。また、ROM読み出しが禁止されるとともに、バス出力マスクが有効となる設定データ(全て「00H」)は、ROM読み出しが禁止データともいう。ROM読み出しが許可されるとともに、バス出力マスクが無効となる設定データは、バス出力マスク無効データともいう。  
20  
30

#### 【0047】

プログラム管理エリアに記憶される機能設定KFC5は、メイン制御部41におけるウォッチドッグタイマの動作設定や、各種機能兼用端子の使用設定を示す。図8(B)は、機能設定KFC5における設定内容の一例を示している。

#### 【0048】

機能設定KFC5のビット番号[7-5]は、例えばリセット／割込コントローラ504における割込要因として設定可能なウォッチドッグタイマの動作許可／禁止や、許可した場合の周期を示している。本実施例では、外部のウォッチドッグタイマを用いるため、ウォッチドッグタイマを動作禁止に設定している。機能設定KFC5のビット番号[4]は、メイン制御部41における所定の機能兼用端子(第1兼用端子)を、シリアル通信回路511が使用する第2チャネル送信端子TXBとするか、アドレスデコード回路512が使用するチップセレクト出力端子XCS13とするかを指定するTXB端子設定である。図8(B)に示す例において、機能設定KFC5のビット番号[4]におけるビット値が“0”であれば、第1兼用端子がシリアル通信回路511での第2チャネル送信に使用される第2チャネル送信端子TXBの設定となる。これに対して、そのビット値が“1”  
40  
50

であれば、第1兼用端子がアドレスコード回路512で使用されるチップセレクト出力端子XCS13の設定となる。本実施例では、機能設定KFC5のビット番号[4]を“0”として、第1兼用端子を第2チャネル送信端子TXBに設定することにより、演出制御基板90との間でのシリアル通信を可能にする。

#### 【0049】

機能設定KFC5のビット番号[3]は、メイン制御部41における所定の機能兼用端子(第2兼用端子)を、シリアル通信回路511が使用する第1チャネル送信端子TXAとするか、アドレスコード回路512が使用するチップセレクト出力端子XCS12とするかを示すTXA端子設定である。図8(B)に示す例において、機能設定KFC5のビット番号[3]におけるビット値が“0”であれば、第2兼用端子がシリアル通信回路511での第1チャネル送信に使用される第1チャネル送信端子TXAの設定となる。これに対して、そのビット値が“1”であれば、第2兼用端子がアドレスコード回路512で使用されるチップセレクト出力端子XCS12の設定となる。本実施例では、機能設定KFC5のビット番号[3]を“0”として、第2兼用端子を第1チャネル送信端子TXAに設定しているが、第1チャンネル受信は未使用とされている。10

#### 【0050】

機能設定KFC5のビット番号[2]は、メイン制御部41における所定の機能兼用端子(第3兼用端子)を、シリアル通信回路511が使用する第1チャネル受信端子RXAとするか、PIP510が使用する入力ポートP5とするかを示すRXA端子設定である。図8(B)に示す例において、機能設定KFC5のビット番号[2]におけるビット値が“0”であれば、第3兼用端子がシリアル通信回路511での第1チャネル受信に使用される第1チャネル受信端子RXAの設定となる。これに対して、そのビット値が“1”であれば、第3兼用端子がPIP510で使用される入力ポートP5の設定となる。本実施例では、機能設定KFC5のビット番号[2]を“0”として、第3兼用端子を第1チャネル受信端子RXAに設定しているが、第1チャンネル受信は未使用とされている。20

#### 【0051】

機能設定KFC5のビット番号[1]は、メイン制御部41における所定の機能兼用端子(第4兼用端子)を、CPU505等に接続される外部ノンマスカブル割込端子XNM1とするか、PIP510が使用する入力ポートP4とするかを示すNMI接続設定である。図8(B)に示す例において、機能設定KFC5のビット番号[1]におけるビット値が“0”であれば、第4兼用端子がCPU505等に接続される外部ノンマスカブル割込端子XNM1の設定となる(CPU接続)。これに対して、そのビット値が“1”であれば、第4兼用端子がPIP510で使用される入力ポートP4の設定となる(CPU非接続)。本実施例では、機能設定KFC5のビット番号[1]を“1”として、第4兼用端子をPIP510で使用される入力ポートP4に設定しているが、入力ポートP4は未使用とされている。30

#### 【0052】

機能設定KFC5のビット番号[0]は、メイン制御部41における所定の機能兼用端子(第5兼用端子)を、CPU505等に接続される外部マスカブル割込端子XINTとするか、PIP510が使用する入力ポートP3とするかを示すXINT接続設定である。図8(B)に示す例において、機能設定KFC5のビット番号[0]におけるビット値が“0”であれば、第5兼用端子がCPU505等に接続される外部マスカブル割込端子XINTの設定となる(CPU接続)。これに対して、そのビット値が“1”であれば、第5兼用端子がPIP510で使用される入力ポートP3の設定となる(CPU非接続)。本実施例では、機能設定KFC5のビット番号[0]を“0”として、第4兼用端子をCPU505等に接続される外部マスカブル割込端子XINTに設定することにより、電断検出回路48からの電圧低下信号が入力される。40

#### 【0053】

プログラム管理エリアに記憶される第1乱数初期設定KRS1及び第2乱数初期設定KRS2は、乱数回路509の初期設定を示す。図9(A)は、第1乱数初期設定KRS150

における設定内容の一例を示している。図9( B )は、第2乱数初期設定KRS2における設定内容の一例を示している。

#### 【0054】

第1乱数初期設定KRS1のビット番号[3]は、乱数回路509を使用するか否かを示す乱数回路使用設定である。図9( A )に示す例において、第1乱数初期設定KRS1のビット番号[3]におけるビット値が“0”であれば、乱数回路509を使用しない設定となる一方(未使用)、“1”であれば、乱数回路509を使用する設定となる(使用)。本実施例では、第1乱数初期設定KRS1のビット番号[3]を“1”として、乱数回路509を使用可能に設定する。

#### 【0055】

第1乱数初期設定KRS1のビット番号[2]は、乱数回路509における乱数値となる数値データの更新に用いられる乱数更新クロックRGK(図12参照)を、内部システムクロックSCLKとするか、乱数用クロックRCLKの2分周とするかを示す乱数更新クロック設定である。図9( A )に示す例において、第1乱数初期設定KRS1のビット番号[2]におけるビット値が“0”であれば、内部システムクロックSCLKを乱数更新クロックRGKに用いる設定となる一方、“1”であれば、乱数用クロックRCLKを2分周して乱数更新クロックRGKに用いる設定となる。本実施例では、第1乱数初期設定KRS1のビット番号[2]を“1”として、乱数用クロックRCLKを2分周して乱数更新クロックRGKに用いる設定とする。

#### 【0056】

第1乱数初期設定KRS1のビット番号[1-0]は、乱数回路509における乱数更新規則を変更するか否かや、変更する場合における変更方式を示す乱数更新規則設定である。図9( A )に示す例において、第1乱数初期設定KRS1のビット番号[1-0]におけるビット値が“00”であれば、乱数更新規則を変更しない設定となり、“01”であれば、2周目以降にて乱数更新規則をソフトウェアにより変更する設定となり、“10”であれば、2周目以降にて乱数更新規則を自動で変更する設定となる。

#### 【0057】

第2乱数初期設定KRS2のビット番号[3-2]は、固定のビット値“00”が設定される。尚、図9( B )における「00B」の“B”は2進数表示であることを示す。第2乱数初期設定KRS2のビット番号[1-0]は、乱数回路509における乱数値となる数値データでのスタート値に関する設定を示す。図9( B )に示す例において、第2乱数初期設定KRS2のビット番号[1]におけるビット値が“0”であれば、スタート値が所定のデフォルト値0001Hに設定される一方、“1”であるときには、メイン制御部41毎に付与された固有の識別情報であるIDナンバーに基づく値がスタート値に設定される。また、図9( B )に示す例では、第2乱数初期設定KRS2のビット番号[0]におけるビット値が“0”であれば、システムリセット毎にスタート値を変更しない設定となる一方、“1”であるときには、システムリセット毎にスタート値を変更する設定となる。尚、スタート値をIDナンバーに基づく値に設定する場合には、IDナンバーに所定のスクランブル処理を施す演算や、IDナンバーを用いた加算・減算・乗算・除算などの演算の一部又は全部を実行して、算出された値をスタート値に用いるようにすれば良い。また、スタート値をシステムリセット毎に変更する場合には、例えばメイン制御部41に内蔵されたフリーランカウンタのカウント値を、システムリセットの発生時にメイン制御部41が備える所定の内蔵レジスタ(乱数スタート値用レジスタ)に格納する。そして、初期設定時に乱数スタート値用レジスタの格納値をそのまま用いること、或いは、その格納値を所定の演算関数(例えばハッシュ関数)に代入して得られた値を用いることなどにより、スタート値がランダムに決定されれば良い。フリーランカウンタは、遊技制御基板40におけるバックアップ箇所と共にバックアップ電源を用いてバックアップされるものであれば良い。或いは、フリーランカウンタは、RAM507におけるバックアップ領域などに用いられるバックアップ電源とは別個に設けられた電源によりバックアップされても良い。こうして、フリーランカウンタがバックアップ電源によってバックアップさ

10

20

30

40

50

れることで、電力供給が停止した場合でも、所定期間はフリーランカウンタにおけるカウント値が保存されることになる。

#### 【0058】

尚、乱数回路509にて乱数値となる数値データを生成するための回路が2系統（第1及び第2チャネル対応）設けられる場合には、図9（A）及び（B）に示す第1乱数初期設定KRS1のビット番号[3-0]と第2乱数初期設定KRS2のビット番号[3-0]とを、第1チャネルにおける初期設定を示すものとして使用する。その一方で、第1乱数初期設定KRS1のビット番号[7-4]や第2乱数初期設定KRS2のビット番号[7-4]を（図9（A）及び（B）では省略）、第2チャネルにおける初期設定を示すものとして使用すれば良い。

10

#### 【0059】

プログラム管理エリアに記憶される割込初期設定KVISは、メイン制御部41にて発生するマスカブル割込の取扱いに関する初期設定を示す。図9（C）は、割込初期設定KVISにおける設定内容の一例を示している。

#### 【0060】

割込初期設定KVISのビット番号[7-4]では、割込ベクタの上位4ビットを設定する。割込初期設定KVISのビット番号[3-0]では、マスカブル割込要因の優先度の組合せを設定する。図9（C）に示す例において、割込初期設定KVISのビット番号[3-0]により「00H」～「02H」及び「06H」のいずれかが指定されれば、CTC508からのマスカブル割込要因を最優先とする優先度の組合せが設定される。これに対して、「03H」及び「07H」のいずれかが指定されれば、乱数回路509からのマスカブル割込要因を最優先とする優先度の組合せが設定される。また、「04H」及び「05H」のいずれかが指定されれば、シリアル通信回路511からのマスカブル割込要因を最優先とする優先度の組合せが設定される。尚、同一回路からのマスカブル割込要因を最優先とする優先度の組合せでも、指定値が異なる場合には、最優先となるマスカブル割込要因の種類や第2順位以下における優先度の組合せなどが異なっている。

20

#### 【0061】

プログラム管理エリアに記憶されるセキュリティ時間設定KSESは、乱数用クロックRCLKの周波数を監視する場合に異常を検知する周波数や、メイン制御部41の動作開始時などに移行するセキュリティモードの時間（セキュリティ時間）に関する設定を示す。ここで、メイン制御部41の動作モードがセキュリティモードであるときには、所定のセキュリティチェック処理が実行されて、ROM506の記憶内容が変更されたか否かが検査される。図10（A）は、セキュリティ時間設定KSESにおける設定内容の一例を示している。

30

#### 【0062】

セキュリティ時間設定KSESのビット番号[7-6]は、乱数用クロックRCLKの周波数を監視する場合に異常が検出される周波数を示す乱数用クロック異常検出設定である。図10（B）は、セキュリティ時間設定KSESのビット番号[7-6]における設定内容の一例を示している。セキュリティ時間設定KSESのビット番号[7-6]は、内部システムクロックSCLKの周波数に基づき、乱数用クロックRCLKの周波数が異常と検知される基準値（判定値）を指定する。セキュリティ時間設定KSESのビット番号「5」は、固定のビット値“0”が設定される。

40

#### 【0063】

セキュリティ時間設定KSESのビット番号[4-3]は、セキュリティ時間をシステムリセット毎にランダムな時間分延長する場合の時間設定を示す。図10（C）は、セキュリティ時間設定KSESのビット番号[4-3]における設定内容の一例を示している。図10（C）に示す例において、セキュリティ時間設定KSESのビット番号[4-3]におけるビット値が“00”であれば、ランダムな時間延長を行わない設定となる。これに対して、そのビット値が“01”であればショートモードの設定となり、“10”であればロングモードの設定となる。ここで、ショートモードやロングモードが指定された

50

場合には、例えばメイン制御部41に内蔵されたフリーランカウンタのカウント値を、システムリセットの発生時にメイン制御部41が備える所定の内蔵レジスタ（可変セキュリティ時間用レジスタ）に格納する。そして、初期設定時に可変セキュリティ時間用レジスタの格納値をそのまま用いること、或いは、その格納値を所定の演算関数（例えばハッシュ関数）に代入して得られた値を用いることなどにより、セキュリティ時間を延長する際の延長時間がランダムに決定されれば良い。一例として、内部システムクロックSCLKの周波数が6.0MHzである場合には、ショートモードにおいて0~680μs（マイクロ秒）の範囲で延長時間がランダムに決定され、ロングモードにおいて0~348,160μsの範囲で延長時間がランダムに決定される。また、他の一例として、内部システムクロックSCLKの周波数が10.0MHzである場合には、ショートモードにおいて0~408μsの範囲で延長時間がランダムに決定され、ロングモードにおいて0~208,896μsの範囲で延長時間がランダムに決定される。尚、セキュリティ時間を延長する際の延長時間をシステムリセット毎にランダムに決定するために用いられるフリーランカウンタは、乱数回路509にて生成される乱数のスタート値をシステムリセット毎にランダムに決定するために用いられるフリーランカウンタと、同一のカウンタであっても良いし、別個に設けられたカウンタであっても良い。10

#### 【0064】

セキュリティ時間設定KSESのビット番号[2-0]は、セキュリティ時間を固定時間に加えて予め選択可能な複数の延長時間のいずれかとする場合の時間設定を示す。図10(D)は、セキュリティ時間設定KSESのビット番号[2-0]における設定内容の一例を示している。図10(D)に示す例において、セキュリティ時間設定KSESのビット番号[2-0]におけるビット値が“000”であれば、固定時間に加える延長時間がなく延長しない設定となる。これに対して、そのビット値が“000”以外の値であれば、内部システムクロックSCLKの周期T<sub>SCLK</sub>を用いて定められる複数の延長時間のいずれかに設定される。この場合には、指定されたビット値に応じて、異なる延長時間の設定となる。一例として、内部システムクロックSCLKの周波数が6.0MHzである場合に、セキュリティ時間設定KSESのビット番号[2-0]におけるビット値が“001”であれば、固定時間に加えて約699.1ms(ミリ秒)の延長時間が設定される。また、他の一例として、内部システムクロックSCLKの周波数が10.0MHzである場合に、セキュリティ時間設定KSESのビット番号[2-0]におけるビット値が“001”であれば、固定時間に加えて約419.4msの延長時間が設定される。2030

#### 【0065】

また、セキュリティ時間設定KSESのビット番号[4-3]におけるビット値によりショートモード又はロングモードを設定するとともに、セキュリティ時間設定KSESのビット番号[2-0]におけるビット値を“000”以外とすることにより固定時間に加える延長時間を設定することもできる。この場合には、ビット番号[2-0]におけるビット値に対応した延長時間と、ビット番号[4-3]におけるビット値に基づいてランダムに決定された延長時間との双方が、固定時間に加算されて、メイン制御部41がセキュリティモードとなるセキュリティ時間が決定されることになる。

#### 【0066】

図5(A)に示すメイン制御部41が備える外部バスインターフェース501は、メイン制御部41を構成するチップの外部バスと内部バスとのインターフェース機能や、アドレスバス、データバス及び各制御信号の方向制御機能などを有するバスインターフェースである。例えば、外部バスインターフェース501は、メイン制御部41に外付けされた外部メモリや外部入出力装置などに接続され、これらの外部装置との間でアドレス信号やデータ信号、各種の制御信号などを送受信するものであれば良い。この実施の形態において、外部バスインターフェース501には、内部リソースアクセス制御回路501Aが含まれている。40

#### 【0067】

内部リソースアクセス制御回路501Aは、外部バスインターフェース501を介した外50

部装置からメイン制御部41の内部データに対するアクセスを制御して、例えばROM506に記憶されたゲーム制御用プログラムや固定データといった、内部データの不適切な外部読出を制限するための回路である。ここで、外部バスインターフェース501には、例えばインサーキットエミュレータ(ICE)といった回路解析装置が、外部装置として接続されることがある。

#### 【0068】

一例として、ROM506のプログラム管理エリアに記憶されたヘッダKHDRの内容に応じて、ROM506における記憶データの読み出しを禁止するか許可するかを切り替えられるようにする。例えば、ヘッダKHDRがバス出力マスク無効データとなっている場合には、外部装置によるROM506の読み出しを可能にして、内部データの外部読出を許可する。これに対して、ヘッダKHDRがバス出力マスク有効データとなっている場合には、例えば外部バスインターフェース501におけるアドレスバス出力、データバス出力及び制御信号出力にマスクをかけることなどにより、外部装置からROM506の読み出しを不能にして、内部データの外部読出を禁止する。この場合、外部バスインターフェース501に接続された外部装置から内部データの読み出しが要求されたときには、予め定められた固定値を出力することで、外部装置からは内部データを読み出すことができないようにする。また、ヘッダKHDRがROM読出禁止データとなっている場合には、ROM506自身を読み出不能として、ROM506における記憶データの読み出しを防止しても良い。そして、例えば製造段階のROMでは、ヘッダKHDRをROM読出禁止データとして、ROM自体を読み出不能としておき、開発用ROMとするのであればバス出力マスク無効データをヘッダKHDRに書き込むことで、外部装置による内部データの検証を可能にする。これに対して、量産用ROMとするのであればバス出力マスク有効データをヘッダKHDRに書き込むことで、CPU505などによるメイン制御部41の内部におけるROM506の読み出しは可能とする一方で、外部装置によるROM506の読み出しはできないようにすれば良い。

10

20

#### 【0069】

他の一例として、内部リソースアクセス制御回路501Aは、ROM506における記憶データの全部又は一部といった、メイン制御部41の内部データの読み出しが、外部バスインターフェース501に接続された外部装置から要求されたことを検出する。この読み出しが検出したときに、内部リソースアクセス制御回路501Aは、メイン制御部41の内部データの読み出しを許可するか否かの判定を行う。例えば、ROM506における記憶データの全部又は一部に暗号化処理が施されているものとする。この場合、内部リソースアクセス制御回路501Aは、外部装置からの読み出しがROM506に記憶された暗号化処理プログラムや鍵データ等に対する読み出しだければ、この読み出しが拒否して、メイン制御部41の内部データの読み出しを禁止する。外部バスインターフェース501では、ROM506の記憶データが出力される出力ポートと、内部バスとの間にスイッチ素子を設け、内部リソースアクセス制御回路501Aが内部データの読み出しを禁止した場合には、このスイッチ素子をオフ状態とすると制御すれば良い。このように、内部リソースアクセス制御回路501Aは、外部装置からの読み出しが所定の内部データ（例えばROM506の所定領域）の読み出しが要求するものであるか否かに応じて、内部データの読み出しを禁止するか許可するかを決定するようにしても良い。

30

40

#### 【0070】

或いは、内部リソースアクセス制御回路501Aは、内部データの読み出しが検出されたときに、所定の認証コードが外部装置から入力されたか否かを判定しても良い。この場合には、例えば内部リソースアクセス制御回路501Aの内部或いはROM506の所定領域に、認証コードとなる所定のコードパターンが予め記憶されていれば良い。そして、外部装置から認証コードが入力されたときには、この認証コードを内部記憶された認証コードと比較して、一致すれば読み出しが受容して、メイン制御部41の内部データの読み出しを許可する。これに対して、外部装置から入力された認証コードが内部記憶された認証コードと一致しない場合には、読み出しが拒否して、メイン制御部41の内部データの読み出しが拒否される。

50

み出しを禁止する。このように、内部リソースアクセス制御回路 501A は、外部装置から入力された認証コードが内部記憶された認証コードと一致するか否かに応じて、内部データの読み出しを禁止するか許可するかを決定するようにしても良い。これにより、検査機関などが予め知得した正しい認証コードを用いて、メイン制御部 41 の内部データを損なうことなく読み出すことができ、内部データの正当性を適切に検査することなどが可能になる。

#### 【0071】

さらに他の一例として、内部リソースアクセス制御回路 501A に読出禁止フラグを設け、読出禁止フラグがオン状態であれば外部装置による ROM506 の読み出しを禁止する。その一方で、読出禁止フラグがオフ状態であるときには、外部装置による ROM506 の読み出しが許可される。ここで、読出禁止フラグは、初期状態ではオフ状態であるが、読出禁止フラグを一旦オン状態とした後には、読出禁止フラグをクリアしてオフ状態に復帰させることができないように構成されれば良い。すなわち、読出禁止フラグはオフ状態からオン状態に不可逆的に変更することが可能とされている。例えば、内部リソースアクセス制御回路 501A には、読出禁止フラグをクリアしてオフ状態とする機能が設けられておらず、どのような命令によっても読出禁止フラグをクリアすることができないように設定されれば良い。そして、内部リソースアクセス制御回路 501A は、外部装置から ROM506 における記憶データといったメイン制御部 41 の内部データの読み出しが要求されたときに、読出禁止フラグがオンであるか否かを判定する。このとき、読出禁止フラグがオンであれば、外部装置からの読出要求を拒否して、メイン制御部 41 の内部データの読み出しを禁止する。他方、読出禁止フラグがオフであれば、外部装置からの読出要求を受容して、メイン制御部 41 の内部データの読み出しを許可にする。このような構成であれば、ゲーム制御用のプログラムを作成して ROM506 に格納する提供者においては、読出禁止フラグがオフとなっている状態でデバッグの終了したプログラムを ROM506 から外部装置に読み込むことができる。そして、デバッグの作業が終了した後に出荷する際には、読出禁止フラグをオン状態にセットすることにより、それ以後は ROM506 の外部読出を制限することができ、スロットマシン 1 の使用者などによる ROM506 の読出を防止することができる。このように、内部リソースアクセス制御回路 501A は、読出禁止フラグといった内部フラグがオフであるかオンであるかに応じて、内部データの読み出しを禁止するか許可するかを決定するようにしても良い。

#### 【0072】

尚、読出禁止フラグを不可逆に設定するのではなく、オン状態からオフ状態に変更することも可能とする一方で、読出禁止フラグをオン状態からオフ状態に変更して内部データの読み出しが許可されるときには、ROM506 の記憶データを消去（例えばフラッシュ消去など）することにより、ROM506 の外部読出を制限するようにしても良い。

#### 【0073】

メイン制御部 41 が備えるクロック回路 502 は、例えば制御用外部クロック端子 EXC に入力される発振信号を 2 分周することなどにより、内部システムクロック SCLK を生成する回路である。本実施例では、制御用外部クロック端子 EXC に制御用クロック生成回路 42 が生成した制御用クロック CCLK が入力される。クロック回路 502 により生成された内部システムクロック SCLK は、例えば CPU505 といった、メイン制御部 41 において遊技の進行を制御する各種回路に供給される。また、内部システムクロック SCLK は、乱数回路 509 にも供給され、乱数用クロック生成回路 43 から供給される乱数用クロック RCLK の周波数を監視するために用いられる。さらに、内部システムクロック SCLK は、クロック回路 502 に接続されたシステムクロック出力端子 CLKO から、メイン制御部 41 の外部へと出力されても良い。尚、内部システムクロック SCLK は、メイン制御部 41 の外部へは出力されないことが望ましい。このように、内部システムクロック SCLK の外部出力を制限することにより、メイン制御部 41 の内部回路（CPU505 など）の動作周期を外部から特定することが困難になり、乱数値となる数値データをソフトウェアにより更新する場合に、乱数値の更新周期が外部から特定されて

10

20

30

40

50

しまうことを防止できる。

**【0074】**

メイン制御部41が備える固有情報記憶回路503は、例えばメイン制御部41の内部情報となる複数種類の固有情報を記憶する回路である。一例として、固有情報記憶回路503は、ROMコード、チップ個別ナンバー、IDナンバーといった3種類の固有情報を記憶する。ROM506コードは、ROM506の所定領域における記憶データから生成される4バイトの数値であり、生成方法の異なる4つの数値が準備されれば良い。チップ個別ナンバーは、メイン制御部41の製造時に付与される4バイトの番号であり、メイン制御部41を構成するチップ毎に異なる数値を示している。IDナンバーは、メイン制御部41の製造時に付与される8バイトの番号であり、メイン制御部41を構成するチップ毎に異なる数値を示している。ここで、チップ個別ナンバーはユーザプログラムから読み取ることができる一方、IDナンバーはユーザプログラムから読み取ることができないよう10に設定されれば良い。尚、固有情報記憶回路503は、例えばROM506の所定領域を用いることなどにより、ROM506に含まれるようにも良い。或いは、固有情報記憶回路503は、例えばCPU505の内蔵レジスタを用いることなどにより、CPU505に含まれるようにも良い。

**【0075】**

メイン制御部41が備えるリセット／割込コントローラ504は、メイン制御部41の内部や外部にて発生する各種リセット、割込要求を制御するためのものである。リセット／割込コントローラ504が制御するリセットには、システムリセットとユーザリセットが含まれている。システムリセットは、外部システムリセット端子XSRSTに一定の期間間にわたりローレベル信号（システムリセット信号）が入力されたときに発生するリセットである。ユーザリセットは、外部ユーザリセット端子XURSTに一定の期間間にわたりローレベルの信号（ユーザリセット信号）が入力されたとき、または内蔵ウォッチドッグタイマ（WDT）のタイムアウト信号が発生したことや、指定エリア外走行禁止（IAT）が発生したことなど、所定の要因により発生するリセットである。尚、本実施例では前述のように内蔵ウォッチドッグタイマを使用せずにリセット回路49に搭載されたウォッチドッグタイマ（WDT）を用いているため、外部ユーザリセット端子XURSTにユーザリセット信号が入力されるか、指定エリア外走行禁止（IAT）が発生することでユーザリセットが発生することとなる。  
30

**【0076】**

本実施例では、図5(B)に示すように、ウォッチドッグタイマ49aを内蔵するリセット回路49を遊技制御基板40に搭載している。リセット回路49は、スロットマシン1への供給電源が安定電圧となり一定時間が経過するまでシステムリセット信号をメイン制御部41に対して出力する。また、ウォッチドッグタイマ49aがタイムアウトした場合には、ユーザリセット信号をメイン制御部41に対して出力する。

**【0077】**

図5(B)に示すように、遊技制御基板40では、LED駆動回路47からクレジット表示器11へ接続される信号線のうち、クレジット表示器11を構成する複数のセグメントの駆動信号のうち下1桁Bセグメント信号、下1桁Cセグメント信号、上1桁Bセグメント信号、上1桁Cセグメント信号の信号線が分岐し、or回路を介してリセット回路49のウォッチドッグタイマクリア信号端子に接続されている。  
40

**【0078】**

本実施例では、メイン制御部41が、クレジット表示器11の下1桁Bセグメント、下1桁Cセグメント、上1桁Bセグメント、上1桁Cセグメントのいずれかのセグメントを必ずダイナミック点灯させる制御を行っており、これらのセグメントをダイナミック点灯させるため、メイン制御部41が正常に動作していれば、これら4つのセグメントのいずれかの駆動信号が定期的に出力されるはずであり、これら4つのセグメントのいずれかの駆動信号が定期的に出力されているか否かを監視することにより、メイン制御部41が正常に動作しているか否かを判定することが可能となる。  
50

**【0079】**

そして、これら4つのセグメントの駆動信号をor回路を介して1つにまとめられた信号がリセット回路49のウォッチドッグタイマクリア信号端子に入力され、ウォッチドッグタイマ49aがクリアされるようになっており、上記4つのセグメントの駆動信号の出力が停止して、ウォッチドッグタイマ49aがクリアされず、タイムアップすることで、ユーザリセット信号がメイン制御部41に対して出力されるようになっている。

**【0080】**

このように本実施例では、定期的に駆動信号が与えられるLEDのセグメント信号を分岐してウォッチドッグタイマ49aをクリアするようになっており、メイン制御部41のCPU505が個別にウォッチドッグタイマ49aをクリアする処理を行うことなく、メイン制御部41aが正常に動作しているか否かを監視することが可能となることから好ましいが、メイン制御部41からリセット回路49のウォッチドッグタイマクリア信号端子に個別のクリア信号を入力することでウォッチドッグタイマ49aをクリアするようにしても良い。10

**【0081】**

また、本実施例では、メイン制御部41の外部に設けられたリセット回路49にウォッチドッグタイマ49aを搭載する構成であるが、メイン制御部41に内蔵されたウォッチドッグタイマを用いてメイン制御部41の動作を監視するようにしても良い。

**【0082】**

リセットノ割込コントローラ504が制御する割込には、ノンマスカブル割込NMIとマスカブル割込INTが含まれている。ノンマスカブル割込NMIは、CPU505の割込禁止状態でも無条件に受け付けられる割込であり、外部ノンマスカブル割込端子XNM<sub>I</sub>(入力ポートP4と兼用)に一定の期間にわたりローレベル信号が入力されたときに発生する割込である。マスカブル割込INTは、CPU505の設定命令により、割込要求の受け付けを許可/禁止できる割込であり、優先順位設定による多重割込の実行が可能である。マスカブル割込INTの要因としては、外部マスカブル割込端子XINT(入力ポートP3と兼用)に一定の期間にわたりローレベル信号が入力されたこと、CTC508に含まれるタイマ回路にてタイムアウトが発生したこと、シリアル通信回路511にてデータ送信による割込要因が発生したこと、乱数回路509にて乱数値となる数値データの取込による割込要因が発生したことなど、複数種類の割込要因が予め定められていれば良い。2030

**【0083】**

リセットノ割込コントローラ504は、図7(B)に示すようなメイン制御部41が備える内蔵レジスタのうち、割込マスクレジスタIMR(アドレス2028H)、割込待ちモニタレジスタIRR(アドレス2029H)、割込中モニタレジスタISR(アドレス202AH)、内部情報レジスタCIF(アドレス208CH)などを用いて、割込の制御やリセットの管理を行う。割込マスクレジスタIMRは、互いに異なる複数の要因によるマスカブル割込INTのうち、使用するものと使用しないものを設定するレジスタである。割込待ちモニタレジスタIRRは、割込初期設定KIIISにより設定されたマスカブル割込要因のそれぞれについて、マスカブル割込要求信号の発生状態を確認するレジスタである。割込中モニタレジスタISRは、割込初期設定KIIISにより設定されたマスカブル割込要因のそれぞれについて、マスカブル割込要求信号の処理状態を確認するレジスタである。内部情報レジスタCIFは、直前に発生したリセット要因を管理したり、乱数用クロックRCLKの周波数異常を記録したりするためのレジスタである。40

**【0084】**

図11(A)は、内部情報レジスタCIFの構成例を示している。図11(B)は、内部情報レジスタCIFに格納される内部情報データの各ビットにおける設定内容の一例を示している。内部情報レジスタCIFのビット番号[4]に格納される内部情報データCIF4は、乱数用クロックRCLKにおける周波数異常の有無を示す乱数用クロック異常指示である。図11(B)に示す例では、乱数用クロックRCLKの周波数異常が検知さ50

れないときに、内部情報データC I F 4のビット値が“0”となる一方、周波数異常が検知されたときには、そのビット値が“1”となる。内部情報レジスタC I Fのビット番号[2]に格納される内部情報データC I F 2は、直前に発生したリセット要因がシステムリセットであるか否かを示すシステムリセット指示である。図11(B)に示す例では、直前のリセット要因がシステムリセットではないときに(システムリセット未発生)、内部情報データC I F 2のビット値が“0”となる一方、システムリセットであるときには(システムリセット発生)、そのビット値が“1”となる。

#### 【0085】

内部情報レジスタC I Fのビット番号[1]に格納される内部情報データC I F 1は、直前に発生したリセット要因がウォッチドッグタイマ(WDT)のタイムアウトによるユーザリセットであるか否かを示すWDTタイムアウト指示である。図11(B)に示す例では、直前のリセット要因がウォッチドッグタイマのタイムアウトによるユーザリセットではないときに(タイムアウト未発生)、内部情報データC I F 1のビット値が“0”となる一方、ウォッチドッグタイマのタイムアウトによるユーザリセットであるときに(タイムアウト発生)、そのビット値が“1”となる。内部情報レジスタC I Fのビット番号[0]に格納される内部情報データC I F 0は、直前に発生したリセット要因が指定エリア外走行禁止(IAT)によるユーザリセットであるか否かを示すIAT発生指示である。図11(B)に示す例では、直前のリセット要因が指定エリア外走行の発生によるユーザリセットではないときに(IAT発生なし)、内部情報データC I F 0のビット値が“0”となる一方、指定エリア外走行の発生によるユーザリセットであるときに(IAT発生あり)、そのビット値が“1”となる。

#### 【0086】

メイン制御部41が備えるCPU505は、ROM506から読み出したプログラムを実行することにより、スロットマシン1におけるゲームの進行を制御するための処理などを実行する。このときには、CPU505がROM506から固定データを読み出す固定データ読出動作や、CPU505がRAM507に各種の変動データを書き込んで一時記憶させる変動データ書込動作、CPU505がRAM507に一時記憶されている各種の変動データを読み出す変動データ読出動作、CPU505が外部バスインターフェース501やPIP510などを介してメイン制御部41の外部から各種信号の入力を受け付ける受信動作、CPU505が外部バスインターフェース501やシリアル通信回路511などを介してメイン制御部41の外部へと各種信号を出力する送信動作等も行われる。

#### 【0087】

このように、メイン制御部41では、CPU505がROM506に格納されているプログラムに従って制御を実行するので、以下、メイン制御部41(又はCPU505)が実行する(又は処理を行う)ということは、具体的には、CPU505がプログラムに従って制御を実行することである。このことは、遊技制御基板40以外の他の基板に搭載されているマイクロコンピュータについても同様である。

#### 【0088】

メイン制御部41が備えるROM506には、ゲーム制御用のユーザプログラムや固定データ等が記憶されている。また、ROM506には、セキュリティチェックプログラム506Aが記憶されている。CPU505は、スロットマシン1の電源投入やシステムリセットの発生に応じてメイン制御部41がセキュリティモードに移行したときに、ROM506に記憶されたセキュリティチェックプログラム506Aを読み出し、ROM506の記憶内容が変更されたか否かを検査するセキュリティチェック処理を実行する。尚、セキュリティチェックプログラム506Aは、ROM506とは異なる内蔵メモリに記憶されても良い。また、セキュリティチェックプログラム506Aは、例えば外部バスインターフェース501を介してメイン制御部41に外付けされた外部メモリの記憶内容を検査するセキュリティチェック処理に対応したものであっても良い。

#### 【0089】

メイン制御部41が備えるRAM507は、ゲーム制御用のワークエリアを提供する。

10

20

30

40

50

ここで、RAM507の少なくとも一部は、バックアップ電源によってバックアップされているバックアップRAMであれば良い。すなわち、スロットマシンへの電力供給が停止しても、所定期間はRAM507の少なくとも一部の内容が保存される。尚、本実施例では、RAM507の全ての領域がバックアップRAMとされており、スロットマシンへの電力供給が停止しても、所定期間はRAM507の全ての内容が保存される。

#### 【0090】

メイン制御部41が備えるCTC508は、例えば8ビットのプログラマブルタイマを3チャネル(PTC0-PTC2)内蔵して構成され、リアルタイム割込の発生や時間計測を可能とするタイマ回路を含んでいる。各プログラマブルタイマPTC0-PTC2は、内部システムクロックSCLKに基づいて生成されたカウントクロックの信号変化(例えばハイレベルからローレベルへと変化する立ち下がりタイミング)などに応じて、タイマ値が更新されるものであれば良い。また、CTC508は、例えば8ビットのプログラマブルカウンタを4チャネル(PCC0-PCC3)内蔵しても良い。各プログラマブルカウンタPCC0-PCC3は、内部システムクロックSCLKの信号変化、或いは、プログラマブルカウンタPCC0-PCC3のいずれかにおけるタイムアウトの発生などに応じて、カウント値が更新されるものであれば良い。CTC508は、セキュリティ時間を延長する際の延長時間(可変設定時間)をシステムリセット毎にランダムに決定するために用いられるフリーランカウンタや、乱数回路509にて生成される乱数のスタート値をシステムリセット毎にランダムに決定するために用いられるフリーランカウンタなどを、含んでも良い。或いは、これらのフリーランカウンタは、例えばRAM507のバックアップ領域といった、CTC508とは異なるメイン制御部41の内部回路に含まれても良い。10

#### 【0091】

メイン制御部41が備える乱数回路509は、例えば16ビット乱数といった、所定の更新範囲を有する乱数値となる数値データを生成する回路である。本実施例では、遊技制御基板40の側において、後述する内部抽選用の乱数値を示す数値データがカウント可能に制御される。尚、遊技効果を高めるために、これら以外の乱数値が用いられても良い。CPU505は、乱数回路509から抽出した数値データに基づき、乱数回路509とは異なるランダムカウンタを用いて、ソフトウェアによって各種の数値データを加工或いは更新することで、内部抽選用の乱数値を示す数値データをカウントするようにしても良い。20。以下では、内部抽選用の乱数値を示す数値データが、ハードウェアとなる乱数回路509からCPU505により抽出された数値データをソフトウェアにより加工しないものとする。尚、乱数回路509は、メイン制御部41に内蔵されるものであっても良いし、メイン制御部41とは異なる乱数回路チップとして、メイン制御部41に外付けされるものであっても良い。30

#### 【0092】

内部抽選用の乱数値は、複数種類の入賞について発生を許容するか否かを判定するために用いられる値であり、本実施例では、「0」～「65535」の範囲の値をとる。

#### 【0093】

図12は、乱数回路509の一構成例を示すブロック図である。乱数回路509は、図12に示すように、周波数監視回路551、クロック用フリップフロップ552、乱数生成回路553、スタート値設定回路554、乱数列変更回路555、乱数列変更設定回路556、ラッチ用フリップフロップ557A、557B、乱数ラッチセレクタ558A、558B、乱数値レジスタ559A、559Bを備えて構成される。尚、乱数値レジスタ559Aと乱数値レジスタ559Bはそれぞれ、図7(B)に示すようなメイン制御部41の内蔵レジスタに含まれる乱数値レジスタR1D(アドレス2038H-2039H)と乱数値レジスタR2D(アドレス203AH-203BH)に対応している。尚、本実施例では、取得する乱数が内部抽選用の乱数値のみであり、ラッチ用フリップフロップ557A、557B、乱数ラッチセレクタ558A、558B、乱数値レジスタ559A、559Bのうちラッチ用フリップフロップ557B、乱数ラッチセレクタ558B、乱数

40
50

値レジスタ 559B は未使用とされている。

**【0094】**

周波数監視回路 551 は、乱数用クロック生成回路 43 により生成された乱数用クロック RCLK の周波数を監視して、その異常発生を検知するための回路である。周波数監視回路 551 は、例えば乱数用外部クロック端子 ERC に入力される発振信号を監視して、内部システムクロック SCLK に基づきセキュリティ時間設定 KSES のビット番号 [7 - 6] における設定内容（図 10（B）参照）に応じた周波数異常を検知したときに、内部情報レジスタ CIF のビット番号 [4] を “1” にセットする。本実施例では、乱数用外部クロック端子 ERC に乱数用クロック生成回路 43 が生成した乱数用クロック RCLK が入力される。

10

**【0095】**

クロック用フリップフロップ 552 は、例えば D 型フリップフロップなどを用いて構成され、乱数用外部クロック端子 ERC からの乱数用クロック RCLK がクロック端子 CK に入力される。また、クロック用フリップフロップ 552 では、逆相出力端子（反転出力端子） Q バーが D 入力端子に接続されている。そして、正相出力端子（非反転出力端子） Q から乱数更新クロック RGK を出力する一方で、逆相出力端子（反転出力端子） Q バーからラッチ用クロック RC0 を出力する。この場合、クロック用フリップフロップ 552 は、クロック端子 CK に入力される乱数用クロック RCLK における信号状態が所定の変化をしたときに、正相出力端子（非反転出力端子） Q 及び逆相出力端子（反転出力端子） Q バーからの出力信号における信号状態を変化させる。例えば、クロック用フリップフロップ 552 は、乱数用クロック RCLK の信号状態がローレベルからハイレベルへと変化する立ち上がりのタイミング、或いは、乱数用クロック RCLK の信号状態がハイレベルからローレベルへと変化する立ち下がりのタイミングのうち、いずれか一方のタイミングにて、D 入力端子における入力信号を取り込む。このとき、正相出力端子（非反転出力端子） Q からは、D 入力端子にて取り込まれた入力信号が反転されることなく出力される一方で、逆相出力端子（反転出力端子） Q バーからは、D 入力端子にて取り込まれた入力信号が反転されて出力される。こうして、クロック用フリップフロップ 552 の正相出力端子（非反転出力端子） Q からは乱数用クロック RCLK における発振周波数（例えば 20 MHz）の 1/2 となる発振周波数（例えば 10 MHz）を有する乱数更新クロック RGK が出力される一方、逆相出力端子（反転出力端子） Q バーからは乱数更新クロック RGK の逆相信号（反転信号）、すなわち乱数更新クロック RGK と同一周波数で乱数更新クロック RGK とは位相が (= 180°) だけ異なるラッチ用クロック RC0 が出力される。

20

**【0096】**

クロック用フリップフロップ 552 から出力された乱数更新クロック RGK は、乱数生成回路 553 のクロック端子に入力されて、乱数生成回路 553 におけるカウント値の歩進に用いられる。また、クロック用フリップフロップ 552 から出力されたラッチ用クロック RC0 は、分岐点 BR1 にてラッチ用クロック RC1 とラッチ用クロック RC2 とに分岐される。したがって、ラッチ用クロック RC1 とラッチ用クロック RC2 とは、互いに同一の発振周波数を有し、互いに共通の周期で信号状態が変化することになる。ここで、ラッチ用クロック RC1 やラッチ用クロック RC2 における信号状態の変化としては、例えばローレベルからハイレベルへと変化する立ち上がりや、ハイレベルからローレベルへと変化する立ち下がりなどがある。ラッチ用クロック RC1 は、ラッチ用フリップフロップ 557A のクロック端子 CK に入力されて、ゲーム開始時ラッチ信号 SL1 の生成に用いられる乱数取得用クロックとなる。

30

**【0097】**

乱数用クロック RCLK の発振周波数と、制御用クロック生成回路 42 によって生成される制御用クロック CCLK の発振周波数とは、互いに異なる周波数となっており、また、いずれか一方の発振周波数が他方の発振周波数の整数倍になることがない。一例として、制御用クロック CCLK の発振周波数が 11.0 MHz である一方で、乱数用クロック

40

50

R C L K の発振周波数は 9 . 7 M H z であれば良い。そのため、乱数更新クロック R G K やラッチ用クロック R C 1、R C 2 はいずれも、C P U 5 0 5 に供給される制御用クロック C C L K とは異なる周期で信号状態が変化する発振信号となる。すなわち、クロック用フリップフロップ 5 5 2 は、乱数用クロック生成回路 4 3 によって生成された乱数用クロック R C L K に基づき、カウント値を更新するための乱数更新クロック R G K や、複数の乱数取得用クロックとなるラッチ用クロック R C 1、R C 2 として、制御用クロック C C L K や内部システムクロック S C L K ( 制御用クロック C C L K を 2 分周したもの ) とは異なる周期で信号状態が変化する発振信号を生成する。

#### 【 0 0 9 8 】

乱数生成回路 5 5 3 は、例えば 1 6 ビットのカウンタなどから構成され、クロック用フリップフロップ 5 5 2 から出力される乱数更新クロック R G K などの入力に基づき、数値データを更新可能な所定の範囲において所定の初期値から所定の最終値まで循環的に更新する回路である。例えば乱数生成回路 5 5 3 は、所定のクロック端子への入力信号である乱数更新クロック R G K における立ち上がりエッジに応答して、「 0 」から「 6 5 5 3 5 」までの範囲内で設定された初期値から「 6 5 5 3 5 」まで 1 ずつ加算するように数値データをカウントアップして行く。そして、「 6 5 5 3 5 」までカウントアップした後には、「 0 」から初期値よりも 1 小さい最終値となる数値まで 1 ずつ加算するようにカウントアップすることで、数値データを循環的に更新する。

#### 【 0 0 9 9 】

スタート値設定回路 5 5 4 は、第 2 乱数初期設定 K R S 2 のビット番号 [ 1 - 0 ] におけるビット値 ( 図 9 ( B ) 参照 ) に応じて、乱数生成回路 5 5 3 により生成されるカウント値におけるスタート値を設定する。例えば、スタート値設定回路 5 5 4 は、第 2 乱数初期設定 K R S 2 のビット番号 [ 1 - 0 ] が “ 0 0 ” であればスタート値をデフォルト値である「 0 0 0 0 H 」に設定し、“ 1 0 ” であれば I D ナンバーに基づく値に設定し、“ 0 1 ” であればシステムリセット毎に変更される値に設定する。本実施例では、第 2 乱数初期設定 K R S 2 のビット番号 [ 1 - 0 ] が “ 1 0 ” に設定されており、乱数生成回路 5 5 3 により生成されるカウント値におけるスタート値として I D ナンバーに基づく値が設定される。

#### 【 0 1 0 0 】

乱数列変更回路 5 5 5 は、乱数生成回路 5 5 3 により生成された数値データが一巡したときに、数値データの更新順である順列を所定の乱数更新規則に従った順列に変更可能とする回路である。例えば、乱数列変更回路 5 5 5 は、乱数生成回路 5 5 3 から出力される数値データにおけるビットの入れ替えや転置などのビットスクランブル処理を実行する。また、乱数列変更回路 5 5 5 は、例えばビットスクランブル処理に用いるビットスクランブル用キーやビットスクランブルテーブルを変更することなどにより、数値データの更新順である順列の変更を行うことができる。

#### 【 0 1 0 1 】

乱数列変更設定回路 5 5 6 は、第 1 乱数初期設定 K R S 1 のビット番号 [ 1 - 0 ] におけるビット値 ( 図 9 ( A ) 参照 ) などに応じて、乱数列変更回路 5 5 5 における数値データの更新順を変更する設定を行うための回路である。例えば、乱数列変更設定回路 5 5 6 は、第 1 乱数初期設定 K R S 1 のビット番号 [ 1 - 0 ] が “ 0 0 ” であれば 2 周目以降も乱数更新規則を変更しない設定とする一方、“ 0 1 ” であれば 2 周目以降はソフトウェアでの変更要求に応じて乱数更新規則を変更し、“ 1 0 ” であれば自動で乱数更新規則を変更する。本実施例では、第 1 乱数初期設定 K R S 1 のビット番号 [ 1 - 0 ] が “ 1 0 ” に設定されており、自動で乱数更新規則が変更される。

#### 【 0 1 0 2 】

乱数列変更設定回路 5 5 6 は、第 1 乱数初期設定 K R S 1 のビット番号 [ 1 - 0 ] が “ 0 1 ” であることに対応してソフトウェアによる乱数更新規則の変更を行う場合に、図 7 ( B ) に示すようなメイン制御部 4 1 が備える内蔵レジスタのうち、乱数列変更レジスタ R D S C ( アドレス 2 0 3 4 H ) を用いて、乱数更新規則の変更を制御する。図 1 3 ( A )

10

20

30

40

50

)は、乱数列変更レジスタR D S Cの構成例を示している。図13(B)は、乱数列変更レジスタR D S Cに格納される乱数列変更要求データの各ビットにおける設定内容の一例を示している。乱数列変更レジスタR D S Cのビット番号[0]に格納される乱数列変更要求データR D S C 0は、乱数更新規則をソフトウェアにより変更する場合に、乱数列の変更要求の有無を示している。図13(B)に示す例では、ソフトウェアにより乱数列の変更要求がないときに、乱数列変更要求データR D S C 0のビット値が“0”となる一方、乱数列の変更要求があったときには、そのビット値が“1”となる。

#### 【0103】

図14は、乱数更新規則をソフトウェアにより変更する場合の動作例を示している。この場合、乱数生成回路553から出力されるカウント値順列R C Nが所定の初期値から所定の最終値まで循環的に更新されたときに、乱数列変更要求データR D S C 0が“1”であることに応答して、乱数更新規則を変更する。図14に示す動作例では、始めに乱数列変更回路555から出力される乱数列R S Nが、「0 1 ... 65535」となっている。この後、C P U 505がR O M 506に格納されたユーザプログラムを実行することによって、所定のタイミングで乱数列変更レジスタR D S Cのビット番号[0]に“1”が書き込まれたものとする。

#### 【0104】

そして、第1乱数初期設定K R S 1のビット番号[1 - 0]が“01”であることに応じて、乱数列変更設定回路556が乱数列変更要求データR D S C 0を読み出し、そのビット値が“1”であることに応答して、乱数更新規則を変更するための設定を行う。このとき、乱数列変更設定回路556は、乱数生成回路553から出力されたカウント値順列R C Nが所定の最終値に達したことに応じて、例えは予め用意された複数種類の乱数更新規則のいずれかを選択することなどにより、乱数更新規則を変更する。図14に示す動作例では、乱数列変更回路555が乱数生成回路553から出力されたカウント値順列R C Nにおける最終値に対応する数値データ「65535」を出力した後、乱数列変更要求データR D S C 0に応じて乱数更新規則を変更する。その後、乱数列変更回路555は、変更後の乱数更新規則に従った乱数列R S Nとして、「65535 65534 ... 0」を出力する。乱数列変更レジスタR D S Cは、乱数列変更設定回路556により乱数列変更要求データR D S C 0が読み出されたときに初期化される。そのため、再び乱数列変更レジスタR D S Cのビット番号[0]にビット値“1”が書き込まれるまでは、乱数列変更回路555から出力される乱数列R S Nが、「65535 65534 ... 0」となる。

#### 【0105】

C P U 505によって、乱数列変更レジスタR D S Cのビット番号[0]に再びビット値“1”が書き込まれると、乱数更新規則が再度変更される。図14に示す動作例では、乱数列変更回路555が乱数列R S Nにおける最終値に対応する数値データ「0」を出力したときに、乱数列変更要求データR D S C 0としてビット値“1”が書き込まれたことに応じて乱数更新規則を変更する。その後、乱数列変更回路555は、変更後の乱数更新規則に従った乱数列R S Nとして、「0 2 ... 65534 1 ... 65535」を出力する。

#### 【0106】

図15は、乱数更新規則を自動で変更する場合の動作例を示している。この場合、乱数生成回路553から出力されるカウント値順列R C Nが所定の初期値から所定の最終値まで循環的に更新されたことに応じて、乱数列変更設定回路556が自動的に乱数更新規則を変更する。図15に示す動作例では、始めに乱数列変更回路555から出力される乱数列R S Nが、「0 1 ... 65535」となっている。

#### 【0107】

そして、乱数列変更回路555から出力された乱数列R S Nが所定の最終値に達したときに、乱数列変更設定回路556は、予め用意された複数種類の更新規則のうちから予め定められた順序に従って更新規則を選択することにより、更新規則を変更するようにして

10

20

30

40

50

も良い。或いは、乱数列変更設定回路 556 は、複数種類の更新規則のうちから任意の更新規則を選択することにより、更新規則を変更するようにしても良い。図 15 に示す動作例では、1 回目の乱数更新規則の変更により、乱数列変更回路 555 から出力される乱数列 RSN が、「65535 65534 ... 0」となる。その後、2 回目の乱数更新規則の変更により、乱数列変更回路 555 から出力される乱数列 RSN は、「0 2 ... 65534 1 ... 65535」となる。図 15 に示す動作例では、3 回目の乱数更新規則の変更により、乱数列変更回路 555 から出力される乱数列 RSN は、「65534 0 ... 32768」となる。4 回目の乱数更新規則の変更が行われたときには、乱数列変更回路 555 から出力される乱数列 RSN が、「16383 49151 ... 49150」となる。5 回目の乱数更新規則の変更が行われたときには、乱数列変更回路 555 から出力される乱数列 RSN が、「4 3 ... 465531」となる。  
10

#### 【0108】

ラッチ用フリップフロップ 557A は、例えば D 型フリップフロップなどを用いて構成される。ラッチ用フリップフロップ 557A では、D 入力端子に PIP510 が備える入力ポート P0 からの配線が接続され、クロック端子 CK にラッチ用クロック RC1 を伝送する配線が接続されている。本実施例では、入力ポート P0 にスタートスイッチ 7 からのゲーム開始信号 SS1 が入力される。ラッチ用フリップフロップ 557A は、ラッチ用クロック RC1 の立ち上がりエッジなどに応答して、ゲーム開始信号 SS1 を取り込み、ゲーム開始時ラッチ信号 SL1 として出力する。これにより、ラッチ用フリップフロップ 557A では、ラッチ用クロック RC1 の立ち上がりエッジに同期して、ゲーム開始信号 SS1 がゲーム開始時ラッチ信号 SL1 として出力される。  
20

#### 【0109】

尚、ゲーム開始信号 SS1 は、スタートスイッチ 7 から直接伝送されるものに限定されない。一例として、スタートスイッチ 7 からの出力信号がオン状態となっている時間を計測し、計測した時間が所定の時間（例えば 3ms）になったときに、ゲーム開始信号 SS1 を出力するタイマ回路を設けても良い。

#### 【0110】

乱数ラッチセレクタ 558A は、ラッチ用フリップフロップ 557A から伝送されるゲーム開始時ラッチ信号 SL1 と、ソフトウェアによる乱数ラッチ要求信号とを取り込み、いずれかを乱数ラッチ信号 LL1 として選択的に出力する回路である。乱数ラッチセレクタ 558A は、図 7 (B) に示すようなメイン制御部 41 が備える内蔵レジスタのうち、乱数値取込レジスタ RDLT (アドレス 2032H) を用いて、乱数ラッチ信号 LL1 の出力を制御する。乱数値取込レジスタ RDLT は、乱数列変更回路 555 から出力された乱数列 RSN における数値データを、ソフトウェアにより乱数値レジスタ 559A に取り込むために用いられるレジスタである。乱数ラッチ選択レジスタ RDLS は、乱数列変更回路 555 から出力された乱数列 RSN における数値データを、乱数値レジスタ 559A に、ソフトウェアにより取り込むか、入力ポート P0 への信号入力により取り込むかの取込方法を示すレジスタである。  
30

#### 【0111】

図 16 (A) は、乱数値取込レジスタ RDLT の構成例を示している。図 16 (B) は、乱数値取込レジスタ RDLT に格納される乱数値取込指定データの各ビットにおける設定内容の一例を示している。乱数値取込レジスタ RDLT のビット番号 [0] に格納される乱数値取込指定データ RDLT0 は、乱数値レジスタ R1D となる乱数値レジスタ 559A に対する乱数値取込指定の有無を示している。図 16 (B) に示す例では、ソフトウェアにより乱数値レジスタ R1D に対する乱数値の取込指定がないときに、乱数値取込指定データ RDLT0 のビット値が “0” となる一方、乱数値の取込指定があったときには、そのビット値が “1” となる。  
40

#### 【0112】

図 17 (A) は、乱数ラッチ選択レジスタ RDLS の構成例を示している。図 17 (B) は、乱数ラッチ選択レジスタ RDLS に格納される乱数ラッチ選択データの各ビットに  
50

おける設定内容の一例を示している。乱数ラッチ選択レジスタ R D L S のビット番号 [ 0 ] に格納される乱数ラッチ選択データ R D L S 0 は、乱数値レジスタ R 1 D となる乱数値レジスタ 5 5 9 A への取込方法を示している。図 17 ( B ) に示す例では、ソフトウェアによる乱数値取込指定データ R D L T 0 の書き込みに応じて乱数値となる数値データを乱数値レジスタ R 1 D に取り込む場合に、乱数ラッチ選択データ R D L S 0 のビット値を“0”とする。これに対して、入力ポート P 0 への信号入力に応じて乱数値となる数値データを乱数値レジスタ R 1 D に取り込む場合には、乱数ラッチ選択データ R D L S 0 のビット値を“1”とする。本実施例では、乱数ラッチ選択データ R D L S 0 のビット値が“1”とされており、入力ポート P 0 への信号入力に応じて乱数値となる数値データが乱数値レジスタ R 1 D に取り込まれる。

10

#### 【 0 1 1 3 】

乱数値レジスタ 5 5 9 A は、乱数列変更回路 5 5 5 から出力された乱数列 R S N における数値データを乱数値として格納するレジスタである。図 18 ( A ) 及び ( B ) は、乱数値レジスタ R 1 D となる乱数値レジスタ 5 5 9 A の構成例を示している。尚、図 18 ( A ) は、乱数値レジスタ R 1 D の下位バイト R 1 D ( L ) を示し、図 18 ( B ) は、乱数値レジスタ R 1 D の上位バイト R 1 D ( H ) を示している。乱数値レジスタ 5 5 9 A は 16 ビット ( 2 バイト ) のレジスタであり、16 ビットの乱数値を格納することができる。

#### 【 0 1 1 4 】

乱数値レジスタ 5 5 9 A は、乱数ラッチセレクタ 5 5 8 A から供給される乱数ラッチ信号 L L 1 がオン状態となったことに応答して、乱数列変更回路 5 5 5 から出力された乱数列 R S N における数値データを乱数値として取り込んで格納する。乱数値レジスタ 5 5 9 A は、C P U 5 0 5 から供給されるレジスタリード信号 R R S 1 がオン状態となったときに、読出可能 ( イネーブル ) 状態となり、格納されている数値データを内部バス等に出力する。これに対して、レジスタリード信号 R R S 1 がオフ状態であるときには、常に同じ値 ( 例えば「 6 5 5 3 5 H 」など ) を出力して、読出不能 ( ディセーブル ) 状態となれば良い。また、乱数値レジスタ 5 5 9 A は、乱数ラッチ信号 L L 1 がオン状態である場合に、レジスタリード信号 R R S 1 を受信不可能な状態となるようにしても良い。さらに、乱数値レジスタ 5 5 9 A は、乱数ラッチ信号 L L 1 がオン状態となるより前にレジスタリード信号 R R S 1 がオン状態となっている場合に、乱数ラッチ信号 L L 1 を受信不可能な状態となるようにしても良い。

20

#### 【 0 1 1 5 】

乱数値レジスタ 5 5 9 A は、図 7 ( B ) に示すようなメイン制御部 4 1 が備える内蔵レジスタのうち、乱数ラッチフラグレジスタ R D F M ( アドレス 2 0 3 3 H ) と、乱数割込制御レジスタ R D I C ( アドレス 2 0 3 1 H ) とを用いて、乱数ラッチ時の動作管理や割込制御を可能にする。乱数ラッチフラグレジスタ R D F M は、乱数値レジスタ 5 5 9 A に対応して、乱数値となる数値データがラッチされたか否かを示す乱数ラッチフラグを格納するレジスタである。乱数割込制御レジスタ R D I C は、乱数値レジスタ 5 5 9 A に乱数値となる数値データがラッチされたときに発生する割込の許可 / 禁止を設定するレジスタである。

30

#### 【 0 1 1 6 】

図 19 ( A ) は、乱数ラッチフラグレジスタ R D F M の構成例を示している。図 19 ( B ) は、乱数ラッチフラグレジスタ R D F M に格納される乱数ラッチフラグデータの各ビットにおける設定内容の一例を示している。乱数ラッチフラグレジスタ R D F M のビット番号 [ 0 ] に格納される乱数ラッチフラグデータ R D F M 0 は、乱数値レジスタ R 1 D となる乱数値レジスタ 5 5 9 A に数値データが取り込まれたか否かを示す乱数ラッチフラグとなる。図 19 ( B ) に示す例では、乱数値レジスタ R 1 D に数値データが取り込まれていないときに ( 亂数値取込なし ) 、乱数ラッチフラグデータ R D F M 0 のビット値が“0”となる一方、数値データが取り込まれたときには ( 亂数値取込あり ) 、そのビット値が“1”となる。乱数ラッチフラグデータ R D F M 0 が“1”的状態、すなわち乱数値レジスタ R 1 D に数値データが取り込まれている状態では、新たな乱数値の取込要求が発生し

40

50

た場合でも、新たな数値データを乱数値レジスタ R 1 D に取り込まないようになっており、このような状態では、乱数値レジスタ R 1 D の数値データが読み出されて、乱数ラッチフラグデータ R D F M 0 がクリアされるまで新たな数値データを乱数値レジスタ R 1 D に取り込むことが不可能となる。

#### 【 0 1 1 7 】

図 2 0 ( A ) は、乱数割込制御レジスタ R D I C の構成例を示している。図 2 0 ( B ) は、乱数割込制御レジスタ R D I C に格納される乱数割込制御データの各ビットにおける設定内容の一例を示している。乱数割込制御レジスタ R D I C のビット番号 [ 0 ] に格納される乱数割込制御データ R D I C 0 は、乱数値レジスタ R 1 D となる乱数値レジスタ 5 5 9 A に数値データが取り込まれたときに発生する割込を、許可するか禁止するかの割込制御設定を示している。図 2 0 ( B ) に示す例では、乱数値レジスタ R 1 D への取込における割込を禁止する場合に（割込禁止）、乱数割込制御データ R D I C 0 のビット値を“ 0 ”とする一方、この割込を許可する場合には（割込許可）、そのビット値を“ 1 ”とする。本実施例では、乱数割込制御データ R D I C 0 のビット値が“ 0 ”に設定されており、乱数値レジスタ R 1 D となる乱数値レジスタ 5 5 9 A に数値データが取り込まれても割込は発生しない。10

#### 【 0 1 1 8 】

メイン制御部 4 1 が備える P I P 5 1 0 は、例えば 6 ビット幅の入力専用ポートであり、専用端子となる入力ポート P 0 ~ 入力ポート P 2 と、機能兼用端子となる入力ポート P 3 ~ 入力ポート P 5 を含んでいる。入力ポート P 3 は、C P U 5 0 5 等に接続される外部マスカブル割込端子 X I N T と兼用される。入力ポート P 4 は、C P U 5 0 5 等に接続される外部ノンマスカブル割込端子 X N M I と兼用される。入力ポート P 5 は、シリアル通信回路 5 1 1 が使用する第 1 チャネル受信端子 R X A と兼用される。入力ポート P 3 ~ 入力ポート P 5 の使用設定は、プログラム管理エリアに記憶される機能設定 K F C S により指示される。20

#### 【 0 1 1 9 】

P I P 5 1 0 は、図 7 ( B ) に示すようなメイン制御部 4 1 が備える内蔵レジスタのうち、入力ポートレジスタ P I ( アドレス 2 0 9 0 H ) などを用いて、入力ポート P 0 ~ 入力ポート P 5 の状態管理等を行う。入力ポートレジスタ P I は、入力ポート P 0 ~ 入力ポート P 5 のそれぞれに対応して、外部信号の入力状態を示すビット値が格納されるレジスタである。30

#### 【 0 1 2 0 】

図 2 1 ( A ) は、入力ポートレジスタ P I の構成例を示している。図 2 1 ( B ) は、入力ポートレジスタ P I に格納される入力ポートデータの各ビットにおける設定内容の一例を示している。入力ポートレジスタ P I のビット番号 [ 5 ] に格納される入力ポートデータ P I 5 は、第 1 チャネル受信端子 R X A と兼用される入力ポート P 5 における端子状態（オン / オフ）を示している。入力ポートレジスタ P I のビット番号 [ 4 ] に格納される入力ポートデータ P I 4 は、外部ノンマスカブル割込端子 X N M I と兼用される入力ポート P 4 における端子状態（オン / オフ）を示している。入力ポートレジスタ P I のビット番号 [ 3 ] に格納される入力ポートデータ P I 3 は、外部マスカブル割込端子 X I N T と兼用される入力ポート P 3 における端子状態（オン / オフ）を示している。入力ポートレジスタ P I のビット番号 [ 2 ] に格納される入力ポートデータ P I 2 は、入力ポート P 2 における端子状態（オン / オフ）を示している。入力ポートレジスタ P I のビット番号 [ 1 ] に格納される入力ポートデータ P I 1 は、入力ポート P 1 における端子状態（オン / オフ）を示している。入力ポートレジスタ P I のビット番号 [ 0 ] に格納される入力ポートデータ P I 0 は、入力ポート P 0 における端子状態（オン / オフ）を示している。40

#### 【 0 1 2 1 】

図 5 に示すメイン制御部 4 1 が備えるアドレスデコード回路 5 1 2 は、メイン制御部 4 1 の内部における各機能ブロックのデコードや、外部装置用のデコード信号であるチップセレクト信号のデコードを行うための回路である。チップセレクト信号により、メイン制50

御部41の内部回路、或いは、周辺デバイスとなる外部装置を、選択的に有効動作させて、C P U 5 0 5からのアクセスが可能となる。

#### 【0122】

メイン制御部41が備えるR O M 5 0 6には、ゲーム制御用のユーザプログラムやセキュリティチェックプログラム506Aの他に、ゲームの進行を制御するために用いられる各種の選択用データ、テーブルデータなどが格納される。例えば、R O M 5 0 6には、C P U 5 0 5が各種の判定や決定、設定を行うために用意された複数の判定テーブルや決定テーブル、設定テーブルなどを構成するデータが記憶されている。また、R O M 5 0 6には、C P U 5 0 5が遊技制御基板40から各種の制御コマンドとなる制御信号を送信するために用いられる複数のコマンドテーブルを構成するテーブルデータなどが記憶されている。10

#### 【0123】

メイン制御部41が備えるR A M 5 0 7には、スロットマシン1におけるゲームの進行などを制御するために用いられる各種のデータを保持する領域として、遊技制御用データ保持エリア590が設けられている。R A M 5 0 7としては、例えばD R A Mが使用されており、記憶しているデータ内容を維持するためのリフレッシュ動作が必要になる。C P U 5 0 5には、このリフレッシュ動作を行うためのリフレッシュレジスタが内蔵されている。例えば、リフレッシュレジスタは8ビットからなり、そのうち下位7ビットはC P U 5 0 5がR O M 5 0 6から命令フェッチするごとに自動的にインクリメントされる。したがって、リフレッシュレジスタにおける格納値の更新は、C P U 5 0 5における1命令の実行時間ごとに行われることになる。20

#### 【0124】

メイン制御部41は、遊技制御基板40に接続された各種スイッチ類の検出状態が入力ポートから入力される。そしてメイン制御部41は、これら入力ポートから入力される各種スイッチ類の検出状態に応じて段階的に移行する基本処理を実行する。

#### 【0125】

また、メイン制御部41は、割込の発生により基本処理に割り込んで割込処理を実行できるようになっている。本実施例では、C T C 5 0 8に含まれるタイマ回路にてタイムアウトが発生したこと、すなわち一定時間間隔（本実施例では、約0.56ms）毎に後述するタイマ割込処理（メイン）を実行する。30

#### 【0126】

また、メイン制御部41は、割込処理の実行中に他の割込を禁止するように設定されているとともに、複数の割込が同時に発生した場合には、予め定められた順位によって優先して実行する割込が設定されている。尚、割込処理の実行中に他の割込要因が発生し、割込処理が終了してもその割込要因が継続している状態であれば、その時点で新たな割込が発生することとなる。

#### 【0127】

メイン制御部41は、基本処理として遊技制御基板40に接続された各種スイッチ類の検出状態が変化するまでは制御状態に応じた処理を繰り返しループし、各種スイッチ類の検出状態の変化に応じて段階的に移行する処理を実行する。また、メイン制御部41は、電断検出回路48から出力された電圧低下信号の入力に応じて電断割込処理（メイン）を実行し、一定時間間隔（本実施例では、約0.56ms）毎にタイマ割込処理（メイン）を実行する。尚、タイマ割込処理（メイン）の実行間隔は、基本処理において制御状態に応じて繰り返す処理が一巡する時間とタイマ割込処理（メイン）の実行時間とを合わせた時間よりも長い時間に設定されており、今回と次回のタイマ割込処理（メイン）との間で必ず制御状態に応じて繰り返す処理が最低でも一巡することとなる。40

#### 【0128】

メイン制御部41は、演出制御基板90に各種のコマンドを送信する。遊技制御基板40から演出制御基板90へ送信されるコマンドは一方方向のみで送られ、演出制御基板90から遊技制御基板40へ向けてコマンドが送られることはない。50

**【0129】**

演出制御基板90には、スロットマシン1の前面扉1bに配置された液晶表示器51(図1参照)、演出効果LED52、スピーカ53、54、前述したリールLED55等の演出装置が接続されており、これら演出装置は、演出制御基板90に搭載された後述のサブ制御部91による制御に基づいて駆動されるようになっている。

**【0130】**

尚、本実施例では、演出制御基板90に搭載されたサブ制御部91により、液晶表示器51、演出効果LED52、スピーカ53、54、リールLED55等の演出装置の出力制御が行われる構成であるが、サブ制御部91とは別に演出装置の出力制御を直接的に行う出力制御部を演出制御基板90または他の基板に搭載し、サブ制御部91がメイン制御部41からのコマンドに基づいて演出装置の出力パターンを決定し、サブ制御部91が決定した出力パターンに基づいて出力制御部が演出装置の出力制御を行う構成としても良く、このような構成では、サブ制御部91及び出力制御部の双方によって演出装置の出力制御が行われることとなる。

10

**【0131】**

また、本実施例では、演出装置として液晶表示器51、演出効果LED52、スピーカ53、54、リールLED55を例示しているが、演出装置は、これらに限られず、例えば、機械的に駆動する表示装置や機械的に駆動する役モノなどを演出装置として適用しても良い。

20

**【0132】**

演出制御基板90には、サブCPU91a、ROM91b、RAM91c、I/Oポート91dなどを備えたマイクロコンピュータにて構成され、演出の制御を行うサブ制御部91、演出制御基板90に接続された液晶表示器51の表示制御を行う表示制御回路92、演出効果LED52、リールLED55の駆動制御を行うLED駆動回路93、スピーカ53、54からの音声出力制御を行う音声出力回路94、電源投入時または電源遮断時にサブCPU91aにリセット信号を与えるリセット回路95、日付情報及び時刻情報を含む時間情報を出力する時計装置97、スロットマシン1に供給される電源電圧を監視し、電圧低下を検出したときに、その旨を示す電圧低下信号をサブ制御部91に対して出力する電断検出回路98が搭載されており、サブ制御部91は、メイン制御部から送信されるコマンドを受けて、演出を行うための各種の制御を行うとともに、演出制御基板90に搭載された制御回路の各部を直接的または間接的に制御する。

30

**【0133】**

サブ制御部91は、メイン制御部41と同様に、割込機能を備えており、メイン制御部41からのコマンド受信時に割込を発生させて、メイン制御部41から送信されたコマンドを取得し、バッファに格納するコマンド受信割込処理を実行する。また、サブ制御部91は、システムクロックの入力数が一定数に到達する毎、すなわち一定間隔毎に割込を発生させて後述するタイマ割込処理(サブ)を実行する。

**【0134】**

また、サブ制御部91は、メイン制御部41とは異なり、コマンドの受信に基づいて割込が発生した場合には、タイマ割込処理(サブ)の実行中であっても、当該処理に割り込んでコマンド受信割込処理を実行し、タイマ割込処理(サブ)の契機となる割込が同時に発生してもコマンド受信割込処理を最優先で実行するようになっている。

40

**【0135】**

また、サブ制御部91にも、停電時においてバックアップ電源が供給されており、バックアップ電源が供給されている間は、RAM91cに記憶されているデータが保持されるようになっている。

**【0136】**

本実施例のスロットマシン1は、設定値に応じてメダルの払出率が変わるものである。詳しくは、後述する内部抽選において設定値に応じた当選確率を用いることにより、メダルの払出率が変わらくなっている。設定値は1~6の6段階からなり、6が最も払出

50

率が高く、5、4、3、2、1の順に値が小さくなるほど払出率が低くなる。すなわち設定値として6が設定されている場合には、遊技者にとって最も有利度が高く、5、4、3、2、1の順に値が小さくなるほど有利度が段階的に低くなる。

#### 【0137】

設定値を変更するためには、設定キースイッチ37をON状態としてからスロットマシン1の電源をONする必要がある。設定キースイッチ37をON状態として電源をONすると、設定値表示器24にRAM507から読み出された設定値が表示値として表示され、リセット／設定スイッチ38の操作による設定値の変更操作が可能な設定変更状態に移行する。設定変更状態において、リセット／設定スイッチ38が操作されると、設定値表示器24に表示された表示値が1ずつ更新されていく（設定6からさらに操作されたときは、設定1に戻る）。そして、スタートスイッチ7が操作されると表示値を設定値として確定する。そして、設定キースイッチ37がOFFされると、確定した表示値（設定値）がメイン制御部41のRAM507に格納され、遊技の進行が可能な状態に移行する。

#### 【0138】

また、設定値を確認するためには、ゲーム終了後、賭数が設定されていない状態で設定キースイッチ37をON状態とすれば良い。このような状況で設定キースイッチ37をON状態とすると、設定値表示器24にRAM507から読み出された設定値が表示されることで設定値を確認可能な設定確認状態に移行する。設定確認状態においては、ゲームの進行が不能であり、設定キースイッチ37をOFF状態とすることで、設定確認状態が終了し、ゲームの進行が可能な状態に復帰することとなる。

#### 【0139】

本実施例のスロットマシン1においては、メイン制御部41は、タイマ割込処理（メイン）を実行する毎に、電断検出回路48からの電圧低下信号が検出されているか否かを判定する停電判定処理を行い、停電判定処理において電圧低下信号が検出されていると判定した場合に、電断処理（メイン）を実行する。電断処理（メイン）では、レジスタを後述するRAM507のスタックに退避し、RAM507にいずれかのビットが1となる破壊診断用データ（本実施例では、5AH）、すなわち0以外の特定のデータを格納するとともに、RAM507の全ての領域に格納されたデータに基づくRAMパリティが0となるようにRAMパリティ調整用データを計算し、RAM507に格納する処理を行うようになっている。尚、RAMパリティとはRAM507の該当する領域（本実施例では、全ての領域）の各ビットに格納されている値の排他的論理和として算出される値である。このため、RAM507の全ての領域に格納されたデータに基づくRAMパリティが0であれば、RAMパリティ調整用データは0となり、RAM507の全ての領域に格納されたデータに基づくRAMパリティが1であれば、RAMパリティ調整用データは1となる。

#### 【0140】

そして、メイン制御部41は、システムリセットによるかユーザリセットによるかに関わらず、その起動時においてRAM507の全ての領域に格納されたデータに基づいてRAMパリティを計算するとともに、破壊診断用データの値を確認し、RAMパリティが0であり、かつ破壊診断用データの値も正しいことを条件に、RAM507に記憶されているデータに基づいてメイン制御部41の処理状態を電断前の状態に復帰させるが、RAMパリティが0でない場合（1の場合）や破壊診断用データの値が正しくない場合には、RAM異常と判定し、RAM異常エラーコードをレジスタにセットしてRAM異常エラー状態に制御し、遊技の進行を不能化させるようになっている。尚、RAM異常エラー状態は、通常のエラー状態と異なり、リセットスイッチ23やリセット／設定スイッチ38を操作しても解除されないようにになっており、前述した設定変更状態において新たな設定値が設定されるまで解除されることがない。

#### 【0141】

尚、本実施例では、RAM507に格納されている全てのデータが停電時においてもバックアップ電源により保持されるとともに、メイン制御部41は、電源投入時においてRAM507のデータが正常であると判定した場合に、RAM507の格納データに基づい

10

20

30

40

50

て電断前の制御状態に復帰する構成であるが、RAM507に格納されているデータのうち停電時において制御状態の復帰に必要なデータのみをバックアップし、電源投入時においてバックアップされているデータに基づいて電断前の制御状態に復帰する構成としても良い。

#### 【0142】

また、電源投入時において電断前の制御状態に復帰させる際に、全ての制御状態を電断前の制御状態に復帰させる必要はなく、遊技者に対して不利益とならない最低限の制御状態を復帰させる構成であれば良く、例えば、入力ポートの状態などを全て電断前の状態に復帰させる必要はない。

#### 【0143】

また、サブ制御部91もタイマ割込処理(サブ)において電断検出回路98からの電圧低下信号が検出されているか否かを判定し、電圧低下信号が検出されていると判定した場合に電断処理(サブ)を実行する。電断処理(サブ)では、レジスタを後述するRAM91cのスタックに退避し、RAM91cにいずれかのビットが1となる破壊診断用データを格納するとともに、RAM91cの全ての領域に格納されたデータに基づくRAMパリティが0となるようにRAMパリティ調整用データを計算し、RAM91cに格納する処理を行うようになっている。

#### 【0144】

そして、サブ制御部91は、その起動時においてRAM91cの全ての領域に格納されたデータに基づいてRAMパリティを計算し、RAMパリティが0であることを条件に、RAM91cに記憶されているデータに基づいてサブ制御部91の処理状態を電断前の状態に復帰させるが、RAMパリティが0でない場合(1の場合)には、RAM異常と判定し、RAM91cを初期化するようになっている。この場合、メイン制御部41と異なり、RAM91cが初期化されるのみで演出の実行が不能化されることはない。

#### 【0145】

尚、本実施例では、RAM91cに格納されている全てのデータが停電時においてもバックアップ電源により保持されるとともに、サブ制御部91は、電源投入時においてRAM91cのデータが正常であると判定した場合に、RAM91cの格納データに基づいて電断前の制御状態に復帰する構成であるが、RAM91cに格納されているデータのうち停電時において制御状態の復帰に必要なデータのみをバックアップし、電源投入時においてバックアップされているデータに基づいて電断前の制御状態に復帰する構成としても良い。

#### 【0146】

また、電源投入時において電断前の制御状態に復帰させる際に、全ての制御状態を電断前の制御状態に復帰させる必要はなく、遊技者に対して不利益とならない最低限の制御状態を復帰させる構成であれば良く、入力ポートの状態や、演出が途中で中断された場合の途中経過などを全て電断前の状態に復帰させる必要はない。

#### 【0147】

次に、メイン制御部41のRAM507の初期化について説明する。メイン制御部41のRAM507の格納領域は、重要ワーク、一般ワーク、特別ワーク、設定値ワーク、非保存ワーク、未使用領域、スタック領域に区分されている。

#### 【0148】

重要ワークは、各種表示器やLEDの表示用データ、I/Oの入出力データ、遊技時間の計時カウンタ等、BB終了時に初期化すると不都合があるデータが格納されるワークである。一般ワークは、停止制御テーブル、停止図柄、メダルの払出枚数、BB中のメダル払出総数等、BB終了時に初期化可能なデータが格納されるワークである。特別ワークは、演出制御基板90へコマンドを送信するためのデータ、各種ソフトウェア乱数等、設定開始前にのみ初期化されるデータが格納されるワークである。設定値ワークは、内部抽選処理で抽選を行う際に用いる設定値が格納されるワークである。非保存ワークは、各種スイッチ類の状態を保持するワークであり、起動時にRAM507のデータが破壊されてい

10

20

30

40

50

るか否かに関わらず必ず値が設定されることとなる。未使用領域は、RAM507の格納領域のうち使用していない領域であり、後述する複数の初期化条件のいずれか1つでも成立すれば初期化されることとなる。スタック領域は、メイン制御部41のレジスタから退避したデータが格納される領域であり、このうちの未使用スタック領域は、未使用領域と同様に、後述する複数の初期化条件のいずれか1つでも成立すれば初期化されることとなるが、使用中スタック領域は、プログラムの続行のため、初期化されることはない。

#### 【0149】

本実施例においてメイン制御部41は、設定キースイッチ37がONの状態での起動時、RAM異常エラー発生時、BB終了時、設定キースイッチ37がOFFの状態での起動時においてRAM507のデータが破壊されていないとき、1ゲーム終了時の5つからなる初期化条件が成立した際に、各初期化条件に応じて初期化される領域の異なる4種類の初期化を行う。10

#### 【0150】

初期化1は、起動時において設定キースイッチ37がONの状態であり、設定変更状態へ移行する場合において、その前に行う初期化、またはRAM異常エラー発生時に行う初期化であり、初期化1では、RAM507の格納領域のうち、使用中スタック領域を除く全ての領域（未使用領域及び未使用スタック領域を含む）が初期化される。初期化2は、BB終了時に行う初期化であり、初期化2では、RAM507の格納領域のうち、一般ワーク、未使用領域及び未使用スタック領域が初期化される。初期化3は、起動時において設定キースイッチ37がOFFの状態であり、かつRAM507のデータが破壊されていない場合において行う初期化であり、初期化3では、非保存ワーク、未使用領域及び未使用スタック領域が初期化される。初期化4は、1ゲーム終了時に行う初期化であり、初期化4では、RAM507の格納領域のうち、未使用領域及び未使用スタック領域が初期化される。20

#### 【0151】

尚、本実施例では、初期化1を設定変更状態の移行前に行っているが、設定変更状態の終了時に行ったり、設定変更状態移行前、設定変更状態終了時の双方で行うようにしても良い。この場合、設定値ワークを初期化してしまうと確定した設定値が失われてしまうこととなるので、設定変更状態終了時の初期化では、設定値ワークの初期化は行われない。

#### 【0152】

本実施例のスロットマシン1は、前述のように遊技状態に応じて設定可能な賭数の規定数が定められており、遊技状態に応じて定められた規定数の賭数が設定されたことを条件にゲームを開始させることが可能となる。本実施例では、後に説明するが、遊技状態として、レギュラーボーナス（以下ではRBと称す）（ビッグボーナス（以下ではBBと称す））、通常遊技状態があり、このうちRB（BB）では賭数の規定数として2が定められており、通常遊技状態では賭数の規定数として3が定められている。このため、遊技状態がRB（BB）であれば、賭数として2が設定されるとゲームを開始させることが可能となり、通常遊技状態であれば、賭数として3が設定されるとゲームを開始させることが可能となる。尚、本実施例では、遊技状態に応じた規定数の賭数が設定された時点で、全ての入賞ラインL1～L5が有効化されるようになっており、RB（BB）では賭数として2が定められた時点で全ての入賞ラインL1～L5が有効化されることなり、通常遊技状態では賭数として3が設定された時点で全ての入賞ラインL1～L5が有効化されることとなる。30

#### 【0153】

本実施例のスロットマシン1は、全てのリール2L、2C、2Rが停止した際に、有効化された入賞ライン（本実施例の場合、常に全ての入賞ラインが有効化されるため、以下では、有効化された入賞ラインを単に入賞ラインと呼ぶ）上に役と呼ばれる図柄の組合せが揃うと入賞となる。役は、同一図柄の組合せであっても良いし、異なる図柄を含む組合せであっても良い。入賞となる役の種類は、遊技状態に応じて定められているが、大きく分けて、メダルの払い出しを伴う小役と、賭数の設定を必要とせずに次のゲームを開始可40

能となる再遊技役と、遊技状態の移行を伴う特別役と、がある。以下では、小役と再遊技役をまとめて一般役とも呼ぶ。遊技状態に応じて定められた各役の入賞が発生するためには、後述する内部抽選に当選して、当該役の当選フラグがRAM507に設定されている必要がある。

#### 【0154】

尚、これら各役の当選フラグのうち、小役及び再遊技役の当選フラグは、当該フラグが設定されたゲームにおいてのみ有効とされ、次のゲームでは無効となるが、特別役の当選フラグは、当該フラグにより許容された役の組合せが揃うまで有効とされ、許容された役の組合せが揃ったゲームにおいて無効となる。すなわち特別役の当選フラグが一度当選すると、例え、当該フラグにより許容された役の組合せを揃えることができなかった場合にも、その当選フラグは無効とされずに、次のゲームへ持ち越されることとなる。10

#### 【0155】

このスロットマシン1における役としては、図22に示すように、特別役としてビッグボーナス（以下ではビッグボーナスをBBとする）、レギュラーボーナス（以下ではレギュラーボーナスをRBとする）が、小役として1枚、スイカ、チェリー、ベルが、再遊技役としてリプレイが定められている。

#### 【0156】

チェリーは、いずれの遊技状態においても右リールについて入賞ラインのいずれかに「白チェリー」の図柄が導出されたときに入賞となり、いずれの遊技状態においても1枚のメダルが払い出される。尚、「白チェリー」の図柄が右リールの上段または下段に停止した場合には、入賞ラインL2、L5または入賞ラインL3、L4の2本の入賞ラインにチェリーの組合せが揃うこととなり、2本の入賞ライン上でチェリーに入賞したこととなるので、2枚のメダルが払い出されることとなる。20

#### 【0157】

スイカは、いずれの遊技状態においても入賞ラインのいずれかに「スイカ - スイカ - スイカ」の組合せまたは「スイカ - スイカ - BAR」の組合せが揃ったときに入賞となり、RB(BB)では15枚のメダルが払い出され、通常遊技状態では12枚のメダルが払い出される。ベルは、いずれの遊技状態においても入賞ラインのいずれかに「ベル - ベル - ベル」の組合せが揃ったときに入賞となり、RB(BB)では15枚のメダルが払い出され、通常遊技状態では10枚のメダルが払い出される。30

#### 【0158】

リプレイは、通常遊技状態において入賞ラインのいずれかに「リプレイ - リプレイ - リプレイ」の組合せ、「BAR - リプレイ - リプレイ」の組合せ、または「黒7 - リプレイ - リプレイ」の組合せのうちいずれかの組合せが揃ったときに入賞となる。リプレイが入賞したときには、メダルの払い出しはないが次のゲームを改めて賭数を設定することなく開始できるので、次のゲームで設定不要となった賭数に対応した3枚のメダルが払い出されるのと実質的には同じこととなる。

#### 【0159】

RBは、通常遊技状態において入賞ラインのいずれかに「網7 - 網7 - 黒7」の組合せが揃ったときに入賞となり、遊技状態がRBに移行する。RBは、小役、特にベルの当選確率が高まることによって他の遊技状態よりも遊技者にとって有利となる遊技状態であり、RBが開始した後、12ゲームを消化したとき、または8ゲーム入賞（役の種類は、いずれでも可）したとき、のいずれか早いほうで終了する。40

#### 【0160】

BBは、通常遊技状態において入賞ラインのいずれかに「黒7 - 黒7 - 黒7」の組合せ、「網7 - 網7 - 網7」の組合せまたは「白7 - 白7 - 白7」の組合せが揃ったときに入賞となる。

#### 【0161】

BBが入賞すると、遊技状態がBBに移行するとともに同時にRBに移行し、RBが終了した際に、BBが終了していなければ、再度RBに移行し、BBが終了するまで繰り返50

し R B に制御される。すなわち B B 中は、常に R B に制御されることとなる。そして、B B は、当該 B B 中において遊技者に払い出したメダルの総数が 465 枚を超えたときに終了する。B B の終了時には、R B の終了条件が成立しているか否かに関わらず R B も終了する。

#### 【0162】

以下、本実施例の内部抽選について説明する。内部抽選は、上記した各役への入賞を許容するか否かを、全てのリール 2 L、2 C、2 R の表示結果が導出表示される以前に（実際には、スタートスイッチ 7 の検出時）決定するものである。内部抽選では、まず、スタートスイッチ 7 の検出時に内部抽選用の乱数値（0 ~ 65535 の整数）を取得する。詳しくは、スタートスイッチ 7 の操作の検出と同時にラッチされ、乱数値レジスタ R 1 D に格納された数値データを読み出し、RAM 507 に割り当てられた乱数値ワークに格納する。そして、遊技状態及び特別役の持ち越しの有無に応じて定められた各役について、乱数値ワークに格納された数値データと、遊技状態、賭数及び設定値に応じて定められた各役の判定値数に応じて行われる。10

#### 【0163】

本実施例では、図 22 に示すように、遊技状態が、通常遊技状態であるか、R B (B B) であるか、によって内部抽選の対象となる役が異なる。さらに遊技状態が通常遊技状態においては、特別役の持越しの有無に応じて定められた各役について、乱数値ワークに格納された数値データと、遊技状態、賭数及び設定値に応じて定められた各役の判定値数に応じて行われる。

#### 【0164】

遊技状態が通常遊技状態であり、いずれの特別役も持越しされていない状態では、B B、R B、リプレイ、スイカ、チェリー、ベルが内部抽選の対象役として順に読み出される。20

#### 【0165】

遊技状態が通常遊技状態であり、いずれかの特別役が持越しされている状態では、リプレイ、スイカ、チェリー、ベルが内部抽選の対象役として順に読み出される。

#### 【0166】

遊技状態が R B では、スイカ、チェリー、ベルが内部抽選の対象役として順に読み出される。

#### 【0167】

内部抽選では、内部抽選の対象となる役、現在の遊技状態及び設定値に対応して定められた判定値数を、内部抽選用の乱数値に順次加算し、加算の結果がオーバーフローしたときに、当該役に当選したものと判定される。このため、判定値数の大小に応じた確率（判定値数 / 65536）で役が当選することとなる。30

#### 【0168】

そして、いずれかの役の当選が判定された場合には、当選が判定された役に対応する当選フラグを RAM 507 に割り当てられた内部当選フラグ格納ワークに設定する。内部当選フラグ格納ワークは、2 バイトの格納領域にて構成されており、そのうちの上位バイトが、特別役の当選フラグが設定される特別役格納ワークとして割り当てられ、下位バイトが、一般役の当選フラグが設定される一般役格納ワークとして割り当てられている。詳しくは、特別役が当選した場合には、当該特別役が当選した旨を示す特別役の当選フラグを特別役格納ワークに設定し、一般役格納ワークに設定されている当選フラグをクリアする。また、一般役が当選した場合には、当該一般役が当選した旨を示す一般役の当選フラグを一般役格納ワークに設定する。尚、いずれの役及び役の組合せにも当選しなかった場合には、一般役格納ワークのみクリアする。40

#### 【0169】

次に、リール 2 L、2 C、2 R の停止制御について説明する。

#### 【0170】

メイン制御部 41 は、リールの回転が開始したとき、及びリールが停止し、かつ未だ回転中のリールが残っているときに、ROM 506 に格納されているテーブルインデックス及びテーブル作成用データを参照して、回転中のリール別に停止制御テーブルを作成する50

。そして、ストップスイッチ 8 L、8 C、8 R のうち、回転中のリールに対応するいずれかの操作が有効に検出されたときに、該当するリールの停止制御テーブルを参照し、参照した停止制御テーブルの滑りコマ数に基づいて、操作されたストップスイッチ 8 L、8 C、8 R に対応するリール 2 L、2 C、2 R の回転を停止させる制御を行う。

#### 【0171】

テーブルインデックスには、内部抽選による当選フラグの設定状態（以下、内部当選状態と呼ぶ）別に、テーブルインデックスを参照する際の基準アドレスから、テーブル作成用データが格納された領域の先頭アドレスを示すインデックスデータが格納されているアドレスまでの差分が登録されている。これにより内部当選状態に応じた差分を取得し、基準アドレスに対してその差分を加算することで該当するインデックスデータを取得することが可能となる。尚、役の当選状況が異なる場合でも、同一の制御が適用される場合においては、インデックスデータとして同一のアドレスが格納されており、このような場合には、同一のテーブル作成用データを参照して、停止制御テーブルが作成されることとなる。

10

#### 【0172】

テーブル作成用データは、停止操作位置に応じた滑りコマ数を示す停止制御テーブルと、リールの停止状況に応じて参照すべき停止制御テーブルのアドレスと、からなる。

#### 【0173】

リールの停止状況に応じて参照される停止制御テーブルは、全てのリールが回転しているか、左リールのみ停止しているか、中リールのみ停止しているか、右リールのみ停止しているか、左、中リールが停止しているか、左、右リールが停止しているか、中、右リールが停止しているか、によって異なる場合があり、更に、いずれかのリールが停止している状況においては、停止済みのリールの停止位置によっても異なる場合があるので、それぞれの状況について、参照すべき停止制御テーブルのアドレスが回転中のリール別に登録されており、テーブル作成用データの先頭アドレスに基づいて、それぞれの状況に応じて参照すべき停止制御テーブルのアドレスが特定可能とされ、この特定されたアドレスから、それぞれの状況に応じて必要な停止制御テーブルを特定できるようになっている。尚、リールの停止状況や停止済みのリールの停止位置が異なる場合でも、同一の停止制御テーブルが適用される場合においては、停止制御テーブルのアドレスとして同一のアドレスが登録されているものもあり、このような場合には、同一の停止制御テーブルが参照されることとなる。

20

#### 【0174】

停止制御テーブルは、停止操作が行われたタイミング別の滑りコマ数を特定可能なデータである。本実施例では、リールモータ 32 L、32 C、32 R に、168 ステップ（0 ~ 167）の周期で 1 周するステッピングモータを用いている。すなわちリールモータ 32 L、32 C、32 R を 168 ステップ駆動させることでリール 2 L、2 C、2 R が 1 周することとなる。そして、リール 1 周に対して 16 ステップ（1 図柄が移動するステップ数）毎に分割した 21 の領域（コマ）が定められており、これらの領域には、リール基準位置から 0 ~ 20 の領域番号が割り当てられている。一方、1 リールに配列された図柄数も 21 であり、各リールの図柄に対して、リール基準位置から 0 ~ 20 の図柄番号が割り当てられているので、0 番図柄から 20 番図柄に対して、それぞれ 0 ~ 20 の領域番号が順に割り当てられることとなる。そして、停止制御テーブルには、領域番号別の滑りコマ数が所定のルールで圧縮して格納されており、停止制御テーブルを展開することによって領域番号別の滑りコマ数を取得できるようになっている。

30

#### 【0175】

前述のようにテーブルインデックス及びテーブル作成用データを参照して作成される停止制御テーブルは、領域番号に対応して、各領域番号に対応する領域が停止基準位置（本実施例では、透視窓 3 の下段図柄の領域）に位置するタイミング（リール基準位置からのステップ数が各領域番号のステップ数の範囲に含まれるタイミング）でストップスイッチ 8 L、8 C、8 R の操作が検出された場合の滑りコマ数がそれぞれ設定されたテーブルで

40

50

ある。

#### 【0176】

次に、停止制御テーブルの作成手順について説明すると、まず、リール回転開始時においては、そのゲームの内部当選状態に応じたテーブル作成用データの先頭アドレスを取得する。具体的には、まずテーブルインデックスを参照し、内部当選状態に対応するインデックスデータを取得し、そして取得したインデックスデータに基づいてテーブル作成用データを特定し、特定したテーブル作成用データから全てのリールが回転中の状態に対応する各リールの停止制御テーブルのアドレスを取得し、取得したアドレスに格納されている各リールの停止制御テーブルを展開して全てのリールについて停止制御テーブルを作成する。

10

#### 【0177】

また、いずれか1つのリールが停止したとき、またはいずれか2つのリールが停止したときには、リール回転開始時に取得したインデックスデータ、すなわちそのゲームの内部当選状態に応じたテーブル作成用データの先頭アドレスに基づいてテーブル作成用データを特定し、特定したテーブル作成用データから停止済みのリール及び当該リールの停止位置の領域番号に対応する未停止リールの停止制御テーブルのアドレスを取得し、取得したアドレスに格納されている各リールの停止制御テーブルを展開して未停止のリールについて停止制御テーブルを作成する。

#### 【0178】

次に、メイン制御部41がストップスイッチ8L、8C、8Rのうち、回転中のリールに対応するいずれかの操作を有効に検出したときに、該当するリールに表示結果を導出させる際の制御について説明すると、ストップスイッチ8L、8C、8Rのうち、回転中のリールに対応するいずれかの操作を有効に検出すると、停止操作を検出した時点のリール基準位置からのステップ数に基づいて停止操作位置の領域番号を特定し、停止操作が検出されたリールの停止制御テーブルを参照し、特定した停止操作位置の領域番号に対応する滑りコマ数を取得する。そして、取得した滑りコマ数分リールを回転させて停止させる制御を行う。具体的には、停止操作を検出した時点のリール基準位置からのステップ数から、取得した滑りコマ数引き込んで停止させるまでのステップ数を算出し、算出したステップ数分リールを回転させて停止させる制御を行う。これにより、停止操作が検出された停止操作位置の領域番号に対応する領域から滑りコマ数分先の停止位置となる領域番号に対応する領域が停止基準位置（本実施例では、透視窓3の下段図柄の領域）に停止することとなる。

20

#### 【0179】

本実施例のテーブルインデックスには、一の遊技状態における一の内部当選状態に対応するインデックスデータとして1つのアドレスのみが格納されており、更に、一のテーブル作成用データには、一のリールの停止状況（及び停止済みのリールの停止位置）に対応する停止制御テーブルの格納領域のアドレスとして1つのアドレスのみが格納されている。すなわち一の遊技状態における一の内部当選状態に対応するテーブル作成用データ、及びリールの停止状況（及び停止済みのリールの停止位置）に対応する停止制御テーブルが一意的に定められており、これらを参照して作成される停止制御テーブルも、一の遊技状態における一の内部当選状態、及びリールの停止状況（及び停止済みのリールの停止位置）に対して一意となる。このため、遊技状態、内部当選状態、リールの停止状況（及び停止済みのリールの停止位置）の全てが同一条件となった際に、同一の停止制御テーブル、すなわち同一の制御パターンに基づいてリールの停止制御が行われることとなる。

30

#### 【0180】

また、本実施例では、滑りコマ数として0～4の値が定められており、停止操作を検出してから最大4コマ図柄を引き込んでリールを停止させることが可能である。すなわち停止操作を検出した停止操作位置を含め、最大5コマの範囲から図柄の停止位置を指定できるようになっている。また、1図柄分リールを移動させるのに1コマの移動が必要であるので、停止操作を検出してから最大4図柄を引き込んでリールを停止させることが可能で

40

50

あり、停止操作を検出した停止操作位置を含め、最大 5 図柄の範囲から図柄の停止位置を指定することとなる。

#### 【 0 1 8 1 】

本実施例では、いずれかの役に当選している場合には、当選役を入賞ライン上に 4 コマの範囲で最大限引き込み、当選していない役が入賞ライン上に揃わないように引き込む滑りコマ数が定められた停止制御テーブルを作成し、リールの停止制御を行う一方、いずれの役にも当選していない場合には、いずれの役も揃わない滑りコマ数が定められた停止制御テーブルを作成し、リールの停止制御を行う。これにより、停止操作が行われた際に、入賞ライン上に最大 4 コマの引込範囲で当選している役を揃えて停止させることができれば、これを揃えて停止させる制御が行われ、当選していない役は、最大 4 コマの引込範囲でハズシテ停止させることとなる。10

#### 【 0 1 8 2 】

特別役が前ゲーム以前から持ち越されている状態で小役が当選した場合など、特別役と小役が同時に当選している場合には、当選した小役を入賞ラインに 4 コマの範囲で最大限に引き込むように滑りコマ数が定められているとともに、当選した小役を入賞ラインに最大 4 コマの範囲で引き込めない停止操作位置については、当選した特別役を入賞ラインに 4 コマの範囲で最大限に引き込むように滑りコマ数が定められた停止制御テーブルを作成し、リールの停止制御を行う。これにより、停止操作が行われた際に、入賞ライン上に最大 4 コマの引込範囲で当選している小役を揃えて停止させることができれば、これを揃えて停止させる制御が行われ、入賞ライン上に最大 4 コマの引込範囲で当選している小役を引き込めない場合には、入賞ライン上に最大 4 コマの引込範囲で当選している特別役を揃えて停止させることができれば、これを揃えて停止させる制御が行われ、当選していない役は、4 コマの引込範囲でハズシテ停止させる制御が行われることとなる。すなわちこのような場合には、特別役よりも小役を入賞ライン上に揃える制御が優先され、小役を引き込めない場合にのみ、特別役を入賞させることが可能となる。尚、特別役と小役を同時に引き込める場合には、小役のみを引き込み、特別役と同時に小役が入賞ライン上に揃わないようになっている。20

#### 【 0 1 8 3 】

尚、本実施例では、特別役が前ゲーム以前から持ち越されている状態で小役が当選した場合や新たに特別役と小役が同時に当選した場合など、特別役と小役が同時に当選している場合には、当選した特別役よりも当選した小役が優先され、小役が引き込めない場合のみ、特別役を入賞ライン上に揃える制御を行っているが、特別役と小役が同時に当選している場合に、小役よりも特別役を入賞ライン上に揃える制御が優先され、特別役を引き込めない場合にのみ、小役を入賞ライン上に揃える制御を行っても良い。30

#### 【 0 1 8 4 】

特別役が前ゲーム以前から持ち越されている状態で再遊技役が当選した場合など、特別役と再遊技役が同時に当選している場合には、停止操作が行われた際に、入賞ライン上に最大 4 コマの引込範囲で再遊技役の図柄を揃えて停止させる制御が行われる。尚、この場合、再遊技役を構成する図柄または同時当選する再遊技役を構成する図柄は、リール 2 L、2 C、2 R のいずれについても 5 図柄以内、すなわち 4 コマ以内の間隔で配置されており、4 コマの引込範囲で必ず任意の位置に停止させることができるので、特別役と再遊技役が同時に当選している場合には、遊技者によるストップスイッチ 8 L、8 C、8 R の操作タイミングに関わらずに、必ず再遊技役が揃って入賞することとなる。すなわちこのような場合には、特別役よりも再遊技役を入賞ライン上に揃える制御が優先され、必ず再遊技役が入賞することとなる。尚、特別役と再遊技役を同時に引き込める場合には、再遊技役のみを引き込み、再遊技役と同時に特別役が入賞ライン上に揃わないようになっている。

#### 【 0 1 8 5 】

本実施例においてメイン制御部 4 1 は、リール 2 L、2 C、2 R の回転が開始した後、ストップスイッチ 8 L、8 C、8 R の操作が検出されるまで、停止操作が未だ検出されて4050

いないリールの回転を継続し、ストップスイッチ 8 L、8 C、8 R の操作が検出されたことを条件に、対応するリールに表示結果を停止させる制御を行うようになっている。尚、リール回転エラーの発生により、一時的にリールの回転が停止した場合でも、その後リール回転が再開した後、ストップスイッチ 8 L、8 C、8 R の操作が検出されるまで、停止操作が未だ検出されていないリールの回転を継続し、ストップスイッチ 8 L、8 C、8 R の操作が検出されたことを条件に、対応するリールに表示結果を停止させる制御を行うようになっている。

#### 【0186】

尚、本実施例では、ストップスイッチ 8 L、8 C、8 R の操作が検出されたことを条件に、対応するリールに表示結果を停止させる制御を行うようになっているが、リールの回転が開始してから、予め定められた自動停止時間が経過した場合に、リールの停止操作がなされない場合でも、停止操作がなされたものとみなして自動的に各リールを停止させる自動停止制御を行うようにしても良い。この場合には、遊技者の操作を介さずリールが停止することとなるため、例え、いずれかの役が当選している場合でもいずれの役も構成しない表示結果を導出させることが好ましい。

#### 【0187】

次に、メイン制御部 4 1 が演出制御基板 9 0 に対して送信するコマンドについて説明する。

#### 【0188】

本実施例では、メイン制御部 4 1 が演出制御基板 9 0 に対して、B E T コマンド、クレジットコマンド、内部当選コマンド、リール回転開始コマンド、リール停止コマンド、入賞判定コマンド、払出開始コマンド、払出終了コマンド、遊技状態コマンド、待機コマンド、打止コマンド、エラーコマンド、復帰コマンド、設定開始コマンド、確認開始コマンド、確認終了コマンド、操作検出コマンドを含む複数種類のコマンドを送信する。

#### 【0189】

これらコマンドは、コマンドの種類を示す 1 バイトの種類データとコマンドの内容を示す 1 バイトの拡張データとからなり、サブ制御部 9 1 は、種類データからコマンドの種類を判別できるようになっている。

#### 【0190】

B E T コマンドは、メダルの投入枚数、すなわち賭数の設定に使用されたメダル枚数を特定可能なコマンドであり、ゲーム終了後（設定変更後）からゲーム開始までの状態であり、規定数の賭数が設定されていない状態において、メダルが投入されるか、1 枚 B E T スイッチ 5 または M A X B E T スイッチ 6 が操作されて賭数が設定されたときに送信される。また、B E T コマンドは、賭数の設定操作がなされたときに送信されるので、B E T コマンドを受信することで賭数の設定操作がなされたことを特定可能である。

#### 【0191】

クレジットコマンドは、クレジットとして記憶されているメダル枚数を特定可能なコマンドであり、ゲーム終了後（設定変更後）からゲーム開始までの状態であり、規定数の賭数が設定されている状態において、メダルが投入されてクレジットが加算されたときに送信される。

#### 【0192】

内部当選コマンドは、内部当選フラグの当選状況、並びに成立した内部当選フラグの種類を特定可能なコマンドであり、スタートスイッチ 7 が操作されてゲームが開始したときに送信される。また、内部当選コマンドは、スタートスイッチ 7 が操作されたときに送信されるので、内部当選コマンドを受信することでスタートスイッチ 7 が操作されたことを特定可能である。

#### 【0193】

リール回転開始コマンドは、リールの回転の開始を通知するコマンドであり、リール 2 L、2 C、2 R の回転が開始されたときに送信される。

#### 【0194】

10

20

30

40

50

リール停止コマンドは、停止するリールが左リール、中リール、右リールのいずれかであるか、該当するリールの停止操作位置の領域番号、該当するリールの停止位置の領域番号、を特定可能なコマンドであり、各リールの停止操作に伴う停止制御が行われる毎に送信される。また、リール停止コマンドは、ストップスイッチ 8 L、8 C、8 R が操作されたときに送信されるので、リール停止コマンドを受信することでストップスイッチ 8 L、8 C、8 R が操作されたことを特定可能である。

#### 【0195】

入賞判定コマンドは、入賞の有無、並びに入賞の種類、入賞時のメダルの払出枚数を特定可能なコマンドであり、全リールが停止して入賞判定が行われた後に送信される。

#### 【0196】

払出開始コマンドは、メダルの払出開始を通知するコマンドであり、入賞やクレジット（賭数の設定に用いられたメダルを含む）の精算によるメダルの払出が開始されたときに送信される。また、払出終了コマンドは、メダルの払出終了を通知するコマンドであり、入賞及びクレジットの精算によるメダルの払出が終了したときに送信される。

#### 【0197】

遊技状態コマンドは、次ゲームの遊技状態を特定可能なコマンドであり、設定変更状態の終了時及びゲームの終了時に送信される。

#### 【0198】

待機コマンドは、待機状態へ移行する旨を示すコマンドであり、1ゲーム終了後、賭数が設定されずに一定時間経過して待機状態に移行するとき、クレジット（賭数の設定に用いられたメダルを含む）の精算によるメダルの払出が終了し、払出終了コマンドが送信された後に送信される。

#### 【0199】

打止コマンドは、打止状態の発生または解除を示すコマンドであり、B B 終了後、エンディング演出待ち時間が経過した時点で打止状態の発生を示す打止コマンドが送信され、リセット操作がなされて打止状態が解除された時点で、打止状態の解除を示す打止コマンドが送信される。

#### 【0200】

エラーコマンドは、エラー状態の発生または解除、エラー状態の種類を示すコマンドであり、エラーが判定され、エラー状態に制御された時点でエラー状態の発生及びその種類を示すエラーコマンドが送信され、リセット操作がなされてエラー状態が解除された時点で、エラー状態の解除を示すエラーコマンドが送信される。

#### 【0201】

復帰コマンドは、メイン制御部 4 1 が電断前の制御状態に復帰した旨を示すコマンドであり、メイン制御部 4 1 の起動時において電断前の制御状態に復帰した際に送信される。

#### 【0202】

設定開始コマンドは、設定変更状態の開始を示すコマンドであり、設定変更状態に移行する際に送信される。また、設定変更状態への移行に伴ってメイン制御部 4 1 の制御状態が初期化されるため、設定開始コマンドによりメイン制御部 4 1 の制御状態が初期化されたことを特定可能である。

#### 【0203】

確認開始コマンドは、設定確認状態の開始を示すコマンドであり、設定確認状態に移行する際に送信される。確認終了コマンドは、設定確認状態の終了を示すコマンドであり、設定確認状態の終了時に送信される。

#### 【0204】

操作検出コマンドは、操作スイッチ類（1枚 B E T スイッチ 5、MAX B E T スイッチ 6、スタートスイッチ 7、ストップスイッチ 8 L、8 C、8 R）の検出状態（ON / OFF）を示すコマンドであり、一定間隔毎に送信される。

#### 【0205】

これらコマンドのうち設定開始コマンド、RAM異常または乱数回路異常を示すエラー

10

20

30

40

50

コマンド、復帰コマンドは、起動処理において割込が許可される前の段階で生成され、RAM507の特別ワークに割り当てられた特定コマンド送信用バッファに格納され、直ちに送信される。

#### 【0206】

設定開始コマンド、RAM異常または乱数回路異常を示すエラーコマンド、復帰コマンド、操作検出コマンド以外のコマンドは、ゲーム処理においてゲームの進行状況に応じて生成され、RAM507の特別ワークに設けられた通常コマンド送信用バッファに一時格納され、タイマ割込処理（メイン）において実行するコマンド送信処理において送信される。

#### 【0207】

操作検出コマンドは、コマンド送信処理が5回実行される毎に生成され、前述の特定コマンド送信用バッファに格納され、直ちに送信される。また、通常コマンド送信用バッファに未送信のコマンドが格納されている状態で操作検出コマンドの送信時期に到達した場合には、操作検出コマンドの送信を優先し、通常コマンド送信用バッファに格納されている未送信のコマンドは、次回のコマンド送信処理において送信されるようになっており、1度のコマンド送信処理において複数のコマンドが送信されることがないようになっている。

#### 【0208】

次に、メイン制御部41が演出制御基板90に対して送信するコマンドに基づいてサブ制御部91が実行する演出の制御について説明する。

20

#### 【0209】

サブ制御部91は、メイン制御部41からのコマンドを受信した際に、コマンド受信割込処理を実行する。コマンド受信割込処理では、RAM91cに設けられた受信用バッファに、コマンド伝送ラインから取得したコマンドを格納する。

#### 【0210】

サブ制御部91は、タイマ割込処理（サブ）において、受信用バッファに未処理のコマンドが格納されているか否かを判定し、未処理のコマンドが格納されている場合には、そのうち最も早い段階で受信したコマンドに基づいてROM91bに格納された制御パターンテーブルを参照し、制御パターンテーブルに登録された制御内容に基づいて液晶表示器51、演出効果LED52、スピーカ53、54、リールLED55等の各種演出装置の出力制御を行う。

30

#### 【0211】

制御パターンテーブルには、複数種類の演出パターン毎に、コマンドの種類に対応する液晶表示器51の表示パターン、演出効果LED52の点灯態様、スピーカ53、54の出力態様、リールLEDの点灯態様等、これら演出装置の制御パターンが登録されており、サブ制御部91は、コマンドを受信した際に、制御パターンテーブルの当該ゲームにおいてRAM91cに設定されている演出パターンに対応して登録された制御パターンのうち、受信したコマンドの種類に対応する制御パターンを参照し、当該制御パターンに基づいて演出装置の出力制御を行う。これにより演出パターン及び遊技の進行状況に応じた演出が実行されることとなる。

40

#### 【0212】

尚、サブ制御部91は、あるコマンドの受信を契機とする演出の実行中に、新たにコマンドを受信した場合には、実行中の制御パターンに基づく演出を中止し、新たに受信したコマンドに対応する制御パターンに基づく演出を実行するようになっている。すなわち演出が最後まで終了していない状態でも、新たにコマンドを受信すると、受信した新たなコマンドが新たな演出の契機となるコマンドではない場合を除いて実行していた演出はキャンセルされて新たなコマンドに基づく演出が実行されることとなる。

#### 【0213】

特に、本実施例では、演出の実行中に賭数の設定操作がなされたとき、すなわちサブ制御部91が、賭数が設定された旨を示すBETコマンドを受信したときに、実行中の演出

50

を中止するようになっている。このため、遊技者が、演出を最後まで見るよりも次のゲームを進めたい場合には、演出がキャンセルされ、次のゲームを開始できるので、このような遊技者に対して煩わしい思いをさせることがない。また、演出の実行中にクレジットまたは賭数の精算操作がなされたとき、すなわちサブ制御部91が、ゲームの終了を示す遊技状態コマンドを受信した後、ゲームの開始を示す内部当選コマンドを受信する前に、払出開始コマンドを受信した場合には、実行中の演出を中止するようになっている。クレジットや賭数の精算を行うのは、遊技を終了する場合であり、このような場合に実行中の演出を終了させることで、遊技を終了する意志があるのに、不要に演出が継続してしまわないようになっている。

## 【0214】

10

演出パターンは、内部当選コマンドを受信した際に、内部当選コマンドが示す内部抽選の結果に応じた選択率にて選択され、RAM91cに設定される。演出パターンの選択率は、ROM91bに格納された演出テーブルに登録されており、サブ制御部91は、内部当選コマンドを受信した際に、内部当選コマンドが示す内部抽選の結果に応じて演出テーブルに登録されている選択率を参照し、その選択率に応じて複数種類の演出パターンからいずれかの演出パターンを選択し、選択した演出パターンを当該ゲームの演出パターンとしてRAM91cに設定するようになっており、同じコマンドを受信しても内部当選コマンドの受信時に選択された演出パターンによって異なる制御パターンが選択されるため、結果として演出パターンによって異なる演出が行われることがある。

## 【0215】

20

次に、本実施例におけるメイン制御部41が実行する各種制御内容を、図23～図36に基づいて以下に説明する。

## 【0216】

メイン制御部41は、リセット回路49からシステムリセット信号が入力されると、図23のフローチャートに示す起動処理（メイン）を行う。また、ユーザリセット信号が入力された場合には、起動処理（メイン）のSa2のステップから処理を開始する。すなわち電源投入に伴う起動の場合のみセキュリティチェック処理を行うセキュリティモードから開始する一方、ウォッチドッグタイマ49aのタイムアップによる起動の場合には、セキュリティチェック処理を省略してROM506に記憶されたユーザプログラムを実行するユーザモードから開始されることになる。

30

## 【0217】

システムリセット信号の入力に伴う起動処理（メイン）では、まず、CPU505がROM506から読み出したセキュリティチェックプログラム506Aに基づき、図24に示すセキュリティチェック処理を実行する（Sa1）。このとき、メイン制御部41は、セキュリティモードとなり、ROM506に記憶されているゲーム制御用のユーザプログラムは未だ実行されない状態となる。

## 【0218】

40

セキュリティチェック処理では、図24に示すように、まず、セキュリティモードに制御する時間（セキュリティ時間）を決定するための処理を実行する。このとき、CPU505は、ROM506のプログラム管理エリアに記憶されるセキュリティ時間設定KSEのビット番号[2-0]におけるビット値を読み出す（Sa101）。そして、この読み出しが“000”であるか否かを判定する（Sa102）。

## 【0219】

Sa102にて読み出しが“000”と判定された場合には、定常設定時間を既定の固定時間に設定する（Sa103）。ここで、定常設定時間は、セキュリティ時間のうち、システムリセットの発生等に基づくセキュリティチェック処理の実行回数（メイン制御部41がセキュリティモードとなる回数）に関わりなく、一定となる時間成分である。また、固定時間は、セキュリティ時間のうち、メイン制御部41の仕様などに基づいて予め定められた不变時間成分であり、例えばセキュリティ時間として設定可能な最小値となるものであれば良い。

50

**【0220】**

Sa102にて読み出しが“000”以外と判定された場合には、その読み出しが対応して、固定時間に加えて図10(D)に示す設定内容により選択される延長時間を、定常設定時間に設定する(Sa104)。こうして、セキュリティ時間設定KSESのビット番号[2-0]におけるビット値が“000”以外の値である場合には、セキュリティチェック処理の実行時間であるセキュリティ時間を、固定時間に加えて予め選択可能な複数の延長時間のいずれかに設定することができる。

**【0221】**

Sa103、Sa104の処理のいずれかを実行した後には、セキュリティ時間設定KSESのビット番号[4-3]におけるビット値を読み出す(Sa105)。そして、この読み出しが“00”であるか否かを判定する(Sa106)。

10

**【0222】**

Sa106にて読み出しが“00”と判定された場合には、定常設定時間をセキュリティ時間に設定する(Sa107)。これに対して、読み出しが“00”以外と判定された場合には、その読み出しが対応して決定される可変設定時間を、定常設定時間に加算してセキュリティ時間に設定する(Sa108)。ここで、可変設定時間は、セキュリティ時間のうち、セキュリティチェック処理が実行されるごとに変化する時間成分であり、セキュリティ時間設定KSESのビット番号[4-3]におけるビット値が“01”(ショートモード)であるか“10”(ロングモード)であるかに応じて異なる所定の時間範囲で変化する。例えば、システムリセットの発生時に、所定のフリーランカウンタにおけるカウント値がメイン制御部41に内蔵された可変セキュリティ時間用レジスタに格納される場合には、Sa108の処理において、可変セキュリティ時間用レジスタの格納値をそのまま用いること、あるいは、その格納値を所定の演算関数(例えばハッシュ関数)に代入して得られた値を用いることなどにより、可変設定時間がシステムリセット毎に所定の時間範囲でランダムに変化するように決定されれば良い。こうして、セキュリティ時間設定KSESのビット番号[4-3]におけるビット値が“00”以外の値である場合には、セキュリティチェック処理の実行時間であるセキュリティ時間を、システムリセットの発生等に基づくセキュリティチェック処理が実行されるごとに所定の時間範囲で変化させることができる。

20

**【0223】**

30

ここで、セキュリティ時間設定KSESのビット番号[2-0]におけるビット値が“000”以外の値であり、かつ、セキュリティ時間設定KSESのビット番号[4-3]におけるビット値が“00”以外の値である場合には、ステップS4にて設定される延長時間と、ステップS8にて設定される可変設定時間との双方が、固定時間に加算されて、セキュリティ時間が決定されることになる。

**【0224】**

Sa107、Sa108の処理のいずれかを実行した後には、ROM506の所定領域に記憶されたセキュリティコードを読み出す(Sa109)。ここで、ROM506の所定領域には、記憶内容のデータを所定の演算式によって演算した演算結果のセキュリティコードが予め記憶されている。セキュリティコードの生成方法としては、例えばハッシュ関数を用いてハッシュ値を生成するもの、エラー検出コード(CRCコード)を用いるもの、エラー訂正コード(ECCコード)を用いるもののいずれかといった、予め定められた生成方法を使用すれば良い。また、ROM506のセキュリティコード記憶領域とは異なる所定領域には、セキュリティコードを演算により特定するための演算式が、暗号化して予め記憶されている。

40

**【0225】**

ステップSa109の処理に続いて、暗号化された演算式を復号化して元に戻す(Sa110)。その後、Sa110で復号化した演算式により、ROM506の所定領域における記憶データを演算してセキュリティコードを特定する(Sa111)。このときセキュリティコードを特定するための演算に用いる記憶データは、例えばROM506の記憶

50

データのうち、セキュリティチェックプログラム 506A とは異なるユーザプログラムの全部又は一部に相当するプログラムデータ、あるいは、所定のテープルデータを構成する固定データの全部又は一部であれば良い。そして、Sa109 にて読み出したセキュリティコードと、Sa111 にて特定されたセキュリティコードとを比較する (Sa112)。このときには、比較結果においてセキュリティコードが一致したか否かを判定する (Sa113)。

#### 【0226】

Sa113 にてセキュリティコードが一致しない場合には、ROM506 に不正な変更が加えられたと判断して、CPU505 の動作を停止状態 (HALT) へ移行させる。これに対して、Sa113 にてセキュリティコードが一致した場合には、プログラム管理エリアに記憶されている第1乱数初期設定 KRS1 や第2乱数初期設定 KRS2 を読み出して、図25 に示す乱数回路設定処理を実行した後 (Sa114)、乱数回路 509 における動作異常の有無を検査するために図26 に示す乱数回路異常検査処理を実行する (Sa115)。

10

#### 【0227】

図25 に示す乱数回路設定処理では、まず、第1乱数初期設定 KRS1 のビット番号 [3] におけるビット値に基づき、乱数回路 509 を使用するための設定を行う (Sa201)。本実施例では、第1乱数初期設定 KRS1 のビット番号 [3] におけるビット値が予め “1” とされており、このビット値に対応して乱数回路 509 を使用する設定が行われる。続いて、第1乱数初期設定 KRS1 のビット番号 [2] におけるビット値に基づき、乱数回路 509 における乱数更新クロック RGK の設定を行う (Sa202)。本実施例では、第1乱数初期設定 KRS1 のビット番号 [2] におけるビット値が予め “1” とされていることに対応して、乱数用クロック生成回路 43 で生成された乱数用クロック RCLK を2分周して乱数更新クロック RGK とする設定が行われる。

20

#### 【0228】

Sa202 での設定を行った後には、第1乱数初期設定 KRS1 のビット番号 [1-0] におけるビット値に基づき、乱数回路 509 における乱数更新規則の設定を行う (Sa203)。本実施例では、第1乱数初期設定 KRS1 のビット番号 [1-0] におけるビット値として予め “10” が設定されており、乱数列 RSN における乱数更新規則を2周目以降は自動で変更する設定がなされる。

30

#### 【0229】

続いて、第2乱数初期設定 KRS2 のビット番号 [1] におけるビット値に基づき、乱数値となる数値データにおける起動時スタート値を決定する (Sa204)。本実施例では、第2乱数初期設定 KRS2 のビット番号 [1] におけるビット値として予め “1” が設定されており、乱数のスタート値をIDナンバーに基づく値とする設定がなされる。

#### 【0230】

さらに、第2乱数初期設定 KRS2 のビット番号 [0] におけるビット値に基づき、乱数値となる数値データのスタート値をシステムリセット毎に変更するか否かの設定を行う (Sa205)。本実施例では、第2乱数初期設定 KRS2 のビット番号 [0] におけるビット値が予め “0” とされており、Sa204 にて設定した起動時スタート値をそのまま用いて、乱数回路 509 におけるスタート値としている。Sa205 の処理による設定が完了すると、乱数回路 509 では乱数値の生成動作が開始されることとなる。尚、第2乱数初期設定 KRS2 のビット番号 [0] におけるビット値が予め “1” とされている場合には、乱数のスタート値をシステムリセット毎に変更する設定がなされる。この場合には、例えば、システムリセットの発生時に、所定のフリーランカウンタにおけるカウント値がメイン制御部 41 に内蔵された乱数スタート値用レジスタに格納される場合には、Sa205 の処理において、乱数スタート値用レジスタの格納値をそのまま用いることなどにより、乱数のスタート値がシステムリセット毎に所定の数値範囲（例えば乱数生成回路 553 にて生成されるカウント値順列 RCN に含まれる数値データの全部又

40

50

は一部を含む範囲)でランダムに変化するように決定されれば良い。

#### 【0231】

尚、乱数回路設定処理による設定は、CPU505の処理が介在することなく、乱数回路509がプログラム管理エリアの記憶データに基づき自律的に行うようにしても良い。この場合、乱数回路509は、メイン制御部41がセキュリティモードとなっているときには初期設定を行わず、乱数値の生成動作が行われないようにもしても良い。そして、メイン制御部41にてCPU505がROM506に記憶されたユーザプログラムを読み出してユーザモードが開始されたときに、例えばCPU505から乱数回路509に対して初期設定を指示する制御信号が伝送されたことなどに応答して、乱数回路509が初期設定を行ってから乱数値の生成動作を開始するようにしても良い。また、特に乱数回路509がメイン制御部41に外付けされる場合などには、メイン制御部41がセキュリティモードとなっているときでも、乱数回路509がCPU505における処理とは独立して、プログラム管理エリアの記憶データに基づく初期設定を行ってから、乱数値の生成動作を開始するようにしても良い。10

#### 【0232】

図26に示す乱数回路異常検査処理では、まず乱数用クロック異常判定カウンタをクリアして、そのカウント値である乱数用クロック異常判定カウント値を「0」に初期化する(Sa301)。続いて、内部情報レジスタCIFのビット番号[4]に格納されている内部情報データCIF4を読み出す(Sa302)。そして、Sa302での読み出しが“1”であるか否かを判定する。このとき、乱数回路509が備える周波数監視回路551によって、乱数用外部クロック端子ERCからの乱数用クロックRCLKに周波数異常が検知された場合には、内部情報データCIF4としてビット値“1”が書き込まれる。20

#### 【0233】

そこで、Sa303にて読み出しが“1”と判定された場合には、乱数用クロック異常判定カウント値を1加算するように更新する(Sa304)。このときには、Sa304での更新後におけるカウント値が所定のクロック異常判定値に達したか否かを判定する(Sa305)。ここで、クロック異常判定値は、周波数監視回路551により乱数用クロックRCLKの周波数異常が連続して検知された場合にクロック異常と判定するために予め定められた数値であれば良い。Sa305にてクロック異常判定値に達していないければ、Sa302の処理に戻り、再び内部情報データCIF4のビット値に基づく判定を行う。30

#### 【0234】

Sa305にてクロック異常判定値に達した場合には、乱数用クロックが正常に機能していないと判断して、CPU505の動作を停止状態(HALT)へ移行させる。

#### 【0235】

Sa303にて読み出しが“0”と判定された場合には、乱数値異常判定カウンタをクリアして、そのカウント値である乱数値異常判定カウント値を「0」に初期化する(Sa306)。尚、Sa303の処理では、Sa302にて読み出した内部情報データCIF4のビット値が複数回(例えば2回など)連続して“0”となったときに、読み出しが“0”であると判定しても良い。

#### 【0236】

Sa306の処理に続いて、乱数値における異常の有無をチェックするために用いるチェック値を、初期値「0000H」に設定する(Sa307)。そして、乱数回路509が備える乱数値レジスタR1Dとなる乱数値レジスタ559Aから、格納されている乱数値となる数値データを読み出す(Sa308)。例えば、Sa308の処理では、乱数ラッチ選択レジスタRDLのビット番号[0]に格納される乱数ラッチ選択データRDLのビット値を“0”として、ソフトウェアによる乱数値の取り込みを指定する。続いて、乱数値取込指定レジスタRDLTのビット番号[0]に格納される乱数値取込指定データRDLTのビット値を“1”として、乱数値レジスタR1Dへの取り込みを指定する。なお、乱数値取込指定レジスタRDLTのビット番号[0]におけるビット値を“1”とすることは、CPU505から乱数回路509に対して数値データの取り込み(ラッ40  
50

チ)を指示するラッチ信号を出力することに相当する。その後、乱数値レジスタ R 1 D に供給するレジスタリード信号 R R S 1 をオン状態とすることにより、格納されている乱数値となる数値データを読み出すようにすれば良い。

#### 【0237】

S a 3 0 8 にて数値データを読み出した後には、その読出値を乱数検査基準値に設定する(S a 3 0 9)。続けて、さらに乱数値レジスタ 5 5 9 A から乱数値となる数値データを読み出す(S a 3 1 0)。尚、S a 3 1 0 での読出動作は、S a 3 0 8 での読出動作と同様の手順で行われれば良い。また、S a 3 0 8 での読出動作と、S a 3 1 0 での読出動作との間には、乱数回路 5 0 9 で生成される乱数列 R S N における数値データが変化するために十分な遅延時間を設けると良い。S a 3 1 0 にて数値データを読み出した後には、乱数検査基準値と、S a 3 1 0 での読出値との排他的論理和演算を実行する(S a 3 1 1)。また、S a 3 1 1 での演算結果と、チェック値との論理和演算を実行し、演算結果を新たなチェック値とするように更新させる(S a 3 1 2)。例えば、チェック値は R A M 5 0 7 の所定領域に記憶しておき、S a 3 1 2 の処理が実行される毎に、その処理で得られた演算結果を新たなチェック値として保存すれば良い。これにより、乱数値レジスタ 5 5 9 A から読み出した数値データにおける全ビットの変化が記録され、複数回の読出中に少なくとも 1 回は値が変化したビットであれば、チェック値において対応するビット値が“1”となる。

#### 【0238】

そこで、チェック値が「F F F F H」となったか否かを判定し(S a 3 1 3)、なっていれば、全ビットについてビット値の変化が認められることから、乱数値が正常に更新されていると判断して、乱数回路異常検査処理を終了する。尚、乱数値が正常に更新されていることを確認できた場合には、乱数ラッチ選択レジスタ R D L S のビット番号[0]に格納される乱数ラッチ選択データ R D L S 0 のビット値を“1”として、入力ポート P 0 への信号入力に応じた乱数値レジスタ R 1 D への乱数値取込を、指示するよう設定される。本実施例では、入力ポート P 0 にスタートスイッチ 7 からのゲーム開始信号 S S 1 を伝送する配線が接続される。これにより、ゲーム開始信号 S S 1 がオン状態となったときに乱数値レジスタ R 1 D への乱数値取込を行うことができる。

#### 【0239】

S a 3 1 3 にてチェック値が「F F F F H」以外と判定された場合には、乱数値異常判定カウント値を 1 加算するように更新する(S a 3 1 4)。このときには、S a 3 1 4 での更新後におけるカウント値が所定の乱数値異常判定値に達したか否かを判定する(S a 3 1 5)。ここで、乱数値異常判定値は、乱数回路 5 0 9 が正常動作していれば、乱数値レジスタ 5 5 9 A から読み出される数値データの全ビットが少なくとも 1 回は変化するのに十分な判定回数となるように、予め定められた数値であれば良い。S a 3 1 5 にて乱数値異常判定値に達していないければ、S a 3 1 0 の処理に戻り、再び乱数回路 5 0 9 から乱数値となる数値データを読み出して異常の有無をチェックするための判定などを行う。

#### 【0240】

S a 3 1 5 にて乱数値異常判定値に達した場合には、乱数用クロックが正常に機能していないと判断して、C P U 5 0 5 の動作を停止状態(HALT)へ移行させる。

#### 【0241】

尚、乱数回路異常検査処理は、C P U 5 0 5 が実行するものに限定されず、C P U 5 0 5 以外のメイン制御部 4 1 における内蔵回路により乱数回路異常検査処理が実行されても良い。一例として、乱数回路 5 0 9 が乱数回路異常検査処理を実行する機能を有し、乱数用クロック R C L K の周波数異常が検知されたときや、乱数値の異常が検知されたときには、エラーの発生を C P U 5 0 5 に通知するようにしても良い。また、乱数回路異常検査処理は、起動時に実行されるものに限定されず、例えばメイン制御部 4 1 にてタイマ割込みが発生する毎に、乱数回路異常検査処理の一部又は全部が実行されるようにしても良い。

#### 【0242】

図 2 4 に戻り、S a 1 1 4 の乱数回路設定処理及び S a 1 1 5 の乱数回路異常検査処理

10

20

30

40

50

の後、Sa107やSa108の処理で設定されたセキュリティ時間が経過したか否かを判定する(Sa116)。このとき、セキュリティ時間が経過していないければ、Sa116の処理を繰り返し実行して、セキュリティ時間が経過するまで待機する。そして、Sa116にてセキュリティ時間が経過したと判定された場合には、セキュリティチェック処理を終了し、CPU505に内蔵されたプログラムカウンタの値をROM506におけるユーザプログラムの先頭アドレス(アドレス0000H)に設定することなどにより、メイン制御部41の動作状態がセキュリティモードからユーザモードへと移行し、ROM506に記憶されたユーザプログラムの実行が開始されることになる。尚、前述のようにユーザリセット信号の入力に伴う起動時には、セキュリティモードを経ずにユーザモードから開始することとなる。

10

#### 【0243】

ユーザモードではまず、内蔵デバイスや周辺IC、割込モード、スタックポインタ等を初期化した後(Sa2)、Iレジスタ及びIYレジスタの値を初期化する(Sa3)。Iレジスタ及びIYレジスタの初期化により、Iレジスタには、割込発生時に参照する割込テーブルのアドレスが設定され、IYレジスタには、RAM507の格納領域を参照する際の基準アドレスが設定される。これらの値は、固定値であり、起動時には常に初期化されることとなる。

#### 【0244】

次いで、RAM507へのアクセスを許可し(Sa4)、RAM507の全ての格納領域(未使用領域及び未使用スタック領域を含む)のRAMパリティを計算する(Sa5)。ついで、打止スイッチ36\_a、自動精算スイッチ36\_bの状態を取得し、メイン制御部41の特定のレジスタに打止機能、自動精算機能の有効/無効を設定した後(Sa6)。後述するポート入力処理において取得した各スイッチの入力データ、前回と今回の入力データが同じ状態を示す各スイッチの確定データ、前回と今回の確定データが異なる状態を示す各スイッチのエッジデータをそれぞれクリアし(Sa7)、さらに停電が検知された旨を示す電断フラグをクリアする(Sa8)。

20

#### 【0245】

次いで、Sa5のステップにおいて計算したRAMパリティが0か否かを判定する(Sa9)。正常に電断割込処理(メイン)が行われていれば、RAMパリティが0になるはずであり、Sa9のステップにおいてRAMパリティが0でなければ、RAM507に格納されているデータが正常ではなく、この場合には、RAM507の格納領域のうち、使用中スタック領域を除く全ての格納領域を初期化する初期化1を実行した後(Sa23)、設定キースイッチ37がONか否かを判定し(Sa24)、設定キースイッチ37がONであれば、設定開始コマンドを特定コマンド送信用バッファに設定し、サブ制御部91に対して送信する(Sa21)。次いで、割込を許可して(Sa22)、設定変更処理、すなわち設定変更状態に移行する。

30

#### 【0246】

Sa24のステップにおいて設定キースイッチ37がOFFであれば、RAM異常を示すエラーコードをレジスタに設定し(Sa25)、RAM異常を示すエラーコマンドを特定コマンド送信用バッファに設定し、サブ制御部91に対して送信し(Sa26)、割込を許可して(Sa27)、エラー処理、すなわちRAM異常エラー状態に移行する。

40

#### 【0247】

Sa9のステップにおいて、RAMパリティが0であれば、更に破壊診断用データが正常か否かを判定する(Sa10)。正常に電断処理(メイン)が行われていれば、破壊診断用データが設定されているはずであり、Sa10のステップにおいて破壊診断用データが正常でない場合(破壊診断用データが電断時に格納される5A(H)以外の場合)にも、RAM507のデータが正常ではないので、Sa23のステップに移行して初期化1を実行し、その後、Sa24のステップにおいて設定キースイッチ37がONであれば、設定開始コマンドの送信(Sa21)の後、割込を許可して(Sa22)、設定変更処理に移行する。また、Sa24のステップにおいて設定キースイッチ37がOFFであれば、

50

R A M異常を示すエラーコードの設定(S a 2 5)、R A M異常を示すエラーコマンドの送信(S 2 6)の後、割込を許可して(S a 2 7)、エラー処理に移行する。

#### 【0 2 4 8】

S a 1 0 のステップにおいて破壊診断用データが正常であると判定した場合には、R A M 5 0 7 のデータは正常であるので、破壊診断用データをクリアし(S a 1 1)、R A M 5 0 7 の非保存ワーク、未使用領域及び未使用スタック領域を初期化する初期化3を行った後(S a 1 2)、設定キースイッチ3 7 がONか否かを判定し(S a 1 3)、設定キースイッチ3 7 がONであれば、S a 2 0 のステップに移行して初期化1を実行し、設定開始コマンドの送信(S a 2 1)の後、割込を許可して(S a 2 2)、設定変更処理に移行する。

10

#### 【0 2 4 9】

S a 1 3 のステップにおいて設定キースイッチ3 7 がOFFであれば、各レジスタを電断前の状態、すなわちスタックに保存されている状態に復帰し(S a 1 4)、復帰コマンドを特定コマンド送信用バッファに設定し、サブ制御部9 1 に対して送信する(S a 1 5)。そして、乱数ラッチフラグレジスタの値に基づいて乱数値がラッチされているか否か、すなわち乱数値レジスタ5 5 9 A に数値データが取り込まれているか否かを判定し(S a 1 6)、乱数値がラッチされていなければS a 1 8 のステップに進み、割込を許可して電断前の最後に実行していた処理に戻る。S a 1 6 のステップにおいて乱数値がラッチされていれば、乱数値レジスタ5 5 9 A から内部抽選用の乱数値を読み出し(S a 1 7)、S a 1 8 のステップに進み、割込を許可して電断前の最後に実行していた処理に戻る。尚、S a 1 7 のステップにおいて乱数値レジスタ5 5 9 から内部抽選用の乱数値が読み出されると乱数ラッチフラグレジスタがクリアされ、新たな数値データの取込が許可されることとなる。また、S a 1 7 のステップにおいては、内部抽選用の乱数値を読み出すものの、読み出した乱数値を用いる訳ではなく、スタートスイッチ7の操作に応じて新たな乱数値の取込を可能とするためにダミーとして読み出すものである。

20

#### 【0 2 5 0】

このように起動処理では、システムリセット時のセキュリティモードの後、ユーザモードに移行するのに対してユーザリセット時には、セキュリティモードを省き、ユーザモードから開始するようになっている。

30

#### 【0 2 5 1】

システムリセット時のセキュリティモードでは、乱数回路5 0 9 の設定を行う乱数回路設定処理を行うとともに、乱数回路設定処理において乱数の更新規則の設定や乱数のスタート値の設定などが行われる一方でユーザリセット時には、セキュリティモードは省略され、ユーザモードから開始することとなる。このため、システムリセット時には、乱数の初期値が設定し直されるのに対して、ユーザリセット時には、乱数回路5 0 9 の設定は行われず、リセット前からの状態のまま乱数となる数値データの更新が継続されることとなる。また、セキュリティモードの間は、割込が許可されることなく、ユーザモードへ移行した後に割込が許可されるので、システムリセットにおいては、割込が許可されるよりも前に、これら乱数回路5 0 9 の設定が行われることとなる。

40

#### 【0 2 5 2】

また、システムリセットであるかユーザリセットであるかに関わらず、R A M 5 0 7へのアクセスが許可された後、他の処理を行うことなく、まず、R A M 5 0 7 の全ての格納領域(未使用領域及び未使用スタック領域を含む)のR A Mパリティを計算し、その後、計算したR A Mパリティから格納データが正常か否かの判定を行う前に、R A M 5 0 7 の格納データが正常か否かに関わらず共通して行う必要のある処理として、打止機能、自動精算機能の有効/無効の設定、各スイッチの入力データ、確定データ、エッジデータのクリア、電断フラグのクリアを行い、その後、R A Mパリティに基づいてR A M 5 0 7 の格納データが正常か否かの判定を行うようになっている。

#### 【0 2 5 3】

図2 7は、メイン制御部4 1 が実行するエラー処理の制御内容を示すフローチャートで

50

ある。

**【0254】**

エラー処理では、まず、現在の遊技補助表示器12の表示状態をスタックに退避し(Sb1)、レジスタに格納されているエラーコードを遊技補助表示器12に表示し(Sb2)、エラーコードがRAM異常エラーであるか否かを判定し(Sb3)、RAM異常エラーを示すエラーコードである場合には、いずれの処理も行わないループ処理に移行する。

**【0255】**

Sb3のステップにおいてRAM異常エラーを示すエラーコードではないと判定された場合には、エラー状態の発生及びその種類を示すエラーコマンドを通常コマンド送信用バッファに設定する(Sb4)。Sb4のステップで設定されたエラーコマンドは、その後のタイマ割込処理(メイン)にてサブ制御部91に対して送信される。10

**【0256】**

次いで、リセット/設定スイッチ38の操作が検出されているか否かを判定し(Sb5)、リセット/設定スイッチ38の操作が検出されていなければ、更にリセットスイッチ23の操作が検出されているか否かを判定し(Sb6)、リセットスイッチ23の操作も検出されていなければ、Sb5のステップに戻る。すなわちリセット/設定スイッチ38またはリセットスイッチ23の操作が検出されるまで、遊技の進行が不能な状態で待機する。

**【0257】**

そして、Sb5のステップにおいてリセット/設定スイッチ38の操作が検出された場合、またはSb6のステップにおいてリセットスイッチ23の操作が検出された場合には、レジスタに格納されているエラーコードをクリアし(Sb7)、遊技補助表示器12の表示状態をSb1のステップにおいてスタックに退避した表示状態に復帰させ(Sb8)、エラー状態が解除された旨を示すエラーコマンドを通常コマンド送信用バッファに設定して(Sb9)、もとの処理に戻る。Sb9で設定されたエラーコマンドは、その後のタイマ割込処理(メイン)にてサブ制御部91に対して送信される。20

**【0258】**

このようにエラー処理においては、RAM異常以外によるエラー処理であれば、リセット/設定スイッチ38またはリセットスイッチ23が操作されることで、エラー状態を解除してもとの処理に復帰するが、RAM異常によるエラー処理であれば、リセット/設定スイッチ38またはリセットスイッチ23が操作されてもエラー状態が解除され、元の状態に復帰することはない。30

**【0259】**

図28は、メイン制御部41が実行する設定変更処理の制御内容を示すフローチャートである。

**【0260】**

設定変更処理では、まず、RAM507の設定値ワークに格納されている設定値を読み出して、読み出した値を表示値とし(Sc1)、表示値が設定可能な範囲(1~6)外か否かを判定し(SC2)、表示値が設定可能な範囲内であればSc4のステップに進み、表示値が設定可能な範囲外であれば、表示値を1に補正し、Sc5のステップに進む。40

**【0261】**

Sc5のステップでは、設定値表示器24に表示値を表示させた後、リセット/設定スイッチ38とスタートスイッチ7の操作の検出待ちの状態となり(SC5、SC6)、Sc5のステップにおいてリセット/設定スイッチ38のonが検出されると、操作スイッチの立上りを示す立上りエッジをクリアし(SC7)、表示値を1加算し(SC8)、Sc2のステップに戻る。

**【0262】**

また、Sc6のステップにおいてスタートスイッチ7のonが検出された場合には、立上りエッジをクリアし(SC9)、設定値表示器24に表示されている値を0に更新し(SC10)、設定キースイッチ37がOFFの状態となるまで待機する(SC11)。50

**【0263】**

S c 1 1 のステップにおいて設定キースイッチ 3 7 の O F F が判定されると、表示値を設定値ワークに格納して ( S c 1 2 ) 、遊技状態コマンドを生成し、コマンド送信用バッファに設定し ( S c 1 3 ) 、ゲーム処理に移行する。 S c 1 3 のステップにおいて設定された遊技状態コマンドは、その後のタイマ割込処理 ( メイン ) にてサブ制御部 9 1 に対して送信される。

**【0264】**

図 2 9 は、メイン制御部 4 1 が実行するゲーム処理の制御内容を示すフローチャートである。

**【0265】**

ゲーム処理では、 B E T 処理 ( S d 1 ) 、内部抽選処理 ( S d 2 ) 、リール回転処理 ( S d 3 ) 、入賞判定処理 ( S d 4 ) 、払出処理 ( S d 5 ) 、ゲーム終了時処理 ( S d 6 ) を順に実行し、ゲーム終了時処理が終了すると、再び B E T 処理に戻る。

10

**【0266】**

S d 1 のステップにおける B E T 処理では、賭数を設定可能な状態で待機し、遊技状態に応じた規定数の賭数が設定され、スタートスイッチ 7 が操作された時点でゲームを開始させる処理を実行する。

**【0267】**

S d 2 のステップにおける内部抽選処理では、 S d 1 のステップにおけるスタートスイッチ 7 の検出によるゲーム開始と同時にラッ奇された内部抽選用の乱数値に基づいて上記した各役への入賞を許容するかどうかを決定する処理を行う。この内部抽選処理では、それぞれの抽選結果に基づいて、 R A M 5 0 7 に当選フラグが設定される。

20

**【0268】**

S d 3 のステップにおけるリール回転処理では、各リール 2 L 、 2 C 、 2 R を回転させる処理、遊技者によるストップスイッチ 8 L 、 8 C 、 8 R の操作が検出されたことに応じて対応するリール 2 L 、 2 C 、 2 R の回転を停止させる処理を実行する。

**【0269】**

S d 4 のステップにおける入賞判定処理では、 S d 3 のステップにおいて全てのリール 2 L 、 2 C 、 2 R の回転が停止したと判定した時点で、各リール 2 L 、 2 C 、 2 R に導出された表示結果に応じて入賞が発生したか否かを判定する処理を実行する。

30

**【0270】**

S d 5 のステップにおける払出処理では、 S d 4 のステップにおいて入賞の発生が判定された場合に、その入賞に応じた払出枚数に基づきクレジットの加算並びにメダルの払出等の処理を行う。

**【0271】**

S d 6 のステップにおけるゲーム終了時処理では、次のゲームに備えて遊技状態を設定する処理を実行する。

**【0272】**

また、ゲーム処理では、ゲームの進行制御に応じてコマンドを生成し、通常コマンド送信用バッファに設定することで、その後のタイマ割込処理 ( メイン ) においてサブ制御部 9 1 に対して設定されたコマンドが送信されるようになっている。

40

**【0273】**

図 3 0 ~ 図 3 2 は、メイン制御部 4 1 が S d 1 のステップにおいて実行する B E T 処理の制御内容を示すフローチャートである。

**【0274】**

B E T 処理では、まず、 R A M 5 0 7 において賭数の値が格納される B E T カウンタの値をクリアし ( S e 1 ) 、遊技状態に応じた規定数 ( 本実施例では遊技状態に関わらず 3 ) を R A M 5 0 7 に設定し ( S e 2 ) 、 R A M 5 0 7 にリプレイゲームである旨を示すリプレイゲームフラグが設定されているか否かに基づいて当該ゲームがリプレイゲームであるか否かを判定する ( S e 3 ) 。

50

**【0275】**

S e 3 のステップにおいて当該ゲームがリプレイゲームであると判定された場合には、賭数が 3 加算された旨を示す B E T コマンドを通常コマンド送信用バッファに設定し ( S e 4 )、 B E T カウンタの値を 1 加算し ( S e 5 )、 R A M 5 0 7 に設定された賭数の規定数を参照し、 B E T カウンタの値が規定数であるか否か、すなわちゲームの開始条件となる賭数が設定されているか否かを判定し ( S e 6 )、 B E T カウンタの値が規定数でなければ S e 5 のステップに戻り、 B E T カウンタの値が規定数であれば、メダルの投入不可を示す投入不可フラグを R A M 5 0 7 に設定し ( S e 7 )、 S e 1 2 のステップに進む。 S e 4 のステップで設定された B E T コマンドは、その後のタイマ割込処理 ( メイン ) でサブ制御部 9 1 に対して送信される。

10

**【0276】**

S e 3 のステップにおいて当該ゲームがリプレイゲームでないと判定されれば、投入待ち前の設定を行い ( S e 8 )、 S e 9 のステップに進む。投入待ち前の設定では、 R A M 5 0 7 に設定されている投入不可フラグをクリアし、メダルの投入が可能な状態とする。

**【0277】**

S e 9 のステップでは、 B E T カウンタの値が規定数であるか否か、すなわちゲームの開始条件となる賭数が設定されているか否かを判定し、 B E T カウンタの値が規定数でなければ S e 1 4 のステップに進み、 B E T カウンタの値が規定数であれば、さらにいずれかのスイッチが O F F に変化したか否かを判定する ( S e 1 0 )。 S e 1 0 のステップでは、いずれかのスイッチの立下りを示す立下りエッジが設定されているか否かに基づいていずれかのスイッチが O F F に変化したか否かが判定される。さらに立下りエッジは、いずれかのスイッチが O F F に変化し、かつ全てのスイッチが O F F である場合にのみ設定されるので、 S e 1 0 のステップでは、いずれかのスイッチが O F F に変化したか否かに加えて他のスイッチが O F F であるかどうかについても判定されることとなる。

20

**【0278】**

S e 1 0 のステップにおいていずれのスイッチも O F F に変化していないと判定された場合、またはいずれかのスイッチが O F F に変化したものの未だ O N が継続しているスイッチがある場合には、 S e 1 4 のステップに進み、いずれかのスイッチが O F F に変化し、かつ全てのスイッチが O F F であると判定された場合には、立下りエッジをクリアし ( S e 1 1 )、 S e 1 2 のステップに進む。

30

**【0279】**

S e 1 2 のステップでは、乱数ラッチフラグレジスタの値に基づいて乱数値がラッチされているか否か、すなわち乱数値レジスタ 5 5 9 A に数値データが取り込まれているか否かを判定し、乱数値がラッチされていなければ S e 1 4 のステップに進み、乱数値がラッチされていれば乱数値レジスタ 5 5 9 A から内部抽選用の乱数値を読み出し ( S e 1 3 )、 S e 1 4 のステップに進む。尚、 S e 1 3 のステップにおいて乱数値レジスタ 5 5 9 から内部抽選用の乱数値が読み出されると乱数ラッチフラグレジスタがクリアされ、新たな数値データの取込が許可されることとなる。また、 S e 1 3 のステップにおいては、内部抽選用の乱数値を読み出すものの、読み出した乱数値を用いる訳ではなく、スタートスイッチ 7 の操作に応じて新たな乱数値の取込を可能とするためにダミーとして読み出すものである。このため、ゲームの開始条件となる賭数が設定されている状態であるが、スタートスイッチ 7 以外のスイッチが操作されており、スタートスイッチ 7 の操作が無効化されている状態でスタートスイッチ 7 が操作されたために乱数値がラッチされ、新たな乱数値の取込が禁止されている状態であっても、全てのスイッチが O F F の状態となり、スタートスイッチ 7 の操作が有効化された時点でラッチされている乱数値がダミーとして読み出され、その後のスタートスイッチ 7 が操作されたタイミングで新たに乱数値をラッチすることが可能となる。

40

**【0280】**

S e 1 4 のステップでは、レジスタにエラーコードが設定されているか否か、すなわちエラーが検知されたか否かを判定し、エラーコードが設定されれば、図 2 7 に示すエ

50

ラー処理に移行する。

#### 【0281】

S e 1 4 のステップにおいてエラーコードが設定されていなければ、R A M 5 0 7 に投入不可フラグが設定されているか否かに基づいてメダルの投入が可能な状態か否かを判定する ( S e 1 5 )。S e 1 5 のステップにおいてメダルの投入が可能な状態であると判定された場合には、流路切替ソレノイド 3 0 を o n の状態とし、メダルの流路をホッパー・ターンク側の経路としてメダルの投入が可能な状態とし ( S e 1 6 )、S e 1 8 のステップに進み、メダルの投入が可能な状態でないと判定された場合には、流路切替ソレノイド 3 0 を o f f の状態とし、メダルの流路をメダル払出口 9 側の経路として新たなメダルの投入を禁止し ( S e 1 7 )、S e 1 8 のステップに進む。

10

#### 【0282】

S e 1 8 のステップにおいては、設定キースイッチ 3 7 が o n の状態か否かを判定し、設定キースイッチ 3 7 が o n であれば、B E T カウンタの値が 0 か否かを判定する ( S e 1 9 )。そして、S e 1 9 のステップにおいて B E T カウンタの値が 0 であれば、確認開始コマンドを通常コマンド送信用バッファに設定し ( S e 2 0 )、設定確認処理 ( S e 2 1 )、すなわち設定確認状態に移行する。S e 2 0 のステップで設定された確認開始コマンドは、その後のタイマ割込処理 ( メイン ) でサブ制御部 9 1 に対して送信される。また、S e 2 1 のステップにおける設定確認処理が終了した後は、S e 9 のステップに戻る。

#### 【0283】

S e 1 8 のステップにおいて設定キースイッチ 3 7 が o n ではない場合または S e 1 9 のステップにおいて B E T カウンタの値が 0 ではない場合には、S e 2 2 のステップに進み、投入メダルセンサ 3 1 により投入メダルの通過が検出されたか否か、すなわち投入メダルの通過が検出された旨を示す投入メダルフラグの有無を判定する。S e 2 2 のステップにおいて投入メダルの通過が検出されていなければ、S e 3 3 のステップに進み、投入メダルの通過が検出されれば、投入メダルフラグをクリアし ( S e 2 3 )、R A M 5 0 7 に投入不可フラグが設定されているか否かに基づいてメダルの投入が可能な状態か否かを判定し ( S e 2 4 )、メダルの投入が可能な状態でなければ S e 3 3 のステップに進む。

20

#### 【0284】

S e 2 4 のステップにおいてメダルの投入が可能な状態であれば、R A M 5 0 7 に設定された賭数の規定数を参照し、B E T カウンタの値が規定数であるか否を判定し ( S e 2 0 )、B E T カウンタの値が規定数でなければ、賭数が 1 加算された旨を示す B E T コマンドを通常コマンド送信用バッファに設定し ( S e 2 6 )、B E T カウンタの値を 1 加算し ( S e 2 7 )、B E T カウンタの値が規定数であるか否か、すなわちゲームの開始条件となる賭数が設定されたか否かを判定し ( S e 2 8 )、B E T カウンタの値が規定数でなければ S e 9 のステップに戻る。S e 2 6 のステップで設定された B E T コマンドは、その後のタイマ割込処理 ( メイン ) でサブ制御部 9 1 に対して送信される。

30

#### 【0285】

また、S e 2 8 のステップにおいて B E T カウンタの値が規定数であれば、すなわちゲームの開始条件となる賭数が設定された場合には、S e 1 2 のステップに戻り、ゲームの開始条件となる賭数が設定される前に乱数値がラッチされていれば、S a 1 3 のステップにて乱数値が読み出され、新たな乱数値の取込が可能となる。このため、ゲームの開始条件となる賭数が設定されておらず、スタートスイッチ 7 の操作が無効化されている状態でスタートスイッチ 7 が操作されたために乱数値がラッチされ、新たな乱数値の取込が禁止されている状態であっても、ゲームの開始条件となる既定数の賭数が設定され、スタートスイッチ 7 の操作が有効化された時点でラッチされている乱数値がダミーとして読み出され、その後のスタートスイッチ 7 が操作されたタイミングで新たに乱数値をラッチすることが可能となる。

40

#### 【0286】

S e 2 5 のステップにおいて B E T カウンタの値が規定数であれば、現在のクレジット

50

カウンタの値を示すクレジットコマンドを通常コマンド送信用バッファに設定し（S e 2 9）、R A M 5 0 7においてクレジットの値が格納されるクレジットカウンタの値を1加算し（S e 3 0）、クレジットカウンタの値が上限値である50であるか否かを判定し（S e 3 1）、クレジットカウンタの値が50でなければ、S e 9のステップに戻り、クレジットカウンタの値が50であれば投入不可フラグをR A M 5 0 7に設定し（S e 3 2）、S e 9のステップに戻る。S e 2 9のステップで設定されたクレジットコマンドは、その後のタイマ割込処理（メイン）でサブ制御部91に対して送信される。

#### 【0287】

S e 3 3のステップでは、スタートスイッチ7の操作が検出されているか否か、すなわちスタートスイッチ7の立上りを示す立上りエッジが設定されているか否かを判定する。  
さらに立上りエッジは、いずれかのスイッチがONに変化し、かつ全てのスイッチがOFFである場合にのみ設定されるので、S e 3 3のステップでは、いずれかのスイッチがONに変化したか否かに加えて他のスイッチがOFFであるかどうかについても判定されることとなる。

10

#### 【0288】

S e 3 3のステップにおいてスタートスイッチ7の操作が検出されていないと判定された場合、またはスタートスイッチ7が操作されたものの他のスイッチも操作されている場合には、S e 4 4のステップに進み、スタートスイッチ7の操作が有効に検出されれば、立上りエッジをクリアし（S e 3 4）、R A M 5 0 7に設定された賭数の規定数を参照し、B E Tカウンタの値が規定数であるか、すなわちゲームの開始条件となる賭数が設定されているか否かを判定する（S e 3 5）。

20

#### 【0289】

S e 3 5のステップにおいてB E Tカウンタの値が規定数でなければ、S e 9のステップに戻り、B E Tカウンタの値が規定数であれば、割込を禁止し（S e 3 6）、スタートスイッチ7の操作により乱数値レジスタ559Aに取り込まれた乱数値がR A M 5 0 7に割り当てられた乱数値ワークに格納されずに、電断割込処理（メイン）が実行されることで、電断前にラッチされた乱数値がバックアップされなくなってしまうことを防止した後、S e 3 7のステップに進む。

#### 【0290】

S e 3 7のステップでは、乱数ラッチフラグレジスタの値に基づいて乱数値がラッチされているか否か、すなわち乱数値レジスタ559Aに数値データが取り込まれているか否かを判定し、乱数値がラッチされていなければS e 3 6のステップにおいて禁止した割込を許可し（S e 3 8）、S e 9のステップに戻る。

30

#### 【0291】

S e 3 7のステップにおいて乱数値がラッチされていれば、乱数値レジスタ559Aから内部抽選用の乱数値を読み出し（S e 3 9）、読み出した乱数値をR A M 5 0 7の乱数値ワークに格納し（40）、S e 4 1のステップに進む。乱数値ワークは、内部抽選において用いる乱数値を格納するワークであり、S e 4 0のステップにおいて乱数値ワークに格納された乱数値が内部抽選に用いられることとなる。

#### 【0292】

40

S e 4 1のステップでは、S e 3 6のステップにおいて禁止した割込を許可し、投入不可フラグをR A M 5 0 7に設定するとともに、流路切替ソレノイド30をoffの状態とし、メダルの流路をメダル払出口9側の経路として新たなメダルの投入を禁止し（S e 4 2）、ゲーム開始時の設定を行う（S e 4 3）。そして、S e 4 3のステップの後、B E T処理を終了して図29のフローチャートに復帰する。

#### 【0293】

S e 4 4のステップにおいては、1枚B E Tスイッチ5の操作が検出されているか否か、すなわち1枚B E Tスイッチ5の立上りを示す立上りエッジが設定されているか否かを判定する。S e 4 4のステップにおいて1枚B E Tスイッチ5の操作が検出されていなければ、S e 5 2のステップに進み、1枚B E Tスイッチ5の操作が検出されていれば立上

50

リエッジをクリアし（Se 45）、RAM507に設定された賭数の規定数を参照し、BETカウンタの値が規定数であるか否かを判定する（Se 46）。Se 46のステップにおいてBETカウンタの値が規定数であればSe 9のステップに戻り、BETカウンタの値が規定数でなければ、クレジットカウンタの値が0であるか否かを判定し（Se 47）、クレジットカウンタの値が0であればSe 9のステップに戻る。Se 47のステップにおいてクレジットカウンタの値が0でなければ、賭数が1加算された旨を示すBETコマンドを通常コマンド送信用バッファに設定し（Se 48）、クレジットカウンタの値を1減算し（Se 49）、BETカウンタの値を1加算して（Se 50）、BETカウンタの値が規定数であるか否か、すなわちゲームの開始条件となる賭数が設定されたか否かを判定し（Se 51）、BETカウンタの値が規定数でなければSe 9のステップに戻る。Se 48のステップで設定されたBETコマンドは、その後のタイマ割込処理（メイン）でサブ制御部91に対して送信される。  
10

#### 【0294】

また、Se 51のステップにおいてBETカウンタの値が規定数であれば、すなわちゲームの開始条件となる賭数が設定された場合には、Se 12のステップに戻り、ゲームの開始条件となる賭数が設定される前に乱数値がラッチされていれば、Sa 13のステップにて乱数値が読み出され、新たな乱数値の取込が可能となる。このため、ゲームの開始条件となる賭数が設定されておらず、スタートスイッチ7の操作が無効化されている状態でスタートスイッチ7が操作されたために乱数値がラッチされ、新たな乱数値の取込が禁止されている状態であっても、ゲームの開始条件となる既定数の賭数が設定され、スタートスイッチ7の操作が有効化された時点でラッチされている乱数値がダミーとして読み出され、その後のスタートスイッチ7が操作されたタイミングで新たに乱数値をラッチすることが可能となる。  
20

#### 【0295】

Se 52のステップにおいては、MAXBETスイッチ6の操作が検出されているか否か、すなわちMAXBETスイッチ6の立上り示す立上りエッジが設定されているか否かを判定する。Se 52のステップにおいてMAXBETスイッチ6の操作が検出されていなければ、Se 61のステップに進み、MAXBETスイッチ6の操作が検出されれば、立上りエッジをクリアし（Se 53）、RAM507に設定された賭数の規定数を参照し、BETカウンタの値が規定数であるか否かを判定する（Se 54）。Se 54のステップにおいてBETカウンタの値が規定数であれば、Se 58のステップに進み、BETカウンタの値が規定数でなければ、クレジットカウンタの値が0であるか否かを判定し（Se 55）、クレジットカウンタの値が0であれば、Se 58のステップに進む。Se 55のステップにおいてクレジットカウンタの値が0でなければ、クレジットカウンタの値を1減算し（Se 56）、BETカウンタの値を1加算して（Se 57）、Se 54のステップに戻る。Se 58のステップでは、BETカウンタが加算されたか否かを判定し、BETカウンタが加算されなければ、Se 9のステップに戻り、BETカウンタが加算されれば、加算された数分賭数が加算された旨を示すBETコマンドを通常コマンド送信用バッファに設定し（Se 59）、Se 60のステップに進む。Se 59のステップで設定されたBETコマンドは、その後のタイマ割込処理（メイン）でサブ制御部91に対して送信される。  
30

#### 【0296】

Se 60のステップでは、BETカウンタの値が規定数であるか否か、すなわちゲームの開始条件となる賭数が設定されたか否かを判定し、BETカウンタの値が規定数でなければSe 9のステップに戻る。また、Se 60のステップにおいてBETカウンタの値が規定数であれば、すなわちゲームの開始条件となる賭数が設定された場合には、Se 12のステップに戻り、ゲームの開始条件となる賭数が設定される前に乱数値がラッチされれば、Sa 13のステップにて乱数値が読み出され、新たな乱数値の取込が可能となる。このため、ゲームの開始条件となる賭数が設定されておらず、スタートスイッチ7の操作が無効化されている状態でスタートスイッチ7が操作されたために乱数値がラッチされ  
40

、新たな乱数値の取込が禁止されている状態であっても、ゲームの開始条件となる既定数の賭数が設定され、スタートスイッチ7の操作が有効化された時点でラッチされている乱数値がダミーとして読み出され、その後のスタートスイッチ7が操作されたタイミングで新たに乱数値をラッチすることが可能となる。

#### 【0297】

S e 6 1 のステップにおいては、精算スイッチ10の操作が検出されているか否か、すなわち精算スイッチ10の立上りを示す立上りエッジが設定されているか否かを判定する。S e 6 1 のステップにおいて精算スイッチ10の操作が検出されていなければ、S e 9 のステップに戻り、精算スイッチ10の操作が検出されていれば、立上りエッジをクリアし(S e 6 2)、R A M 5 0 7 にリプレイゲームフラグが設定されているか否かに基づいて当該ゲームがリプレイゲームであるか否かを判定し(S e 6 3)、当該ゲームがリプレイゲームであればS e 9 のステップに戻る。S e 6 3 のステップにおいて当該ゲームがリプレイゲームでなければ、B E T カウンタの値が0か否かを判定し(S e 6 4)、B E T カウンタの値が0であればS e 6 6 のステップに進み、B E T カウンタの値が0でなければ、既に設定済み賭数の精算を行う旨を示す賭数精算フラグをR A M 5 0 7 に設定し(S e 6 5)、S e 6 6 のステップに進む。S e 6 6 のステップにおいては、ホッパーモータ34\_bを駆動してクレジットカウンタまたはB E T カウンタに格納された値分のメダルを払い出す制御、すなわちクレジットとして記憶されているメダルまたは賭数の設定に用いられたメダルを返却する制御が行われる精算処理を行う。そして、S e 6 6 のステップにおける精算処理の後、R A M 5 0 7 に設定されている投入不可フラグをクリアして(S e 6 7)、S e 9 のステップに戻る。10

#### 【0298】

図33及び図34は、メイン制御部41が一定間隔(0.56msの間隔)で起動処理やゲーム処理に割り込んで実行するタイマ割込処理(メイン)の制御内容を示すフローチャートである。尚、タイマ割込処理(メイン)の実行期間中は自動的に他の割込が禁止される。

#### 【0299】

タイマ割込処理(メイン)においては、まず、使用中のレジスタをスタック領域に退避した後(S k 1)、入力ポートから各種スイッチ類の検出データを入力するポート入力処理を行う(S k 2)。30

#### 【0300】

次いで、4種類のタイマ割込1~4から当該タイマ割込処理(メイン)において実行すべきタイマ割込を識別するための分岐用カウンタを1進める(S k 3)。S k 3 のステップでは、分岐用カウンタ値が0~2の場合に1が加算され、カウンタ値が3の場合に0に更新される。すなわち分岐用カウンタ値は、タイマ割込処理(メイン)が実行される毎に、0 1 2 3 0 ... の順番でループする。

#### 【0301】

次いで、分岐用カウンタ値を参照して2または3か、すなわちタイマ割込3またはタイマ割込4かを判定し(S k 4)、タイマ割込3またはタイマ割込4ではない場合、すなわちタイマ割込1またはタイマ割込2の場合には、リールモータ32L、32C、32Rの始動時または定速回転中か否かを確認し、リールモータ32L、32C、32Rの始動時または定速回転中であれば、後述するS k 8 のモータステップ処理において変更した位相信号データや後述するS k 21 の最終停止処理において変更した位相信号データを出力するモータ位相信号出力処理を実行する(S k 5)。40

#### 【0302】

次いで、分岐用カウンタ値を参照して1か否か、すなわちタイマ割込2か否かを判定し(S k 7)、タイマ割込2ではない場合、すなわちタイマ割込1の場合には、リールモータ32L、32C、32Rの始動時のステップ時間間隔の制御を行うリール始動処理(S k 8)、リールモータ32L、32C、32Rの位相信号データの変更を行うモータステップ処理(S k 8)、リールモータ32L、32C、32Rの停止後、一定時間経過後に50

位相信号を1相励磁に変更するモータ位相信号スタンバイ処理(Sk9)を順次実行した後、Sk21のステップに進む。

#### 【0303】

また、Sk6のステップにおいてタイマ割込2の場合には、各種表示器をダイナミック点灯させるLEDダイナミック表示処理(Sk10)、各種LED等の点灯信号等のデータを出力ポートへ出力する制御信号等出力処理(Sk11)、各種時間カウンタを更新する時間カウンタ更新処理(Sk12)、通常コマンド送信用バッファに格納されているコマンドまたは操作検出コマンドをサブ制御部91に対して送信するコマンド送信処理(Sk13)、外部出力信号を更新する外部出力信号更新処理(Sk14)を順次実行した後、Sk21のステップに進む。

10

#### 【0304】

また、Sk4のステップにおいてタイマ割込3またはタイマ割込4であれば、更に、分岐用カウンタ値を参照して3か否か、すなわちタイマ割込4か否かを判定し(Sk15)、タイマ割込4でなければ、すなわちタイマ割込3であれば、回転中のリール2L、2C、2Rの原点通過(リール基準位置の通過)をチェックし、リール回転エラーの発生を検知するとともに、停止準備が完了しているか(停止準備完了コードが設定されているか)を確認し、停止準備が完了しており、かつ定速回転中であれば、回転中のリールに対応するストップスイッチの操作を有効化する原点通過時処理(Sk16)、スイッチ類の検出状態に変化があったか否かを判定するスイッチ入力判定処理(Sk17)を順次実行した後、Sk21のステップに進む。

20

#### 【0305】

また、Sk15のステップにおいてタイマ割込4であれば、ストップスイッチ8L、8C、8Rの検出に伴って停止リールのワークに停止操作位置が格納されたときに、停止リールのワークに格納された停止操作位置から停止位置を決定し、何ステップ後に停止すれば良いかを算出する停止スイッチ処理(Sk19)、停止スイッチ処理で算出された停止までのステップ数をカウントして、停止する時期になったら2相励磁によるブレーキを開始する停止処理(Sk20)、停止処理においてブレーキを開始してから一定時間後に3相励磁とする最終停止処理(Sk21)を順次実行した後、Sk21のステップに進む。

#### 【0306】

Sk21のステップでは、停電判定処理を行う。停電判定処理では、電断検出回路48から電圧低下信号が入力されているか否かを判定し、電圧低下信号が入力されれば、前回の停電判定処理でも電圧低下信号が入力されていたか否かを判定し、前回の停電判定処理でも電圧低下信号が入力されていた場合には停電と判定し、その旨を示す電断フラグを設定する。

30

#### 【0307】

Sk22のステップでは、電断フラグが設定されているか否かを判定し、電断フラグが設定されていなければ、Sk1においてスタック領域に退避したレジスタを復帰し(Sk18)、割込前の処理に戻る。

#### 【0308】

一方、Sk22のステップにおいて電断フラグが設定されていた場合には、電断処理(メイン)に移行する。

40

#### 【0309】

このように本実施例では、一定間隔毎に基本処理に割り込んでタイマ割込処理(メイン)を実行するとともに、タイマ割込処理(メイン)を実行する毎に処理カウンタを更新し、処理カウンタ値に応じて定められた処理を行うようになっており、一度のタイマ割込処理(メイン)に要する負荷を分散できるうえに、処理カウンタ値に関わらず、電圧低下信号に基づいて電断の条件が成立しているか否かを判定する停電判定処理を行い、電断の条件が成立していれば、電断処理を行うようになっており、電断が検知された場合には速やかに電断処理を行うことが可能となる。

#### 【0310】

50

また、タイマ割込処理（メイン）内で、電断の条件が成立しているか否かの判定を行い、電断の条件が成立していれば、そのまま電断処理に移行することとなり、タイマ割込処理（メイン）の実行中に電断に伴う割込が発生することもないため、タイマ割込処理（メイン）の実行中に電断処理を割り込ませたり、タイマ割込処理（メイン）の終了を待って電断に伴う割込処理を行う必要がないため、電断条件の成立に伴う処理が複雑化してしまうことがない。

#### 【0311】

図35は、メイン制御部41が前述したタイマ割込処理（メイン）のタイマ割込3内において実行するスイッチ入力判定処理の制御内容を示すフローチャートである。

#### 【0312】

スイッチ入力判定処理では、ポート入力処理において取得した各スイッチの入力データを更新し（S k 1 0 1）、前回の入力データが示す検出状態と今回の入力データが示す検出状態と同じであるか否かを判定し（S k 1 0 2）、前回の入力データが示す検出状態と今回の入力データが示す検出状態と同じでなければ、図34のフローチャートに復帰する。

#### 【0313】

S k 1 0 2 のステップにおいて、前回の入力データが示す検出状態と今回の入力データが示す検出状態と同じ場合、すなわち  $2.24 \text{ m s}$  の間同じ検出状態を示している場合には、該当するスイッチの確定データを更新し（S k 1 0 3）、S k 1 0 4 のステップに進む。S k 1 0 3 のステップでは、今回の確定データを前回の確定データに移動し、前回と今回が同じと判定された入力データが示す検出状態を今回の確定データとして設定する。

#### 【0314】

S k 1 0 4 のステップでは、更新後の前回の確定データと今回の確定データと同じか否かを判定し、前回の確定データと今回の確定データと同じであれば、図34のフローチャートに復帰する。

#### 【0315】

S k 1 0 4 のステップにおいて前回の確定データと今回の確定データと同じでなければ、前回の確定データと今回の確定データとが異なるスイッチがOFFからONに変化したかを判定し（S k 1 0 5）、OFFからONに変化した場合には、さらに他のスイッチがONか否かを判定し（S k 1 0 6）、他のスイッチがONであればS k 1 0 8 のステップに進み、他のスイッチがONでなければ、該当するスイッチがOFFからONに変化した旨を示す立上りエッジを設定し（S k 1 0 7）、S k 1 0 8 のステップに進む。

#### 【0316】

S k 1 0 8 のステップでは、前回の確定データと今回の確定データとが異なるスイッチがONからOFFに変化したかを判定し、ONからOFFに変化した場合には、他のスイッチがONか否かを判定し（S k 1 0 9）、他のスイッチがONであれば図34のフローチャートに復帰し、他のスイッチがONでなければ、該当するスイッチがONからOFFに変化した旨を示す立下りエッジを設定し（S k 1 1 0）、図34のフローチャートに復帰する。

#### 【0317】

図36は、メイン制御部41が前述したタイマ割込処理（メイン）において電断フラグが設定されていると判定した場合に実行する電断処理（メイン）の制御内容を示すフローチャートである。

#### 【0318】

電断処理（メイン）においては、まず、使用している可能性がある全てのレジスタをスタック領域に退避する（S m 1）。尚、前述したIレジスタ及びIYレジスタの値は使用されているが、起動時の初期化に伴って常に同一の固定値が設定されるため、ここでは保存されない。

#### 【0319】

10

20

30

40

50

次いで、破壊診断用データ（本実施例では、5A(H)）をセットして（Sm2）、全ての出力ポートを初期化する（Sm3）。次いでRAM507の全ての格納領域（未使用領域及び未使用スタック領域を含む）の排他的論理和が0になるようにRAMパリティ調整用データを計算してセットし（Sm4）、RAM507へのアクセスを禁止する（Sm5）。

#### 【0320】

その後、電圧が低下してCPU505が停止するか、ユーザリセット信号が入力されて再起動するか、の待機状態に移行する。この待機状態では、乱数ラッチフラグレジスタの値に基づいて乱数値がラッチされているか否か、すなわち乱数値レジスタ559Aに数値データが取り込まれているか否かを判定し（Sm6）、乱数値がラッチされていれば、乱数値レジスタ559Aから内部抽選用の乱数値を読み出す（Sm7）ようになっており、これら以外の処理は行われない。Sm7のステップにおいて乱数値レジスタ559から内部抽選用の乱数値が読み出されると乱数ラッチフラグレジスタがクリアされ、新たな数値データの取込が許可されることとなる。また、Sm7のステップにおいては、内部抽選用の乱数値を読み出すものの、読み出した乱数値を用いる訳ではなく、スタートスイッチ7の操作に応じて新たな乱数値の取込を可能とするためにダミーとして読み出すものである。

#### 【0321】

そしてこの待機状態のまま電圧が低下すると内部的に動作停止状態になる。よって、電断時に確実にメイン制御部41は動作停止する。また、この待機状態において、電圧が低下せずにユーザリセット信号が入力されると前述した起動処理（メイン）がユーザモードから実行され、RAM507の格納データが正常であれば、元の処理に復帰することとなる。

#### 【0322】

このように本実施例のメイン制御部41は、システムリセットであるかユーザリセットであるかに関わらず、RAM507へのアクセスが許可された後、他の処理を行うことなく、まず、RAM507の全ての格納領域（未使用領域及び未使用スタック領域を含む）のRAMパリティを計算し、その後、計算したRAMパリティから格納データが正常か否かの判定を行う前に、RAM507の格納データが正常か否かに関わらず共通して行う必要のある処理を行い、その後、RAMパリティに基づいてRAM507の格納データが正常か否かの判定を行うようになっている。

#### 【0323】

このため、これらRAM507の格納データが正常か否かに関わらず共通して行う必要のある処理を、RAM507の格納データが正常と判定された場合と異常と判定された場合とで個別に実行する必要がないため、起動時の処理に用いるプログラム容量を削減することができる。

#### 【0324】

また、メイン制御部41の起動後、RAM507へのアクセスが許可された後、他の処理を行う前にRAMパリティの計算が行われるので、RAM507に対して何らのアクセスも行われる前の状態で算出されたRAMパリティに基づいて格納データが正常か否かの判断が行われるため、データが正常か否かの判定の信頼性を高めることができる。

#### 【0325】

また、電断処理において、RAM507の未使用領域及び未使用スタック領域を含む全てのデータに基づくRAMパリティが0となるようにRAMパリティ調整用データを計算し、格納するとともに、復旧時においてRAM507における未使用領域及び未使用スタック領域を含む全ての領域に格納されているデータに基づいて計算したRAMパリティが0か否かを判定することで、RAM507のデータが正常か否かを判定しているので、当該判定を正確にかつ簡便に行うことができるうえに、未使用領域や未使用スタック領域に不正なプログラムやデータが格納された場合でも、これら不正なプログラムやデータが格納されたまま復帰してしまうことを防止できる。

10

20

30

40

50

**【 0 3 2 6 】**

また、RAMパリティを計算した後、計算したRAMパリティから格納データが正常か否かの判定を行う前に、RAM507の格納データが正常か否かに関わらず共通して行う処理として、RAM507に格納されている各スイッチ入力データ、確定データ、エッジデータ、すなわち各スイッチの検出状況を示すデータをクリアするようになっており、メイン制御部41の起動時においてRAM507のデータが正常であり、以前の制御状態に復帰させる場合でも以前の制御状態には復帰させない場合でも、割込が許可された後、速やかにスイッチの検出状態のチェックを開始することが可能となる。

**【 0 3 2 7 】**

本実施例のスロットマシン1におけるメイン制御部41は、システムリセット時、すなわち電源投入などに伴いシステムリセット信号が入力されたときに、CPU505がROM506などに記憶されているセキュリティチェックプログラム506Aを読み出して実行することにより、セキュリティモードとなる。

10

**【 0 3 2 8 】**

このときには、セキュリティチェックプログラム506Aに対応した処理としてセキュリティチェック処理が実行される。ここで、メイン制御部41がセキュリティモードとなるセキュリティ時間は、ROM506のプログラム管理エリアに記憶されているセキュリティ時間設定KSESのビット番号[2-0]やビット番号[4-3]に予め格納されたビット値に応じて、一定の固定時間とは異なる時間成分を含むことができる。

**【 0 3 2 9 】**

20

例えば、セキュリティ時間設定KSESのビット番号[2-0]におけるビット値が“000”以外の値であれば、図10(D)に示す設定内容に対応して、固定時間に加えて予め選択可能な複数の延長時間のいずれかを、セキュリティ時間に含まれる時間成分として設定することができる。また、セキュリティ時間設定KSESのビット番号[4-3]におけるビット値が“00”以外の値であれば、図10(C)に示すショートモード又はロングモードに対応して、システムリセットや電源投入がなされるごとに所定の時間範囲で変化する可変設定時間を、セキュリティ時間に含まれる時間成分として設定することができる。

**【 0 3 3 0 】**

30

こうして設定されたセキュリティ時間が経過するまでは、ROM506に記憶されているユーザプログラムの実行が開始されない。そして、乱数回路509による乱数値となる数値データの生成動作も、メイン制御部41がセキュリティモード中である期間では、開始されないようにすれば良い。これにより、電源投入等のシステムリセットによる動作開始タイミングから、乱数回路509の動作開始タイミングや更新される数値データなどを特定することが困難になり、プログラムの解析結果に基づく狙い撃ちや、いわゆる「ぶら下げ基板」を接続して所定タイミングで不正信号を入力することで、不正に特別役を当選させるなどの行為を、確実に防止することができる。

**【 0 3 3 1 】**

一例として、スロットマシン1の機種毎に、セキュリティ時間設定KSESのビット番号[2-0]におけるビット値を異なる値に設定する。この場合には、セキュリティモードの延長時間を、スロットマシン1の機種毎に異ならせることができ、スロットマシン1の動作開始タイミングから乱数回路509の動作開始タイミングを特定することが困難になる。また、セキュリティ時間設定KSESのビット番号[4-3]におけるビット値を“01”又は“10”に設定することにより、可変設定時間を、システムリセット毎に異ならせることができる。これにより、スロットマシン1の動作開始タイミングから乱数回路509の動作開始タイミングを特定することは著しく困難になる。

40

**【 0 3 3 2 】**

図37は、メイン制御部41に内蔵された乱数回路509の動作を説明するためのタイミングチャートである。また、図37(A)では、遊技制御基板40に搭載された制御用クロック生成回路42により生成される制御用クロックCCLKを示している。図37(

50

B)では、乱数用クロック生成回路43により生成される乱数用クロックRCLKを示している。図37(A)及び(B)に示すように、制御用クロックCCLKの発振周波数と、乱数用クロックRCLKの発振周波数とは、互いに異なる周波数となっており、また、いずれか一方の発振周波数が他方の発振周波数の整数倍になることがない。

#### 【0333】

図37(B)に示すように、乱数用クロックRCLKは、タイミングT10, T11, T12, ...においてローレベルからハイレベルに立ち上がる。そして、乱数用クロックRCLKは、メイン制御部41の乱数用外部クロック端子ERCに供給され、図12に示す乱数回路509が備えるクロック用フリップフロップ552におけるクロック端子CKに入力される。クロック用フリップフロップ552は、逆相出力端子(反転出力端子)QバーからD入力端子へとフィードバックされるラッチ用クロックRC0を、クロック端子CKに入力される乱数用クロックRCLKの立ち上がりエッジに応答して取り込み(ラッチして)、正相出力端子(非反転出力端子)Qから乱数更新クロックRGKとして出力する。これにより、乱数更新クロックRGKは、図37(C)に示すように、タイミングT10, T12, T14, ...において、ローレベルからハイレベルへと立ち上がり、乱数用クロックRCLKの発振周波数の1/2の発振周波数を有する信号となる。例えば、乱数用クロックRCLKの発振周波数が20MHzであれば、乱数更新クロックRGKの発振周波数は10MHzとなる。そして、乱数用クロックRCLKの発振周波数は制御用クロックCCLKの発振周波数の整数倍にも整数分の1にもならないことから、乱数更新クロックRGKの発振周波数は、制御用クロックCCLKの発振周波数とは異なる周波数となる。  
乱数生成回路553は、例えば乱数更新クロックRGKの立ち上がりエッジに応答して、カウント値順列RCNにおける数値データを更新する。乱数列変更回路555は、乱数列変更設定回路556による乱数更新規則の設定に基づき、乱数生成回路553から出力されたカウント値順列RCNにおける数値データの更新順を変更したものを、乱数列RSNとして出力する。こうして、乱数列RSNにおける数値データは、例えば図37(D)に示すように、乱数更新クロックRGKの立ち上がりエッジなどに応答して更新される。

#### 【0334】

クロック用フリップフロップ552から出力されるラッチ用クロックRC0は、乱数更新クロックRGKの反転信号となり、その発振周波数は乱数更新クロックRGKの発振周波数と同一で、その位相は乱数更新クロックRGKの位相と(=180°)だけ異なる。ラッチ用クロックRC0は、分岐点BR1にてラッチ用クロックRC1とラッチ用クロックRC2とに分岐される。したがって、例えば図37(E)に示すように、各ラッチ用クロックRC0、RC1、RC2はいずれも、共通の周期で信号状態が変化する発振信号となる。ラッチ用クロックRC1は、ラッチ用フリップフロップ557Aのクロック端子CKに入力される。ラッチ用クロックRC2は、ラッチ用フリップフロップ557Bのクロック端子CKに入力される。

#### 【0335】

ラッチ用フリップフロップ557Aは、ラッチ用クロックRC1の立ち上がりエッジに応答して、スタートスイッチ7から伝送されて入力ポートP0に供給されたゲーム開始信号SS1を取り込み(ラッチして)、ゲーム開始時ラッチ信号SL1として出力端子Qから出力する。そして、乱数ラッチセレクタ558Aにおける取込方法が入力ポートP0への信号入力に指定されていれば、ゲーム開始時ラッチ信号SL1が乱数ラッチ信号LL1として出力される。これにより、例えば図37(F)に示すようなタイミングでオフ状態(ローレベル)とオン状態(ハイレベル)とで信号状態が変化するゲーム開始信号SS1は、ラッチ用クロックRC1が立ち上がるタイミングT11、T13、T15、...にてラッチ用フリップフロップ557Aに取り込まれた後、図37(G)に示すようなタイミングT11、T13で信号状態がオフ状態とオン状態とで変化する乱数ラッチ信号LL1となって、乱数ラッチセレクタ558Aから出力される。

#### 【0336】

乱数値レジスタR1Dとなる乱数値レジスタ559Aは、乱数列変更回路555から出

10

20

30

40

50

力される乱数列 R S N における数値データを、乱数ラッチセレクタ 5 5 8 A からクロック端子へと入力される乱数ラッチ信号 L L 1 の立ち上がりエッジに応答して取り込み（ラッチして）、記憶データとなる数値データを更新する。

#### 【 0 3 3 7 】

例えは図 3 7 ( G ) に示すように、タイミング T 1 1 にて乱数ラッチ信号 L L 1 がオフ状態からオン状態に変化する立ち上がりエッジが生じた場合には、このタイミング T 1 1 にて乱数列変更回路 5 5 5 から出力されている乱数列 R S N における数値データが、図 3 7 ( H ) に示すように、乱数値レジスタ R 1 D に取り込まれ、乱数値となる数値データとして取得される。これにより、乱数値レジスタ R 1 D となる乱数値レジスタ 5 5 9 A では、スタートスイッチ 7 の操作が検出されたことに基づき、乱数値として用いられる数値データを取得して記憶することができる。10

#### 【 0 3 3 8 】

また、図 9 ( B ) に示すような第 2 亂数初期設定 K R S 2 のビット番号 [ 0 ] におけるビット値を “ 1 ” とすれば、例えは乱数回路 5 0 9 にて生成される乱数値となる数値データのスタート値を、システムリセット毎に変更することができる。これにより、たとえ乱数回路 5 0 9 の動作開始タイミングを特定することができたとしても、乱数回路 5 0 9 が備える乱数値レジスタ 5 5 9 A から読み出される数値データを特定することは困難になり、プログラムの解析結果に基づく狙い撃ちや、いわゆる「ぶら下げ基板」を接続して所定タイミングで不正信号を入力することで、不正に特別役を当選させるなどの行為を、確実に防止することができる。20

#### 【 0 3 3 9 】

このように本実施例では、セキュリティ時間設定 K S E S のビット番号 [ 4 - 3 ] におけるビット値を “ 0 0 ” 以外の値とすることにより、電源投入等に伴うシステムリセットごとに所定の時間範囲で変化する可変設定時間を、セキュリティ時間に含まれる時間成分として設定することができる。これにより、電源投入等に伴うシステムリセットによる動作開始タイミングから、乱数回路 5 0 9 の動作開始タイミングや更新される数値データなどを特定することが困難になり、プログラムの解析結果に基づく狙い撃ちや、いわゆる「ぶら下げ基板」を接続して所定タイミングで不正信号を入力することで、不正に特別役を当選させるなどの行為を、確実に防止することができる。

#### 【 0 3 4 0 】

また、セキュリティ時間設定 K S E S のビット番号 [ 2 - 0 ] におけるビット値を “ 0 0 0 ” 以外の値とすることにより、固定時間に加えて予め選択可能な複数の延長時間のいずれかを、セキュリティ時間に含まれる時間成分として設定することができる。これにより、電源投入等に伴うシステムリセットによる動作開始タイミングから、乱数回路 5 0 9 の動作開始タイミングや更新される数値データなどを特定することが困難になり、プログラムの解析結果に基づく狙い撃ちや、いわゆる「ぶら下げ基板」を接続して所定タイミングで不正信号を入力することで、不正に特別役を当選させるなどの行為を、確実に防止することができる。30

#### 【 0 3 4 1 】

また、乱数用クロック生成回路 4 3 により生成される乱数用クロック R C L K の発振周波数と、制御用クロック生成回路 4 2 により生成される制御用クロック C C L K の発振周波数とは、互いに異なっており、また、一方の発振周波数が他方の発振周波数の整数倍となることもない。そのため、乱数回路 5 0 9 のクロック用フリップフロップ 5 5 2 により生成される乱数更新クロック R G K やラッチ用クロック R C 0 の発振周波数は、乱数用クロック R C L K の発振周波数の 1 / 2 となるが、制御用クロック C C L K の発振周波数や、制御用クロック C C L K の発振周波数の 1 / 2 となる内部システムクロック S C L K の発振周波数とは、異なるものとなる。こうして、制御用クロック C C L K や内部システムクロック S C L K と、乱数更新クロック R G K とに同期が生じることを防ぎ、C P U 5 0 5 の動作タイミングからは、乱数回路 5 0 9 にて乱数生成回路 5 5 3 や乱数列変更回路 5 5 5 により生成される乱数列 R S N における数値データの更新タイミングを特定すること4050

が困難になる。これにより、CPU505の動作タイミングから乱数回路509における乱数値となる数値データの更新動作を解析した結果に基づく狙い撃ちなどを、確実に防止することができる。また、ラッチ用クロックRC0を分岐することにより生成されるラッチ用クロックRC1、RC2の発振周波数も、制御用クロックCCLKや内部システムクロックSCLKの発振周波数とは、異なるものとなる。こうして、制御用クロックCCLKや内部システムクロックSCLKと、ラッチ用クロックRC1、RC2とに同期が生じることを防ぎ、CPU505の動作タイミングからは、乱数回路509にて乱数値となる数値データが取り込まれる動作タイミングを特定することが困難になる。これにより、CPU505の動作タイミングから乱数回路509における乱数値となる数値データの取込動作を解析した結果に基づく狙い撃ちなどを、確実に防止することができる。

10

#### 【0342】

メイン制御部41に内蔵又は外付けされた乱数回路509では、第2乱数初期設定KRS2のビット番号[0]におけるビット値が“1”とされたことに対応して、乱数生成回路553から出力されるカウント値順列RCNや乱数列変更回路555から出力される乱数列RSNにおける数値データのスタート値を、システムリセット毎に変更することができる。これにより、たとえ乱数回路509の動作開始タイミングを特定することができたとしても、乱数回路509が備える乱数値レジスタ559Aから読み出される数値データを特定することは困難になり、プログラムの解析結果に基づく狙い撃ちや、いわゆる「ぶら下げ基板」の接続による不正信号の入力などを、確実に防止することができる。

20

#### 【0343】

メイン制御部41が備える外部バスインターフェース501では、内部リソースアクセス制御回路501Aにより、例えばROM506の記憶データといった、メイン制御部41の内部データにつき、CPU505等の内部回路以外による外部読出が制限される。これにより、例えばROM506に記憶されているユーザプログラムといった、ゲームを制御するプログラムがメイン制御部41の外部から読み出されて解析などに提供されることを防止できる。そして、遊技制御処理プログラムの解析結果に基づく狙い撃ちや、いわゆる「ぶら下げ基板」の接続による不正信号の入力などを、確実に防止することができる。

#### 【0344】

メイン制御部41に内蔵又は外付けされた乱数回路509では、周波数監視回路551により乱数用クロックRCLKにおける周波数異常が検知されたときに、内部情報レジスタCIFのビット番号[4]に格納される内部情報データCIF4のビット値が“1”に設定される。そして、CPU505では、内部情報データCIF4の読出値が“1”であると連続して判定された回数が、ステップS65にてクロック異常判定値に達したと判定されたときに、乱数回路509の動作状態に異常が発生したと判定する。そして乱数回路509の動作状態に異常が発生したと判定された場合には、エラー状態に制御され、ゲームの進行が不能化されることとなる。これにより、乱数用クロックRCLKとして不正信号を入力することによる不正行為を確実に防止することができる。

30

#### 【0345】

メイン制御部41に内蔵されたCPU505は、乱数回路509に対するラッチ信号の出力に相当する乱数値取込指定レジスタRDLTへのビット値“1”的書き込みを行い、乱数値レジスタ559Aを複数回読み出す。そして、読み出した数値データの全ビットを監視して、変化しないビットデータの有無に基づき、乱数回路509の動作状態に異常が発生したと判定する。そして乱数回路509の動作状態に異常が発生したと判定された場合には、エラー状態に制御され、ゲームの進行が不能化されることとなる。これにより、乱数回路509の動作状態に異常が発生していることを確実かつ容易に検知して、不正行為を防止することができる。特に、乱数回路509により数値データが正常に更新されないまま、すなわち数値データが固定されたままの状態で内部抽選を行わせることで、常に特別役を当選させるなどの不正を防止できる。

40

#### 【0346】

メイン制御部41に内蔵又は外付けされた乱数回路509は、第2乱数初期設定KRS

50

2 のビット番号 [ 0 ] におけるビット値が “ 1 ” である場合に、システムリセット毎に乱数値となる数値データのスタート値を変更する。このときには、例えばメイン制御部 4 1 に内蔵されたフリーランカウンタのカウント値などを用いて、システムリセット毎に変更されるスタート値を決定すれば良い。これにより、システムリセット等のタイミングにより異なる初期値決定用データを用いて初期値を決定することができ、狙い撃ちなどによる不正行為を防止することができる。

#### 【 0 3 4 7 】

尚、本実施例では、セキュリティモードの延長時間が、セキュリティ時間設定 K S E S のビット番号 [ 2 - 0 ] におけるビット値に対応して、予め選択可能な複数の延長時間のいずれかとなり、この延長時間はシステムリセット毎に変更されないものであったが、例えば R O M 5 0 6 に記憶されたユーザプログラムにおける設定などにより、固定時間に加算される延長時間を、システムリセット毎に複数の延長時間のいずれかに決定するよりも良い。この場合には、延長時間がいずれも、最長の可変設定時間に比べて、長くなるように定義しておく。そして、大まかな延長時間を決定した後、詳細な延長時間を決定すれば良い。これにより、電源投入等に伴うシステムリセット時にセキュリティモードとなるセキュリティ時間を、システムリセット毎に大きく変化させることが可能になり、スロットマシン 1 の動作開始タイミングから乱数回路 5 0 9 の動作開始タイミングや更新される数値データなどを特定することができる、より困難になる。

#### 【 0 3 4 8 】

また、セキュリティモードの延長時間などは、メイン制御部 4 1 を構成するチップ毎に付与される I D ナンバーを用いて決定されるようにしても良い。一例として、I D ナンバーに所定のスクランブル処理を施す演算や、I D ナンバーを用いた加算・減算・乗算・除算などの演算の一部又は全部を実行して、算出された値に対応して延長時間を設定しても良い。この場合には、例えばシステムリセット毎に延長時間を決定するために用いる演算式を変更することなどにより、システムリセット毎に延長時間がランダムに決定されるようにしても良い。さらに、例えば I D ナンバーを用いて延長時間を決定するための演算式をシステムリセット時に格納したフリーランカウンタのカウント値に対応して決定するといったように、フリーランカウンタのカウント値と、I D ナンバーとを組み合わせて使用することなどにより、システムリセット毎に延長時間がランダムに決定されるようにしても良い。また、乱数回路 5 0 9 にて生成される乱数のスタート値をシステムリセット毎に変更する場合にも、フリーランカウンタのカウント値と、I D ナンバーとを組み合わせて使用することなどにより、乱数のスタート値を決定しても良い。

#### 【 0 3 4 9 】

また、システムリセット時には、セキュリティモードから開始し、乱数回路 5 0 9 の設定を行う乱数回路設定処理を行うとともに、乱数回路設定処理において乱数の更新規則の設定や乱数のスタート値の設定などが行われる一方でユーザリセット時には、セキュリティモードは省略され、ユーザモードから開始することとなる。すなわち、システムリセット時には、乱数の初期値が設定し直されるのに対して、ユーザリセット時には、乱数回路 5 0 9 の設定は行われず、リセット前からの状態のまま乱数となる数値データの更新が継続されることとなる。これにより、意図的にユーザリセットを行った場合でもユーザリセットのタイミングから乱数回路 5 0 9 における乱数値となる数値データを特定することは困難となり、狙い撃ちなどによる不正行為をさらに効果的に防止することができる。

#### 【 0 3 5 0 】

また、セキュリティモード内で乱数のスタート値の設定などを含む乱数回路の設定が行われ、数値データの更新が開始され、その後ユーザモードへ移行した後に、タイマ割込処理の実行が許可されるようになっている。すなわちタイマ割込処理の実行が許可される前に乱数のスタート値の設定が行われ、この設定された乱数のスタート値から数値データの更新が開始されるため、タイマ割込処理中に実行されるスイッチ類の判定状況からタイマ割込処理の許可されたタイミングまではある程度特定することはできるものの、乱数回路 5 0 9 の数値データの更新が開始されたタイミングを特定することはできないため、乱数

10

20

30

40

50

回路 509 における乱数値となる数値データを特定することは困難となり、狙い撃ちなどによる不正行為をさらに効果的に防止することができる。

#### 【0351】

本実施例では、メイン制御部 41 の外部に設けられた乱数用クロック生成回路 43 により、制御用クロック生成回路 42 で生成される制御用クロック CCLK の発振周波数とは異なる発振周波数を有する乱数用クロック RCLK を生成して、乱数回路 509 に供給するものであったが、メイン制御部 41 の CPU505 に供給されるクロック信号と、乱数回路 509 に供給されるクロック信号とが、共通のクロック生成回路に含まれる 1 つの発振器により生成された発振信号を用いて、生成されるようにしても良い。この場合には、例えば乱数用クロック RCLK と制御用クロック CCLK をそれぞれ生成するための分周器などを設け、ラッチ用クロック RC0、RC1、RC2 と制御用クロック CCLK あるいは内部システムクロック SCLK との同期が生じにくくなるように、各分周器における分周比などを設定すれば良い。制御用クロック生成回路 42 と乱数用クロック生成回路 43 とは、その全部又は一部が、メイン制御部 41 の内部に設けられても良いし、メイン制御部 41 の外部に設けられても良い。10

#### 【0352】

本実施例では、乱数回路 509 が乱数用クロック生成回路 43 により生成された乱数用クロック RCLK の供給を受け、クロック用フリップフロップ 552 により、乱数更新クロック RGK とラッチ用クロック RC0 とを生成するものであったが、例えば乱数用クロック生成回路 43 といった、乱数回路 509 の外部において、乱数更新クロック RGK やラッチ用クロック RC0 となる発振信号が生成されるようにしても良い。あるいは、乱数回路 509 の内部にて、乱数更新クロック RGK を生成するための回路と、ラッチ用クロック RC0 を生成するための回路とを、別個に設けるようにしても良い。一例として、クロック用フリップフロップ 552 と同様のフリップフロップにより乱数更新クロック RGK を生成する一方で、乱数更新クロック RGK の信号状態を反転させる反転回路を設け、その反転回路から出力される信号を、ラッチ用クロック RC0 として用いるようにしても良い。20

#### 【0353】

本実施例に適用した乱数回路 509 では、乱数ラッチフラグデータ RDFM0 が“1”的状態、すなわち乱数値レジスタ R1D に数値データが取り込まれている状態では、新たな乱数値の取込要求が発生した場合でも、新たな数値データを乱数値レジスタ R1D に取り込まないようになっており、このような状態では、乱数値レジスタ R1D の数値データが読み出されて、乱数ラッチフラグデータ RDFM0 がクリアされるまで新たな数値データを乱数値レジスタ R1D に取り込むことが不可能となる。30

#### 【0354】

このため、図 38 に示すように、スタートスイッチ 7 の操作によりゲーム開始信号 SS1 が入力されて数値データがラッチされ、乱数値レジスタ R1D に格納された後、この格納された数値データが読み出されるまでは、乱数ラッチフラグデータ RDFM0 が“1”的状態となることで、格納されている数値データが保持され、その間にゲーム開始信号 SS1 が入力されても新たな数値データに上書きされてしまうことがないので、静電気などによりゲーム開始信号 SS1 の信号線にノイズがのっても数値データが変わってしまうことがなく、このような場合であってもスタートスイッチ 7 が操作されたタイミングでラッチされた数値データとは異なる数値データを用いて内部抽選が行われてしまうことがない。40

#### 【0355】

尚、乱数値レジスタ R1D に数値データが取り込まれている状態において新たな数値データを乱数値レジスタ R1D に取り込まないようにする構成としては、新たな数値データのラッチを禁止する構成であっても良いし、新たな数値データをラッチするものの、乱数値レジスタ R1D への書き込みを禁止する構成であっても良い。

#### 【0356】

10

20

30

40

50

上記のように、本実施例では、乱数値レジスタ R 1 D に数値データが取り込まれている状態では、この格納された数値データが読み出されるまで、格納されている数値データを保持する構成を採用しているが、この場合には、スタートスイッチ 7 の操作がゲームの進行上有効でない期間においてなされた場合でも、ラッチされて乱数値レジスタ R 1 D に数値データが保持されたままとなり、その後、スタートスイッチ 7 の操作が有効となってスタートスイッチ 7 が操作された場合、本来、ゲームを開始させるためにスタートスイッチ 7 が操作されたタイミングとは異なるタイミングでラッチされた数値データによって内部抽選が行われてしまうという新たな問題が生じることとなる。

#### 【 0 3 5 7 】

これに対して本実施例では、図 3 9 に示すように、規定数の賭数が設定されていない状態において規定数の賭数が設定され、ゲームの開始条件を満たした時点で、乱数値レジスタ R 1 D に数値データがラッチされているか否か、すなわち乱数ラッチフラグが設定されているか否かを確認し、乱数値レジスタ R 1 D に数値データがラッチされている場合には、ダミーで乱数値レジスタ R 1 D の数値データを読み出すようになっている。これにより、乱数ラッチフラグがクリアされ、新たな数値データの取込が可能な状態となる。

#### 【 0 3 5 8 】

このため、規定数の賭数が設定される前にスタートスイッチ 7 が操作されて乱数値レジスタ R 1 D に数値データが格納されており、その状態が保持されても、規定数の賭数が設定され、ゲームの開始条件を満たすと同時に、乱数値レジスタ R 1 D に新たにラッチされた数値データを格納可能な状態となるので、ゲームの開始条件となる賭数が設定される前にラッチされ乱数値レジスタ R 1 D に保持されていた数値データ、すなわちゲームを開発可能な状態でスタートスイッチ 7 が操作されたタイミングとは異なるタイミングでラッチされた数値データを用いて内部抽選が行われてしまうことがない。

#### 【 0 3 5 9 】

尚、本実施例では、規定数の賭数が設定され、ゲームの開始条件を満たした時点で乱数ラッチフラグが設定されているか否か、すなわち規定数の賭数が設定される前にラッチされた数値データが乱数値レジスタ R 1 D に格納されているか否かを確認し、乱数ラッチフラグが設定されている場合にのみ、乱数値レジスタ R 1 D に格納されている数値データを読み出すようになっているが、規定数の賭数が設定され、ゲームの開始条件を満たした時点で一律に乱数値レジスタ R 1 D に格納されている数値データを読み出すようにしても良い。

#### 【 0 3 6 0 】

また、本実施例では、図 4 0 に示すように、ゲームの開始条件を満たす規定数の賭数が設定されている状態においてスタートスイッチ 7 が操作されても、スタートスイッチ 7 以外のスイッチが操作されている状態では、ゲームの開始が禁止されるようになっているが、規定数の賭数が設定されている状態であるもののスタートスイッチ 7 以外のスイッチが操作されており、ゲームの開始が禁止されている状態では、全てのスイッチが OFF となり、ゲームの開始が許可された時点で、乱数値レジスタ R 1 D に数値データがラッチされているか否か、すなわち乱数ラッチフラグが設定されているか否かを確認し、乱数値レジスタ R 1 D に数値データがラッチされている場合には、ダミーで乱数値レジスタ R 1 D の数値データを読み出すようになっている。これにより、乱数ラッチフラグがクリアされ、新たな数値データの取込が可能な状態となる。

#### 【 0 3 6 1 】

このため、ゲームの開始条件を満たす規定数の賭数が設定されている状態であるもののスタートスイッチ 7 以外のスイッチが操作されており、ゲームの開始が禁止されている状態においてスタートスイッチ 7 が操作されて乱数値レジスタ R 1 D に数値データが格納され、その状態が保持されても、全てのスイッチが OFF となりゲームの開始が許可されると同時に、乱数値レジスタ R 1 D に新たにラッチされた数値データを格納可能な状態となるので、ゲームの開始が禁止されている状態においてラッチされ乱数値レジスタ R 1 D に保持されていた数値データ、すなわちゲームを開発可能な状態でスタートスイッチ 7

10

20

30

40

50

が操作されたタイミングとは異なるタイミングでラッチされた数値データを用いて内部抽選が行われてしまうことがない。

#### 【0362】

尚、本実施例では、ゲームの開始条件を満たす規定数の賭数が設定されている状態であるもののスタートスイッチ7以外のスイッチが操作されており、ゲームの開始が禁止されている状態においては、全てのスイッチがOFFとなりゲームの開始が許可された時点で乱数ラッチフラグが設定されているか否か、すなわちゲームの開始が許可される前にラッチされた数値データが乱数値レジスタR1Dに格納されているか否かを確認し、乱数ラッチフラグが設定されている場合にのみ、乱数値レジスタR1Dに格納されている数値データを読み出すようになっているが、ゲームの開始が許可された時点で一律に乱数値レジスタR1Dに格納されている数値データを読み出すようにしても良い。10

#### 【0363】

また、本実施例では、電源投入時においてメイン制御部41の制御状態を電断前の制御状態に復帰させることや瞬停時に一時的にメイン制御部41の動作が停止してもユーザリセット信号の入力により、動作停止前の制御状態に復帰させることが可能とされており、ゲームの開始条件を満たす規定数の賭数が設定されている状態に復帰することもある。

#### 【0364】

一方で、電断時や瞬停時は電源電圧が不安定な状態であり、このような状況においてはノイズなどによってゲーム開始信号SS1が入力されて数値データがラッチされ、乱数値レジスタR1Dに格納され、そのまま保持されてしまう虞がある。20

#### 【0365】

これに対して、本実施例では、電断処理（メイン）の終了後、乱数値レジスタR1Dに数値データがラッチされているか否か、すなわち乱数ラッチフラグが設定されているか否かを確認し、乱数値レジスタR1Dに数値データがラッチされている場合には、ダミーで乱数値レジスタR1Dの数値データを読み出す処理を、電圧低下によりメイン制御部41の動作が停止するか、ユーザリセット信号の入力により再起動するまでの間、繰り返し行うようになっている。これにより、乱数ラッチフラグがクリアされ、新たな数値データの取込が可能な状態となる。

#### 【0366】

このため、瞬停などにより電断処理（メイン）が行われた後、ユーザリセット信号の入力により再起動する場合において、その間に、ノイズなどによって乱数値レジスタR1Dに数値データがラッチされてしまった場合でも、すぐに読み出されることにより乱数値レジスタR1Dに新たにラッチされた数値データを格納可能な状態となるので、瞬停時にノイズなどによってラッチされた数値データ、すなわちゲームを開始可能な状態でスタートスイッチ7が操作されたタイミングとは異なるタイミングでラッチされた数値データを用いて内部抽選が行われてしまうことがない。30

#### 【0367】

尚、本実施例では、電断処理（メイン）の終了後、乱数値レジスタR1Dに数値データがラッチされているか否か、すなわち数値データが乱数値レジスタR1Dに格納されているか否かを確認し、乱数ラッチフラグが設定されている場合にのみ、乱数値レジスタR1Dに格納されている数値データを読み出すようになっているが、電断処理（メイン）の終了後、数値データが乱数値レジスタR1Dに格納されているか否かに関わらず一律に乱数値レジスタR1Dに格納されている数値データを読み出す処理を繰り返し行うようにしても良い。40

#### 【0368】

さらに、図41に示すように、電源投入またはユーザリセットに伴いメイン制御部41が起動した後、電断前またはユーザリセット前の制御状態に復帰してゲームの開始が許可される前の段階でスタートスイッチ7が操作されて乱数値レジスタR1Dに数値データが格納され、その状態が保持されていても、制御状態が復帰してゲームの開始が許可されるのと同時に、乱数値レジスタR1Dに新たにラッチされた数値データを格納可能な状態と50

なるので、起動後、電断前またはユーザリセット前の制御状態に復帰する前の段階でラッチされ乱数値レジスタ R 1 D に保持されていた数値データ、すなわちゲームを開始可能な状態でスタートスイッチ 7 が操作されたタイミングとは異なるタイミングでラッチされた数値データを用いて内部抽選が行われてしまうことがない。

#### 【 0 3 6 9 】

尚、本実施例では、起動後、電断前またはユーザリセット前の制御状態に復帰した時点で乱数ラッチフラグが設定されているか否か、すなわちゲームの開始が許可される前にラッチされた数値データが乱数値レジスタ R 1 D に格納されているか否かを確認し、乱数ラッチフラグが設定されている場合にのみ、乱数値レジスタ R 1 D に格納されている数値データを読み出すようになっているが、起動後、電断前またはユーザリセット前の制御状態に復帰した時点で一律に乱数値レジスタ R 1 D に格納されている数値データを読み出すようにも良い。10

#### 【 0 3 7 0 】

また、電源投入後、起動後、電断前またはユーザリセット前の制御状態に復帰する場合には、復帰する制御状態がゲームを開始可能な状態か否かに関わらず一律に乱数値レジスタ R 1 D に格納されている数値データを読み出すようになっているが、復帰する制御状態がゲームを開始可能な状態であるか否かを確認し、復帰する制御状態がゲームを開始可能な状態である場合にのみ乱数値レジスタ R 1 D に格納されている数値データを読み出すようにも良い。

#### 【 0 3 7 1 】

メイン制御部 4 1 は、ゲームを開始可能な状態でスタートスイッチ 7 が操作されたか否かをスタートスイッチ 7 の立上りを示す立上りエッジが設定されているか否かに基づいて判定する。20

#### 【 0 3 7 2 】

スタートスイッチ 7 の立上りエッジは、一定間隔毎に割り込んで実行するタイマ割込処理（メイン）4 回に 1 回毎（約 2 . 24 ms 毎）に実行するスイッチ入力判定処理において、スタートスイッチ 7 の検出状態に基づく確定データが OFF から ON に変化したことを条件に設定される。確定データは、スイッチ入力判定処理毎に前回の検出状態と今回の状態が一致する場合にのみ更新されるデータであることから、スタートスイッチ 7 の立上りエッジは、スタートスイッチ 7 の検出状態が OFF の状態である場合に、スイッチ入力判定処理において 2 回連続してスタートスイッチ 7 の ON が検知されることで設定されることとなる。30

#### 【 0 3 7 3 】

一方、スイッチ入力判定処理の実行間隔は約 2 . 24 ms であることから、スタートスイッチ 7 が ON となってから最低でも約 2 . 24 ms 以上 ON が継続して検知されることを条件にスタートスイッチ 7 の立上りエッジが設定され、スタートスイッチ 7 の操作が検出されることとなる。

#### 【 0 3 7 4 】

このように本実施例では、図 4 2 に示すように、ゲームを開始可能な状態においてスタートスイッチ 7 の ON が一定期間（最低でも約 2 . 24 ms ）以上継続して検知されたことを条件に、スタートスイッチ 7 の操作が検出され、ゲームが開始されるようになっており、静電気などのノイズによってスタートスイッチ 7 の ON が誤って検出されたにも関わらず、ゲームが開始してしまうことを防止できる。40

#### 【 0 3 7 5 】

尚、本実施例では、ゲームを開始可能な状態においてスタートスイッチ 7 の ON が一定期間（最低でも約 2 . 24 ms ）以上継続して検知されたことを条件に、スタートスイッチ 7 の操作が検出され、ゲームが開始されるようになっているが、例えば、1 回のスイッチ入力判定処理においてスタートスイッチ 7 の検出状態を複数回確認し、全てにおいて ON が判定されたことを条件に、スタートスイッチの操作が検出され、ゲームが開始されるようにも良く、このような構成であっても静電気などのノイズによってスタートスイ50

ツチ 7 の O N が誤って検出されたにも関わらず、ゲームが開始してしまうことを防止できる。

#### 【 0 3 7 6 】

また、本実施例では、図 4 2 に示すように、ゲームを開始可能な状態において、スタートスイッチ 7 の立上りエッジが検出されている場合に、乱数ラッチフラグが設定されているか否か、すなわちスタートスイッチ 7 の操作により数値データがラッチされ、乱数値レジスタ R 1 D に数値データが格納されているか否かを確認し、乱数値レジスタ R 1 D に数値データが格納されている場合にのみゲームを開始させるようになっている。

#### 【 0 3 7 7 】

このため、乱数値レジスタ R 1 D に数値データが格納されていない、すなわち内部抽選に用いる乱数値がラッチされていないにも関わらず、ゲームが開始して内部抽選が行われてしまうことがなく、スタートスイッチ 7 が操作されたタイミングでラッチされた数値データを用いて確実に内部抽選を行うことができる。

#### 【 0 3 7 8 】

以上、本発明の実施例を図面により説明してきたが、本発明はこの実施例に限定されるものではなく、本発明の主旨を逸脱しない範囲における変更や追加があっても本発明に含まれることは言うまでもない。

#### 【 0 3 7 9 】

例えば、前記実施例では、遊技機の一例としてスロットマシンを適用しているが、電断時や瞬停時において以前の制御状態に復帰可能な構成を備える遊技機或いは乱数回路を備え、当該乱数回路にて抽出した数値データに基づいて遊技に関連する抽選を行う遊技機であれば良く、遊技球を用いて遊技を行う弾球遊技機にも本発明を適用可能である。

#### 【 符号の説明 】

##### 【 0 3 8 0 】

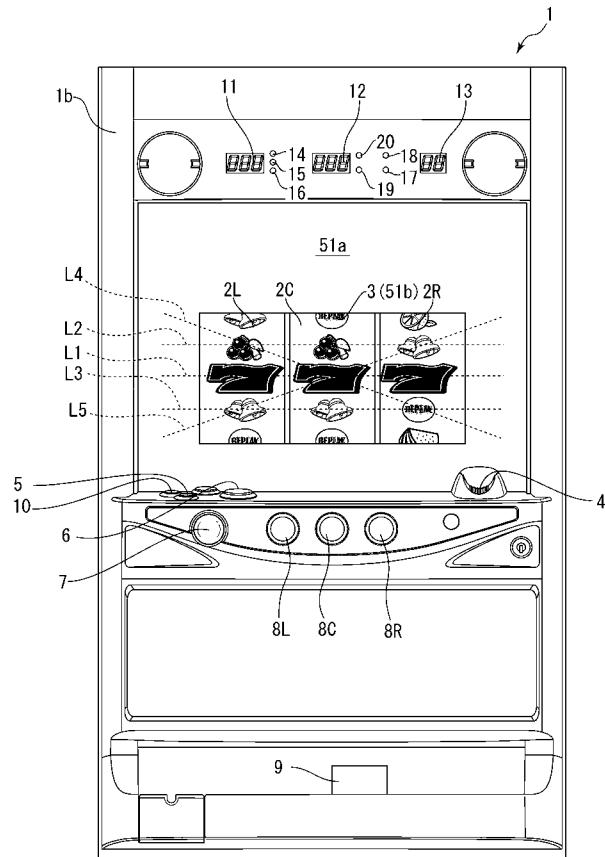
- 1 スロットマシン
- 2 L、2 C、2 R リール
- 6 MAX BET スイッチ
- 7 スタートスイッチ
- 8 L、8 C、8 R ストップスイッチ
- 4 1 メイン制御部
- 9 1 サブ制御部
- 5 0 5 C P U
- 5 0 6 R O M
- 5 0 7 R A M
- 5 0 9 亂数回路
- 5 5 9 A 亂数値レジスタ

10

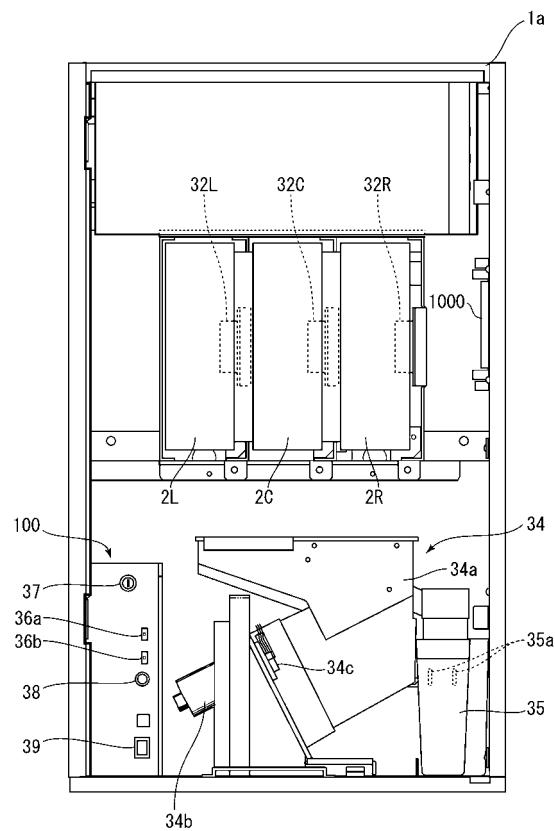
20

30

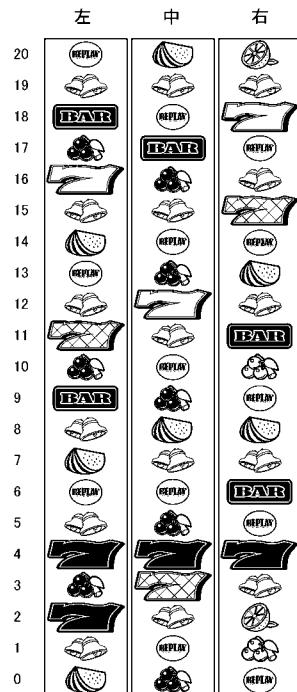
【図1】



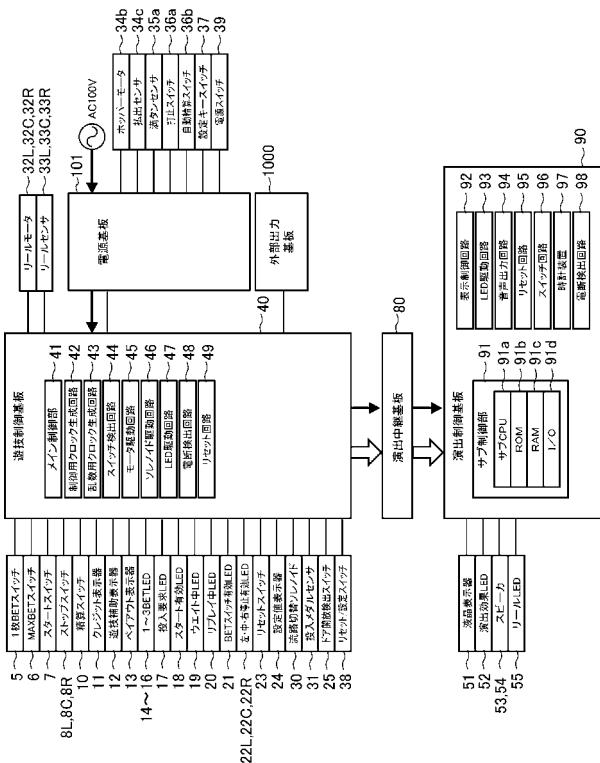
【図2】



【図3】

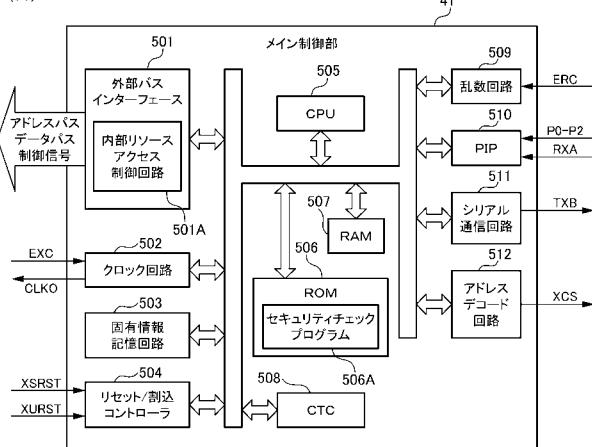


【図4】

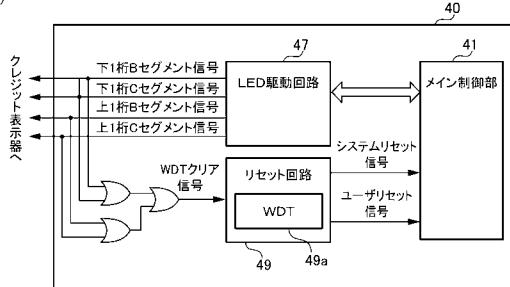


【図5】

(A)



(B)



【図6】

アドレス 種別 内容

0000H	ROM	ユーザプログラムエリア
1FC0H		プログラム管理エリア
1FFFH	内蔵レジスタ	内蔵レジスタエリア
2000H		:
20FFH		:
7E00H	RAM	I/Oマップメモリマップ
7FFFH		:
FDD0H	アドレスデコード回路	XCSデコードエリア
FDFBH		:
FFFFH		:

【図7】

(A) プログラム管理エリア(主要部分)

アドレス(H)	用途	内容
1FC0～1FC7	内部リソースアクセス制御回路	ヘッダ(KHDR)
1FD9	リセット/割込コントローラ等	機能設定(KFCS)
1FDD	乱数回路等	第1乱数初期設定(KRS1)
1FDE	乱数回路等	第2乱数初期設定(KRS2)
1FDF	リセット/割込コントローラ	割込初期設定(KIS)
1FED	CPU等	セキュリティ時間設定(KSES)

(B) 内蔵レジスタエリア(主要部分)

アドレス(H)	用途	内容
201A	シリアル通信回路	第1チャネル通信設定レジスタ(SAFM)
201B	シリアル通信回路	第1チャネル割込制御レジスタ(SAIC)
201C	シリアル通信回路	第1チャネルコマンドレジスタ(SCAM)
201E	シリアル通信回路	第1チャネルステータスレジスタ(SAST)
201F	シリアル通信回路	第1チャネルデータレジスタ(SDAT)
2022	シリアル通信回路	第2チャネル通信設定レジスタ(SBFM)
2023	シリアル通信回路	第2チャネル割込制御レジスタ(SBIC)
2024	シリアル通信回路	第2チャネルコマンドレジスタ(SCBM)
2026	シリアル通信回路	第2チャネルステータスレジスタ(SBST)
2027	シリアル通信回路	第2チャネルデータレジスタ(SBDT)
2028	リセット/割込コントローラ	割込マスクレジスタ(IMR)
2029	リセット/割込コントローラ	割込待ちモニタレジスタ(IRR)
202A	リセット/割込コントローラ	割込モニタレジスタ(ISR)
2030	乱数回路	乱数ラッチ選択レジスタ(RDLS)
2031	乱数回路	乱数割込制御レジスタ(RDIC)
2032	乱数回路	乱数値取込レジスタ(RDLT)
2033	乱数回路	乱数ラッチフレゲジスタ(RDFM)
2034	乱数回路	乱数列変更レジスタ(RDSC)
2038	乱数回路	乱数値レジスタR1D下位バイト
2039	乱数回路	乱数値レジスタR1D上位バイト
203A	乱数回路	乱数値レジスタR2D下位バイト
203B	乱数回路	乱数値レジスタR2D上位バイト
208C	リセット/割込コントローラ等	内部情報レジスタ(CIF)
2090	PIP	入力ポートレジスタ(PI)

【図8】

(A) ヘッダ(KHDR)

アドレス(H)	設定データ(H)	左記以外
1FC0	4C 00	
1FC1	45 00	
1FC2	34 00	
1FC3	33 00	
1FC4	38 00	
1FC5	30 00	
1FC6	41 00	
1FC7	53 00	
ROM読み出し許可／禁止	許可 禁止 許可	
バス出力マスク有効／無効	有効 有効 無効	

(B) 機能設定(KFCS)

ビット番号	設定内容
7-5	ウォッチドッグ動作許可／禁止
4	TXB端子設定 0:シリアル通信第2チャネル送信用 1:チップセレクト(XCS13)
3	TXA端子設定 0:シリアル通信第1チャネル送信用 1:チップセレクト(XCS12)
2	RXA端子設定 0:シリアル通信第1チャネル受信用 1:入力ポートP5
1	NMI接続設定 0:CPU接続 1:CPU非接続(入力ポートP4)
0	XINT接続設定 0:CPU接続 1:CPU非接続(入力ポートP3)

【図 9】

(A) 第1乱数初期設定(KRS1)

ビット番号	設定内容
7-4	省略
3	乱数回路使用設定 0:未使用 1:使用
2	乱数更新クロック設定 0:内部システムクロック(SCLK) 1:乱数用クロック(RCLK)の2分周
1-0	乱数更新規則設定 00:変更しない 01:ソフトウェアで変更(2周目以降) 10:自動で変更(2周目以降)

(B) 第2乱数初期設定(KRS2)

ビット番号	設定内容
7-4	省略
3-2	固定ビット(00B)
1	乱数のスタート値設定 0:デフォルト値(0001H) 1:IDナンバーに基づく値
0	乱数のスタート値変更設定 (システムリセット毎) 0:変更しない 1:変更する

(C) 割込初期設定(KIIS)

ビット番号	設定内容
7-4	割込ベクタ上位4ビット
3-0	マスカブル割込要因の優先度設定 00H~02H:06H:CTCからの割込を最優先 03H:07H:乱数回路からの割込を最優先 04H:05H:シリアル通信回路からの割込を最優先

【図 10】

(A) セキュリティ時間設定(KSES)

ビット番号	設定内容
7-6	乱数用クロック異常検出設定
5	固定値(0B)
4-3	可変セキュリティ時間設定
2-0	延長セキュリティ時間設定

(B) KSES ビット番号7-6

設定値	設定内容
00	RCLK < SCLK
01	RCLK < SCLK/2
10	RCLK < SCLK/4
11	RCLK < SCLK/8

(C) KSES ビット番号4-3

設定値	設定内容
00	延長しない
01	ショートモード
10	ロングモード

(D) KSES ビット番号2-0

設定値	設定内容
000	延長しない
001	$2^{22} \times T_{SCLK} \times 1$
010	$2^{22} \times T_{SCLK} \times 2$
011	$2^{22} \times T_{SCLK} \times 3$
100	$2^{22} \times T_{SCLK} \times 4$
101	$2^{22} \times T_{SCLK} \times 5$
110	$2^{22} \times T_{SCLK} \times 6$
111	$2^{22} \times T_{SCLK} \times 7$

T<sub>SCLK</sub>: 内部システムクロック(SCLK)の周期

【図 11】

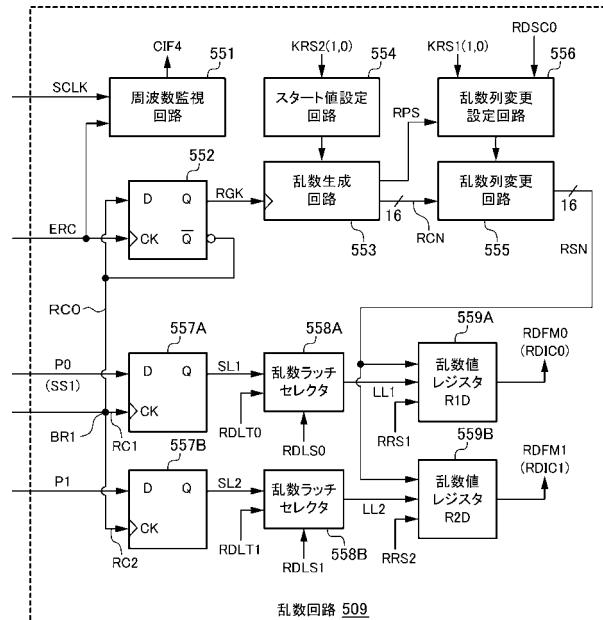
(A)

ビット番号	7	6	5	4	3	2	1	0
CIF	-	-	-	CIF4	-	CIF2	CIF1	CIF0

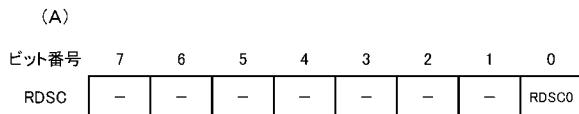
(B) 内部情報データ(CIF)

ビット番号	設定内容
4	乱数用クロック異常指示 0:異常なし 1:異常あり
2	システムリセット指示 0:システムリセット未発生 1:システムリセット発生
1	WDTタイムアウト指示 0:タイムアウト未発生 1:タイムアウト発生
0	IAT発生指示 0:IAT発生なし 1:IAT発生あり

【図 12】



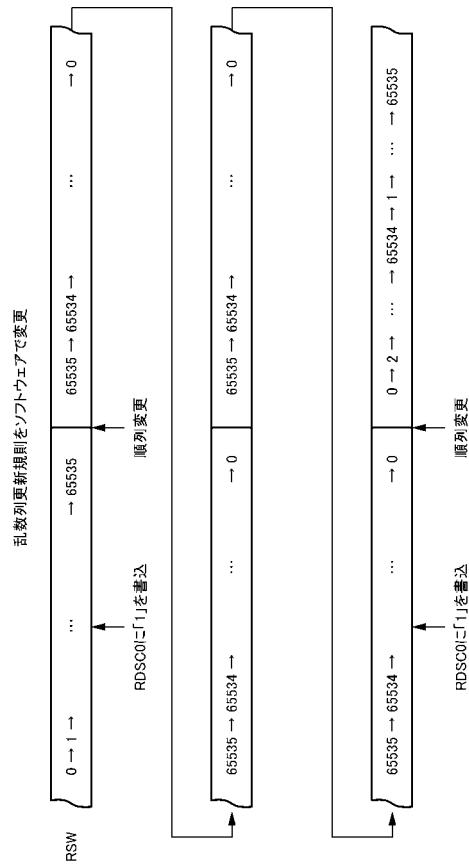
【図13】



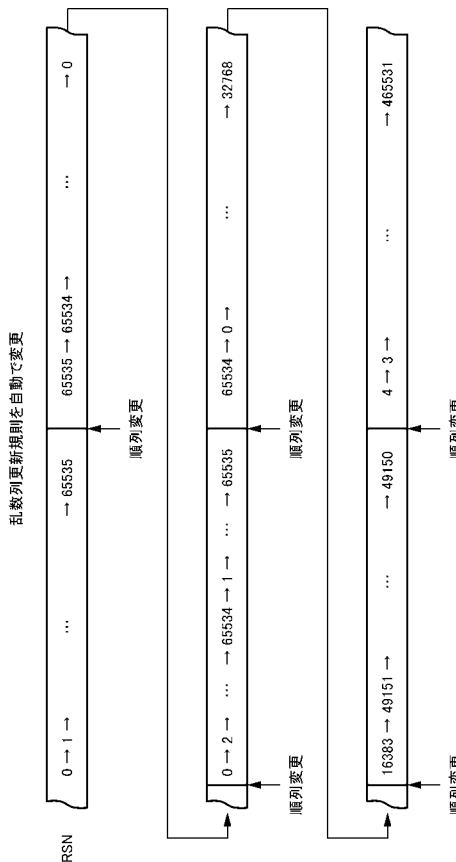
(B)乱数列変更要求データ(RDSC)

ビット番号	設定内容
0	乱数列の変更要求 0:乱数列変更要求なし 1:乱数列変更要求あり

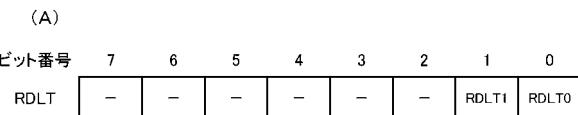
【図14】



【図15】



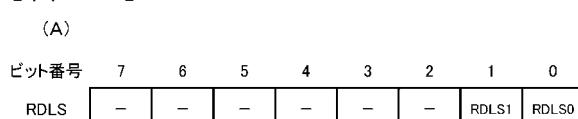
【図16】



(B)乱数値取込指定データ(RDLT)

ビット番号	設定内容
1	乱数値レジスタR2Dへの取込指定 0:取込実行指定なし 1:取込実行指定あり
0	乱数値レジスタR1Dへの取込指定 0:取込実行指定なし 1:取込実行指定あり

【図17】



(B)乱数ラッチ選択データ(RDLS)

ビット番号	設定内容
1	乱数値レジスタR2Dへの取込方法 0:ソフトウェア 1:入力ポートP1
0	乱数値レジスタR1Dへの取込方法 0:ソフトウェア 1:入力ポートP0

【図18】

(A)

ビット番号	7	6	5	4	3	2	1	0
R1D(L)	R1D7	R1D6	R1D5	R1D4	R1D3	R1D2	R1D1	R1D0

(B)

ビット番号	7	6	5	4	3	2	1	0
R1D(H)	R1D15	R1D14	R1D13	R1D12	R1D11	R1D10	R1D9	R1D8

【図19】

(A)

ビット番号	7	6	5	4	3	2	1	0
RDFM	-	-	-	-	-	-	RDFM1	RDFM0

(B)乱数ラッチフラグデータ(RDFM)

ビット番号	設定内容
1	乱数値レジスタR2Dへの取込有無 0:乱数値取込なし 1:乱数値取込みあり
0	乱数値レジスタR1Dへの取込有無 0:乱数値取込なし 1:乱数値取込みあり

【図20】

(A)

ビット番号	7	6	5	4	3	2	1	0
RDIC	-	-	-	-	-	-	RDIC1	RDIC0

(B)乱数割込制御データ(RDIC)

ビット番号	設定内容
1	乱数値レジスタR2Dへの取込時割込許可／禁止 0:割込禁止 1:割込許可
0	乱数値レジスタR1Dへの取込時割込許可／禁止 0:割込禁止 1:割込許可

【図21】

(A)

ビット番号	7	6	5	4	3	2	1	0
PI	-	-	PI5	PI4	PI3	PI2	PI1	PI0

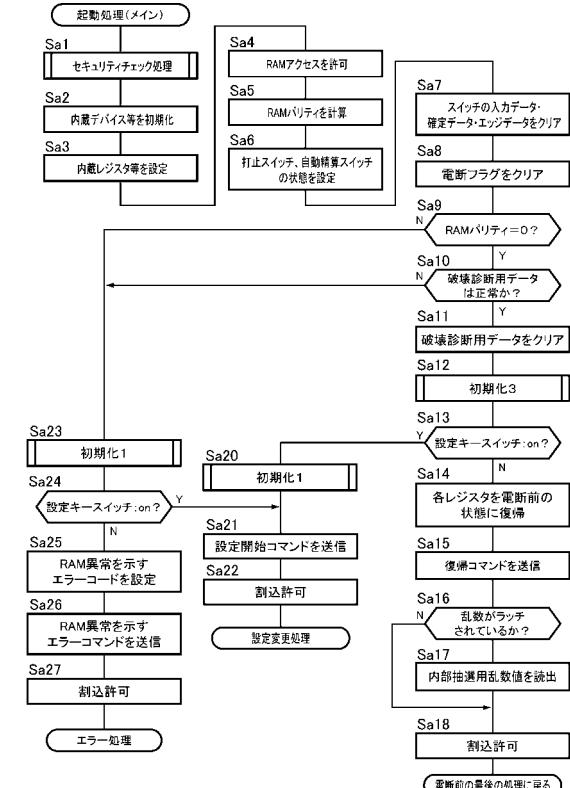
(B)入力ポートデータ(PI)

ビット番号	設定内容
5	P5(RXA)の端子状態
4	P4(XNMI)の端子状態
3	P3(XINT)の端子状態
2	P2の端子状態
1	P1の端子状態
0	P0の端子状態

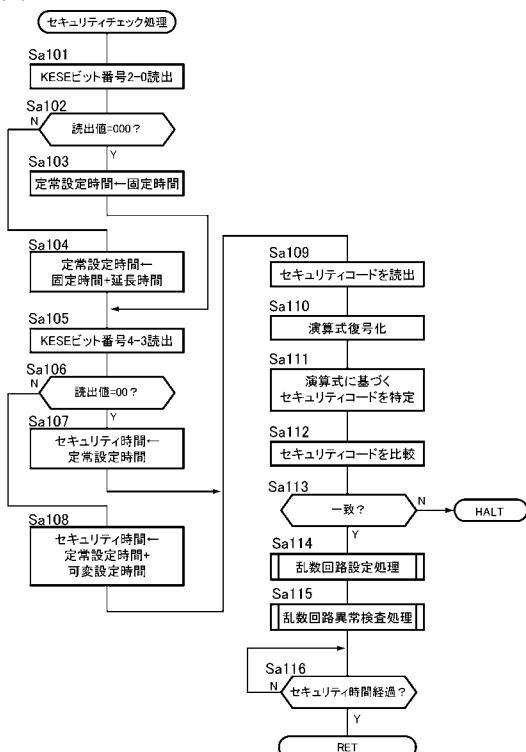
【図22】

役	回路組み合わせ	選択状態		
		通常	当選中	RB(BB)
BB	黒7-黒7-黒7	○	×	×
	緑7-緑7-緑7			
	白7-白7-白7			
RB	緑7-緑7-黒7	○	×	×
	リプレイ-リプレイ-リプレイ	○	○	×
	BAR-リフレイ-リフレイ			
スイカ	黒7-リプレイ-リプレイ			
	スイカ-スイカ-スイカ	○	○	○
	スイカ-スイカ-BAR			
チエリー	ANY-ANY-白チエリー	○	○	○
ベル	ベル-ベル-ベル	○	○	○

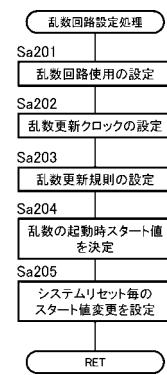
【図23】



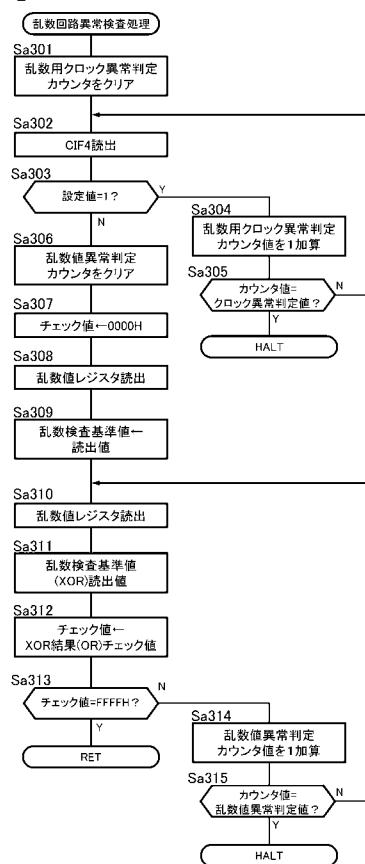
【図24】



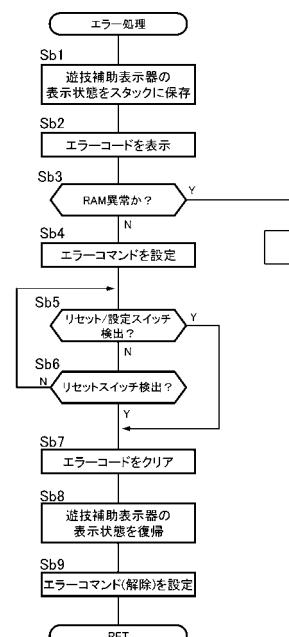
【図25】



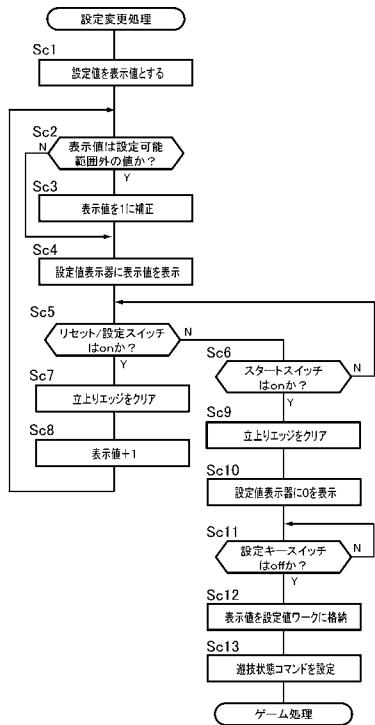
【図26】



【 図 27 】



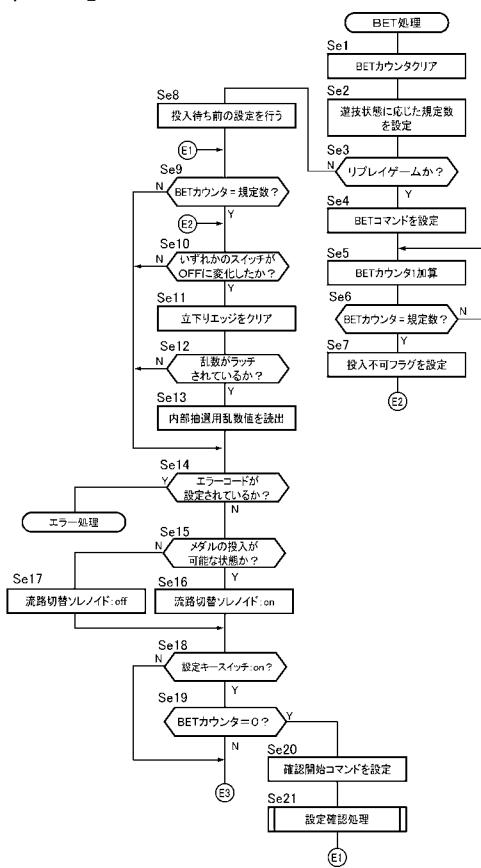
【図28】



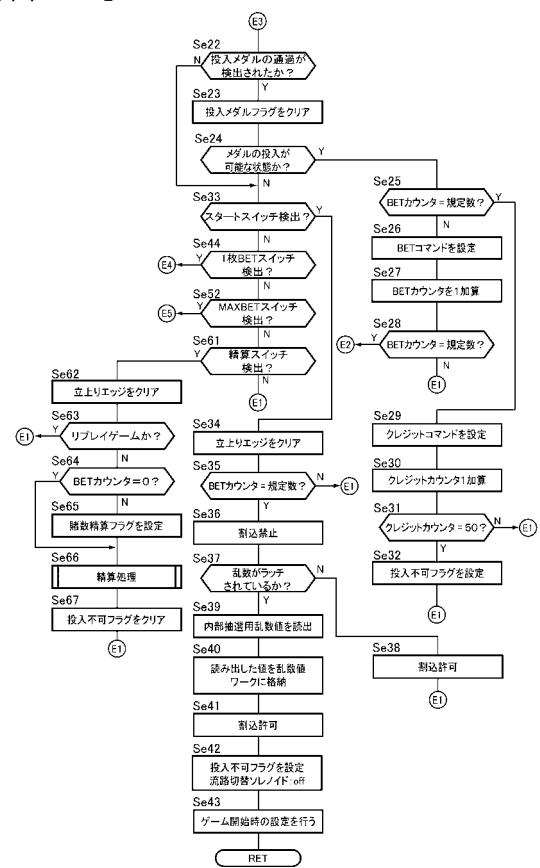
【図29】



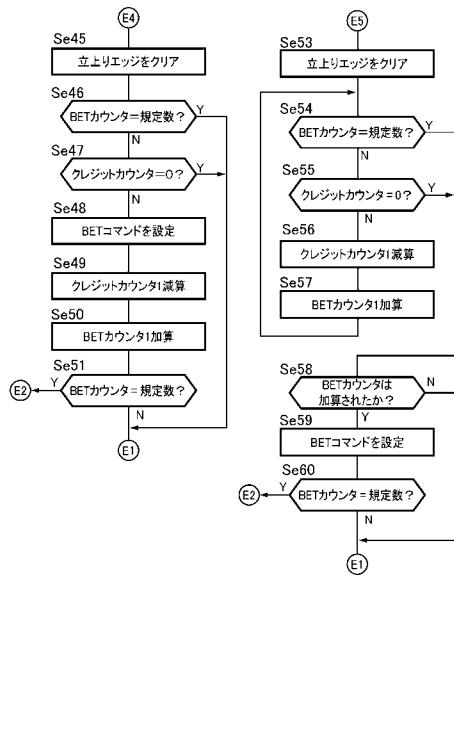
【図30】



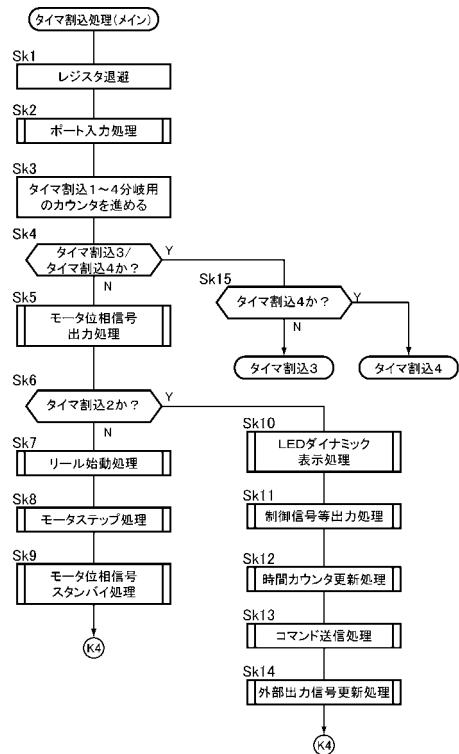
【図31】



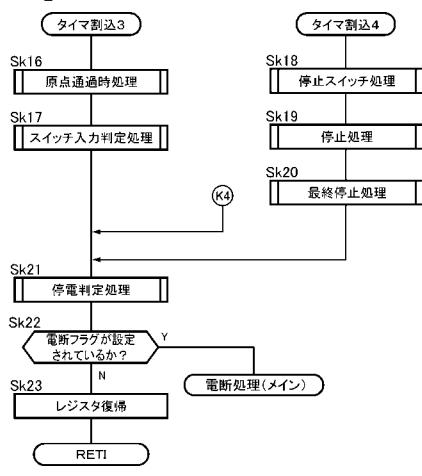
【図3-2】



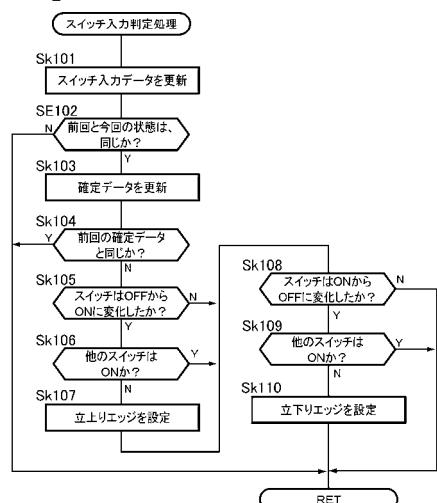
【図3-3】



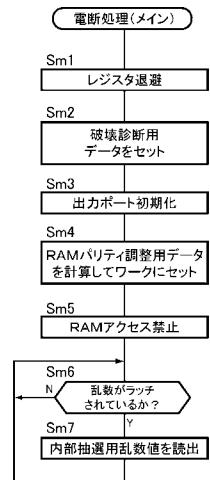
【図3-4】



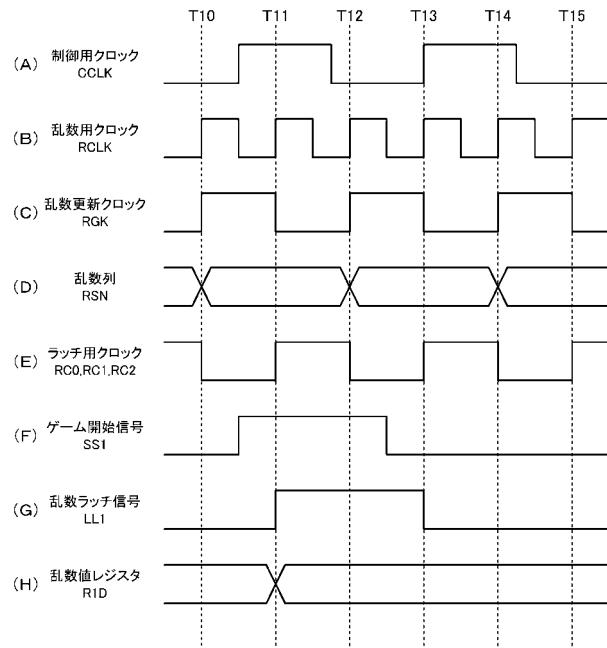
【図3-5】



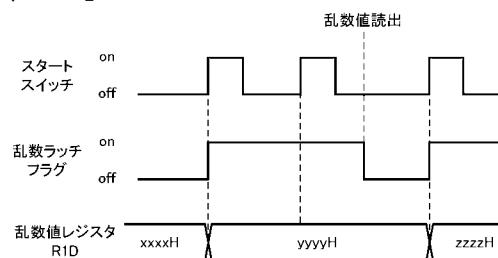
【図36】



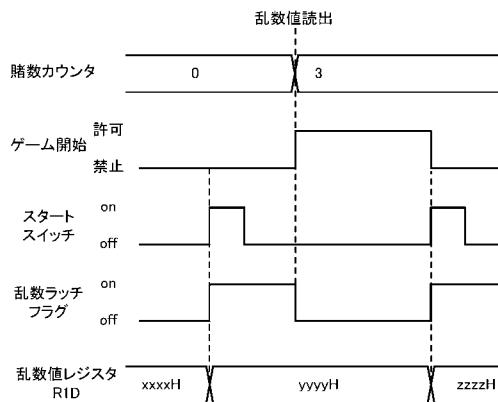
【図37】



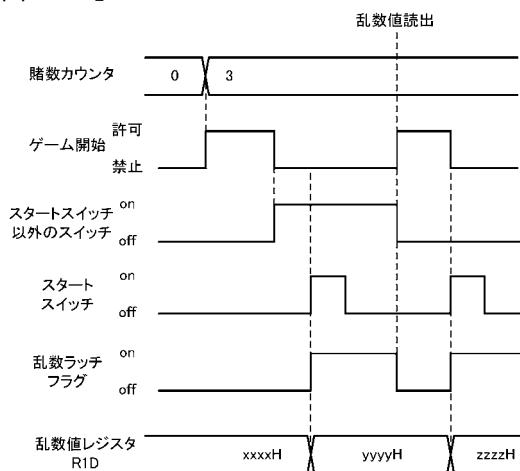
【図38】



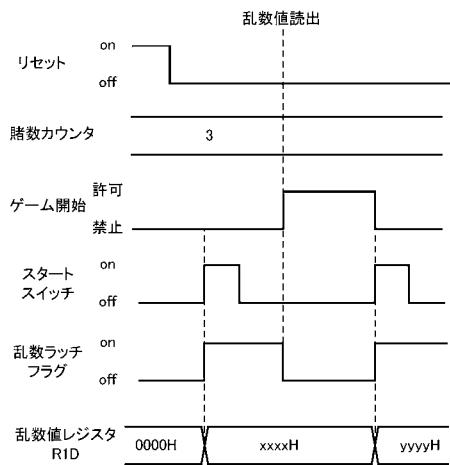
【図39】



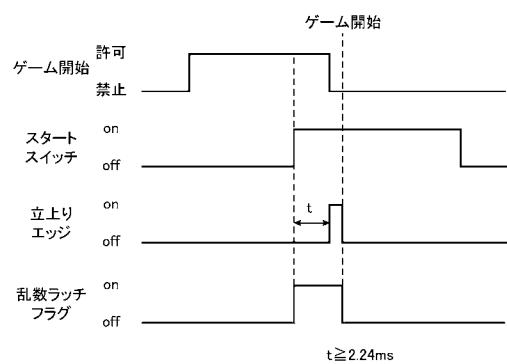
【図40】



【図4-1】



【図4-2】



---

フロントページの続き

(72)発明者 小倉 敏男  
東京都渋谷区渋谷三丁目29番14号 株式会社三共内

(72)発明者 近藤 昇  
東京都渋谷区渋谷三丁目29番14号 株式会社三共内

審査官 太田 恒明

(56)参考文献 特許第4415057(JP, B1)  
特開2010-115296(JP, A)  
特許第5160473(JP, B2)

(58)調査した分野(Int.Cl., DB名)

A 63 F 5 / 04  
A 63 F 7 / 02