

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5552244号
(P5552244)

(45) 発行日 平成26年7月16日(2014.7.16)

(24) 登録日 平成26年5月30日(2014.5.30)

(51) Int.Cl. F 1
A 6 3 F 5/04 (2006.01) 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 7 (全 82 頁)

(21) 出願番号	特願2009-58624 (P2009-58624)	(73) 特許権者	000144153
(22) 出願日	平成21年3月11日 (2009.3.11)		株式会社三共
(65) 公開番号	特開2010-207478 (P2010-207478A)		東京都渋谷区渋谷三丁目29番14号
(43) 公開日	平成22年9月24日 (2010.9.24)	(74) 代理人	100098729
審査請求日	平成24年2月20日 (2012.2.20)		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	100148161
			弁理士 秋庭 英樹

最終頁に続く

(54) 【発明の名称】 スロットマシン

(57) 【特許請求の範囲】

【請求項1】

遊技用価値を用いて1ゲームに対して所定数の賭数を設定することによりゲームが開始可能となるとともに、各々が識別可能な複数種類の図柄を変動表示可能な可変表示装置に表示結果が導出されたことにより1ゲームが終了し、前記可変表示装置に導出された表示結果に応じて入賞が発生可能とされたスロットマシンであって、

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、

ゲームを開始する際に操作される開始操作手段と、

数値データを更新する数値更新手段と、

ゲームを開始させることが可能な開始可能状態において、前記開始操作手段が操作されたことによりゲームを開始させるゲーム開始手段と、

前記開始操作手段が操作されたときに、前記開始可能状態であるか否かに関わらず、前記数値更新手段によって更新された数値データを乱数値として抽出し、第1の数値データ格納領域に格納する乱数抽出手段と、

前記第1の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は、前記乱数抽出手段により新たな数値データが格納されることがなく、該第1の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数値データ保持手段と、

一定時間毎に、前記第1の数値データ格納領域に数値データが格納されているか否かを

判定する数値データ格納判定手段と、

前記第1の数値データ格納判定手段が前記数値データ格納領域に数値データが格納されていると判定したときに、前記第1の数値データ格納領域に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とともに、該読み出した数値データを前記バックアップ領域に設けられた第2の数値データ格納領域に格納する数値データ読出手段と、

前記ゲーム開始手段がゲームを開始させるときに、前記第2の数値データ格納領域に格納されている数値データを用いて入賞の発生を許容するか否かを決定する事前決定手段と、

を備える

ことを特徴とするスロットマシン。

10

【請求項2】

遊技用価値を用いて1ゲームに対して所定数の賭数を設定することによりゲームが開始可能となるとともに、各々が識別可能な複数種類の図柄を変動表示可能な可変表示装置に表示結果が導出されたことにより1ゲームが終了し、前記可変表示装置に導出された表示結果に応じて入賞が発生可能とされたスロットマシンであって、

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、

ゲームを開始する際に操作される開始操作手段と、

数値データを更新する数値更新手段と、

20

ゲームを開始させることが可能な開始可能状態において、前記開始操作手段が操作されたことによりゲームを開始させるゲーム開始手段と、

前記開始操作手段が操作されたときに、前記開始可能状態であるか否かに関わらず、前記数値更新手段によって更新された数値データを乱数値として抽出し、第1の数値データ格納領域に格納する乱数抽出手段と、

前記第1の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は、前記乱数抽出手段により新たな数値データが格納されることがなく、該第1の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数値データ保持手段と、

前記第1の数値データ格納領域に数値データが格納されたときに割込を発生させる割込発生手段と、

30

前記割込発生手段が前記割込を発生させたことに応じて、前記第1の数値データ格納領域に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とともに、該読み出した数値データを前記バックアップ領域に設けられた第2の数値データ格納領域に格納する数値データ読出手段と、

前記ゲーム開始手段がゲームを開始させるときに、前記第2の数値データ格納領域に格納されている数値データを用いて入賞の発生を許容するか否かを決定する事前決定手段と、

を備える

ことを特徴とするスロットマシン。

40

【請求項3】

ゲームを開始させることが不可能な状態から前記開始可能状態となったときに、前記第1の数値データ格納領域に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を備える

ことを特徴とする請求項1または2に記載のスロットマシン。

【請求項4】

前記ゲーム開始手段は、ゲームを開始させることが可能な開始可能状態において、前記開始操作手段の操作が所定期間継続して検出されたことを条件に、ゲームを開始させる

ことを特徴とする請求項1～3のいずれかに記載のスロットマシン。

【請求項5】

50

前記スロットマシン毎に個別に割り当てられた識別符号が記憶される不揮発性メモリと

電力供給が開始したときに前記不揮発性メモリに記憶されている識別符号に基づいて初期数値データを生成する初期数値データ生成手段と、

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から前記数値データの更新を開始する

ことを特徴とする請求項 1 ~ 4 のいずれかに記載のスロットマシン。

【請求項 6】

前記数値更新手段は、ゲームの進行制御を行うゲーム制御手段を動作させるための動作クロックとは周期の異なる動作クロックを入力して前記数値データを更新する

ことを特徴とする請求項 1 ~ 5 のいずれかに記載のスロットマシン。

【請求項 7】

前記数値更新手段は、所定周期の動作クロックを入力して前記数値データを更新し、

前記スロットマシンは、前記数値更新手段に入力される動作クロックの入力状態に基づき、前記数値更新手段の動作状態に異常が発生したか否かを判定する動作クロック異常判定手段を備える

ことを特徴とする請求項 1 ~ 6 のいずれかに記載のスロットマシン。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、各々が識別可能な複数種類の識別情報を変動表示可能な可変表示装置の表示結果に応じて所定の入賞が発生可能なスロットマシンに関する。

【背景技術】

【0002】

スロットマシンは、一般に、外周部に識別情報としての複数種類の図柄が描かれた複数（通常は3つ）のリールを有する可変表示装置を備えており、まず遊技者のBET操作により賭数を設定し、規定の賭数が設定された状態でスタート操作することによりリールの回転が開始し、各リールに対応して設けられた停止ボタンを操作することにより回転を停止する。そして、全てのリールの回転を停止したときに入賞ライン上に予め定められた入賞図柄の組合せ（例えば、7 - 7 - 7、以下図柄の組合せを役とも呼ぶ）が揃ったことによ

って入賞が発生する。すなわち遊技者の操作によってゲームが進行するようになっている。

【0003】

また、この種のスロットマシンにおいては、スタート操作と同時に役の抽選を行い、この抽選に当選したことを条件に当選役の入賞が可能となるものが一般的であり、この役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を設け、スタート操作が検出された際に、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】

【0004】

【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】

【0005】

特許文献1に記載の遊技機のように、スタート操作が検出された際にカウンタの数値を抽出して抽選を行う方法では、スタート操作が検出されてカウンタの数値を抽出した後、静電気などによりスタート操作の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、スタート操作の検出によって抽出された数値とは異なる数値によ

10

20

30

40

50

て抽選が行われてしまうこととなる。

【0006】

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、スタート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、カウンタから抽出された数値が保持されたままとなり、本来のスタート操作がなされたタイミング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。

【0007】

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、ゲームの開始操作がなされたタイミングで抽出された数値とは異なる数値を用いて抽選が行われてしまうことを防止できるスロットマシン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】

【0008】

上記課題を解決するために、本発明の請求項1に記載のスロットマシンは、遊技用価値を用いて1ゲームに対して所定数の賭数を設定することによりゲームが開始可能となるとともに、各々が識別可能な複数種類の図柄を変動表示可能な可変表示装置（リール2L、2C、2R）に表示結果が導出されたことにより1ゲームが終了し、前記可変表示装置（リール2L、2C、2R）に導出された表示結果に応じて入賞が発生可能とされたスロットマシン（スロットマシン1）であって、

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、

ゲームを開始する際に操作される開始操作手段（スタートスイッチ7）と、

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、

ゲームを開始させることが可能な開始可能状態において、前記開始操作手段（スタートスイッチ7）が操作されたことによりゲームを開始させるゲーム開始手段と、

前記開始操作手段（スタートスイッチ7）が操作されたときに、前記開始可能状態であるか否かに関わらず、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、第1の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、

前記第1の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がなく、該第1の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、

一定時間（約2.24ms）毎に、前記第1の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（乱数ラッチフラグが設定されているか否か）を判定する数値データ格納判定手段と、

前記数値データ格納判定手段が前記第1の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（乱数ラッチフラグが設定されている）と判定したときに、前記第1の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とともに、該読み出した数値データを前記バックアップ領域に設けられた第2の数値データ格納領域（乱数値格納ワーク）に格納する数値データ読出手段と、

前記ゲーム開始手段がゲームを開始させるときに、前記第2の数値データ格納領域（乱数値格納ワーク）に格納されている数値データを用いて入賞の発生を許容するか否かを決定する事前決定手段（内部抽選）と、

を備える

ことを特徴としている。

10

20

30

40

50

この特徴によれば、開始操作手段の操作により抽出された数値データが第1の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て第2の数値データ格納領域に格納されるまでは、第1の数値データ格納領域に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、静電気などにより開始操作手段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い。

また、一定時間毎に、第1の数値データ格納領域に数値データが格納されているか否か判定され、第1の数値データ格納領域に数値データが格納されている場合には、第1の数値データ格納領域に格納されている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、開始可能状態となる前に開始操作手段が操作されて数値データ格納領域に数値データが格納され、その数値データが保持されても、その状態が一定時間を超えて継続することがなく、新たな数値データを格納可能な状態となるため、開始可能状態となる前に開始操作手段が操作されて数値データが格納されても、開始可能状態において開始操作手段が操作されたタイミングで抽出した数値データを取得することが可能となる。

また、数値データ格納判定手段が第1の数値データ格納領域に新たな数値データが格納されていると判定し、第1の数値データ格納領域から数値データが読み出される毎に、第2の数値データ格納領域の数値データが第1の数値データ格納領域から読み出された数値データ、すなわち新たに抽出された数値データに更新されるとともに、事前決定手段は、第1の数値データ格納領域に格納されている数値データではなく、第2の数値データ格納領域に格納された数値データを用いるので、第1の数値データ格納領域から数値データが読み出された後に、開始操作手段の信号線にノイズがのって数値データが変わってしまうても事前決定手段が用いる第2の数値データ格納領域の数値データに影響することがなく、このような場合であっても、開始可能状態において開始操作手段が操作されたタイミングで抽出した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を行うことができる。

【0009】

本発明の請求項2に記載のロットマシンは、

遊技用価値を用いて1ゲームに対して所定数の賭数を設定することによりゲームが開始可能となるとともに、各々が識別可能な複数種類の図柄を変動表示可能な可変表示装置（リール2L、2C、2R）に表示結果が導出されたことにより1ゲームが終了し、前記可変表示装置（リール2L、2C、2R）に導出された表示結果に応じて入賞が発生可能とされたロットマシン（ロットマシン1）であって、

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、

ゲームを開始する際に操作される開始操作手段（スタートスイッチ7）と、

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、

ゲームを開始させることが可能な開始可能状態において、前記開始操作手段（スタートスイッチ7）が操作されたことによりゲームを開始させるゲーム開始手段と、

前記開始操作手段（スタートスイッチ7）が操作されたときに、前記開始可能状態であるか否かに関わらず、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、第1の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、

前記第1の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がなく、該第1の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、

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たときに割込（乱数ラッチ割込）を発生させる割込発生手段と、

10

20

30

40

50

前記割込発生手段が前記割込（乱数ラッチ割込）を発生させたことに応じて、前記数値データ格納領域（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とともに、該読み出した数値データを前記バックアップ領域に設けられた第 2 の数値データ格納領域（乱数値格納ワーク）に格納する数値データ読出手段と、

前記ゲーム開始手段がゲームを開始させるときに、前記第 2 の数値データ格納領域（乱数値格納ワーク）に格納されている数値データを用いて入賞の発生を許容するか否かを決定する事前決定手段（内部抽選）と、

を備える

ことを特徴としている。

10

この特徴によれば、開始操作手段の操作により抽出された数値データが第 1 の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て第 2 の数値データ格納領域に格納されるまでは、第 1 の数値データ格納領域に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、静電気などにより開始操作手段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い。

また、数値データ格納領域に数値データが格納されると割込が発生し、割込に応じて数値データ格納領域に格納されている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、開始可能状態となる前に開始操作手段が操作されて数値データ格納領域に数値データが格納されても、割込の発生に応じて直ちに数値データが読み出され、新たな数値データを格納可能な状態となるため、開始可能状態において開始操作手段が操作されたタイミングで抽出した数値データを取得することが可能となる。

20

また、第 1 の数値データ格納領域に新たな数値データが格納されて割込が発生し、第 1 の数値データ格納領域から数値データが読み出される毎に、第 2 の数値データ格納領域の数値データが第 1 の数値データ格納領域から読み出された数値データ、すなわち新たに抽出された数値データに更新されるとともに、事前決定手段は、第 1 の数値データ格納領域に格納されている数値データではなく、第 2 の数値データ格納領域に格納された数値データを用いるので、第 1 の数値データ格納領域から数値データが読み出された後に、開始操作手段の信号線にノイズがのって数値データが変わってしまうても事前決定手段が用いる第 2 の数値データ格納領域の数値データに影響することがなく、このような場合であっても、開始可能状態において開始操作手段が操作されたタイミングで抽出した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を行うことができる。

30

【 0 0 1 0 】

尚、請求項 1、2 において所定数の賭数とは、少なくとも 1 以上の賭数であって、2 以上の賭数が設定されることや最大賭数が設定されることでゲームが開始可能となるようにしても良い。また、複数の遊技状態に応じて定められた賭数が設定されることでゲームが開始可能となるようにしても良い。

また、請求項 1、2 において数値データ保持手段は、前記第 1 の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、第 1 の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よいし、前記第 1 の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第 1 の数値データ格納領域への格納を禁止することにより、第 1 の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よい。

40

【 0 0 1 1 】

本発明の請求項 3 に記載のスロットマシンは、請求項 1 または 2 に記載のスロットマシンであって、

ゲームを開始させることが不可能な状態から前記開始可能状態となったときに、前記数値データ格納領域（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こと

50

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を備える

ことを特徴としている。

この特徴によれば、開始操作手段を操作してもゲームが開始しない状態から、開始操作手段の操作によりゲームを開始させることが可能な開始可能状態となったときにも、数値データ格納領域に格納されている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、開始可能状態となる前に開始操作手段が操作されて数値データ格納領域に数値データが格納され、その状態が保持されたままとなっても、開始可能状態になると同時に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。

10

【0012】

本発明の請求項4に記載のロットマシンは、請求項1～3のいずれかに記載のロットマシンであって、

前記ゲーム開始手段は、前記開始可能状態において、前記開始操作手段（スタートスイッチ7）の操作が所定期間（約2.24ms）継続して検出されたことを条件に、ゲームを開始させる

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開始操作手段が誤って検出されたにも関わらず、ゲームが開始し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を防止できる。

20

【0013】

本発明の請求項5に記載のロットマシンは、請求項1～4のいずれかに記載のロットマシンであって、

前記ロットマシン毎に個別に割り当てられた識別符号（IDナンバー）が記憶される不揮発性メモリ（ROM506）と、

電力供給が開始したときに前記不揮発性メモリ（ROM506）に記憶されている識別符号（ID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）と、

をさらに備え、

前記数値更新手段（乱数列変更回路555）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

30

ことを特徴としている。

この特徴によれば、ロットマシン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、ロットマシン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。

【0014】

本発明の請求項6に記載のロットマシンは、請求項1～5のいずれかに記載のロットマシンであって、

前記数値更新手段（乱数列変更回路555）は、ゲームの進行制御を行うゲーム制御手段（CPU505）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

40

ことを特徴としている。

この特徴によれば、ゲーム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、事前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、ゲーム制御手段に不正基板が接続されてもゲーム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。

【0015】

50

本発明の請求項 7 に記載のロットマシンは、請求項 1 ~ 6 のいずれかに記載のロットマシンであって、

前記数値更新手段（乱数列変更回路 5 5 5）は、所定周期の動作クロック（乱数用クロック）を入力して前記数値データを更新し、

前記ロットマシンは、前記数値更新手段（乱数列変更回路 5 5 5）に入力される動作クロック（乱数用クロック）の入力状態に基づき、前記数値更新手段（乱数列変更回路 5 5 5）の動作状態に異常が発生したか否かを判定する動作クロック異常判定手段（乱数回路異常検査処理）を備える

ことを特徴としている。

この特徴によれば、数値更新手段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が入賞の発生を許容するか否かの決定を行わせる不正を防止できる。

【図面の簡単な説明】

【0016】

【図 1】本発明が適用された実施例のロットマシンの正面図である。

【図 2】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】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。

【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。

【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 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。

。

10

20

30

40

50

【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）において実行する乱数値読出処理の制御内容を示すフローチャートである。

10

【図 36】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 37】メイン制御部が、電断検出回路から電圧低下信号の入力されることによって実行する電断割込処理（メイン）の制御内容を示すフローチャートである。

【図 38】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 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。

20

【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 43】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 44】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。

【図 45】メイン制御部が、乱数をラッチする毎に実行する乱数値ラッチ割込処理の制御内容を示すフローチャートである。

【図 46】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。

30

【発明を実施するための形態】

【0017】

本発明の実施例を以下に説明する。

【実施例 1】

【0018】

本発明が適用されたスロットマシンの実施例 1 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 1 a と、この筐体 1 a の側端に回動自在に枢支された前面扉 1 b と、から構成されている。

【0019】

40

本実施例のスロットマシン 1 の筐体 1 a 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 2 L、2 C、2 R（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 2 L、2 C、2 R に配列された図柄のうち連続する 3 つの図柄が前面扉 1 b 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。

【0020】

リール 2 L、2 C、2 R 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 2 L、2 C、2 R の外周部に

50

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。

【 0 0 2 1 】

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

【 0 0 2 2 】

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 3 3 L、3 3 C、3 3 R と、リール 2 L、2 C、2 R を背面から照射するリール LED 5 5 と、が設けられている。また、リール LED 5 5 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 1 2 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

【 0 0 2 3 】

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 5 1 (図 1 参照) の表示領域 5 1 a が配置されている。液晶表示器 5 1 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 5 1 a の透視窓 3 に対応する透過領域 5 1 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 5 1 a の透過領域 5 1 b を除く領域の裏面には、背後から表示領域 5 1 a を照射するバックライト (図示略) が設けられているとともに、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。

20

【 0 0 2 4 】

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 MAX BET 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 1 0、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

【 0 0 2 5 】

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 1 1、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 1 2、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 1 3 が設けられている。

【 0 0 2 6 】

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 1 BET LED 1 4、賭数が 2 設定されている旨を点灯により報知する 2 BET LED 1 5、賭数が 3 設定されている旨を点灯により報知する 3 BET LED 1 6、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 LED 1 7、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 LED 1 8、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 LED 1 9、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 LED 2 0 が設けられている。

40

【 0 0 2 7 】

MAX BET スイッチ 6 の内部には、1 枚 BET スイッチ 5 及び MAX BET 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

50

LED 21 (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 LED 22 L、22 C、22 R (図4参照) がそれぞれ設けられている。

【0028】

前面扉 1 b 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 1 a 内部に設けられた後述のホッパータンク 34 a (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 34 a 側に流下したメダルを検出する投入メダルセンサ 31 を有するメダルセクタ (図示略)、前面扉 1 b 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) が設けられている。

【0029】

筐体 1 a 内部には、図2に示すように、前述したリール 2 L、2 C、2 R、リールモータ 32 L、32 C、32 R、各リール 2 L、2 C、2 R のリール基準位置をそれぞれ検出可能なリールセンサ 33 L、33 C、33 R (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 34 a、ホッパータンク 34 a 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 34 b、ホッパーモータ 34 b 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 34 c 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。

【0030】

ホッパーユニット 34 の側部には、ホッパータンク 34 a 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 35 a 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留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。

【0031】

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 36 a、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 36 b、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、通常時においてはエラー状態や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、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、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。

【0032】

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 MAX BET 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。

【0033】

入賞ラインとは、各リール2 L、2 C、2 R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1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L1、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L2、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L3、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L4、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L5の5種類が入賞ラインとして定められている。

10

【0034】

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。

【0035】

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L1～L5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L1～L5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

30

【0036】

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40、演出制御基板90、電源基板101が設けられており、遊技制御基板40によって遊技状態が制御され、演出制御基板90によって遊技状態に応じた演出が制御され、電源基板101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。

【0037】

電源基板101には、外部からAC100Vの電源が供給されるとともに、このAC100V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40及び遊技制御基板40を介して接続された演出制御基板90に供給されるようになっている。

40

【0038】

また、電源基板101には、前述したホッパーモータ34b、払出センサ34c、満タンセンサ35a、打止スイッチ36a、自動精算スイッチ36b、設定キースイッチ37、リセット/設定スイッチ38、電源スイッチ39が接続されている。

【0039】

遊技制御基板40には、前述した1枚BETスイッチ5、MAXBETスイッチ6、スタートスイッチ7、ストップスイッチ8L、8C、8R、精算スイッチ10、リセットスイッチ23、投入メダルセンサ31、ドア開放検出スイッチ25、リールセンサ33L、33C、33Rが接続されているとともに、電源基板101を介して前述した払出センサ

50

34c、満タンセンサ35a、打止スイッチ36a、自動精算スイッチ36b、設定キー
スイッチ37、リセット/設定スイッチ38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。

【0040】

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L
ED21、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。

10

【0041】

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4
6、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。

【0042】

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ROM5
06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。

20

【0043】

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CLK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CLKは、例えば図5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4
3は、メイン制御部41の外部にて、制御用クロックCLK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CLKの発振周波数以
下となるようにすれば良い。

30

【0044】

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3
2L、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にリセット信号を与える。

40

【0045】

図5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(パラレルインポートポ
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とを備えて構成

50

される。

【 0 0 4 6 】

図 6 は、メイン制御部 4 1 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 0 0 0 0 H ~ アドレス 1 F F F H の領域は、ROM 5 0 6 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、ROM 5 0 6 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 2 0 0 0 H ~ アドレス 2 0 F F H の領域は、メイン制御部 4 1 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 7 E 0 0 H ~ アドレス 7 F F F H の領域は、RAM 5 0 7 に割り当てられたワークエリアであり、I/O マップやメモリアドレスに割り付けることができる。アドレス F D D 0 H ~ アドレス F D F B H の領域は、アドレスデコード回路 5 1 2 に割り当てられる X C S デコードエリアである。

10

【 0 0 4 7 】

プログラム管理エリアは、CPU 5 0 5 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 K H D R、機能設定 K F C S、第 1 乱数初期設定 K R S 1、第 2 乱数初期設定 K R S 2、割込初期設定 K I I S、セキュリティ時間設定 K S E S などが、含まれている。

【 0 0 4 8 】

プログラム管理エリアに記憶されるヘッダ K H D R は、メイン制御部 4 1 における内部データの読出設定を示す。図 8 (A) は、ヘッダ K H D R 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 4 1 では、ROM 読出防止機能と、バス出力マスク機能とを設定可能である。ROM 読出防止機能は、メイン制御部 4 1 が備える ROM 5 0 6 の記憶データについて、読出動作を許可又は禁止する機能であり、読出禁止に設定された状態では、ROM 5 0 6 の記憶データを読み出すことができない。バス出力マスク機能は、外部バスインタフェース 5 0 1 に接続された外部装置からメイン制御部 4 1 の内部データに対する読出要求があった場合に、外部バスインタフェース 5 0 1 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 K H D R の設定データに対応して、ROM 読出防止機能やバス出力マスク機能の動作組合せが異なるように設定される。図 8 (A) に示す設定データのうち、ROM 読出が許可されるとともに、バス出力マスクが有効となる設定データは、バス出力マスク有効データともいう。また、ROM 読出が禁止されるとともに、バス出力マスクが有効となる設定データ(全て「00H」)は、ROM 読出禁止データともいう。ROM 読出が許可されるとともに、バス出力マスクが無効となる設定データは、バス出力マスク無効データともいう。

20

30

【 0 0 4 9 】

プログラム管理エリアに記憶される機能設定 K F C S は、メイン制御部 4 1 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 K F C S における設定内容の一例を示している。

【 0 0 5 0 】

機能設定 K F C S のビット番号 [7 - 5] は、例えばリセット / 割込コントローラ 5 0 4 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。機能設定 K F C S のビット番号 [4] は、メイン制御部 4 1 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 5 1 1 が使用する第 2 チャネル送信端子 T X B とするか、アドレスデコード回路 5 1 2 が使用するチップセレクト出力端子 X C S 1 3 とするかを指定する T X B 端子設定である。図 8 (B) に示す例において、機能設定 K F C S 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 5 1 1 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 T X B 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 1 兼用端子がアドレスデコード回路 5 1 2 で使用されるチップセレクト出力端子 X C S 1 3 の設定

40

50

となる。本実施例では、機能設定K F C Sのビット番号[4]を“ 0 ”として、第1兼用端子を第2チャンネル送信端子T X Bに設定することにより、演出制御基板9 0との間でのシリアル通信を可能にする。

【 0 0 5 1 】

機能設定K F C Sのビット番号[3]は、メイン制御部4 1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5 1 1が使用する第1チャンネル送信端子T X Aとするか、アドレスデコード回路5 1 2が使用するチップセレクト出力端子X C S 1 2とするかを示すT X A端子設定である。図8 (B) に示す例において、機能設定K F C S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5 1 1での第1チャンネル送信に使用される第1チャンネル送信端子T X A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デコード回路5 1 2で使用されるチップセレクト出力端子X C S 1 2の設定となる。本実施例では、機能設定K F C Sのビット番号[3]を“ 0 ”として、第2兼用端子を第1チャンネル送信端子T X A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

【 0 0 5 2 】

機能設定K F C Sのビット番号[2]は、メイン制御部4 1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5 1 1が使用する第1チャンネル受信端子R X Aとするか、P I P 5 1 0が使用する入力ポートP 5とするかを示すR X A端子設定である。図8 (B) に示す例において、機能設定K F C S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5 1 1での第1チャンネル受信に使用される第1チャンネル受信端子R X A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P I P 5 1 0で使用される入力ポートP 5の設定となる。本実施例では、機能設定K F C Sのビット番号[2]を“ 0 ”として、第3兼用端子を第1チャンネル受信端子R X A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

【 0 0 5 3 】

機能設定K F C Sのビット番号[1]は、メイン制御部4 1における所定の機能兼用端子(第4兼用端子)を、C P U 5 0 5等に接続される外部ノンマスクابل割込端子X N M Iとするか、P I P 5 1 0が使用する入力ポートP 4とするかを示すN M I接続設定である。図8 (B) に示す例において、機能設定K F C S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C P U 5 0 5等に接続される外部ノンマスクابل割込端子X N M Iの設定となる(C P U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P I P 5 1 0で使用される入力ポートP 4の設定となる(C P U非接続)。本実施例では、機能設定K F C Sのビット番号[1]を“ 1 ”として、第4兼用端子をP I P 5 1 0で使用される入力ポートP 4に設定しているが、入力ポートP 4は未使用とされている。

30

【 0 0 5 4 】

機能設定K F C Sのビット番号[0]は、メイン制御部4 1における所定の機能兼用端子(第5兼用端子)を、C P U 5 0 5等に接続される外部マスクابل割込端子X I N Tとするか、P I P 5 1 0が使用する入力ポートP 3とするかを示すX I N T接続設定である。図8 (B) に示す例において、機能設定K F C S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C P U 5 0 5等に接続される外部マスクابل割込端子X I N Tの設定となる(C P U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P I P 5 1 0で使用される入力ポートP 3の設定となる(C P U非接続)。本実施例では、機能設定K F C Sのビット番号[0]を“ 0 ”として、第4兼用端子をC P U 5 0 5等に接続される外部マスクابل割込端子X I N Tに設定することにより、電断検出回路4 8からの電圧低下信号が入力される。

40

【 0 0 5 5 】

プログラム管理エリアに記憶される第1乱数初期設定K R S 1及び第2乱数初期設定K R S 2は、乱数回路5 0 9の初期設定を示す。図9 (A) は、第1乱数初期設定K R S 1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K R S 2にお

50

る設定内容の一例を示している。

【 0 0 5 6 】

第 1 乱数初期設定 K R S 1 のビット番号 [3] は、乱数回路 5 0 9 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 K R S 1 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 5 0 9 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 5 0 9 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 K R S 1 のビット番号 [3] を “ 1 ” として、乱数回路 5 0 9 を使用可能に設定する。

【 0 0 5 7 】

第 1 乱数初期設定 K R S 1 のビット番号 [2] は、乱数回路 5 0 9 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 R G K (図 1 2 参照) を、内部システムクロック S C L K とするか、乱数用クロック R C L K の 2 分周とすること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 K R S 1 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 S C L K を乱数更新クロック R G K に用いる設定となる一方、 “ 1 ” であれば、乱数用クロック R C L K を 2 分周して乱数更新クロック R G K に用いる設定となる。本実施例では、第 1 乱数初期設定 K R S 1 のビット番号 [2] を “ 1 ” として、乱数用クロック R C L K を 2 分周して乱数更新クロック R G K に用いる設定とする。

【 0 0 5 8 】

第 1 乱数初期設定 K R S 1 のビット番号 [1 - 0] は、乱数回路 5 0 9 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 K R S 1 のビット番号 [1 - 0] におけるビット値が “ 0 0 ” であれば、乱数更新規則を変更しない設定となり、 “ 0 1 ” 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、 “ 1 0 ” 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。

【 0 0 5 9 】

第 2 乱数初期設定 K R S 2 のビット番号 [3 - 2] は、固定のビット値 “ 0 0 ” が設定される。尚、図 9 (B) における「 0 0 B 」の “ B ” は 2 進数表示であることを示す。第 2 乱数初期設定 K R S 2 のビット番号 [1 - 0] は、乱数回路 5 0 9 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 K R S 2 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 0 0 0 1 H に設定される一方、 “ 1 ” であるときには、メイン制御部 4 1 毎に付与された固有の識別情報である I D 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 K R S 2 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 I D ナンバーに基づく値に設定する場合には、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 4 1 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 4 1 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 4 0 におけるバックアップ箇所と共通の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、R A M 5 0 7 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

10

20

30

40

50

ント値が保存されることになる。

【 0 0 6 0 】

尚、乱数回路 5 0 9 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 チャンネル対応）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 K R S 1 のビット番号 [3 - 0] と第 2 乱数初期設定 K R S 2 のビット番号 [3 - 0] とを、第 1 チャン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 K R S 1 のビット番号 [7 - 4] や第 2 乱数初期設定 K R S 2 のビット番号 [7 - 4] を（図 9（A）及び（B）では省略）、第 2 チャン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。

【 0 0 6 1 】

プログラム管理エリアに記憶される割込初期設定 K I I S は、メイン制御部 4 1 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 K I I S における設定内容の一例を示している。

【 0 0 6 2 】

割込初期設定 K I I S 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 K I I S 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 K I I S のビット番号 [3 - 0] により「0 0 H」～「0 2 H」及び「0 6 H」のいずれかが指定されれば、C T C 5 0 8 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、「0 3 H」及び「0 7 H」のいずれかが指定されれば、乱数回路 5 0 9 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、「0 4 H」及び「0 5 H」のいずれかが指定されれば、シリアル通信回路 5 1 1 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。

【 0 0 6 3 】

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 K S E S は、乱数用クロック R C L K 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 4 1 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 4 1 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、R O M 5 0 6 の記憶内容が変更されたか否かが検査される。図 1 0（A）は、セキュリティ時間設定 K S E S における設定内容の一例を示している。

【 0 0 6 4 】

セキュリティ時間設定 K S E S のビット番号 [7 - 6] は、乱数用クロック R C L K 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 1 0（B）は、セキュリティ時間設定 K S E S 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 K S E S のビット番号 [7 - 6] は、内部システムクロック S C L K の周波数に基づき、乱数用クロック R C L K 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 K S E S のビット番号「5」は、固定のビット値“0”が設定される。

【 0 0 6 5 】

セキュリティ時間設定 K S E S 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 1 0（C）は、セキュリティ時間設定 K S E S のビット番号 [4 - 3] における設定内容の一例を示している。図 1 0（C）に示す例において、セキュリティ時間設定 K S E S のビット番号 [4 - 3] におけるビット値が“0 0”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“0 1”であればショートモードの設定となり、“1 0”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場合には、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を、シ

10

20

30

40

50

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0~680 μ s(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0~348,160 μ s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0~408 μ s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0~208,896 μ s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。

【0066】

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_{SCLK} 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。

【0067】

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。

【0068】

図5に示すメイン制御部41が備える外部バスインタフェース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フェースである。例えば、外部バスインタフェース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。

【0069】

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

10

20

30

40

50

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。

【0070】

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自体を読出不能として、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読出禁止データとすることで、ROM自体を読出不能としておき、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。

【0071】

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。この読出要求を検出したときに、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、外部装置からの読出要求が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出要求であれば、こ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。外部バスインタフェース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ように制御すれば良い。このように、内部リソースアクセス制御回路501Aは、外部装置からの読出要求が所定の内部データ(例えばROM506の所定領域)の読み出しを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。

【0072】

或いは、内部リソースアクセス制御回路501Aは、内部データの読出要求を検出したときに、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。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

10

20

30

40

50

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。

【0073】

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ていなければならない。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ていなければならない。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。

【0074】

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。

【0075】

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CLK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CLK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しまうことを防止できる。

【0076】

メイン制御部 4 1 が備える固有情報記憶回路 5 0 3 は、例えばメイン制御部 4 1 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 5 0 3 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。ROM 5 0 6 コードは、ROM 5 0 6 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 4 1 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 4 1 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 4 1 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 4 1 を構成するチップ毎に異なる数値を示している。ここで、チップ個別ナンバーはユーザプログラムから読み取ることができる一方、ID ナンバーはユーザプログラムから読み取ることができないように設定されていれば良い。尚、固有情報記憶回路 5 0 3 は、例えば ROM 5 0 6 の所定領域を用いることなどにより、ROM 5 0 6 に含まれるようにしても良い。或いは、固有情報記憶回路 5 0 3 は、例えば CPU 5 0 5 の内蔵レジスタを用いることなどにより、CPU 5 0 5 に含まれるようにしても良い。

【 0 0 7 7 】

メイン制御部 4 1 が備えるリセット / 割込コントローラ 5 0 4 は、メイン制御部 4 1 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 5 0 4 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 X S R S T に一定の期間にわたりローレベル信号が入力されたときに発生するリセットである。ユーザリセットは、ウォッチドッグタイマ (W D T) 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 (I A T) 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。

【 0 0 7 8 】

リセット / 割込コントローラ 5 0 4 が制御する割込には、ノンマスカブル割込 N M I とマスカブル割込 I N T が含まれている。ノンマスカブル割込 N M I は、CPU 5 0 5 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 X N M I (入力ポート P 4 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 I N T は、CPU 5 0 5 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 I N T の要因としては、外部マスカブル割込端子 X I N T (入力ポート P 3 と兼用) に一定の期間にわたりローレベル信号が入力が入力されたこと、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、シリアル通信回路 5 1 1 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 5 0 9 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。

【 0 0 7 9 】

リセット / 割込コントローラ 5 0 4 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、割込マスクレジスタ I M R (アドレス 2 0 2 8 H)、割込待ちモニタレジスタ I R R (アドレス 2 0 2 9 H)、割込中モニタレジスタ I S R (アドレス 2 0 2 A H)、内部情報レジスタ C I F (アドレス 2 0 8 C H) 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 I M R は、互いに異なる複数の要因によるマスカブル割込 I N T のうち、使用するものと使用しないものとを設定するレジスタである。割込待ちモニタレジスタ I R R は、割込初期設定 K I I S 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 I S R は、割込初期設定 K I I S 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 C I F は、直前に発生したりセット要因を管理したり、乱数用クロック R C L K の周波数異常を記録したりするためのレジスタである。

【 0 0 8 0 】

図 1 1 (A) は、内部情報レジスタ C I F の構成例を示している。図 1 1 (B) は、内

10

20

30

40

50

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れないときに、内部情報データCIF4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CIFのビット番号[2]に格納される内部情報データCIF2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CIF2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。

10

【0081】

内部情報レジスタCIFのビット番号[1]に格納される内部情報データCIF1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CIF1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CIFのビット番号[0]に格納される内部情報データCIF0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CIF0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。

20

【0082】

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。

30

【0083】

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。

40

【0084】

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

50

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。

【0085】

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。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。

【0086】

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チャンネル(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CLK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チャンネル(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CLK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。

【0087】

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。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。

【0088】

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。

【0089】

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)

10

20

30

40

50

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、乱数値レジスタ559Bは未使用とされている。

【0090】

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

【0091】

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20MHz）の1/2となる発振周波数（例えば10MHz）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

30

【0092】

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。

40

【0093】

50

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 CLK 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 CLK の発振周波数が 11.0 MHz である一方で、乱数用クロック RCLK の発振周波数は 9.7 MHz であれば良い。そのため、乱数更新クロック RGK やラッチ用クロック RC1、RC2 はいずれも、CPU505 に供給される制御用クロック CLK 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 552 は、乱数用クロック生成回路 43 によって生成された乱数用クロック RCLK 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 RGK や、複数の乱数取得用クロックとなるラッチ用クロック RC1、RC2 として、制御用クロック CLK や内部システムクロック SCLK (制御用クロック CLK 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。

10

【0094】

乱数生成回路 553 は、例えば 16 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 552 から出力される乱数更新クロック RGK 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 553 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 RGK における立ち上がりエッジにตอบสนองして、「0」から「65535」までの範囲内で設定された初期値から「65535」まで 1 ずつ加算するように数値データをカウントアップして行く。そして、「65535」までカウントアップした後は、「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。

20

【0095】

スタート値設定回路 554 は、第 2 乱数初期設定 KRS2 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 553 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 554 は、第 2 乱数初期設定 KRS2 のビット番号 [1-0] が “00” であればスタート値をデフォルト値である「0000H」に設定し、“10” であれば ID ナンバーに基づく値に設定し、“01” 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 KRS2 のビット番号 [1-0] が “10” に設定されており、乱数生成回路 553 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 ID ナンバーに基づく値が設定される。

30

【0096】

乱数列変更回路 555 は、乱数生成回路 553 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 555 は、乱数生成回路 553 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 555 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。

40

【0097】

乱数列変更設定回路 556 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 (図 9 (A) 参照) などに応じて、乱数列変更回路 555 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 556 は、第 1 乱数初期設定 KRS1 のビット番号 [1-0] が “00” 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、“01” 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、“10” 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 KRS1 のビット番号 [1-0] が “10” に設定されており、自動で乱数更新規則が変更される。

【0098】

50

乱数列変更回路556は、第1乱数初期設定KRS1のビット番号[1-0]が“01”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41が備える内蔵レジスタのうち、乱数列変更レジスタRDS C(アドレス2034H)を用いて、乱数更新規則の変更を制御する。図13(A)は、乱数列変更レジスタRDS Cの構成例を示している。図13(B)は、乱数列変更レジスタRDS 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RDS Cのビット番号[0]に格納される乱数列変更要求データRDS C0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RDS C0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。

10

【0099】

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RCN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RDS C0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RSNが、「0 1 ... 65535」となっている。この後、CPU505がROM506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RDS Cのビット番号[0]に“1”が書き込まれたものとする。

20

【0100】

そして、第1乱数初期設定KRS1のビット番号[1-0]が“01”であることに対応して、乱数列変更設定回路556が乱数列変更要求データRDS C0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RCNが所定の最終値に達したことに応じて、例えば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RCN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RDS C0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RSNとして、「65535 65534 ... 0」を出力する。乱数列変更レジスタRDS 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RDS C0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RDS 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RSNが、「65535 65534 ... 0」となる。

30

【0101】

CPU505がROM506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、乱数列変更レジスタRDS 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RSN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RDS C0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RSNとして、「0 2 ... 65534 1 ... 65535」を出力する。

40

【0102】

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RCN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RSNが、「0 1 ... 65535」となっている。

50

【 0 1 0 3 】

そして、乱数変更回路 5 5 5 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 5 5 6 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も良い。或いは、乱数列変更設定回路 5 5 6 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 1 5 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 5 5 5 から出力される乱数列 R S N が、「6 5 5 3 5 6 5 5 3 4 ... 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 5 5 5 から出力される乱数列 R S N は、「0 2 ... 6 5 5 3 4 1 ... 6 5 5 3 5」となる。図 1 5 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 5 5 5 から出力される乱数列 R S N は、「6 5 5 3 4 0 ... 3 2 7 6 8」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 5 5 5 から出力される乱数列 R S N が、「1 6 3 8 3 4 9 1 5 1 ... 4 9 1 5 0」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 5 5 5 から出力される乱数列 R S N が、「4 3 ... 4 6 5 5 3 1」となる。

10

【 0 1 0 4 】

ラッチ用フリップフロップ 5 5 7 A 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 5 5 7 A では、D 入力端子に P I P 5 1 0 が備える入力ポート P 0 からの配線が接続され、クロック端子 C K にラッチ用クロック R C 1 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 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 が入力される。ラッチ用フリップフロップ 5 5 7 A は、ラッチ用クロック R C 1 の立ち上がりエッジなどに応答して、ゲーム開始信号 S S 1 を取り込み、ゲーム開始時ラッチ信号 S L 1 として出力する。これにより、ラッチ用フリップフロップ 5 5 7 A では、ラッチ用クロック R C 1 の立ち上がりエッジに同期して、ゲーム開始信号 S S 1 がゲーム開始時ラッチ信号 S L 1 として出力される。

20

【 0 1 0 5 】

尚、ゲーム開始信号 S S 1 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 3 m s）になったときに、ゲーム開始信号 S S 1 を出力するタイマ回路を設けても良い。

30

【 0 1 0 6 】

乱数ラッチセクタ 5 5 8 A は、ラッチ用フリップフロップ 5 5 7 A から伝送されるゲーム開始時ラッチ信号 S L 1 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 L L 1 として選択的に出力する回路である。乱数ラッチセクタ 5 5 8 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、乱数値取込レジスタ R D L T (アドレス 2 0 3 2 H) を用いて、乱数ラッチ信号 L L 1 の出力を制御する。乱数値取込レジスタ R D L T 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、ソフトウェアにより乱数値レジスタ 5 5 9 A 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 R D L S 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、乱数値レジスタ 5 5 9 A に、ソフトウェアにより取り込むか、入力ポート P 0 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。

40

【 0 1 0 7 】

図 1 6 (A) は、乱数値取込レジスタ R D L T の構成例を示している。図 1 6 (B) は、乱数値取込レジスタ R D L T 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 R D L T のビット番号 [0] に格納される乱数値取込指定データ R D L T 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 に対する乱数値取込指定の有無を示している。図 1 6 (B) に示す例では、ソフトウェアにより乱数値レジスタ R 1 D 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 R D L T 0 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

50

、そのビット値が“ 1 ”となる。

【 0 1 0 8 】

図 1 7 (A) は、乱数ラッチ選択レジスタ R D L S の構成例を示している。図 1 7 (B) は、乱数ラッチ選択レジスタ R D L S に格納される乱数ラッチ選択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 1 7 (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 に取り込む場合には、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 0 9 】

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 1 8 (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 1 8 (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 1 8 (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 1 6

20

【 0 1 1 0 】

乱数値レジスタ 5 5 9 A は、乱数ラッチセクタ 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、CPU 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。

30

【 0 1 1 1 】

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。

40

【 0 1 1 2 】

図 1 9 (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 1 9 (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 1 9 (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

50

いないときに（乱数値取込なし）、乱数ラッチフラグデータRDFM0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（乱数値取込あり）、そのビット値が“1”となる。乱数ラッチフラグデータRDFM0が“1”の状態、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しており、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。

【0113】

図20(A)は、乱数割込制御レジスタRDICの構成例を示している。図20(B)は、乱数割込制御レジスタRDIC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RDICのビット番号[0]に格納される乱数割込制御データRDIC0は、乱数値レジスタR1Dとなる乱数値レジスタ559A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20(B)に示す例では、乱数値レジスタR1Dへの取込時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RDIC0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RDIC0のビット値が“0”に設定されており、乱数値レジスタR1Dとなる乱数値レジスタ559Aに数値データが取り込まれても割込は発生しない。

【0114】

メイン制御部41が備えるPIP510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P0～入力ポートP2と、機能兼用端子となる入力ポートP3～入力ポートP5とを含んでいる。入力ポートP3は、CPU505等に接続される外部マスカブル割込端子XINTと兼用される。入力ポートP4は、CPU505等に接続される外部ノンマスカブル割込端子XNMIと兼用される。入力ポートP5は、シリアル通信回路511が使用する第1チャンネル受信端子RXAと兼用される。入力ポートP3～入力ポートP5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KFC5により指示される。

【0115】

PIP510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、入力ポートレジスタPI（アドレス2090H）などを用いて、入力ポートP0～入力ポートP5の状態管理等を行う。入力ポートレジスタPIは、入力ポートP0～入力ポートP5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。

【0116】

図21(A)は、入力ポートレジスタPIの構成例を示している。図21(B)は、入力ポートレジスタPI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PIのビット番号[5]に格納される入力ポートデータPI5は、第1チャンネル受信端子RXAと兼用される入力ポートP5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PIのビット番号[4]に格納される入力ポートデータPI4は、外部ノンマスカブル割込端子XNMIと兼用される入力ポートP4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PIのビット番号[3]に格納される入力ポートデータPI3は、外部マスカブル割込端子XINTと兼用される入力ポートP3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PIのビット番号[2]に格納される入力ポートデータPI2は、入力ポートP2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PIのビット番号[1]に格納される入力ポートデータPI1は、入力ポートP1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PIのビット番号[0]に格納される入力ポートデータPI0は、入力ポートP0における端子状態（オン/オフ）を示している。

10

20

30

40

50

【 0 1 1 7 】

図5に示すメイン制御部41が備えるアドレスデコード回路512は、メイン制御部41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、CPU505からのアクセスが可能となる。

【 0 1 1 8 】

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、ROM506には、CPU505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、ROM506には、CPU505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

【 0 1 1 9 】

メイン制御部41が備えるRAM507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。RAM507としては、例えばDRAM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。CPU505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CPU505がROM506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、CPU505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

【 0 1 2 0 】

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。

【 0 1 2 1 】

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、外部マスカブル割込端子XINTが、前述した電断検出回路48と接続されており、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後述する電断割込処理(メイン)を実行する。また、メイン制御部41は、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、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

【 0 1 2 2 】

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。本実施例では、タイマ割込処理(メイン)よりも電断割込処理(メイン)の方が優先順位が高く、これらの割込が同時に発生した場合には電断割込処理(メイン)が優先して実行される。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。例えば、タイマ割込処理(メイン)の実行中に電圧低下信号が入力された場合に、その時点で電断割込処理(メイン)が実行されることはないが、実行中のタイマ割込処理(メイン)が終了した時点で、電圧低下信号の入力が継続している場合には、その時点で電断割込処理(メイン)が実行されることとなる。

40

【 0 1 2 3 】

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

50

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。

【0124】

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。

10

【0125】

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。

【0126】

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。

20

【0127】

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。

30

【0128】

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。

40

【0129】

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。また、サブ制御部91の割込端子の1つは、電断検出回路98と接続されており、サブ制御部91は、電断検出回路98か

50

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（サブ）を実行する。

【0130】

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。尚、電断割込処理（サブ）の実行中には、コマンド受信割込処理も禁止されるが、電断割込処理（サブ）の契機となる割込が同時に発生した場合にはコマンド受信割込処理を優先して実行する

【0131】

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。

【0132】

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るようになっている。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。

【0133】

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。

【0134】

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。

【0135】

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が電断検出回路48からの電圧低下信号を検出した際に、電断割込処理（メイン）を実行する。電断割込処理（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。

【0136】

そして、メイン制御部 4 1 は、その起動時において R A M 5 0 7 の全ての領域に格納されたデータに基づいて R A M 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、R A M 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、R A M 5 0 7 に記憶されているデータに基づいてメイン制御部 4 1 の処理状態を電断前の状態に復帰させるが、R A M 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、R A M 異常と判定し、R A M 異常エラーコードをレジスタにセットして R A M 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、R A M 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 2 3 やリセット / 設定スイッチ 3 8 を操作しても解除されないようになっており、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。

10

【 0 1 3 7 】

尚、本実施例では、R A M 5 0 7 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 4 1 は、電源投入時において R A M 5 0 7 のデータが正常であると判定した場合に、R A M 5 0 7 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、R A M 5 0 7 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。

【 0 1 3 8 】

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。

20

【 0 1 3 9 】

また、サブ制御部 9 1 も電断検出回路 9 8 からの電圧低下信号を検出した際に、電断割込処理（サブ）を実行する。電断割込処理（サブ）では、レジスタを後述する R A M 9 1 c のスタックに退避し、R A M 9 1 c 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、R A M 9 1 c の全ての領域に格納されたデータに基づく R A M パリティが 0 となるように R A M パリティ調整用データを計算し、R A M 9 1 c に格納する処理を行うようになっている。

30

【 0 1 4 0 】

そして、サブ制御部 9 1 は、その起動時において R A M 9 1 c の全ての領域に格納されたデータに基づいて R A M パリティを計算し、R A M パリティが 0 であることを条件に、R A M 9 1 c に記憶されているデータに基づいてサブ制御部 9 1 の処理状態を電断前の状態に復帰させるが、R A M パリティが 0 でない場合（1 の場合）には、R A M 異常と判定し、R A M 9 1 c を初期化するようになっている。この場合、メイン制御部 4 1 と異なり、R A M 9 1 c 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。

【 0 1 4 1 】

尚、本実施例では、R A M 9 1 c 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 9 1 は、電源投入時において R A M 9 1 c のデータが正常であると判定した場合に、R A M 9 1 c 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、R A M 9 1 c 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。

40

【 0 1 4 2 】

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。

50

【 0 1 4 3 】

次に、メイン制御部 4 1 の R A M 5 0 7 の初期化について説明する。メイン制御部 4 1 の R A M 5 0 7 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。

【 0 1 4 4 】

重要ワークは、各種表示器や L E D 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、B B 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、B B 中のメダル払出総数等、B B 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 9 0 ヘ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 R A M 5 0 7 のデータが破壊されてい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、R A M 5 0 7 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 4 1 のレジスタから回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。

【 0 1 4 5 】

本実施例においてメイン制御部 4 1 は、設定キースイッチ 3 7 が O N の状態での起動時、R A M 異常エラー発生時、B B 終了時、設定キースイッチ 3 7 が O F F の状態での起動時において R A M 5 0 7 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。

【 0 1 4 6 】

初期化 1 は、起動時において設定キースイッチ 3 7 が O N 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 R A M 異常エラー発生時に行う初期化であり、初期化 1 では、R A M 5 0 7 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、B B 終了時に行う初期化であり、初期化 2 では、R A M 5 0 7 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 3 7 が O F F の状態であり、かつ R A M 5 0 7 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、R A M 5 0 7 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。

【 0 1 4 7 】

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ぬ。

【 0 1 4 8 】

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 R B と称す）（ビッグボーナス（以下では B B と称す））、通常遊技状態があり、このうち R B （B B ）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 R B （B B ）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

10

20

30

40

50

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。

【0149】

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。

10

【0150】

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つた場合にも、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。

20

【0151】

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。

【0152】

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。

30

【0153】

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。

40

【0154】

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。

50

【 0 1 5 5 】

R B 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 R B に移行する。R B 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、R B 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。

【 0 1 5 6 】

B B 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。

10

【 0 1 5 7 】

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

【 0 1 5 8 】

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、全てのルール 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、RAM 507 に割り当てられた乱数値格納ワークの値を同じく RAM 507 に割り当てられた抽選用ワークに設定する。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、抽選用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。

20

【 0 1 5 9 】

乱数値格納ワークは、スタートスイッチ 7 の操作と同時に乱数値レジスタ R 1 D にラッチされた数値データが格納される記憶領域であり、乱数値レジスタ R 1 D に新たな数値データがラッチされる毎に、ラッチされた数値データがその後のタイマ割込処理（メイン）において読み出され、乱数値格納ワークに格納された数値データが新たにラッチされた最新の数値データに更新されるようになっている。

30

【 0 1 6 0 】

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、特別役の持越中であるか否かによっても内部抽選の対象となる役が異なる。

【 0 1 6 1 】

遊技状態が通常遊技状態であり、いずれの特別役も持ち越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。

40

【 0 1 6 2 】

遊技状態が通常遊技状態であり、いずれかの特別役が持ち越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。

【 0 1 6 3 】

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。

【 0 1 6 4 】

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（抽選用ワークの値）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

50

大小に応じた確率（判定値数 / 6 5 5 3 6）で役が当選することとなる。

【 0 1 6 5 】

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 R A M 5 0 7 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。

10

【 0 1 6 6 】

次に、リール 2 L、2 C、2 R の停止制御について説明する。

【 0 1 6 7 】

メイン制御部 4 1 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、R O M 5 0 6 A 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。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。

20

【 0 1 6 8 】

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。

30

【 0 1 6 9 】

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。

【 0 1 7 0 】

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。

40

【 0 1 7 1 】

50

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32L、32C、32R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32L、32C、32Rを168ステップ駆動させることでリール2L、2C、2R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ていることとなる。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。

10

【0172】

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8L、8C、8R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ある。

【0173】

20

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。

【0174】

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。

30

【0175】

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。

40

50

【0176】

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。

10

【0177】

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できることとなる。

20

【0178】

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て停止させる制御が行われることとなる。

30

【0179】

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て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。

40

【0180】

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

50

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。

【0181】

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2L、2C、2R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8L、8C、8R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。

10

【0182】

本実施例においてメイン制御部41は、リール2L、2C、2Rの回転が開始した後、ストップスイッチ8L、8C、8R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8L、8C、8R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8L、8C、8R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8L、8C、8R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。

20

【0183】

尚、本実施例では、ストップスイッチ8L、8C、8R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に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。

30

【0184】

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ついて説明する。

【0185】

本実施例では、メイン制御部41が演出制御基板90に対して、BET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。

40

【0186】

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91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。

【0187】

BET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BET

50

スイッチ5またはMAX BET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、BET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、BET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。

【0188】

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。

【0189】

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。

10

【0190】

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2L、2C、2Rの回転が開始されたときに送信される。

【0191】

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8L、8C、8R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8L、8C、8Rが操作されたことを特定可能である。

20

【0192】

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。

【0193】

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。

30

【0194】

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。

【0195】

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。

40

【0196】

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、BB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。

【0197】

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。

50

【 0 1 9 8 】

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。

【 0 1 9 9 】

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。

【 0 2 0 0 】

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。

10

【 0 2 0 1 】

操作検出コマンドは、操作スイッチ類（1枚BETスイッチ5、MAXBETスイッチ6、スタートスイッチ7、ストップスイッチ8L、8C、8R）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。

【 0 2 0 2 】

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、RAM 5 0 7 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。

20

【 0 2 0 3 】

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、RAM 5 0 7 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。

【 0 2 0 4 】

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。

30

【 0 2 0 5 】

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基づいてサブ制御部 9 1 が実行する演出の制御について説明する。

【 0 2 0 6 】

サブ制御部 9 1 は、メイン制御部 4 1 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、RAM 9 1 c 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。

40

【 0 2 0 7 】

サブ制御部 9 1 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ROM 9 1 b 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 5 1、演出効果LED 5 2、スピーカ 5 3、5 4、リールLED 5 5 等の各種演出装置の出力制御を行う。

【 0 2 0 8 】

50

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 5 1 の表示パターン、演出効果 L E D 5 2 の点灯態様、スピーカ 5 3、5 4 の出力態様、リール L E D 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 9 1 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 R A M 9 1 c 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。

【 0 2 0 9 】

尚、サブ制御部 9 1 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。

【 0 2 1 0 】

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 9 1 が、賭数が設定された旨を示す B E T コマンドを受信したときに、実行中の演出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 9 1 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。

【 0 2 1 1 】

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、R A M 9 1 c に設定される。演出パターンの選択率は、R O M 9 1 b 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 9 1 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 R A M 9 1 c に設定するようになっている。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。

【 0 2 1 2 】

次に、本実施例におけるメイン制御部 4 1 が実行する各種制御内容を、図 2 3 ~ 図 3 7 に基づいて以下に説明する。

【 0 2 1 3 】

メイン制御部 4 1 は、リセット要因が発生すると、図 2 3 のフローチャートに示す起動処理（メイン）を行う。尚、リセット要因としては、電源投入時において外部システムリセット端子 X S R S T 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システムリセットや、ウォッチドッグタイマ（W D T）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（I A T）が発生したことなどのユーザリセットがある。このため、起動処理（メイン）は、電源投入に伴うメイン制御部 4 1 の起動時及びメイン制御部 4 1 の不具合に伴う再起動時に行われる処理である。

【 0 2 1 4 】

起動処理（メイン）では、まず、C P U 5 0 5 が R O M 5 0 6 から読み出したセキュリ

10

20

30

40

50

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。

【0215】

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KSES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、この読出値が“000”であるか否かを判定する(Sa102)。

10

【0216】

Sa102にて読出値が“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変時間成分であり、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。

【0217】

Sa102にて読出値が“000”以外と判定された場合には、その読出値に対応して、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。

20

【0218】

Sa103、Sa104の処理のいずれかを実行した後は、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、この読出値が“00”であるか否かを判定する(Sa106)。

【0219】

Sa106にて読出値が“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、読出値が“00”以外と判定された場合には、その読出値に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。

30

40

【0220】

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]

50

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。

【0221】

Sa107、Sa108の処理のいずれかを実行した後は、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。

10

【0222】

ステップS9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(Sa1010)。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、所定のテーブ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)。

20

【0223】

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、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(Sa114)。このとき、セキュリティ時間が経過していなければ、Sa114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、Sa114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。

30

【0224】

メイン制御部41は、Sa1のセキュリティチェック処理の終了後、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)、入力ポートから電圧低下信号の検出データを取得し、電圧低下信号が入力されているか否か、すなわち電圧が安定しているか否かを判定し(Sa3)、電圧低下信号が入力されている場合には、電圧低下信号が入力されているか否かの判定以外は、いずれの処理も行わないループ処理に移行する。

40

【0225】

Sa3のステップにおいて電圧低下信号が入力されていないと判定した場合には、レジスタ及びIYレジスタの値を初期化する(Sa4)。レジスタ及びIYレジスタの初期化により、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。

【0226】

50

次いで、RAM 507へのアクセスを許可し(S a 5)、RAM 507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(S a 6)、打止スイッチ36、自動精算スイッチ29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/無効を設定する(S a 7)。

【0227】

次いで、S a 6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(S a 8)。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、S a 8のステップにおいてRAMパリティが0でなければ、RAM 507に格納されているデータが正常ではなく、この場合には、RAM 507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(S a 31)、設定キースイッチ37がONか否かを判定し(S a 32)、設定キースイッチ37がONであれば、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(S a 26)、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(S a 27)。

10

【0228】

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(S a 201)。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(S a 202)。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CLK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

【0229】

S a 202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(S a 203)。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

【0230】

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(S a 204)。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。

【0231】

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(S a 205)。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、S a 204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。S a 205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、S a 205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、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、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

40

50

乱数生成回路 5 5 3 にて生成されるカウント値順列 R C N に含まれる数値データの全部又は一部を含む範囲) でランダムに変化するように決定されれば良い。

【 0 2 3 2 】

尚、乱数回路設定処理による設定は、C P U 5 0 5 の処理が介在することなく、乱数回路 5 0 9 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 5 0 9 は、メイン制御部 4 1 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しても良い。そして、メイン制御部 4 1 にて C P U 5 0 5 が R O M 5 0 6 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 C P U 5 0 5 から乱数回路 5 0 9 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 5 0 9 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 5 0 9 がメイン制御部 4 1 に外付けされる場合などには、メイン制御部 4 1 がセキュリティモードとなっているときでも、乱数回路 5 0 9 が C P U 5 0 5 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。

【 0 2 3 3 】

図 2 6 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タ値を「 0 」に初期化する (S a 3 0 1) 。続いて、内部情報レジスタ C I F のビット番号 [4] に格納されている内部情報データ C I F 4 を読み出す (S a 3 0 2) 。そして、 S a 3 0 2 での読出値が “ 1 ” であるか否かを判定する。このとき、乱数回路 5 0 9 が備える周波数監視回路 5 5 1 によって、乱数用外部クロック端子 E R C からの乱数用クロック R C L K に周波数異常が検知された場合には、内部情報データ C I F 4 としてビット値 “ 1 ” が書き込まれる。

【 0 2 3 4 】

そこで、 S a 3 0 3 にて読出値が “ 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タ値を 1 加算するように更新する (S a 3 0 4) 。このときには、 S a 3 0 4 での更新後におけるカウンタ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 (S a 3 0 5) 。ここで、クロック異常判定値は、周波数監視回路 5 5 1 により乱数用クロック R C L K 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。 S a 3 0 5 にてクロック異常判定値に達していなければ、 S a 3 0 2 の処理に戻り、再び内部情報データ C I F 4 のビット値に基づく判定を行う。

【 0 2 3 5 】

S a 3 0 5 にてクロック異常判定値に達した場合には、乱数回路異常を示すエラーコードをレジスタに設定し (S a 3 0 6) 、乱数回路異常検査処理を終了する。

【 0 2 3 6 】

S a 3 0 3 にて読出値が “ 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タ値を「 0 」に初期化する (S a 3 0 7) 。尚、 S a 3 0 3 の処理では、 S a 3 0 2 にて読み出した内部情報データ C I F 4 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、読出値が “ 0 ” であると判定しても良い。

【 0 2 3 7 】

S a 3 0 7 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「 0 0 0 0 H 」に設定する (S a 3 0 8) 。そして、乱数回路 5 0 9 が備える乱数値レジスタ R 1 D となる乱数値レジスタ 5 5 9 A 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 (S a 3 0 9) 。例えば、 S a 3 0 9 の処理では、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 “ 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 R D L T のビット番号 [0] に格納される乱数値取込指定データ R D L T 0 のビット値を “ 1 ” として、乱数値レジスタ R 1 D への取り込みを指定する。なお、乱数値取込指定レジスタ R D L T のビット番号 [0] におけるビット値を “ 1

” とすることは、CPU 505 から乱数回路 509 に対して数値データの取り込み（ラッチ）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 R1D に供給するレジスタリード信号 RRS1 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。

【0238】

Sa309 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（Sa310）。続けて、さらに乱数値レジスタ 559A から乱数値となる数値データを読み出す（Sa311）。尚、Sa311 での読出動作は、Sa309 での読出動作と同様の手順で行われれば良い。また、Sa309 での読出動作と、Sa311 での読出動作との間には、乱数回路 509 で生成される乱数列 RSN 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。Sa311 にて数値データを読み出した後には、乱数検査基準値と、Sa311 での読出値との排他的論理和演算を実行する（Sa312）。また、Sa312 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（Sa313）。例えば、チェック値は RAM 507 の所定領域に記憶しておき、Sa313 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 559A 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。

【0239】

そこで、チェック値が「FFFFH」となったか否かを判定し（Sa314）、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 RDL5 のビット番号 [0] に格納される乱数ラッチ選択データ RDL50 のビット値を“1”として、入力ポート P0 への信号入力に応じた乱数値レジスタ R1D 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 P0 にスタートスイッチ 7 からのゲーム開始信号 SS1 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 SS1 がオン状態となったときに乱数値レジスタ R1D への乱数値取込を行うことができる。

【0240】

Sa314 にてチェック値が「FFFFH」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（Sa315）。このときには、Sa315 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（Sa316）。ここで、乱数値異常判定値は、乱数回路 509 が正常動作していれば、乱数値レジスタ 559A 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。Sa316 にて乱数値異常判定値に達していなければ、Sa311 の処理に戻り、再び乱数回路 509 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。

【0241】

Sa316 にて乱数値異常判定値に達した場合には、乱数回路異常を示すエラーコードをレジスタに設定し（Sa317）、乱数回路異常検査処理を終了する。

【0242】

尚、Sa314 の乱数回路異常検査処理は、CPU 505 が実行するものに限定されず、CPU 505 以外のメイン制御部 41 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 509 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 RCLK 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、エラーの発生を CPU 505 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、Sa314 のみにて実行されるものに限定されず、例えばメイン制御部 41 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。

【 0 2 4 3 】

S a 2 6 のステップにおける乱数回路異常検査処理の後、乱数回路異常か否かを判定する (S a 2 7) 。乱数回路異常か否かは、乱数回路異常を示すエラーコードが設定されているか否かによって判定する。S a 2 7 のステップにおいて乱数異常と判定された場合には、乱数回路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 9 1 に対して送信し (S a 2 2) 、割込を許可して (S a 2 3) 、エラー処理、すなわち R A M 異常エラー状態に移行する。

【 0 2 4 4 】

S a 2 7 のステップにおいて乱数異常ではないと判定された場合には、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 9 1 に対して送信する (S a 2 8) 。次いで、割込を許可して (S a 2 9) 、設定変更処理、すなわち設定変更状態に移行する。

10

【 0 2 4 5 】

S a 3 1 のステップにおいて設定キースイッチ 3 7 が O F F であれば、R A M 異常を示すエラーコードをレジスタに設定し (S a 3 2) 、R A M 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 9 1 に対して送信し (S a 3 3) 、割込を許可して (S a 3 4) 、エラー処理、すなわち R A M 異常エラー状態に移行する。

【 0 2 4 6 】

S a 7 のステップにおいて、R A M 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 (S a 8) 。正常に電断割込処理 (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、S a 8 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 5 A (H) 以外の場合) にも、R A M 5 0 7 のデータが正常ではないので、S a 2 4 のステップに移行して初期化 1 を実行し、その後、S a 3 1 のステップにおいて設定キースイッチ 3 7 が O N であれば、乱数回路設定処理 (S a 2 5) 、乱数回路異常検査処理 (S a 2 6) を順次行い、乱数回路異常であれば、乱数回路異常を示すエラーコマンドの送信 (S a 2 2) の後、割込を許可して (S a 2 3) 、エラー処理に移行する。一方、乱数回路異常でなければ、設定開始コマンドの送信 (S a 2 8) の後、割込を許可して (S a 2 9) 、設定変更処理に移行する。また、S a 3 1 のステップにおいて設定キースイッチ 3 7 が O F F であれば、R A M 異常を示すエラーコードの設定 (S a 3 2) 、R A M 異常を示すエラーコマンドの送信 (S a 3 2 3) の後、割込を許可して (S a 3 4) 、エラー処理に移行する。

20

30

【 0 2 4 7 】

S a 8 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 9) 、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 0) 、設定キースイッチ 3 7 が O N か否かを判定し (S a 1 1) 、設定キースイッチ 3 7 が O N であれば、S a 2 4 のステップに移行して初期化 1 を実行し、乱数回路設定処理 (S a 2 5) 、乱数回路異常検査処理 (S a 2 6) を順次行い、乱数回路異常であれば、乱数回路異常を示すエラーコマンドの送信 (S a 2 2) の後、割込を許可して (S a 2 3) 、エラー処理に移行する。一方、乱数回路異常でなければ、設定開始コマンドの送信 (S a 2 8) の後、割込を許可して (S a 2 9) 、設定変更処理に移行する。

40

【 0 2 4 8 】

S a 1 1 のステップにおいて設定キースイッチ 3 7 が O F F であれば、S a 2 5 のステップと同じ乱数回路設定処理 (S a 1 3) 、S a 2 6 のステップと同じ乱数回路異常検査処理 (S a 1 4) を実行し、乱数回路異常か否かを判定する (S a 1 5) 。

【 0 2 4 9 】

S a 1 5 のステップにおいて乱数回路異常であると判定された場合には、乱数回路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 9 1 に対して送信し (S a 2 2) 、割込を許可して (S a 2 3) 、エラー処理、すなわち R A M 異常エラー状態に移行する。

50

【 0 2 5 0 】

S a 1 5 のステップにおいて乱数回路異常ではないと判定された場合には、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 6)、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 7)。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 8)、乱数値がラッチされていなければ S a 2 0 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 から格納されている数値データを読み出し (S a 1 9)、S a 2 0 のステップに進む。尚、S a 1 9 のステップにおいて乱数値レジスタ 5 5 9 から数値データ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 9 のステップにおいては、数値データを読み出すものの、読み出した数値データ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数値データ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。

10

【 0 2 5 1 】

S a 2 0 のステップでは、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 に対して送信し、その後、割込を許可して (S a 2 1)、電断前の最後に行っていた処理に戻る。

【 0 2 5 2 】

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ある。

20

【 0 2 5 3 】

エラー処理では、まず、現在の遊技補助表示器 1 2 の表示状態をスタックに退避し (S b 1)、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 1 2 に表示し (S b 2)、エラーコードが R A M 異常エラーまたは乱数回路異常を示すエラーコードであるか否かを判定し (S b 3)、R A M 異常エラーまたは乱数回路異常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。

【 0 2 5 4 】

S b 3 のステップにおいて R A M 異常エラーを示すエラーコードでも乱数回路異常を示すエラーコードでもない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 (S b 4)。S b 4 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 9 1 に対して送信される。

30

【 0 2 5 5 】

次いで、リセット / 設定スイッチ 3 8 の操作が検出されているか否かを判定し (S b 5)、リセット / 設定スイッチ 3 8 の操作が検出されていなければ、更にリセットスイッチ 2 3 の操作が検出されているか否かを判定し (S b 6)、リセットスイッチ 2 3 の操作も検出されていなければ、S b 5 のステップに戻る。すなわちリセット / 設定スイッチ 3 8 またはリセットスイッチ 2 3 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。

40

【 0 2 5 6 】

そして、S b 5 のステップにおいてリセット / 設定スイッチ 3 8 の操作が検出された場合、または S b 6 のステップにおいてリセットスイッチ 2 3 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 (S b 7)、遊技補助表示器 1 2 の表示状態を S b 1 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 (S b 8)、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用送信バッファに設定して (S b 9)、もとの処理に戻る。S b 9 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 9 1 に対して送信される。

【 0 2 5 7 】

このようにエラー処理においては、R A M 異常、乱数回路異常以外によるエラー処理で

50

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異常や乱数回路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。

【0258】

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。

【0259】

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(S c 1)、表示値が設定可能な範囲(1~6)外か否かを判定し(S c 2)、表示値が設定可能な範囲内であればS c 4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、S c 5のステップに進む。

10

【0260】

S c 5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(S c 5、S c 6)、S c 5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(S c 7)、表示値を1加算し(S c 8)、S c 2のステップに戻る。

【0261】

また、S c 6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(S c 9)、設定値表示器24に表示されている値を0に更新し(S c 10)、設定キースイッチ37がOFFの状態となるまで待機する(S c 11)。

20

【0262】

S c 11のステップにおいて設定キースイッチ37のOFFが判定されると、表示値を設定値ワークに格納して(S c 12)、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 13)、ゲーム処理に移行する。S c 13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91に対して送信される。

【0263】

図29は、メイン制御部41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。

30

【0264】

ゲーム処理では、BET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BET処理に戻る。

【0265】

S d 1のステップにおけるBET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。

【0266】

40

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チされた数値データ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、RAM507に当選フラグが設定される。

【0267】

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2L、2C、2Rを回転させる処理、遊技者によるストップスイッチ8L、8C、8Rの操作が検出されたことに応じて対応するリール2L、2C、2Rの回転を停止させる処理を実行する。

【0268】

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

50

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。

【0269】

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。

【0270】

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。

【0271】

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91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。

【0272】

図30~図32は、メイン制御部41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。

【0273】

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)。

【0274】

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用送信バッファに設定し(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91に対して送信される。

【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。

【0276】

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。

【0277】

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

10

20

30

40

50

、かつ全てのスイッチがOFF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。

【 0 2 7 8 】

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数値データ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、数値データがラッチされていないならば S e 1 4 のステップに進み、数値データがラッチされていれば乱数値レジスタ 5 5 9 A から数値データを読み出し (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 から数値データ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、数値データを
10
読み出すものの、読み出した数値データ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数値データ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数値データがラッチされ、新たな数値データ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OFF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数値データ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数値データをラッチすることが可能となる。

【 0 2 7 9 】

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ていれば、図 2 7 に示すエラー処理に移行する。

【 0 2 8 0 】

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いならば、 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
30
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。

【 0 2 8 1 】

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。
40

【 0 2 8 2 】

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いならば、 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ていれば、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。
50

【0283】

Se 24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、RAM 507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し (Se 20)、BET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BETコマンドを通常コマンド送信用バッファに設定し (Se 26)、BETカウンタの値を1加算し (Se 27)、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 28)、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 26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。

【0284】

また、Se 28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数値データがラッチされていれば、Sa 13のステップにて数値データが読み出され、新たな数値データ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数値データがラッチされ、新たな数値データ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数値データ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数値データをラッチすることが可能となる。

【0285】

Se 25のステップにおいてBETカウンタの値が規定数であれば、現在のクレジット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 (Se 29)、RAM 507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 (Se 30)、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 (Se 31)、クレジットカウンタの値が50でなければ、Se 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RAM 507に設定し (Se 32)、Se 9のステップに戻る。Se 29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。

【0286】

Se 33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である場合のみ設定されるので、Se 33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。

【0287】

Se 33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、Se 39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ていれば、立上りエッジをクリアし (Se 34)、RAM 507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、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 (Se 35)。

【0288】

Se 35のステップにおいてBETカウンタの値が規定数でなければ、Se 9のステップに戻り、BETカウンタの値が規定数であれば、乱数値格納ワークの値を内部抽選用の乱数値として抽選用ワークに設定し (Se 36)、投入不可フラグをRAM 507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 (Se 37)、ゲーム開始時の設定を行う (Se 38)。そして、Se 38のステップの後、BET処理を終了して図29のフロ

10

20

30

40

50

ーチャートに復帰する。

【0289】

Se 39のステップにおいては、1枚BET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BET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。Se 39のステップにおいて1枚BETスイッチ5の操作が検出されていなければ、Se 47のステップに進み、1枚BETスイッチ5の操作が検出されていれば立上りエッジをクリアし(Se 40)、RAM 507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 41)。Se 41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 42)、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 42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 43)、クレジットカウンタの値を1減算し(Se 44)、BETカウンタの値を1加算して(Se 45)、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 46)、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 43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

【0290】

また、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、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数値データがラッチされていれば、Sa 13のステップにて数値データが読み出され、新たな数値データ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数値データがラッチされ、新たな数値データ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数値データ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数値データをラッチすることが可能となる。

20

【0291】

Se 47のステップにおいては、MAX BETスイッチ6の操作が検出されているか否か、すなわちMAX BET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 47のステップにおいてMAX BET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、MAX BETスイッチ6の操作が検出されていれば、立上りエッジをクリアし(Se 48)、RAM 507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 49)。Se 49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 53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 50)、クレジットカウンタの値が0であれば、Se 53のステップに進む。Se 50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 51)、BETカウンタの値を1加算して(Se 52)、Se 49のステップに戻る。Se 53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ていなければ、Se 9のステップに戻り、BETカウンタが加算されていれば、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 54)、Se 55のステップに進む。Se 54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

40

【0292】

Se 55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

50

れば S e 9 のステップに戻る。また、S e 5 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、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数値データがラッチされていれば、S a 1 3 のステップにて数値データが読み出され、新たな数値データ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数値データがラッチされ、新たな数値データ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数値データ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数値データをラッチすることが可能となる。

10

【 0 2 9 3 】

S e 5 6 のステップにおいては、精算スイッチ 1 0 の操作が検出されているか否か、すなわち精算スイッチ 1 0 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 5 6 のステップにおいて精算スイッチ 1 0 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 1 0 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 5 7)、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 5 8)、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 5 8 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 5 9)、B E T カウンタの値が 0 であれば S e 6 1 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 0)、S e 6 1 のステップに進む。S e 6 1 のステップにおいては、ホッパーモータ 3 4 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 1 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 2)、S e 9 のステップに戻る。

20

【 0 2 9 4 】

図 3 3 及び図 3 4 は、メイン制御部 4 1 が一定間隔 (0 . 5 6 m s 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。

30

【 0 2 9 5 】

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)。

【 0 2 9 6 】

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。

40

【 0 2 9 7 】

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 3 2 L、3 2 C、3 2 R 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 3 2 L、3 2 C、3 2 R 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 2 1 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)。

50

【 0 2 9 8 】

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3 2 L、3 2 C、3 2 R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3 2 L、3 2 C、3 2 R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3 2 L、3 2 C、3 2 Rの停止後、一定時間経過後に位相信号を1相励磁に変更するモータ位相信号スタンバイ処理（S k 9）を順次実行した後、S k 1においてスタック領域に退避したレジスタを復帰し（S k 1 9）、割込前の処理に戻る。

【 0 2 9 9 】

また、S k 6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（S k 1 0）、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（S k 1 1）、各種時間カウンタを更新する時間カウンタ更新処理（S k 1 2）、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9 1に対して送信するコマンド送信処理（S k 1 3）、外部出力信号を更新する外部出力信号更新処理（S k 1 4）を順次実行した後、S k 1においてスタック領域に退避したレジスタを復帰し（S k 1 9）、割込前の処理に戻る。

【 0 3 0 0 】

また、S k 4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（S k 1 5）、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2 L、2 C、2 R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（S k 1 6）、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（S k 1 7）、乱数値レジスタR 1 Dから数値データを読み出して乱数値格納ワークに格納する乱数値読出処理（S k 1 8）を順次実行した後、S k 1においてスタック領域に退避したレジスタを復帰し（S k 1 9）、割込前の処理に戻る。

【 0 3 0 1 】

また、S k 1 5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8 L、8 C、8 R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（S k 2 0）、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（S k 2 1）、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（S k 2 2）を順次実行した後、S k 1においてスタック領域に退避したレジスタを復帰し（S k 1 9）、割込前の処理に戻る。

【 0 3 0 2 】

図3 5は、メイン制御部4 1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乱数値読出処理の制御内容を示すフローチャートである。

【 0 3 0 3 】

乱数値読出処理では、乱数ラッチフラグレジスタの値に基づいて数値データ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 k 1 0 1）、数値データがラッチされていなければ、乱数値読出処理を終了して図3 4に示すフローチャートに復帰する。

【 0 3 0 4 】

S k 1 0 1のステップにおいて数値データがラッチされていれば乱数値レジスタ5 5 9 Aから数値データを読み出し（S k 1 0 2）、乱数値格納ワークに格納されている値を、

10

20

30

40

50

S k 1 0 2 において読み出した数値データに更新し (S k 1 0 3)、乱数値読出処理を終了して図 3 4 に示すフローチャートに復帰する。

【 0 3 0 5 】

尚、S k 1 0 2 のステップにおいて乱数値レジスタ 5 5 9 から数値データ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、乱数値レジスタ 5 5 9 への新たな数値データの取込が許可されることとなる。

【 0 3 0 6 】

図 3 6 は、メイン制御部 4 1 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。

【 0 3 0 7 】

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 2 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が同じであるか否かを判定し (S k 2 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が同じでなければ、図 3 4 のフローチャートに復帰する。

【 0 3 0 8 】

S k 2 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が同じ場合、すなわち 2 . 2 4 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 2 0 3)、S k 2 0 4 のステップに進む。S k 2 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。

【 0 3 0 9 】

S k 2 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が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が同じであれば、図 3 4 のフローチャートに復帰する。

【 0 3 1 0 】

S k 2 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が同じでなければ、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 O F F から O N に変化したかを判定し (S k 2 0 5)、O F F から O N に変化した場合には、さらに他のスイッチが O N か否かを判定し (S k 2 0 6)、他のスイッチが O N であれば S k 2 0 8 のステップに進み、他のスイッチが O N でなければ、該当するスイッチが O F F から O N に変化した旨を示す立上りエッジを設定し (S k 2 0 7)、S k 2 0 8 のステップに進む。

【 0 3 1 1 】

S k 2 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 O N から O F F に変化したかを判定し、O N から O F F に変化した場合には、他のスイッチが O N か否かを判定し (S k 2 0 9)、他のスイッチが O N であれば図 3 4 のフローチャートに復帰し、他のスイッチが O N でなければ、該当するスイッチが O N から O F F に変化した旨を示す立下りエッジを設定し (S k 2 1 0)、図 3 4 のフローチャートに復帰する。

【 0 3 1 2 】

図 3 7 は、電断検出回路 4 8 からの電圧低下信号が入力されたときにメイン制御部 4 1 が起動処理やゲーム処理に割り込んで実行する電断割込処理 (メイン) の制御内容を示すフローチャートである。尚、電断割込処理 (メイン) の開始後、自動的に他の割込が禁止され、元の処理に復帰する場合以外は割込が許可されることはない。

【 0 3 1 3 】

電断割込処理 (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 I Y 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。

10

20

30

40

50

【0314】

次いで、入力ポートから電圧低下信号の検出データを取得し、電圧低下信号が入力されているか否かを判定する(Sm2)。この際、電圧低下信号が入力されていなければ、Sm1においてスタック領域に退避したレジスタを復帰し(Sm3)、割込前の処理に戻る。この際、自動的に割込が許可される。

【0315】

また、Sm2のステップにおいて電圧低下信号が入力されていれば、破壊診断用データ(本実施例では、5A(H))をセットして(Sm4)、全ての出力ポートを初期化する(Sm5)。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(Sm6)、RAM507へのアクセスを禁止する(Sm7)。

10

【0316】

そして、電圧低下信号が入力されているか否かの判定(Sm8、尚、Sm8は、Sm2と同様の処理である)を除いて、何らの処理も行わないループ処理に入る。すなわち、そ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、このループ処理において、電圧が回復し、電圧低下信号が入力されない状態となると、前述した起動処理(メイン)が実行され、RAMパリティが0となり、かつ破壊診断用データが正常であれば、元の処理に復帰することとなる。

【0317】

尚、本実施例では、RAM507へのアクセスを禁止した後、電圧低下信号の出力状況を監視して、電圧低下信号が入力されなくなった場合に電圧の回復を判定し、起動処理(メイン)へ移行するようになっているが、ループ処理において何らの処理も行わず、ループ処理が行われている間に、電圧が回復してリセット回路49からリセット信号が入力されたことや、電圧が低下しないまま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に基づいて、起動処理(メイン)へ移行するようにしても良い。

20

【0318】

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、電源投入時や、システムリセットの発生後における再起動時などに、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。

30

【0319】

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。

【0320】

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。

40

【0321】

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、電源投入やシステムリセット等による動作開

50

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。

【0322】

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

10

【0323】

図38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38(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CLKを示している。図38(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CLKを示している。図38(A)及び(B)に示すように、制御用クロックCLKの発振周波数と、乱数用クロックCLK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、

20

【0324】

図38(B)に示すように、乱数用クロックCLK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CLK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CLKの立ち上がりエッジにตอบสนอง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38(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CLK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CLK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CLKの発信周波数は制御用クロックCLK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CLK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ตอบสนอง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38(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ตอบสนองして更新される。

30

40

【0325】

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38(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

50

となる。ラッチ用クロック R C 1 は、ラッチ用フリップフロップ 5 5 7 A のクロック端子 C K に入力される。ラッチ用クロック R C 2 は、ラッチ用フリップフロップ 5 5 7 B のクロック端子 C K に入力される。

【 0 3 2 6 】

ラッチ用フリップフロップ 5 5 7 A は、ラッチ用クロック R C 1 の立ち上がりエッジに
10 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 P 0 に供給されたゲーム開始信号 S S 1 を取り込み（ラッチして）、ゲーム開始時ラッチ信号 S L 1 として出力端子 Q から出力する。そして、乱数ラッチセクタ 5 5 8 A における取込方法が入力ポート P 0 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 S L 1 が乱数ラッチ信号 L L 1 として出力される。これにより、例えば図 3 8 (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 S S 1 は、ラッチ用クロック R C 1 が立ち上がるタイミング T 1 1、T 1 3、T 1 5、... にてラッチ用フリップフロップ 5 5 7 A に取り込まれた後、図 3 8 (G) に示すようなタイミング T 1 1、T 1 3 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 L L 1 となって、乱数ラッチセクタ 5 5 8 A から出力される。

【 0 3 2 7 】

乱数値レジスタ R 1 D となる乱数値レジスタ 5 5 9 A は、乱数列変更回路 5 5 5 から出力される乱数列 R S N における数値データを、乱数ラッチセクタ 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
20 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 2 8 】

例えば図 3 8 (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 8 (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。

【 0 3 2 9 】

また、図 9 (B) に示すような第 2 乱数初期設定 K R S 2 のビット番号 [0] における
30 ビット値を “ 1 ” とすれば、例えば乱数回路 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。

【 0 3 3 0 】

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
40 ビット値を “ 0 0 ” 以外の値とすることにより、システムリセットや電源投入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、電源投入やシステムリセット等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 3 1 】

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
50 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、電源投入やシステムリセット等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。

【0332】

また、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の発振周波数と、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CLK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509のクロック用フリップフロップ552により生成される乱数更新クロックRGKやラッチ用クロックRC0の発振周波数は、乱数用クロックRCLKの発振周波数の1/2となるが、制御用クロックCLKの発振周波数や、制御用クロックCLKの発振周波数の1/2となる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CLKや内部システムクロックSCLKと、乱数更新クロックRGK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にて乱数生成回路553や乱数列変更回路555により生成される乱数列RSNにおける数値データの更新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CLK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CLK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。

【0333】

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。

【0334】

メイン制御部41が備える外部バスインタフェース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。

【0335】

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

10

20

30

40

50

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。

【0336】

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”の書き込みを行い、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に容易に検知して、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すべて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。

10

【0337】

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KRS2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41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。

20

【0338】

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KSES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ROM506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うにしても良い。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、電源投入時や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、より困難になる。

30

【0339】

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41を構成するチップ毎に付与されるID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、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ID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、IDナンバーとを組み合わせる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変更する場合にも、フリーランカウンタのカウント値と、IDナンバーとを組み合わせる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。

40

【0340】

50

本実施例では、メイン制御部 4 1 の外部に設けられた乱数用クロック生成回路 4 3 により、制御用クロック生成回路 4 2 で生成される制御用クロック C C L K 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 R C L K を生成して、乱数回路 5 0 9 に供給するものであったが、メイン制御部 4 1 の C P U 5 0 5 に供給されるクロック信号と、乱数回路 5 0 9 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 R C L K と制御用クロック C C L K 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 R C 0、R C 1、R C 2 と制御用クロック C C L K あるいは内部システムクロック S C L K 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 4 2 と乱数用クロック生成回路 4 3 とは、その全部又は一部が、メイン制御部 4 1 の内部に設けられても良いし、メイン制御部 4 1 の外部に設けられても良い。

10

【 0 3 4 1 】

本実施例では、乱数回路 5 0 9 が乱数用クロック生成回路 4 3 により生成された乱数用クロック R C L K の供給を受け、クロック用フリップフロップ 5 5 2 により、乱数更新クロック R G K とラッチ用クロック R C 0 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 4 3 といった、乱数回路 5 0 9 の外部において、乱数更新クロック R G K やラッチ用クロック R C 0 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 5 0 9 の内部にて、乱数更新クロック R G K を生成するための回路と、ラッチ用クロック R C 0 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 5 5 2 と同様のフリップフロップにより乱数更新クロック R G K を生成する一方で、乱数更新クロック R G K 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 R C 0 として用いるようにしても良い。

20

【 0 3 4 2 】

本実施例に適用した乱数回路 5 0 9 では、乱数ラッチフラグデータ R D F M 0 が “ 1 ” の状態、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、新たな数値データ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 R 1 D に取り込まないようにしており、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。

30

【 0 3 4 3 】

このため、図 3 9 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 S S 1 が入力されて数値データがラッチされ、乱数値レジスタ R 1 D 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て乱数値格納ワークの数値データが更新されるまでは、乱数ラッチフラグデータ R D F M 0 が “ 1 ” の状態となることで、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 S S 1 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 S S 1 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い。

40

【 0 3 4 4 】

尚、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において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 R 1 D への書込を禁止する構成であっても良い。

【 0 3 4 5 】

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

50

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。

【0346】

これに対して本実施例では、約0.56ms毎に基本処理に割り込んで実行するタイマ割込処理4回に1回(約2.24ms)毎に実行される乱数値読出処理において乱数値レジスタ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、乱数値レジスタR1D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。すなわち一定時間間隔(約2.24ms)毎に、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を確認し、ラッチされていれば乱数値レジスタR1Dの数値データを読み出すことで、乱数値レジスタR1Dに数値データが保持されている状態を解除し、乱数値レジスタR1Dに新たな数値データの取込が可能な状態となる。

10

【0347】

このため、図40に示すように、ゲームの開始が許可されていない状態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、一定時間(乱数値読出処理の実行間隔である約2.24ms)以内に乱数値レジスタ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なるので、ゲームの開始が許可される前のスタートスイッチ7の操作によりラッチされた数値データが乱数値レジスタR1Dに保持されても、ゲームの開始が許可された後にスタートスイッチ7が操作されたタイミングで新たにラッチした数値データを取得することが可能となる。

20

【0348】

また、乱数値読出処理では、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、数値データを読み出すだけではなく、RAM507の乱数値格納ワークの値を読み出した数値データに更新することにより、乱数値レジスタR1Dに新たな数値データがラッチされる毎に、ラッチされた数値データがその後の乱数値読出処理において読み出され、乱数値格納ワークに格納された数値データが新たにラッチされた最新の数値データに更新されるようになっている。

【0349】

そして、ゲームの開始時に、乱数値レジスタR1Dに格納されている数値データではなく、RAM507の乱数値格納ワークに格納されている数値データを取得して内部抽選を行うので、乱数値レジスタR1Dから数値データが読み出された後に、スタートスイッチの信号線にノイズがのって数値データが変わってしまっても内部抽選に用いる乱数値格納ワークの数値データに影響することがなく、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した数値データを用いて内部抽選を行うことができる。

30

【0350】

また、本実施例では、電断した際に乱数値レジスタR1Dの値は保持されないが、乱数値格納ワークが割り当てられたRAM507は、電断しても格納データが保持されるようになっているため、スタートスイッチ7の操作が検出された後、内部抽選を開始する前の段階で瞬停などにより乱数値レジスタR1Dの値が消失しても乱数値格納ワークの数値データは維持されるため、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内部抽選に用いることができる。

40

【0351】

また、本実施例では、スタートスイッチ7の操作が検出された時点で、RAM507の乱数値格納ワークに格納されている数値データを同じくRAM507に割り当てられた抽選用ワークに設定し、その後の内部抽選において抽選用ワークに設定された数値データに対して演算を行うことにより役に当選したか否かの判定を行うようになっており、スタートスイッチ7の操作が検出された時点から内部抽選が終了するまでに乱数値格納ワークに

50

格納されている数値データが更新されても、スタートスイッチ7の操作が検出された時点で取得した数値データが内部抽選が終了するまでに変更されてしまうことがないが、スタートスイッチ7の操作が検出された時点でRAM507の乱数値格納ワークに格納されている数値データをCPU505のワークレジスタに設定し、その後の内部抽選においてワークレジスタに設定された数値データに対して演算を行うことにより役に当選したか否かの判定を行うようにしても良い。

【0352】

また、乱数値格納ワークに格納されている数値データに対して演算を行うことにより役に当選したか否かの判定を行うようにしても良いが、この場合には、スタートスイッチ7の操作が検出された時点から内部抽選が終了するまでの期間において、タイマ割込処理(メイン)を禁止することなどにより、乱数値格納ワークに格納されている数値データが新たにラッチされた数値データによって更新されてしまうことがないようにすることで、スタートスイッチ7の操作が検出された時点で取得した数値データが内部抽選が終了するまでに変更されてしまうことを防止できる。

10

【0353】

また、本実施例では、タイマ割込処理(メイン)4回に1回の割合でスイッチ類の検出信号を入力し、2回連続して検出信号の状態が一致した場合、すなわち約2.24msの期間検出信号の状態が同じ場合に、該当するスイッチ類の検出状態を確定させる。このため、スタートスイッチ7の操作(OFFからONへの変化)を検知するのに約2.24msかかることとなるが、乱数値読出処理がこの時間よりも短いと、スタートスイッチがOFFからONに変化した時点から検知されるまでの間に乱数値格納ワークの数値データが、ノイズなどによって新たにラッチされた数値データに更新されてしまう可能性がある。一方、乱数値読出処理がこの時間よりも長いとスタートスイッチ7の操作のタイミングでラッチした数値データが乱数値格納ワークの数値データに反映されない可能性がある。

20

【0354】

これに対して乱数値読出処理もタイマ割込処理(メイン)4回に1回の割合で実行され、乱数値格納ワークの数値データが更新されるのに要する最小時間も同じ約2.24msであり、スタートスイッチ7がOFFからONに変化した時点から検知されるまでに乱数値格納ワークの数値データが新たにラッチされた数値データに更新されてしまうことがなく、かつ、スタートスイッチ7の操作のタイミングでラッチした数値データを乱数値格納ワークの数値データに反映させることができる。

30

【0355】

また、本実施例では、図41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。

【0356】

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R1Dに数値データが格納されており、その状態が保持されていても、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。

40

【0357】

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、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

50

い。

【0358】

また、本実施例では、図42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。

10

【0359】

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、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。

【0360】

20

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。

【0361】

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が、本実施例では、図43に示すように、電源投入時に電断前の制御状態に復帰する場合には、復帰した時点、すなわちゲームの開始が許可された時点で、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。

30

【0362】

このため、図43に示すように、電源投入後、リセット信号がOFFとなってメイン制御部41が動作を開始した後、電断前の制御状態に復帰す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なる。

40

【0363】

また、電源投入時だけではなく、メイン制御部41がリセット要因の発生によりリセットされ、再起動した場合にも、起動処理の先頭から処理が開始することとなるが、この場合も同様で、メイン制御部41が再起動した際に、リセット前の制御状態に復帰する場合には、復帰した時点、すなわちゲームの開始が許可された時点で、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

50

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ことで、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 4 】

このため、リセット要因の発生によりメイン制御部 4 1 が再起動した後、リセット前の制御状態に復帰す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 R 1 D 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 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。

【 0 3 6 5 】

特に、スタートスイッチ 7 からの検出信号の入力を負論理で行っている場合には、瞬停時など電圧供給が不安定な状態では、復帰前にスタートスイッチ 7 の ON (電圧 LOW) の信号が入力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しまう可能性が高くなるが、このような場合で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 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り、復帰前に格納された乱数値レジスタ R 1 D の数値データが保持されたままとなってしまうことを確実に防止できる。

【 0 3 6 6 】

尚、本実施例では、電源投入後またはリセット後、電断前または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しても良い。

【 0 3 6 7 】

また、電源投入後またはリセット後、電断前または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、電源投入後またはリセット後、電断前または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しても良い。さらには、電源投入後またはリセット後、電断前またはリセット前の制御状態に復帰するか否かに関わらず、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しても良い。

【 0 3 6 8 】

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ちるか否かに基づいて判定する。

【 0 3 6 9 】

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 . 2 4 m s 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。

【 0 3 7 0 】

一方、スイッチ入力判定処理の実行間隔は約 2 . 2 4 m s 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 . 2 4 m s 以上 ON が継続して検知されること

10

20

30

40

50

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。

【0371】

このように本実施例では、図44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2.24ms）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。

【0372】

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2.24ms）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しても良く、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。

【0373】

以上、本発明の実施例1を図面により説明してきたが、本発明はこの実施例1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

【0374】

例えば、前記実施例では、メダル並びにクレジットを用いて賭数を設定するスロットマシンを用いているが、本発明はこれに限定されるものではなく、遊技球を用いて賭数を設定するスロットマシンや、クレジットのみを使用して賭数を設定する完全クレジット式のスロットマシンであっても良い。

【0375】

更に、流路切替ソレノイド30や投入メダルセンサ31など、メダルの投入機構に加えて、遊技球の取込を行う球取込装置、球取込装置により取り込まれた遊技球を検出する取込球検出スイッチを設けるとともに、ホッパーモータ34bや払出センサ34cなど、メダルの払出機構に加えて、遊技球の払出を行う球払出装置、球払出装置により払い出された遊技球を検出する払出球検出スイッチを設け、メダル及び遊技球の双方を用いて賭数を設定してゲームを行うことが可能であり、かつ入賞の発生によってメダル及び遊技球が払い出されるスロットマシンに適用しても良い。

【実施例2】

【0376】

次に、本発明が適用されたスロットマシンの実施例2について説明する。尚、本実施例の構成は、前述の実施例1と同一の構成を含むため、ここでは実施例1と相違する点について説明する。

【0377】

実施例1では、乱数割込制御データRDIC0のビット値が“0”に設定されており、乱数値レジスタR1Dとなる乱数値レジスタ559Aに数値データが取り込まれても割込は発生しない構成であったが、本実施例では、乱数割込制御データRDIC0のビット値が“1”に設定されており、乱数値レジスタR1Dとなる乱数値レジスタ559Aに数値データが取り込まれることで、乱数割込が発生するようになっている。

【0378】

そして、メイン制御部41は、乱数割込が発生した際に、基本処理に割り込んで乱数値ラッチ割込処理を実行する。尚、乱数割込が発生した際に、他の割込処理の実行中であれば、実行中の割込処理の終了を待って乱数値ラッチ割込処理を実行する。

【0379】

図45は、メイン制御部41が乱数割込の発生に伴い実行する乱数値ラッチ割込処理の制御内容を示すフローチャートである。尚、乱数値ラッチ割込処理の実行期間中は自動的に他の割込が禁止される。

【0380】

乱数値ラッチ割込処理では、まず、使用中のレジスタをスタック領域に退避した後（Sk1001）、乱数値レジスタ559Aから数値データを読み出し（Sk1002）、乱数値格納ワークに格納されている値を、Sk1002において読み出した数値データに更新し（Sk1003）、Sk1001においてスタック領域に退避したレジスタを復帰し（Sk1004）、割込前の処理に戻る。

【0381】

このように本実施例では、乱数値レジスタR1Dに数値データが取り込まれることで、乱数割込が発生し、これに伴いメイン制御部41は、基本処理に割り込んで乱数値ラッチ割込処理を実行し、乱数値レジスタR1D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。すなわち乱数値レジスタR1Dに数値データが取り込まれる毎に、乱数値ラッチ割込処理を実行して乱数値レジスタR1Dの数値データを読み出すことで、乱数値レジスタR1Dに数値データが保持されている状態を解除し、乱数値レジスタR1Dに新たな数値データの取込が可能な状態となる。

【0382】

このため、図46に示すように、ゲームの開始が許可されていない状態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も、数値データがラッチされたことに伴う割込の発生により乱数値レジスタR1Dの数値データが読み出され、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、ゲームの開始が許可される前のスタートスイッチ7の操作によりラッチされた数値データが乱数値レジスタR1Dに保持されても、ゲームの開始が許可された後にスタートスイッチ7が操作されたタイミングで新たにラッチした数値データを取得することが可能となる。

【0383】

また、乱数値ラッチ割込処理では、乱数値レジスタR1Dにラッチされた数値データを読み出すだけではなく、RAM507の乱数値格納ワークの値を読み出した数値データに更新することにより、乱数値レジスタR1Dに新たな数値データがラッチされる毎に、乱数値格納ワークに格納された数値データが新たにラッチされた最新の数値データに更新されるようになっている。

【0384】

そして、ゲームの開始時に、乱数値レジスタR1Dに格納されている数値データではなく、RAM507の乱数値格納ワークに格納されている数値データを取得して内部抽選を行うので、乱数値レジスタR1Dから数値データが読み出された後に、スタートスイッチの信号線にノイズがのって数値データが変わってしまっても内部抽選に用いる乱数値格納ワークの数値データに影響することがなく、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した数値データを用いて内部抽選を行うことができる。

【0385】

また、本実施例では、スタートスイッチ7の操作が検出された時点で、RAM507の乱数値格納ワークに格納されている数値データを同じくRAM507に割り当てられた抽選用ワークに設定し、その後の内部抽選において抽選用ワークに設定された数値データに対して演算を行うことにより役に当選したか否かの判定を行うようになっており、スタートスイッチ7の操作が検出された時点から内部抽選が終了するまでに乱数値格納ワークに格納されている数値データが更新されても、スタートスイッチ7の操作が検出された時点で取得した数値データが内部抽選が終了するまでに変更されてしまうことがないが、スタートスイッチ7の操作が検出された時点でRAM507の乱数値格納ワークに格納されている数値データをCPU505のワークレジスタに設定し、その後の内部抽選においてワ

10

20

30

40

50

ークレジスタに設定された数値データに対して演算を行うことにより役に当選したか否かの判定を行うようにしても良い。

【0386】

また、乱数値格納ワークに格納されている数値データに対して演算を行うことにより役に当選したか否かの判定を行うようにしても良いが、この場合には、スタートスイッチ7の操作が検出された時点から内部抽選が終了するまでの期間において、乱数割込の発生に伴う乱数値割込処理を禁止することなどにより、乱数値格納ワークに格納されている数値データが新たにラッチされた数値データによって更新されてしまうことがないようにすることで、スタートスイッチ7の操作が検出された時点で取得した数値データが内部抽選が終了するまでに変更されてしまうことを防止できる。

10

【0387】

尚、本実施例では、乱数値ラッチ割込処理において、乱数値レジスタR1Dの数値データを読み出すようになっているが、乱数値ラッチ割込処理では、乱数値レジスタR1Dに数値データがラッチされた旨を示すラッチフラグをRAM507に設定し、BET処理などの基本処理やタイマ割込処理(メイン)などにおいてラッチフラグがRAM507に設定されているか否かを判定し、RAM507にラッチフラグが設定されていると判定した場合に、乱数値レジスタR1Dの数値データを読み出す構成としても良く、このような構成とした場合でも、ゲームの開始が許可されていない状態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たままとなってしまうことがなく、ゲームの開始が許可された後にスタートスイッチ7が操作されたタイミングで新たにラッチした数値データを取得することが可能となる。

20

【0388】

以上、本発明の実施例2を説明してきたが、本発明はこの実施例2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。また、実施例1と同一もしくは類似する構成については、実施例1で説明したものと同様の効果を有するものである。また、実施例1について例示した変形例についても実施例2に適用可能である。

【符号の説明】

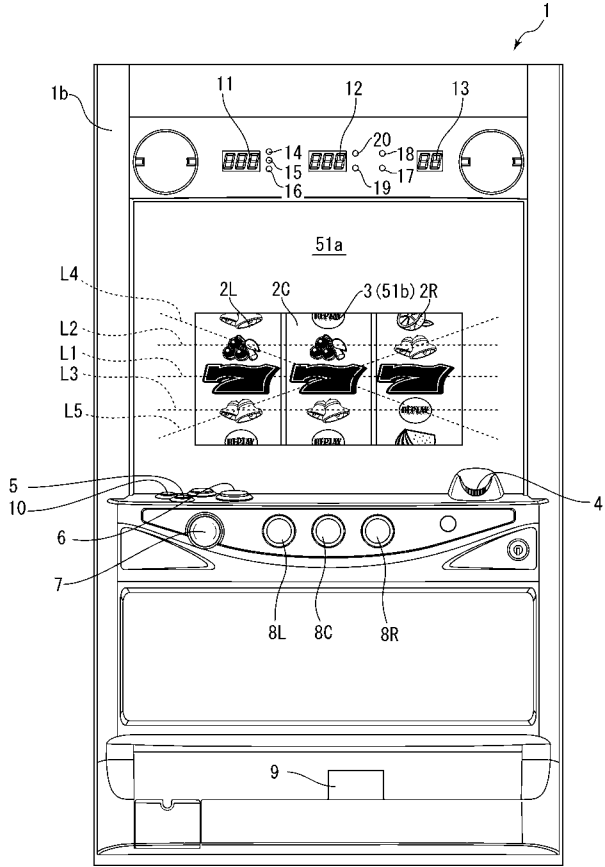
【0389】

- 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
- 41 メイン制御部
- 91 サブ制御部
- 505 CPU
- 506 ROM
- 507 RAM
- 509 乱数回路
- 559A 乱数値レジスタ

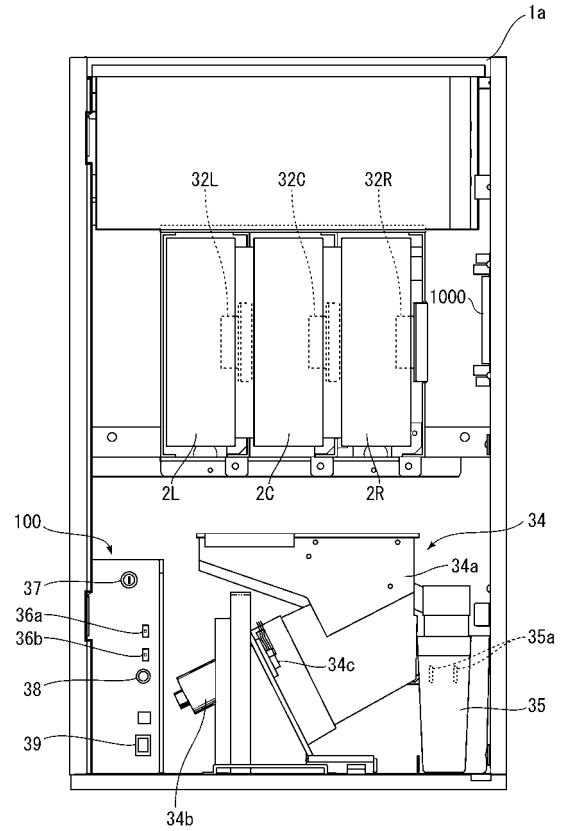
30

40

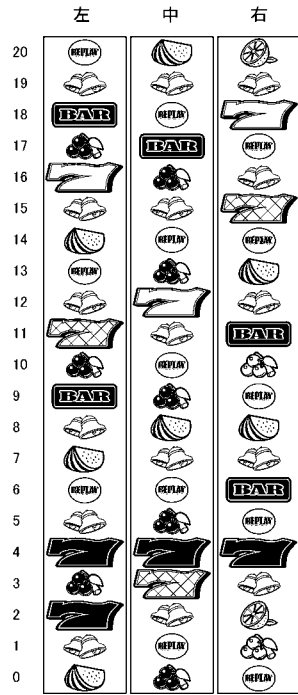
【図1】



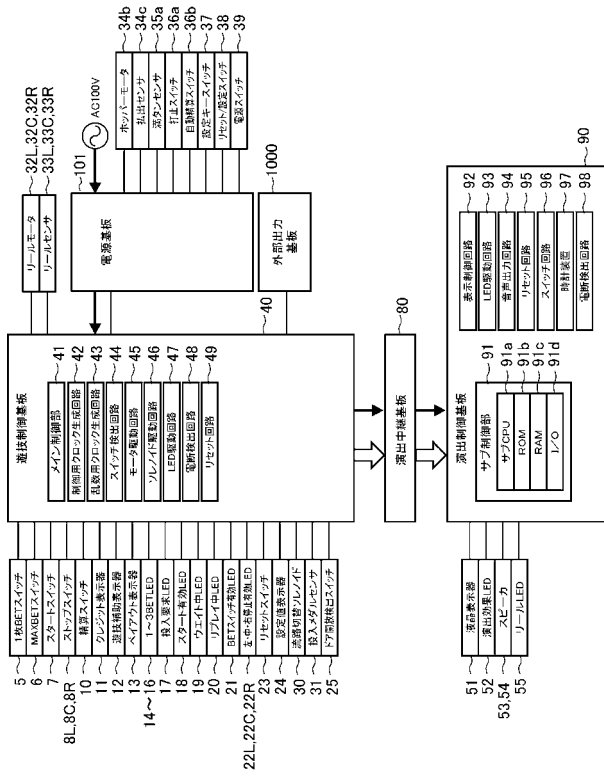
【図2】



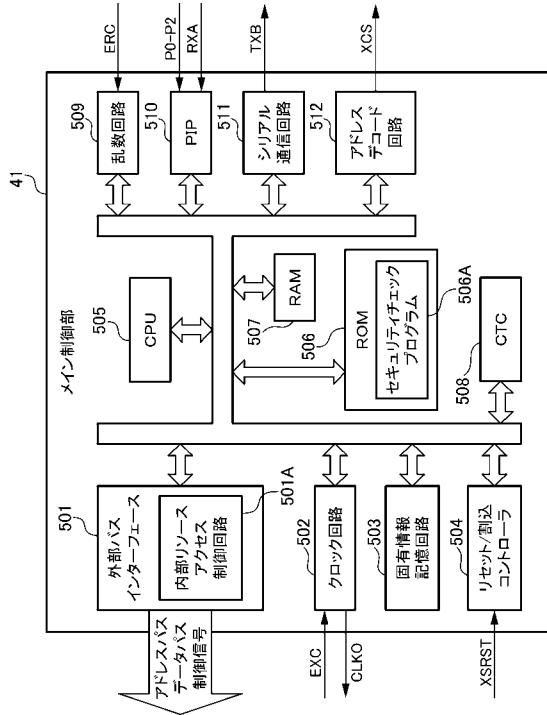
【図3】



【図4】



【図5】



【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		...
7E00H	RAM	I/Oマップ
7FFFH		メモリマップ
FDD0H	アドレスデコード回路	XCSデコードエリア
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(KIIS)
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チャンネル通信設定レジスタ(SAFM)
201B	シリアル通信回路	第1チャンネル割込制御レジスタ(SAIC)
201C	シリアル通信回路	第1チャンネルコマンドレジスタ(SACM)
201E	シリアル通信回路	第1チャンネルステータスレジスタ(SAST)
201F	シリアル通信回路	第1チャンネルデータレジスタ(SADT)
2022	シリアル通信回路	第2チャンネル通信設定レジスタ(SBFM)
2023	シリアル通信回路	第2チャンネル割込制御レジスタ(SBIC)
2024	シリアル通信回路	第2チャンネルコマンドレジスタ(SBCM)
2026	シリアル通信回路	第2チャンネルステータスレジスタ(SBST)
2027	シリアル通信回路	第2チャン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	割込中モニタレジスタ(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	乱数ラッチフラグレジスタ(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)		
1FC0	4C	00	左記以外
1FC1	45	00	
1FC2	34	00	
1FC3	33	00	
1FC4	38	00	
1FC5	30	00	
1FC6	41	00	
1FC7	53	00	
ROM読出許可/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チャンネル送信用 1:チップセレクト(XCS13)
3	TXA端子設定 0:シリアル通信第1チャンネル送信用 1:チップセレクト(XCS12)
2	RXA端子設定 0:シリアル通信第1チャンネル受信用 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: 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

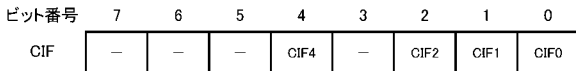
(D) KSES ビット番号2-0

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_{SCLK} : 内部システムクロック(SCLK)の周期

【図11】

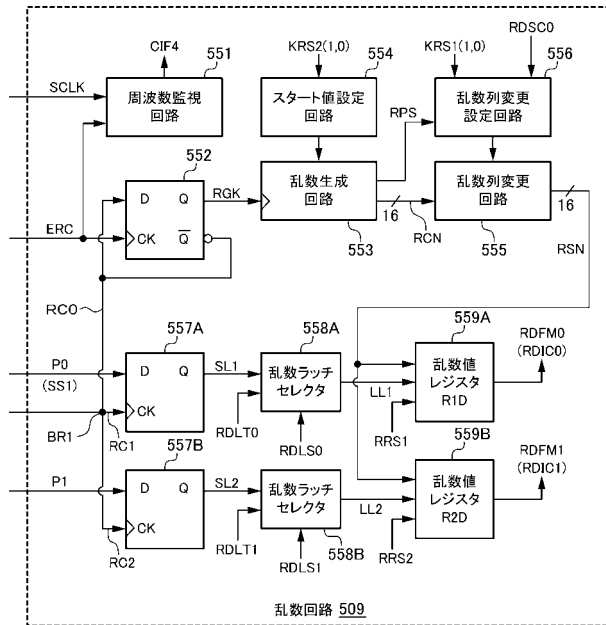
(A)



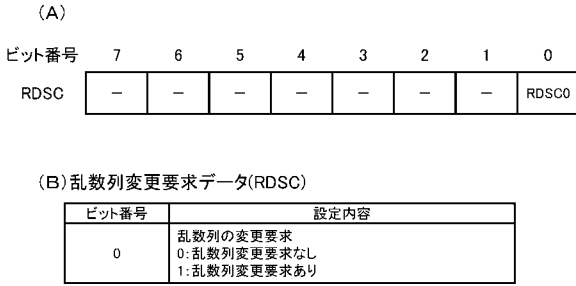
(B) 内部情報データ(CIF)

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

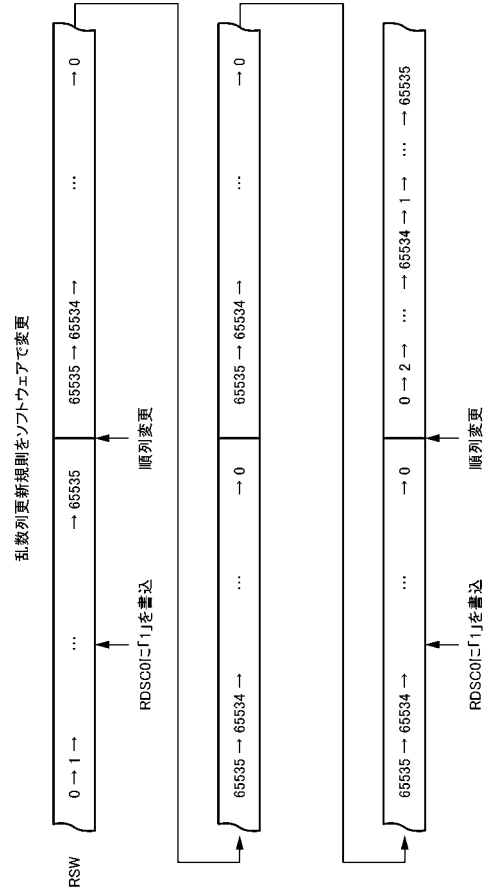
【図12】



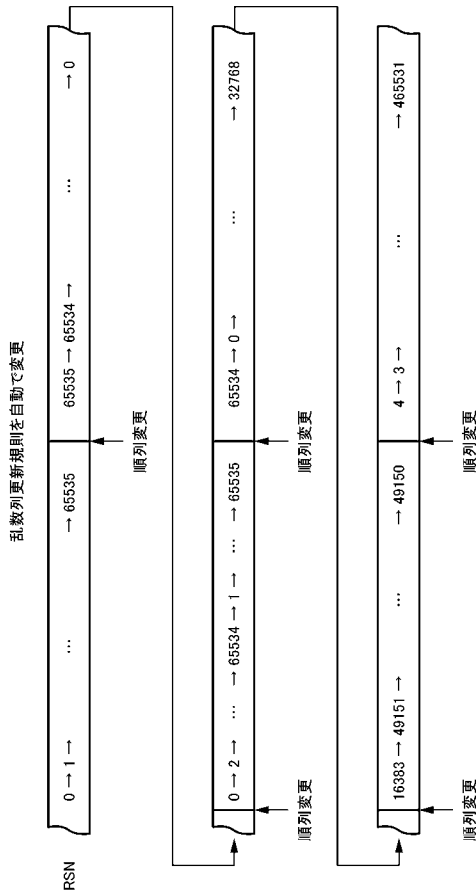
【図13】



【図14】



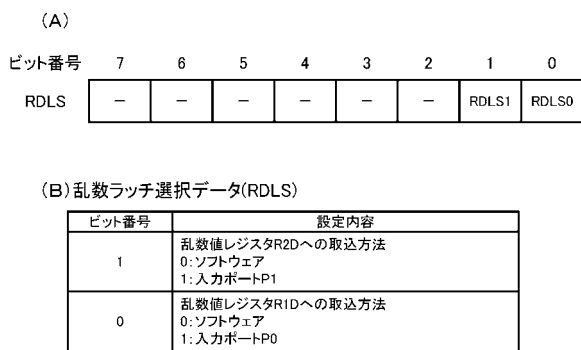
【図15】



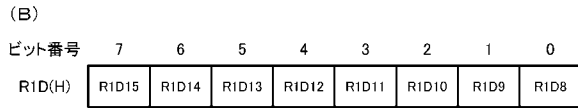
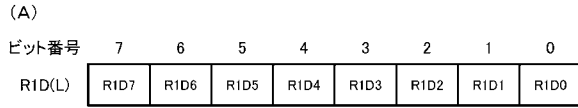
【図16】



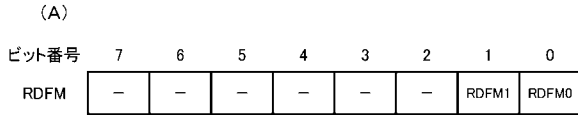
【図17】



【図18】



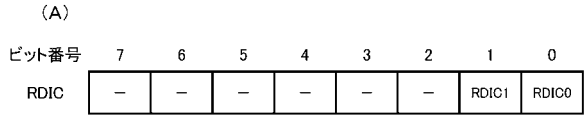
【図19】



(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: 乱数値取込あり

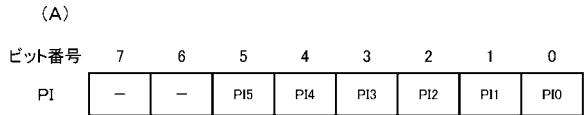
【図20】



(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】



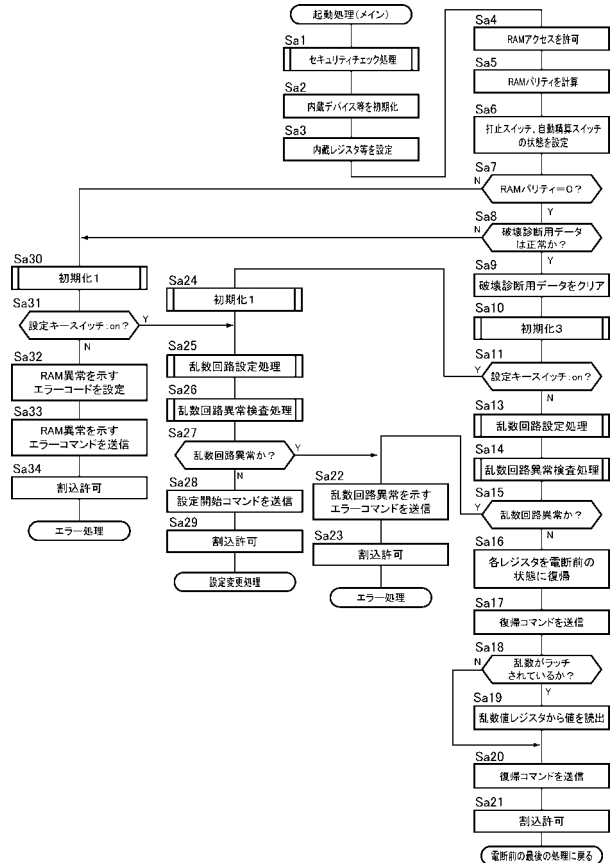
(B) 入力ポートデータ(PI)

ビット番号	設定内容
5	P5(RXA)の端子状態
4	P4(XNMD)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

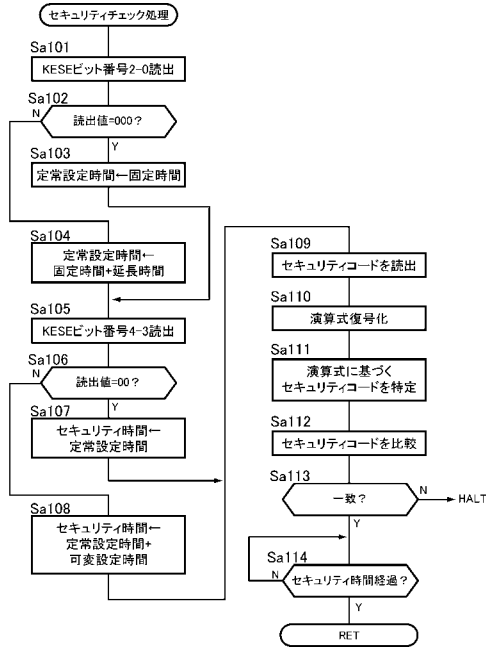
【図22】

役	図柄組み合わせ	遊技状態		
		通常	遊技中	RB(BB)
BB	黒7-黒7-黒7	○	×	×
	緑7-緑7-緑7	○	×	×
	白7-白7-白7	○	×	×
RB	網7-網7-黒7	○	×	×
リプレイ	リプレイ-リプレイ-リプレイ	○	○	×
	BAR-リプレイ-リプレイ	○	○	×
	旗7-リプレイ-リプレイ	○	○	×
スイカ	スイカ-スイカ-スイカ	○	○	○
	スイカ-スイカ-BAR	○	○	○
チェリー	ANY-ANY-白チェリー	○	○	○
ベル	ベル-ベル-ベル	○	○	○

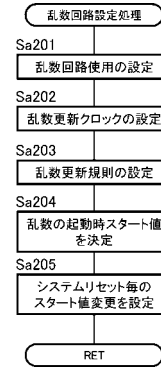
【図23】



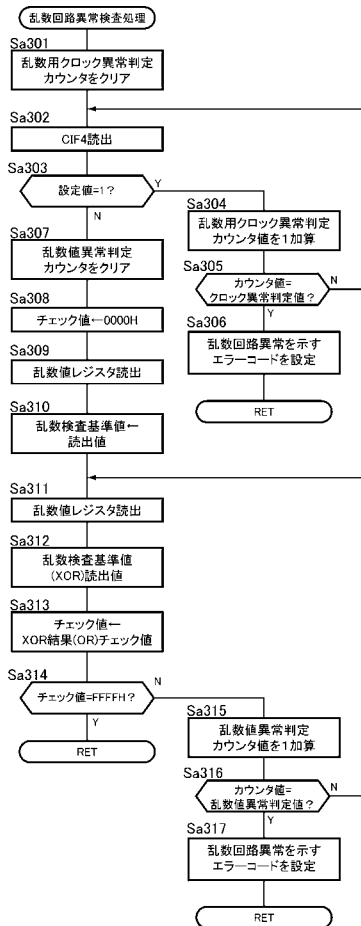
【図24】



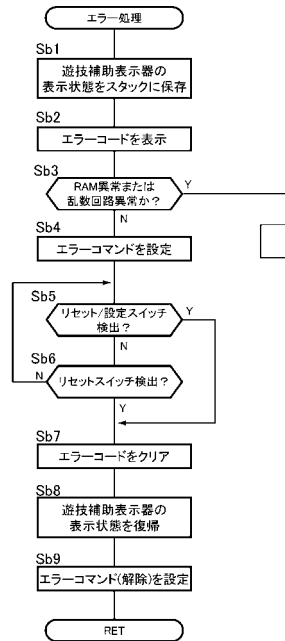
【図25】



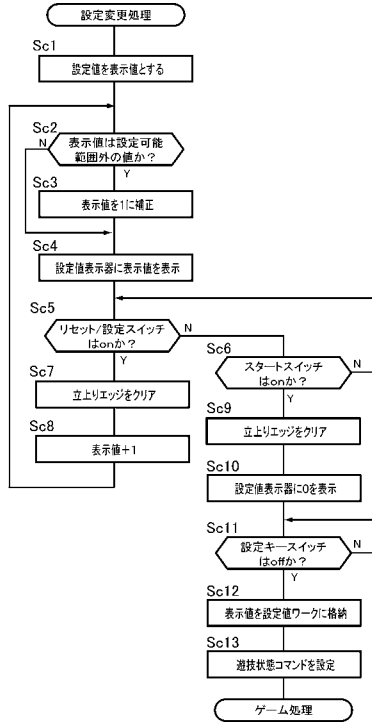
【図26】



【図27】



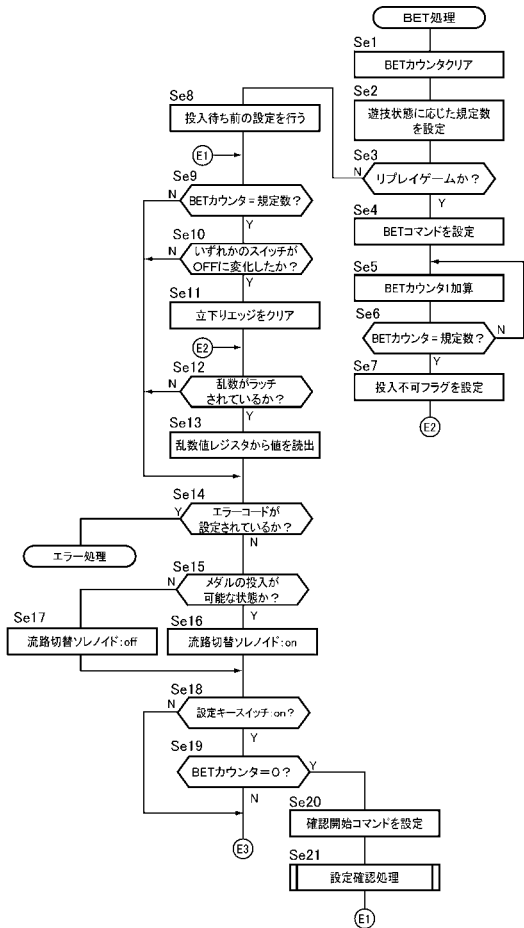
【図28】



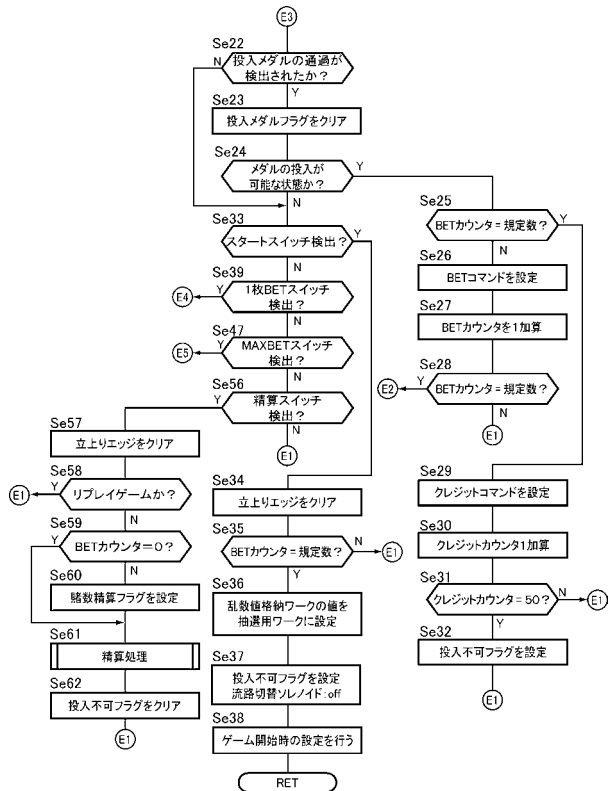
【図29】



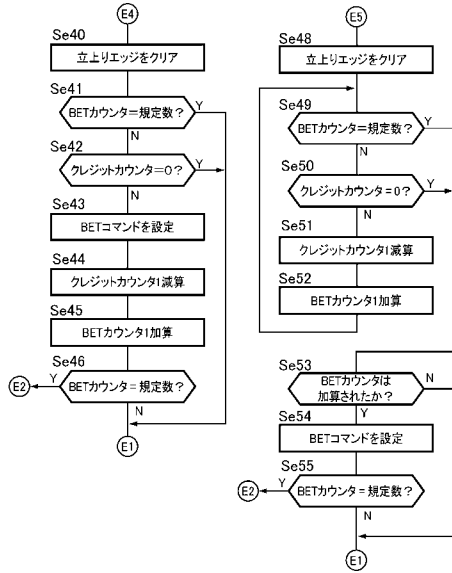
【図30】



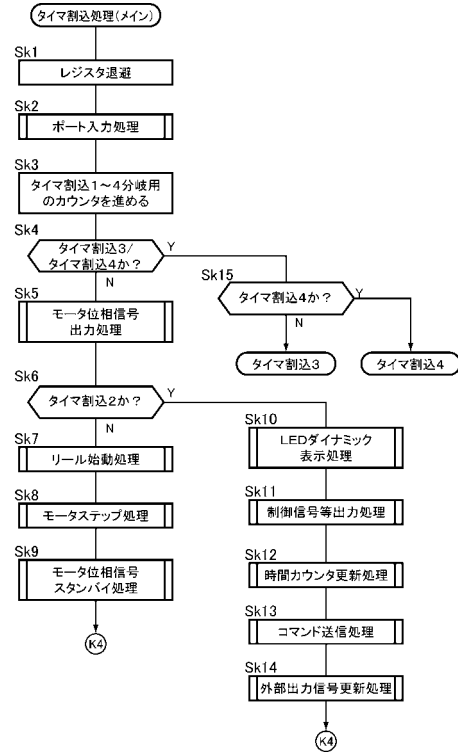
【図31】



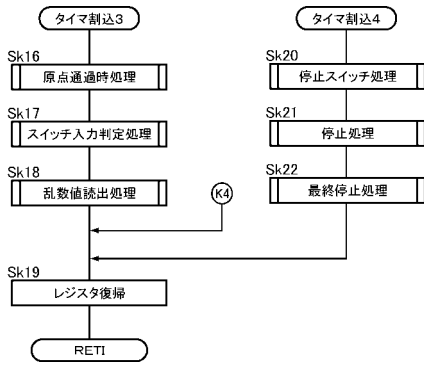
【図32】



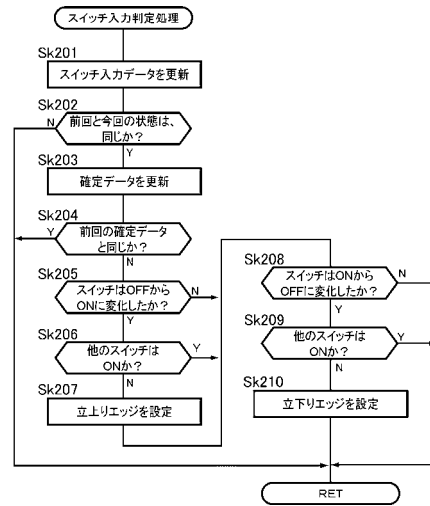
【図33】



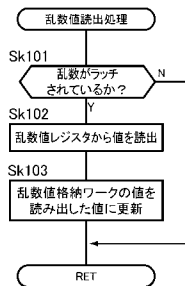
【図34】



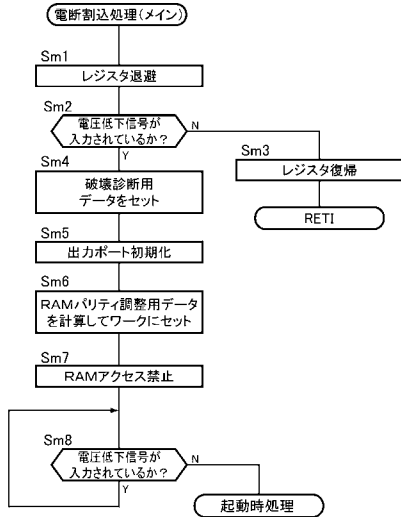
【図36】



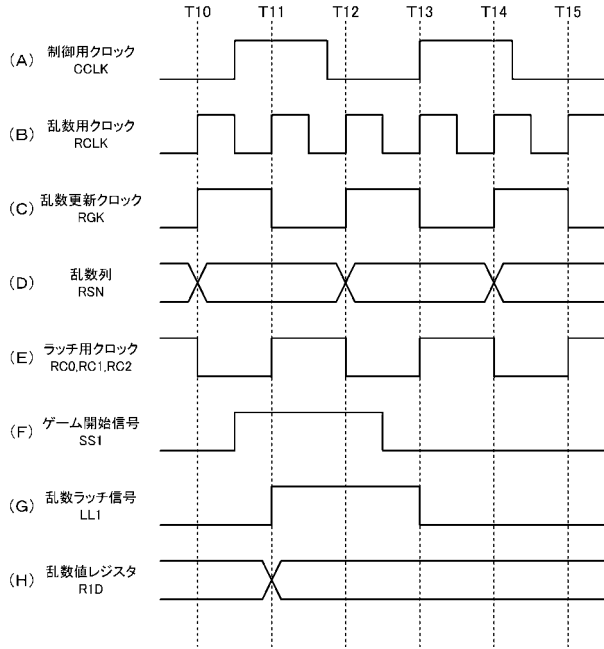
【図35】



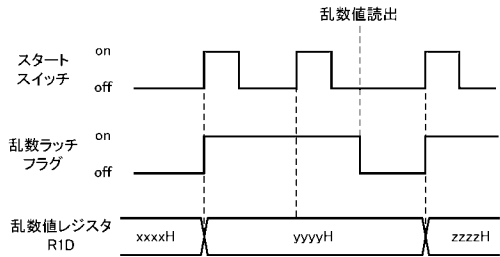
【図37】



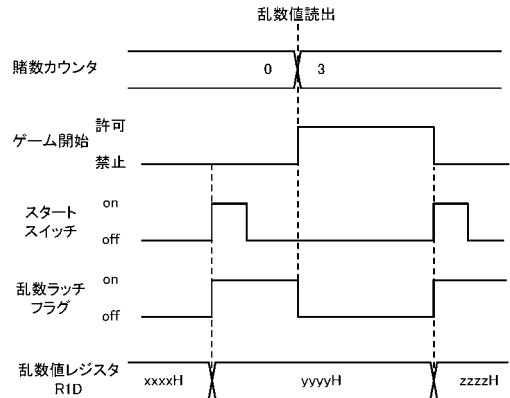
【図38】



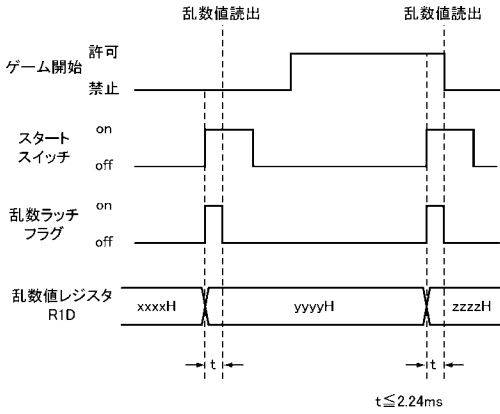
【図39】



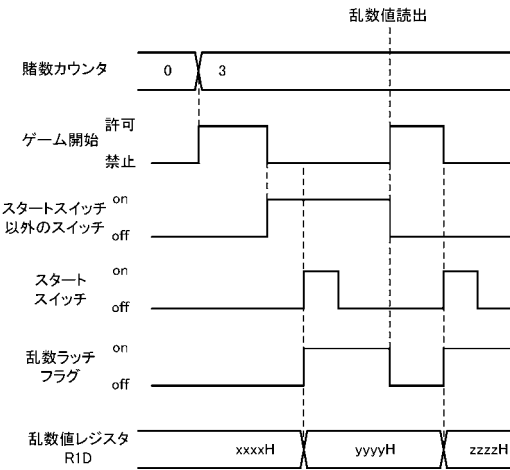
【図41】



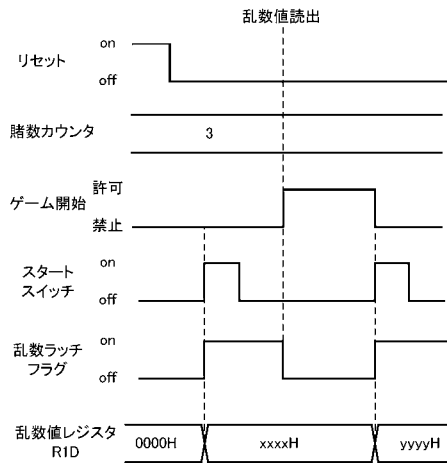
【図40】



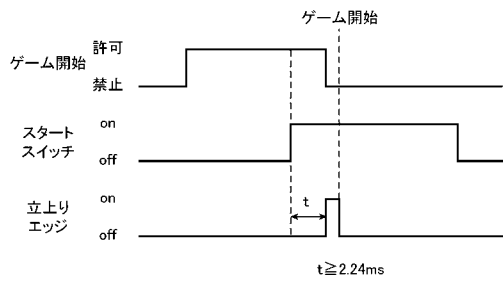
【図42】



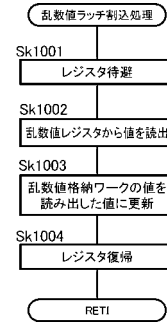
【図43】



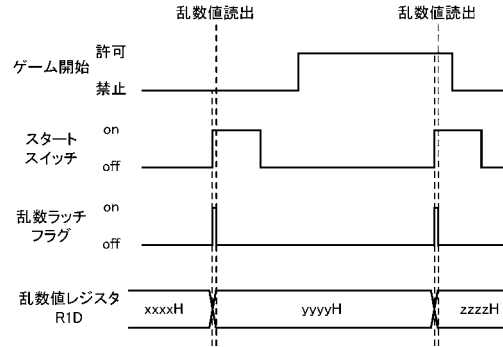
【図44】



【図45】



【図46】



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 渡邊 有一
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)
特許第4498454(JP, B2)
特許第4684328(JP, B2)
特許第4790824(JP, B2)
特許第4964260(JP, B2)
特許第4964261(JP, B2)
特開2010-115296(JP, A)
特開2006-333949(JP, A)
特開2009-089966(JP, A)
LE4380A - ユーザーズマニュアル, 株式会社エルイーテック, 2008年 1月15日
, p.144-145
LE4168B - ユーザーズマニュアル, 株式会社エルイーテック, 2008年 9月25日
, p.144-145

(58)調査した分野(Int.Cl., DB名)

A63F 5/04
A63F 7/02