(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4281138号
(P4281138)
(45)発行日　平成21年6月17日(2009.6.17)　(24)登録日　平成21年3月27日(2009.3.27)

(51)Int.Cl.　F I
A63F 7/02 (2006.01)　A63F 7/02 315Z
A63F 7/02 334

請求項の数 1　(全 17 頁)

(21)出願番号　特願平11-14910
(22)出願日　平成11年1月22日(1999.1.22)
(65)公開番号　特開2000-225260(P2000-225260A)
(43)公開日　平成12年8月15日(2000.8.15)
審査請求日　平成18年1月17日(2006.1.17)
(31)優先権主張番号　特願平10-345923
(32)優先日　平成10年12月4日(1998.12.4)
(33)優先権主張国　日本国(JP)

(73)特許権者　000144522
株式会社三洋物産
愛知県名古屋市千種区今池３丁目９番２１号
(74)代理人　110000534
特許業務法人しんめいセンチュリー
(74)代理人　100103045
弁理士　兼子　直久
(72)発明者　保谷　誠
名古屋市千種区今池３丁目９番２１号
株式会社　三洋物産内

最終頁に続く

(54)【発明の名称】弾球遊技機

(57)【特許請求の範囲】
【請求項１】
定期的に実行される割込処理と、２バイトで構成された乱数カウンタと、その乱数カウンタの値を前記割込処理によって更新する第１更新手段と、所定の契機により前記乱数カウンタの値を読み出す読出手段とを備え、その読出手段により読み出された前記乱数カウンタの値が予め定められた値の１つと一致する場合に、遊技者に所定条件下で所定の遊技価値を付与する弾球遊技機において、
１バイトで構成されたバイトカウンタと、
そのバイトカウンタの値を前記割込処理によって次回のその割込処理が発生するまでの残余時間の間に繰り返し更新する第２更新手段と、
更新中の前記乱数カウンタの初期値を記憶する現初期値メモリと、
前記バイトカウンタの値に基づいて、前記乱数カウンタおよび現初期値メモリの値を変更する変更手段とを備え、
前記現初期値メモリは１バイトで構成されると共に、
前記変更手段は、
その１バイトの現初期値メモリと２バイトの前記乱数カウンタとの一致の判断を、その乱数カウンタの上位バイトが所定値である場合に、その乱数カウンタの下位バイトの値と前記現初期値メモリの値とが一致するか否かにより行う比較手段と、
その比較手段により一致すると判断された場合に、１バイトの前記バイトカウンタの値を２バイトの前記乱数カウンタの下位バイトおよび１バイトの前記現初期値メモリへ書き

込む書き込み手段とを備えていることを特徴とする弾球遊技機。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、パチンコ遊技機などに代表される弾球遊技機に関し、特に、「ぶら下げ基板」等による不正行為を防止することができる弾球遊技機に関するものである。
【０００２】
【従来の技術】
この種のパチンコ遊技機は、複数種類の図柄を変動表示可能な表示装置を備えており、遊技領域に打ち込まれた打球が図柄作動ゲートを通過すると、変動表示を開始するように構成されている。この変動表示が予め定められた図柄の組み合わせと一致して停止すると、大当たりとなって、遊技者に所定の遊技価値が付与され、大量の遊技球が払出可能な状態となる。
【０００３】
かかる大当たりの発生の有無は、打球が図柄作動ゲートを通過するタイミングで決定される。即ち、１カウントずつ定期的に一定の範囲で（例えば、１カウントずつ、２ｍｓ毎に、０から６３０の範囲で）更新されるカウンタを備え、打球が図柄作動ゲートを通過したときに、そのカウンタの値を読み出して、読み出されたカウンタの値が、例えば「７」などの所定値と一致する場合に、大当たりを発生するようにしている。大当たりが発生すると、制御基板のコネクタに接続されたケーブルを介して、大当たりコマンドが表示装置の表示用基板へ送信される。表示装置では、受信された大当たりコマンドに基づいて、変動表示を制御し、所定の図柄の組み合わせで停止する大当たり表示を現出させるのである。
【０００４】
【発明が解決しようとする課題】
ところが、最近、「ぶら下げ基板」と呼ばれる不正な基板を使用した不正行為が報告されている。この不正行為は、制御基板と表示装置の表示用基板との間に、不正な基板をぶら下げて（不正な「ぶら下げ基板」を取り付けて）、不当に大当たりを発生させるというものである。具体的には、前記したパチンコ遊技機に設けられる大当たりを決定するためのカウンタと同様の働きをするカウンタ（１カウントずつ定期的に一定の範囲で更新されるカウンタ）を「ぶら下げ基板」内に設け、そのカウンタの値をパチンコ遊技機の電源投入に合わせてリセット（０クリア）することにより、「ぶら下げ基板」内で大当たりの発生タイミングを把握するのである。そして、その把握した大当たりの発生タイミングに合わせて、「ぶら下げ基板」内で打球の図柄作動ゲート通過信号を不正に生成し、これをパチンコ遊技機の制御基板へ出力して、不当に大当たりを発生させるというものである。遊技場などでは、この「ぶら下げ基板」を用いた不正行為により、多大な被害を被っているという問題点があった。
【０００５】
本発明は上述した問題点を解決するためになされたものであり、大当たりの発生タイミングの把握を不可能にして、「ぶら下げ基板」等を用いた不正行為を防止することができる弾球遊技機を提供することを目的としている。
【０００６】
【課題を解決するための手段】
この目的を達成するために請求項１記載の弾球遊技機は、定期的に実行される割込処理と、２バイトで構成された乱数カウンタと、その乱数カウンタの値を前記割込処理によって更新する第１更新手段と、所定の契機により前記乱数カウンタの値を読み出す読出手段とを備え、その読出手段により読み出された前記乱数カウンタの値が予め定められた値の１つと一致する場合に、遊技者に所定条件下で所定の遊技価値を付与するものであり、１バイトで構成されたバイトカウンタと、そのバイトカウンタの値を前記割込処理によって次回のその割込処理が発生するまでの残余時間の間に繰り返し更新する第２更新手段と、

更新中の前記乱数カウンタの初期値を記憶する現初期値メモリと、前記バイトカウンタの値に基づいて、前記乱数カウンタおよび現初期値メモリの値を変更する変更手段とを備え、前記現初期値メモリは１バイトで構成されると共に、前記変更手段は、その１バイトの現初期値メモリと２バイトの前記乱数カウンタとの一致の判断を、その乱数カウンタの上位バイトが所定値である場合に、その乱数カウンタの下位バイトの値と前記現初期値メモリの値とが一致するか否かにより行う比較手段と、その比較手段により一致すると判断された場合に、１バイトの前記バイトカウンタの値を２バイトの前記乱数カウンタの下位バイトおよび１バイトの前記現初期値メモリへ書き込む書き込み手段とを備えている。

【０００７】

この請求項１記載の弾球遊技機によれば、定期的に実行される割込処理において、乱数カウンタの値は、第１更新手段により更新されると共に、所定の契機により読出手段によって読み出される。読み出された乱数カウンタの値が予め定められた値の１つと一致すると、大当たりとなって、遊技者に所定条件下で所定の遊技価値が付与される。

【０００８】

また、変更手段によって、乱数カウンタおよび現初期値メモリの値がバイトカウンタの値に基づいて変更される。即ち、乱数カウンタの更新の初期値がバイトカウンタの値に基づいて変更される。このように、乱数カウンタの更新の初期値は、固定値ではなく、定期的に変更される値であるので、弾球遊技機の電源投入に合わせて、「ぶら下げ基板」等がその内部の不正なカウンタをリセットしても、その不正なカウンタの値を乱数カウンタの値と一致させることはできない。従って、「ぶら下げ基板」等が大当たりの発生タイミングを把握することを防止することができるのである。

【０００９】

しかも、バイトカウンタの値は、第２更新手段によって、次回の割込処理が発生するまでの残余時間の間に繰り返し更新される。割込処理の残余時間の長さは、遊技の状態に応じて変化するので、「ぶら下げ基板」等で把握することはできない。よって、かかる残余時間の間に繰り返し更新されるバイトカウンタの値に基づいて乱数カウンタの更新の初期値を変更することにより、「ぶら下げ基板」等による乱数カウンタの値の把握を一層効果的に防止することができる。

【００１０】

ところで、バイトカウンタの値を繰り返し更新する割込処理が命令の途中で実行される場合がある。よって、バイトカウンタを２バイト以上で構成すると、割込処理の発生タイミングによっては更新後のバイトカウンタの値が乱数カウンタの更新の範囲外の値になることがある。バイトカウンタの更新は、その値が一旦ＣＰＵ（内部レジスタまたはＡＬＵ）へ読み込まれ、ＣＰＵ内で更新され、更新後の値がバイトカウンタへ書き込まれることにより行われる。よって、バイトカウンタが２バイト以上で構成されると、更新後の値は１バイトずつバイトカウンタへ書き込まれるので、更新後の一部のバイトの書き込み後であって全バイトの書き込み前に次の割込が発生すると、書き込み途中の値が更新後のバイトカウンタの値となり、乱数カウンタの更新の範囲外の値になってしまうことがある。

【００１１】

乱数カウンタの更新の初期値はバイトカウンタの値に基づいて変更されるので、例えば、バイトカウンタの値を乱数カウンタへ書き込んで乱数カウンタの更新の初期値を変更する場合、バイトカウンタの値が乱数カウンタの更新の範囲外の値となると、乱数カウンタの値は本来更新されるべき範囲外の値となって、所定の不具合が生じる。例えば、大当たりの発生確率が予定していた確率と異なったものになったり、乱数カウンタの更新の初期値が以降は変更されなくなるのである。しかし、バイトカウンタは１バイトで構成されるので、割込処理がいかなるタイミングで発生しても、書き込み途中の値が更新後のバイトカウンタの値となることはなく、上記した不具合を回避することができる。

【００１２】

【発明の実施の形態】

以下、本発明の好ましい実施例について、添付図面を参照して説明する。本実施例では

、弾球遊技機の一例としてパチンコ遊技機、特に、第１種パチンコ遊技機を用いて説明する。なお、本発明を第３種パチンコ遊技機や他の弾球遊技機に用いることは、当然に可能である。

【００１３】

図１は、第１実施例のパチンコ遊技機Ｐの遊技盤の正面図である。遊技盤１の周囲には、打球が入賞することにより５個から１５個の遊技球が払い出される複数の入賞口２が設けられている。また、遊技盤１の中央には、複数種類の識別情報としての図柄などを表示する液晶（ＬＣＤ）ディスプレイ３が設けられている。このＬＣＤディスプレイ３の表示画面は横方向に３分割されており、３分割された各表示領域において、それぞれ図柄の変動表示が行われる。

【００１４】

ＬＣＤディスプレイ３の下方には、図柄作動ゲート（第１種始動口）４が設けられ、打球がこの図柄作動ゲート４を通過することにより、前記したＬＣＤディスプレイ３の変動表示が開始される。図柄作動ゲート４の下方には、特定入賞口（大入賞口）５が設けられている。この特定入賞口５は、ＬＣＤディスプレイ３の変動後の表示結果が予め定められた図柄の組み合わせの１つと一致する場合に、大当たりとなって、打球が入賞しやすいように所定時間（例えば、３０秒経過するまで、あるいは、打球が１０個入賞するまで）開放される入賞口である。この特定入賞口５内には、Ｖゾーン５ａが設けられており、特定入賞口５の開放中に、打球がＶゾーン５ａ内を通過すると、継続権が成立して、特定入賞口５の閉鎖後、再度、その特定入賞口５が所定時間（又は、特定入賞口５に打球が所定個数入賞するまで）開放される。この特定入賞口５の開閉動作は、最高で１６回（１６ラウンド）繰り返し可能にされており、開閉動作の行われ得る状態が、いわゆる所定の遊技価値の付与された状態（特別遊技状態）である。

【００１５】

図２は、かかるパチンコ遊技機Ｐの電気的構成を示したブロック図である。パチンコ遊技機Ｐの制御部Ｃは、演算装置であるＣＰＵ１１と、そのＣＰＵ１１により実行される各種の制御プログラムや固定値データを記憶したＲＯＭ１２と、各種のデータ等を一時的に記憶するためのメモリであるＲＡＭ１３とを備えている。図３から図５に示すフローチャートのプログラムは、制御プログラムの一部としてＲＯＭ１２内に記憶されている。

【００１６】

ＣＰＵ１１は、演算を行うＡＬＵのほか、アキュームレータ（以下「Ａｃｃ」と称す）１１ａや複数の内部レジスタ１１ｂ、フラグレジスタ１１ｃを備えている。ＲＡＭ１３内に設けられるカウンタ等の値は、一旦、ＣＰＵ１１の内部レジスタ１１ｂへロードされ（書き込まれ）、その内部レジスタ１１ｂ内で更新された後に、ＲＡＭ１３の元のカウンタ内へセイブされて（書き込まれて）、更新される。

【００１７】

なお、６８系の８ビットＣＰＵ１１では、ペアになっている２バイト（１６ビット）の内部レジスタ１１ｂの値を、連続したアドレスの２バイトのメモリ（ＲＡＭ１３内）へ１命令でセイブする（書き込む）ことができる。この場合の書き込みは、バスライン１４のデータバスは８ビットで構成されるので、上位バイト、下位バイトの順に行われる。また、８０系の８ビットＣＰＵでは、６８系のＣＰＵ１１とは逆に、ペアになっている２バイト（１６ビット）の内部レジスタの値を、連続したアドレスの２バイトのメモリへ、下位バイト上位バイトの順に１命令でセイブすることができる。

【００１８】

ＲＡＭ１３は、乱数カウンタ１３ａと、現初期値メモリ１３ｂと、次初期値メモリ１３ｃと、バイトカウンタ１３ｄとを備えている。乱数カウンタ１３ａは、大当たりの発生を決定するためのカウンタであり、図５の乱数更新処理（Ｓ７）によって、「０～６３０（０～２７６ｈ）」の範囲で、２ｍｓ毎に１カウントずつ更新される。このため乱数カウンタ１３ａは２バイトで構成されている。打球が図柄作動ゲート４を通過したときに取得した乱数カウンタ１３ａの値が例えば「７」であると、大当たりが発生する。大当たりが発

生すると、大当たりコマンドが制御部Ｃから後述する表示装置Ｄへ送られる。表示装置Ｄは、この大当たりコマンドに基づいて、ＬＣＤディスプレイ３の変動表示を大当たりの状態に制御する。

【００１９】

現初期値メモリ１３ｂは、乱数カウンタ１３ａの更新中の初期値を記憶するためのメモリであり、次初期値メモリ１３ｃは、乱数カウンタ１３ａの次回の更新の初期値を記憶するためのメモリである。現初期値メモリ１３ｂ及び次初期値メモリ１３ｃは、いずれも乱数カウンタ１３ａと同様に２バイトで構成されており、いずれも乱数カウンタ１３ａの更新範囲と同じ「０～６３０（０～２７６ｈ）」の範囲内で更新される。本実施例では、乱数カウンタ１３ａの更新の初期値は、乱数カウンタ１３ａの一回りの更新毎に変更される。よって、更新された乱数カウンタ１３ａの値が現初期値メモリ１３ｂの値と一致すると、乱数カウンタ１３ａの一回りの更新が終了したことになるので、両値１３ａ，１３ｂの一致を契機として、そのときの次初期値メモリ１３ｃの値が乱数カウンタ１３ａおよび現初期値メモリ１３ｂに書き込まれ、乱数カウンタ１３ａの更新の初期値が変更される。従って、乱数カウンタ１３ａの更新の初期値を変更しても、乱数の一様性（連続で取得した場合に同じ値を取ることがなく、しかも、すべての値が同じ確率で取り出せること）のある乱数値を得ることができるのである。

【００２０】

バイトカウンタ１３ｄは、次初期値メモリ１３ｃの値を更新するためのカウンタであり、１バイトで構成されている。バイトカウンタ１３ｄの値は、リセット割込処理のＳ１からＳ２０の処理の終了後、次のリセット割込が発生するまでの残余時間の間に「０～２５５（０～０ＦＦｈ）」の範囲内で繰り返し「１」ずつ加算され（インクリメントされ）更新される。更新されたバイトカウンタ１３ｄの値は、乱数更新処理（Ｓ７）の前に実行される次初期値メモリ更新処理（Ｓ６）において、次初期値メモリ１３ｃの値に加算されて、次初期値メモリ１３ｃの値を更新する。

【００２１】

リセット割込処理は２ｍｓ毎に実行されるが、Ｓ１からＳ２０の各処理の処理時間は最大の場合でも２ｍｓ以内に終了するようにプログラムされており、必ず、残余時間が生じるようにされている。１回のリセット割込処理において実行されるＳ１からＳ２０の各処理の処理時間は遊技の状況に応じて変化するので、リセット割込処理の残余時間は、一定な時間ではなく、遊技の状況に応じて変化する不定な時間となる。「ぶら下げ基板」ではこの不定な時間を把握することはできないので、かかる不定な時間内に繰り返し更新されるバイトカウンタ１３ｄの値に基づいて次初期値メモリ１３ｃの値を更新し、その次初期値メモリ１３ｃの値を乱数カウンタ１３ａの更新の初期値として使用することにより、「ぶら下げ基板」による大当たり発生のタイミングの把握を不可能にしている。

【００２２】

なお、バイトカウンタ１３ｄの値の更新は、その値が一旦ＣＰＵ１１内へ読み込まれ、ＣＰＵ１１内で更新された後にバイトカウンタ１３ｄへ書き込まれて更新される。一方、リセット割込は、割込の発生を禁止することができないノンマスカブルな割込であると共に、割込の優先順位が最も高く、ＣＰＵ１１の命令の実行途中であっても強制的に開始される割込である。よって、バイトカウンタ１３ｄの更新中のいかなるタイミングで次のリセット割込が発生しても不具合が生じないように、バイトカウンタ１３ｄを１バイトで構成している。１バイトのデータ書き込みは一度に行われるので、バイトカウンタ１３ｄを１バイトで構成することにより、バイトカウンタ１３ｄの値が書き込み途中の値になってしまうことを防止できるからである。

【００２３】

これらのＣＰＵ１１、ＲＯＭ１２、ＲＡＭ１３は、バスライン１４を介して互いに接続されており、バスライン１４は、また、入出力ポート１５にも接続されている。この入出力ポート１５は表示装置Ｄや他の入出力装置１６と接続されている。制御部Ｃは、入出力ポート１５を介して、表示装置Ｄや他の入出力装置１６へ動作コマンドを送り、それら各

装置を制御する。ＬＣＤディスプレイ３の変動表示や特定入賞口５の開閉動作も、この動作コマンドに基づいて制御される。

【００２４】

表示装置Ｄは、ＣＰＵ２１と、プログラムＲＯＭ２２と、ワークＲＡＭ２３と、ビデオＲＡＭ２４と、キャラクタＲＯＭ２５と、画像コントローラ２６と、入出力ポート２７と、ＬＣＤディスプレイ３とを備えている。表示装置ＤのＣＰＵ２１は、制御部Ｃから出力される動作コマンドに応じて、ＬＣＤディスプレイ３の表示制御（変動表示）を行うものであり、プログラムＲＯＭ２２には、このＣＰＵ２１により実行されるプログラムが記憶されている。ワークＲＡＭ２３は、ＣＰＵ２１によるプログラムの実行時に使用されるワークデータが記憶されるメモリである。

【００２５】

ビデオＲＡＭ２４は、ＬＣＤディスプレイ３に表示されるデータが記憶されるメモリであり、このビデオＲＡＭ２４の内容を書き換えることにより、ＬＣＤディスプレイ３の表示内容が変更される。即ち、各表示領域における図柄の変動表示は、ビデオＲＡＭ２４の内容が書き換えられることにより行われる。キャラクタＲＯＭ２５は、ＬＣＤディスプレイ３に表示される図柄などのキャラクタデータを記憶するメモリである。画像コントローラ２６は、ＣＰＵ２１、ビデオＲＡＭ２４、入出力ポート２７のそれぞれのタイミングを調整して、データの読み書きを介在するとともに、ビデオＲＡＭ２４に記憶される表示データをキャラクタＲＯＭ２５を参照して所定のタイミングでＬＣＤディスプレイ３に表示させるものである。

【００２６】

次に、上記のように構成されたパチンコ遊技機Ｐで実行される各処理を、図３から図５のフローチャートを参照して説明する。図３は、パチンコ遊技機Ｐの制御部Ｃにおいて、２ｍｓ毎に実行されるリセット割込処理のフローチャートである。パチンコ遊技機Ｐの主な制御は、このリセット割込処理によって実行される。

【００２７】

リセット割込処理では、まず、スタックポインタを設定し（Ｓ１）、ＲＡＭ１３の所定エリアに書き込まれているパターンのチェックを行う（Ｓ２）。チェックの結果、所定エリアに所定のパターンが書き込まれていれば、ＲＡＭ１３に異常はなく正常であるので（Ｓ２：正常）、処理をＳ３へ移行する。一方、Ｓ２のチェックの結果、所定エリアに所定のパターンが書き込まれていなければ、電源投入後最初に実行されたリセット割込処理であるか、或いは、ＲＡＭ１３に異常があるので（Ｓ２：異常）、この場合には処理をＳ２３へ移行して、一旦、ＲＡＭ１３の内容をクリアした後、ＲＡＭ１３内へ初期値を書き込んで（Ｓ２３）、次のリセット割込処理の発生を待機する。

【００２８】

Ｓ３の処理ではタイマ割込の設定を行う（Ｓ３）。ここで設定されるタイマ割込としては、ＬＣＤディスプレイ３の表示を制御するコマンドを表示装置Ｄへ送信するためのストローブ信号を発生させるタイマ割込などがある。タイマ割込の設定後は、各割込を許可状態とする（Ｓ４）。割込の許可後は、特別図柄変動処理（Ｓ１６）や、表示データ作成処理（Ｓ１８）、ランプ・情報処理（Ｓ１９）などにより、前回のリセット割込処理において更新された出力データを一度に各ポートへ出力するポート出力処理を実行する（Ｓ５）。その後、後述する次初期値メモリ更新処理（Ｓ６）を実行して、次初期値メモリ１３ｃの値を更新すると共に、乱数更新処理（Ｓ７）を実行して、乱数カウンタ１３ａの値を「＋１」更新し、更に、記憶タイマ減算処理を実行する（Ｓ８）。記憶タイマ減算処理は、大当たり判定の保留球が所定数以上あり、且つ、ＬＣＤディスプレイ３において図柄の変動表示中である場合に、図柄の変動表示時間の短縮を行うものである。

【００２９】

スイッチ読込処理（Ｓ９）は、各スイッチの値を読み込むことにより、遊技領域１へ打ち込まれた打球の入賞口２や大入賞口５（Ｖゾーン５ａを含む）への入賞、図柄作動ゲート４の通過、更には賞球や貸球を検出するための処理である。カウント異常監視処理（Ｓ

１０）は、Ｓ９のスイッチ読込処理によって読み込まれたスイッチデータに異常があるか否かを監視するための処理である。例えば、大入賞口５が開放され、打球のＶゾーン５ａの通過を検出するＶカウントスイッチで打球が検出されたにも拘わらず、Ｖゾーン５ａ以外の大入賞口５への入賞を検出する１０カウントスイッチで１球の打球も検出できない場合には、１０カウントスイッチが抜き取られるなどして、１０カウントスイッチに何らかの異常が発生している。また、賞球を払い出すモータを駆動したにも拘わらず、１球の賞球も払い出されない場合には、賞球の払出装置に何らかの異常が発生している。このようにカウント異常監視処理（Ｓ１０）では、スイッチ読込処理（Ｓ９）によって読み込まれたスイッチデータに基づいて、上記のような異常の有無を監視している。

【００３０】

図柄カウンタ更新処理（Ｓ１１）では、ＬＣＤディスプレイ３で行われる変動表示の結果、停止表示される図柄を決定するためのカウンタの更新処理が行われる。また、図柄チェック処理（Ｓ１２）では、図柄カウンタ更新処理（Ｓ１１）で更新されたカウンタの値に基づいて、特別図柄変動処理（Ｓ１６）で使用される大当たり図柄や、はずれ図柄、更にはリーチ図柄などが決定される。

【００３１】

Ｓ３からＳ１２までの処理において、エラーが発生していなければ（Ｓ１３：正常）、普通図柄変動処理（Ｓ１４）によって、７セグメントＬＥＤの変動表示を行うと共に、その変動表示の結果、当たりが発生した場合には普通電動役物（図示せず）を所定時間開放する当たり処理を実行する。その後、状態フラグをチェックし（Ｓ１５）、ＬＣＤディスプレイ３の図柄の変動表示中であれば（Ｓ１５：図柄変動中）、特別図柄変動処理（Ｓ１６）によって、打球が図柄作動ゲート４を通過するタイミングで読みとられた乱数カウンタ１３ａの値に基づいて、大当たりか否かの判定が行われると共に、ＬＣＤディスプレイ３の表示図柄の変動処理を実行する。一方、状態フラグをチェックした結果、大当たり中であれば（Ｓ１５：大当り中）、大入賞口５を開放するなどの大当たり処理（Ｓ１７）を実行する。更に、状態フラグをチェックした結果、図柄の変動中でも大当たり中でもなければ（Ｓ１５：その他）、Ｓ１６及びＳ１７の処理をスキップして、Ｓ１８の表示データ作成処理へ移行する。なお、Ｓ１３の処理において、エラーが確認された場合には（Ｓ１３：エラー）、Ｓ１４～Ｓ１７の各処理をスキップして、Ｓ１８の表示データ作成処理へ移行する。

【００３２】

表示データ作成処理（Ｓ１８）では、図柄の変動表示以外にＬＣＤディスプレイ３に表示されるデモデータや、７セグメントＬＥＤの表示データなどが作成され、ランプ・情報処理（Ｓ１９）では、保留球のランプデータをはじめ、各種のランプデータが作成される。効果音処理（Ｓ２０）では、遊技の状況に応じた効果音データが作成される。なお、これらの表示データおよび効果音データは、前記したポート出力処理（Ｓ５）やタイマ割込処理によって各装置へ出力される。

【００３３】

効果音処理（Ｓ２０）の終了後は、次のリセット割込処理が発生するまでの残余時間の間、Ｓ１１と同一の処理である図柄カウンタ更新処理（Ｓ２１）と、バイトカウンタ１３ｄの「＋１」加算であるインクリメント処理（Ｓ２２）とを繰り返し実行する。Ｓ１～Ｓ２０の各処理の実行時間は遊技の状態に応じて変化するので、次のリセット割込処理が発生するまでの残余時間は、一定の時間ではなく、遊技の状態に応じて変化する。よって、かかる残余時間を使用して図柄カウンタ更新処理（Ｓ２１）を繰り返し実行することにより、停止図柄をランダムに変更することができる。

【００３４】

また、かかる残余時間を使用してバイトカウンタ１３ｄの値を繰り返し更新することにより、バイトカウンタ１３ｄの値を「ぶら下げ基板」により把握不可能とすることができる。バイトカウンタ１３ｄの値は、次初期値メモリ更新処理（Ｓ６）において、次初期値メモリ１３ｃの値に加算され、次初期値メモリ１３ｃの値を更新するものである。次初期

値メモリ１３ｃの値は、乱数カウンタ１３ａの次回の更新の初期値となる値であるので、バイトカウンタ１３ｄの値を「ぶら下げ基板」により把握不可能とすることにより、大当たりの発生タイミングを「ぶら下げ基板」で把握できなくして、「ぶら下げ基板」による不正行為を防止することができるのである。

【００３５】

図４は、次初期値メモリ更新処理のフローチャートである。次初期値メモリ更新処理（Ｓ６）では、ＣＰＵ１１の内部レジスタ１１ｂを介して、乱数カウンタ１３ａの更新の初期値を記憶する次初期値メモリ１３ｃの値を、バイトカウンタ１３ｄの値に基づいて乱数カウンタ１３ａの更新範囲の「０～６３０（０～２７６ｈ）」の範囲内で更新する。

【００３６】

まず、２バイトで構成される次初期値メモリ１３ｃの値を２バイトの内部レジスタ１１ｂへ書き込む（Ｓ４１）。次に、１バイトのバイトカウンタ１３ｄの値を内部レジスタ１１ｂへ加算し（Ｓ４２）、加算後の内部レジスタ１１ｂの値が「６３１」以上であるか否か、即ち、乱数カウンタ１３ａの更新範囲の値を超えている否かを調べる（Ｓ４３）。

【００３７】

加算後の内部レジスタ１１ｂの値が「６３１」以上であれば（Ｓ４３：Ｙｅｓ）、乱数カウンタ１３ａの更新範囲の値を超えているので、その値を次回の乱数カウンタ１３ａの初期値として、次初期値メモリ１３ｃに書き込むことはできない。よって、内部レジスタ１１ｂの値から乱数カウンタ１３ａの更新範囲の最大値＋１である「６３１（２７７ｈ）」を減算し（Ｓ４４）、内部レジスタ１１ｂの値を「０～６３０（０～２７６ｈ）」の範囲内の値とする。

【００３８】

一方、加算後の内部レジスタ１１ｂの値が「６３０」以下であれば（Ｓ４３：Ｎｏ）、乱数カウンタ１３ａの更新範囲内の値であるので、Ｓ４４の処理をスキップして、Ｓ４５の処理へ移行する。Ｓ４５の処理では、６８系ＣＰＵ１１の２バイト書き込み命令によって、２バイトの内部レジスタ１１ｂの値を、上位バイト、下位バイトの順に次初期値メモリ１３ｃへ書き込み（Ｓ４５）、次初期値メモリ１３ｃの更新を終了する。

【００３９】

図５は、乱数更新処理のフローチャートである。乱数更新処理（Ｓ７）では、ＣＰＵ１１の内部レジスタ１１ｂを介して、乱数カウンタ１３ａの値を「０～６３０（０～２７６ｈ）」の範囲内で「＋１」ずつ更新すると共に、制御部Ｃで使用される他の乱数の更新を行っている。

【００４０】

まず、２バイトで構成される乱数カウンタ１３ａの値を２バイトの内部レジスタ１１ｂへ書き込む（Ｓ５１）。内部レジスタ１１ｂの値を１加算し（Ｓ５２）、加算後の内部レジスタ１１ｂの値が「６３１」以上であるか否か、即ち、乱数カウンタ１３ａの更新範囲の値を超えている否かを調べる（Ｓ５３）。加算後の内部レジスタ１１ｂの値が「６３１」以上であれば（Ｓ５３：Ｙｅｓ）、更新範囲の値を超えているので、内部レジスタ１１ｂの値を「０」クリアする（Ｓ５４）。一方、加算後の内部レジスタ１１ｂの値が「６３０」以下であれば（Ｓ５３：Ｎｏ）、更新範囲内の値であるので、Ｓ５４の処理をスキップして、Ｓ５５の処理へ移行する。

【００４１】

Ｓ５５の処理では、更新後の内部レジスタ１１ｂの値と現初期値メモリ１３ｂの値とが比較される。現初期値メモリ１３ｂには現在更新中の乱数カウンタ１３ａの更新の初期値が記憶されているので、両値が等しい場合には（Ｓ５５：Ｙｅｓ）、乱数カウンタ１３ａの更新は一回り終了したということである。よって、かかる場合には、乱数カウンタ１３ａの次回の更新の初期値を記憶する次初期値メモリ１３ｃの値を内部レジスタ１１ｂへ書き込み（Ｓ５６）、その内部レジスタ１１ｂの値を現初期値メモリ１３ｂ及び乱数カウンタ１３ａへ書き込んで（Ｓ５７，Ｓ５８）、乱数カウンタ１３ａの更新の初期値を変更する。

【００４２】
一方、更新後の内部レジスタ１１ｂの値と現初期値メモリ１３ｂの値とが等しくない場合には（Ｓ５５：Ｎｏ）、乱数カウンタ１３ａの更新は未だ一回り終了していないので、Ｓ５６及びＳ５７の処理をスキップして、Ｓ５２からＳ５４の処理で更新された内部レジスタ１１ｂの値を乱数カウンタ１３ａへ書き込み（Ｓ５８）、乱数カウンタ１３ａの更新を行う。その後は、制御部Ｃで使用される他の乱数の更新処理を行って（Ｓ５９）、この乱数更新処理を終了する。
【００４３】
以上、図３から図５のフローチャートに基づいて説明したように、大当たりの判定に用いられる乱数カウンタ１３ａは一回りの更新毎に更新の初期値が変更されるが、その更新の初期値は、次初期値メモリ１３ｃの値にバイトカウンタ１３ｄの値を加算して算出される。バイトカウンタ１３ｄは、遊技の状態に応じて変化するリセット割込処理の残余時間の間に繰り返し更新されるので、その値を「ぶら下げ基板」で把握することはできない。よって、乱数カウンタ１３ａの値は一回りの更新毎に「ぶら下げ基板」で把握できない値から更新されるので、「ぶら下げ基板」による大当たりの発生タイミングの把握を不可能にして、「ぶら下げ基板」による不正行為を防止することができるのである。
【００４４】
また、リセット割込は割込優先順位の最も高いノンマスカブルな割込であり、命令の実行途中であっても開始される割込であるが、バイトカウンタ１３ｄを１バイトで構成しているので、バイトカウンタ１３ｄの更新中のいかなるタイミングで次のリセット割込が発生しても不具合が生じることはない。１バイトのデータ書き込みは一度に行われるので、バイトカウンタ１３ｄの値が書き込み途中の値になることはないからである。
【００４５】
次に、図６及び図７を参照して第２実施例について説明する。図６は、第２実施例のリセット割込処理のフローチャートであり、図７は、第２実施例の乱数更新処理のフローチャートである。第２実施例では、次初期値メモリ１３ｃを用いずに、乱数カウンタ１３ａの値と現初期値メモリ１３ｂの値とが一致する場合に、現在の乱数カウンタ１３ａの更新の初期値にバイトカウンタ１３ｄの値を加算して、次回の乱数カウンタ１３ａの更新の初期値を算出している。なお、前記した第１実施例と同一の部分には同一の符号を付し、その説明は省略する。
【００４６】
図６のリセット割込処理では、図３の第１実施例のリセット割込処理における次初期値メモリ更新処理（Ｓ６）を削除して、乱数更新処理（Ｓ７）を図７の乱数更新処理（Ｓ７０）に変更している。他の処理（Ｓ１～Ｓ５及びＳ８～Ｓ２３）は、図３のリセット割込処理と同一であるので、その説明は省略する。
【００４７】
図７の乱数更新処理（Ｓ７０）では、まず、まず、２バイトで構成される乱数カウンタ１３ａの値を２バイトの内部レジスタ１１ｂへ書き込み（Ｓ７１）、内部レジスタ１１ｂの値を１加算して（Ｓ７２）、乱数カウンタ１３ａの値を更新する。加算後の内部レジスタ１１ｂの値が「６３１」以上であるか否か、即ち、乱数カウンタ１３ａの更新範囲の値を超えている否かを調べ（Ｓ７３）、「６３１」以上であれば（Ｓ７３：Ｙｅｓ）、乱数カウンタ１３ａの更新範囲の値を超えているので、内部レジスタ１１ｂの値を「０」クリアする（Ｓ７４）。一方、加算後の内部レジスタ１１ｂの値が「６３０」以下であれば（Ｓ７３：Ｎｏ）、乱数カウンタ１３ａの更新範囲内の値であるので、Ｓ７４の処理をスキップして、Ｓ７５の処理へ移行する。
【００４８】
Ｓ７５の処理では、更新後の内部レジスタ１１ｂの値と現初期値メモリ１３ｂの値とが比較される。現初期値メモリ１３ｂには現在更新中の乱数カウンタ１３ａの更新の初期値が記憶されているので、両値が等しい場合には（Ｓ７５：Ｙｅｓ）、乱数カウンタ１３ａの更新は一回り終了したということである。よって、かかる場合には、乱数カウンタ１３

ａの次回の更新の初期値を算出するために、１バイトのバイトカウンタ１３ｄの値を内部レジスタ１１ｂへ加算する（Ｓ７６）。加算後の内部レジスタ１１ｂの値が「６３１」以上であるか否か、即ち、乱数カウンタ１３ａの更新範囲の値を超えている否かを調べ（Ｓ７７）、「６３１」以上であれば（Ｓ７７：Ｙｅｓ）、乱数カウンタ１３ａの更新範囲の値を超えているので、加算後の内部レジスタ１１ｂの値から乱数カウンタ１３ａの更新範囲の最大値＋１である「６３１」を減算して（Ｓ７８）、内部レジスタ１１ｂの値を乱数カウンタ１３ａの更新範囲内の値とする。一方、加算後の内部レジスタ１１ｂの値が「６３０」以下であれば（Ｓ７７：Ｎｏ）、乱数カウンタ１３ａの更新範囲内の値であるので、Ｓ７８の処理をスキップして、Ｓ７９の処理へ移行する。

【００４９】

Ｓ７９の処理では、内部レジスタ１１ｂには、Ｓ７６からＳ７８の処理で算出された次回の乱数カウンタ１３ａの更新の初期値が記憶されているので、その内部レジスタ１１ｂの値を現初期値メモリ１３ｂ及び乱数カウンタ１３ａへ書き込んで（Ｓ７９，Ｓ８０）、乱数カウンタ１３ａの更新の初期値を変更する。

【００５０】

一方、Ｓ７２からＳ７４の処理で更新された内部レジスタ１１ｂの値と現初期値メモリ１３ｂの値とが等しくない場合には（Ｓ７５：Ｎｏ）、乱数カウンタ１３ａの更新は未だ一回り終了していないので、Ｓ７６からＳ７９の処理をスキップし、Ｓ７２からＳ７４の処理で更新された内部レジスタ１１ｂの値を乱数カウンタ１３ａへ書き込み（Ｓ８０）、乱数カウンタ１３ａの更新を行う。その後は、制御部Ｃで使用される他の乱数の更新処理を行って（Ｓ８１）、この乱数更新処理を終了する。

【００５１】

このように、次初期値メモリ１３ｃを用いずに、バイトカウンタ１３ｄの値を直接使用して、乱数カウンタ１３ａの更新の初期値を変更することもできる。

【００５２】

次に、図８を参照して第３実施例について説明する。第３実施例では、次初期値メモリ１３ｃを用いないばかりか、現初期値メモリ１３ｅを１バイトで構成している。即ち、乱数カウンタ１３ａの上位バイトが所定値（第３実施例では「０」）であって、且つ、乱数カウンタ１３ａの下位バイトが１バイトの現初期値メモリ１３ｅの値と一致する場合に、バイトカウンタ１３ｄの値を乱数カウンタ１３ａの下位バイトおよび１バイトの現初期値メモリ１３ｅへ書き込んで、乱数カウンタ１３ａの更新の初期値を変更している。これにより第３実施例では、乱数カウンタ１３ａの更新の初期値は「０００ｈ～０ＦＦｈ」の範囲内で変更される。なお、前記した第２実施例と同一の部分には同一の符号を付し、その説明は省略する。また、第３実施例のリセット割込処理は、図６の第２実施例のリセット割込処理が使用される。

【００５３】

図８の乱数更新処理（Ｓ７０）では、まず、まず、２バイトで構成される乱数カウンタ１３ａの値を２バイトの内部レジスタ１１ｂへ書き込み（Ｓ９１）、内部レジスタ１１ｂの値を１加算して（Ｓ９２）、乱数カウンタ１３ａの値を更新する。加算後の内部レジスタ１１ｂの値が「６３１」以上であるか否か、即ち、乱数カウンタ１３ａの更新範囲の値を超えている否かを調べ（Ｓ９３）、「６３１」以上であれば（Ｓ９３：Ｙｅｓ）、乱数カウンタ１３ａの更新範囲の値を超えているので、内部レジスタ１１ｂの値を「０」クリアする（Ｓ９４）。一方、加算後の内部レジスタ１１ｂの値が「６３０」以下であれば（Ｓ９３：Ｎｏ）、乱数カウンタ１３ａの更新範囲内の値であるので、Ｓ９４の処理をスキップして、Ｓ９５の処理へ移行する。

【００５４】

Ｓ９５の処理では、更新後の内部レジスタ１１ｂの上位バイトの値が「０」であるか否かを調べ（Ｓ９５）、「０」であれば（Ｓ９５：Ｙｅｓ）、更に、内部レジスタ１１ｂの下位バイトの値と１バイトの現初期値メモリ１３ｅの値とを比較する（Ｓ９６）。現初期値メモリ１３ｅには現在更新中の乱数カウンタ１３ａの更新の初期値の下位バイトが記憶

されているので、両値が等しい場合には（Ｓ９６：Ｙｅｓ）、乱数カウンタ１３ａの更新は一回り終了したということである。よって、かかる場合には、乱数カウンタ１３ａの更新の初期値を変更するために、１バイトのバイトカウンタ１３ｄの値を２バイトの内部レジスタ１１ｂの下位バイトへ書き込み（Ｓ９７）、内部レジスタ１１ｂの値を乱数カウンタ１３ａの次回の更新の初期値に変更する。この内部レジスタ１１ｂの下位バイトの値を１バイトの現初期値メモリ１３ｅへ書き込んで（Ｓ９８）、現初期値メモリ１３ｅの値を更新し、更に、内部レジスタ１１ｂの上位バイト及び下位バイトの値を２バイトの乱数カウンタ１３ａへ書き込んで（Ｓ９９）、乱数カウンタ１３ａの更新の初期値を変更する。

【００５５】

一方、Ｓ９５の処理において、Ｓ９２からＳ９４の処理で更新された内部レジスタ１１ｂの上位バイトの値が「０」でない場合や（Ｓ９５：Ｎｏ）、更新後の内部レジスタ１１ｂの上位バイトの値が「０」であっても（Ｓ９５：Ｙｅｓ）、内部レジスタ１１ｂの下位バイトの値と１バイトの現初期値メモリ１３ｅの値とが等しくない場合には（Ｓ９６：Ｎｏ）、乱数カウンタ１３ａの更新は未だ一回り終了していない。よって、この場合には、Ｓ９７及びＳ９８の処理をスキップし、Ｓ９２からＳ９４の処理で更新された内部レジスタ１１ｂの値を乱数カウンタ１３ａへ書き込み（Ｓ９９）、乱数カウンタ１３ａの更新を行う。

【００５６】

Ｓ９９の処理の終了後は、制御部Ｃで使用される他の乱数の更新処理を行って（Ｓ１００）、この乱数更新処理を終了する。このように、次初期値メモリ１３ｃを用いず、現初期値メモリ１３ｅを１バイトで構成し、且つ、バイトカウンタ１３ｄの値を直接使用して、乱数カウンタ１３ａの更新の初期値を変更することもできる。この場合、２バイトの乱数カウンタ１３ａの値と１バイトの現初期値メモリ１３ｅの値との一致の比較は、乱数カウンタ１３ａの上位バイトが所定値である時に（第３実施例では「０」）、乱数カウンタ１３ａの下位バイトと現初期値メモリ１３ｅの値とが一致するか否かにより行われる。

【００５７】

なお、Ｓ９５の処理では、Ｓ９２からＳ９４の処理で更新された内部レジスタ１１ｂの上位バイトの値が「０」と比較されるので、乱数カウンタ１３ａの更新の初期値は「０００ｈ～０ＦＦｈ」の範囲内で変更される。よって、Ｓ９５の処理で、更新後の内部レジスタ１１ｂの上位バイトの値が「１」と比較される場合には、乱数カウンタ１３ａの更新の初期値は「１００ｈ～１ＦＦｈ」の範囲で変更される。また、「２」と比較される場合には、乱数カウンタ１３ａの更新の初期値は「２００ｈ～２７６ｈ」の範囲で変更されるが、この場合には、Ｓ９７の処理で内部レジスタ１１ｂの下位バイトへ書き込まれるバイトカウンタ１３ｄの値が「７７ｈ～０ＦＦｈ」であれば、その値を「００ｈ～７６ｈ」の範囲内に補正する処理が必要となる。

【００５８】

次に、図９を参照して第４実施例について説明する。図９は、第４実施例のリセット割込処理のフローチャートである。第４実施例では、バイトカウンタ１３ｄの値を「０」クリアした後に、リセット割込処理の残余時間を使用して、そのバイトカウンタ１３ｄの値を繰り返し更新するものである。なお、前記した第１実施例と同一の部分には同一の符号を付し、その説明は省略する。

【００５９】

図９に示すように、効果音処理（Ｓ２０）の終了後、バイトカウンタ１３ｄの値を「０」クリアし（Ｓ１０１）、次のリセット割込処理が発生するまでの残余時間の間、図柄カウンタ更新処理（Ｓ２１）と、バイトカウンタ１３ｄの「＋１」加算であるインクリメント処理（Ｓ２２）とを繰り返し実行する。

【００６０】

１回のリセット割込処理の残余時間の間にＳ２２の処理によってバイトカウンタ１３ｄの値が一回り以上更新されない限りにおいて、上記のようにＳ２２の処理の前にバイトカウンタ１３ｄの値を「０」クリアすることによって、次初期値メモリ１３ｃの値をリセッ

ト割込処理の残余時間の間に繰り返し更新することと同様の効果を得ることができる。即ち、第４実施例の構成によれば、次初期値メモリ１３ｃの値を連続的に更新することができるのである。

【００６１】

なお、Ｓ１０１の処理に代えて、図４の次初期値メモリ更新処理（Ｓ６）においてバイトカウンタ１３ｄの値を「０」クリアするようにしても良い。具体的には、Ｓ４５の処理の実行後に、バイトカウンタ１３ｄの値を「０」クリアすれば良い。

【００６２】

また、第２及び３実施例の図６のリセット割込処理においても、図９のリセット割込処理と同様に、Ｓ２０の効果音処理の直後であってＳ２１及びＳ２２の繰り返し処理の前に、Ｓ１０１の処理に相当するバイトカウンタ１３ｄの「０」クリア処理を追加したり、或いは、図７の乱数更新処理におけるＳ８０の処理の直後、又は、図８の乱数更新処理におけるＳ９９の処理の直後に、Ｓ１０１の処理に相当するバイトカウンタ１３ｄの「０」クリア処理を追加するようにしても良い。

【００６３】

上記各実施例において、請求項１記載の割込処理としてはノンマスカブルなリセット割込処理が該当し、第１更新手段としては、図５の乱数更新処理（Ｓ７）のＳ５１からＳ５４及びＳ５８の処理、図７の乱数更新処理（Ｓ７０）のＳ７１からＳ７４及びＳ８０の処理、図８の乱数更新処理（Ｓ７０）のＳ９１からＳ９４及びＳ９９の処理がそれぞれ該当する。また、第２更新手段としては図３及び図６のＳ２２の処理が該当し、変更手段としては、図５の乱数更新処理（Ｓ７）のＳ５５からＳ５８の処理、図７の乱数更新処理（Ｓ７０）のＳ７５からＳ８０の処理、図８の乱数更新処理（Ｓ７０）のＳ９５からＳ９９の処理がそれぞれ該当する。

【００６４】

以上、実施例に基づき本発明を説明したが、本発明は上記実施例に何ら限定されるものではなく、本発明の趣旨を逸脱しない範囲内で種々の改良変形が可能であることは容易に推察できるものである。

【００６５】

例えば、上記各実施例では、バイトカウンタ１３ｄの値の更新は、インクリメントにより加算方向に行われたが、必ずしもこれに限らず、デクリメントなどによって減算方向に行うようにしても良い。また、第１実施例における次初期値メモリ１３ｃの値の更新や（Ｓ４２～Ｓ４４）、第２実施例における現初期値メモリ１３ｂの値の更新（Ｓ７６～Ｓ７９）は、バイトカウンタ１３ｄの値を加算することにより行われたが、これらをバイトカウンタ１３ｄの値を減算することにより行うようにしても良い。

【００６６】

また、上記各実施例において、バイトカウンタ１３ｄはいずれも１バイト（８ビット）で構成されたが、バイトカウンタ１３ｄのサイズは必ずしも８ビットに固定されるものではない。例えば、ＣＰＵが１６ビットのデータを一度に更新可能であればバイトカウンタを１６ビットで構成しても良いし、ＣＰＵが３２ビットのデータを一度に更新可能であればバイトカウンタを３２ビットで構成しても良い。即ち、バイトカウンタのサイズは、ＣＰＵが一度に更新可能な最大ビット数の範囲内で設定することができる。

【００６７】

以下に本発明の変形例を示す。請求項１記載の弾球遊技機において、前記変更手段は、前記乱数カウンタの次回の更新の初期値を記憶する少なくとも２バイトで構成された次初期値メモリと、その次初期値メモリの値を前記バイトカウンタの値に基づいて更新する第３更新手段と、前記現初期値メモリの値と前記乱数カウンタの値とが一致する場合に、前記第３更新手段によって更新された次初期値メモリの値を前記乱数カウンタおよび現初期値メモリに書き込む書込手段とを備えていることを特徴とする弾球遊技機１。

【００６８】

弾球遊技機１において、前記第３更新手段は前記第２更新手段の実行前に実行されるこ

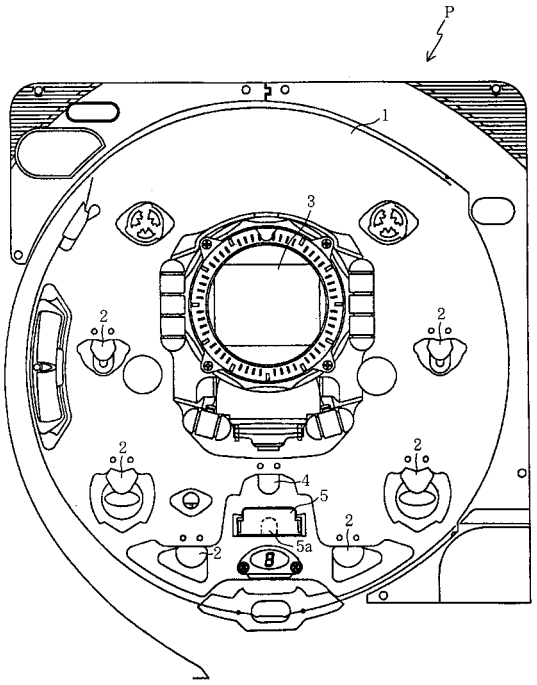
とを特徴とする弾球遊技機２。
【００６９】
請求項１記載の弾球遊技機において、前記変更手段は前記バイトカウンタの値を前記乱数カウンタおよび現初期値メモリの値に加算または減算することによりそれらの値を変更することを特徴とする弾球遊技機３。
【００７０】
請求項１記載の弾球遊技機において、前記現初期値メモリは２バイトで構成されると共に、前記変更手段は１バイトの前記バイトカウンタの値を前記乱数カウンタおよび現初期値メモリの下位バイトに書き込んでそれらの値を変更することを特徴とする弾球遊技機４。
【００７１】
請求項１記載の弾球遊技機において、前記現初期値メモリは１バイトで構成されると共に、前記変更手段は、その１バイトの現初期値メモリと２バイトの前記乱数カウンタとの一致の判断を、その乱数カウンタの上位バイトが所定値（例えば「０」）である場合に、その乱数カウンタの下位バイトの値と前記現初期値メモリの値とが一致するか否かにより行う比較手段と、その比較手段により一致すると判断された場合に、１バイトの前記バイトカウンタの値を２バイトの前記乱数カウンタの下位バイトおよび１バイトの前記現初期値メモリへ書き込む書き込み手段とを備えていることを特徴とする弾球遊技機５。
【００７２】
請求項１記載の弾球遊技機において、前記第２更新手段の実行前に前記バイトカウンタの値をリセットするリセット手段を備えていることを特徴とする弾球遊技機６。
【００７３】
【発明の効果】
本発明の弾球遊技機によれば、大当たりを決定するための乱数カウンタの更新の初期値は、固定値ではなく、定期的に変更される値である。よって、弾球遊技機の電源投入に合わせて、「ぶら下げ基板」等がその内部の不正なカウンタをリセットしても、そのカウンタの値を乱数カウンタの値と一致させることはできない。従って、「ぶら下げ基板」等による大当たりの発生タイミングの把握を不可能にして、「ぶら下げ基板」等による不正行為を防止することができるという効果がある。
【００７４】
しかも、乱数カウンタの更新の初期値の元になるバイトカウンタの値は、定期的に実行される割込処理の残余時間の間に繰り返し更新される。割込処理の残余時間の長さは、遊技の状態に応じて変化するので、「ぶら下げ基板」等で把握することはできない。よって、かかる残余時間の間にバイトカウンタの値を繰り返し更新しているので、この点においても「ぶら下げ基板」等による乱数カウンタの値の把握を防止することができるという効果がある。
【００７５】
また、バイトカウンタは１バイトで構成されるので、割込処理がいかなるタイミングで発生しても、書き込み途中の値が更新後のバイトカウンタの値となることはない。よって、書き込み途中の値が更新後のバイトカウンタの値となることにより生じる不具合を回避できるという効果がある。
【図面の簡単な説明】
【図１】　本発明の第１実施例であるパチンコ遊技機の遊技盤の正面図である。
【図２】　パチンコ遊技機の電気的構成を示したブロック図である。
【図３】　リセット割込処理を示したフローチャートである。
【図４】　次初期値メモリ更新処理を示したフローチャートである。
【図５】　乱数更新処理を示したフローチャートである。
【図６】　第２実施例のリセット割込処理を示したフローチャートである。
【図７】　第２実施例の乱数更新処理を示したフローチャートである。
【図８】　第３実施例の乱数更新処理を示したフローチャートである。

【図９】　第４実施例のリセット割込処理を示したフローチャートである。

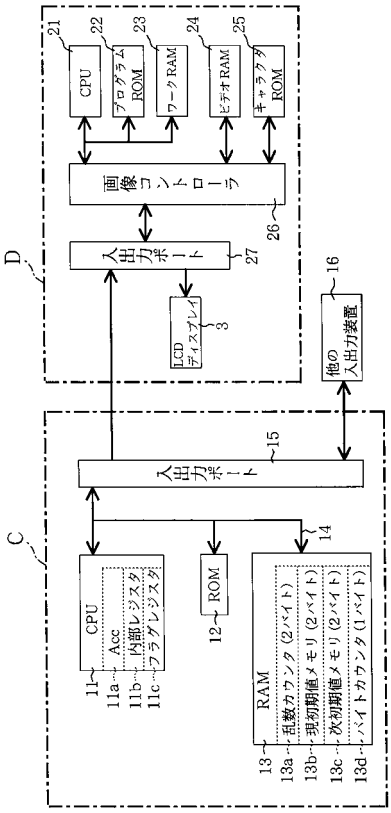
【符号の説明】

１１	制御部のＣＰＵ
１１ｂ	制御部のＣＰＵの内部レジスタ
１３	制御部のＲＡＭ
１３ａ	乱数カウンタ
１３ｂ，１３ｅ	現初期値メモリ
１３ｃ	次初期値メモリ
１３ｄ	バイトカウンタ
Ｃ	制御部
Ｐ	パチンコ遊技機（弾球遊技機）

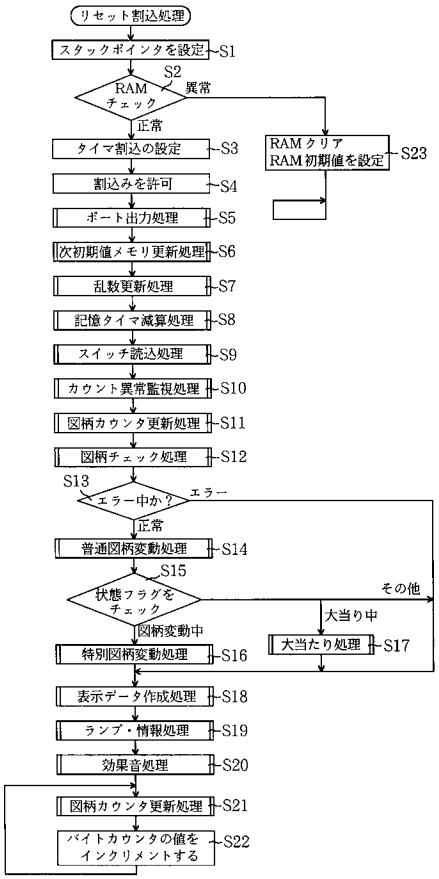
【図１】



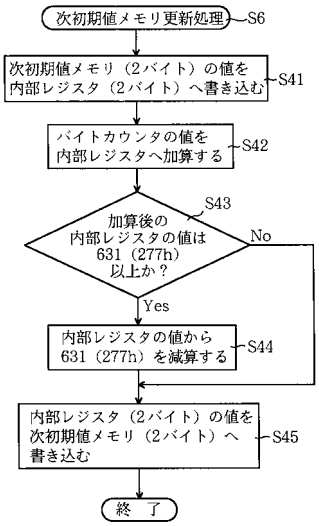
【図２】



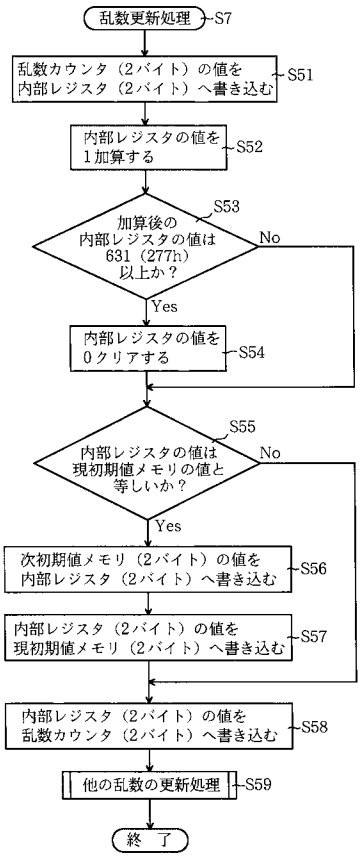
【図３】


リセット割込処理
スタックポインタを設定 S1
S2
RAM チェック
異常
正常
RAMクリア RAM初期値を設定 S23
タイマ割込の設定 S3
割込みを許可 S4
ポート出力処理 S5
次初期値メモリ更新処理 S6
乱数更新処理 S7
記憶タイマ減算処理 S8
スイッチ読込処理 S9
カウント異常監視処理 S10
図柄カウンタ更新処理 S11
図柄チェック処理 S12
S13
エラー中か？
エラー
正常
普通図柄変動処理 S14
S15
状態フラグをチェック
その他
大当り中
図柄変動中
大当たり処理 S17
特別図柄変動処理 S16
表示データ作成処理 S18
ランプ・情報処理 S19
効果音処理 S20
図柄カウンタ更新処理 S21
バイトカウンタの値をインクリメントする S22

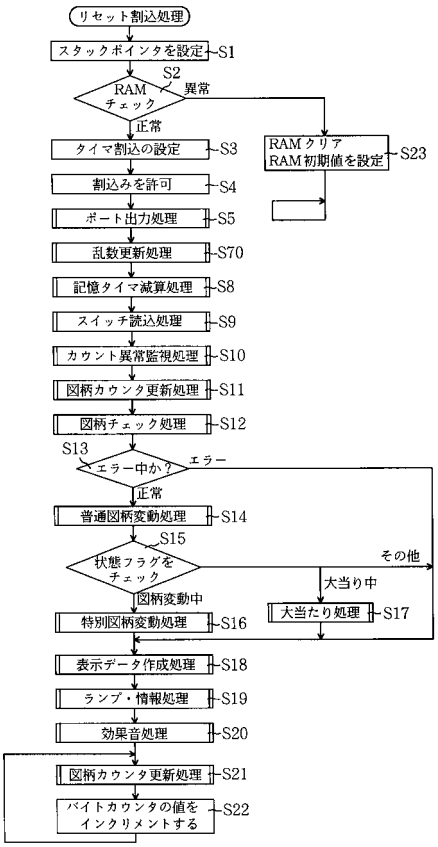
【図４】


次初期値メモリ更新処理 S6
次初期値メモリ（2バイト）の値を内部レジスタ（2バイト）へ書き込む S41
バイトカウンタの値を内部レジスタへ加算する S42
S43
加算後の内部レジスタの値は631（277h）以上か？
No
Yes
内部レジスタの値から631（277h）を減算する S44
内部レジスタ（2バイト）の値を次初期値メモリ（2バイト）へ書き込む S45
終 了

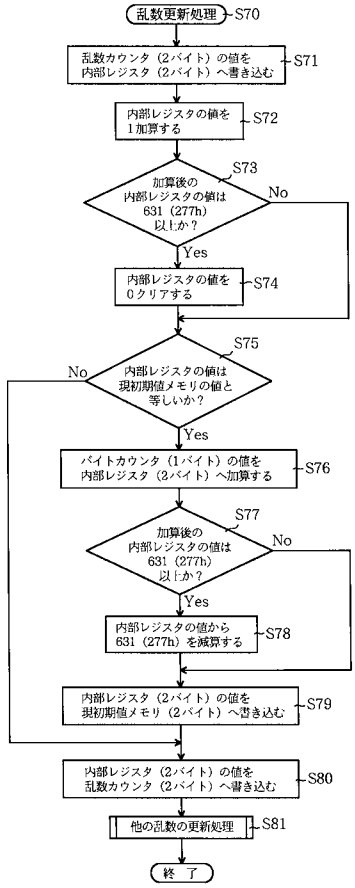
【図５】


乱数更新処理 S7
乱数カウンタ（2バイト）の値を内部レジスタ（2バイト）へ書き込む S51
内部レジスタの値を1加算する S52
S53
加算後の内部レジスタの値は631（277h）以上か？
No
Yes
内部レジスタの値を0クリアする S54
S55
内部レジスタの値は現初期値メモリの値と等しいか？
No
Yes
次初期値メモリ（2バイト）の値を内部レジスタ（2バイト）へ書き込む S56
内部レジスタ（2バイト）の値を現初期値メモリ（2バイト）へ書き込む S57
内部レジスタ（2バイト）の値を乱数カウンタ（2バイト）へ書き込む S58
他の乱数の更新処理 S59
終 了

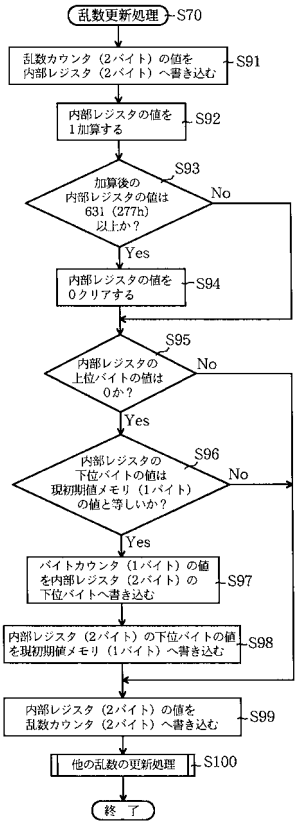
【図６】


リセット割込処理
スタックポインタを設定 S1
S2
RAM チェック
異常
正常
RAMクリア RAM初期値を設定 S23
タイマ割込の設定 S3
割込みを許可 S4
ポート出力処理 S5
乱数更新処理 S70
記憶タイマ減算処理 S8
スイッチ読込処理 S9
カウント異常監視処理 S10
図柄カウンタ更新処理 S11
図柄チェック処理 S12
S13
エラー中か？
エラー
正常
普通図柄変動処理 S14
S15
状態フラグをチェック
その他
大当り中
図柄変動中
大当たり処理 S17
特別図柄変動処理 S16
表示データ作成処理 S18
ランプ・情報処理 S19
効果音処理 S20
図柄カウンタ更新処理 S21
バイトカウンタの値をインクリメントする S22

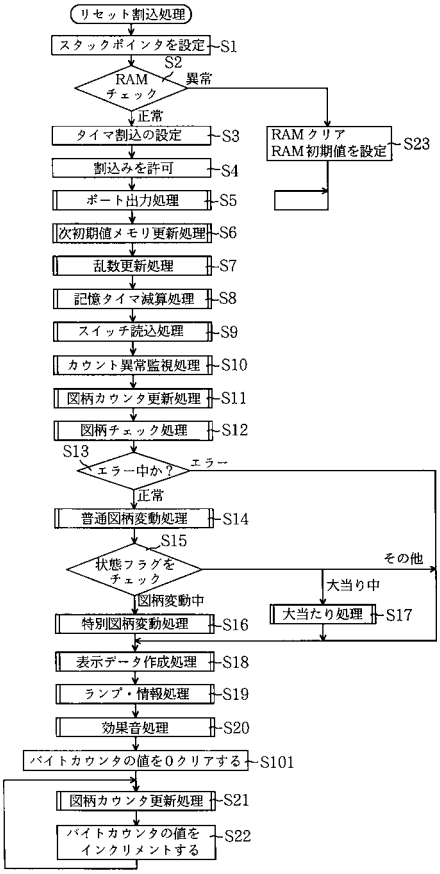
【図７】


乱数更新処理 S70
乱数カウンタ（2バイト）の値を内部レジスタ（2バイト）へ書き込む S71
内部レジスタの値を1加算する S72
S73
加算後の内部レジスタの値は631（277h）以上か？
No
Yes
内部レジスタの値を0クリアする S74
S75
No
内部レジスタの値は現初期値メモリの値と等しいか？
Yes
バイトカウンタ（1バイト）の値を内部レジスタ（2バイト）へ加算する S76
S77
加算後の内部レジスタの値は631（277h）以上か？
No
Yes
内部レジスタの値から631（277h）を減算する S78
内部レジスタ（2バイト）の値を現初期値メモリ（2バイト）へ書き込む S79
内部レジスタ（2バイト）の値を乱数カウンタ（2バイト）へ書き込む S80
他の乱数の更新処理 S81
終了

【図８】


乱数更新処理 S70
乱数カウンタ（2バイト）の値を内部レジスタ（2バイト）へ書き込む S91
内部レジスタの値を1加算する S92
S93
加算後の内部レジスタの値は631（277h）以上か？
No
Yes
内部レジスタの値を0クリアする S94
S95
内部レジスタの上位バイトの値は0か？
No
Yes
S96
内部レジスタの下位バイトの値は現初期値メモリ（1バイト）の値と等しいか？
No
Yes
バイトカウンタ（1バイト）の値を内部レジスタ（2バイト）の下位バイトへ書き込む S97
内部レジスタ（2バイト）の下位バイトの値を現初期値メモリ（1バイト）へ書き込む S98
内部レジスタ（2バイト）の値を乱数カウンタ（2バイト）へ書き込む S99
他の乱数の更新処理 S100
終了

【図９】


リセット割込処理
スタックポインタを設定 S1
S2
RAMチェック
異常
正常
RAMクリア RAM初期値を設定 S23
タイマ割込の設定 S3
割込みを許可 S4
ポート出力処理 S5
次初期値メモリ更新処理 S6
乱数更新処理 S7
記憶タイマ減算処理 S8
スイッチ読込処理 S9
カウント異常監視処理 S10
図柄カウンタ更新処理 S11
図柄チェック処理 S12
S13
エラー中か？
エラー
正常
普通図柄変動処理 S14
S15
状態フラグをチェック
その他
大当り中
図柄変動中
特別図柄変動処理 S16
大当たり処理 S17
表示データ作成処理 S18
ランプ・情報処理 S19
効果音処理 S20
バイトカウンタの値を0クリアする S101
図柄カウンタ更新処理 S21
バイトカウンタの値をインクリメントする S22

フロントページの続き

(72)発明者　那須　隆
名古屋市千種区春岡通７丁目４９番地　株式会社　ジェイ・ティ内
(72)発明者　山崎　好男
名古屋市千種区春岡通７丁目４９番地　株式会社　ジェイ・ティ内
(72)発明者　佐藤　秀昭
名古屋市千種区春岡通７丁目４９番地　株式会社　ジェイ・ティ内

審査官　大浜　康夫

(56)参考文献　特開平１１－０５７１８３（ＪＰ，Ａ）
特開平１０－００５４００（ＪＰ，Ａ）
特開平１０－００５４０２（ＪＰ，Ａ）
特開平１１－３４７２２７（ＪＰ，Ａ）
特開平０７－１００２４６（ＪＰ，Ａ）
特開平１１－０５７１７０（ＪＰ，Ａ）
特開２０００－１０７３８７（ＪＰ，Ａ）
特開２０００－０７９２５３（ＪＰ，Ａ）
特開２０００－１８９６０７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
A63F　7/02