

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2015-107188

(P2015-107188A)

(43) 公開日 平成27年6月11日 (2015.6.11)

(51) Int.Cl.	F 1	テーマコード (参考)
A 6 3 F 5/04 (2006.01)	A 6 3 F 5/04 5 1 2 D	2 C 0 8 2
	A 6 3 F 5/04 5 1 2 C	

審査請求 未請求 請求項の数 1 O L (全 96 頁)

(21) 出願番号	特願2013-250649 (P2013-250649)	(71) 出願人	390031783
(22) 出願日	平成25年12月3日 (2013.12.3)		サミー株式会社
			東京都豊島区東池袋三丁目1番1号 サン
			シャイン60
		(74) 代理人	100113228
			弁理士 中村 正
		(72) 発明者	藤生 直樹
			東京都豊島区東池袋三丁目1番1号サンシ
			シャイン60 サミー株式会社内
		(72) 発明者	瀬藤 潤
			東京都豊島区東池袋三丁目1番1号サンシ
			シャイン60 サミー株式会社内
		(72) 発明者	矢嶋 悠紀
			東京都豊島区東池袋三丁目1番1号サンシ
			シャイン60 サミー株式会社内
			最終頁に続く

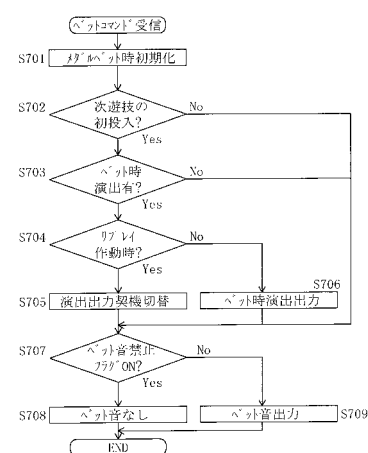
(54) 【発明の名称】 遊技機

(57) 【要約】

【課題】遊技媒体のベットを契機として特定演出を出力する場合に、リプレイの入賞時であっても、遊技者の操作に基づいて特定演出を出力する。

【解決手段】サブ制御手段は、ベットを契機として特定演出を出力する場合を有し、メイン制御手段からベットを示すコマンドデータを受信したときは、遊技者の投入操作に基づくベットであるか、リプレイの入賞に基づく自動ベットであるかを判別する。特定演出がセットされている場合において、リプレイの入賞に基づく自動ベット時にはその自動ベットを契機として特定演出を出力せず、次遊技でのスタートスイッチの操作時に特定演出を出力するように制御する (ステップ S 7 0 5)。

【選択図】 図 4 5



【特許請求の範囲】**【請求項 1】**

遊技者が遊技媒体を投入するときの遊技媒体の入口となる遊技媒体投入口と、
前記遊技媒体投入口から投入された遊技媒体を検知する検知手段と、
所定数の範囲内において遊技媒体を貯留する貯留手段と、
前記遊技媒体投入口から投入された遊技媒体、又は前記貯留手段に貯留された遊技媒体
を、その遊技における限界数の範囲内においてベットするベット手段と、
リプレイが入賞したときに、その遊技でベットされていた数の遊技媒体を自動でベット
する自動ベット手段と、
遊技を開始するときに遊技者が操作するスタートスイッチと、
遊技の進行を制御するメイン制御手段と、
前記メイン制御手段から送信されるコマンドに基づいて、演出に係る制御を行うサブ制
御手段と
を備え、
前記メイン制御手段は、
遊技者の投入操作に基づく遊技媒体のベット時と、リプレイの入賞に基づく遊技媒体の
自動ベット時とで、ベットされる遊技媒体数に応じたコマンドデータを前記サブ制御手段
に送信し、
前記サブ制御手段は、
遊技媒体のベットを契機として特定演出を出力する場合を有し、
前記メイン制御手段から遊技媒体のベットを示すコマンドデータを受信したときは、遊
技者の投入操作に基づく遊技媒体のベットであるか、リプレイの入賞に基づく遊技媒体の
自動ベットであるかを判別し、
前記特定演出がセットされている場合において、リプレイの入賞に基づく遊技媒体の自
動ベット時にはその自動ベットを契機として前記特定演出を出力せずに、次遊技での前記
スタートスイッチの操作時に前記特定演出を出力するように制御する
ことを特徴とする遊技機。

10

20

【発明の詳細な説明】**【技術分野】****【0001】**

30

本発明は、遊技媒体（メダル等）のベットを契機として特定演出を出力する遊技機（ス
ロットマシン等）に関するものである。

【背景技術】**【0002】**

従来のスロットマシンにおいて、リプレイが入賞したときは、当該遊技でベットしてい
た枚数のメダルが自動ベットされ、いわゆる再遊技が可能となる。

ここで、従来の一般的なスロットマシンにおいて、リプレイの入賞によってメダルが自
動ベットされた状態では、メダルの手入れ投入を受け付けていなかった。このため、遊技
者がメダルを手入れしたとしても、払出し口から返却され、メダルを貯留することができ
なかった。

40

なお、リプレイの入賞により自動ベットされた状態で、メダルを投入可能とする技術思
想について開示された文献も存在する（たとえば、特許文献 1 参照）。

【先行技術文献】**【特許文献】****【0003】**

【特許文献 1】特開 2003-325749 号公報

【発明の概要】**【発明が解決しようとする課題】****【0004】**

前述の従来の技術において、スロットマシンから出力する演出のうち、複数遊技にわた

50

って一連の演出を出力する連続演出が知られている。連続演出では、たとえば、当該遊技の終了時においても演出が継続しており、メダルが次遊技のためにベットされると、演出を発展させるものが知られている。

ここで、次遊技のためにメダルがベットされたタイミングで演出を発展させる連続演出が選択されている場合に、遊技者が自らメダルを手入れするか、又はベットスイッチを操作したタイミングで、連続演出を発展させることができる。しかし、当該遊技でリプレイが入賞すると、リプレイの入賞に基づき自動ベットされるので、この自動ベット時に演出が発展することとなり、演出効果を損ねるおそれがあった。

【0005】

したがって、本発明が解決しようとする課題は、遊技媒体のベットを契機として特定演出を出力する場合に、リプレイの入賞時であっても、遊技者の操作に基づいて特定演出を出力することである。

【課題を解決するための手段】

【0006】

本発明は、以下の解決手段によって、上述の課題を解決する。なお、かっこ書きにて、対応する実施形態を示す。

請求項1の発明は、

遊技者が遊技媒体（メダル）を投入するときの遊技媒体の入口となる遊技媒体投入口（メダル投入口43）と、

前記遊技媒体投入口から投入された遊技媒体を検知する検知手段（メダルセクタ）と

所定数の範囲内において遊技媒体を貯留する貯留手段（メモリ53、貯留枚数表示装置47b）と、

前記遊技媒体投入口から投入された遊技媒体、又は前記貯留手段に貯留された遊技媒体を、その遊技における限界数の範囲内においてベットするベット手段（メダル投入口43、ベットスイッチ40）と、

リプレイが入賞したときに、その遊技でベットされていた数の遊技媒体を自動でベットする自動ベット手段（メインCPU54）と、

遊技を開始するときに遊技者が操作するスタートスイッチ（41）と、

遊技の進行を制御するメイン制御手段（メイン制御基板50）と、

前記メイン制御手段から送信されるコマンドに基づいて、演出に係る制御を行うサブ制御手段（サブ制御基板80）と

を備え、

前記メイン制御手段は、

遊技者の投入操作に基づく遊技媒体のベット時と、リプレイの入賞に基づく遊技媒体の自動ベット時とで、ベットされる遊技媒体数に応じたコマンドデータ（制御コマンドデータ）を前記サブ制御手段に送信し、

前記サブ制御手段は、

遊技媒体のベットを契機として特定演出（連続演出）を出力する場合を有し、

前記メイン制御手段から遊技媒体のベットを示すコマンドデータを受信したときは、遊技者の投入操作に基づく遊技媒体のベットであるか、リプレイの入賞に基づく遊技媒体の自動ベットであるかを判別し、

前記特定演出がセットされている場合において、リプレイの入賞に基づく遊技媒体の自動ベット時にはその自動ベットを契機として前記特定演出を出力せずに、次遊技での前記スタートスイッチの操作時に前記特定演出を出力するように制御する（ステップS705）

ことを特徴とする。

【0007】

本発明においては、メイン制御手段から送信されてくるベットに関するコマンドデータをサブ制御手段で受信する。サブ制御手段は、このコマンドデータに基づいて演出を出力

10

20

30

40

50

するように制御する。

ここで、遊技媒体のベットを契機として特定演出を出力する場合において、リプレイの入賞に基づき遊技媒体が自動ベットされたときは、その自動ベットを契機として特定演出を出力せずに、スタートスイッチの操作時に出力するように、特定演出の出力タイミングを変更する。

【発明の効果】

【0008】

本発明によれば、遊技媒体のベットを契機として特定演出を出力する場合において、リプレイの入賞に基づき遊技媒体が自動ベットされたときは、その自動ベットを契機として特定演出を出力しないようにし、他の契機すなわちスタートスイッチの操作時に特定演出を出力する。これにより、特定演出の出力契機を、リプレイの入賞時であっても遊技者の操作に基づく契機に変更することができる。

【図面の簡単な説明】

【0009】

【図1】本実施形態におけるスロットマシン（遊技機）の制御の概略を示すブロック図である。

【図2】リールの図柄配列を示す図である。

【図3】表示窓とリールとの関係、及び有効ラインを示す図である。

【図4】役の種類、払出し枚数等、及び図柄の組合せを示す図である。

【図5】抽選される役の種類と当選確率とを示す図である。

【図6】ストップスイッチの押し順と入賞役との関係を示す図である。

【図7】メイン制御基板の入力ポート及び出力ポートと、各周辺機器の信号の入出力との関係を示す図である。

【図8】メモリに記憶されるデータの一部を示す図である。

【図9】立ち上がりデータの生成の第1例（正常データ取得時）を示す図である。

【図10】立ち上がりデータの生成の第2例（異常データ取得時）を示す図である。

【図11】メモリに記憶されるデータの他の一部を示す図である。

【図12】本実施形態におけるメインループ（M_MAIN）処理を示すフローチャートである。

【図13】図12のステップS102における遊技開始セット（MS_GAME_SET）を示すフローチャートである。

【図14】図13のステップS121における遊技状態セット（MS_ACTION_SET）を示すフローチャートである。

【図15】図13のステップS122における遊技状態出力（MS_STATUS_OUT）を示すフローチャートである。

【図16】貯留枚数読み込み（S_CREDIT_READ）を示すフローチャートである。

【図17】図13のステップS127におけるブロックオン（MS_BLOCKER_ON）を示すフローチャートである。

【図18】定義データの種類と内容を示す図である。

【図19】制御コマンドセット1（S_CMD_SET）を示すフローチャートである。

【図20】図19のステップS503における制御コマンドセット2（SS_CMD_SET）を示すフローチャートである。

【図21】図20のステップS516における指定アドレスデータセット（S_ADDR_SET）を示すフローチャートである。

【図22】図13のステップS133におけるメダル1枚のベット処理（MS_MEDAL_INC）を示すフローチャートである。

【図23】ベットメダルの読み込み処理（S_PLAYM_READ）を示すフローチャートである。

【図24】メダル限界枚数のセット処理（MS_MMAX_SET）を示すフローチャートである。

【図25】図12のステップS106におけるメダル管理処理（MS_MEDAL_CHK）を示すフローチャートである。

10

20

30

40

50

【図 26】図 25 のステップ S 2 2 2 における手入れメダルチェック処理 (MS_INSERT_CHK) を示すフローチャートである。

【図 27】図 26 に続くフローチャートである。

【図 28】図 27 のステップ S 2 5 7 におけるブロックオフ (MS_BLOCKER_OFF) を示すフローチャートである。

【図 29】ブロックのオン／オフと、投入センサの信号のオン／オフと、エラー監視とのタイミングを示すタイムチャートである。

【図 30】図 25 におけるステップ S 2 2 3 の貯留ベット処理 (MS_BET_IN) を示すフローチャートである。

【図 31】クレジット減算処理 (MS_CREDIT_DEC) を示すフローチャートである。

10

【図 32】図 25 のステップ S 2 2 1 における精算処理 (MS_MEDAL_RET) を示すフローチャートである。

【図 33】図 32 のステップ S 3 2 5 における貯留メダルの精算処理 (MS_CREDIT_RET) を示すフローチャートである。

【図 34】メダル 1 枚の払出し処理 (MS_1MEDAL_PAY) を示すフローチャートである。

【図 35】図 34 に続くフローチャートである。

【図 36】割込み処理を示すフローチャートである。

【図 37】図 36 のステップ S 6 0 5 における電断処理を示すフローチャートである。

【図 38】サブ制御基板での制御コマンド受信時の処理を説明する図である。

【図 39】受信した制御コマンドデータと、処理関数及び引数を定めたデータテーブルを示す図である。

20

【図 40】遊技システム全体の概要を説明する図である。

【図 41】マイスロ遊技において、新規会員登録時の操作内容を説明する図である。

【図 42】マイスロ遊技において、遊技開始時の操作内容を説明する図である。

【図 43】マイスロ遊技において、遊技終了時の操作内容を説明する図である。

【図 44】サブ制御基板のメモリに設けられたフラグの種類を示す図である。

【図 45】サブ CPU において、ベットコマンド受信時の処理を示すフローチャートである。

【図 46】サブ CPU において、精算コマンド受信時の処理を示すフローチャートである。

30

【図 47】図 46 に続くフローチャートである。

【図 48】二次元コードのバックアップ処理を示すフローチャートである。

【図 49】メニュー表示禁止フラグをオフにする処理を示すフローチャートである。

【発明を実施するための形態】

【0010】

本明細書において、用語の意味は、以下の通りである。

「遊技媒体」とは、遊技の用に供する媒体をいい、本実施形態では「メダル」である。ただしこれに限らず、遊技球を使用することも可能である。また、遊技媒体には、実際のメダルの他に、遊技機内部に電氣的に貯留（クレジット、記憶）された遊技媒体（遊技媒体に係るデータ）も含まれる。

40

「ベット」とは、遊技を行うためにメダル（遊技媒体）を賭けることをいう。本実施形態において、ベット可能な最大（限界）枚数は、通常遊技では「3」枚、MB 遊技中では「2」枚に設定されている。

【0011】

「貯留」とは、上記「ベット」とは異なり、スロットマシン内部にメダルをクレジットすることをいう。「貯留」は、ベットを含む意味で用いられる場合もあるが、本明細書では、「貯留」というときは、「ベット」を含まない意味で使用する。本実施形態において、貯留可能な最大（限界）枚数は、遊技状態等にかかわらず、「50」枚に設定されている。

【0012】

50

「手入れ」とは、遊技者が、後述するメダル投入口４３からメダルを直接投入することをいう。

「手入れベット」とは、遊技者が、メダル投入口４３からメダルを手入れすることにより、メダルをベットすることをいう。

「手入れ貯留」とは、遊技者が、メダル投入口４３からメダルを手入れすることにより、メダルを貯留する（クレジットを加算する）ことをいう。

「ベットメダル」とは、ベットされているメダルをいう。

「貯留メダル」とは、クレジットとして貯留されているメダルをいう。

【００１３】

「貯留ベット」とは、遊技者が後述するベットスイッチ４０を操作することにより、当該遊技でベット可能な範囲内において、クレジットとして貯留されているメダルの一部又は全部を、遊技を行うためにベットすることをいう。

「自動ベット」とは、リプレイが入賞したときに、スロットマシンの内部制御処理により、当該遊技でベットされていた数のメダルを自動でベットすることをいう。なお、上記の手入れベットしたメダル、貯留ベットしたメダル、及び貯留メダルは、その後に精算可能であるが、リプレイの入賞により自動ベットされたメダルは精算を行うことができないように設定されている。

【００１４】

「投入」とは、上記の手入れベット、手入れ貯留、貯留ベット、及び自動ベットを含み、メダルをベット又は貯留することをいう。

「精算」とは、ベットメダル及び／又は貯留メダルを遊技者に対して払い出すことをいう。

「払出し」とは、上記精算によりメダルを払い出すこと、又は役の入賞に基づきメダルを遊技者に払い出すことをいい、クレジットとして貯留すること、又は払出し口からの実際のメダルを払い出すことをいう。本実施形態における払出しは、「５０」枚を限界枚数として貯留し、「５０」を超えた分のメダルは、直接払い出すように制御する。

【００１５】

以下、図面等を参照して、本発明の一実施形態について説明する。

図１は、本実施形態におけるスロットマシン１０（遊技機）の制御の概略を示すブロック図である。スロットマシン１０は、メイン制御基板５０とサブ制御基板８０とを備える。

メイン制御基板５０は、入力ポート５１、出力ポート５２、メモリ５３、メインＣＰＵ５４等を備える（図１で図示したもののみを備える意味ではない）。

【００１６】

メイン制御基板５０と、図１で図示した操作スイッチ等の遊技進行用の周辺機器とは、入力ポート５１又は出力ポート５２を介して電氣的に接続されている。入力ポート５１は、操作スイッチ等の信号が入力される接続部であり、出力ポート５２は、モータ３２等の周辺機器に対して信号を送信する接続部である。

【００１７】

なお、図１において、入力用の周辺機器は、その周辺機器からの信号がメイン制御基板５０又はサブ制御基板８０に向かう矢印で表示しており、出力用の周辺機器は、メイン制御基板５０又はサブ制御基板８０からその周辺機器に向かう矢印で示している。

【００１８】

メモリ５３は、遊技の進行等に必要なプログラム等を記憶しておくＲＯＭ、及びメインＣＰＵ５４が各種の制御を行うときに取り込んだデータ等を一時的に記憶しておくＲＷＭ（Read Write Memory）からなる。また、メインＣＰＵ５４に備えられたレジスタもメモリ５３に含まれる。

メインＣＰＵ５４は、メイン制御基板５０上に設けられたＣＰＵを指し、遊技の進行に必要なプログラムの実行、演算等を行い、具体的には、役の抽選、リール３１の駆動制御、及び入賞時の払出し等を実行する。

10

20

30

40

50

【0019】

また、サブ制御基板80は、遊技中及び遊技待機中における演出（情報）の選択・出力等を制御するものである。サブ制御基板80と、演出用の周辺機器とは、入力ポート81又は出力ポート82を介して電氣的に接続されている。また、メイン制御基板50と同様に、サブ制御基板80は、メモリ83及びサブCPU84を備える。

【0020】

メモリ83は、演出用のデータ等（演出パターン等）を記憶しておくROM、サブCPU84が各種の演出を出力するときに取り込んだデータ等を一時的に記憶しておくRWMからなる。

サブCPU84は、所定のプログラムに従って、演出の決定及び実行、AT（サブボーナス）に関する抽選等を実行する。

【0021】

サブ制御基板80は、メイン制御基板50の下位に属する制御基板である。そして、メイン制御基板50とサブ制御基板80とは電氣的に接続されており、メイン制御基板50のメインCPU54内にあるシリアル通信回路により、サブ制御基板80に一方向で演出の出力に必要な信号や情報（制御コマンド等）を送信する。

【0022】

メイン制御基板50からサブ制御基板80に送信される情報としては、たとえば、メダルが投入（ベット、貯留）された旨の情報、スタートスイッチ41が操作された旨の情報、役の抽選結果（当選役）の情報、リール31の回転が開始された旨の情報、ストップスイッチ42が操作された旨の情報、リール31が停止した旨の情報、各リール31の停止位置（停止図柄）の情報、入賞役の情報、メダルの払出し（リプレイの入賞による自動ベットを含む）の情報、遊技状態の情報、フリーズに関する情報、精算時の情報等が挙げられる。

【0023】

図1において、メダル投入口43は、遊技者が実際にメダルを投入（手入れ）する部分であり、スロットマシン10の筐体前面側に設けられている。メダル投入口43から投入されたメダルは、メダルセレクトを通過する。

メダルセレクトは、図1に示すように、通路センサ43a、ブロック45、投入センサ44a及び44bを備える（ただし、これらに限定されるものではない）。

メダル投入口43からメダルが投入されると、最初に、通路センサ43aにより検知されるように構成されている。

【0024】

さらに、通路センサ43aの下流側には、ブロック45が設けられている。ブロック45は、メダルの投入を許可／不許可にするためのものであり、メダルの投入が不許可状態のときは、メダル投入口43から投入されたメダルを払出し口から返却するメダル通路を形成する。これに対し、メダルの投入が許可状態のときは、メダル投入口43から投入されたメダルをスロットマシン10内部のホッパー（メダル溜め）に案内するメダル通路を形成する。

【0025】

ここで、ブロック45は、遊技中（リール31の回転開始時から、全リール31が停止し、役の入賞時には入賞役に対応する払出しの終了時まで）は、メダルの投入を不許可状態とする。すなわち、ブロック45がメダルの投入を許可するのは、少なくとも遊技が行われていないときである。

【0026】

ブロック45のさらに下流側には、投入センサ44a及び44b（光学センサ）が設けられている。したがって、メダル投入口43から投入されたメダルは、通路センサ43aによって検知された後、さらに、投入センサ44a（上流側）及び44b（下流側）により検知されるように構成されている。なお、図1に示すように、後述する説明においては、上流側の投入センサ44aを投入センサ1、下流側の投入センサ44bを投入センサ2

10

20

30

40

50

と称する場合もある。

【0027】

また、図1に示すように、メイン制御基板50には、遊技者が操作する操作スイッチとして、精算スイッチ46、1ベットスイッチ40a、3ベットスイッチ40b、スタートスイッチ41、(左、中、右)ストップスイッチ42が電氣的に接続されている。

【0028】

精算スイッチ46は、スロットマシン10内部に貯留(クレジット)されたメダルを払い戻す(ペイアウトする)ときに遊技者が操作するスイッチである。

ベットスイッチ40は、貯留されたメダルを当該遊技のためにベットするときに遊技者が操作するスイッチであり、本実施形態では、1枚投入用の1ベットスイッチ40aと、3枚投入用の3ベットスイッチ40bとを備える。なお、これに限らず、2枚投入用のベットスイッチを設けてもよい。また、1枚、2枚、3枚投入用のベットスイッチのうち、スロットマシン10の仕様に応じて、2つ又は3つ設けることも可能である。

なお、「ベットスイッチ40」というときは、1ベットスイッチ40a及び／又は3ベットスイッチ40bを指すものとする。

【0029】

また、スタートスイッチ41は、(左、中、右のすべての)リール31を始動させるときに遊技者が操作するスイッチである。

さらにまた、ストップスイッチ42は、3つ(左、中、右)のリール31に対応して3つ設けられ、対応するリール31を停止させるときに遊技者が操作するスイッチである。

【0030】

図1に示すように、メイン制御基板50には、ベット枚数表示装置47a及び貯留枚数表示装置47bが電氣的に接続されている。ベット枚数表示装置47aは、現時点でベットされているメダル枚数を表示する装置であり、「0」～「3」(整数)を表示する。

また、貯留枚数表示装置47bは、貯留されているメダルをセグメント表示するものであり、本実施形態では、「00」～「50」(整数)の間の数字を表示する。

なお、貯留枚数(数字)に限らず、エラー発生時に、エラーコード(アルファベット)を表示する場合もある。

【0031】

たとえば、メダルが全くベット及び貯留されていない状態で、1枚のメダルが手入れされると、当該遊技のためにその1枚のメダルがベットされる。ベットされたメダルは、ベット枚数表示装置47aによって表示される。さらに2枚を追加投入すると、当該遊技のために3枚のメダルがベットされる(ベット限界枚数が3枚の場合)。したがって、手入れされたメダルが3枚までのときは、そのメダルはベットされ、貯留されない。さらにメダルが手入れされ続けると、スロットマシン10内部にメダルが貯留されるとともに、その貯留枚数が貯留枚数表示装置47bによって表示される。貯留されたメダル枚数は、メモリ53の所定の格納領域に記憶される。

【0032】

上述のように、本実施形態では、最大で50枚までのメダルを貯留可能となっている。したがって、貯留枚数が50枚となったとき(貯留枚数表示装置47bに「50」と表示されたとき)は、それ以上、メダルは貯留されない。この状態で、仮に、メダル投入口43からメダルが手入れされると、ブロック45により、手入れされたメダルは、払出し口から返却される。

【0033】

また、たとえばメダル払出しのある役(リプレイを除く)が入賞してその役に対応するメダルが払い出されるときは、払出し口から払い出されることよりも優先して、スロットマシン10内部にメダルが貯留される。たとえば、役の入賞時、その役の入賞に対応する払出し枚数が8枚であり、役の入賞前の貯留枚数が「10」であるときは、その役の入賞により、貯留枚数が「10」から「18」に変更されるとともに、貯留枚数表示装置47bによる表示数も「10」から「18」に更新される。

【0034】

さらにまた、役の入賞時に、貯留枚数が「50」を超えるときは、「50」を超えた分については払出し口から払い出さされる。たとえば、役の入賞前に貯留枚数が「47」であり、役の入賞によって8枚のメダルが払い出されるとき、3枚は貯留されて貯留枚数が「50」となり、「50」を超える5枚については払出し口から払い出される。

【0035】

さらに、リプレイの入賞時は、メダルの貯留及び払出しは行われず、当該遊技でベットされていた枚数のメダルが再遊技のために自動ベットされる。たとえば、当該遊技を2ベット（2枚）で行い、リプレイが入賞したときは、2枚のメダルが自動ベットされる。同様に、当該遊技を3ベット（3枚）で行い、リプレイが入賞したときは、3枚のメダルが自動ベットされる。そして、リプレイの入賞に基づく自動ベットは、再遊技を行うためのメダルの投入であるので、その後に精算（返却）操作を行っても、当該メダルを精算することはできない。

【0036】

なお、「遊技機の認定及び型式の検定等に関する規則」では、リプレイに対応する図柄の組合せが有効ラインに停止したときは、再遊技に係る条件装置の作動であって「入賞」ではないと解釈されている。しかし、本願（本明細書等）では、リプレイについても役の1つとして扱い（再遊技役）、リプレイに対応する図柄の組合せが有効ラインに停止したことを「リプレイの入賞」と称する。

【0037】

メイン制御基板50の出力ポート52には、図柄表示装置のモータ32等が電氣的に接続されている。

図柄表示装置は、図柄を表示する（本実施形態では3つの）リール31と、各リール31をそれぞれ駆動するモータ32と、リール31の位置を検出するためのインデックスセンサ33を含む。

【0038】

モータ32は、リール31を回転させるためのものであり、各リール31の回転中心部に連結され、後述するリール制御手段62によって制御される。ここで、リール31は、左リール31、中リール31、右リール31からなり、左リール31を停止させるときに操作するストップスイッチ42が左ストップスイッチ42であり、中リール31を停止させるときに操作するストップスイッチ42が中ストップスイッチ42であり、右リール31を停止させるときに操作するストップスイッチ42が右ストップスイッチ42である。

【0039】

リール31は、リング状のものであって、その外周面には複数種類の図柄（役に対応する図柄の組合せを構成している図柄）を印刷したリールテープを貼付したものである。

図2は、本実施形態におけるリール31の図柄配列を示す図である。図2では、図柄番号を併せて図示している。図2に示すように、本実施形態では、各リール31ごとに、21個の図柄表示領域が等間隔で配置されているとともに（図柄コマ数が21個）、各図柄表示領域にそれぞれ所定の図柄が表示されている。

なお、図柄コマ数は、21個以外に、20個の場合が挙げられる。

【0040】

また、各リール31には、1個（2個以上であってもよい）のインデックスが設けられている。インデックスは、リール31の例えば周側面に凸状に設けられており、リール31が所定位置を通過したか否かや、1回転したか否か等を検出するときに用いられる。そして、各インデックスを検知するための（左、中、右）インデックスセンサ（フォトセンサ）33が設けられている。各インデックスセンサ33は、メイン制御基板50の入力ポート51に電氣的に接続されている。そして、インデックスがインデックスセンサ33を検知する（切る）と、その入力信号がメイン制御基板50に入力され、そのリール31が所定位置を通過したことが検知される。

【0041】

また、インデックスセンサ 33 がリール 31 のインデックスを検知した瞬間の基準位置上の図柄が予めメモリ 53（たとえば ROM）に記憶されている。これにより、インデックスを検知した瞬間の基準位置上の図柄を検知することができる。

【0042】

図 3 は、スロットマシン 10 のフロントマスク部（前面扉。図示せず。）に設けられた表示窓（透明窓）11 と、各リール 31 との位置関係を示す図である。各リール 31 は、本実施形態では横方向に並列に 3 個（左リール 31、中リール 31、及び右リール 31）設けられている。さらに、各リール 31 は、表示窓 11 から、上下に連続する 3 図柄が見えるように配置されている。よって、スロットマシン 10 の表示窓 11 から、合計 $3 \times 3 = 9$ 個の図柄が見えるように配置されている。

10

【0043】

なお、本明細書では、図 3 中、左リール 31 の「RP」図柄、中リール 31 の「スイカ」図柄、及び右リール 31 の「チェリー」図柄が停止している位置を「上段」と称し、左、中及び右リール 31 の「赤 7」図柄が停止している位置を「中段」と称し、左リール 31 の「スイカ」図柄、中リール 31 の「RP」図柄、及び右リール 31 の「ベル」図柄が停止している位置を「下段」と称する。

【0044】

そして、左、中及び右リール 31 のそれぞれ表示窓 11 内の 1 図柄を通るラインが、1 つの図柄組合せラインとなる。ここで、「図柄組合せライン」とは、リール 31 の停止時における図柄の並びラインであって図柄の組合せを形成させるラインである。

20

本実施形態では、図柄組合せラインのうち、10 本が有効ラインに設定されている。ここで「有効ライン」とは、いずれかの役に対応する図柄の組合せがそのラインに停止したときに、その役の入賞となるラインである。

【0045】

特に、中リール 31 の上段の図柄を通る有効ラインは、「上段（左リール 31）」－「上段（中リール 31）」－「上段（右リール 31）」の 1 本である。また、中リール 31 の中段の図柄を通る有効ラインは、図 3 に示す 9 本である。なお、中リール 31 の下段の図柄を通る有効ラインは設けられていない。

【0046】

そして、図柄組合せラインのうち、上記有効ライン以外は、無効ラインとなる。ここで、「無効ライン」とは、図柄組合せラインのうち、有効ラインとして設定されないラインであって、いずれかの役に対応する図柄の組合せがそのラインに停止した場合であっても、その役に応じた利益の付与（メダルの払出し等）を行わないラインである。すなわち、無効ラインは、そもそも図柄の組合せの成立対象となっていないラインである。

30

【0047】

有効ライン及び無効ラインは、メダルのベット枚数や遊技状態等に応じて設定されるが、本実施形態では、3 枚又は 2 枚のメダルをベットして遊技を行うように設定されている。そして、常に 10 本が有効ラインとなる。

なお、これに限らず、メダルのベット枚数として、1 枚の場合を設けてもよい。また、メダルのベット枚数や遊技状態等に応じて、図柄組合せラインのうち、有効ライン及び無効ラインの種類及び数を設定してもよい。

40

【0048】

また、メイン制御基板 50 には、メダル払出し装置が電氣的に接続されている。メダル払出し装置は、メダルの貯留部となるホッパーのメダルを払出し口から払い出すときに駆動するホッパーモータ 36 と、ホッパーモータ 36 から払い出されたメダルを検出するための払出しセンサ 37a 及び 37b と、ホッパーの満杯を検出するための満杯センサ 38 とを備える。

【0049】

メダル投入口 43 から手入れされ、受け付けられたメダルは、所定の通路を通してホッパー内に収容されるように形成されている。

50

払出しセンサ 37 a 及び 37 b は、投入センサ 44 a 及び 44 b と同様に、上流側に払出しセンサ 37 a が設けられ、下流側に払出しセンサ 37 b が設けられている。

なお、図 1 に示すように、後述する説明においては、上流側の払出しセンサ 37 a を払出しセンサ 1、下流側の払出しセンサ 37 b を払出しセンサ 2 と称する場合もある。

【0050】

払出しセンサ 37 a と 37 b とは、所定距離を隔てて配置され、メダルが払出しセンサ 37 a により検知されてから所定時間を経過した後に払出しセンサ 37 b により検知されるように構成されている。そして、払出しセンサ 37 a 及び 37 b がそれぞれオン／オフとなるタイミングに基づいて、メダルが正しく払い出されたか否かを判断する。

【0051】

たとえば、ホッパーモータ 36 が駆動しているにもかかわらず、払出しセンサ 37 a 及び 37 b の信号がいずれもオフであるときは、メダルが払い出されていないと判断し、ホッパーエラー（メダルなし）と検知される。

一方、払出しセンサ 37 a 及び 37 b の信号がオンのままとなったときは、メダル詰まりが生じたと検知する。

【0052】

満杯センサ 38 は、ホッパー自体、あるいはホッパーから溢れたメダルを収容するサブタンクの満杯を検知するセンサであり、たとえばホッパー又はサブタンクのメダルが満杯となったときにメダルが接触することで通電する回路から構成される。

【0053】

さらに、図 1 において、サブ制御基板 80 の入力ポート 81 には、メニューキー 85 が電氣的に接続されている。メニューキー 85 の使用については後述するが、遊技者が意図する情報を表示させたりするときや、演出を出力する際に用いられ、カーソルを移動させるための十字キー 85 a と、確定キーとして機能するエンターキー 85 b とを備える。

【0054】

また、サブ制御基板 80 の出力ポート 82 には、ランプ 21、スピーカ 22、及び画像表示装置 23 等の演出用の周辺機器が電氣的に接続されている。

ランプ 21 は、スロットマシン 10 の演出用のランプ（LED 等）であり、所定の条件を満たしたときに、それぞれ所定のパターンで点灯する。なお、ランプ 21 には、各リール 31 の内周側に配置され、リール 31 に表示された図柄（表示窓 11 から見える上下に連続する 3 図柄）を背後から照らすためのバックランプ、リールの上部からリール上の図柄を照光する蛍光灯、スロットマシン 10 の筐体前面に配置され、役の入賞時等に点滅する上部ランプ及びサイドランプ（いずれも図示せず）等が含まれる。

【0055】

また、スピーカ 22 は、遊技中に各種の演出を行うべく、所定の条件を満たしたときに、所定のサウンドを出力するものである。

さらにまた、画像表示装置 23 は、液晶ディスプレイ、有機 EL ディスプレイ、ドットディスプレイ等からなるものであり、遊技中に各種の演出画像（フリーズ中の演出画像、AT 中の押し順、役の抽選結果に対応する演出等）や、遊技情報（AT 中の遊技回数や獲得枚数等）、メニュー画面（後述）等を表示するものである。

【0056】

図 4 は、本実施形態における役（後述する役抽選手段 61 で抽選される役）の種類、払出し枚数等、及び図柄の組合せ等を示す図である。図 4 に示すように、役としては、遊技状態移行役、小役、及びリプレイが設けられている。

そして、各役に対応する図柄の組合せ及び入賞時の払出し枚数等が定められている。これにより、すべてのリール 31 の停止時に、いずれかの役に対応する図柄の組合せが有効ラインに停止すると、その役の入賞となり、その役に対応する枚数のメダルの払出し又は自動ベットが行われる（ただし、遊技状態移行役を除く。）。

本明細書では、「いずれかの役に対応する図柄の組合せがいずれかの有効ラインに停止する」ことを、「その役が入賞する」と称する。

10

20

30

40

50

【0057】

役において、まず、遊技状態移行役とは、その役が当選又は入賞すると、役の当選確率がそれまでと異なる他の遊技状態に移行する役であり、本実施形態では、MB（ミドルボーナス。第2種ビッグボーナス（2BB）ともいう。）、及びCB（チャレンジビッグ）が設けられている。

なお、他の遊技状態移行役としては、1BB（第1種ビッグボーナス）、RB（レギュラーボーナス）、SB（シングルボーナス）が挙げられるが、本実施形態では設けられていない。

【0058】

また、遊技状態移行役の入賞によって移行する遊技には、通常遊技以上に出玉率が高く、遊技者にとって有利な遊技と、メダルの増加を目的としない遊技とを有する。

ここで、本実施形態のMBは、MBに当選している遊技状態（MB内部中遊技）を作り出すことを目的とするものであり、MBを入賞させてMB遊技に移行させ、そのMB遊技で遊技者のメダルを増加させることを直接の目的とするものではない。

【0059】

MBに当選し、かつMBが入賞すると、次遊技からMB遊技に移行する。MB遊技は、CBに当選することなくCB遊技が実行される（いわゆる、CBが連続作動する）遊技である。MB遊技は、メダルを規定枚数獲得するまで継続される。MB遊技においてメダルを規定枚数獲得したときは、MB遊技を終了して、次遊技から通常遊技（非内部中）に移行する。

また、CBに当選し、かつCBが入賞すると、次遊技からCB遊技に移行する。CB遊技は、原則として1遊技のみ実行する遊技である。当該1遊技（CB遊技）を終了すると、次遊技から通常遊技（非内部中）に移行する。

【0060】

また、小役とは、予め定められた枚数のメダルが払い出される役であり、本実施形態では、チェリー、スイカ、ベルA～Dを備えており、各小役ごとに所定の図柄の組合せが設定されている。なお、ベルAにおける「ANY」とは、いずれの図柄でもよいことを意味している。

【0061】

特に、ベルは、ベルA～Dの4種類を備える。ここで、ベルB～Dは、ストップスイッチ42の押し順に応じて、ベルAである「ANY」－「ベル」－「ANY」の図柄の組合せをどの有効ラインに停止させるかを制御するための「制御役」としての役割を有し、入賞を本来の目的とするものではない。この点については後述する。

【0062】

また、リプレイ（再遊技役）とは、当該遊技で投入したメダル枚数を維持した再遊技が行えるようにした役である。

本実施形態のリプレイは、ノーマルリプレイ、ベルリプレイ、特殊リプレイA～Cを備え、それぞれ図柄の組合せが異なる。

【0063】

ベルリプレイとは、実際の入賞役は「リプレイ」であるが、遊技者には小役であるベルに見せるようにしたリプレイである。

上述したように、たとえば「中段」－「中段」－「中段」（中段ライン）は有効ラインであるが、「下段」－「下段」－「下段」（下段ライン）は無効ラインに設定されている。

そして、ベルリプレイの当選時には、中段ライン（有効ライン）には、ベルリプレイに対応する図柄の組合せを停止させる。なお、図4中、「／」は、「又は」を意味する（後述する特殊リプレイCも同様である）。たとえばベルリプレイの左リール31の図柄は、「青7」、「BAR」又は「スイカ」のいずれか（いずれでもよい）という意味である。

【0064】

ベルリプレイの当選時に、たとえば「青7（8番）」－「青7（2番）」－「スイカ（

10

20

30

40

50

1 3 番)」が中段ラインに停止したとする。なお、かっこ書きで示す番号は、図 2 の図柄番号である。これにより、中段ラインには、ベルリプレイの図柄の組合せが停止したこととなるのでベルリプレイの入賞となる。さらに、この図柄の組合せの停止時に、下段ライン（無効ライン）には、「ベル（7 番）」－「ベル（1 番）」－「ベル（1 2 番）」、すなわちベル揃いが停止する。

【0065】

そして、ベルリプレイの入賞時には、有効ラインである中段ライン上の図柄ではなく、下段ラインの「ベル」の図柄を点滅する等して、「ベル」（小役）が入賞したかのような演出を出力する。

なお、図 4 及び図 2 に示すように、ベルリプレイの他の図柄の組合せ、たとえば「BAR」－「RP」－「赤 7」等であっても、常に、下段ラインにはベル揃いが停止するようにリール 3 1 の図柄が配列されている。

【0066】

さらに、ベルリプレイの入賞時に、直ちにメダルの自動ベットを行うと、リプレイの入賞に見えてしまうため、ベルリプレイの入賞と同時にたとえばフリーズを実行する。

ここで「フリーズ」とは、遊技の進行を所定期間一時停止状態にして遅延させることであり、たとえば、メダルの受付け、ベットスイッチ 4 0 の操作の受付け、スタートスイッチ 4 1、ストップスイッチ 4 2 の操作の受付け、又はリール 3 1 の停止操作の受付けに関する機能を一時停止状態にすることである。さらに、前記一時停止状態の間に、リール 3 1 が通常動作とは異なる動作をする等の演出期間として使用される（いわゆる擬似遊技を実行する）こともある。

【0067】

本実施形態では、ベルリプレイの入賞と同時にフリーズを開始するとともに、その経過時間を計測する。そして、所定時間（たとえば 20 秒）を経過する前にベットスイッチ 4 0 が操作されたときは、そのベットスイッチ 4 0 の操作を契機としてフリーズを解除（キャンセル）する。メイン CPU 5 4 は、フリーズを解除すると、ベルリプレイの入賞に基づくメダルの自動ベットを行うように制御する。

【0068】

よって、3 枚のメダルをベットして遊技を行い、ベルリプレイが入賞したとき、その入賞時から 20 秒を経過する前に遊技者がベットスイッチ 4 0（3 ベットスイッチ 4 0 b）を操作したときは、3 枚のメダルが自動ベットされる。これにより、遊技者に対し、3 ベットスイッチ 4 0 b の操作により 3 枚のメダルがベットされた印象を与えることができるので、ベルリプレイを、擬似的に小役（ベル）に見せることができる。

【0069】

また、フリーズの解除は、ベットスイッチ 4 0 の操作時に限らず、遊技者によるメダル投入口 4 3 からのメダルの手入れ時に行うことも可能である。

ベルリプレイの入賞時に、遊技者は、ベルリプレイを小役であると認識していれば、3 枚のメダルがクレジットに加算されると考える。したがって、その 3 枚のクレジットを用いて（3 ベットスイッチ 4 0 b の操作により）次遊技を開始できると考える。

【0070】

さらに、小役の入賞後、次遊技の開始前（スタートスイッチ 4 1 を操作する前）に、メダルを手入れベットして遊技を行うことは、当然に可能である。ここで、ベルリプレイの入賞後、次遊技の開始前に、遊技者がメダルを手入れしたときに、そのメダルを受け付けずに、払出し口から返却してしまうと、ベルリプレイを小役のように見せることができない。そこで、ベルリプレイの入賞時に、メダル投入口 4 3 からのメダルの手入れを検出したときに、フリーズを解除するとともに、そのメダルを受け付け、貯留するように制御する。

【0071】

この場合、通路センサ 4 3 a によりメダルが検知された時にフリーズを解除するとともに、ベルリプレイの入賞に基づくメダルの自動ベットを行う。さらに、手入れされたメダ

10

20

30

40

50

ルについては、貯留処理を行う。なお、通路センサ 4 3 aに限らず、投入センサ 4 4 によりメダルが検知されたときにフリーズを解除してもよい。あるいは、スタートスイッチ 4 1 が操作されたときにフリーズを解除してもよい。

【0072】

さらにまた、リプレイのうち、特殊リプレイ A 及び B は、複数種類のリプレイを重複当選させるために用いられるものであり、入賞を本来の目的とするものではない。

さらに、特殊リプレイ C は、次遊技から A T (サブボーナス) 遊技を開始させるための役である。上述したように、本実施形態では、遊技状態移行役として 1 B B は設けられていないが、遊技状態を移行することなく A T 遊技を開始する役として、特殊リプレイ C を設けている。

【0073】

上述した各役において、役に当選した遊技でその役に対応する図柄の組合せが有効ラインに停止しなかったときは、次遊技以降に持ち越される役と、持ち越されない役とが定められている。

持ち越される役としては、遊技状態移行役である M B が挙げられる。M B に当選したときは、リール 3 1 の停止時に、M B に対応する図柄の組合せが有効ラインに停止するまでの遊技において、M B の当選を次遊技以降に持ち越すように制御される。

【0074】

このように、M B の当選は持ち越されるのに対し、M B 以外の役 (C B、小役及びリプレイ) は、持ち越されない。役の抽選において、M B 以外の役に当選したときは、当該遊技でのみその当選役が有効となり、その当選は次遊技以降に持ち越されない。すなわち、これらの役に当選した遊技では、その当選した役に対応する図柄の組合せ (重複当選時はいずれか 1 つの役) が有効ラインに停止可能となる場合を有するようにリール 3 1 が停止制御されるが、その当選役の入賞の有無にかかわらず、その遊技の終了時に、その当選役に係る権利は消滅する。

【0075】

遊技の開始時には、遊技者は、ベットスイッチ 4 0 を操作して予め貯留されたメダルをベットするか、又はメダル投入口 4 3 からメダルを手入れする。遊技開始前にメダルが貯留されているときは、その貯留枚数の範囲内で、1 ベットスイッチ 4 0 a を操作すれば 1 枚のメダルがベットされ、3 ベットスイッチ 4 0 b が操作されれば 3 枚のメダルがベット

される。なお、M B 遊技中のように 2 枚のベットで遊技を行う場合において、3 ベットスイッチ 4 0 b が操作されたときは 2 枚のメダルがベットされるように制御してもよい。

また、貯留の有無にかかわらず、ベットメダル無しの状態から、メダル投入口 4 3 から 1 枚のメダルが手入れされると 1 ベットされ、3 枚のメダルが手入れされれば 3 ベットされる。

そして、ベットが行われたときは、その信号がメイン制御基板 5 0 に入力される。さらに、ベットが行われた状態でスタートスイッチ 4 1 が操作されると、その信号がメイン制御基板 5 0 に送信される。

【0076】

メイン C P U 5 4 (具体的には、後述するリール制御手段 6 2) は、スタートスイッチ 4 1 の操作信号を受信すると、すべてのモータ 3 2 を駆動制御して、すべてのリール 3 1 を回転させるように制御する (ただし、フリーズの実行時には回転させない場合もある)。このようにしてリール 3 1 がモータ 3 2 によって回転されることで、リール 3 1 上の図柄は、所定の速度で表示窓内で上下方向 (図柄が上段から下段に移動する方向) に移動表示される。

【0077】

また、メイン制御基板 5 0 の役抽選手段 6 1 は、スタートスイッチ 4 1 が操作された信号を検知したときは、役の抽選を行う。役抽選手段 6 1 は、スタートスイッチ 4 1 が操作されたときに乱数値を抽出し、その乱数値がどの当選役に該当するかを役抽選テーブルと照合することにより、当該遊技での当選役を決定する。

【0078】

役抽選テーブルは、抽選される役の種類と、各役の当選確率とを定めたものである。図5は、遊技状態ごとの抽選される役の種類と当選確率とを示す図である。役抽選テーブルは、それぞれ所定の範囲の抽選領域を有し、この抽選領域は、各役の当選領域及び非当選領域に分けられているとともに、抽選される役が、予め設定された当選確率となるように所定の割合に設定されている。

【0079】

先ず、非内部中遊技では、遊技状態移行役であるMB及びCBと、ベル重複当選1～6、ノーマルリプレイ単独当選が抽選される。これに対し、MB内部中遊技及びMB遊技中は、MBは抽選されない。すなわち、一旦MBに当選すると、そのMBの入賞に基づくMB遊技が終了するまで、MBは抽選されない。

10

【0080】

また、図5において、「+」とは、当該遊技で同時に重複当選することを意味している。たとえばベル重複当選4である「ベル(A+B+C)」とは、ベルA、ベルB、ベルCの3種類のベルが当該遊技で同時に重複当選することを意味する。同様に、たとえばリプレイ重複当選3である「ノーマルリプレイ+特殊リプレイ(A+C)」とは、ノーマルリプレイ、特殊リプレイA、特殊リプレイCの3種類のリプレイが当該遊技で同時に重複当選することを意味する。

なお、本実施形態では例示していないが、たとえば遊技状態移行役(MB、CB)と小役やリプレイが重複当選するようにしてもよいのはもちろんである。

20

【0081】

ベルは、いずれが1つが単独当選する場合はなく、ベルAと、他の少なくとも1つのベルとの重複当選(ベル重複当選1～6)を有する。

また、リプレイは、ノーマルリプレイのみが当選するノーマルリプレイ単独当選と、ベルリプレイのみが当選するベルリプレイ単独当選と、ノーマルリプレイ及び少なくとも1つの特殊リプレイの重複当選(リプレイ重複当選1～3)を有する。

【0082】

MB内部中遊技では、ノーマルリプレイ単独当選、ベルリプレイ単独当選、及びリプレイ重複当選の双方が設けられており、その合算の当選確率は、 $1/2$ 弱に設定されている。また、MB内部中遊技におけるベル重複当選の合算確率は $1/2$ に設定されている。すなわち、MB内部中遊技では、非当選がほとんどないように設定されている。

30

【0083】

これに対し、非内部中遊技では、リプレイは、ノーマルリプレイ単独当選のみであり、その当選確率は、 $1/7.3$ である。

さらに、MB遊技及びCB遊技では、小役及びリプレイは抽選されない。ただし、MB遊技中及びCB遊技中は、全小役の当選フラグ63aがオンとなり、全小役に当選しているときと同一の状態となる。

【0084】

遊技者は、ストップスイッチ42の操作受けが有効となっているときにストップスイッチ42を押すことで、そのストップスイッチ42に対応するリール31(例えば、左ストップスイッチ42に対応する左リール31)の回転を停止させる。ストップスイッチ42が操作されると、その信号がメイン制御基板50に入力される。

40

【0085】

メインCPU54(具体的にはリール制御手段62)は、この信号を受信すると、役抽選手段61での役抽選結果と、ストップスイッチ42が操作された瞬間のリール31の位置とから、そのストップスイッチ42に対応するモータ32を駆動制御し、そのモータ32に係るリール31の停止制御を行う。

そして、すべてのリール31の停止時に、いずれかの役に対応する図柄の組合せが有効ラインに停止したとき(すなわち、その役の入賞時)は、入賞した役に対応するメダルの払出し等が行われる。

50

【0086】

図6は、当選役に対する、ストップスイッチ42の押し順と入賞役との関係を示す図である。

図6に示すように、MB当選時は、ストップスイッチ42の押し順が右中左であることを条件に、MBが入賞可能となるように設定されている。したがって、それ以外の押し順のときは、たとえ目押しが正確であっても、MBは入賞しないように設定されている。

【0087】

ここで、本実施形態におけるストップスイッチ42の押し順は、押し順報知が行われたとき（ほとんどは、AT遊技中）以外は、常に第一停止左、すなわち左中右か左右中と定めている。そして、仮に遊技者が（押し順非報知時に）第一停止中又は右で操作したときは、ペナルティを課すように制御する。ペナルティとしては、たとえば、AT遊技の抽選に関して所定の制限を課すこと、具体的にはペナルティ発生時から一定の遊技回数を消化するまでAT遊技を実行するか否かの抽選を行わないこと等が挙げられる。

10

【0088】

特に本実施形態のAT遊技は、MB内部中遊技でのみ実行されるので、MB当選時の遊技がAT遊技中である場合はない。したがって、MB当選時の遊技では、遊技者は、常に第一停止左でストップスイッチ42を操作するはずであるから、当選したMBが入賞する場合はない。

【0089】

さらに本実施形態では、MB内部中遊技では、小役とリプレイの当選とで、約99.9%を占めており、非当選となることはきわめて稀である。したがって、MB内部中遊技であっても、MBが入賞可能な遊技となることは極めて稀であり、かつ、MBは上述したように右中左の押し順であることが入賞の条件に設定されているので、MB内部中遊技であってもMBが入賞するのはきわめて稀である。

20

なお、仮に、MB内部中遊技においてMBが入賞可能となった遊技では、第一停止左となるいずれかの押し順を遊技者に指示（報知）することで、MBの入賞を回避するように設定されている。

【0090】

また、ベル重複当選1は、ベル(A+B)の重複当選であり、ストップスイッチ42の押し順が「左中右」であるときは、中リール31の停止時に中段に「ベル」図柄を停止させるとともに、ストップスイッチ42の押し順が「左中右」以外であるときは、中リール31の停止時に上段に「ベル」図柄を停止させるようにリール31を停止制御する。

30

【0091】

ここで、ベルAの図柄の組合せは、「ANY」－「ベル」－「ANY」であり（図4参照）、中リール31の「ベル」図柄が中段に停止すれば、（左及び右リール31の停止図柄にかかわらず）ベルAの入賞となり、かつベルAに対応する図柄の組合せが9本の有効ラインに停止したこととなる（図3参照）。

【0092】

これに対し、中リール31の「ベル」図柄が上段に停止したときは、1本の有効ラインに停止したこととなる。

40

以上のように、本実施形態では、ベルAが9本の有効ラインに停止する押し順が「正解押し順」であり、ベルAが1本の有効ラインに停止する押し順が「不正解押し順」となる。

【0093】

さらに、本実施形態では、ベルAの入賞時は、遊技者に対し、「ベル」図柄が有効ラインに停止したことを一見して理解できるようにするために、いずれかの一直線状のラインに、「ベル」－「ベル」－「ベル」図柄の組合せを停止させるように、各リール31の停止位置が定められている。どのラインに「ベル」揃いを停止させるかは任意である。

【0094】

ベル重複当選2は、ベル(A+C)の重複当選であり、ストップスイッチ42の押し順

50

が「左右中」であるとき（押し順正解時）は、中リール 3 1 の停止時に中段に「ベル」図柄を停止させるとともに、ストップスイッチ 4 2 の押し順が「左右中」以外であるとき（押し順不正解時）は、中リール 3 1 の停止時に上段に「ベル」図柄を停止させるようにリール 3 1 を停止制御する。

【0095】

同様に、ベル重複当選 3 は、ベル（A + D）の重複当選であり、ストップスイッチ 4 2 の押し順が「中左右」であるとき（押し順正解時）は、中リール 3 1 の停止時に中段に「ベル」図柄を停止させるとともに、ストップスイッチ 4 2 の押し順が「中左右」以外であるとき（押し順不正解時）は、中リール 3 1 の停止時に上段に「ベル」図柄を停止させるようにリール 3 1 を停止制御する。

10

【0096】

ベル重複当選 4 は、ベル（A + B + C）の重複当選であり、ストップスイッチ 4 2 の押し順が「中右左」であるとき（押し順正解時）は、中リール 3 1 の停止時に中段に「ベル」図柄を停止させるとともに、ストップスイッチ 4 2 の押し順が「中右左」以外であるとき（押し順不正解時）は、中リール 3 1 の停止時に上段に「ベル」図柄を停止させるようにリール 3 1 を停止制御する。

【0097】

また、ベル重複当選 5 は、ベル（A + B + D）の重複当選であり、ストップスイッチ 4 2 の押し順が「右左中」であるとき（押し順正解時）は、中リール 3 1 の停止時に中段に「ベル」図柄を停止させるとともに、ストップスイッチ 4 2 の押し順が「右左中」以外であるとき（押し順不正解時）は、中リール 3 1 の停止時に上段に「ベル」図柄を停止させるようにリール 3 1 を停止制御する。

20

【0098】

同様に、ベル重複当選 6 は、ベル（A + C + D）の重複当選であり、ストップスイッチ 4 2 の押し順が「右中左」であるとき（押し順正解時）は、中リール 3 1 の停止時に中段に「ベル」図柄を停止させるとともに、ストップスイッチ 4 2 の押し順が「右中左」以外であるとき（押し順不正解時）は、中リール 3 1 の停止時に上段に「ベル」図柄を停止させるようにリール 3 1 を停止制御する。

【0099】

図 6 では、上述したベルの 6 種類の重複当選と入賞役との関係を示している。図 6 中、「9 枚」とは、9 枚の払出しとなるように中リール 3 1 の停止時に中段に「ベル」図柄が停止することを意味し、「1 枚」とは、1 枚の払出しとなるように中リール 3 1 の停止時に上段に「ベル」図柄が停止することを意味する。

30

【0100】

なお、通常遊技（非 A T 遊技）の押し順非報知時は、第一停止左が前提であるので、ベル重複当選 3 ～ 6 時には、遊技者のストップスイッチ 4 2 の操作ミス等がない限り、押し順正解にはならない。

【0101】

また、図 2 に示すように、すべてのリール 3 1 において、「ベル」図柄は、5 図柄以内の間隔で配置されている。一方、本実施形態では、リール制御手段 6 2 は、ストップスイッチ 4 2 が操作された瞬間からリール 3 1 を停止させるまでの移動コマ（図柄）数を 4 コマ以内（ストップスイッチ 4 2 が操作された瞬間からリール 3 1 を停止させるまでの時間が 190ms 以内）に設定している。

40

これにより、リール 3 1 がどの瞬間に位置するときにストップスイッチ 4 2 が操作されても、リール制御手段 6 4 は、リール 3 1 の停止制御の範囲内において、（M B 遊技中及び C B 遊技中を除き）所望の位置に「ベル」図柄を停止させることができる。よって、ベル重複当選時には、リール制御手段 6 2 は、M B 遊技及び C B 遊技以外は、常に、所望の図柄組合せラインに、「A N Y」－「ベル」－「A N Y」の図柄の組合せを停止させることができる。

【0102】

50

なお、リール 3 1 がどの瞬間に位置するときにストップスイッチ 4 2 が操作されても、リール 3 1 の停止制御の範囲内において、所望の有効ラインに対象図柄を停止させることができること（常に入賞させることができること）を、「引込み率（P B）＝1」という。これに対し、遊技者の目押しによらなければ有効ラインに対象図柄を停止させることができないこと（常に入賞させることができないこと）を、「P B ≠ 1」という。

【0103】

また、図 2 に示すように、すべてのリール 3 1 において、「R P」図柄は、5 図柄以内の間隔で配置されている。これにより、ノーマルリプレイについても、ベル A と同様に、「P B＝1」である。

さらに、左リール 3 1 の停止時には、常に「青 7」、「B A R」又は「スイカ」のいずれかを中段に停止させることができる。同様に、中リール 3 1 の停止時には、常に「青 7」又は「R P」のいずれかを中段に停止させることができる。さらに同様に、右リール 3 1 の停止時には、常に「赤 7」又は「スイカ」のいずれかを中段に停止させることができる。

したがって、ベルリプレイの当選時には、常に、中段ラインにベルリプレイの図柄の組合せを停止させ、かつ下段ラインにベル揃いを停止させることができる。

【0104】

リプレイ重複当選 1 は、ノーマルリプレイと特殊リプレイ A との重複当選であり、ストップスイッチ 4 2 の押し順にかかわらず、ノーマルリプレイを入賞させるようにリール 3 1 を停止制御する（図 6）。

リプレイ重複当選 2 は、ノーマルリプレイと特殊リプレイ B との重複当選であり、ストップスイッチ 4 2 の押し順にかかわらず、ノーマルリプレイを入賞させるようにリール 3 1 を停止制御する（図 6）。

【0105】

リプレイ重複当選 3 は、ノーマルリプレイ、特殊リプレイ A、及び特殊リプレイ C の重複当選であり、ストップスイッチ 4 2 の押し順が「右中左」であるときは特殊リプレイ C を中段ラインに入賞させ、「右中左」以外の押し順であるときはノーマルリプレイを入賞させるようにリール 3 1 を停止制御する（図 6）。

【0106】

ここで、図 2 に示すように、「赤 7」図柄は、左リール 3 1 には 2 か所に、中及び右リール 3 1 には 1 か所に設けられている。したがって、遊技者が、右中左の押し順でストップスイッチ 4 2 を操作するとともに、すべてのリール 3 1 について「赤 7」図柄を狙えば、中段ラインに「赤 7」－「赤 7」－「赤 7」を停止させることができる。

【0107】

また、右及び中リール 3 1 について「赤 7」図柄を中段に停止させることができないときは、「R P」図柄を中段に停止させる。さらにまた、左リール 3 1 について「赤 7」図柄を中段に停止させることができないときは「ベル」図柄を中段に停止させる。上述のように、「ベル」及び「R P」図柄は、いずれも「P B＝1」の配置であるので、常に中段に特殊リプレイ C の図柄の組合せを停止させることができる。

なお、リプレイの入賞時には、どのリプレイが入賞しても、当該遊技で遊技者に与える利益（再遊技）は同一である。ただし、特殊リプレイ C の入賞時には、サブ制御基板 8 0 によって次遊技から A T 遊技を開始させる。

【0108】

ここで、図 6 では、右中左の押し順でストップスイッチ 4 2 を操作したときに特殊リプレイ C が入賞し、他の押し順ではノーマルリプレイが入賞するように設定しているが、これに限らず、第一停止操作が「左」でない（通常遊技ではペナルティとなる）押し順（中左右、中右左、右左中、右中左）の全部又は一部の押し順時に特殊リプレイ C を入賞させるように制御することも可能である。

また、サブ制御基板 8 0 が A T 遊技を開始することが可能に制御されている（サブ制御基板 8 0 における A T 遊技の抽選に当選し、A T 準備中となっている）状態において、リ

10

20

30

40

50

プレイ重複当選3となり、かつ特殊リプレイCが入賞可能となる押し順でストップスイッチ42が操作されたことを条件として、サブ制御基板80は、AT遊技を開始させる。

したがって、リプレイ重複当選3となり、特殊リプレイCが入賞したとしても、サブ制御基板80側でAT遊技開始の条件を満たしていなければAT遊技は開始されない。

【0109】

また、リプレイ重複当選3となり、特殊リプレイCを入賞させる押し順でストップスイッチ42が操作されたときは、所定時間、メダルの投入や精算ができないフリーズを実行し、このフリーズ中に、AT遊技を開始する旨の演出等を出力する。なお、フリーズは、AT遊技を実行するときだけに発生させてもよいが、AT遊技の実行にかかわらず、特殊リプレイCの入賞時には、一律、所定時間のフリーズを実行してもよい。

10

特に本実施形態のようにサブ制御基板80でAT遊技の抽選を行う場合には、メイン制御基板50側ではその抽選結果を知らないで、後者のように、特殊リプレイCの入賞時には一律にフリーズを実行することとなる。仮に、メイン制御基板50でAT遊技の抽選を行う場合には、AT遊技を開始することとなる特殊リプレイCの入賞時にのみフリーズを実行することができる。

なお、上記のフリーズの実行中は、サブ制御基板80の制御により、メニューキー85の操作によってメニュー画面を表示することはできない。

【0110】

なお、以上のリプレイ重複当選1及び2において、ノーマルリプレイが停止する有効ラインを、当選役（条件装置）ごとに異ならせてもよく、同一の有効ラインであってもよい。あるいは、一部のリプレイの重複当選時は平行の有効ラインに停止させ、他のリプレイの重複当選時は斜めの有効ラインに停止させるようにしてもよい。また、いずれかのリプレイ重複当選時に、ストップスイッチ42の押し順によって、ノーマルリプレイが入賞する有効ラインを異ならせてもよい。

20

【0111】

また、チェリー、スイカ、ノーマルリプレイ、及びベルリプレイの当選時は、ストップスイッチ42の押し順にかかわらず、これらの当選役が入賞可能である。ただし、これらの役の当選時の遊技であっても、押し順非報知時の第一停止中又は右の操作は、ペナルティとなる。

さらにまた、ノーマルリプレイ及びベルリプレイは、ストップスイッチ42の操作タイミングにかかわらず常に入賞させることができるが、チェリー及びスイカは、対象図柄が有効ラインに停止するタイミングでストップスイッチ42を操作すること（目押し）が必要となる。

30

【0112】

さらに、MB遊技中及びCB遊技中は、いずれかの小役に対応する図柄の組合せが有効ラインに停止するようにリール31を停止制御する。

また、役の非当選時は、いずれの役に対応する図柄の組合せも有効ラインに停止しないようにリール31を停止制御する。

【0113】

さらにまた、MB内部中遊技でいずれかの小役の当選となったときは、まず、当該遊技で当選した小役の入賞を優先し、その小役を入賞させることができないときは、次に、当選を持ち越しているMBを入賞可能に制御し、さらにMBも入賞させることができないときは、いずれの役に対応する図柄の組合せも有効ラインに停止させないように制御する。

40

【0114】

たとえばMB内部中遊技においてチェリーに当選した場合において、第一停止左でストップスイッチ42が操作されたときに、「チェリー」図柄を有効ラインに停止可能であれば「チェリー」図柄を有効ラインに停止させ、チェリーを入賞させる。また、第一停止左であれば、その時点でMBの非入賞が確定する。さらにまた、第一停止左でストップスイッチ42が操作されたときに「チェリー」図柄を有効ラインに停止させることができなかったときは、当該遊技ではチェリーもMBも入賞しない。

50

【0115】

同様に、MB内部中遊技においてスイカに当選した場合において、第一停止左でストップスイッチ42が操作されたときに、「スイカ」図柄を有効ラインに停止可能であれば「スイカ」図柄を有効ラインに停止させる。第二、第三停止時においても同様に、「スイカ」図柄を有効ラインに停止可能であれば「スイカ」図柄を有効ラインに停止させ、スイカを入賞させるように制御する。

また、第一停止左であれば、その時点でMBの非入賞が確定する。さらにまた、ストップスイッチ42が操作されたときに「スイカ」図柄を有効ラインに停止させることができなかったときは、第一停止左の遊技ではスイカもMBも入賞しない。

【0116】

さらにまた、MB内部中遊技において、ベル重複当選となったときは、ベルAは「PB=1」であるから、当該遊技では常にベルAが入賞し、MBが入賞する場合はない。

同様に、MB内部中遊技において、いずれかのリプレイの単独当選又は重複当選となったときは、まず、当該遊技で当選したリプレイの入賞を優先し、そのリプレイを入賞させることができないときは、次に、当選を持ち越しているMBを入賞可能に制御し、さらにMBも入賞させることができないときは、いずれの役に対応する図柄の組合せも有効ラインに停止させない停止位置決定テーブル65が用いられる。

ここで、本実施形態のリプレイは「PB=1」であるから、当該遊技では常にリプレイが入賞し、MBが入賞する場合はない。

【0117】

説明を図1に戻す。

メインCPU54の入賞判定手段63は、すべてのリール31の停止時に、いずれかの役に対応する図柄の組合せが有効ラインに停止したか否かを判断する。入賞判定手段63は、たとえばインデックスセンサ33がインデックスを検知してからのモータ32のステップ数を検知することにより、有効ライン上の図柄を判断する。ただし、入賞判定手段63は、ストップスイッチ42が操作され、リール31の停止位置が決定された時に、そのリール31が停止したか否かにかかわらず、停止図柄を判断することが可能である。

【0118】

払出し手段64は、すべてのリール31の停止時に、いずれかの役に対応する図柄の組合せが有効ラインに停止したと判断され、その役の入賞となったときに、その入賞役に応じて所定枚数のメダルを遊技者に対して払い出す。払出しは、上述したように、貯留枚数として加算するか、又は貯留枚数が「50」を超えるときは実際にメダルを払出し口から払い出す。メダルを実際に払い出すときは、ホッパーモータ36を駆動制御して、所定枚数のメダルを払い出す。メダルの払出し時には、払い出されたメダルを払出しセンサ37により検知し、正しく払い出されたか否かをチェックする。

【0119】

また、サブ制御基板80は、上述したランプ21、スピーカ22、及び画像表示装置23からの演出の出力を制御する。

サブ制御基板80は、遊技ごとに、遊技の開始時に、役抽選手段61による役の抽選結果（メイン制御基板50側から送信された役抽選結果の情報）に基づいて、ソフトウェア乱数を用いた抽選によって、演出を選択する。

【0120】

具体的には、遊技の進行に伴って、どのようなタイミングで（スタートスイッチ41の操作時や各ストップスイッチ42の操作時等）、どのような演出を出力するか（ランプ21をどのように点灯、点滅又は消灯させるか、スピーカ22からどのようなサウンドを出力するか、及び画像表示装置23にどのような画像を表示させるか等）を選択する。そして、この選択に従って、演出を出力する。

したがって、これらのランプ21、スピーカ22、及び画像表示装置23からの演出の出力は、サブ制御基板80側で制御されるものであり、メイン制御基板50側で制御されるものではない。

10

20

30

40

50

【0121】

なお、上述のベット枚数表示装置47a及び貯留枚数表示装置47bは、メインCPU54によってその表示が制御されるものである。

これに対し、画像表示装置23の画像表示領域内に、現在のベット枚数及び貯留枚数を画像表示する場合があるが、この画像表示については、サブCPU84によって制御される。

【0122】

また、図1において、サブ制御基板80は、AT制御手段91を備える。AT制御手段91は、AT（サブボーナス）遊技を実行するか否かを抽選で決定するとともに、この抽選に当選したときは、リプレイ重複当選3時には、逆押しすべきこと、及び「赤7」揃いを狙うべきことを報知する。そして、特殊リプレイCが入賞すると、次遊技から、AT技を開始する。

AT遊技は、所定の終了条件を満たすまで継続される。終了条件としては、たとえば払出し枚数が所定枚数（たとえば300枚）に到達するまで、又はベル重複当選が30回になるまで、等が挙げられる。

AT遊技中は、ベル重複当選時に、9枚の払出しとなる押し順が報知される。

【0123】

図7は、メイン制御基板50の入力ポート51及び出力ポート52と、各周辺機器の信号の入出力との関係を示す図である。

図7に示すように、入力ポート51として、入力ポート51a、51b、51c、・・・を備え、出力ポート52として、出力ポート52a、52b、52c、・・・を備える。本実施形態の入力ポート51及び出力ポート52は、D0～D7の8ビットが入力又は出力可能な1バイトのポートである。

【0124】

図7において、「正」、「負」とは、入力又は出力される信号を意味し、たとえば入力ポート51aのD0ビットにおける精算スイッチ信号では、操作されていないときにはD0ビットに「正（high）」の信号が入力され、操作されたときはD0ビットに「負（low）」の信号が入力される。一方、たとえば3ベットスイッチ信号では、操作されていないときにはD2ビットに「負（low）」の信号が入力され、操作されたときはD2ビットに「正（high）」の信号が入力される。

【0125】

また、図7中、信号の入力又は出力を図示していないポート（たとえば、入力ポート51aのD4）は、未使用であるか、又は本実施形態において説明を省略する信号の入出力ポートを意味する（信号の入出力がないポートは、すべて未使用という意味ではない）。

【0126】

さらにまた、入力ポート51aにおいて、精算スイッチ信号及び1ベットスイッチ信号は、そのスイッチの操作時にいずれも操作時に負の信号が入力されるが、3ベットスイッチ信号は、そのスイッチの操作時に正の信号が入力される。これは、本実施形態では、精算スイッチ46及び1ベットスイッチ40aと、3ベットスイッチ40bとで、異なるスイッチを用いているためである。

【0127】

たとえば、精算スイッチ46及び1ベットスイッチ40aは、押されること（あるいは押圧）に応じて接地／非接地を検出するボタンスイッチであり、押圧された場合に電流が流れる回路を備えるスイッチである。これに対し、3ベットスイッチ40bは、フォトセンサを内蔵したボタンスイッチであり、フォトセンサの受光素子が遮光片により発光素子側の光を遮光したときに電圧が変化するスイッチである。

このように、設計上の都合に応じて、同じベットスイッチ40であっても異なる構造のスイッチを採用している。なお、これに限らず、同一構造のスイッチを用いることも、もちろん可能である。

【0128】

図7に示すように、入力ポート51bのD0ビットには、電断信号（電断が発生したときに出力される信号）が入力される。また、D1～D5ビットには、それぞれメダル払出し装置の所定の信号が入力される。そして、割込み処理時ごとに、これらの信号に基づくデータが、後述する図8において、7E02番地のデータとして記憶される。

さらに、出力ポート52bにおいては、D6ビットからは、ブロック45の信号が出力される。また、D7ビットからは、ホッパーモータ36を駆動するための信号が出力される。これらの信号の出力は、後述する図8において、7E03番地のそれぞれD6及びD7ビットのデータに基づき行う。

【0129】

図8は、本実施形態において、メモリ53（本実施形態ではRWM）に記憶されるデータの一部を示す図である。

10

まず、図8において、7E00番地の入力ポート51aレベルデータとは、入力ポート51aに入力されているデータを示している。また、「7E00」番地とは、メモリ53内における入力ポート51aレベルデータのアドレスを示しており、入力ポート51aと同様に、D0ビット～D7ビット（合計8ビット）からなる1バイトデータであり、各ビットの値は、「0」又は「1」である（以下の7E01番地以降も同様である）。このように、本実施形態では、1バイトで8個のスイッチのデータ（8ビットのデータ列）を記憶できるように構成されている。

【0130】

図7で示したように、入力ポート51aのD0～D7（ただし、D4を除く）には、正又は負の信号が入力されるが、入力ポート51aに入力された信号を取得した後、正論理又は負論理に変換（演算）した後のデータを、入力ポート51aレベルデータとして、7E00番地に記憶する。たとえば、入力ポート51aにおいて、D0、D1、D3ビットには、操作時には負の信号が入力され、入力された信号が「負」の場合は「1」、「正」の場合は「0」となるように記憶する。

20

【0131】

一方、入力ポート51aにおいて、D2、D5、D6、D7ビットには、操作時には正の信号が入力されるので、入力された信号が「正」の場合は「1」、「負」の場合は「0」となるように、7E00番地に記憶する。

【0132】

30

上記の論理変換処理をより具体的に説明する。

まず、定義データとして、以下の定義データ1及び定義データ2を備える。これらの定義データ1及び定義データ2は、メモリ53のROMに記憶されている。

定義データ1：0，0，0，0，1，0，1，1

定義データ2：1，1，1，0，1，1，1，1

そして、たとえば入力ポート51aに信号が入力されると仮定する。ここでは、1ビットスイッチ40aがオンにされたものとする。1ビットスイッチ信号は、オンのときは負の信号が入力されるので、入力される信号は「0」である。また、入力ポート51aにおける他のスイッチは操作されないものとする。精算スイッチ信号（D0ビット）及びスタートスイッチ信号（D3ビット）は「1」、その他は「0」であるので、

40

(D7, D6, D5, D4, D3, D2, D1, D0) = (0, 0, 0, 0, 1, 0, 0, 1) : データX

となる。

このデータXは、たとえばメモリ53内のAレジスタに記憶される。

なお、Aレジスタ、及び以下で登場するB、C、・・・レジスタは、いずれも、メモリ53内に設けられ、データを一時記憶する回路である。

【0133】

次に、上記データXと、定義データ1とを排他的論理和（XOR；エクスクルーシブオア）による論理演算を行う。これにより、

(D7, D6, D5, D4, D3, D2, D1, D0) = (0, 0, 0, 0, 0, 0, 1, 0) : データY

となり、操作されたスイッチに係るデータ（D1ビット）のみが「1」となる。

50

したがって、上記のように演算することで、1ベットスイッチ40aのみが操作されたことを検知することができる。

【0134】

また、入力ポート51aのD4ビットのように未使用ビットを有する場合には、この未使用ビットに対応するデータを確実に「0」にするために、上記のデータYと定義データ2とを「AND」演算（論理積）してもよい。これにより、

(D7, D6, D5, D4, D3, D2, D1, D0) = (0, 0, 0, 0, 0, 0, 1, 0) : データZ
となる。

なお、未使用ビットを有さないときは、データYの算出までで十分であり、データZの算出は不要である。また、未使用ビットを有する場合であっても、データYの算出で終了することも可能であり、データZの算出は、ノイズをより確実に除去するために行われる。

10

なお、上記の演算、記憶等の処理は、1割込み内で行われる。

【0135】

後述する図12に示すように、遊技を進行する情報処理として、1遊技あたり1回行うメインループないしメイン処理(M_MAIN)が設けられている。メインループは、投入されたメダルの検知から、全リール31が停止して入賞処理が行われるまでの処理である。

【0136】

このメインループ中に、メインループを一旦抜けて、割込み処理として、入力ポート51aを検知する処理を実行し、その処理の実行後、再度、メインループに戻る処理を定期的に行っている。その割込み時間の間隔は、本実施形態では2.235msである。すなわち、2.235ms間隔の割込み処理ごとに、入力ポート51aのデータを取得する。そして、取得したデータに基づいて、入力ポート51aレベルデータ及び入力ポート51a立ち上がりデータを生成し、記憶する。したがって、これらのデータは、2.235msごとに更新されていく。

20

ただし、図30のステップS290及びステップS291の処理が行われたときは、1ベットスイッチ信号及び3ベットスイッチ信号の入力ポート立ち上がりデータを「0」に更新する場合がある。

【0137】

また、割込み処理がいつ行われたかにかかわらず、D0～D7ビットのすべてを検知する。たとえば、リール31の回転中（ストップスイッチ42が操作される前）は、遊技者によってベットスイッチ40が操作されることはあり得ないので、必ずしも入力ポート51aのD1及びD2ビットのデータを取得する必要はないが、本発明では、すべてのビットのデータを取得する。そして、全ビットのデータを取得すれば、エラー判定を行うことも可能となる。たとえばリール31の回転中にベットスイッチ40の入力がオフからオンになった場合が挙げられる。

30

また、7E00番地のデータ、すなわち一の格納領域における1バイトのデータ列を取得すれば、すべての操作スイッチのオン／オフの状況を知ることができる。

【0138】

なお、詳細は後述するが、メインループ内で、割込み処理を行わせたくない場合がある。たとえば、A処理とB処理とを一連で行いたいときには、A処理とB処理との間には割込み処理を入れたくない事情がある。この場合には、A処理の前に割込み禁止処理を入れ、B処理後に割込み許可処理を入れる。

40

【0139】

以上のようにして、2.235msごとの割込み処理で、入力ポート51aの各ビットの信号（「正」又は「負」）を取得し、正論理又は負論理演算によって「0」又は「1」のデータに変換し、記憶する。これが、図8に示す入力ポート51aレベルデータ（8ビットのデータ列である1バイトデータ）である。

さらに、今回の割込み処理でレベルデータを生成したとき、その1つ前（前回）、すなわち2.235ms前に生成したレベルデータを保持しておく。

50

【0140】

そして、前回のレベルデータと今回のレベルデータとを「XOR」演算する。なお、「XOR」演算では、前回及び今回「0」、又は前回及び今回「1」の場合、すなわち前回と今回で変化なしの場合は「0」となり、前回「0」で今回「1」、あるいは前回「1」で今回「0」のときは、「1」となる。さらに、この「XOR」の演算結果と、今回のレベルデータとを「AND」演算し、今回のレベルデータが「1」であるときは「1」とし、今回のレベルデータが「0」であるときは「0」とする。この値が入力ポート51a立ち上がりデータ（8ビットのデータ列である1バイトデータ）となり、図8に示す7E01番地に記憶する。

【0141】

すなわち、入力ポート51aレベルデータは、その割込み処理時に、実際の操作の有無を示す操作スイッチのデータであり、操作時が「1」、非操作時が「0」で示されるデータである。

これに対し、入力ポート51a立ち上がりデータは、前回の割込み処理には非操作と検知されたが、今回の割込み処理では操作されたと検知された操作スイッチのデータであり、前回の割込み処理時から変化のあった操作スイッチが「1」、変化のなかった操作スイッチが「0」で示されるデータである。

【0142】

図8に示すように、入力ポート51aのレベルデータ及び立ち上がりデータとして、1バイト（8ビット）内に、精算スイッチ46、1ビットスイッチ40a、3ビットスイッチ40bの各信号のデータが記憶される。これにより、1つの番地、たとえば7E00番地のデータを読み込むことにより、精算スイッチ46、1ビットスイッチ40a、3ビットスイッチ40bのデータを同時に読み込むことができるので、ビット処理や精算処理をスムーズに実行できるようになる。

なお、図8に示すように、入力ポート51aレベルデータの番地（7E00）と、入力ポート51a立ち上がりデータの番地（7E01）とは、異なっている。

【0143】

また、図8に示すように、7E03番地には、ブロック信号及びホッパーモータ信号のオン／オフ状態を示すデータを記憶する。図7に示すように、出力ポート52bからは、ブロック信号及びホッパーモータ信号が出力されるが、7E03番地のデータを参照し、これらの信号を出力するか否かを決定する。7E03番地のデータが「1」の場合には、正論理によりオン（出力）となり、「0」の場合には、オフ（出力なし）となる。

【0144】

ここで、ブロック45は、ブロック信号が「1」（オン）であるときは、メダル投入口43とホッパーとを連結するメダル通路を形成し、「0」（オフ）であるときは、メダル投入口43と払出し口とを連結する通路（返却通路）を形成する。

【0145】

さらにまた、7E04番地には、スタートスイッチ受け付け状態、ブロック信号状態、メダル限界判定を含むデータを記憶する（その他は説明を省略する）。スタートスイッチ受け付け状態としては、受け付け可能時は「1」、受け付け不可時は「0」が記憶される。

また、ブロック信号状態は、上記のブロック信号と同義であるが、制御上、別個に使用されるために、別個の番地に記憶しておくデータである。このデータは、メダル管理時等において、ブロック45が現在どのようなになっているかを知るための制御で用いるものである。

【0146】

ブロック45の駆動時には、割込み処理によって、出力ポート52bのD6ビット（図7参照）からブロック信号を出力する。

なお、ブロック信号状態が入力ポート51に入力されたり、出力ポート52から出力されたりすることはない。ブロック信号とブロック信号状態とを分けて記憶することにより、出力に用いられる番地（7E03）と、出力に用いられない番地（7E04）とを分け

10

20

30

40

50

ている。

【0147】

以上のようにして、1割込み内で、入力ポート51a、51bの信号に基づくデータを記憶するとともに、出力ポート52bに対しては、7E03番地のデータに基づき信号を送信する。

【0148】

次に、入力ポート51aの立ち上がりデータの生成について、具体例を挙げて説明する。

図9は、立ち上がりデータの生成の第1例を示す図である。

図9では、3ベットスイッチ40bがオンされた状態で、1ベットスイッチ40aがオンされた場合の例であって、正常にデータを取得した例である。なお、実際の遊技では、遊技者が意図的に操作しない限り、双方のベットスイッチ40が同時にオンされるケースはない。

【0149】

先ず、図9(A)に示すように、今回の割込み処理において、入力ポート51aの信号を取得し、オンが「1」、かつオフが「0」となるように論理変換したデータを取得し、それをCレジスタに記憶する。なお、以下の説明では、立ち上がりデータの生成時に用いるレジスタとして、Aレジスタ～Cレジスタを備えるものとする。

【0150】

図9(A)において、1ベットスイッチ40a及び3ベットスイッチ40bがオンであるので、入力ポート51aのD1及びD2ビットのデータが「1」となり、他は「0」となる。この状態のデータをAレジスタに記憶する。

次に、同図(B)に示すように、前回の割込み処理時に生成したレベルデータを、Bレジスタに記憶する。ここで、割込み処理時に生成したレベルデータは、次の割込み処理時に使用する。

【0151】

次に、同図(C)に示すように、今回の割込み処理において、入力ポート51aの信号を再度取得し、Aレジスタに記憶する。そして、図10の例では、正常にデータを取得した例であるので、(A)のデータと(C)のデータとは、一致している(D1及びD2ビットデータが「1」)。

【0152】

割込み処理が開始されると、先ず、図9(A)に示す1回目のデータ取得を行い、50μS(マイクロ秒)の経過後、2回目のデータ取得を行う。なお、この50μS間にも、割込み処理時における所定の処理、たとえばタイマ減算処理等を実行している。また、割込み処理で実行される処理の時間は、500μS(0.5ms)程度である。

【0153】

次に、同図(D)に示すように、CレジスタとAレジスタの各データとを「AND」演算(論理積)し、その生成したデータを入力ポートレベルデータとする。CレジスタとAレジスタとを「AND」演算すれば、両者ともデータが「1」である場合に限り「1」となる。

【0154】

次に、同図(E)に示すように、Bレジスタのデータと、(D)で生成したデータすなわち入力ポートレベルデータとを「XOR」演算(排他的論理和)し、変化ビットデータを生成する。これにより、変化があったデータのみが「1」となる。この例では、D1ビットのデータのみが「1」となる。

【0155】

次に、同図(F)に示すように、(D)で生成したデータすなわち入力ポートレベルデータと、(E)で生成した変化ビットデータとを「AND」演算し、その生成したデータを入力ポート立ち上がりデータとする。これにより、変化があったD1ビットのデータのみが「1」となる。

10

20

30

40

50

この立ち上がりデータにより、1ビットスイッチ40aの立ち上がりデータは生成される（「1」となる）が、3ビットスイッチ40bの立ち上がりデータは生成されない（「0」となる）ことを意味する。

【0156】

特に、図9の例では、3ビットスイッチ40bがオンの状態で1ビットスイッチ40aがオンされた場合であるので、3ビットスイッチ40bのオン／オフは変化がなく、1ビットスイッチ40aのみがオフからオンに変化している。この変化を示すデータが、入力ポート立ち上がりデータとなり、D1ビットすなわち1ビットスイッチ信号のみが「1」（オン）となっている。

【0157】

また、3ビットスイッチ40bがオンの状態で1ビットスイッチ40aがオンされた場合には、3ビットスイッチ40b及び1ビットスイッチ40aの双方がオンであるので、入力ポートレベルデータでは、D1及びD2ビットの双方のデータが「1」となっている。

【0158】

図10（第2例）は、図9と同一条件（3ビットスイッチ40bがオンされた状態で、1ビットスイッチ40aがオンされた場合）であるが、異常なデータを取得した場合の例である。

先ず、図10（A）に示すように、今回の割込み処理において、入力ポート51aの信号を取得し（1回目）、オンが「1」、かつオフが「0」となるように論理変換したデータを取得し、それをCレジスタに記憶する。1回目のデータ取得は、正常なデータを取得した例である。

【0159】

次に、同図（B）に示すように、前回の割込み処理時に生成したレベルデータを、Bレジスタに記憶する。このレベルデータは、図9（B）と同一である。

次に、同図（C）に示すように、今回の割込み処理において、入力ポート51aの信号を再度取得する（2回目）。このときに、ノイズ等が発生し、D1ビットにおける1ビットスイッチ40aの信号を「正」と読み取り、AレジスタのD1ビットにはデータを「0」として記憶した例である。

【0160】

次に、同図（D）に示すように、CレジスタのデータとAレジスタのデータとを「AND」演算し、その生成したデータを入力ポートレベルデータとする。CレジスタのデータとAレジスタのデータとを「AND」演算すれば、両者ともデータが「1」である場合に限り「1」となるので、D1ビットは「0」、D2ビットは「1」となる。この（D）の処理により、1回目の取得データと2回目の取得データとで異なる場合には、データを「0」にすることができる。したがって、チャタリングやノイズがあった場合には、そのデータを「0」にすることができる。

【0161】

次に、同図（E）に示すように、Bレジスタのデータと、（D）で生成した入力ポートレベルデータとを「XOR」演算し、変化ビットデータを生成する。これにより、変化があったデータのみが「1」となるが、この例では、全ビットのデータが「0」となる。

【0162】

次に、同図（F）に示すように、（D）で生成した入力ポートレベルデータと、（E）で生成した変化ビットデータとを「AND」演算し、その生成したデータを入力ポート立ち上がりデータとする。これにより、変化があったビットのデータのみが「1」となるが、この例では全ビットのデータが「0」となる。

【0163】

以上より、変化があったビットのデータが「1」となるが、この例のように、1回目の取得データと2回目の取得データとが異なっていた場合には、全ビットのデータを「0」とすることができる。

10

20

30

40

50

このように、1回の割込み処理でデータを2回（複数回）取得する場合には、取得した双方のデータが一致している場合のみレベルデータを記憶し、一致していない場合には、そのレベルデータを記憶しない（「0」とする）。

【0164】

なお、1回目と2回目の取得データが不一致のときは、前回の入力ポートレベルデータを採用するようにしてもよい。

すなわち、図9及び図10の例では、1回目と2回目の取得データが不一致のときは、当該ビットのデータについては常に「0」となる演算を行ったが、1回目と2回目の取得データが不一致のときは、当該ビットのデータについては前回割込み時の値としてもよい。

【0165】

ここで、前回割込み時のデータを用いることにより、以下の効果を有する。

たとえば図9及び図10の例において、前回割込み時のデータが「1」であり、今回割込み時には、1回目と2回目の取得データが不一致であったために「0」になったと仮定する。そうすると、前回割込み時のデータが「1」、今回割込み時のデータが「0」となり、変化ビットのデータ（「1」から「0」に変化する、立ち下がり）を生成してしまう。ここで、立ち下がりデータを有するときに所定の制御（たとえば設定変更）を実行する場合がある。よって、チャタリングやノイズが入ったために、立ち下がりデータを生成し、意図しない制御を実行してしまうおそれがある。

【0166】

これに対し、前回割込み時のデータを用いるようにすれば、チャタリングやノイズによって立ち上がりデータ及び立ち下がりデータのいずれも生成しないので、立ち上がり／立ち下がりデータに基づく意図しない処理が実行されてしまうことはない。

【0167】

図11は、図8と同様に、メモリ53に記憶されるデータの他の一部を示す図である。

図11において、7E11番地のデータは、役の抽選結果に係る作動状態フラグを記憶するためのデータである。図11に示すように、D0、D1、D2の各ビットには、それぞれリプレイ（再遊技役）、MB（2種BB）、CBが割り当てられており、たとえばリプレイ作動時は、D0ビットのみが「1」（オン）となる。

【0168】

本実施形態では、遊技状態に応じて、ベット可能なメダル枚数が異なるが、7E11番地のデータの作動状態フラグを判別することで、遊技状態（ベット可能なメダル枚数）を判別可能となる。また、リプレイ作動時とリプレイ非作動時とで、たとえばメダルの精算処理が異なるので、当該処理時に、リプレイの作動状態フラグをチェックする。

【0169】

また、7E12番地の自動ベット数データ（_NB_REP_MEDAL）とは、リプレイが入賞したときに、自動ベットするメダル枚数を定めるデータであり、たとえば通常遊技では「3」であり、MB遊技中及びCB遊技中は「2」である。

さらにまた、7E13番地のベット数データ（_NB_PLAY_MEDAL）とは、当該遊技のベット数を定めるデータである。7E13番地のベット数データは、前遊技のベット数を意味し、毎遊技、参照される。

【0170】

7E14番地の制御コマンド読み込みポインタ（_NB_CMDREAD）とは、メイン制御基板50からサブ制御基板80に対して制御コマンドを送信するときに、どの制御コマンドバッファ（後述する7E16～7E35番地）に記憶されたデータを読み込むか、すなわち読み取り番地を指定するデータを記憶するものである。制御コマンド読み込みポインタ及び後述する制御コマンド書き込みポインタは、いずれも、「0」～「FF」（H；16進数）、すなわち「0」～「255」（10進数）の範囲の値を記憶する。

【0171】

なお、以下の説明において、「H」は、16進数（ヘキサデシマル）という意味で使用

10

20

30

40

50

し、「B」は、2進数（バイナリ）という意味で使用する。

7 E 1 5 番地の制御コマンド書き込みポインタ（_NB_CMDSTORE）とは、メインCPU 5 4が制御コマンドを記憶するときに、どの制御コマンドバッファ（7 E 1 6～7 E 3 5 番地）に記憶するか、すなわち書き込む番地を指定するデータを記憶するものである。

【0172】

また、7 E 1 6～7 E 3 5 番地は、いずれも、制御コマンドバッファの記憶領域であり、制御コマンドを記憶しておくためのものである。

制御コマンドバッファは、上述した入力ポートレベルデータ等と同様に、8ビットの1バイトデータである。特に、本実施形態の制御コマンドは、一对の第1制御コマンドと第2制御コマンドとからなる。たとえば、図11において、7 E 1 6 番地の「第1制御コマンド0」と、7 E 1 7 番地の「第2制御コマンド0」とが一对となる。

10

このため、7 E 1 6～7 E 3 5 番地は、「0」～「15」の第1及び第2制御コマンドを記憶できるように構成されている。したがって、一对の第1制御コマンド及び第2制御コマンドを、16個まで記憶可能である。

【0173】

続いて、本実施形態における情報処理の流れを、フローチャートに基づき説明する。

図12は、本実施形態におけるメインループ（M_MAIN）処理を示すフローチャートである。そして、このメインループの処理の実行中に、2. 235msごとに割込み処理を行い、入力ポートレベルデータ、及び入力ポート立ち上がりデータを生成し、記憶する。

このメインループと別に実行される割込み処理については後述する。

20

【0174】

図12において、ステップS101では、スタックポインタをセットする。ここで、スタックポインタとは、電断が生じた場合に、電断発生時のデータを保存するRWM領域を指し、スタックポインタのセットとは、そのRWM領域のレジスタ値を初期値にセットする処理である。

次のステップS102では、遊技開始セット（MS_GAME_SET）を行う。ステップS102に進むと、後述する図13の処理に移行する。

【0175】

次のステップS103ではメダルの読み込みを行う。この処理は、現時点においてベットされているメダル枚数が何枚であるかを読み込む処理である。以下の説明において、「メダル読み込み」とは、ベット枚数の読み込みを指し、貯留枚数の読み込みとは異なるものである。

30

次のステップS104では、ステップS103で読み込んだベット枚数に基づき、ベットメダルの有無を判断する。

【0176】

ステップS104でベットメダルありと判断したときはステップS106に進み、ベットメダルなしと判断したときはステップS105に進んでメダル投入待ち表示（メダルの投入可能を示す表示）を行い、ステップS106に進む。

【0177】

ステップS106では、投入されたメダルの管理処理（MS_MEDAL_CHK）を行う。ステップS106に進むと、後述する図25の処理に移行する。

40

次のステップS107では、ソフト乱数の更新処理を行う。この処理は、役抽選手段61で使用する乱数を更新（「1」加算）する処理である。ソフト乱数は、0～65535の範囲を有する16ビット乱数であり、更新時には、更新前の値に、割込みカウント値（割込み時にインクリメントされるカウント値）を加算する処理を実行する。

【0178】

次のステップS108では、スタートスイッチ41の受付けフラグをセットする（オン（「1」）にする処理を行う）。このフラグは、図8中、7 E 0 4 番地のD0ビットのデータであり、このデータが「1」（オン）であるときにスタートスイッチ41が操作されると、その操作は有効となる。

50

次のステップS109では、メインCPU54は、スタートスイッチ41が操作されたか否かを判断する。スタートスイッチ41が操作されたと判断したときは、ステップS110に進み、スタートスイッチ41が操作されていないと判断したときはステップS103に戻る。

【0179】

ステップS110では、役抽選手段61は、スタートスイッチ41が操作されたタイミングで、すなわちスタートスイッチ41の操作信号の受信時に、乱数値を抽出し、役の抽選を実行する。

次のステップS111では、リール制御手段62は、リール31の回転を開始する。次にステップS112に進み、リール31の停止受け付けをチェックする。ここでは、ストップスイッチ42の操作信号を受信したか否かを検知し、操作信号を受信したときは、役の抽選結果とリール31の位置とに基づいてリール31の停止位置を決定し、その決定した位置にリール31を停止させるように制御する。

【0180】

次のステップS113では、リール制御手段62は、全リール31が停止したか否かをチェックし、ステップS114に進む。ステップS114では、全リール31が停止したか否かを判断し、全リール31が停止したと判断したときはステップS115に進み、全リール31が停止していないと判断したときはステップS112に戻る。

【0181】

ステップS115では、図柄の表示判定を行う。ここでは、入賞判定手段63により、有効ラインに、役に対応する図柄の組合せが停止したか否かを判断する。

次にステップS116に進み、役の入賞があったときは、払出し手段64は、入賞役に対応するメダルの払出しを行う。

【0182】

次にステップS117に進み、メインCPU54は、遊技終了チェックを行う。この処理は、役抽選結果によって立てた当選フラグのクリア、ウェイトのセット等を実行する処理である。なお、図11において、7E11番地の作動状態フラグは、当該遊技の開始時（役抽選後）に、当選役に応じてオンとなり、このステップS117においてオフにされる。

【0183】

次のステップS118では、遊技終了時の出力要求をセットする。この処理は、1遊技が終了した旨をサブ制御基板80に送信するための制御コマンドデータをセットする処理である。なお、「出力要求のセット」は、以下に説明する処理においてもしばしば実行されるものであるが、このステップS118と同様に、サブ制御基板80に送信するための制御コマンドデータをセットする処理を意味する。

【0184】

次にステップS119に進み、メインCPU54は、制御コマンドセット1を実行する。この処理は、上述した制御コマンドバッファ（7E16～7E35番地のいずれか）に、サブ制御基板80に送信するための制御コマンドデータを記憶する処理である。なお、「制御コマンドセット1」は、以下に説明する処理においてもしばしば実行されるものであるが、このステップS119と同様に、サブ制御基板80に送信するための制御コマンドデータを記憶する処理を意味する。また、制御コマンドセット1の具体的処理内容については、後述する（図19）。

そして、ステップS119の処理を終了すると、再度、ステップS101に戻る。

【0185】

図13は、図12のステップS102における遊技開始セット（MS_GAME_SET）を示すフローチャートである。

まず、ステップS121では、メインCPU54は、遊技状態のセット処理（MS_ACTION_SET）を行う。この処理は、遊技の開始前にリプレイ作動時やMB遊技、CB遊技であることをセットする処理である。次のステップS122では、メインCPU54は、遊技

10

20

30

40

50

状態出力 (MS_STATUS_SET) を行う。この処理は、サブ制御基板 80 に対し、遊技状態に係るコマンドを送信する処理である。

【0186】

ステップ S 1 2 3 では、メイン CPU 54 は、作動状態フラグをチェックする。作動状態フラグのチェックは、上述した 7 E 1 1 番地の作動状態フラグをオン／オフを判断することである。より具体的には、本実施形態では、次のステップでリプレイの作動状態であるか否かを判断するために、7 E 1 1 番地の D 0 ビットのオン (「1」) / オフ (「0」) を判断する。

【0187】

次にステップ S 1 2 4 に進み、ステップ S 1 2 3 におけるチェックに基づいて、リプレイ作動時であるか否かを判断する。ここで、リプレイ作動時であると判断されるときは、リプレイの入賞に基づいてメダルが自動ベットされているときに相当する。

ステップ S 1 2 4 でリプレイ作動時であると判断したときはステップ S 1 2 5 に進み、リプレイ作動時でないと判断したときはステップ S 1 2 7 に進む。

【0188】

ステップ S 1 2 5 では、メダル貯留枚数の読み込み処理 (S_CREDIT_READ) (図 1 6) を行う。次のステップ S 1 2 6 では、貯留枚数が限界枚数に達しているか否かを判断する。本実施形態では、上述したように、貯留 (クレジット) メダルは 50 枚まで可能であり、現時点で貯留枚数が 50 枚であるときは、貯留限界枚数であると判断する。貯留限界枚数であると判断したときはステップ S 1 3 0 に進み、貯留限界枚数でないと判断したときはステップ S 1 2 7 に進む。

【0189】

ステップ S 1 2 7 では、ブロック 45 をオンにする処理 (MS_BLOCKER_ON) (図 1 7) を実行する。ここで、ブロック 45 がオンであるときは、メダル通路を形成するときである。すなわち、貯留限界枚数に達していないときは、その後のメダルの手入れによる貯留を許可するためブロック 45 をオンにするが、貯留限界枚数に達しているときはその後のメダルの手入れを不許可にするため、ブロック 45 のオフを維持する (図 1 3 のステップ S 1 2 7 の直前では、ブロック 45 がオフになっているものとする)。

なお、ブロック 45 のオン／オフにはある程度の時間を要するので、最初はオフの状態を維持しておき、メダルの手入れを許可するときにはオフからオンとなるように設定している。

【0190】

次のステップ S 1 2 8 では、ステップ S 1 2 3 と同様に、作動状態フラグをチェックし、次のステップ S 1 2 9 では、リプレイ作動時であるか否かを判断する。リプレイ作動時であると判断したときはステップ S 1 3 0 に進み、リプレイ作動時でないと判断したときは本フローチャートによる処理を終了する。

【0191】

ステップ S 1 3 0 では、リプレイ表示の LED (リプレイ作動時に自動ベット有であることを示すために点灯する LED) の点灯処理を行う。そして、次のステップ S 1 3 1 では、サブ制御基板 80 に対し、自動ベットが行われたことを示す制御コマンドを送信するための出力要求セット処理を行い、次のステップ S 1 3 2 で制御コマンドセット 1 処理を実行する。

【0192】

次のステップ S 1 3 3 では、メダルを 1 枚ベットする処理、すなわち、リプレイの入賞に基づき、3 枚又は 2 枚の自動ベットを行うために、1 枚ずつメダルの自動ベット処理を行う (MS_MEDAL_INC) (図 2 2)。次のステップ S 1 3 4 では、メダル限界枚数になったか否か (たとえば通常遊技時では、3 枚) を判断し、メダル限界枚数になったと判断したときは本フローチャートによる処理を終了する。一方、メダル限界枚数になっていないと判断したときはステップ S 1 3 3 に戻り、メダル 1 枚の自動ベット処理を継続する。

なお、上記処理では、自動ベット時の出力要求処理を行った後 (自動ベット時の制御コ

10

20

30

40

50

マンドの送信後)、自動ベット処理を行うが、これに限らず、自動ベット処理の終了後に、自動ベット時の出力要求処理(自動ベット時の制御コマンドの送信)を行ってもよい。

【0193】

図14は、図13のステップS121における遊技状態のセット処理(MS_ACTION_SET)を示すフローチャートである。

まず、ステップS141では、図柄組合せ表示フラグのデータを取得する。この処理では、前遊技でのリール31の停止後における有効ライン上の図柄組合せ、すなわち入賞役を判別する処理である。次のステップS142では、作動状態フラグを生成する。たとえばリプレイの入賞時には、リプレイに係る作動状態フラグを生成する。次のステップS143では、作動状態フラグの更新を行う。すなわち、それまでの作動状態フラグに代えて、ステップS142で生成した作動状態フラグに置き換える。

10

【0194】

次にステップS144に進み、ステップS143で更新した作動状態フラグを保存(記憶)する。この処理は、図11中、7E11番地の作動状態フラグをオン/オフする処理である。

次のステップS145では、ステップS141で取得した図柄フラグデータに基づいて、リプレイの図柄の組合せが表示されたか(リプレイが入賞したか)否かを判断する。リプレイの図柄の組合せが表示されたと判断したときはステップS146に進み、表示されていないと判断したときは本フローチャートによる処理を終了する。

ステップS146では、メインCPU54は、ベットメダルの読み込みを行う。この処理は、前遊技でベットされていたメダル枚数を読み込む処理である。具体的には、後述する図23の処理と同一である。次にステップS147に進み、自動ベット数データをセットする。この処理は、図11中、7E12番地のデータをセットする処理である。そして、本フローチャートによる処理を終了する。

20

【0195】

図15は、図13のステップS122における遊技状態出力(MS_STATUS_SET)を示すフローチャートである。

まず、ステップS148では、作動状態の出力要求をセットする。次のステップS149では、制御コマンドセット1を実行する。そして、本フローチャートによる処理を終了し、図13のステップS123に進む。

30

【0196】

図16は、図13のステップS125における貯留枚数読み込み処理(S_CREDIT_READ)を示すフローチャートである。先ず、ステップS151において、メインCPU54は、貯留枚数のデータを読み込む。ここでの処理としては、たとえば読み取った貯留枚数のデータを所定のレジスタに記憶する処理である。次にステップS152において、ステップS151で読み込んだ貯留枚数のチェックを行う。そして本フローチャートによる処理を終了する。

【0197】

図17は、図13のステップS127におけるブロッカオンの処理(MS_BLOCKER_ON)を示すフローチャートである。

40

先ず、ステップS161において、メインCPU54は、ブロッカ監視時間の経過をチェックする。ブロッカ監視時間は、予め所定値に設定されている。次にステップS162に進み、タイマによる所定時間が経過したか否かを判断する。経過したと判断したときはステップS163に進み、経過していないと判断したときはステップS161に戻る。

【0198】

ステップS163では、メインCPU54は、この時点からの割込みを禁止する。上述したように、メインループ(M_MAIN)の実行中には、2.235msごとに1回のタイマ割り込み処理が入るが、ステップS163における「割込み禁止」の処理の実行後は、当該割込み禁止が解除されるまで、割込みを許可しないように制御する。

【0199】

50

次のステップ S 1 6 4 では、メイン C P U 5 4 は、ブロッカ信号（7 E 0 3 番地の D 6 ビット）をオンにする処理を行う。さらに次のステップ S 1 6 5 では、投入監視カウンタの値をクリアする。ここで、投入監視カウンタとは、通路センサ 4 3 a がメダルを検知すると、「+1」とし、投入センサ 4 4 a 及び 4 4 b がメダルを検知すると、「0」とするカウンタである。すなわち、正常時には、「+1」と「0」とを繰り返すカウンタである。ブロッカ 4 5 をオン、すなわちブロッカ 4 5 によりメダル通路を形成したときは、通路センサ 4 3 a をメダルが検知するようになるからである。

【0200】

次のステップ S 1 6 6 では、ブロッカ信号オンのフラグをセットする（7 E 0 4 番地の D 2 ビットをオンにする）。そしてステップ S 1 6 7 に進み、ステップ S 1 6 3 で設定した割込み禁止の解除、すなわち割込みを許可する。そして、本フローチャートによる処理を終了する。

【0201】

上記において、ステップ S 1 6 4 ～ステップ S 1 6 6 の処理間に割込みを発生させないのは、以下の理由による。

本実施形態では、ステップ S 1 6 4 において、7 E 0 3 番地の D 6 ビット（ブロッカ信号）をオンにし、ステップ S 1 6 6 において、7 E 0 4 番地の D 2 ビット（ブロッカ状態信号）をオンにする処理を行う。この場合、ステップ S 1 6 4 とステップ S 1 6 6 との間に割込みが入ると、一方がオン、他方がオフの状態になってしまうが、そのような状態を避け、双方の値を一気に更新するために、割込み処理を禁止している。

なお、上述したように、ブロッカ信号のデータ番地と、ブロッカ信号状態のデータ番地とを分けているのは、出力に用いられる番地（7 E 0 3）と、出力に用いられない番地（7 E 0 4）とを分けているためである。

【0202】

続いて、図 1 3 中、ステップ S 1 3 1 における自動ベット時の出力要求セット、及びステップ S 1 3 2 における制御コマンドセット 1（S_CMD_SET）について、より詳しく説明する。

図 1 1 で示したように、7 E 1 6 ～7 E 3 5 番地は、制御コマンドバッファの記憶領域に設定されており、これらの記憶領域には、サブ制御基板 8 0 に送信するための制御コマンドデータが記憶される。

【0203】

まず、図 1 3 のステップ S 1 3 1 における自動ベット時の出力要求セットでは、D レジスタ及び E レジスタに、定義データを記憶する。ここで、D レジスタ及び E レジスタに記憶される定義データは、制御コマンドバッファに記憶する制御コマンドデータを生成する際に用いられるデータである。また、D レジスタ及び E レジスタは、メモリ 5 3 内の一部の記憶領域に設けられたものである。

【0204】

図 1 8 は、各処理ごとの定義データを示す図である。各処理ごとに、D レジスタ及び E レジスタに記憶する定義データが設けられている。この例では、自動ベット時の出力であるので、D レジスタには「9 0（H）」、E レジスタには「1 2（H）」の値を記憶する。したがって、

D レジスタ値 = 9 0（1 0 0 1 0 0 0 0）

E レジスタ値 = 1 2（0 0 0 1 0 0 1 0）

となる。

なお、図 1 8 では、各定義データについて、1 6 進数（H）及び 2 進数（B）の値を示している。実際には、2 進数の値が記憶される。

【0205】

自動ベット時の出力要求セット（ステップ S 1 3 1）において、D レジスタ及び E レジスタにそれぞれ上記のように定義データが記憶されると、次の制御コマンドセット 1 では、以下の処理が実行される。

図19は、制御コマンドセット1 (S_CMD_SET) の処理を示すフローチャートである。本実施形態において、制御コマンドセット1の処理時には、図19に示すステップS501以降の処理が実行される（他のフローチャートにおける制御コマンドセット1についても同様である）。

【0206】

まず、ステップS502では、割込み禁止処理を実行する。次のステップS503では、制御コマンドセット2 (SS_CMD_SET) (図20) を実行する。そしてステップS504に進み、ステップS502で禁止した割込み処理を解除（すなわち割込み許可）を実行する。以上の処理により、制御コマンドセット2の実行中は、割込み処理が禁止される。

そして、制御コマンドセット2では、以下に示すように、制御コマンドデータの書き込みを行うが、制御コマンドセット2の実行中は、上述したように割込み処理が禁止されている。したがって、制御コマンドデータの書き込み処理中に割込み処理が実行されることによる誤作動（正常の書き込み番地とは異なる番地に書き込んでしまうこと等）を防止することができる。

【0207】

ここで、割込みの禁止／解除を記憶するための割込みフラグが設けられている。割込みフラグは、ステップS502で割込み処理を禁止した後、次の割込み処理を実行するタイミング（割込み処理を禁止した時から2.235ms以内）が到来したときにオンにされる。そして、メインCPU54は、禁止した割込み処理を解除するときは、その後、割込みフラグをクリアする。

メインCPU54は、割込み処理を実行するタイミングが到来したときは、割込みフラグのオン／オフを判断し、割込み処理の禁止／許可を判断する。

【0208】

図20は、制御コマンドセット2 (SS_CMD_SET) を示すフローチャートである。

まず、ステップS511において、RWM上位アドレスセットでは、Hレジスタに、「7E(H)」(01111110(B))の値をセットする。よって、

Hレジスタ値 = 7E(01111110)

となる。

なお、Hレジスタとは、上述したD及びEレジスタと同様に、メモリ53内に設けられたものである。

【0209】

次のステップS512の「RWM指定？」では、Dレジスタに記憶された値の上位ビットが「0」か「1」かを判定する。ここで、本実施形態では、レジスタに記憶されるデータは、8ビット構成の1バイトデータ（2進数）であって、上位ビットとは、7ビット目（最上位ビット）を指す。そして、Dレジスタには上述のステップS131において「10010000」が記憶されているので、7ビット目、すなわち最上位ビットは、「1」である。

【0210】

ここで、Dレジスタ値の上位ビットが「0」になるのは、図18に示すように、手入れベット時や貯留ベット時である。これに対し、自動ベット時には上位ビットが「1」となる。上位ビットが「1」であるときは、「RWM指定」と判断され、ステップS513及び514の処理を実行する。これに対し、上位ビットが「0」であるときは、RWM指定でないと判断され、ステップS513及びS514の処理はスキップされる。

【0211】

ステップS513のRWM下位アドレスセットでは、Eレジスタの値をLレジスタに書き込む処理が行われる。ここで、Lレジスタとは、上述のD、E、Hレジスタと同様に、メモリ53内に設けられたものである。

上述のステップS131において、Eレジスタに記憶されている値は、「12(00010010)」であるので、

Lレジスタ値 = 12(00010010)

となる。

【0212】

次のステップS514の「第2制御コマンドセット」では、HLレジスタ値で指定される番地が指定するアドレスのデータを取得し、Eレジスタに書き込む。

ここで、

Hレジスタ値 = 7E (01111110)

Lレジスタ値 = 12 (00010010)

であるので、HLレジスタ値で指定される番地は、7E12番地となる。そして、図11に示すように、7E12番地のデータは、自動ベット数データであるので、このデータを読み込む。たとえば通常遊技では「3」となる（本例では、通常遊技であると仮定する）。

10

そして、この値を、Eレジスタに書き込む。よって、

Eレジスタ値 = 3 (00000011)

となる。

【0213】

すなわち、上述のステップS131では、Eレジスタ値として「12 (00010010)」が記憶されたが、この時点で、Eレジスタ値が変更となる。

なお、MB遊技中では、7E12番地のデータは、「2 (00000010)」であるので、その場合には、Eレジスタ値は「2」となる。

【0214】

20

次のステップS515において、制御コマンド書き込みポインタセットでは、以下の処理を行う。

1) Lレジスタに、制御コマンド書き込みポインタの番地である7E15の下位バイトデータを読み込む。ここで、下位バイトデータとは、7E15のうち、下位4ビットのデータを指す。したがって、下位バイトデータは「15」であるから、

Lレジスタ値 = 15

となる。

【0215】

2) HLレジスタ値で指定される番地のデータをAレジスタに読み込む。

Hレジスタ値 = 7E、Lレジスタ値 = 15であるので、HLレジスタ値で指定される番地は、7E15である。よって、7E15番地（制御コマンド書き込みポインタ）のデータを、Aレジスタに記憶する。本例では、7E15番地（制御コマンド書き込みポインタ）の値を「0 (H)」と仮定する。したがって、本処理により、Aレジスタ値は「0」となる。

30

【0216】

3) Aレジスタのデータと、「00001111 (10進数で「15」)」とをAND演算し、AND演算後の値をAレジスタ値とする。この演算により、「0」～「255」の範囲の値は、「0」～「15」の範囲の値に変換される。本例では、上述のようにAレジスタ値は「0」であるので、AND演算後の値は、「0」のままである。7E14及び7E15のデータは、いずれも範囲が「0」～「255」であるが、制御コマンドバッファの番地の範囲（7E16～7E35）、すなわち16個（一对の第1制御コマンド及び第2制御コマンドを記憶可能な最大数）に対応させるため、「0」～「15」の範囲の値に変換する。上記演算により、1バイトデータの範囲（0～255）をフルに使用することができるとともに、たとえば「16」(10進数)になったときは「0」に戻す演算処理が不要となり、容量削減を図ることができる。Aレジスタのデータが「255 (10進数)」、すなわち「11111111」となったときは、「1」インクリメントすれば、自動的に「0」(00000000)にすることができるからである。

40

【0217】

4) Aレジスタ値を2倍にする。この処理により、Aレジスタの値は、偶数になる。なお、本例では、Aレジスタ値は「0」であるので、2倍にする処理の実行後もAレジスタ

50

値は「0」である。

5) Lレジスタ値を「+1」にする。この時点では、Lレジスタ値は「15」であるから、「+1」により、

Lレジスタ値 = 16

となる。この処理は、制御コマンドバッファの先頭番地を指定するための処理である。

【0218】

次にステップS516に進み、「指定アドレスデータセット (S_ADDR_SET)」処理を実行する。

図21は、指定アドレスデータセット (S_ADDR_SET) 処理を示すフローチャートである。図20中、ステップS516に進むと、図21のステップS531に進む。

10

【0219】

まず、ステップS531では、「オフセット値によりワークテーブルのアドレス補正」を行う。この処理では、以下の2つを実行する。

1) Aレジスタの値と、Lレジスタの値とを加算し、Aレジスタに書き込む。

この時点において、Aレジスタの値は「0」であるので、Lレジスタの値は上述のように「16」であるから、「0 + 16 = 16」となる。よって、

Aレジスタ値 = 16

となる。

2) Aレジスタ値をLレジスタに読み込む。この処理により、

Lレジスタ値 = 16

20

となる。

【0220】

次のステップS532における「アドレスデータ取得」では、HLレジスタ値で指定される番地のデータを、Aレジスタに書き込む。上述のように、Hレジスタ値 = 7E、Lレジスタ値 = 16であるから、HLレジスタ値で指定される番地は、7E16である。よって、7E16番地の制御コマンドバッファの値を、Aレジスタに書き込む。

【0221】

次のステップS533の「指定アドレスデータゼロチェック」では、Aレジスタの値同士のAND演算を実行する。ここでAND演算を実行するのは、7E16番地（制御コマンドバッファ）のデータの有無を判断するためである。データ有の場合には、AND演算をすることで、「0」以外の値となる。一方、AND演算をした結果、値が「0」となったときは、データがないことを意味する。

30

【0222】

なお、本実施形態では、制御コマンドバッファに記憶される第1制御コマンドデータの7ビット目は、常に「1」となるように設定されている。したがって、制御コマンドバッファに第1制御コマンドデータが記憶されているときは、そのデータをAND演算して「0」になることはない。

また、本実施形態では、データの有無を判断するためのゼロフラグを備える。そして、上記AND演算の結果、データがないときは、ゼロフラグをオンにし、データがあるときは、ゼロフラグをオフにする。そして、図20のステップS517に戻る。

40

【0223】

図20のステップS517の「制御コマンド数 < 32?」では、上述のゼロフラグのオン/オフを判断する。ゼロフラグがオンのときは「Yes」と判断し、ステップS518に進み、ゼロフラグがオフのときは、「No」と判断し、本フローチャートによる処理を終了する。すなわち、ゼロフラグがオフであるときは、HLレジスタ値で指定される番地にデータがあることを意味するので、書き込み処理を実行しないためである。これにより、指定番地にデータがあるときは、上書きが防止される。

【0224】

ステップS518では、第1制御コマンドセット（第1書き込み）を実行する。具体的には、以下の処理を実行する。

50

1) Dレジスタ値を、HLレジスタ値で指定される番地の制御コマンドバッファに書き込む。上述のように、Dレジスタの値は、「90 (10010000)」である。また、Hレジスタ値=7E、Lレジスタ値=16であるので、HLレジスタ値で指定される番地は、7E16である。したがって、

7E16番地の制御コマンドデータ=10010000
となる。

【0225】

2) HLレジスタ値で指定される番地のデータの7ビット目を「1」にする。HLレジスタ値で指定される番地は、上記の通り7E16であるので、

7E16番地の制御コマンドバッファのデータ=10010000
となる。

10

【0226】

ここで、HLレジスタ値で指定される番地の7ビット目は、本例では、最初から「1」であるので、HLレジスタ値で指定される番地のデータの7ビット目を「1」にする処理を行っても、7E16番地の制御コマンドデータに変更はない。

一方、後述するように、HLレジスタ値で指定される番地の制御コマンドデータにおいて、7ビット目が「0」である場合があり、この場合には「0」から「1」に変更する。

【0227】

7ビット目を「1」にするのは、第1制御コマンドと第2制御コマンドとの区別のためである。本実施形態では、第1制御コマンドの7ビット目を「1」とし、第2制御コマンドの7ビット目を「0」とすることにより、サブ制御基板80側で第1制御コマンドであるか第2制御コマンドであるかを認識可能としている。

20

一方、上述したように、ステップS512において、RWM指定であるか否かは、7ビット目が「0」か「1」かを判断している。すなわち、本実施形態では、RWM指定であるか否かを判断するビット目と、第1制御コマンドであるか否かを判断するビット目とは、同一に設定されている。

【0228】

次にステップS519に進み、第2制御コマンドセット（第2書き込み）を実行する。具体的には、以下の処理を実行する。

1) Lレジスタの値を「+1」とする。この時点では、Lレジスタ値は「16」であるから、「+1」により、

30

Lレジスタ値=17
となる。

【0229】

2) Eレジスタの値をHLレジスタ値で指定される番地に書き込む。Eレジスタの値は、上述のように、「3 (00000011)」である。また、Hレジスタ値=7E、Lレジスタ値=17であるので、HLレジスタ値で指定される番地は、7E17である。よって、

7E17番地の制御コマンドバッファのデータ=3 (00000011)
となる。

40

【0230】

以上より、

7E16番地の制御コマンドバッファのデータ（第1制御コマンドデータ）=10010000 (90)

7E17番地の制御コマンドバッファのデータ（第2制御コマンドデータ）=00000011 (3)

となる。

なお、制御コマンドデータの書き込みにおいて、本実施形態のプログラムでは、第1制御コマンドデータから書き始めるという決まりを設けている。したがって、メインCPU84は、いずれが第1制御コマンドデータであるかを判断することができる。

50

【0231】

次のステップS520では、「制御コマンド書き込みポインタ値＝＋1」を実行する。具体的には、以下の処理を実行する。

1) Lレジスタ値を「15」にする。本例では、直前のLレジスタ値は「17」であるが、この値を「15」に変更する。よって、

Lレジスタ値＝15

となる。

【0232】

2) HLレジスタ値で指定される番地のデータを「＋1」にする。HLレジスタ値で指定される番地は、Hレジスタ値＝7E、Lレジスタ値＝15より、7E15である。そして、7E15番地のデータは、制御コマンド書き込みポインタである(図11)。よって、この処理により、制御コマンド書き込みポインタの値が「＋1」される。本例では、上記処理により、制御コマンド書き込みポインタの値は「0」から「1」となる。

【0233】

以上説明したように、図13中、ステップS131における自動ベット時の出力要求セット、及びステップS132における制御コマンドセット1(S_CMD_SET)を実行することにより、自動ベット時の制御コマンドデータ(サブ制御基板80に送信するデータ)が所定番地の制御コマンドバッファに書き込まれる。そして、これらの制御コマンドデータは、後述する割込み処理により、サブ制御基板80に送信される。

【0234】

なお、以上の処理に示すように、ステップS131における「自動ベット時の出力要求セット」では、Eレジスタには、定義データとして「12」が記憶されるが、自動ベット時は、ステップS512において、RWM指定であると判断される。このため、ステップS514において、HLレジスタ値が指定する番地(7E12)のデータを取得し、Eレジスタに書き込まれる。よって、Eレジスタ値は、最初の定義データから、7E12番地に記憶されているデータに変換される。これにより、第2制御コマンドデータは、7E12番地に記憶されているデータとなる。

【0235】

これに対し、たとえば後述するメダルの手入れによるベット時には、出力要求セット時に、図18に示すように、Dレジスタ及びEレジスタには、定義データとして、それぞれ「10」及び「71」が記憶される。したがって、手入れベット時は、Dレジスタに記憶された値の上位ビットは、「0」となる。よって、次のステップS512の「RWM指定？」では、Dレジスタに記憶された値の上位ビットは「0」と判断される。このため、ステップS514の処理はスキップされ、ステップS519において、Eレジスタの値がHLレジスタ値が指定する番地に書き込まれる。よって、第2制御コマンドデータは、最初にEレジスタに記憶した定義データ(「71」)となる。

【0236】

図22は、図13のステップS133におけるメダル1枚のベット処理(MS_MEDAL_INC)を示すフローチャートである。

まず、ステップS171では、メインCPU54は、ベットメダルの読み込みを行う(S_PLAYM_READ)(後述する図23)。

次のステップS172では、メダル枚数を「1」加算する処理(「＋1」)を行う。次にステップS173に進んで、メダル枚数表示の出力要求セットを行う。次のステップS174では、制御コマンドセット1を実行する。

なお、ステップS173でのメダル枚数表示の出力要求セット及び制御コマンドセット1は、メダル枚数の表示をサブCPU84が行うときに必要な制御コマンドデータをセットする処理である。

【0237】

次のステップS175では、メインCPU54は、獲得枚数表示クリアを行う。この処理では、前遊技でたとえば小役が入賞したときには、その入賞に基づく獲得枚数が表示さ

10

20

30

40

50

れるが、その表示をクリアする処理である。

次にステップ S 1 7 6 に進み、メイン C P U 5 4 は、ベット枚数表示 L E D 点灯データを生成する。これは、メダルが 1 枚ベットされるごとに点灯する L E D が設けられているので、メダル 1 枚ベットに対して 1 つの L E D を点灯させる処理を行うため、そのデータを生成する。

【 0 2 3 8 】

次のステップ S 1 7 7 では、メイン C P U 5 4 は、ベット枚数分の点灯データの生成を終了したか否かを判断する。ベット枚数分の点灯データの生成を終了していないと判断したときはステップ S 1 7 6 に戻り、点灯データの生成を継続する。一方、ベット枚数分の点灯データの生成を終了したと判断したときはステップ S 1 7 8 に進み、メイン C P U 5 4 は、生成した点灯データに対応するベット枚数表示 L E D を点灯する。

次にステップ S 1 7 9 に進み、メイン C P U 5 4 は、メダル限界枚数をセットする (MS_MMAL_SET) (後述する図 2 4) 。

【 0 2 3 9 】

ここで、メダル限界枚数とは、そのスロットマシン 1 0 でベット可能となっているメダル枚数の最大値 (又はその遊技でベットしなければいけないメダル枚数) を意味する。具体的には、本実施形態ではベット可能な最大メダル枚数が遊技状態で異なっているので、その遊技状態でベット可能な最大メダル枚数意味する。特に本実施形態では、通常遊技では、最大 3 枚のメダルがベット可能であり、M B ゲーム中では、最大で 2 枚のメダルをベット可能である。よって、M B ゲーム中を除くメダル限界枚数は 3 枚となり、M B ゲーム中のメダル限界枚数は 2 枚となる。

【 0 2 4 0 】

次にステップ S 1 8 0 に進み、メイン C P U 5 4 は、ベットメダルの読み込み処理 (S_PLAYM_READ) を行う (ステップ S 1 7 1 と同一) 。

そして、次のステップ S 1 8 1 において、メイン C P U 5 4 は、ベット枚数が限界枚数であるか否かを判断する。ベット枚数が限界枚数でないと判断したときは本フローチャートによる処理を終了し、限界枚数であると判断したときはステップ S 1 8 2 に進む。ステップ S 1 8 2 では、メダル限界フラグをセットする。すなわち、現時点でベット限界枚数になっていることを示すフラグをオンにし (図 8 中、7 E 0 4 番地の D 7 ビット)、本フローチャートによる処理を終了する。

【 0 2 4 1 】

図 2 3 は、ベットメダルの読み込み処理 (S_PLAYM_READ) (図 2 2 のステップ S 1 7 1 、ステップ S 1 8 0 等) を示すフローチャートである。先ず、ステップ S 1 9 1 において、メイン C P U 5 4 は、ベットメダル枚数のデータを読み込む。ここでの処理としては、たとえば読み取ったベットメダル枚数のデータを所定のレジスタに記憶する処理である。

次にステップ S 1 9 2 において、メイン C P U 5 4 は、ステップ S 1 9 1 で読み込んだベットメダル枚数のチェックを行う。ここでチェックを行うのは、たとえば読み取った値が正常の範囲内であるか否かをチェックするためである。そして、本フローチャートによる処理を終了する。

【 0 2 4 2 】

図 2 4 は、図 2 2 のステップ S 1 7 9 におけるメダル限界枚数のセット処理 (MS_MMAL_SET) を示すフローチャートである。ステップ S 2 0 1 では、自動ベット時の限界枚数をセットする。なお、自動ベットとは、リプレイ入賞時におけるメダルの自動ベットを指し、当該遊技でベットされた枚数が自動ベット時の限界枚数となる。次のステップ S 2 0 2 では、作動状態フラグをチェックする。この処理は、図 1 1 中、7 E 1 1 番地の作動状態フラグを確認する処理である。

【 0 2 4 3 】

そして、次のステップ S 2 0 3 では、メイン C P U 5 4 は、リプレイ作動時、すなわち 7 E 1 1 番地の D 0 ビットがオンであるか否かを判断する。オンであるときは、本フローチャートによる処理を終了し、オンでないときはステップ S 2 0 4 に進む。ステップ S 2

04では、メダルの限界枚数として2枚をセットする。

【0244】

次にステップS205に進み、メインCPU54は、当該遊技がメダル限界枚数2枚の遊技時であるか否かを判断する。上述したように、たとえば通常遊技であるときはメダル限界枚数が2枚でないと判断し、MBゲーム中であるときはメダル限界枚数が2枚であると判断する。ここでの処理は、上記と同様に、図11中、7E11番地の作動状態フラグを確認する処理である。そして、たとえばD1及びD2ビットがオンであるとき（すなわちMB遊技中）は、メダル限界枚数が2枚であると判断し、オフであるときはメダル限界枚数が3枚であると判断する。

【0245】

メダル限界枚数が2枚であるときは、既にステップS204で2枚をセットしているので本フローチャートによる処理を終了する。一方、メダル限界枚数が2枚でないと判断したときはステップS206に進み、メダル限界枚数3枚をセットする。そして、本フローチャートによる処理を終了する。

【0246】

図25は、図12のステップS106におけるメダル管理処理（MS_MEDAL_CHK）を示すフローチャートである。

図25において、ステップS211では、メインCPU54は、ブロック信号がオンであるか否かを検知する。ブロック信号がオンであるときはステップS212に進み、ブロック信号がオンでないときはステップS213に進む。ここで、「ブロック信号がオン」であるというのは、ブロック45によりメダル通路が形成されている場合に相当する。すなわち、図8中、7E03番地のD6ビットのデータを取得し、オン（「1」）であるか否かを検知する。

【0247】

ステップS212では、メインCPU54は、図8の7E02番地のデータを確認することにより、投入センサ信号に係るデータ（D1及びD2ビット）がオン（「1」）であるか否かを検知する。投入センサ信号に係るデータがオンであると判断したときは、メダルの手入れ投入を意味するので、ステップS222（MS_INSERT_CHK：図26）に進み、メダルの手入れ時のチェック処理を実行する。これに対し、オンでないとは判断したときは、ステップS213に進む。

【0248】

ステップS213では、1枚ベット、3枚ベット、精算操作の受付が可能であるか否かをチェックする要求をセットする。そして、次のステップS214において、これらの操作受付が可能であるか否かをチェックする。

次にステップS215に進み、メインCPU54は、操作受付が可能であるか否かを判断する。可能であると判断したときはステップS216に進み、可能でないと判断したときは本フローチャートによる処理を終了し、メインループに戻る。

【0249】

上記のステップS213～S215の処理により、ベット処理又は精算処理を行う前に、ベットスイッチ40及び精算スイッチ46の操作受付が可能な状態であるか否かを判断し、可能な状態である場合のみ、ベット処理又は精算処理に移行するようにしている。

【0250】

ステップS216では、ベットスイッチ信号及び精算スイッチ信号の確認要求をセットする。この処理は、図8の7E00番地の入力ポート51aレベルデータ（D2～D0ビット）を確認する処理である。

【0251】

次のステップS217では、ステップS216での確認に基づき、1ベットスイッチ40a、3ベットスイッチ40b、精算スイッチ46のうち、いずれかの立ち上がりデータが「1」であるか否かを判断する。いずれの立ち上がりデータも「1」でないと判断したとき、すなわち1ベットスイッチ40a、3ベットスイッチ40b、及び精算スイッチ4

10

20

30

40

50

6のいずれも操作の変化がないと判断したときは本フローチャートによる処理を終了し、メインループに戻る。

【0252】

ステップS217で、いずれかの立ち上がりデータが「1」とであると判断されたときはステップS218に進む。ステップS218では、スタートスイッチ41の受付け許可フラグをクリアする(図8中、7E04番地のD0ビット)。ここで、スタートスイッチ41の受付け許可フラグは、「0」又は「1」の値をとり、クリアとは、「0」にすることである。そして、スタートスイッチ41の受付け許可フラグが「0」とあるときは、スタートスイッチ41は有効にはならない(操作されても無効となる)。一方、スタートスイッチ41の受付け許可フラグが「1」のときは、スタートスイッチ41が有効となっているときである。

10

なお、その後にスタートスイッチ41の受付け許可フラグが有効になるのは、メインループのステップS108である。

【0253】

したがって、1ベットスイッチ40a、3ベットスイッチ40b、精算スイッチ46のうち、いずれかに変化があったとき(立ち上がりデータ「1」時)は、スタートスイッチ41を受付けを許可しないようにしてから、ベットスイッチ又は精算スイッチの操作に基づく処理(図25中、MS_MEDAL_RET、又はMS_BET_IN)を実行する。

【0254】

次にステップS219に進み、メインCPU54は、精算スイッチ信号に係るデータの確認要求をセットする。この処理は、図8の7E00番地の入力ポート51aレベルデータ(D0ビット)を確認する処理である。

20

【0255】

そして、次のステップS220で、メインCPU54は、ステップS219での確認に基づき、精算スイッチ信号の立ち上がりデータが「1」とあるか否かを判断する。精算スイッチ信号の立ち上がりデータが「1」とあるときはステップS221に進み、精算処理(MS_MEDAL_RET:図32)を実行する。一方、精算スイッチ信号の立ち上がりデータが「1」でないと判断したときは、ベットスイッチ信号の立ち上がりデータが「1」とあることを意味するので、ステップS223に進み、貯留ベット処理(MS_BET_IN:図30)を実行する。

30

【0256】

図26及び図27は、図25のステップS222におけるメダルの手入れ時のチェック処理(MS_INSERT_CHK)を示すフローチャートである。図27は、図26に続くフローチャートである。

図26において、ステップS231では、メインCPU54は、スタートスイッチ41の受付け許可フラグをクリアする(「0」にする)。

次にステップS232に進み、メインCPU54は、図8中、7E01番地のD1ビットのデータを確認し、投入センサ1(44a)信号に係るデータがオンであるか否かを判断する。

【0257】

40

なお、以下の説明では、図8中、7E01番地のD1ビットのデータを「投入センサ1信号に係るデータ」と称し、D2ビットのデータを「投入センサ2信号に係るデータ」と称する。また、データが「1」の場合はオンであり、「0」の場合はオフである。

【0258】

投入センサ1信号に係るデータがオンであるときはステップS233に進み、オンでないときはステップS238に進む。ステップS233では、投入センサ1の通過チェック時間をセットする。ここで、投入センサ1によりメダルが検知されると投入センサ1信号に係るデータがオンとなるが、その後、所定時間を経過しても投入センサ1信号に係るデータがオフにならないときは、メダル詰まり等が考えられるので、投入センサ1の通過チェック時間を計測する。

50

【0259】

次にステップS234に進み、メインCPU54は、ステップS232と同様にして、投入センサ1及び2信号に係るデータがオフであるか否かを判断する。投入センサ1及び2信号に係るデータがオフであると判断したときは本フローチャートによる処理を終了する。

【0260】

ここで、ステップS232で投入センサ1信号に係るデータがオンになった後、ステップS234で投入センサ1信号に係るデータがオフと判断されることは稀であるが、ノイズ等により投入センサ1信号に係るデータが一時的にオンとなり、ステップS232でそのオンが検出される場合がある。そして、ノイズ等により投入センサ1信号に係るデータが一時的にオンとなっても、ステップS234の処理により、ステップS235以下に進まないようにしている。

10

【0261】

ステップS234において、投入センサ1及び2信号に係るデータがオフでないと判断したときはステップS235に進む。ステップS235では、投入センサ1及び2信号に係るデータがオンであるか否かを判断する。投入センサ1及び2信号に係るデータがオンであると判断したときはステップS240に進み、オンでないと判断したときはステップS236に進む。

【0262】

ステップS236では、メインCPU54は、投入センサ1信号に係るデータがオンであるか否かを判断する。オンであると判断したときはステップS237に進み、オンでないと判断したときはステップS238に進む。

20

ステップS237では、メインCPU54は、投入センサ1の通過チェック時間が経過したか否かを判断する。すなわち、ステップS233でセットした時間が所定時間を経過したか否かを判断する。通過チェック時間が経過したと判断したときはステップS248に進み、通過チェック時間が経過していないと判断したときはステップS234に戻る。

【0263】

ステップS232において投入センサ1信号に係るデータがオンでないと判断されたとき、又はステップS236において投入センサ1信号に係るデータがオンでないと判断されてステップS238に進むと、メインCPU54は、メダルの不正通過エラーの表示要求をセットする。そしてステップS239に進み、本処理（MS_INSERT_CHK）からメダルエラー処理（MS_ERROR_DSP）に移行する。

30

【0264】

また、ステップS235において投入センサ1及び2信号に係るデータがオンであると判断され、ステップS240に進むと、メインCPU54は、投入センサ2の通過チェック時間をセットする。次にステップS241に進み、メインCPU54は、投入センサ2信号に係るデータがオンであるか否かを判断する。投入センサ2信号に係るデータがオンであるときはステップS245に進み、オンでないときはステップS242に進む。ステップS242では、メインCPU54は、投入センサ1及び2信号に係るデータがオンであるか否かを判断する。オンであるときはステップS243に進み、オンでないときはステップS238に進む。

40

【0265】

ステップS243では、メインCPU54は、投入センサ2の通過チェック時間が経過したか否かを判断する。すなわち、ステップS240でセットした時間が所定時間を経過したか否かを判断する。投入センサ2の通過チェック時間が経過したと判断したときはステップS248に進み、通過チェック時間が経過していないと判断したときはステップS244に進む。ステップS244では、メインCPU54は、投入センサ1の通過チェック時間が経過したか否かを判断する。投入センサ1の通過チェック時間が経過したと判断したときはステップS248に進み、通過チェック時間が経過していないと判断したときはステップS241に戻る。

50

【0266】

ステップS241において投入センサ2信号に係るデータがオンであると判断され、ステップS245に進むと、メインCPU54は、投入センサ1及び2信号に係るデータがオフであるか否かを判断する。投入センサ1及び2信号に係るデータがオフであると判断したときは図27のステップS249に進み、オフでないと判断したときはステップS246に進む。ステップS246では、メインCPU54は、投入センサ2信号に係るデータがオンであるか否かを判断する。

【0267】

投入センサ2信号に係るデータがオンでないと判断したときはステップS238に進み、オンであると判断したときはステップS247に進む。ステップS247では、メインCPU54は、投入センサ2の通過チェック時間が経過したか否かを判断する。投入センサ2の通過チェック時間が経過したと判断したときはステップS248に進み、通過チェック時間が経過していないと判断したときはステップS245に戻る。

10

【0268】

ステップS248に進むと、メダル滞留エラーの表示要求をセットする。メダル滞留エラーとは、投入センサ1や2信号に係るデータが、通過チェック時間の経過後もオンの状態となり、メダル通路にメダルが滞留していることを意味するので、そのエラーを表示するためである。そして、上記と同様に、本処理を終了してエラー処理に移行する。

【0269】

ステップS245において投入センサ1及び2信号に係るデータがオフであると判断し、図27のステップS249に進むと、メインCPU54は、メダル異常投入エラーの表示要求をセットする。ここで、メダル異常投入エラーとは、通路センサ43aと投入センサ1、2の通過異常があったことを示すエラーである。

20

【0270】

次にステップS250に進み、メインCPU54は、投入監視カウンタの値を「1」減算する。ここで、投入監視カウンタとは、上述したように、通路センサ43aがメダルを検知すると、「+1」とし、投入センサ1及び2がメダルを検知すると、「0」とするカウンタである。すなわち、正常時には、「+1」と「0」とを繰り返す。

【0271】

次にステップS251に進み、メインCPU54は、投入監視カウンタの値が「0」～「3」の範囲であるか否かを判断する。当該範囲内であると判断したときはステップS252に進み、範囲内でないと判断したときは、図26のステップS239に進み、上記と同様にエラー処理に移行する。

30

【0272】

たとえば、当該カウンタの値が「-1」となるのは、通路センサ43aを通過せずに投入センサ1、2を通過した場合が挙げられる。

また、当該カウンタの値が「+4」となるのは、通路センサ43aを通過したメダルが投入センサ1、2を通過する前に滞留している場合が挙げられる。

【0273】

ステップS252では、メダル限界枚数をセットする(MS_MMAX_SET：図24)。次にステップS253に進み、ベットメダル(現時点でベットされているメダル枚数)の読み込みを実行する(S_PLAYM_READ：図23)。

40

次のステップS254では、貯留枚数の読み込みを実行する(S_CREDIT_READ：図16)。

【0274】

次にステップS255に進み、メインCPU54は、現時点でベットされているメダル枚数及び貯留枚数の合計数を算出する。次のステップS256では、メダルの投入を有効とした後に、その後のメダル投入は不可能であるか否かを判断する。本実施形態では、

「現在のベット枚数」+「現在の貯留枚数」-49=「メダル限界枚数」

という式が成立するか否かを演算する。ここで、「現在のベット枚数」とは、現在通過

50

したメダルを加算する前のメダル枚数を指す。たとえば、現在のベット数が「2」であり、現在の貯留数が「50」であるとき、メダル限界枚数は「3」となり、この式が成立するので、現在通過したメダルの投入後は、その後のメダル投入は不可能であると判断する。

【0275】

そして、投入不可能と判断したときはステップS257に進んでブロック45をオフにする処理を実行する（MS_BLOCKER_OFF：図28）。そしてステップS258に進む。一方、投入可能と判断したときは、ステップS257をスキップしてステップS258に進む。

【0276】

ここで、後述したように、ブロック信号は、オン時には正の信号を出力し、正論理により、オン時は「1」が記憶される（図8中、7E03番地のD6ビット）。

そして、ブロック信号がオンであるときは、投入センサ1及び2を通過するメダル通路を形成させ、オフであるときは、メダル投入口43から投入されたメダルを払出し口から返却するメダル通路を形成する。

【0277】

次のステップS258では、メインCPU54は、メダル手入れ時の出力要求をセットし、次のステップS259では、制御コマンドセット1を実行する（このステップS258～ステップS259の制御については後述する）。

次のステップS260では、メインCPU54は、現時点でメダルのベット枚数が限界枚数（上限枚数）であるか否かを判断する。ベット枚数が限界値であると判断したときはステップS262に進み、貯留（クレジット）枚数の加算処理（MS_CREDIT_ADD：詳細は省略）を実行する。これに対し、ベット枚数が限界枚数でないと判断したときはステップS261に進み、メダル1枚のベット加算処理を行う（MS_MEDAL_INC：図22）。

【0278】

図28は、図27のステップS257におけるブロックオフの処理（MS_BLOCKER_OFF）を示すフローチャートである。上述したように、メダルの手入れが有効になった後、メダルの手入れが不可能（受付け不可能）となったときは、ブロック45を駆動して、メダル投入口43から手入れされたメダルを払出し口から返却するメダル通路を形成するように制御する。

【0279】

先ず、ステップS271において、メインCPU54は、ブロック信号の確認処理を行う。この処理は、図8中、7E03番地のD6ビットのデータ（「0」又は「1」）を確認する処理である。ここで、D6ビットのデータが「0」であるときは、既にブロック45がオフにされていることを意味する。

【0280】

次にステップS272に進み、メインCPU54は、ブロック信号がオフであるか否かを判断する。オフでないと判断したときはステップS273に進み、オフであると判断したときは本フローチャートによる処理を終了する。すなわち、既にブロック45がオフであるときは、ブロック45をオフにする本処理を進めることなく本フローチャートによる処理を終了する。

【0281】

ステップS272においてブロック信号がオフでないと判断し、ステップS273に進むと、メインCPU54は、この時点からの割込みを禁止する。上述したように、メインループ（M_MAIN）の実行中には、2.235msごとに1回のタイマ割り込み処理が入るが、ステップS273のような「割込み禁止」の記述があるときは、当該割込み禁止が解除されるまで、割込みを許可しないように制御する。

【0282】

次のステップS274では、メインCPU54は、ブロック信号をオフにする処理を行う。さらに次のステップS275では、ブロック信号状態をオフにする処理を行う。

そしてステップ S 2 7 6 に進み、メイン C P U 5 4 は、投入センサ 2 の異常入力の検出時間をセットする。

【0283】

ここで、本実施形態では、ブロック 4 5 をオン（メダル通過）からオフ（メダル返却）にした後、所定時間を経過する前には投入センサ 2 のオン信号を検出しても投入センサ 2 の異常入力とは判断しないが、所定時間を経過した後に投入センサ 2 のオン信号を検出したときは投入センサ 2 の異常入力と判断する。このため、ステップ S 2 7 6 のタイミングで、投入センサ異常入力の検出時間（以下の 500.64ms）をセットする。

【0284】

図 29 は、ブロック 4 5 のオン／オフと、投入センサ 2 信号に係るデータのオン／オフと、エラー監視とのタイミングを示すタイムチャートである。図 29 に示すように、ブロック 4 5 がオンからオフになったタイミングで、時間の計測を開始し、所定時間、特にこの例では、500.64ms の経過前までに投入センサ 2 信号に係るデータのオンを検出しても、その間は、エラー監視をしていないので、エラーと判断されることはない。

【0285】

図 29 の例では、ブロック 4 5 をオフにした後、エラー監視を有効にする前に、投入センサ 2 信号に係るデータがオンとなっている（メダルが投入センサ 2 により検知されている）。この場合、当該メダルはエラーと判断されず（遊技の進行を邪魔せず）、ベットされる。

なお、500.64ms の計測は、割込み回数のカウント値「224」をセットすることにより行う（1 割込みの時間「2.235ms」×回数「224」=500.64ms）。

【0286】

次に、所定時間（500.64ms）の経過後、エラー監視を開始する。エラー監視を開始した後、投入センサ 2 のオン信号を検出したときは、その時点からエラー監視を無効とする。また、エラーを検知してから一定時間を経過したときはエラー表示を開始する。エラー表示後は、エラーが解除されるまで、エラー表示が継続するとともに遊技停止状態となる。

エラー要因が除去され、投入センサ 2 の信号がオフとなったときは、その後、エラーを解除し、エラー監視を再度有効にする。

【0287】

図 28 において、ステップ S 2 7 6 で投入センサ異常入力の検出時間をセットすると、ステップ S 2 7 7 に進み、ステップ S 2 7 3 で設定した割込み禁止の解除、すなわち割込みを許可する。そして、本フローチャートによる処理を終了する。

【0288】

上記処理において、ステップ S 2 7 4 ～ステップ S 2 7 6 の処理間に割込みを発生させないのは、ブロックオンの処理時と同様である。

すなわち、ブロック信号（7E03 番地の D6 ビット）をオンにする処理、及びブロック状態信号（7E04 番地の D2 ビット）をオンにする処理との間に割込みが入ると、一方がオン、他方がオフの状態になってしまうが、そのような状態を避け、双方の値を一気に更新するために、割込み処理を禁止している。

【0289】

図 27 において、ステップ S 2 5 8 におけるメダル手入れ時の出力要求セット、及びステップ S 2 5 9 の制御コマンドセット 1 では、上述したステップ S 1 3 1 及びステップ S 1 3 2 と同様の処理を実行する。

上述のステップ S 1 3 1 及びステップ S 1 3 2 の処理は、リプレイの入賞に基づいてメダルを自動ベットしたときに、その制御コマンドを送信する処理である。

これに対し、ステップ S 2 5 8 及びステップ S 2 5 9 の処理は、遊技者によりメダルが手入れされたときに、その制御コマンドを送信する処理である。

【0290】

ステップ S 2 5 8 において、メダル手入れ時の出力要求セットは、ステップ S 1 3 1 の処理と同様に、D 及び E レジスタに定義データを記憶する。図 1 8 に示すように、メダル手入れ時に D 及び E レジスタに記憶される定義データは、

D レジスタ値 = 1 0 (0 0 0 1 0 0 0 0)

E レジスタ値 = 7 1 (0 1 1 1 0 0 0 1)

となる。

【 0 2 9 1 】

次に、ステップ 2 5 9 に進むと、ステップ S 1 3 2 と同様に、図 1 9 の制御コマンドセット 1 (S_CMD_SET) の処理に移行する。

制御コマンドセット 1 の処理において、ステップ S 5 0 2 では、割込み禁止処理を実行する。次のステップ S 5 0 3 では、制御コマンドセット 2 (SS_CMD_SET) (図 2 0) を実行する。そしてステップ S 5 0 4 に進み、ステップ S 5 0 2 で禁止した割込み処理を解除 (すなわち割込み許可) を実行する。

【 0 2 9 2 】

ステップ S 5 0 3 (制御コマンドセット 2) では、図 2 0 のステップ S 5 1 1 に移行する。ステップ S 5 1 1 において、RWM 上位アドレスセットでは、H レジスタに、「 7 E (H) 」 (0 1 1 1 1 1 1 0 (B)) をセットする。よって、

H レジスタ値 = 7 E (0 1 1 1 1 1 1 0)

となる。

【 0 2 9 3 】

次のステップ S 5 1 2 の「 RWM 指定 ? 」では、D レジスタに記憶された値の上位ビットが「 0 」か「 1 」かを判定する。ここで、上述のように、リプレイの入賞に基づくメダルの自動ベット時は、上位ビットが「 1 」であったが、メダル手入れ時には、D レジスタの値は「 0 0 0 1 0 0 0 0 (1 0) 」であるので、上位ビット (7 ビット目) は「 0 」となる。これにより、ステップ S 5 1 2 では、RWM 指定でないと判断され、ステップ S 5 1 3 及びステップ S 5 1 4 の処理が行われることなくステップ S 5 1 5 に進む。

【 0 2 9 4 】

次のステップ S 5 1 5 の制御コマンド書き込みポインタセットでは、以下の処理を行う。

1) L レジスタに、制御コマンド書き込みポインタの番地である 7 E 1 5 の下位バイトデータを読み込む。ここで、下位バイトデータとは、8 ビット中、下位 4 ビット (0 ~ 3 ビット目) のデータを指す。したがって、下位バイトデータは「 1 5 」であるので、

L レジスタ値 = 1 5

となる。

2) H L レジスタで指定される番地のデータを A レジスタを読み込む。

H L レジスタ値は、H レジスタ値 = 7 E、L レジスタ値 = 1 5 より、7 E 1 5 であるから、7 E 1 5 番地 (制御コマンド書き込みポインタ) のデータを、A レジスタに記憶する。本例では、7 E 1 5 番地 (制御コマンド書き込みポインタ) の値を「 1 (H) 」と仮定する。したがって、A レジスタ値は「 1 」となる。

【 0 2 9 5 】

3) A レジスタのデータと「 0 0 0 0 1 1 1 1 (1 0 進数で「 1 5 」) 」とを AND 演算し、AND 演算後の値を A レジスタ値とする。この演算により、「 0 」 ~ 「 2 5 5 」の範囲の値は、「 0 」 ~ 「 1 5 」の範囲の値に変換される。本例では、AND 演算後の値は、「 1 (H) 」 (0 0 0 0 0 0 0 1 (B)) のままである。

4) A レジスタ値を 2 倍にする。この処理により、A レジスタの値は、偶数になる。なお、本例では、この時点で A レジスタ値は「 1 」であるので、2 倍にすると、A レジスタ値は「 2 」となる。

5) L レジスタ値を「 + 1 」にする。この時点では、L レジスタ値は「 1 5 」であるから、「 + 1 」の処理により、

L レジスタ値 = 1 6

となる。

【0296】

次にステップS516に進み、指定アドレスデータセット (S_ADDR_SET) (図21) を実行する。

図21において、ステップS531では、「オフセット値によりワークテーブルのアドレス補正」を行う。この処理では、以下の2つを実行する。

1) Aレジスタの値と、Lレジスタの値とを加算し、Aレジスタに書き込む。

この時点において、Aレジスタの値は「2」であるので、Lレジスタの値は上述のように「16」であるから、「 $2 + 16 = 18$ 」となる。よって、

Aレジスタ値 = 18

となる。

2) Aレジスタの値をLレジスタに読み込む。この処理により、

Lレジスタ値 = 18

となる。

【0297】

次のステップS532における「アドレスデータ取得」では、HLレジスタ値で指定される番地のデータをAレジスタに書き込む。上述のように、Hレジスタ値 = 7E、Lレジスタ値 = 18であるから、HLレジスタ値 = 7E18である。よって、7E18番地の制御コマンドバッファの値を、Aレジスタに書き込む。

【0298】

次のステップS533の「指定アドレスデータゼロチェック」では、Aレジスタの値同士のAND演算を実行する。このAND演算の結果、データがないときは、上述したゼロフラグをオンにし、データがあるときは、ゼロフラグをオフにする。そして、図20のステップS517に戻る。

【0299】

図20のステップS517の「制御コマンド数 < 32?」では、ゼロフラグのオン/オフを判断する。ゼロフラグがオンのときは「Yes」と判断し、ステップS518に進み、ゼロフラグがオフのときは、「No」と判断し、本フローチャートによる処理を終了する。すなわち、ゼロフラグがオフのときは、HLレジスタ値で指定される番地（制御コマンドバッファ）にデータがあることを意味するので、書き込み処理を実行しないためである。

【0300】

ステップS518では、第1制御コマンドセット（第1書き込み）を実行する。具体的には、以下の処理を実行する。

1) Dレジスタ値を、HLレジスタ値で指定される番地の制御コマンドバッファに書き込む。上述のように、Dレジスタ値は、メダル手入れ時は「10 (00010000)」であり、HLレジスタ値で指定される番地は、Hレジスタ値 = 7E、Lレジスタ値 = 18より、7E18であるので、

7E18番地の制御コマンドバッファのデータ = 00010000

となる。

【0301】

2) HLレジスタ値で指定される番地のデータの7ビット目を「1」にする。この処理により、

7E18番地の制御コマンドバッファのデータ = 10010000

となる。

7ビット目を「1」にするのは、上述したように、第1制御コマンドデータであることを示すためである。

【0302】

次にステップS519に進み、第2制御コマンドセット（第2書き込み）を実行する。具体的には、以下の処理を実行する。

10

20

30

40

50

1) Lレジスタの値を「+1」とする。この時点では、Lレジスタ値は「18」であるから、「+1」処理により、

Lレジスタ値 = 19

となる。

【0303】

2) Eレジスタ値をHLレジスタ値で指定される番地書き込む。Eレジスタ値は、上述のように、「71 (01110001)」である。また、HLレジスタ値で指定される番地は、Hレジスタ値 = 7E、Lレジスタ値 = 19より、7E19である。よって、

7E19番地の制御コマンドバッファのデータ = 01110001

となる。

以上より、

第1制御コマンドデータ (7E18番地) = 10010000 (90)

第2制御コマンドデータ (7E19番地) = 01110001 (71)

となる。

【0304】

次のステップS520では、「制御コマンド書き込みポインタ値 = +1」を実行する。具体的には、以下の処理を実行する。

1) Lレジスタ値を「15」に設定する。本例では、直前のLレジスタ値は「19」であるが、この値を「15」とする。よって、

Lレジスタ値 = 15

となる。

2) HLレジスタ値で指定される番地のデータを「+1」にする。HLレジスタ値で指定される番地は、Hレジスタ値 = 7E、Lレジスタ値 = 15より、7E15番地であり、7E15番地は、制御コマンド書き込みポインタである。よって、この処理により、制御コマンド書き込みポインタの値が「+1」される。本例では、上記処理により、制御コマンド書き込みポインタの値は「1」から「2」となる。

【0305】

以上のように、メダルの手入れ時と、リプレイ作動に基づくメダルの自動ベット時とで、第1制御コマンドデータは、同一 (90 (H)) であるが、第2制御コマンドデータが異なる。これにより、第1制御コマンドデータ及び第2制御コマンドデータを受信したサブ制御基板80側では、第1制御コマンドデータが「90 (H)」であることによりベットであること、及び第2制御コマンドデータにより、手入れか自動ベットかを判断可能となる。

【0306】

また、上述より明らかであるが、制御コマンド書き込みポインタ (7E15番地) の値が「1」の場合でも、制御コマンドバッファ (第1制御コマンド1) の番地 (7E18) にコマンドデータを記憶することができる。

換言すれば、制御コマンド書き込みポインタの値を「1」インクリメントするだけで、制御コマンドバッファの偶数の番地を指定して、第1制御コマンドデータを記憶することができる。

【0307】

図30は、図25のステップS223における貯留ベット処理 (MS_BET_IN) を示すフローチャートである。

先ず、ステップS281において、メインCPU54は、メダル限界フラグがオン (「1」) であるか否かを判断する。ここで、メダル限界フラグは、図8中、7E04番地のD7ビットのデータを判断することにより行う。メダルが限界枚数 (たとえば通常遊技では3枚) までベットされているときはメダル限界フラグがオンとされ、図8中、7E04番地のD7ビットのデータとして「1」が記憶される。そして、メインCPU54は、このD7ビットのデータを読み込むことにより、メダル限界フラグがオン (「1」) であるか否かを判断する。

10

20

30

40

50

【0308】

ステップS281において、メダル限界フラグがオンであるときは、既に最大ベット状態であることを意味するので、本フローチャートによるベット処理を終了する。すなわち、図25のステップS220においてベットスイッチの操作を検知し、「MS_BET_IN」の処理に移行したとしても、最大ベットが既になされているときは、ベット処理を行わない。

【0309】

メダル限界フラグがオンでないと判断したときはステップS282に進み、ベット要求枚数として「1枚」をセットする。次にステップS283に進み、図8中、7E01番地の入力ポート立ち上がりデータのうち、1ベットスイッチ信号に係るデータ（D1ビットのデータ）がオン（「1」）であるか否かを判断する。1ベットスイッチ信号に係るデータがオンであるときはステップS288に進み、1ベットスイッチ信号に係るデータがオンでないときはステップS284に進む。

10

【0310】

ステップS284では、メダル限界枚数（たとえば、通常遊技では3枚）をセットする。この処理は、図24で示した「MS_MMAX_SET」と同じである。

次のステップS285では、ベットメダルの読み込み処理を行う。この処理は、図23で示した「S_PLAYM_READ」と同じである。次に、ステップS286に進み、投入要求枚数を演算する。

【0311】

20

この処理は、メダル限界枚数セット（ステップS284）の値から、ベットメダルの読み込み（ステップS283）で読み込んだ値を減算する処理である。たとえばメダル限界枚数セットで3枚がセットされ、ベットメダルの読み込みで1枚が既にベットされていると判断されたときは、「3-1」を実行する処理に相当する。

次に、ステップS287に進み、ベット要求枚数修正を行う。この処理は、ステップS286で算出した値をベット要求枚数としてセットする処理である。

【0312】

次に、ステップS288に進み、現時点における貯留枚数の読み込みを行う。この処理は、図16で示した「S_CREDIT_READ」と同じである。

次のステップS289では、メインCPU54は、メダルの貯留の有無を判断する。メダルの貯留なしと判断したときは本フローチャートによる処理を終了し、メダルの貯留ありと判断したときはステップS290に進む。

30

【0313】

ステップS290では、3ベットスイッチ信号の立ち上がりデータをクリアする。具体的には、図8中、7E01番地の入力ポート51aの立ち上がりデータのうち、D2ビットのデータを「0」にする。

次のステップS291では、1ベットスイッチ信号の立ち上がりデータをクリアする。具体的には、上記と同様に、入力ポート51aの立ち上がりデータのうち、D1ビットのデータを「0」にする。そして、ステップS292に進む。

【0314】

40

以上の処理のように、ステップS289で貯留メダルありの場合には、3ベットスイッチ信号及び1ベットスイッチ信号の立ち上がりデータをクリアした後、ベット処理を実行する。

上述したように、図8で示した（入力ポートの）立ち上がりデータは、割込み処理によって生成され、記憶される。

一方、ベット処理を実行する場合には、割込み処理によって生成・記憶した立ち上がりデータを、メインループ内でクリアする処理を実行する。

【0315】

ここで、立ち上がりデータをクリアしなかった場合には、再度割込み処理が実行されるまで、立ち上がりデータが維持される。この場合、メインループのベット処理では、立ち

50

上がりデータが「1」であることに基づいて、再度、1ベット処理が実行されてしまう。

このような不都合を回避するために、メダルの貯留がある場合には、ベット処理の開始前に、立ち上がりデータをクリアしている。

【0316】

ステップS292では、メダル貯留枚数（ステップS288で読み込んだ枚数）が、ベット要求枚数（ステップS286でセットした枚数）以上であるか否かを判断する。メダル貯留枚数がベット要求枚数以上であると判断したときはステップS294に進み、メダル貯留枚数がベット要求枚数以上でないと判断したときはステップS293に進む。

ステップS293では、メダル貯留枚数をセットする。この処理は、貯留枚数をベットする処理である。すなわち、ステップS292で「No」のときは、貯留枚数がベット枚数に満たない場合であるので、この場合には貯留枚数をベット枚数とする処理を行う。そしてステップS294に進む。

【0317】

ステップS294では、貯留ベット時の出力要求をセットする。次にステップS295に進み、制御コマンドセット1を実行する（この点の具体的説明については後述する）。

次のステップS296では、メダル1枚のベット処理を行う（MS_MEDAL_INC：図22）。次にステップS297に進み、貯留枚数から1枚を減算する処理を行う（MS_CREDIT_DEC：後述する図31）。次のステップS298では、メインCPU54は、要求枚数のベットが終了したか否かを判断する。終了したと判断したときは本フローチャートによる処理を終了し、終了していないと判断したときはステップS296に戻る。これにより、2枚以上をベットするときは、1枚ベットの処理（ステップS296及びS297）を繰り返す。

【0318】

上記のステップS294及びステップS295では、上述したステップS258及びステップS259と同様の処理を行う。ここで、図18に示すように、貯留ベット時の定義データは、

Dレジスタ値＝10（00010000）

Eレジスタ値＝1（00000001）

となる。

したがって、メダルの手入れベット時に対し、Dレジスタ値が同一で、Eレジスタ値が異なる。

【0319】

これにより、制御コマンドセット1が実行されると、

第1制御コマンドデータ＝10010000（90）（たとえば7E16番地の制御コマンドバッファデータ）

第2制御コマンドデータ＝00000001（1）（たとえば7E17番地の制御コマンドバッファデータ）

が送信すべき制御コマンドデータとしてセットされる。

よって、第1制御コマンドデータは、手入れベット時と同一であるが、第2制御コマンドデータが異なる。

【0320】

また、貯留ベット時は、Dレジスタ値の定義データ中、7ビット目が「0」であるので、図20のステップS512中、RWM指定でないと判断される。これにより、第2制御コマンドデータは、Eレジスタ値となる。

【0321】

図31は、図30のステップS297における貯留枚数から1枚を減算する処理（MS_CREDIT_DEC）を示すフローチャートである。

ステップS301では、貯留枚数の読み込み処理（S_CREDIT_READ）を行う。この処理は、図16の処理と同一である。

次のステップS302では、貯留枚数から「1」を減算する処理（「-1」）を行う。

そして本フローチャートによる処理を終了する。

【0322】

図32は、図25のステップS221における精算処理（MS_MEDAL_RET）を示すフローチャートである。

先ず、ステップS311では、メインCPU54は、貯留枚数の読み込み処理（S_CREDIT_READ）を行う。この処理は、図16の処理と同一である。次のステップS312では、メインCPU54は、ベットメダルを読み込み処理（S_PLAYM_READ）を行う。この処理は、図23の処理と同一である。

【0323】

次に、ステップS313に進み、メインCPU54は、作動状態フラグ（図11中、7E11番地のデータ）をチェックする。続いて、ステップS314では、メインCPU54は、当該遊技がリプレイ作動時であるか否か（リプレイの作動状態フラグがオンであるか否か）を判断する。

ステップS314においてリプレイ作動時でないときはステップS316に進み、リプレイ作動時であるときはステップS315に進む。

【0324】

ステップS315では、精算可能なベット数をクリアする。この処理は、ステップS312で読み込んだベット枚数を、リプレイ作動時には「0」にする処理である。ここで、ステップS312では、読み取ったベット枚数をAレジスタに記憶する。たとえば、リプレイの入賞に基づく自動ベット時には、「3（H）」の値が記憶される。そして、このステップS315では、Aレジスタ値をXOR演算し、「0」にする（クリアする）。

なお、本フローチャートに示すように、リプレイの作動による自動ベットを有するときであっても、その自動ベットを維持した状態で、貯留メダルの精算が可能である。

【0325】

次のステップS316では、メインCPU54は、精算可能なメダルの有無を判断する。この処理は、ステップS311で読み込んだ貯留（クレジット）枚数と、ステップS312で読み込んだベット数とを「OR」演算し、精算可能なメダルの有無を判断する。なお、ステップS311で読み込んだ貯留枚数及びステップS312で読み込んだベット枚数（又はステップS315による処理後のベット枚数）は、いずれも、上述したようにレジスタに記憶されている。ここで、リプレイ作動時の場合には、ステップS315において精算可能なベットメダル枚数を「0」（クリア）にしているため、貯留枚数がある場合にのみ「Yes」となる。

ステップS316において精算可能なメダルありと判断したときはステップS317以降の処理に進むが、精算可能なメダルなしと判断したときは、本フローチャートによる処理を終了する。

【0326】

ステップS317では、ブロッカ45をオフにする処理（MS_BROCKER_OFF）を行う。この処理は、上述した図28の処理と同じである。

次に、ステップS318に進み、メインCPU54は、精算開始時の出力要求をセットし、次のステップS319では制御コマンドセット1を実行する（この点の具体的説明は後述する）。

次のステップS320及びステップS321の処理は、上述したステップS313及びステップS314の処理と同一である。

【0327】

そして、ステップS321において、リプレイ作動時であると判断したときはステップS324に進み、リプレイ作動時でないとは判断したときはステップS322に進む。

ステップS322では、ベットメダル（現在ベットされているメダル枚数）の読み込み処理（S_PLAYM_READ）（ステップS312と同じ）を行う。

【0328】

次にステップS323に進み、メインCPU54は、ベットメダルの有無を判断する。

ベットメダルありと判断されたときはステップ S 3 2 7 以降の処理に進み、ベットメダルなしと判断されたときはステップ S 3 2 4 以降の処理に進む。

ここで、ベットメダルありの場合には、ステップ S 3 2 7 以降の処理を行うことによって、ベットメダルの精算処理を行う。これに対し、ベットメダルなしの場合には、ステップ S 3 2 4 以降の処理を行うことによって、貯留メダル、すなわちクレジットされているメダルの精算処理を行う。

したがって、ベットメダル及び貯留メダルの双方を有する場合には、ベットメダルの精算を先に行う。

【0329】

ステップ S 3 2 3 からステップ S 3 2 4 に進むと、メイン CPU 5 4 は、打ち止め表示 LED を点灯する処理を行う。本実施形態では、貯留メダルの精算処理を行っている間は、打ち止め表示 LED を点灯させる。次にステップ S 3 2 5 に進み、貯留メダルの精算処理 (MS_CREDIT_RET) を行う (後述する図 3 3)。この精算処理後は、ステップ S 3 2 6 に進み、ステップ S 3 2 4 で点灯させた打ち止め表示 LED の消灯処理を行う。そして、ステップ S 3 3 6 に進む。

なお、打ち止め表示 LED の点灯／消灯を示すデータは、図 8 中、7 E 0 4 番地のデータ中、D 4 ビットに記憶される。

【0330】

一方、ステップ S 3 2 3 においてベットメダルありと判断され、ステップ S 3 2 7 に進むと、メイン CPU 5 4 は、メダル 1 枚の払出し処理 (MS_1MEDAL_PAY) を行う (後述する図 3 4 ~ 図 3 5)。

【0331】

次にステップ S 3 2 8 に進み、メイン CPU 5 4 は、ベット枚数を表示する LED を 1 減算するように制御する。次のステップ S 3 2 9 では、ベットメダルの読み込みを行う (S_PLAYM_READ)。次にステップ S 3 3 0 に進み、メダル枚数の減算処理を行う。次のステップ S 3 3 1 では、メダル枚数表示の出力要求をセットする。この処理は、1 枚のメダルを減算した後のメダル枚数の表示の要求である。次のステップ S 3 3 2 では、メイン CPU 5 4 は、制御コマンドセット 1 を実行し、次のステップ S 3 3 3 でベットメダルの読み込みを行う (S_PLAYM_READ)。このステップ S 3 3 3 では、1 枚減算後のメダル枚数が読み込まれる。

【0332】

そしてステップ S 3 3 4 に進み、メイン CPU 5 4 は、ステップ S 3 2 3 と同様に、ベットメダルの有無を判断する。ベットメダルありと判断されたときはステップ S 3 2 7 に戻り、上記と同様に、メダル 1 枚の払い出し処理を行う。これに対し、ベットメダルなしと判断されたときはステップ S 3 3 5 に進み、メイン CPU 5 4 は、メダル限界フラグをクリアする処理を行う。次にステップ S 3 3 6 に進み、メイン CPU 5 4 は、精算終了時の出力要求をセットする。次のステップ S 3 3 7 では、メイン CPU 5 4 は、制御コマンドセット 1 を実行する (この点の具体的説明は後述する)。そして次に、ブロッカ 4 5 をオンにする処理 (MS_BROCKER_ON) (図 1 7) に移行する。

【0333】

以上の精算処理において、ステップ S 3 1 8 及びステップ S 3 1 9 では、精算を開始したことをサブ制御基板 8 0 に送信するため、制御コマンドデータのセットを行う。

また、ステップ S 3 3 6 及びステップ S 3 3 7 では、精算を終了したことをサブ制御基板 8 0 に送信するため、制御コマンドデータのセットを行う。

【0334】

次に、ステップ S 3 1 8 及びステップ S 3 1 9 における精算開始時の処理をより詳細に説明する。

まず、ステップ S 3 1 8 (精算開始時) では、D レジスタ及び E レジスタに定義データをセットする処理を行う。図 1 8 に示すように、

D レジスタ値 = F (0 0 0 0 1 1 1 1)

10

20

30

40

50

Eレジスタ値 = 2 (00000010)

をセットする。

【0335】

次にステップS319に進み、制御コマンドセット1が実行される。

ステップS319に進むと、図19に示すステップS501以降の制御コマンドセット1 (S_CMD_SET) の処理に移行する。

【0336】

まず、ステップS502では、割込み禁止処理を実行する。次のステップS503では、制御コマンドセット2 (SS_CMD_SET) を実行する。そしてステップS504に進み、ステップS502で禁止した割込み処理を解除 (すなわち割込み許可) を実行する。以上の処理により、制御コマンドセット2の実行中は、割込み処理が禁止される。

10

【0337】

また、ステップS503に進むと、図20に示した制御コマンドセット2 (SS_CMD_SET) を実行する。

まず、ステップS511において、RWM上位アドレスセットでは、Hレジスタに、「7E (H)」 (01111110 (B)) の値をセットする。よって、

Hレジスタ値 = 7E (01111110)

となる。

【0338】

次のステップS512の「RWM指定？」では、Dレジスタに記憶された値の上位ビット、すなわち7ビット目が「0」か「1」かを判定する。上述したように、Dレジスタ値は「F」であり、2進法で示したときの7ビット目は「0」であるから、RWM指定でないと判断され、ステップS513及びS514の処理はスキップされる。

20

【0339】

次のステップS515において、制御コマンド書き込みポインタセットでは、以下の処理を行う。

1) Lレジスタに、制御コマンド書き込みポインタの番地である7E15の下位バイトデータを読み込む。ここで、下位バイトデータとは、7E15のうち、下位4ビットのデータを指す。したがって、下位バイトデータは「15」であるから、

Lレジスタ値 = 15

となる。

2) HLレジスタ値で指定される番地のデータをAレジスタに読み込む。

Hレジスタ値 = 7E、Lレジスタ値 = 15であるので、HLレジスタ値で指定される番地は、7E15である。よって、7E15番地 (制御コマンド書き込みポインタ) のデータを、Aレジスタに記憶する。本例では、7E15番地 (制御コマンド書き込みポインタ) のデータを「64 (H)」と仮定する。したがって、Aレジスタ値は「64」となる。

30

【0340】

3) Aレジスタのデータと「00001111」とをAND演算する。この演算により、「0」～「255」 (10進数) の範囲の値は、「0」～「15」 (10進数) の範囲の値に変換される。本例では、

40

Aレジスタ値 = 01100100 (64 (H))

AND演算値 = 00001111

であるので、AND演算により、「00000100」、すなわち「4 (H)」となる。

。

4) Aレジスタ値を2倍にする。この処理により、Aレジスタの値は、偶数になる。本例では、Aレジスタ値は「4」であるので、2倍にすると、Aレジスタ値は「8」となる。

。

5) Lレジスタ値を「+1」にする。この時点では、Lレジスタ値は「15」であるから、「+1」により、

Lレジスタ値 = 16

50

となる。この処理は、制御コマンドバッファの先頭番地を指定するための処理である。

【0341】

次にステップS516に進み、指定アドレスデータセット (S_ADDR_SET) 処理を実行する。ステップS516に進むと、図21の指定アドレスデータセット (S_ADDR_SET) に移行する。

【0342】

ステップS531では、「オフセット値によりワークテーブルのアドレス補正」を行う。この処理では、以下の2つを実行する。

1) Aレジスタの値と、Lレジスタの値とを加算し、Aレジスタに書き込む。

この時点において、Aレジスタ値は「8」であり、Lレジスタの値は上述のように「16」であるから、「 $8 + 16 = 1E$ 」となる。よって、

Aレジスタ値 = 1E

となる。

2) Aレジスタ値をLレジスタに読み込む。この処理により、

Lレジスタ値 = 1E

となる。

【0343】

次のステップS532における「アドレスデータ取得」では、HLレジスタ値で指定される番地のデータを、Aレジスタに書き込む。上述のように、Hレジスタ値 = 7E、Lレジスタ値 = 1Eであるから、HLレジスタ値で指定される番地は、7E1Eである。よって、7E1E番地の制御コマンドバッファの値を、Aレジスタに書き込む。

【0344】

次のステップS533の「指定アドレスデータゼロチェック」では、Aレジスタの値同士のAND演算を実行する。ここでAND演算を実行するのは、7E1E番地（制御コマンドバッファ）のデータの有無を判断するためである。データ有の場合には、AND演算をすることで、「0」以外の値となる。一方、AND演算より値が「0」であるときは、データがないことを意味する。そして、AND演算の結果、データがないときは、ゼロフラグをオンにし、データがあるときは、ゼロフラグをオフにする。そして、図20のステップS517に戻る。

【0345】

図20のステップS517の「制御コマンド数<32?」では、上述のゼロフラグのオン/オフを判断する。ゼロフラグがオンのときは「Yes」と判断し、ステップS518に進み、ゼロフラグがオフのときは、「No」と判断し、本フローチャートによる処理を終了する。すなわち、ゼロフラグがオフであるときは、HLレジスタ値で指定される番地にデータがあることを意味するので、書き込み処理を実行しないためである。

【0346】

ステップS518では、第1制御コマンドセット（第1書き込み）を実行する。具体的には、以下の処理を実行する。

1) Dレジスタ値を、HLレジスタ値で指定される番地の制御コマンドバッファに書き込む。上述のように、Dレジスタの値は、「F(00001111)」である。また、Hレジスタ値 = 7E、Lレジスタ値 = 1Eであるので、HLレジスタ値で指定される番地は、7E1Eである。したがって、

7E1E番地の制御コマンドバッファのデータ = 00001111

となる。

【0347】

2) HLレジスタ値で指定される番地のデータの7ビット目を「1」にする。HLレジスタ値で指定される番地は、上記の通り7E1Eであるので、

7E1E番地の制御コマンドバッファのデータ = 10001111 (8F)

となる。

上述したように、第1制御コマンドと第2制御コマンドとの区別のため、第1制御コマ

10

20

30

40

50

ンドの7ビット目を「1」に処理である。

【0348】

次にステップS519に進み、第2制御コマンドセット（第2書き込み）を実行する。具体的には、以下の処理を実行する。

1) Lレジスタの値を「+1」とする。この時点では、Lレジスタ値は「1E」であるから、「+1」により、

Lレジスタ値 = 1F

となる。

【0349】

2) Eレジスタの値をHLレジスタ値で指定される番地に書き込む。Eレジスタの値は、上述のように、「2(00000010)」である。また、Hレジスタ値 = 7E、Lレジスタ値 = 1Fであるので、HLレジスタ値で指定される番地は、7E1Fである。よって、

7E1F番地の制御コマンドバッファのデータ = 00000010

となる。

【0350】

以上より、

第1制御コマンドデータ(7E1E番地) = 10001111(8F)

第2制御コマンドデータ(7E1F番地) = 00000010(2)

となる。

【0351】

次のステップS520では、「制御コマンド書き込みポインタ値 = +1」を実行する。具体的には、以下の処理を実行する。

1) Lレジスタ値を「15」にする。本例では、直前のLレジスタ値は「1F」であるが、この値を「15」に変更する。よって、

Lレジスタ値 = 15

となる。

【0352】

2) HLレジスタ値で指定される番地のデータを「+1」にする。HLレジスタ値で指定される番地は、Hレジスタ値 = 7E、Lレジスタ値 = 15より、7E15である。そして、7E15番地のデータは、制御コマンド書き込みポインタである。よって、この処理により、制御コマンド書き込みポインタの値が「+1」される。本例では、上記処理により、制御コマンド書き込みポインタの値は「64」から「65」となる。

【0353】

以上のようにして、ステップS318及びステップS319では、精算開始時の制御コマンドデータ(サブ制御基板80に送信するデータ)が制御コマンドバッファに書き込まれる。そして、これらの制御コマンドデータは、後述する割込み処理により、サブ制御基板80に送信される。

【0354】

また、ステップS336及びステップS337における精算終了時の処理は、以下の通りである。

まず、ステップS336(精算終了時)では、Dレジスタ及びEレジスタに定義データをセットする処理を行う。図18に示すように、

Dレジスタ値 = F(00001111)

Eレジスタ値 = 3(00000011)

をセットする。

【0355】

次にステップS319に進み、制御コマンドセット1が実行される。この処理は、上記と同様に行われる。Eレジスタ値が精算開始時と異なるが、それ以外についてはすべて同一である。

10

20

30

40

50

これにより、

第1制御コマンドデータ=10001111(8F)

第2制御コマンドデータ=00000011(3)

となる。

したがって、ステップS336及びステップS337では、精算終了時の制御コマンドデータ(サブ制御基板80に送信するデータ)が制御コマンドバッファに書き込まれる。そして、これらの制御コマンドデータは、後述する割込み処理により、サブ制御基板80に送信される。

【0356】

なお、メインCPU54の処理において、「出力要求セット」及び「制御コマンドセット1」は、上述した説明箇所以外に、

- 1) 図12のステップS118~S119(遊技終了時の出力要求セット)
- 2) 図15のステップS148~S149(作動状態の出力要求セット)
- 3) 図22のステップS173~S174(メダル枚数表示の出力要求セット)
- 4) 図32のステップS331~S332(メダル枚数表示の出力要求セット)

が挙げられるが、これらの処理においても、上記と同様に、それぞれその処理特有の第1制御コマンド及び第2制御コマンドデータをセットしてサブ制御基板80に送信する。

【0357】

さらに、図15では説明を省略したが、遊技状態出力処理では、たとえば設定値指定時の出力要求セット及び制御コマンドセット1や、RT状態の出力要求セット及び制御コマンドセット1が実行される。

そして、出力要求に応じて異なる第1制御コマンドデータ及び第2制御コマンドデータが送信される。

【0358】

図33は、図32のステップS325における貯留メダルの精算処理(MS_CREDIT_RET)を示すフローチャートである。

図33において、まず、ステップS341では、メインCPU54は、現時点における貯留枚数の読み込み処理(S_CREDIT_READ)を行う。次にステップS342に進み、メインCPU54は、ステップS341で読み込んだ結果に基づいて、貯留メダル、すなわちクレジットの有無を判断する。貯留メダルありと判断したときはステップS343に進み、貯留メダルなしと判断したときは本フローチャートによる処理を終了する。

【0359】

ステップS343では、貯留メダルから、メダル1枚の払出し処理(MS_1MEDAL_PAY)(後述する図34~図35)を行う。次にステップS334に進み、貯留メダル枚数から1枚を減算する処理(MS_CREDIT_DEC)(図31)を行う。その後、ステップS341に戻る。

【0360】

図34及び図35は、図32のステップS327(及び図33のステップS343)におけるメダル1枚の払出し処理(MS_1MEDAL_PAY)を示すフローチャートである。図35は、図34に続くフローチャートである。

図34のステップS351では、メインCPU54は、エラー未検出をセットする。すなわち、初期状態ではエラーを検出していない状態をセットする。次のステップS352では、メインCPU54は、メダル詰まりエラー表示要求をセットする。

【0361】

次にステップS353に進み、メインCPU54は、エラーを検出したか否かを判断する。エラーを検出したと判断したときはステップS354に進んでエラー表示を行う。そしてステップS355に進む。一方、ステップS353でエラーを検出していないと判断したときはステップS355に進む。

ステップS355では、メインCPU54は、メダル払出し装置の制御時間をセットする。ここでは、制御時間の計時を開始する。次にステップS356に進み、ホッパーモー

10

20

30

40

50

タ 3 6 を駆動する。次にステップ S 3 5 7 に進み、メイン C P U 5 4 は、ステップ S 3 5 5 でセット後の制御時間の読み込みを行う。

【 0 3 6 2 】

そして、次のステップ S 3 5 8 において、メイン C P U 5 4 は、制御時間が所定時間を経過したか否かを判断する。所定時間を経過したと判断したときはステップ S 3 6 1 に進み、所定時間を経過していないと判断したときはステップ S 3 5 9 に進む。

ステップ S 3 6 1 では、ホッパーモータ 3 6 の駆動信号をオフにする処理を行う。そして、次のステップ S 3 6 2 では、払出しセンサ 1 の検出時間をセットする。ここでは、検出時間の計時を開始する。次にステップ S 3 6 3 に進み、メイン C P U 5 4 は、払出しセンサ 1 信号に係るデータがオンであるか否かを判断する。オンであると判断したときはステップ S 3 5 5 に進み、オンでないと判断したときはステップ S 3 6 4 に進む。

10

【 0 3 6 3 】

ステップ S 3 6 4 では、メイン C P U 5 4 は、払出しセンサ 1 の検出時間が所定時間を経過したか否かを判断する。所定時間を経過したと判断したとき、すなわち払出しセンサ 1 信号に係るデータがオンでなく、かつ払出しセンサ 1 の検出時間が所定時間を経過したときは、メダルが払い出されていないこととなるので、ステップ S 3 6 5 に進み、メイン C P U 5 4 は、メダル空エラーの表示要求をセットする。そしてステップ S 3 5 3 に進む。一方、ステップ S 3 6 4 において払出しセンサ 1 の検出時間が所定時間を経過していないと判断したときはステップ S 3 6 3 に戻る。

【 0 3 6 4 】

20

ステップ S 3 5 8 において制御時間が所定時間を経過していないと判断され、ステップ S 3 5 9 に進むと、メイン C P U 5 4 は、払出しセンサ 1 信号に係るデータがオンであるか否かを判断する。オンであると判断したときはステップ S 3 6 0 に進み、オンでないと判断したときはステップ S 3 5 7 に戻る。

【 0 3 6 5 】

ステップ S 3 6 0 では、払出しセンサ 1 の検出時間をセットする。この処理は、ステップ S 3 6 2 と同様である。次にステップ S 3 6 6 に進み、メイン C P U 5 4 は、メダル詰まりを検出したか否かを判断する。メダル詰まりを検出したと判断したときはステップ S 3 5 2 に進み、検出していないと判断したときはステップ S 3 6 7 に進む。

ステップ S 3 6 7 では、メイン C P U 5 4 は、払出しセンサ 1 信号に係るデータがオンであるか否かを判断する。オンであると判断されたときはステップ S 3 6 8 に進み、オンでないと判断されたときはステップ S ステップ S 3 5 7 に進む。

30

【 0 3 6 6 】

ステップ S 3 6 8 では、メイン C P U 5 4 は、払出しセンサ 1 及び 2 信号に係るデータがオンであるか否かを判断する。双方がオンであると判断されたときはステップ S 3 6 9 に進み、双方がオンでないと判断されたときはステップ S 3 6 6 に戻る。

ステップ S 3 6 9 では、メイン C P U 5 4 は、払出しセンサ 2 の検出時間をセットする。次に、図 3 5 のステップ S 3 7 0 に進み、メイン C P U 5 4 は、メダル詰まりを検出したか否かを判断する。メダル詰まりを検出したと判断したときはステップ S 3 5 2 に進み、検出していないと判断したときはステップ S 3 7 1 に進む。

40

【 0 3 6 7 】

ステップ S 3 7 1 では、メイン C P U 5 4 は、払出しセンサ 2 信号に係るデータがオフであるか否かを判断する。払出しセンサ 2 信号に係るデータがオフでないと判断したときはステップ S 3 7 0 に戻り、オフであると判断したときはステップ S 3 7 2 に進む。

ステップ S 3 7 2 では、メイン C P U 5 4 は、メダル払出しが無効であるか否かを判断する。無効であると判断したときはステップ S 3 5 7 に進み、無効でないと判断したときはステップ S 3 7 3 に進む。ステップ S 3 7 3 では、メイン C P U 5 4 は、払出しセンサ 1 信号に係るデータがオフであるか否かを判断する。オフであると判断したときはステップ S 3 7 4 に進み、オフでないと判断したときはステップ S 3 7 0 に戻る。

【 0 3 6 8 】

50

ステップ S 3 7 4 では、残り払出し数（カウント値）を「-1」（「1」減算）する。次にステップ S 3 7 5 に進み、メイン C P U 5 4 は、メダル払出しを終了したか否かを判断する。メダル払出しを終了したと判断したときはステップ S 3 7 6 に進み、終了していないと判断したときは本フローチャートを終了する。ステップ S 3 7 6 では、メイン C P U 5 4 は、ホッパーモータ 3 6 の駆動信号をオフにする。そして本フローチャートによる処理を終了する。

【0369】

次に、メイン C P U 5 4 で行われる割込み処理について説明する。図 3 6 は、本実施形態の割込み処理を示すフローチャートである。

ステップ S 6 0 1 で割込み処理が実行されると、次のステップ S 6 0 2 では、割込みフラグをクリアする。割込みフラグは、上述したように、割込み処理を禁止した後、次の割込み処理を実行するタイミングが到来したときにオンにされるフラグである。そして、メイン C P U 5 4 は、禁止した割込み処理を解除するときは、その後、割込みフラグをクリアする。

【0370】

次のステップ S 6 0 3 では、入力ポート 5 1 b（ただし、これに限らず、全入力ポートでもよい）のデータ入力を行う。この処理は、入力ポート 5 1 b の D 0 ビットの信号を入力するために行う。次のステップ S 6 0 4 では、ステップ S 6 0 3 で入力した入力ポート 5 1 b の D 0 ビットの信号（電断信号）を正論理に変換し、「1」であるか否かを判断する。「1」であるときは、電断の発生を検知したと判断する。

なお、ステップ S 6 0 3 における入力ポート 5 1 b のデータ入力は、上述の 2 回のデータ取得を行わず、1 回のみ取得する。

ステップ S 6 0 4 において電断の発生を検知したときはステップ S 6 0 5 に進んで電断処理（IS_POWER_DOWNM）を実行し（図 3 7）、ステップ S 6 0 6 に進む。これに対し、電断を検知しないときはステップ S 6 0 5 をスキップしてステップ S 6 0 6 に進む。

【0371】

ステップ S 6 0 6 では、全入力ポート（本実施形態では 5 1 a 及び 5 1 b）のデータ入力・生成処理を行う。次のステップ S 6 0 7 では、全出力ポート（本実施形態では 5 2 b）のデータ入力・生成を行う。そして、ステップ S 6 0 8 に進む。

【0372】

ステップ S 6 0 8 では、制御コマンド読み込みポインタの取得を実行する。具体的には、以下の処理を実行する。

1) D 及び E レジスタに、制御コマンド読み込みポインタ（_NB_CMDREAD）の番地の上位及び下位の値を読み込む。図 1 1 に示すように、制御コマンド読み込みポインタの番地は、7 E 1 4 である。よって、

D レジスタ値 = 7 E

E レジスタ値 = 1 4

となる。

【0373】

2) H 及び L レジスタに、制御コマンドバッファの先頭番地の上位及び下位の値を読み込む。制御コマンドバッファの先頭番地は、図 1 1 に示すように、7 E 1 6 である。よって、

H レジスタ値 = 7 E

L レジスタ値 = 1 6

となる。

3) D E レジスタ値で指定される番地のデータを A レジスタに書き込む。D E レジスタ値は、上記のように 7 E 1 4 であるので、7 E 1 4 番地のデータ、すなわち制御コマンド読み込みポインタのデータ（0 ~ F F（H））を A レジスタに書き込む。ここで、7 E 1 4 番地の値を「5（H）」と仮定すると、

A レジスタ値 = 5

となる。

4) Aレジスタ値と「00011110」(定義データ)とをAND演算し、その値をAレジスタに書き込む。本例では、

Aレジスタ値 = 00000101 (5 (H))

AND演算値 = 00011110

であるので、AND演算後の値は、「00000100」(4 (H))となる。この処理により、Aレジスタ値は、常に偶数となる。

【0374】

次にステップS609に進み、指定アドレスデータセット(S_ADDR_SET)を行う。この指定アドレスデータセットは、図21に示したフローをそのまま使用する(プログラム容量の削減のためである)。よって、ステップS609に進むと、図21のステップS531に移行する。

まず、ステップS531における「オフセット値によりワークテーブルのアドレス補正」では、以下の処理を実行する。

【0375】

1) Aレジスタ値とLレジスタ値とを加算し、その加算後の値をAレジスタに書き込む。本例では、この時点におけるAレジスタ値は「4」であるので、

Aレジスタ値 = $4 + 16 = 1A$

となる。

2) Aレジスタ値をLレジスタに読み込む。よって、

Lレジスタ値 = 1A

となる。

【0376】

次のステップS532におけるアドレスデータ取得では、HLレジスタ値で指定される番地のデータをAレジスタに書き込む。この例では、Hレジスタ値 = 7E、Lレジスタ値 = 1Aであるので、HLレジスタ値 = 7E1Aとなる。よって、7E1A(制御コマンドバッファ)番地のデータをAレジスタに記憶する。

【0377】

次のステップS533における指定アドレスデータゼロチェックでは、Aレジスタ値同士をAND演算する。AND演算した結果、「0」のときは、Aレジスタにデータがないことを意味する。この場合には、上述したゼロフラグをオンにする。

一方、AND演算した結果、「0」でないときは、Aレジスタにデータがあることを意味する。この場合には、ゼロフラグをオフにする。

そして、図36のステップS610に戻る。

【0378】

ステップS610の「制御コマンド有？」では、ゼロフラグのオン/オフを判断する処理を行う。ゼロフラグがオンであるときは、HLレジスタ値で指定される番地のデータ(制御コマンドデータ)がないことを意味するので、送信すべき制御コマンドデータなしと判断する。よって、本フローチャートによる処理を終了する。

これに対し、ゼロフラグがオフであるときは、送信すべき制御コマンドデータがあると判断し、ステップS611に進む。

【0379】

ステップS611の「第1コマンドライト」では、以下の処理を実行する。

1) DEレジスタ値と、HLレジスタ値とを交換する。上記の例では、

DEレジスタ値 = 7E14

HLレジスタ値 = 7E1A

であるので、

DEレジスタ値 = 7E1A

HLレジスタ値 = 7E14

とする。

2) 7 E 1 A 番地のデータ (第 1 制御コマンドデータ) を送信用レジスタの番地に書き込む。これにより、第 1 制御コマンドデータが送信される。

【0380】

次のステップ S 6 1 2 における「第 2 コマンドライト」では、以下の処理を実行する。

1) E レジスタ値を「+1」する。上記の例では、E レジスタ値 = 1 A であるので、E レジスタ値 = 1 B となる。

2) D E レジスタ値で指定される番地のデータを A レジスタに書き込む。この例では、E レジスタ値 = 1 B であるので、D E レジスタ値 = 7 E 1 B である。よって、7 E 1 B 番地のデータ (第 2 制御コマンドデータ) を A レジスタに書き込む。

3) A レジスタの値 (第 2 制御コマンドデータ) を送信用レジスタの番地に書き込む。これにより、第 2 制御コマンドデータが送信される。

10

【0381】

次にステップ S 6 1 3 に進み、制御コマンド送信処理が完了したか否かを判断する。ここで、制御コマンドデータは、1 個のデータにつき、2 回送信するため、2 回の送信が完了したか否かを判断する。具体的には、7 E 1 4 番地のデータ (制御コマンド読み込みポインタ値) の 0 ビット目が「0」か「1」かを判断する。ここで、「0 (偶数)」であるときは、送信 1 回目であり、「1 (奇数)」であるときは、送信 2 回目であると判断する。

そして、送信が 2 回目であるときは、送信が完了したと判断し、ステップ S 6 1 4 に進む。これに対し、送信が 1 回目であるときは、送信が完了していないと判断し、ステップ S 6 1 5 に進む。

20

【0382】

ステップ S 6 1 4 のコマンド保存バッファクリアでは、以下の処理を実行する。

1) A レジスタ値同士を X O R 演算し、その値を A レジスタに読み込む。ここで、同一値同士を X O R 演算すると、常に値が「0」となる。よって、

A レジスタ値 = 0
となる。

【0383】

2) D E レジスタ値で指定される番地に、A レジスタ値を読み込む。D レジスタ値 = 7 E、E レジスタ値 = 1 B であるので、D E レジスタ値 = 7 E 1 B である。したがって、7 E 1 B 番地の制御コマンドバッファに、A レジスタ値 (「0」) を読み込む。これにより、送信済みの第 2 制御コマンドデータ (7 E 1 B 番地の制御コマンドバッファに記憶されていたデータ) がクリアされる。

30

【0384】

3) E レジスタ値を「-1」する。E レジスタ値は、この例では「1 B」であるので、E レジスタ値 = 1 A となる。

4) D E レジスタ値で指定される番地に、A レジスタ値を読み込む。D レジスタ値 = 7 E、E レジスタ値 = 1 A であるので、D E レジスタ値 = 7 E 1 A である。したがって、7 E 1 A 番地の制御コマンドバッファに、A レジスタ値 (「0」) を読み込む。これにより、送信済みの第 1 制御コマンドデータ (7 E 1 A 番地の制御コマンドバッファに記憶されていたデータ) がクリアされる。

40

【0385】

したがって、第 2 制御コマンドデータをクリアした後、第 1 制御コマンドデータをクリアするので、制御コマンド読み込みポインタを移すことなく、送信済みの 2 つの制御コマンドデータ (7 E 1 A 及び 7 E 1 B 番地のデータ) を消去することができる。

【0386】

次のステップ S 6 1 5 では、制御コマンド読み込みポインタ値を「+1」にする。ここでは、H L レジスタ値で指定される番地のデータを「+1」にする。上記のように、H L レジスタ値 = 7 E 1 4 であるので、7 E 1 4 番地のデータ、すなわち制御コマンド読み込みポインタ値を「+1」にする。本例では、7 E 1 4 番地のデータは「5 (H)」である

50

ので、上記処理により、7 E 1 4 番地のデータは、「5」から「6」となる。

そして、本フローチャートによる処理を実行する。

【0387】

以上の制御コマンドデータの送信処理において、ステップS 6 0 8における「制御コマンド読み込みポインタの取得」中、4)では、Aレジスタ値と定義データとをAND演算し、その値をAレジスタに書き込むことで、Aレジスタ値を常に偶数となるようにしている。この結果、7 E 1 4 番地の制御コマンド読み込みポインタの値が読み取られ、Aレジスタに記憶された後、その値は常に偶数にされる。

【0388】

したがって、たとえば制御コマンド読み込みポインタの値が「0」であるときは、AND演算後も「0」であるが、制御コマンド読み込みポインタの値が「+1」にされ、「1」となったときは、AND演算後は「0」となる。よって、制御コマンド読み込みポインタの値が「n」であるときと、「n+1」であるときとで、同一番地の制御コマンドバッファが指定されるので、同一の制御コマンドデータの送信処理が2回行われる。

よって、一割込みで、第1制御コマンドデータ及び第2制御コマンドデータの2バイトが送信されるとともに、前記一割込みの次の割込みで、前記一割込みで送信した制御コマンドと同一の第1制御コマンドデータ及び第2制御コマンドデータの2バイトが送信される。

【0389】

一方、この制御コマンドデータを受信したサブ制御基板80側では、2回受信した制御コマンドが同一であるか否かを判断し、同一であるときは、その制御コマンドの一方を記憶し、同一でないときはその制御コマンドを記憶しない。したがって、2回の制御コマンドデータの送信において、ノイズの発生等により、同一でない制御コマンドデータが送信されたときは、サブ制御基板80側では、その制御コマンドデータに基づく演出処理を実行しないので、誤りのある制御コマンドデータに基づく演出の実行を防止することができる。

【0390】

図37は、図36のステップS 6 0 5の電断処理（IS_POWER_DOWN）を示すフローチャートである。

図37において、ステップS 6 2 1では、ブロック信号及びホッパーモータ駆動信号をクリアする。電断が発生すると、図7中、入力ポート51bのD0ビットに電断信号が入力され、7 E 0 2 番地のD0ビット値が「1」にされる（図8）。

また、ステップS 6 2 1の処理では、7 E 0 3 番地のD6及びD7ビットのデータがクリア（「0」に）される。

【0391】

次のステップS 6 2 2では、スタックポインタを保存する。正常時には、図12のステップS 1 0 1において、電断発生時のデータを保存するためのRWM領域がセットされるので、このRWM領域に電断発生時のデータを保存する処理を行う。

次のステップS 6 2 3では、コマンド送信履歴フラグをクリアする。この処理は、7 E 1 4 番地（_NB_CMDREAD）のデータ、すなわち制御コマンド読み込みポインタ値の0ビット目を「0」にする処理である。

【0392】

たとえば、制御コマンド読み込みポインタ値が「11」（奇数）であるときに電断が発生したときは、ステップS 6 2 3の処理により、制御コマンド読み込みポインタ値が「10」（偶数）にされる。これにより、制御コマンド読み込みポインタ値＝「10」で1回目の制御コマンドデータが送信され、制御コマンド読み込みポインタ値＝「+1」、すなわち「11」になったときに2回目の制御コマンドデータが送信される。よって、制御コマンドデータの送信1回目と2回目との間に電断が発生しても、制御コマンドデータを正しく送信することができる。

【0393】

10

20

30

40

50

次のステップ S 6 2 4 では、R W M、すなわちメモリ 5 3 の書き込みを禁止する。そして、ステップ S 6 2 5 に進み、リセット待ち状態とする。

なお、リセット後は、再度、ステップ S 6 0 1 以降の割込み処理を実行するので、この割込み処理において、制御コマンド読み込みポインタ値を改めて読み取り、その番地の制御コマンドバッファに制御コマンドデータが記憶されているときは、送信処理を実行する。

以上より、制御コマンド読み込みポインタ値が奇数値のときに電断が発生し、制御コマンドバッファにデータが残っているときは、制御コマンド読み込みポインタ値が偶数にされるとともに、制御コマンドバッファに残っているデータを、改めて 2 回送信することになる。

10

【0394】

特に、電断発生時に、制御コマンド読み込みポインタ値の 0 ビット目を「0」にするだけで、必ず 2 回送信することができる。このように処理すれば、制御コマンド読み込みポインタ値が偶数値のときに電断が発生しても、電断発生時が送信 1 回目のときか 2 回目のときかを判断することなく、必ず 2 回送信することができる。

【0395】

続いて、サブ制御基板 8 0 側での制御コマンドデータの受信処理について説明する。図 3 8 は、制御コマンドデータの受信処理を説明する図である。

図 3 8 において、第 1 制御コマンドが「9 0 (1 0 0 1 0 0 0 0)」、第 2 制御コマンドが「7 1 (0 1 1 1 0 0 0 1)」のデータを受信するものとする。

20

【0396】

サブ制御基板 8 0 側には、制御コマンドのバッファ領域として、1 6 ビットからなる記憶領域が設けられている。最初に第 1 制御コマンドデータを受信するが、受信した第 1 制御コマンドデータは、図 3 8 に示すように、1 6 ビット中、下位 8 ビットに記憶する。

ここで、メイン制御基板 5 0 から送信される第 1 制御コマンドの 7 ビット目は、常に「1」である。よって、7 ビット目が「1」であることを確認することで、第 1 制御コマンドデータであることを確認する。そして、その確認後、すでに記憶している第 1 制御コマンドデータを、下位 8 ビットから上位 8 ビットに移動させる。

【0397】

次に第 2 制御コマンドを受信したときは、下位 8 ビットに、受信した第 2 制御コマンドデータを記憶する。ここで、メイン制御基板 5 0 から送信される第 2 制御コマンドデータの 7 ビット目は、常に「0」である。よって、7 ビット目が「0」であることを確認することで、受信した制御コマンドデータが第 2 制御コマンドデータであることを確認する。そして、その確認後、第 2 制御コマンドデータの 7 ビット目を「0」から「1」に変換する処理を行う。

30

【0398】

また、図 3 8 に示すように、サブ制御基板 8 0 側においても、メモリ 8 3 内に、制御コマンド読み込みポインタを記憶する領域 (7 E 1 4 番地)、制御コマンド書き込みポインタを記憶する領域 (7 E 1 5 番地)、制御コマンドバッファ (7 E 1 6 ~ 7 E 3 5 番地) を備える。

40

【0399】

メイン制御基板 5 0 から制御コマンドを受信したときは、制御コマンド書き込みポインタで指定された番地の制御コマンドバッファに、第 1 及び第 2 制御コマンドデータを書き込む。また、受信した制御コマンドに基づいてサブ制御基板 8 0 側の演出処理を実行するときは、制御コマンド読み込みポインタで指定された番地の制御コマンドバッファから、第 1 及び第 2 制御コマンドデータを読み込む。

【0400】

上記の制御コマンドデータの受信において、サブ制御基板 8 0 は、制御コマンドを受信するごとに所定のカウンターを用いて受信回数をカウントする。そして、カウント値が定められた値に到達すると、1 つの制御コマンドデータを受信したと判断する。たとえば一

50

対の制御コマンドデータが本実施形態のように第1及び第2制御コマンドデータからなり、かつ本実施形態のように同一の制御コマンドデータを2回送信する場合には、カウント値が「4」となったときに、1つの制御コマンドデータを受信したと判断する。

【0401】

ここで、第1及び第2制御コマンドデータを受信したとき、その制御コマンドデータ（16ビット）と、「8080（H）（15ビット目及び7ビット目が「1」で、それ以外は「0」のデータ）」とをAND演算し、「8080（H）」になるかを確認する。これにより、第1及び第2制御コマンドを受信したか否かを確認する（受信済みチェックする）ことができる。

【0402】

そして、制御コマンドデータがいわゆるコマンドごけにより正常に受信できずに抜けてしまったとき、たとえば第1制御コマンドデータを受信した後、次に受信した制御コマンドデータが別のメインコマンドに係る第1制御コマンドデータであったとき等は、不完全な制御コマンドデータであるので、処理されない。そして、次の制御コマンドデータによって上書きされてしまうため、コマンドごけした制御コマンドデータは、処理されない場合がある。

そこで、本実施形態では、第1及び第2制御コマンドデータの2回送信を行うことで、コマンドごけにより処理されない制御コマンドデータが発生してしまうことを防止している。

【0403】

また、第1及び第2制御コマンドデータを書き込んだときは、制御コマンド書き込みポインタ値を「+1」する。同様に、第1及び第2制御コマンドデータを読み取ったとき（制御コマンドデータに基づく演出処理を実行したとき）は、制御コマンド読み込みポインタ値を「+1」する。さらに、読み取った制御コマンドバッファの記憶領域をクリアする。

制御コマンド書き込みポインタ及び制御コマンド読み込みポインタのいずれも、指定番地が7E35であるときにポインタ値を「+1」したときは、7E16となるように設定されている。

【0404】

サブCPU84は、16msごとに、制御コマンドバッファに、実行待ちの制御コマンドデータが記憶されているか否かの判断を行う。制御コマンド読み込みポインタ値で指定される番地に制御コマンドデータが記憶されているときは、その制御コマンドデータに基づく演出処理を実行する。

さらに、制御コマンド読み込みポインタ値と制御コマンド書き込みポインタ値とが一致したときは、実行待ちの制御コマンドデータがないと判断する。

【0405】

図39は、受信した制御コマンドデータと、それに対応する処理関数及び引数とを定めたデータテーブルを示す図である。

たとえば、貯留ベット1枚が行われたとき（1ベットスイッチ40aが操作されたとき）には、受信する第1制御コマンドは「90（10010000）」であり、第2制御コマンドは「01（00000001）」である。すなわち、16ビットからなる制御コマンドデータは、「9001（H）」となる。

そして、図39に示すように、受信した制御コマンドデータが「9001」であったときは、このデータテーブルによって、その制御コマンドデータに対応する処理関数は「Cmd_First10」、引数は「0x01」とであると判断される。そして、その関数及び引数を呼び出して、プログラムを処理する。

【0406】

図40は、本発明のスロットマシン10を含む遊技システム1の一実施形態を示す概要図である。図40において、遊技システム1は、スロットマシン10と、携帯通信端末100と、サーバーコンピュータ200とから構成されている。

【0407】

携帯通信端末100は、サーバーコンピュータ200と通信機能を有する携帯電話機、モバイルコンピュータ（ノートブック型パーソナルコンピュータ等）、パーソナルデジタルアシスタント（PDA；携帯情報端末。いわゆるスマートフォンを含む。）等であり、特に本実施形態では、例えば有効画素数が少なくとも数10万ピクセル程度の解像度を有するカメラ（CMOSセンサ等）を内蔵し、二次元コードの読取り機能を有するものである。

【0408】

さらに、携帯通信端末100には、情報表示用のディスプレイ（液晶ディスプレイ、有機EL（エレクトロルミネッセンスディスプレイ）等）、入力キー（数字、英文字、平仮名、カタカナ、クリア、リターン等のキー。また、キーとして機能するタッチセンサーキーを含む。）、及びデータの（電波による）送受信装置等を備える。

また、携帯通信端末100には、携帯通信端末100ごとに固有番号が割り当てられている。固有番号は、たとえば携帯電話機で使用されている電話番号を特定するための固有のID番号が挙げられる。以下の実施形態では、SIM（Subscriber Identity Module）IDを用いている。

【0409】

そして、本実施形態の遊技システム1を利用して、マイスロ遊技（後述）を行う場合には、スロットマシン10の遊技者は、携帯通信端末100を所持していることが必要である。ただし、携帯通信端末100を所持していなくても、スロットマシン10で遊技を行うこと自体はもちろん可能である。

【0410】

また、サーバーコンピュータ200は、携帯通信端末100とデータの送受信が可能なコンピュータ（ホールとは別に設置された外部コンピュータ）であり、独自のウェブサイト所有している。さらにまた、サーバーコンピュータ200内には、ハードディスクドライブ等の記憶手段が設けられており、各遊技者ごとのデータ記憶領域を有している。そして、サーバーコンピュータ200は、携帯通信端末100からデータのダウンロードの要求があると、それに応じて、携帯通信端末100に対し、データの送信を行う。また、携帯通信端末100からアップロードされてきたデータ（たとえば、遊技履歴）を記憶可能である。

【0411】

図1において、サブ制御基板80のサブCPU84は、マイスロ制御手段92を備える。ここで、「マイスロ遊技」とは、以下の内容である。

スロットマシン10は、遊技者の遊技履歴（遊技回数、ボーナス当選回数等）を記憶しておく。遊技終了時に、スロットマシン10は、遊技履歴を二次元コードとして画像表示する。

【0412】

遊技者は、その二次元コードを自身の携帯通信端末100で読み取り、サーバーコンピュータ200にアクセスする。サーバーコンピュータ200には、その遊技者の最新の遊技履歴が記憶される。また、サーバーコンピュータ200は、遊技者に対し、パスワードを発行する。遊技者は、遊技開始時に、そのパスワードを入力して遊技を開始すれば、遊技履歴を引き継ぐことができる、というものである。

以上のような遊技を、本明細書では「マイスロ遊技」と称する。

【0413】

また、マイスロ制御手段92は、マイスロ遊技の開始時に、その遊技者に対して、その遊技者の遊技実績等の情報を含むパスワード（サーバーコンピュータ200によって発行されたもの）を入力させるための入力画面を画像表示する。

【0414】

さらにまた、マイスロ制御手段92は、入力されたパスワードを読み取り、その遊技者の遊技実績等の情報を一時的に記憶するとともに、遊技の進行に伴って、その遊技者の遊

10

20

30

40

50

技実績（遊技回数等）を更新しつつ記憶する。

【0415】

さらに、遊技者がたとえば特定の遊技実績を達成したとき（たとえば、予め定めた所定のミッションをクリアしたとき）等には、特定の遊技実績を達成した旨の情報を遊技履歴として記憶する。マイスロ制御手段92は、遊技者がマイスロ遊技を終了するときに、サーバーコンピュータ200にアクセスするためのコードであって、パスワードが入力されてからの遊技実績（遊技履歴）を含む二次元コードを生成し、画像表示装置23に画像表示するように制御する。

【0416】

図41～図43は、マイスロ遊技の手順を説明する図である。

図41は、マイスロ遊技を初めて行うときに、遊技者が新規に会員登録をするときの操作方法を説明する図である。

サブCPU84は、遊技が行われていないとき（遊技待機中）、すなわち全てのリール31が停止している状態において、エンターキー85bが操作されたときは、画像表示装置23に、メニュー画面を表示するように制御する。

【0417】

図41は、メニュー画面の例を示している。図41の例では、メニューとして、「会員登録」、「パスワード入力」、「二次元コード表示」、及び「終了」の4項目が設けられている。そして、サブCPU84は、これら4つの項目のうち、いずれか1が選択されたときは、さらにその下位メニューに係る画面を画像表示装置23に画像表示するように制御する。

【0418】

マイスロ遊技を初めて行う場合には、遊技者は、会員登録を行う。図41のメニュー画面において「会員登録」を選択する（十字キー85aでカーソルを「会員登録」に配置させ、その位置でエンターキー85bを押す）と、マイスロ制御手段92は、図41に示すように、会員登録用の二次元コードを画像表示する（なお、この二次元コードを画像表示するためのデータは、予めマイスロ制御手段92に設けられている。）。

【0419】

この二次元コードには、サーバーコンピュータ200のウェブサイトのURL（ユニフォームリソースロケータ；ウェブページのアドレス）の情報が含まれる。そして、遊技者は、携帯通信端末100のカメラによってこの二次元コードを読み取る。二次元コードが読み取られると、携帯通信端末100は、この二次元コードをURLに変換し、携帯通信端末100のディスプレイに表示する。

なお、二次元コードをカメラで読み取る方法ではなく、例えば文字コードを表示し、遊技者が、その文字コードを携帯通信端末100に手入力するようにしてもよい。さらに、光・音声通信によるものを採用してもよい。

【0420】

そして、表示されたURLを指定すれば、携帯通信端末100からサーバーコンピュータ200のウェブサイトにアクセスすることができる。サーバーコンピュータ200は、上記のようにして携帯通信端末100からアクセスがあると、そのときに送信されてきたデータに基づいて、会員登録の要求があったと判断する。そして、サーバーコンピュータ200は、会員登録のページを表示するデータを携帯通信端末100に送信する。

【0421】

会員登録時には、遊技者は、所定事項（自己の氏名等）を入力し、会員登録を実行すると、入力された情報がサーバーコンピュータ200に送信される。サーバーコンピュータ200は、この情報を受信すると、登録された会員専用の記憶領域を確保し、これらの情報を記憶する。また、正しく会員登録された旨を携帯通信端末100に送信する（図41）。

一方、スロットマシン10では、エンターキー85bをオンすると、メニュー画面から、遊技待機画面に戻る。

10

20

30

40

50

【0422】

一方、サーバーコンピュータ200は、携帯通信端末100からの要求に応じて、この会員専用のパスワードを発行し、それを携帯通信端末100に送信する(図42)。ここで発行されるパスワードは、会員ID(遊技者を特定可能な個人情報)、この遊技者固有のURL、及び遊技実績(累積遊技回数、累積差枚数、ボーナス当選回数等の遊技履歴)を含むデータである。

【0423】

図42は、マイスロ遊技の開始時における操作内容を説明する図である。図42では、携帯通信端末100のディスプレイにパスワードが表示された例を示している。

遊技者は、遊技開始時には、エンターキー85bをオンし、メニュー画面を表示させ、このメニュー画面から「パスワード入力」を選択する。「パスワード入力」が選択されると、マイスロ制御手段92は、図42に示すように、パスワード入力画面を画像表示するように制御する。そして、遊技者は、メニューキー85を使用して、サーバーコンピュータ200によって発行されたパスワード(図42の例では、「BEAST」)を入力する。

10

【0424】

パスワードを入力したら、エンターキー85bをオンして入力内容を確定させると、パスワードの入力が完了した旨の画像表示を行うとともに、入力したパスワードがスロットマシン10(サブ制御基板80)に読み取られる。そして、再度、エンターキー85bをオンされると、サブCPU84は、メニュー画面を終了して、遊技待機中のデモンストレーション(以下、「デモ」と略称する。)画面を画像表示する。

20

【0425】

一方、サブCPU84は、パスワードが入力されると、そのパスワードに基づいて、会員ID、及び遊技履歴等の情報を取得し、メモリ83に記憶する。さらに、サブCPU84は、上記のようにしてパスワードが入力されてから、遊技履歴を遊技ごとに更新・記憶していくように制御する。たとえば遊技が行われるごとに、累積遊技回数、差枚数、ボーナス当選回数、ミッションクリア情報等を更新するように制御する。

【0426】

図43は、マイスロ遊技の終了時における操作内容を説明する図である。

遊技者は、遊技を終了するときは、図43に示すように、遊技待機中(非遊技中)に、再度、エンターキー85bをオンして、メニュー画面を画像表示装置23に画像表示する。

30

次に、遊技者は、メニューキー85を用いて「二次元コード表示」の項目を選択する。この項目が選択されると、マイスロ制御手段92は、遊技の開始時に入力されたパスワード、パスワードが入力されてからの遊技実績、及びこの遊技者固有のURLのデータを含む二次元コードを生成するように制御する。そして、マイスロ制御手段92は、作成した二次元コードを画像表示するように制御する。図43は、このときの画像表示例を示している。

【0427】

次いで、遊技者は、画像表示された二次元コードを、携帯通信端末100のカメラで読み取る。これにより、携帯通信端末100には、読み取られた二次元コードに対応するURL(サーバーコンピュータ200に記憶された、その会員専用の記憶領域内のデータフォルダー)が表示される。

40

そして、このURLをオンすると、サーバーコンピュータ200にアクセスされるとともに、サーバーコンピュータ200に記憶された、その会員固有のフォルダー内のデータが更新される。

【0428】

これにより、遊技者が、携帯通信端末100にて「パスワード表示」の機能を選択すると、それまでのその会員の遊技実績を反映した最新のパスワードが表示される(図42)。そして、ここで表示されたパスワードを、「パスワード入力」画面で入力すれば、前回

50

の遊技実績から遊技を開始することができる。すなわち、遊技者は、台を移動するときは、上述のようにして一旦、二次元コードを読み取り、それに基づいて新たなパスワードを取得し、このパスワードを、次に遊技を行うスロットマシン10に入力することで、今までの遊技実績を受け継いで（最新の遊技履歴で）遊技を行うことができる。

【0429】

なお、マイスロ制御手段92は、図43の画面（二次元コード）を画像表示した後は、自動で、例えば120秒経過後に、この画面を終了して、初期画面（遊技待機画面）に戻るよう制御する。したがって、遊技者は、二次元コードを携帯通信端末100のカメラで読み取り、それをサーバーコンピュータ200に送信した後は、それ以後は特に何もせずに席を離れることができる。さらに、マイスロ制御手段92は、二次元コードを作成して画像表示したときは、その後、それまでの遊技者に関する情報（メモリ83に記憶されている情報）を消去するように制御する。これにより、次にこの台で遊技を行う遊技者に対しては、それ以前の遊技者のデータが全く残らないので、プライバシー及びセキュリティ性が担保される。

なお、二次元コードの画面表示時から初期画面に戻るまでの秒数は、遊技終了後に無操作の場合に初期画面に戻るまでの秒数よりは長く設定される。

【0430】

また、サーバーコンピュータ200は、パスワードの発行時には、乱数を使用する。

具体的には、サーバーコンピュータ200がパスワードを発行する場合には、0～127の7ビット乱数で生成した乱数値を決定し、その乱数値を含むパスワードを生成し、遊技者に送信（発行）する。なお、乱数値の決定は、パスワードの発行時ごとに行われる。

サーバーコンピュータ200は、このときに決定した乱数値を併せて記憶する。図40の例では、遊技者Aに対してパスワードを発行したときに、乱数値を「127」に決定した例を示している。

【0431】

遊技者は、パスワードを携帯通信端末100で受信し、そのパスワードをスロットマシン10に入力すると、スロットマシン10では、入力されたパスワードから乱数値を特定し、その乱数値も併せて記憶する。

【0432】

そして、遊技者がそのスロットマシン10で遊技を終了するために、二次元コードの発行を要求したときは、遊技開始時に入力されたパスワードに含まれていた乱数値の情報を含む二次元コードを発行する。

二次元コードが携帯通信端末100で読み取られ、この二次元コードがURLに変換されてサーバーコンピュータ200にアクセスがあったときは、その二次元コードに含まれる乱数値と、サーバーコンピュータ200に記憶されている乱数値（最後に発行したパスワードのときに決定した乱数値）とが一致するか否かを判断し、一致していると判断したときは、その遊技者の遊技履歴を更新するように制御する。したがって、二次元コードに基づきサーバーコンピュータ200にアクセスがあっても、乱数値が一致しなければ、遊技者の遊技履歴は更新されない。

【0433】

さらに本実施形態では、携帯通信端末100からサーバーコンピュータ200にアクセスがあったときは、サーバーコンピュータ200は、その携帯通信端末100の固有のID、具体的には上述したSIMIDを読み取る。そして、サーバーコンピュータ200は、SIMIDと乱数値との双方が一致していれば、遊技履歴の更新を許可するように制御する。図40の例では、遊技者AのSIMIDは「123456」であり、前回発行したパスワードに含めた乱数値が「127」であるので、これらの双方のデータが一致している場合のみ、その遊技者の遊技履歴を更新する。

【0434】

このようにすれば、他人が、遊技者Aの二次元コードを読み取ってサーバーコンピュータ200にアクセスしても、遊技者Aの遊技履歴や、当該他人の遊技履歴が更新されるこ

とはない。

なお、携帯通信端末100からサーバーコンピュータ200にアクセスがあったときは、その携帯通信端末100のSIMIDを読み取ることで、遊技者Aを特定する（遊技者Aからのアクセスであると判断する）。

【0435】

また、遊技者Aが、サーバーコンピュータ200から発行されたパスワードと異なるパスワード（以下、「別パスワード」という。）をスロットマシン10に入力して遊技を開始したとする。例えば、遊技者Aが他人の別パスワードを何らかの方法で入手し、スロットマシン10に入力した場合が挙げられる。

【0436】

ここで、別パスワードであっても、遊技者Aのパスワードと同様に、乱数値の情報が含まれている。たとえば、別パスワードの乱数値は「050」とであると仮定する。また、遊技者Aに対して発行された（最新の）パスワードに含まれる乱数値は、図40に示すように「127」とであると仮定する。

【0437】

そして、遊技者Aが遊技を終了し、スロットマシン10で二次元コードを発行し、サーバーコンピュータ200にアクセスしたとする。

ここで、二次元コードにはパスワードが入力されてからの遊技履歴とともに、入力されたパスワード（別パスワード）に含まれていた乱数値「050」の情報が含まれる。

【0438】

サーバーコンピュータ200は、遊技者Aの携帯通信端末100からアクセスがあると、その携帯通信端末100のSIMIDを読み取って、遊技者Aであると特定する。図40の例では、SIMIDは「123456」と読み取られるので、遊技者Aであると特定される。

さらに、このアクセス時には、乱数値「050」が読み取られる。しかし、サーバーコンピュータ200には、遊技者Aの携帯通信端末100に対して、最後に発行したパスワードに含めた乱数値「127」が記憶されている。

【0439】

サーバーコンピュータ200は、サーバーコンピュータ200内に記憶された乱数値「127」と、アクセス時の乱数値「050」とが一致するか否かを判断する。この例では、一致しないと判断される。これにより、サーバーコンピュータ200は、遊技者Aの遊技履歴の更新を拒否するように制御する。したがって、遊技者Aが、別パスワードを入手してスロットマシン10に入力し、それに基づいて二次元コードを発行してサーバーコンピュータ200にアクセスしても、遊技者Aの遊技履歴の更新は拒否されることとなる。

【0440】

ここで、本実施形態のように7ビット乱数を用いて「0」～「127」のいずれか1つの乱数値を決定するので、 $1/128$ の確率で、乱数値が偶然に一致する場合はあり得る。しかし、その一致率は約0.8%程度であるので、99%以上の確率で、別パスワードに基づくアクセスを防止することができる。

また、上記の例では7ビット乱数を用いたが、例えば16ビット乱数等を用いれば、偶然に一致する確率をさらに低くすることができる。

【0441】

図44は、サブ制御基板80のメモリ83に設けられているフラグを示す図である。なお、フラグは、図44に示したものに限定されるものではない。

リプレイ作動フラグ83aは、リプレイの入賞時に、メイン制御基板50から送信されてくる制御コマンドに基づいてオンとなるフラグである。リプレイの自動ベットによる遊技が終了したとき（再遊技の終了時）にオフとなる。

【0442】

ベット時演出フラグ83bは、サブCPU84側で選択される演出のうち、遊技者のベット操作を契機として演出内容を切り替える又は発展させる演出（特定演出）が選択され

10

20

30

40

50

たときにオンとなるフラグである。特定演出としては、たとえば、遊技開始時（役の抽選時）に演出を開始し、主人公キャラクターと敵キャラクターとが対決し、当該遊技の終了時（全リール31の停止時）に、主人公キャラクターが敗北する画像を表示する。次に、次遊技のためのベット時に、主人公キャラクターが再度、敵キャラクターと対決する画像を表示し、最終的に主人公キャラクターが勝利する演出を出力する場合がある。このような場合には、特定演出の選択時（遊技開始時）にベット時演出フラグ83bをオンとし、ベット時における演出（上記の逆転演出や、複数遊技継続すると期待値が高まるステップアップ演出等）の出力後、ベット時演出フラグ83bをオフにする。

【0443】

ベット音禁止フラグ83cは、たとえば連続演出（複数遊技にわたって出力する一連の演出）中にリプレイの入賞による自動ベット音を出力したくないとき等、ベット音を禁止するときにオンにするフラグである。

【0444】

たとえば連続演出が選択されたときは、連続演出を終了するまで、ベット音禁止フラグ83cをオンにすることが挙げられる。なお、連続演出中はベット音禁止フラグ83cを常にオンにするとは限らず、必要に応じて設定される。また、ベット音禁止フラグ83cがオンである場合において、リプレイの入賞に基づく自動ベット音のみを禁止にする場合が挙げられる。そして、メダルの手入れやベットスイッチ40の操作に基づく貯留メダルからのベット時に、ベット音を禁止するか否かは、種々設定することが可能である。

たとえば、連続演出中において、自動ベットや貯留ベットによるベット音は出力するが、手入れベット時はベット音を出力しない、といった設定でもよい。

【0445】

払出し中表示フラグ83dは、貯留メダルを有する場合において、精算スイッチ46が操作されたことに基づき、精算処理（メダルの払出し処理）を行っている間、「払出し中」である旨を画像表示し、精算音を出力するときにオンにされるフラグである。すなわち、メイン制御基板50側から精算開始コマンド（前述）を受信したときは、払出し中表示フラグ83dをオンにするとともに、「払出し中」を画像表示し、精算音を出力する。

そして、メイン制御基板50側から精算終了コマンド（前述）を受信したときは、払出し中表示フラグ83dをオフにするとともに、「払出し中」の画像表示及び精算音の出力を終了する。

【0446】

ベット有フラグ83eは、ベットメダルを有するときはオンとなり、ベットメダル無しのときはオフにされるフラグである。リプレイの入賞に基づき自動ベットされたときは、ベット有フラグ83eがオンとなる。リプレイの入賞によって自動ベットされたメダルは、精算（返却）が不可であるので、当該遊技を消化しないとベット有フラグ83eはオフにならない。なお、ベット有フラグ83eがオンである場合において、いつオフにするか（オフにするタイミング）は任意である。遊技の開始時でもよく、遊技の終了時でもよい。

【0447】

ベット有フラグ83eがオフの状態において、メダルのベット（手入れベット、貯留ベット、又はリプレイの入賞による自動ベット）を検出したときは、サブCPU84は、ベット有フラグ83eをオンにする。

ベット有フラグ83eがオンになったときは、

- 1) バックランプ、リール31の上部の蛍光灯等の照明装置のリセット
- 2) 後述するメニュー表示禁止フラグ83fをオン
- 3) サブCPU84側で制御される演出機器の表示リセット

等を行う。

ただし、ベット有フラグ83eがオンの状態では、メダルの手入れ貯留が行われても、上記1)～3)の処理は行わない。

【0448】

また、ベット有フラグ83eがオンの状態で、精算終了のコマンドを受信した場合において、リプレイの非作動時であればベット有フラグ83eをオフとするが、リプレイ作動時（自動ベットを有する状態）であれば、ベット有フラグ83eのオン状態を維持する。

ここで、ベット有フラグ83eがオフからオンに切り替わるタイミングで、演出を切り替えたり、表示をリセットする処理が設けられている。

【0449】

そして、たとえばリプレイ作動時に精算処理が行われ、再度、メダルの手入れ投入（貯留）が行われても、この間は、ベット有フラグ83eのオン状態が維持されている（ベット有フラグ83eが一旦オフとなり、再度、オンになることはない）ので、演出の切り替えや表示のリセット処理が二重に行われることはない。

10

【0450】

メニュー表示禁止フラグ83fは、図41～図43で示したメニュー画面を表示禁止に設定しているときにオンにするフラグである。メニュー画面の表示を禁止する場合としては、たとえばリプレイ作動時の自動ベット有りの場合、連続演出中等のマイスロ遊技の遊技履歴に影響を与える場合が挙げられる。メニュー画面の表示を禁止するというのは、二次元コードの出力を禁止することを意味する。したがって、本実施形態の場合、パスワードを入力して遊技を行っていないとき（マイスロ遊技中でないとき）には、メニュー表示禁止フラグ83fはオフのままでよい。あるいは、リプレイ作動に基づく自動ベット有りのときにだけオンにしてもよい。

【0451】

次に、図44で示した各フラグに関する処理の流れを、フローチャートに基づき説明する。図45は、ベットコマンドを受信したときの処理を示すフローチャートである。

20

サブCPU84は、メイン制御基板50から送信されたベットに係る制御コマンドを受信すると、ステップS701で、メダルベット時の初期化を行う。メダルベット時の初期化としては、以下の処理が挙げられる（ここで例示した処理に限定されるものではない）。

1) ベット有フラグ83eをオン

2) 照明装置のリセット（リール31の内側に設けられたバックランプ、リール上部に設けられた蛍光灯、その他のランプの点灯初期化等）

3) メニュー表示禁止フラグ83fをオン

30

4) 画像表示装置23のリセット（特に、遊技待機中（デモ）画面となっているとき）

【0452】

次にステップS702に進み、受信したベットに係る制御コマンドが、次遊技のための初投入であるか否かを判断する。ここで、「初投入」とは、たとえばリプレイの入賞に基づく自動ベットであるときは、最初の自動ベットに係る制御コマンドの受信のときに相当する。また、手入れによるベットであるときは、ベットなしの状態での最初のメダル投入に相当する。したがって、当該遊技のベット限界枚数が3枚であるときは、最初の1枚目の投入が「初投入」に相当し、2枚目や3枚目は初投入に相当しない。

次遊技の初投入であると判断されたときはステップS703に進み、初投入でないと判断されたときはステップS707に進む。なお、一度「初投入」が行われた後、精算処理でベットが解除され、その後、再度メダルが投入（ベット）されたときは、その投入は「初投入」に該当しない。

40

【0453】

ステップS703では、ベット時演出がセットされているか否かを判断する。ここでの判断は、ベット時演出フラグ83bのオン／オフを判断することにより行う。そして、ベット時演出有りと判断されたときはステップS704に進み、ベット時演出なしと判断されたときはステップS707に進む。

ステップS704では、リプレイ作動時であるか否かを判断する。ここでの判断は、リプレイ作動フラグ83aのオン／オフをハンダすることにより行う。そして、リプレイ作動時であると判断されたときはステップS705に進み、リプレイ作動時でないと判断さ

50

れたときはステップ S 7 0 6 に進む。

【0454】

ステップ S 7 0 5 では、サブ C P U 8 4 は、演出の出力契機（出力時期）の切り替えを行う。ステップ S 7 0 3 においてベット時演出有りと判断された場合であっても、リプレイ作動時には、ステップ S 7 0 5 により、ベット時に出力する演出を、本実施形態ではスタートスイッチ 4 1 の操作時に出力するように切り替える。すなわち、ベット時にはこの演出を出力しない。そしてステップ S 7 0 7 に進む。

【0455】

ここで、演出の出力タイミングを切り替える方法としては、たとえば演出番号を切り替える方法が挙げられる。たとえば、本実施形態の例では、ベット時に逆転演出 X を出力する演出が選択されているとしたとき（演出番号「11」と仮定する）、スタートスイッチ 4 1 の操作時に逆転演出を出力する演出（たとえば、演出番号「15」）を選択し直すことである。ここで、演出番号「15」においてスタートスイッチ 4 1 の操作時に出力する演出は、上記の逆転演出 X と同一であってもよく、あるいは逆転演出 X と異なる内容の逆転演出 Y、又は逆転演出 X と類似する内容の逆転演出 X' 等、いかなるものでもよい。

これに対し、リプレイ作動時でないと判断されてステップ S 7 0 6 に進んだときは、ベット時の演出を出力する。そしてステップ S 7 0 7 に進む。

【0456】

ステップ S 7 0 7 では、ベット音禁止フラグ 8 3 c のオン／オフを判断する。なお、ベット音禁止フラグ 8 3 c は、たとえば前遊技で連続演出が選択された場合等、前遊技で既にオンにされている。

ベット音禁止フラグがオンであると判断したときはステップ S 7 0 8 に進み、ベット時のメダル投入音を出力しないように制御する。これに対し、ベット音禁止フラグ 8 3 c がオフであるときはステップ S 7 0 9 に進み、ベット時のメダル投入音を出力するように制御する。

なお、ベット音は、リプレイ作動時の自動ベット、手入れベット、ベットスイッチ 4 0 の操作に基づく貯留ベットで、同一音とするか異なる音とするかは、任意に設定することができる。

【0457】

図 4 6 及び図 4 7 は、メダル精算時の流れを示すフローチャートである。図 4 7 は、図 4 6 に続くフローチャートである。

図 4 6 のステップ S 7 2 1 では、サブ C P U 8 4 は、精算開始に係る制御コマンドを受信したか否かを判断する。精算開始に係る制御コマンドを受信したと判断したときはステップ S 7 2 2 に進み、払出し中表示フラグ 8 3 d をオンにする。さらに、次のステップ S 7 2 3 では、メニュー表示禁止フラグ 8 3 f をオンにし、メニュー画面を出力不可に設定する。精算処理中は、メニュー画面を表示させないためである。

【0458】

次のステップ S 7 2 4 では、リプレイ非作動であるか否かを判断する。リプレイ非作動であると判断したときはステップ S 7 2 6 に進み、リプレイ非作動でないと判断したときはステップ S 7 2 5 に進む。

ステップ S 7 2 6 では、サブ C P U 8 4 は、ベット有フラグ 8 3 e をオフにする。次のステップ S 7 2 7 では、サブ C P U 8 4 は、通常精算音を出力する。そしてステップ S 7 2 8 に進む。

これに対し、ステップ S 7 2 4 でリプレイ非作動でないと判断されてステップ S 7 2 5 に進むと、サブ C P U 8 4 は、特殊精算音を出力する。そしてステップ S 7 2 8 に進む。

【0459】

ここで、ベット有フラグ 8 3 e がオンのときとオフのときとで、精算処理中であることを示す画像表示を異ならせるように制御してもよい。たとえばベット有フラグ 8 3 e がオフのときは「精算中」とだけ画像表示し、ベット有フラグ 8 3 e がオンのときは「精算中ーリプレイの入賞によるメダルが残っています」等の画像表示を行うことが挙げられる。

【0460】

また、ステップS727における通常精算音は、たとえば「ブルルル・・・」といったごく標準の音を出力するが、ステップS725における特殊精算音は、たとえば「ピーピー・・・」といった特殊の音（警告音そのものではないが、警告音に類似するような音）を出力することが挙げられる。

【0461】

次のステップS728では、サブCPU84は、精算終了に係る制御コマンドを受信したか否かを判断し続ける。受信したと判断したときはステップS729に進み、払出し中表示フラグ83dをオフにする。次のステップS730では、ステップS725又はステップS727で出力を開始した精算音を停止する。

10

【0462】

次にステップS731に進み、マイスロ制御手段92は、マイスロ遊技中であるか否かを判断する。マイスロ制御手段92は、たとえば遊技者のパスワードが記憶されているか否かを判断することで、マイスロ遊技中であるか否かを判断する。マイスロ遊技中であると判断したときはステップS732に進み、マイスロ遊技中でないと判断したときはステップS737（図47）に進む。

【0463】

ステップS732では、マイスロ制御手段92は、ベット有フラグ83eがオンであるか否かを判断する。なお、この時点では、精算処理が終了しているので、ベット有フラグ83eがオンであるときは、自動ベットを有するとき、すなわちリプレイ作動時であることを意味する。ベット有フラグ83eがオンであると判断されるとステップS733に進み、マイスロ制御手段92は、二次元コードの表示を促進する演出1を画像表示（及び音声出力）する。この演出1としては、たとえば「リプレイ遊技を消化した後、二次元コードを表示してください」等のコメントを画像表示することが挙げられる。そして、本フローチャートによる処理を終了する。

20

【0464】

ステップS732においてベット有フラグ83eがオフであると判断され、ステップS734に進むと、マイスロ制御手段92は、連続演出中であるか否かを判断する。ここでの連続演出は、マイスロ遊技の遊技履歴に影響を与える連続演出であるものとする。そして、連続演出中でないと判断したときはステップS736に進み、連続演出中であると判断したときはステップS735に進む。

30

【0465】

ステップS735では、マイスロ制御手段92は、二次元コードの表示を促進する演出2を画像表示（及び音声出力）する。演出2としては、たとえば「連続演出を消化した後、二次元コードを表示してください」等のコメントを画像表示することが挙げられる。そして、本フローチャートによる処理を終了する。

【0466】

一方、ステップS734からステップS736に進むと、マイスロ制御手段92は、二次元コードの表示を促進する演出3を画像表示（及び音声出力）する。演出3としては、たとえば「二次元コードを忘れずに表示してください」等のコメントを画像表示することが挙げられる。そしてステップS740に進む。

40

【0467】

また、ステップS731においてマイスロ遊技中でないと判断され、ステップS737に進むと、サブCPU84は、ベット有フラグ83eがオンであるか否かを判断する。ベット有フラグ83eがオンであると判断したとき（すなわち、リプレイの入賞による自動ベットを有するとき）はステップS739に進み、オンでないと判断したときはステップS738に進む。

ステップS739では、特殊精算終了演出を出力する。この特殊精算終了演出は、たとえばたとえば「あと1プレイできます」等の画像表示（及び音声出力）である。そして、本フローチャートによる処理を終了する。

50

【0468】

一方、ステップS738では、通常精算終了演出を出力する。この通常精算終了演出は、たとえば、次の来店を誘致する演出であり、「またのお越しをお待ちしています」等の画像表示（及び音声出力）である。そして、ステップS740に進む。

ステップS738又はステップS736からステップS740に進むと、サブCPU84は、精算終了後の待機画面に移行する準備として、バックランプや、リール上部の蛍光灯を消灯する。次のステップS741では、メニュー表示禁止フラグ83fをオフにする。これにより、メニュー画面の表示が許可される。

【0469】

次にステップS742に進み、サブCPU84は、タイマーのカウント値を「0」にする。このタイマーは、前遊技の終了時からカウントダウンしているタイマーであり（ただし、連続演出中を除く）、「0」となったときは待機画面への移行を許可する。ステップS742では、ベット有フラグ83eがオフかつ連続演出中でないので、精算処理終了後に速やかに待機画面に移行させるため、タイマー値を「0」にする処理を行う。次にステップS743に進み、サブCPU84は、待機画面の表示に移行するように制御する。そして本フローチャートによる処理を終了する。

【0470】

以上の処理により、リプレイの作動時、すなわち自動ベットを有するときは、精算処理が終了しても、バックランプや蛍光灯の消灯処理を行わず、待機画面の表示（遊技待機状態）に移行しないように制御する。

また、精算開始のコマンドを受信した場合において、リプレイ非作動時は、ステップS726においてベット有フラグ83eがオフにされるが、リプレイ作動時は、ベット有フラグ83eは、オンのままとなる。

なお、ステップS726の処理（ベット有フラグ83eのオフ）は、精算開始コマンドを受信したときに行ったが、精算開始コマンドの受信時には行わずに精算終了コマンドを受信したときに行ってもよい。あるいは、精算開始コマンドの受信時及び精算終了コマンドの受信時の双方で行ってもよい。

【0471】

さらにまた、図46及び図47の例では、リプレイの入賞に基づく自動ベットを有するときは、ステップS741の処理（メニュー表示禁止フラグ83fのオフ）を行わないので、メニュー画面を表示させることができないように設定している。

【0472】

さらに、精算処理が行われた場合に、マイスロ遊技中かつ連続演出中（遊技履歴に影響を与える演出中）であるときも、ステップS741の処理（メニュー表示禁止フラグ83fのオフ）を行わないので、メニュー画面を表示させることができない。

なお、マイスロ遊技中かつ連続演出中であっても、遊技履歴に影響を与えない連続演出中であるときは、ステップS734の判断で「No」となるので、ステップS741の処理（メニュー表示禁止フラグ83fのオフ）を経由し、メニュー画面を表示させることが可能となる。

【0473】

図48は、マイスロ遊技の終了時における二次元コードのバックアップ処理を示すフローチャートである。

まず、ステップS751では、サブCPU84は、エンターキー85bがオンされたか否かを判断し続ける。エンターキー85bがオンされたと判断されたときはステップS752に進み、サブCPU84は、遊技開始待ち状態であるか否かを判断する。遊技開始待ち状態であると判断したときは、メニュー画面の表示が可能であるためステップS753に進む。これに対し、遊技開始待ち状態でないとき、たとえば遊技中（リールの回転中）や精算処理中等にはメニュー画面の表示に移行しない。

【0474】

ステップS753では、サブCPU84は、メニュー画面の表示を実行するように制御

する。メニュー画面は、図 4 1 等で示した内容である。なお、メニュー画面の表示は、メニュー表示禁止フラグ 8 3 f がオフとなっていることが前提である。

【0 4 7 5】

次のステップ S 7 5 4 では、メニュー画面中、「二次元コード表示」が選択されるのを待ち、次のステップ S 7 5 5 では、「二次元コード」の位置でエンターキー 8 5 b がオンされたか否かを判断する。オンされたと判断したときはステップ S 7 5 6 に進み、オンされていないと判断したときはステップ S 7 5 3 に戻る。なお、ステップ S 7 5 5 でエンターキー 8 5 b がオンされると、その時点で遊技履歴の更新をストップする。

【0 4 7 6】

次のステップ S 7 5 6 では、マイスロ制御手段 9 2 は、二次元コードを画像表示するためのバックアップデータを有するか否かを判断する。バックアップデータを有しないと判断したときはステップ S 7 5 7 に進み、バックアップデータを有すると判断したときはステップ S 7 5 9 に移行する。

ステップ S 7 5 7 では、マイスロ制御手段 9 2 は、上述した遊技者の I D（すなわち S I M I D）、乱数値、それまでの遊技履歴をメモリ 8 3 内から読み取り、二次元コードを所定のプログラム（図示せず）によって生成する。

【0 4 7 7】

次のステップ S 7 5 8 では、メモリ 8 3 内に記憶していたその遊技者の遊技履歴を削除（消去）する。また、本フローチャートでは図示しないが、たとえばマイスロ遊技において演出をカスタマイズする機能（その遊技者の好みに応じて登場キャラクタの服を変更すること等）を有するときは、そのカスタマイズ演出をキャンセルし、初期（デフォルト）状態に設定する。そしてステップ S 7 6 1 に進み、生成した二次元コードを画像表示する。この状態は、図 4 3 の「二次元コード表示」で示す状態である。

【0 4 7 8】

一方、ステップ S 7 5 6 からステップ S 7 5 9 に進むと、メモリ 8 3 内に記憶していたバックアップデータ（二次元コードを画像表示するためのデータ）を読み出す。次にステップ S 7 6 0 に進み、ステップ S 7 5 9 で読み出したバックアップデータに基づいて二次元コードを再生する。そしてステップ S 7 6 1 に進んで、再生した二次元コードを画像表示する。

【0 4 7 9】

ステップ S 7 6 1 からステップ S 7 6 2 に進むと、二次元コードの画像表示を終了する指示コマンド（制御コマンド）を受信したか否かを判断する。この指示コマンドは、本実施形態では、メダルの手入れに係るコマンドに設定されている。メダルの手入れは、遊技者が自らの意思でその操作を行ったと考えられ、遊技者が、自らの意思で、二次元コードを画像表示した（遊技履歴更新をストップさせた）後、再度、メダルを手入れして遊技を再開することは考えにくいからである。このような指示コマンドを受信したときはステップ S 7 6 3 に進む。

【0 4 8 0】

また、二次元コードの画像表示を終了する指示コマンドの他の例としては、遊技者の意思による所定の操作に基づくコマンドが挙げられる。たとえば、二次元コードの画像表示中のエンターキー 8 5 b の操作、マイスロ遊技開始時のパスワードの入力操作、所定のスイッチ（操作スイッチ又はメニューキー 8 5）の長押し、複数のスイッチの同時押し等が挙げられる。このような所定の操作を事前に設定しておき、その所定の操作が行われたときは、二次元コードの画像表示を終了する指示コマンドを送信して、二次元コードの画像表示を終了させる。この所定の操作は、遊技者が自らの意思で二次元コードの画像表示を終了するときに遊技者に操作させる。

【0 4 8 1】

次のステップ S 7 6 3 では、二次元コードのバックアップデータを有するか否かを判断する。バックアップデータを有すると判断したときはステップ S 7 6 4 に進み、そのバックアップデータを消去し、ステップ S 7 6 5 に進む。したがって、指示コマンドの受信は

10

20

30

40

50

、二次元コードのバックアップデータの消去契機となる。一方、ステップ S 7 6 3 でバックアップデータを有しないと判断したときはステップ S 7 6 5 に進む。

ステップ S 7 6 5 では、二次元コードの画像表示画面をクリアし、通常画面に移行する。そして本フローチャートによる処理を終了する。

【0482】

これに対し、ステップ S 7 6 2 において指示コマンドを受信していないと判断したときはステップ S 7 6 6 に進む。ステップ S 7 6 6 では、制御コマンドのうち、出力を切り替えることとなるコマンド（出力切替コマンド）を受信したか否かを判断する。出力切替コマンドを受信していないと判断したときは、ステップ S 7 6 1 に戻って二次元コードの画像表示状態を継続し、出力切替コマンドを受信したと判断したときはステップ S 7 6 7 に進む。

10

ここで、出力切替コマンドとしては、たとえばベットスイッチ 4 0 の操作、スタートスイッチ 4 1 の操作、精算スイッチ 4 6 の操作、エラー検知、及び電断信号の受信が挙げられる。

【0483】

なお、本実施形態では、貯留メダルを有するときであっても、遊技待ち状態であれば、エンターキー 8 5 b の操作によりメニュー画面を表示し、二次元コードを画像表示可能である。そして、二次元コードが画像表示された状態で、ベットスイッチ 4 0 が操作されると、規定枚数がベットされるとともに、二次元コードの画像表示を終了し、通常画面（遊技開始画面）に切り替えるように設定されている。

20

【0484】

また、本実施形態では、メダルがベットされている状態であっても、遊技開始待ち状態でエンターキー 8 5 b がオンされれば、メニュー画面を表示し、二次元コードを画像表示可能である。そして、二次元コードが画像表示された状態でスタートスイッチ 4 1 が操作されると、遊技を開始する（リール 3 1 の回転を開始する）とともに、遊技中の演出画面に切り替えるように設定されている。

【0485】

以上より、二次元コードを画像表示している状態で、ベットスイッチ 4 0 やスタートスイッチ 4 1 が操作されると、二次元コードの画像表示状態は終了する。

しかし、二次元コードを画像表示している途中で、遊技者が自分の携帯通信端末 1 0 0 を取り出す際等に、誤ってベットスイッチ 4 0 やスタートスイッチ 4 1 を操作してしまう（これらのスイッチに接触してしまう）可能性がある。そこで、本実施形態では、そのような誤操作（遊技者の意思に反する操作）により、携帯通信端末 1 0 0 にて二次元コードを読み取る前に、二次元コードが消去されかつ再生不可能になってしまうことを防止するため、二次元コードデータのバックアップをとる。

30

【0486】

精算スイッチ 4 6 の操作により精算処理が実行されると、精算処理の終了後は、自動ベットを有さない限り、待機画面に移行する。このため、精算処理が実行されると、二次元コードの画像表示を終了する。しかし、二次元コードの画像表示中に、遊技者が誤って精算スイッチ 4 6 を操作してしまう（精算スイッチ 4 6 に接触してしまう）可能性があるの

40

で、二次元コードの画像表示中に精算スイッチ 4 6 が操作されたときも、二次元コードデータのバックアップをとる。

【0487】

また、出力切替コマンドを送信させる契機となるエラーとしては、たとえばメダルセレクトエラー（図 2 6 のステップ S 2 3 9 等）、ホッパーエラー（精算時の払出しエラー）、フロントドアオープン時のエラー等が挙げられる。二次元コードの画像表示中にこのようなエラーが発生したときは、二次元コードの画像表示よりも優先して、エラーが発生したことを画像表示（報知）するため、優先順位がエラー報知よりも低い二次元コードの画像表示は強制終了してしまう。このような場合においても、二次元コードデータのバックアップをとる。

50

【0488】

二次元コードの画像表示中に、上述のような出力切替コマンドを受信したときは、ステップS767に進み、マイスロ制御手段92は、バックアップデータを有するか否かを判断する。バックアップデータを有しないと判断したときはステップS768に進み、画像表示中の二次元コードデータをバックアップする（メモリ83内の所定の記憶領域に記憶する）。そしてステップS769に進み、サブCPU84は、画面表示の切替えを行う。この切替えにより、二次元コードの画像表示を終了する。そしてステップS751に戻る。一方、ステップS767において既にバックアップデータを有すると判断したときはステップS769に進む。

【0489】

これにより、二次元コードのデータがバックアップされた後、ステップS751に進んで、遊技開始待ち状態でエンターキー85bをオンすれば、メニュー画面を経由して、ステップS759～ステップS761の処理により、再度、二次元コードを画像表示することができる。

さらに、二次元コードを誤って消去してしまうことが複数回続いても、ステップS762、ステップS766、ステップS767、ステップS769の処理を繰り返すことになるので、何度でも、二次元コードを画像表示することが可能となる。

【0490】

なお、エラー検知による二次元コードの画像表示終了の場合には、エラーからの復帰後、二次元コードの画像表示状態に自動復帰するようにしてもよい。また、フロントドアオープンのようなエラーにおいては、エラー表示よりも優先して二次元コードの画像表示を継続してもよい（ただし、ゴト対策として、フロントドアオープン時の音は出力する）。

さらにまた、二次元コードのデータのバックアップ後、再度、メニュー画面を表示する場合には、「二次元コード表示」という表示に変えて、「二次元コード再表示」としてもよい。

【0491】

図49は、メニュー表示禁止フラグ83fの処理を示すフローチャートである。また、図49の例では、エンターキー85bは、その裏面側に設けられたランプにより点灯／消灯可能に形成されたボタンであり、エンターキー85bが操作可能状態であるときは点灯し、操作不可能状態であるときは消灯するものとする。

【0492】

まず、ステップS781では、サブCPU84は、リプレイ作動時であるか否かを判断する。リプレイ作動時であると判断したときはステップS782に進み、リプレイ作動時でないと判断したときは本フローチャートによる処理を終了する。

ステップS782では、サブCPU84は、遊技者の操作スイッチ及び／又はメニューキー85による操作で、解除操作が行われたか否かを判断する。ここで、解除操作は、予めメモリ83に登録されており、その登録内容と、操作された内容とが一致するか否かを判断する。解除操作が行われたと判断したときはステップS784に進み、解除操作が行われていないと判断したときはステップS783に進む。

【0493】

ステップS784に進むと、メニュー表示禁止フラグ83fをオフにするとともに、メニューキー85を点灯させる。これにより、遊技者は、操作スイッチ及び／又はメニューキー85により解除操作を行えば、手動でメニュー表示禁止フラグ83fをオフにして（すなわち、メニュー表示禁止を解除して）、図43に示したようにメニュー画面を表示可能となる。

【0494】

ここで、「解除操作」としては、種々設定することが可能であるが、具体例を挙げれば、たとえば以下の通りである。

1) エンターキー85bを長押し（3秒以上）する等、いずれか1つのスイッチを長押しすること

10

20

30

40

50

2) 3ベットスイッチ40bとエンターキー85bとを同時押しする等、通常では生じ得ない複数のスイッチを同時押しすること

3) ホール店員による所定の操作(たとえばフロントドアオープン後、リセットボタンをオンする等の操作)を行うこと、又はその操作後に遊技者による所定の操作(たとえばエンターキー85bのオン)を行うこと

4) 上記1)~3)の組合せ

が挙げられる。

【0495】

ステップS782において解除操作が行われていないと判断され、ステップS783に進むと、サブCPU84は、精算終了コマンドを受信したか否かを判断する。この判断は、図46のステップS728と同様である。精算終了コマンドを受信したと判断したときはステップS784に進んでメニュー画面の表示を許可するように制御する。これに対し、精算終了コマンドを受信していないと判断したときは本フローチャートによる処理を終了する。

【0496】

以上より、リプレイ作動時、すなわちリプレイの入賞による自動ベットを有するままで、精算スイッチ46が操作され、精算処理が終了したときは、メニュー画面の表示を許可するようにする。これにより、たとえばホールが閉店時間を迎えた時点でART遊技等が継続しており、リプレイが入賞して自動ベットを有するときでも、メダルを精算し、かつメニュー画面を表示させ、マイスロ遊技を終了させる(二次元コードを読み取る)ことができる。

【0497】

なお、上述した図46及び図47の例では、リプレイの入賞による自動ベットを有するままで精算処理が終了したときは、メニュー表示禁止フラグ83fをオフにしていない。

したがって、図46及び図47の例は、自動ベットを有するままで精算処理が終了しただけではメニュー画面の表示を禁止状態にしておく例である。一方、図48の例は、それとは逆に、メニュー画面の表示を許可する例である。

【0498】

以上、本発明の一実施形態について説明したが、本発明は、上記実施形態に限定されるものではなく、たとえば、以下のような種々の変形が可能である。

A. メイン制御基板50に関する変形例

(1) 本実施形態では、制御コマンドの送信時には、第1及び第2制御コマンドを送信するようにしたが、これに限らず、第1~第4制御コマンドを送信することも可能である。この場合、たとえば、送信する制御コマンドデータの6~7ビット目を、「0」、「1」、「10」、及び「11」の4種類に分けるとともに、

第1制御コマンドデータの6~7ビット目：0

第2制御コマンドデータの6~7ビット目：1

第3制御コマンドデータの6~7ビット目：10

第4制御コマンドデータの6~7ビット目：11

とする。これにより、第1~第4制御コマンドデータを区別してサブ制御基板80に送信することが可能となる。

【0499】

(2) 本実施形態では、リプレイ作動後の自動ベットを有する状態で、一律に、貯留メダルを精算可能とした。しかし、これに限らず、リプレイの入賞に基づく自動ベットを有するときに、入賞したリプレイの種類に応じて、精算可能な場合と精算不可能な場合とを設けてもよい。

たとえば、リプレイの種類として、ノーマル(通常)リプレイと、レア(特殊)リプレイ(たとえば、上述したベルリプレイを含む)とを設ける。そして、ノーマルリプレイの入賞時は、その自動ベットを有する状態で精算可能に設定し、レアリプレイの入賞時は、その自動ベットを有する状態では精算不可能に設定することが挙げられる。

【0500】

また、このように制御するときは、7E11番地の作動状態フラグとして、D0ビットをノーマルリプレイとし、D3ビットをレアリプレイに設定する。そして、リプレイが入賞したときに、ノーマルリプレイかレアリプレイかに応じて、D0ビット又はD3ビットのデータをオンにする。

一方、精算スイッチ46が操作されたときは、7E11番地の作動状態フラグを判別し、D0ビットがオンであるときは精算可能とし（精算処理に移行し）、D3ビットがオンであるときは精算処理に移行しないように制御することが挙げられる。

【0501】

(3) 本実施形態では説明を省略したが、リプレイの当選時に、ストップスイッチ42の押し順に応じて内部状態を移行させるスロットマシンが知られている。

たとえば第1に、複数種類のリプレイを重複当選させるとともに、リプレイの重複当選パターンを複数設ける。そして、一のリプレイの重複当選時に、ストップスイッチ42の押し順がたとえば中第一停止であるときは、内部状態(RT)を移行させる図柄の組合せを表示してRT移行させ、左又は右第一停止であるときは、内部状態を移行させない図柄の組合せを表示してRT移行させないことが挙げられる。

【0502】

第2に、リプレイの単独当選又は重複当選時に、ストップスイッチ42の押し順にかかわらず同一の図柄の組合せを停止表示するが、ストップスイッチ42の押し順に応じて内部状態を移行させる(RTは同一。たとえば、MB内部中の場合。)ことが挙げられる。

そして、内部状態が移行するときは、移行前の内部状態と移行後の内部状態とで、フリーズに関する制御（実行頻度等）を異ならせることが挙げられる。

【0503】

このような場合に、内部状態が移行しない押し順でストップスイッチ42が操作されたときに入賞したリプレイの自動ベットを有するときは、その自動ベットを有するままで精算可能に設定する。一方、内部状態が移行する押し順でストップスイッチ42が操作されたときに入賞したリプレイの自動ベットを有するときは、その自動ベットを有するままで精算不可能に設定することが挙げられる。

【0504】

上記のように制御する場合には、精算禁止フラグを設けておき、内部状態が移行可能となるリプレイの当選時に、内部状態が移行可能となる押し順でストップスイッチ42が操作されたときは精算禁止フラグをオンにし、そのリプレイの入賞に基づく再遊技を消化したときに精算禁止フラグをオフにする。

リプレイの入賞に基づく自動ベットを有する状態において精算スイッチ46が操作されたときは、精算禁止フラグを確認し、オンのときは精算処理に移行しないように制御する。

一方、内部状態が移行しない押し順でストップスイッチ42が操作されたときは精算禁止フラグをオフのオフのままとする。

【0505】

(4) また、リプレイ作動後の自動ベットを有する状態では、原則として、貯留メダルを精算不可能とし、所定の操作を行ったときに限り精算可能に設定してもよい。

この場合、精算許可フラグを設けておく。そして、リプレイの当選時に、たとえば左右中（順挟み）の押し順（ただし、ペナルティにならない押し順を設定する）で操作されたときは、精算許可フラグをオンにし、それ以外の押し順では、精算許可フラグはオフのままとする。

【0506】

そして、リプレイの入賞に基づく自動ベットを有するときに精算スイッチ46が操作されたときは、精算許可フラグのオン／オフを判断し、精算許可フラグがオンであるときは精算処理を開始する。

このように設定すれば、遊技者は、左右中の押し順でストップスイッチ42を操作すれ

10

20

30

40

50

ば、当該遊技の終了時に貯留メダルを精算することができる。たとえばリプレイの当選確率を高く設定したR T状態で遊技を行っているときに、遊技者が精算を希望するときは、迅速に精算を可能にすることができる。

【0507】

(5) さらにまた、R T状態等、遊技状態（内部状態）に応じて、自動ベットを有するときに精算処理を許可するか否かを判断してもよい。

たとえば、いわゆる通常遊技中（非A R T遊技中、非特別遊技中）であるときは、リプレイの入賞に基づく自動ベットを有するときに精算スイッチ46が操作されたときは、本実施形態と同様に、精算処理を許可する。

これに対し、たとえば本実施形態のM B遊技中等に、リプレイの入賞に基づく自動ベットを有するときに精算スイッチ46が操作されたときは、精算処理を不許可にすることが挙げられる。

【0508】

ここで、仮に、M B遊技中には精算処理を不許可に設定するときは、7 E 1 1番地の作動状態フラグにより精算可能な遊技状態であるか否かを判別すればよい。

これに対し、たとえばリプレイの当選確率の異なる複数の遊技状態（R T）を設けておき、R T移行する仕様のスロットマシンであるときに、リプレイ通常確率のR Tでは自動ベットを有するときに精算を許可し、リプレイ高確率のR Tでは自動ベットを有するときの精算を不許可に設定することが挙げられる。この場合には、リプレイ高確率のR T中にオンとなるR Tフラグを設けておき、精算スイッチ46が操作されたときは、このR Tフラグのオン／オフに基づき精算可能であるか否かを判別すればよい。

【0509】

(6) 実施形態では、精算スイッチ46が操作された場合において、ベットメダルを有するときは、ベットメダルを先に精算するように制御した。したがって、ベットメダルと貯留メダルの双方を有する場合において、双方とも精算するときは、精算スイッチ46を2回操作する必要がある。

しかし、これに限らず、1回の精算スイッチ46の操作で、ベットメダルと貯留メダルの双方を精算できるように制御してもよい（ただし、リプレイ非作動時に限る）。また、ベットメダルと貯留メダルの双方を精算する場合、いずれを先に精算してもよい。

【0510】

(7) 図20において、ステップS516の指定アドレスデータセット（S_ADDR_SET）では、図21のサブルーチンに移行し、ステップS531～ステップS533の処理を行っている。これは、メインCPU54（チップ）の性能に依存するものである。したがって、メインCPU54によっては、このようなサブルーチン処理を設けることに限らず、ステップS516の中で一括で処理を行うことも可能である。

すなわち、ステップS516において、メインCPU54の性能によっては、

- a) Aレジスタ値とLレジスタ値との合計値をAレジスタに記憶する処理
- b) Aレジスタ値をLレジスタに記憶する処理
- c) H Lレジスタ値で指定される番地のデータをAレジスタに記憶する処理
- d) Aレジスタ値をAND演算し、0かどうかを判断する処理

を一括で行うことが挙げられる。

【0511】

(8) 本実施形態では、精算スイッチ46に係る立ち上がりデータに基づいて精算処理を開始するように制御した。すなわち、精算スイッチ46が操作されることで精算処理を開始するようにした。

しかし、これに限らず、精算スイッチ46を一定時間操作し続けること（押しっぱなし）により精算処理を開始するようにしてもよい。

たとえば、精算スイッチ46に係るレベルデータが、500回の割込み（約1100ms）の間、連続でオンであるときに精算処理を開始することが挙げられる。

【0512】

10

20

30

40

50

ここで、上記実施形態（図 9 及び図 10）の例では、1 回目割込み時と 2 回目割込み時とで取得したデータが異なる場合には、今回割込み時のデータを「0」とした。しかし、これに限らず、1 回目割込み時と 2 回目割込み時とで取得したデータが異なる場合には、前回割込み時のデータを採用することも可能である。

【0513】

そして、上記のような精算スイッチ 46 の押しっぱなしを検出する場合には、後者を採用することが好ましい。精算スイッチ 46 が押しっぱなしとされた場合には、立ち上がりデータはずっと「1」のままとなるが、その途中でチャタリングやノイズ等によって「0」のデータを稀に単発で取得しても、その場合には前回割込み時のデータを採用することで、立ち上がりデータをずっと「1」にすることができる。これにより、精算スイッチ 46 の押しっぱなしを正しく検知することができる。

10

【0514】

（9）上記実施形態では、リプレイの入賞に基づく自動ベットを有するときであっても、メダルの手入れ貯留を可能とした。しかし、これに限らず、メダルの手入れ貯留ができる場合とできない場合とを設けてもよい。

たとえば、ノーマルリプレイの入賞に基づく自動ベット時は、メダルの手入れ貯留を不可に設定する。これに対し、ベルリプレイの入賞に基づく自動ベット時は、メダルの手入れ貯留を可能に設定する。

【0515】

ここで、図 11 中、7E11 番地の作動状態フラグとして、D0 ビットのノーマルリプレイに加え、D3 ビットをベルリプレイに設定する。そして、リプレイが入賞したときに、ノーマルリプレイかベルリプレイかに応じて、D0 ビット又は D3 ビットのデータをオンにする。

20

【0516】

また、メイン CPU 54 は、ノーマルリプレイの入賞に基づく自動ベットを有するときは、ブロック 45 をオフにし、メダル返却通路を形成する（メダルの手入れ貯留を不許可とする）。これに対し、メイン CPU 54 は、ベルリプレイの入賞に基づく自動ベットを有するときは、ブロック 45 をオンにし、メダル投入通路を形成する（メダルの手入れ貯留を許可する）。

【0517】

さらに、上述したように、ノーマルリプレイとレアリプレイとを有する場合において、レアリプレイの入賞時は、遊技状態、内部状態等に移行させる場合がある。このような場合においても、ノーマルリプレイの入賞による自動ベットを有するときはメダルの手入れ貯留を不許可とするが、レアリプレイの入賞による自動ベットを有するときはメダルの手入れ貯留を許可することが挙げられる。

30

【0518】

（10）本実施形態では、通常遊技中は 3 枚ベットとし、MB 遊技及び CB 遊技中は 2 枚ベットとした。しかし、これに限らず、全遊技状態で 3 枚ベットとしてもよい。この場合には、メダル限界枚数セット（MS_MMAX_SET）で、限界枚数を一律に 3 枚にセットする。また、MB 遊技中等が 3 枚ベットであっても、同一のモジュールを使用することにより、製品の開発段階で、MB 遊技中等のベット数を容易に変更することができる。さらに、製品ごとにベット数が異なる場合であってもベット数を容易に変更することができる。

40

【0519】

（11）本実施形態では、図 11 に示すように、制御コマンドバッファとして 7E16 ～ 7E35 の 32 個を設けたが、これに限るものではなく、制御コマンドバッファを何個設けるかは任意である。

（12）本実施形態では、2 回の割込みにおいて、同一の制御コマンドデータをサブ制御基板 80 に送信した。これは、上述したように、コマンドごけ等が生じてもサブ制御基板 80 に対して制御コマンドデータを確実に送信するためであるが、必ずしもこのように制御する必要はない。つまり、1 回の割込みごとに、それぞれ異なる制御コマンドデータ

50

を送信してもよい。

【0520】

(13) 図9の例では、1割込み処理において、入力ポートのデータを2回取得したが、必ずしもこれに限られるものではなく、1回のみとしてもよい。ただし、1割込みでデータの取得を2回行うことにより、チャタリングやノイズによる誤検知をより確実に防止することができる。

【0521】

(14) 本実施形態では、1BBやRB等の遊技状態移行役（当選を持ち越す、いわゆるボーナス役）を設けていないが、これらを設けることは、もちろん、可能である。

さらに、本実施形態では、MBの当選を持ち越して内部中遊技を作り出したが、これに限らず、1BBやRBを当選をさせて内部中遊技を作り出してもよい。

(15) 本実施形態では、サブ制御基板80側のAT制御手段91により、AT遊技を実行するか否かの抽選を行ったが、これに限らず、メイン制御基板50側でAT遊技を実行するか否かの抽選や、AT遊技中の制御を行ってもよい。

【0522】

B. サブ制御基板80に関する変形例

(1) 上記実施形態では説明を省略したが、メダルの投入（ベット又は貯留）が許可状態であるときに点灯するランプ（インサートランプ）を設けてもよい。そして、リプレイの入賞による自動ベットを有するときであっても、貯留枚数が限界枚数に到達していないときは、貯留の追加が可能であるので、インサートランプを点灯させることで、遊技者に対し、視覚的にメダルの手入れ貯留が可能であることを表示してもよい。

さらに、インサートランプを設けたときのインサートランプの制御は、メイン制御基板50又はサブ制御基板80のいずれで行ってもよい。

【0523】

(2) リプレイの入賞による自動ベットを有する場合におけるメダルの手入れ貯留時と、自動ベットではない（手入れベット又は貯留ベットである）場合におけるメダルの手入れ貯留時とで、メダルの投入（貯留）音を異ならせてもよい。これにより、手入れ貯留がいずれに該当するかを遊技者に感覚的に知らせることができる。また、投入音自体は同一であるが、音量のみを異ならせることも可能である。もちろん、音色及び音量の双方を異ならせるものでもよい。

【0524】

(3) さらに、リプレイの入賞による自動ベットを有する場合におけるメダルの手入れ貯留において、上述したノーマルリプレイの入賞による自動ベットを有するときは、メダル貯留時に通常の投入音を出力する。

一方、ベルリプレイやレアリプレイによる自動ベットを有する場合におけるメダルの手入れ貯留時には、前記通常の投入音と異なる音を出力してもよい。

あるいは、ベルリプレイやレアリプレイによる自動ベットを有する場合におけるメダルの手入れ貯留時には、自動ベットを有さない場合におけるメダルの手入れ貯留時と同一の投入音を出力してもよい。

【0525】

(4) 図45の実施形態において、ベット時に演出が切り替わる場合においてリプレイが入賞したときは、ベット時の演出切替えからスタートスイッチ41の操作時に演出を切り替えるようにした。

しかし、この演出の切り替えは、リプレイの種類によって異ならせることも可能である。上述したように、本実施形態では、ベルリプレイの入賞時はフリーズを実行し、所定時間（たとえば20秒）を経過する前にベットスイッチ40の操作又は手入れ貯留があったときは、その時点でフリーズを解除（キャンセル）し、自動ベットを行うようにした。

【0526】

そこで、ノーマルリプレイの入賞時には図45に示すフローチャートによる処理を実行するが、ベルリプレイの入賞時に、ベットスイッチ40の操作又は手入れ貯留があったと

きは、自動ベットを行うと同時にベット時演出を出力するようにしてもよい。

さらに、ベット時演出の出力を開始するタイミングとしては、ベットスイッチ 40 の操作を検知したとき、フリーズの解除（キャンセル）を検知したとき、（投入センサ 44 a 及び 44 b による）手入れメダルを検知したとき等が挙げられる。

【0527】

（5）図 48 の実施形態では、遊技者によるメダルの手入れがあったときは、二次元コードのバックアップデータを削除するようにした。しかし、これ以外に、バックアップデータの削除は、種々のタイミングで行うことが可能である。たとえば、次の遊技者がパスワードを入力したとき、入力が全くない状態で所定時間（たとえば 30 秒）を経過したとき等が挙げられる。

10

【0528】

（6）サブ制御基板 80 において、待機画面への移行禁止を示す待機画面移行禁止フラグを設けてもよい。そして、二次元コードの画像表示中は、待機画面移行禁止フラグをオンにする。ここで、上記実施形態では、二次元コードの画像表示中に精算スイッチ 46 が操作されると、精算処理に係るコマンドは出力切替えコマンドになるため、二次元コードの画像表示が終了してしまう。

これに対し、二次元コードの画像表示中は待機画面移行禁止フラグをオンにしておくことで、精算処理後に待機画面に移行することを禁止すれば、精算処理により二次元コードの画像表示が終了してしまうことを防止することができる。

【0529】

20

（7）上記実施形態では、二次元コードの画像表示中にエラーが発生したときは、一律に、二次元コードの画像表示よりも優先してエラーが発生したことを画像表示した。しかし、これに限らず、エラーの内容に応じて二次元コードの画像表示を終了するか否かを設定してもよい。たとえばメダルセレクトやメダル払出し装置のエラー等、重要度の高いエラー（ゴト行為に関係するエラー）の発生時は、エラー報知を優先するために二次元コードの画像表示を終了する。一方、フロントドアのオープンエラー等の重要度の低いエラーについては、たとえば警告音は出力するが、二次元コードの画像表示については終了せずに継続するように設定してもよい。

【0530】

（8）二次元コードの画像表示中に出力切替えコマンドを受信したために、二次元コードのデータをバックアップしたときは、画面表示を切り替えた後、二次元コードがバックアップされていることを遊技者に知らせるために、切替え後の画面にその旨を画像表示してもよい。たとえば画像表示装置 23 の画像表示領域中、右下等に画像表示することが挙げられる。

30

【0531】

（9）サブ制御基板 80 にメニュー起動フラグを設け、図 46 及び図 47 の実施形態において、精算処理の終了後、メニュー表示禁止フラグ 83 f がオンであるとき（たとえば、自動ベット有、連続演出中等）は、メニュー起動フラグをオンにしてもよい。そして、メニュー表示禁止フラグ 83 f がオンからオフになったとき、メニュー起動フラグを参照し、メニュー起動フラグがオンであるときは、「メニュー画面（二次元コード）を画像表示しますか？」等のメッセージを画像表示してもよい。

40

【0532】

（10）メニュー表示禁止フラグ 83 f がオンであるときは、画面中に、「メニュー画面（二次元コード）表示不可状態」と画像表示し、メニュー表示禁止フラグ 83 f がオフであるときは、「メニュー画面（二次元コード）表示可状態」と画像表示することで、遊技者に対し、メニュー表示禁止フラグ 83 f のオン／オフを報知してもよい。

【0533】

（11）図 49 の実施形態では、リプレイ作動時に、二次元コードを画像表示させるときの解除操作について例示した。しかし、これに限らず、たとえば連続演出中のときに、解除操作を行うことで、メニュー表示禁止フラグ 83 f をオフにできるようにしてもよい

50

。このようにすれば、連続演出中であるが、メダルがなくなったために遊技を終了したい遊技者が、所定の解除操作を行うことで、二次元コードを画像表示することができる。

【0534】

さらに、連続演出の途中で（連続演出を終了することなく）二次元コードを画像表示するときは、遊技履歴を、当該連続演出の開始前に戻して（当該連続演出を開始してからの遊技を含めずに）二次元コードを生成する場合と、当該連続演出中の遊技を含めて二次元コードを生成する場合とが挙げられる。

【0535】

（12）図49の実施形態では、リプレイ作動中に精算終了コマンドを受信したときは、メニュー表示禁止フラグ83fをオフにするようにした。しかし、これに限らず、リプレイ作動による自動ベットを有する場合において、貯留メダルが「0」であるとき（精算処理を行ったか否かにかかわらず）は、メニュー表示禁止フラグ83fをオフにするように設定してもよい。

【0536】

（13）図49中、ステップS783では、精算終了コマンドを受信したか否かを判断したが、この判断においては、たとえば精算スイッチ46の開始コマンドを受信したときにメニュー表示禁止フラグ83fを解除してもよく、あるいは、精算スイッチ46の終了コマンドを受信したときに解除してもよい。

【0537】

C. その他の変形例

（1）本実施形態では、サーバーコンピュータ200から発行されたパスワードを、遊技者が、スロットマシン10のメニューキー85を使用して入力するようにした。しかし、これに限らず、以下のような入力方法も可能である。

【0538】

例えば、スロットマシン10の前面側に、小型の画像読取り装置（カメラ、CCD、撮像装置）を設ける。そして、サーバーコンピュータ200から携帯通信端末100に発行するパスワードを二次元コードとし、携帯通信端末100のディスプレイにその二次元コードを画像表示する。それをスロットマシン10の上記画像読取り装置にかざすことで、二次元コード（パスワード）がスロットマシン10に入力されるようにする。

なお、二次元コードからなるパスワードをスロットマシン10に入力する際には、スロットマシン10の所定の操作スイッチ等を操作する。このようにすれば、パスワードを遊技者が手入力する必要がないので、遊技者の労力を軽減することができる。

【0539】

（2）また、たとえば遊技者がパスワード等を入力しなくても、遊技履歴を記憶するようにしてもよい。たとえば遊技の開始時に所定の操作を行うこと等により、それまでの遊技履歴をリセットする。そして、その遊技者の遊技開始時からの遊技履歴を記憶する。その遊技者が遊技を終了するときに、上記実施形態で示した二次元コード等を画像表示することで、その遊技者の遊技履歴を出力することも可能である。

【0540】

（3）本実施形態で示したフローチャートにおいて、処理の順序は、図で示した順序に限られるものではなく、処理の順序を入れ替え可能な場合には、処理の順序を入れ替えたものでもよい。

たとえば図13において、ステップS126における限界枚数の判断の前に、リプレイ作動時であるか否かを判断しているが（ステップS124）、ステップS126の限界枚数の判断の後にリプレイ作動時であるか否かを判断してもよい。

また、たとえば図46において、ステップS731のマイスロ遊技中であるか否かを判断する前に、ステップS732のベット有フラグ83eがオンであるか否かを判断することが挙げられる。

さらにまた、たとえば、図49において、ステップS781のリプレイ作動中であるか否かを判断する前に、ステップS782の解除操作が行われたか否かを判断すること等が

10

20

30

40

50

挙げられる。

【０５４１】

（４）本実施形態では、遊技機の一例としてスロットマシン１０を例に挙げたが、メイン制御基板５０（メイン制御手段）からサブ制御基板８０（サブ制御手段）への制御コマンドデータの送信方法に関しては、弾球遊技機や封入式遊技機等に適用することも可能である。

【０５４２】

D．本実施形態及び上記の各種の変形例は、単独で実施されることに限らず、適宜組み合わせることで実施することが可能である。

【符号の説明】

10

【０５４３】

１ 遊技システム

１０ スロットマシン（遊技機）

２１ ランプ

２２ スピーカ

２３ 画像表示装置

３１ リール

３２ モータ

３３ インデックスセンサ

３５ メダル払出し装置

20

３６ ホッパーモータ

３７ a、３７ b 払出しセンサ

３８ 満杯センサ

４０ a １ベットスイッチ

４０ b ３ベットスイッチ

４１ スタートスイッチ

４２ ストップスイッチ

４３ メダル投入口

４３ a 通路センサ

４４ a、４４ b 投入センサ

30

４５ ブロッカ

４６ 精算スイッチ

４７ a ベット枚数表示装置

４７ b 貯留枚数表示装置

５０ メイン制御基板

５１ 入力ポート

５２ 出力ポート

５３ メモリ

５４ メインCPU

６１ 役抽選手段

40

６２ リール制御手段

６３ 入賞判定手段

６４ 払出し手段

８０ サブ制御基板

８１ 入力ポート

８２ 出力ポート

８３ メモリ

８４ サブCPU

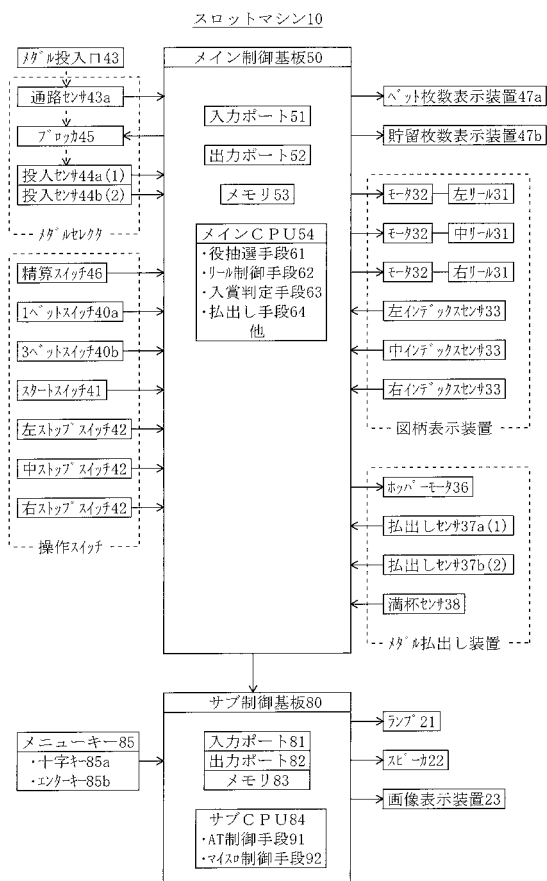
８５ メニューキー

８５ a 十字キー

50

85b エンターキー
 91 AT制御手段
 92 マイスロ制御手段
 100 携帯通信端末
 200 サーバコンピュータ

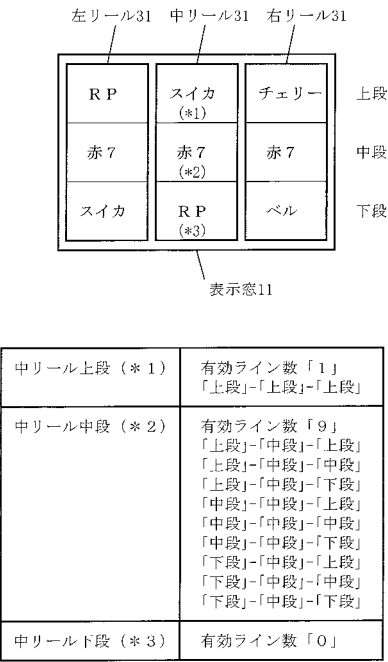
【図1】



【図2】

図柄番号	左リール31	中リール31	右リール31
21.	ベル	スイカ	スイカ
20.	RP	RP	ベル
19.	BAR	ベル	BAR
18.	ベル	BAR	RP
17.	RP	スイカ	チェリー
16.	赤7	赤7	赤7
15.	スイカ	RP	ベル
14.	ベル	ベル	RP
13.	RP	チェリー	スイカ
12.	赤7	RP	ベル
11.	スイカ	ベル	RP
10.	ベル	チェリー	スイカ
9.	RP	スイカ	ベル
8.	青7	RP	RP
7.	ベル	ベル	チェリー
6.	スイカ	チェリー	ベル
5.	ベル	スイカ	スイカ
4.	RP	RP	ベル
3.	スイカ	ベル	RP
2.	ベル	青7	チェリー
1.	チェリー	ベル	青7

【図 3】



【図 7】



【図 8】

番地	ビット	内容	データ
7E00	D 0	精算スイッチ信号	0 or 1
	D 1	1ベットスイッチ信号	0 or 1
	D 2	3ベットスイッチ信号	0 or 1
	D 3	スタートスイッチ信号	0 or 1
	D 4	(未使用)	0
	D 5	左ストップスイッチ信号	0 or 1
	D 6	中ストップスイッチ信号	0 or 1
7E01	D 7	右ストップスイッチ信号	0 or 1
	D 0	精算スイッチ信号	0 or 1
	D 1	1ベットスイッチ信号	0 or 1
	D 2	3ベットスイッチ信号	0 or 1
	D 3	スタートスイッチ信号	0 or 1
	D 4	(未使用)	0
	D 5	左ストップスイッチ信号	0 or 1
7E02	D 6	中ストップスイッチ信号	0 or 1
	D 7	右ストップスイッチ信号	0 or 1
	D 0	電断信号	0 or 1
	D 1	投入センサ1信号	0 or 1
	D 2	投入センサ2信号	0 or 1
	D 3	払出しセンサ1信号	0 or 1
	D 4	払出しセンサ2信号	0 or 1
7E03	D 5	満杯センサ信号	0 or 1
	D 6	(未使用)	0
	D 7	(未使用)	0
	D 0	(未使用)	0
	D 1	(未使用)	0
	D 2	(未使用)	0
	D 3	(未使用)	0
7E04	D 4	(未使用)	0
	D 5	(未使用)	0
	D 6	プロッカ信号	0 or 1
	D 7	ホッパモータ駆動信号	0 or 1
	D 0	スタートスイッチ受け状態	0 or 1
	D 1	(未使用)	0
	D 2	プロッカ信号状態	0 or 1
	D 3	リプレイLED	0 or 1
	D 4	打ち止めLED	0 or 1
	D 5	(未使用)	0
	D 6	設定変更可能フラグ	0 or 1
	D 7	メダル限界判定	0 or 1

【図 9】

3ベットスイッチがオンの状態で、
1ベットスイッチがオンされた場合
(正常データ取得時)

(A) Cレジスタ (今回割込み時における入力ポートの取得データ (1回目))

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	1	1	0

(B) Bレジスタ (前回割込み時に生成した入力ポートレベルデータ)

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	1	0	0

(C) Aレジスタ (今回割込み時における入力ポートの取得データ (2回目))

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	1	1	0

(D) 入力ポートレベルデータ (Cレジスタ AND Aレジスタ)

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	1	1	0

(E) 変化ビットデータ (Bレジスタ XOR (D))

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	0	1	0

(F) 入力ポート立ち上がりデータ ((D) AND (E))

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	0	1	0

【図 10】

3ベットスイッチがオンの状態で、
1ベットスイッチがオンされた場合
(異常データ取得時)

(A) Cレジスタ (今回割込み時における入力ポートの取得データ (1回目))

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	1	1	0

(B) Bレジスタ (前回割込み時に生成した入力ポートレベルデータ)

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	1	0	0

(C) Aレジスタ (今回割込み時における入力ポートの取得データ (2回目))

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	1	0	0

(D) 入力ポートレベルデータ (Cレジスタ AND Aレジスタ)

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	1	0	0

(E) 変化ビットデータ (Bレジスタ XOR (D))

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	0	0	0

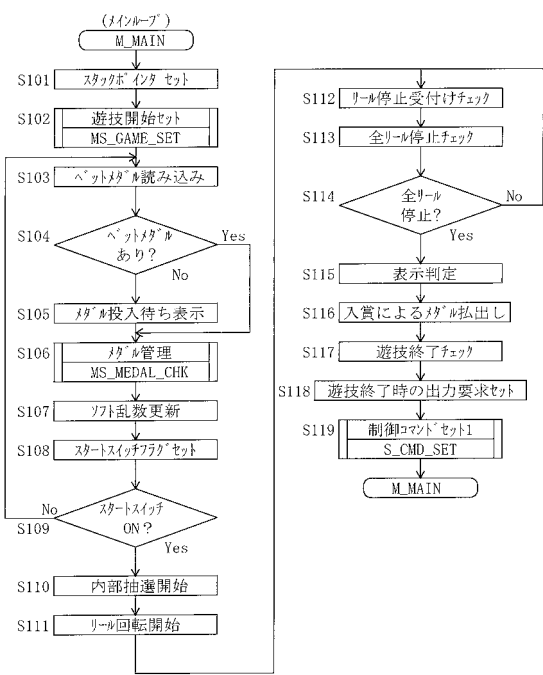
(F) 入力ポート立ち上がりデータ ((D) AND (E))

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
0	0	0	0	0	0	0	0

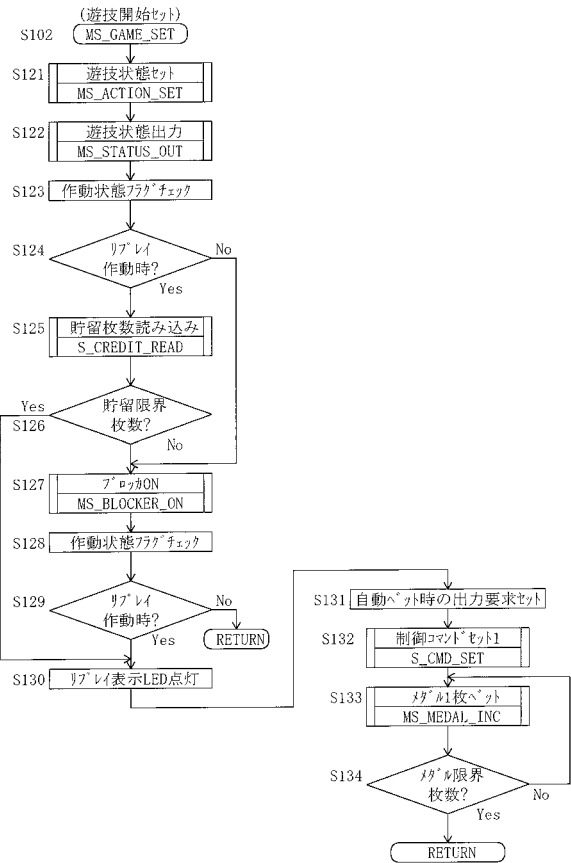
【図 1 1】

番地	ラベル	内容		データ
7E11	_FL_ACTION	作動状態フラグ		
		D0	リプレイ	0 or 1
		D1	MB	0 or 1
		D2	C B	0 or 1
		D3	未使用	0
		D4	未使用	0
		D5	未使用	0
		D6	未使用	0
		D7	未使用	0
7E12	_NB_REP_MEDAL	自動ベット数データ		3 or 2
7E13	_NB_PLAY_MEDAL	ベット数データ		3 or 2
7E14	_NB_CMDREAD	制御コマンド読み込みポインタ		0~FF (H)
7E15	_NB_CMDSTORE	制御コマンド書き込みポインタ		0~FF (H)
7E16	_BF_SUBCMD	+0	制御コマンドバッファ (第1制御コマンド0)	
7E17		+1	制御コマンドバッファ (第2制御コマンド0)	
7E18		+2	制御コマンドバッファ (第1制御コマンド1)	
7E19		+3	制御コマンドバッファ (第2制御コマンド1)	
}				
7E34		+30	制御コマンドバッファ (第1制御コマンド15)	
7E35		+31	制御コマンドバッファ (第2制御コマンド15)	

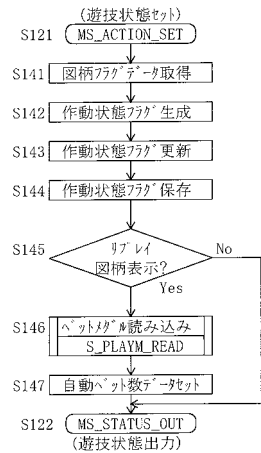
【図 1 2】



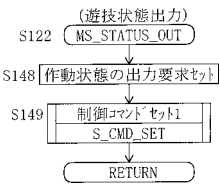
【図 1 3】



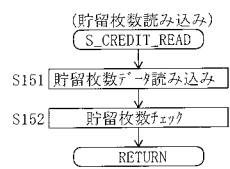
【図 1 4】



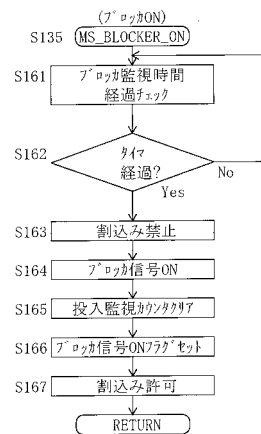
【図 1 5】



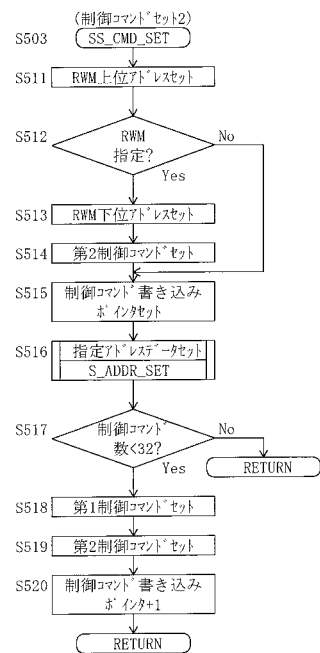
【 図 1 6 】



【 図 1 7 】



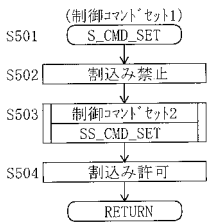
【 図 2 0 】



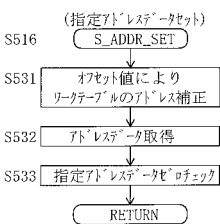
【 図 1 8 】

定義データ		
内容	Dレジスタ値(H(B))	Eレジスタ値(H(B))
メダル手入れ時の出力	10(00010000)	71(01110001)
貯留ベット時の出力	10(00010000)	1(00000001)
自動ベット時の出力	90(10010000)	12(00010010)
精算開始時の出力	F(00001111)	2(00000010)
精算終了時の出力	F(00001111)	3(00000011)

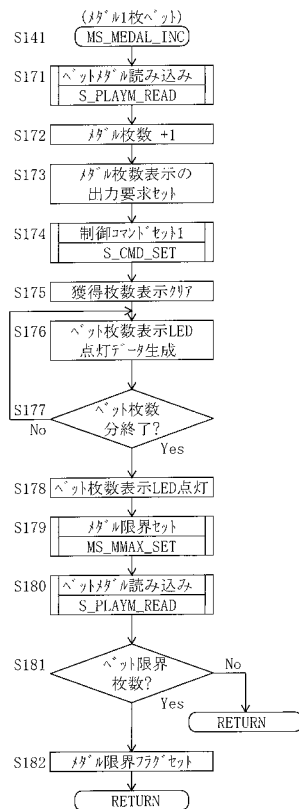
【 図 1 9 】



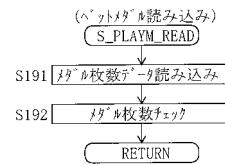
【 図 2 1 】



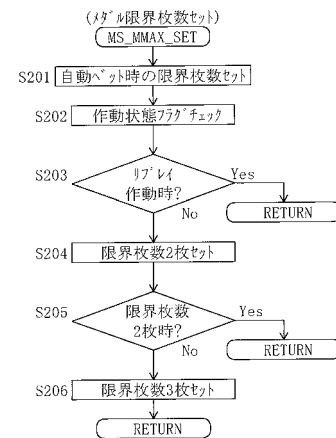
【図 22】



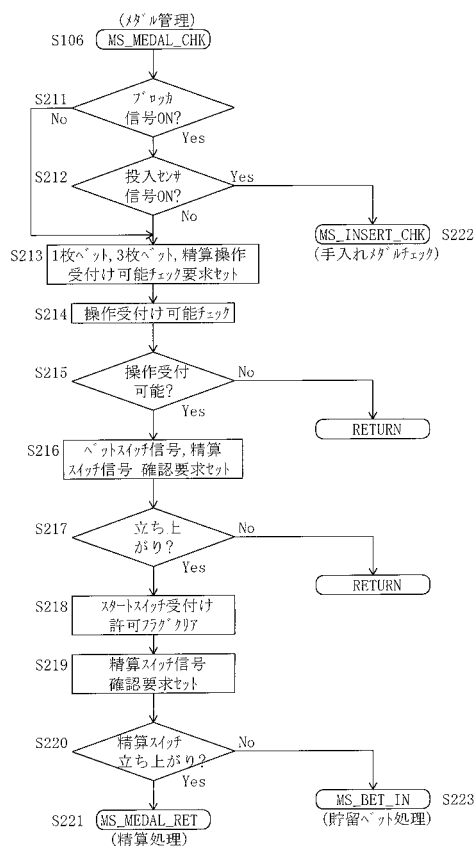
【図 23】



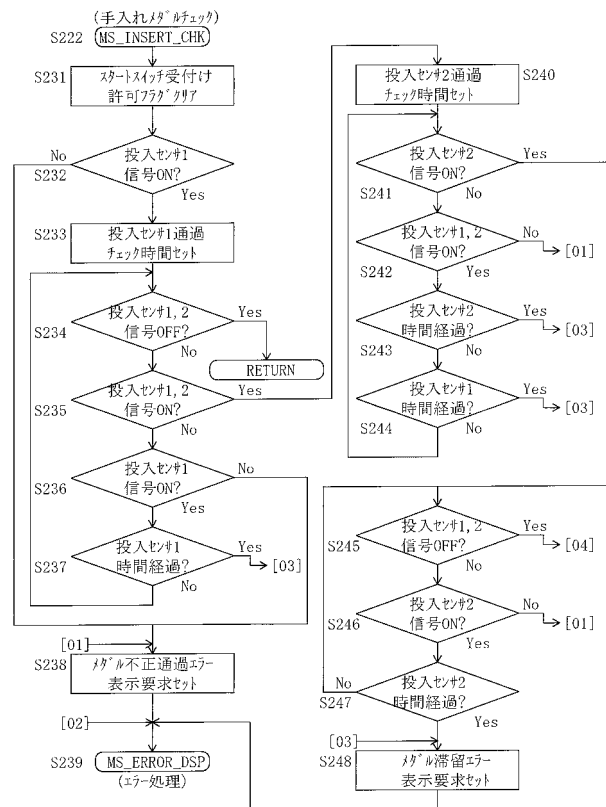
【図 24】



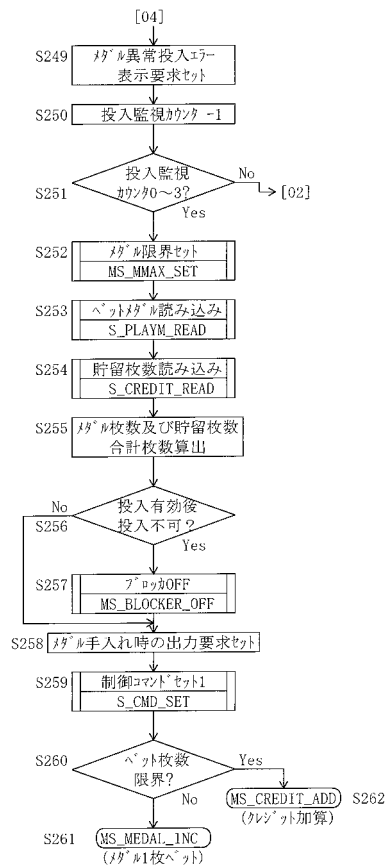
【図 25】



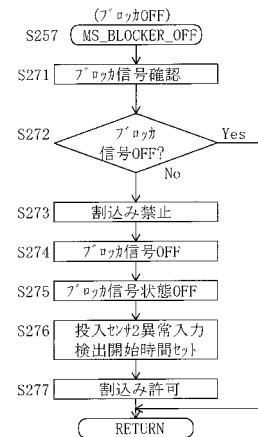
【図 26】



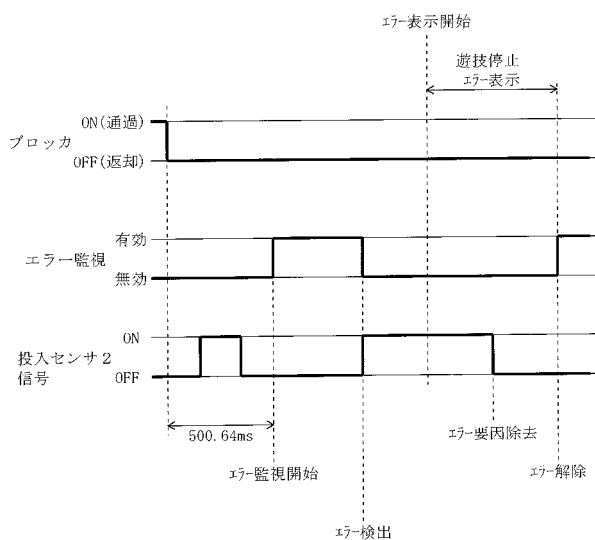
【図 27】



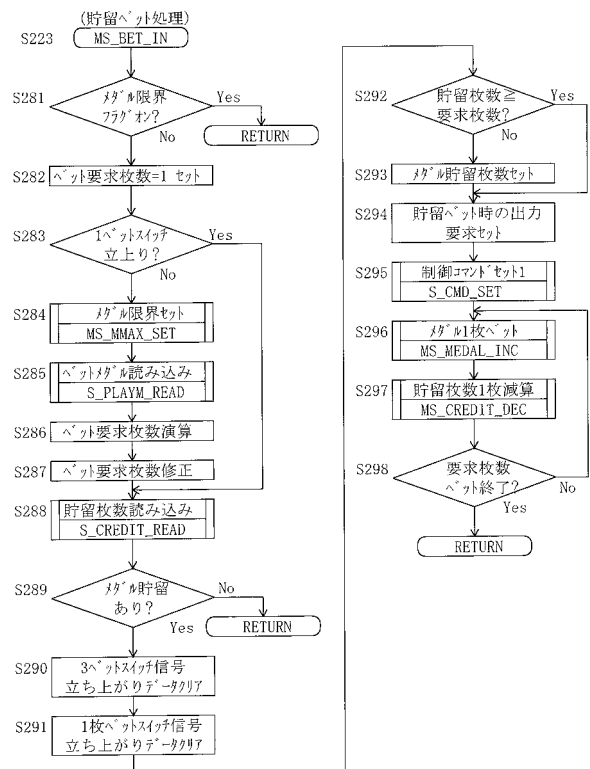
【図 28】



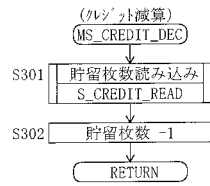
【図 29】



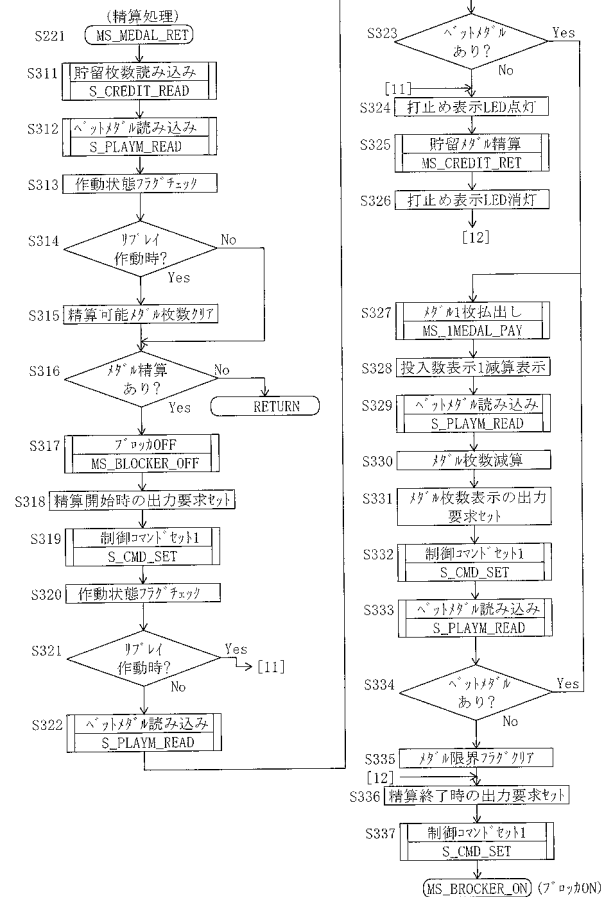
【図 30】



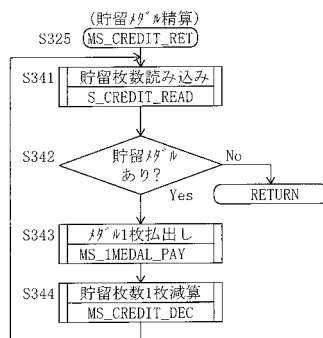
【図 3 1】



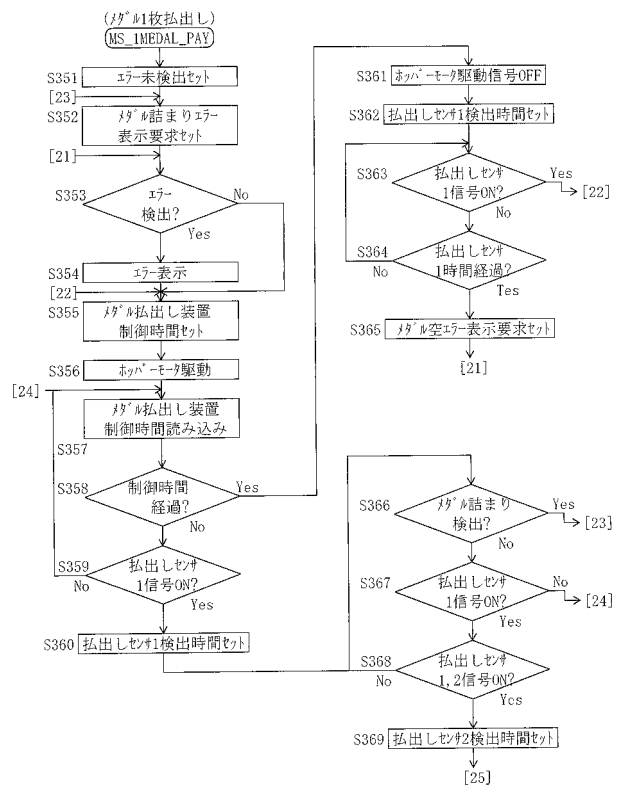
【図 3 2】



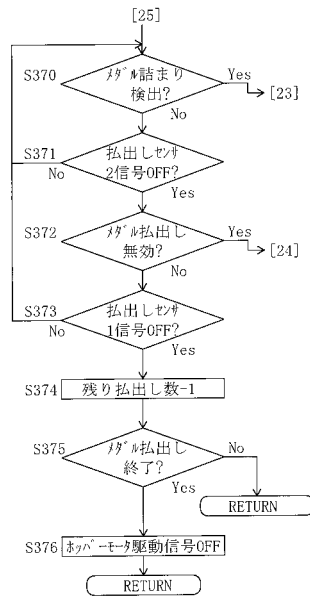
【図 3 3】



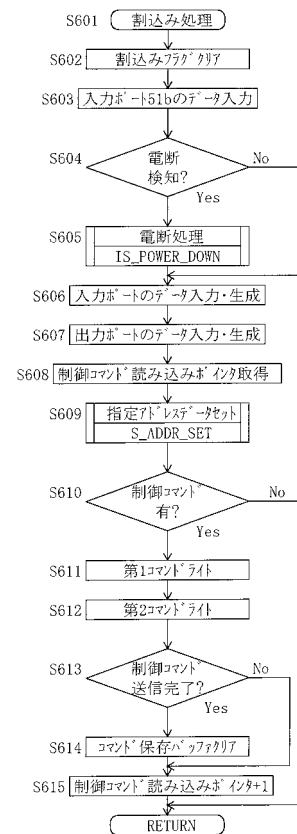
【図 3 4】



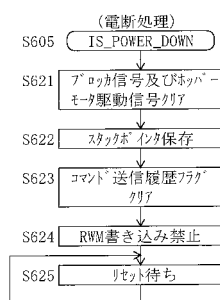
【図 35】



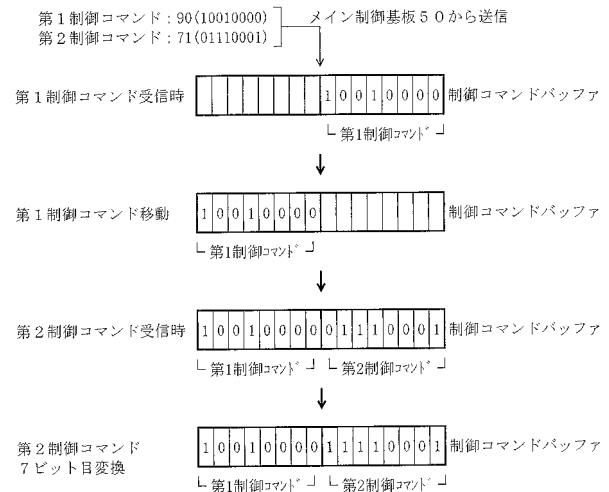
【図 36】



【図 37】



【図 38】



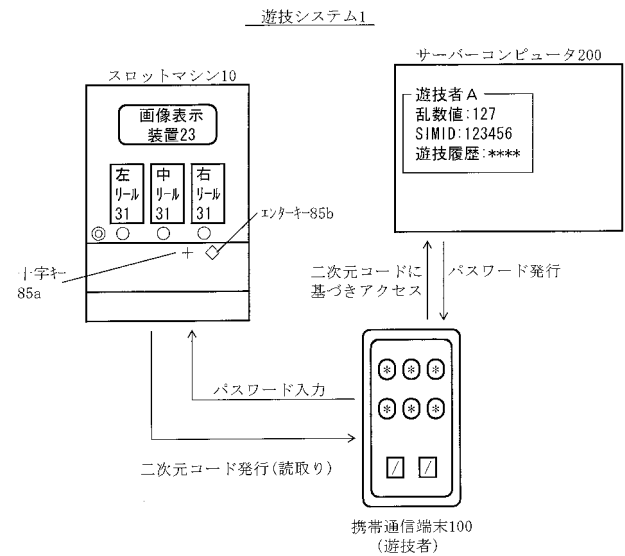
(メモリ83)

番地	内容	データ
7E14	制御コマンド読み込みポインタ	00~FF(H)
7E15	制御コマンド書き込みポインタ	00~FF(H)
7E16	制御コマンドバッファ+0	
7E17	制御コマンドバッファ+1	
7E18	制御コマンドバッファ+2	
7E19	制御コマンドバッファ+3	
{ }		
7E34	制御コマンドバッファ+30	
7E35	制御コマンドバッファ+31	

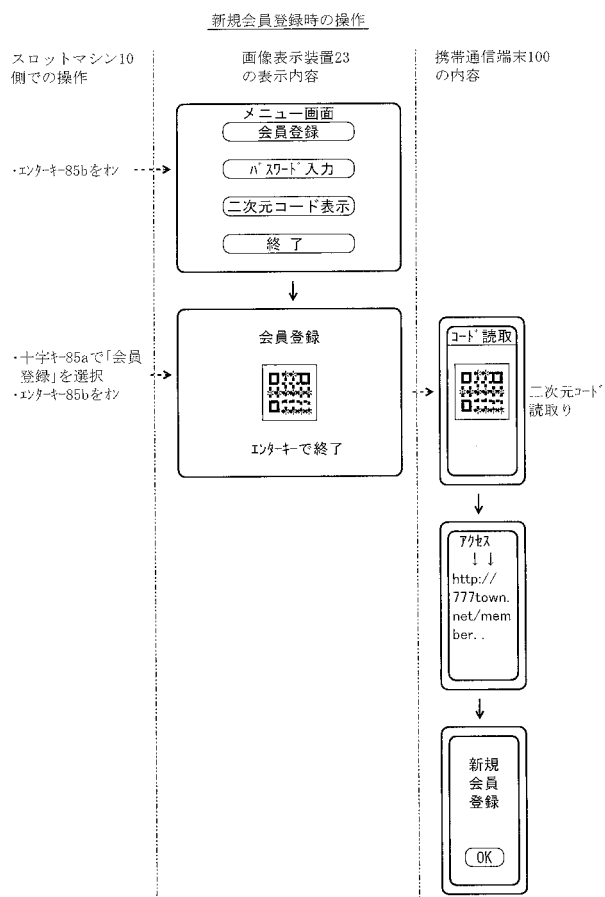
【図 39】

制御コマンドデータ	内容	処理関数	引数
：	：	：	：
：	：	：	：
：	：	：	：
8 F 0 3 (H)	精算終了	<Cmd_First0F>	[0x03]
9 0 0 1 (H)	貯留ベット(1枚)	<Cmd_First10>	[0x01]
9 0 0 2 (H)	貯留ベット(2枚)	<Cmd_First10>	[0x02]
：	：	：	：
：	：	：	：

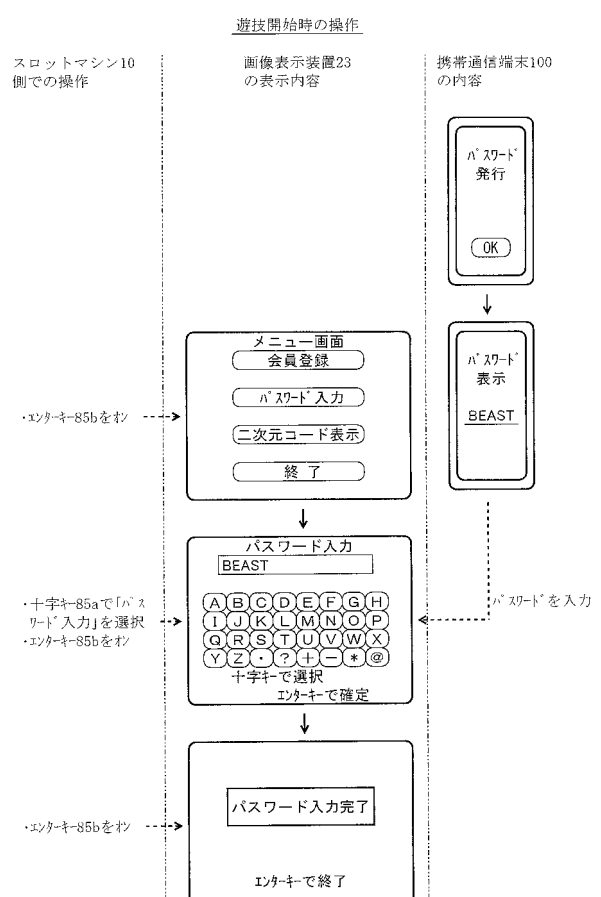
【図 40】



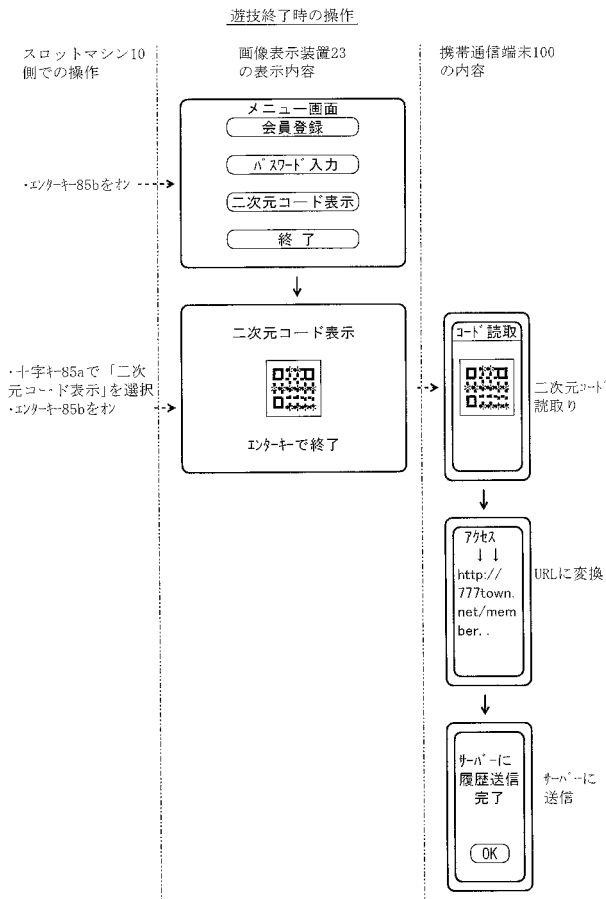
【図 41】



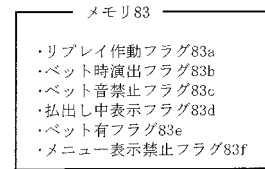
【図 42】



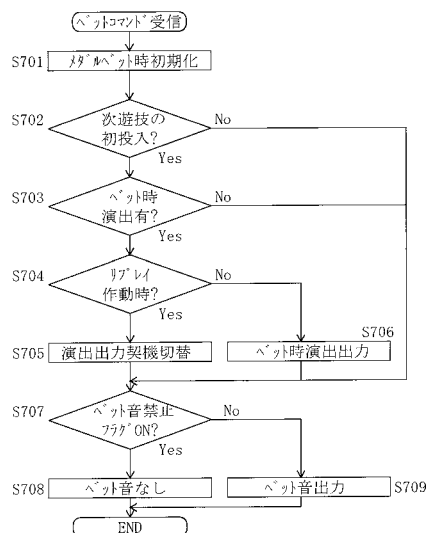
【図 4 3】



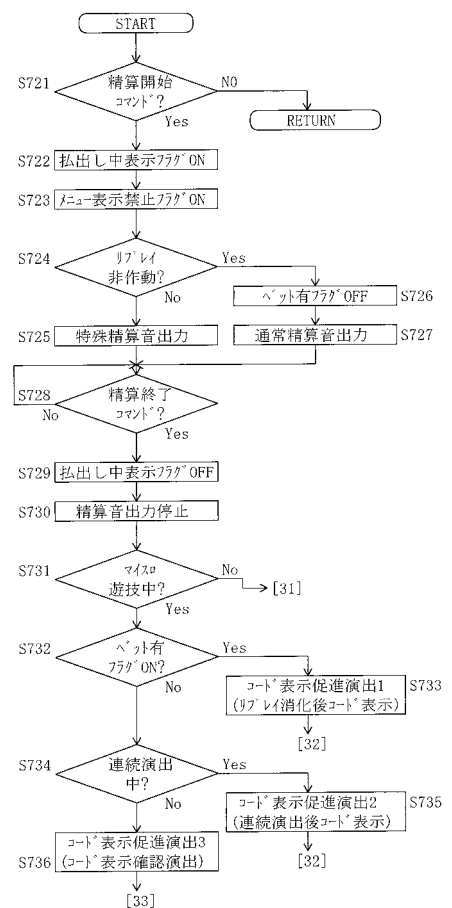
【図 4 4】



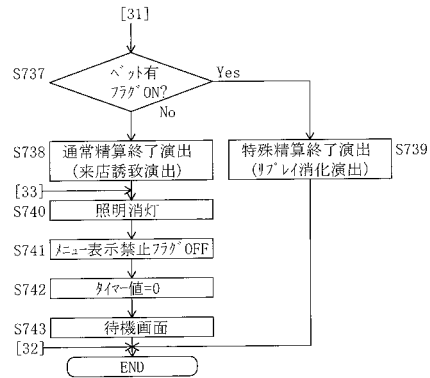
【図 4 5】



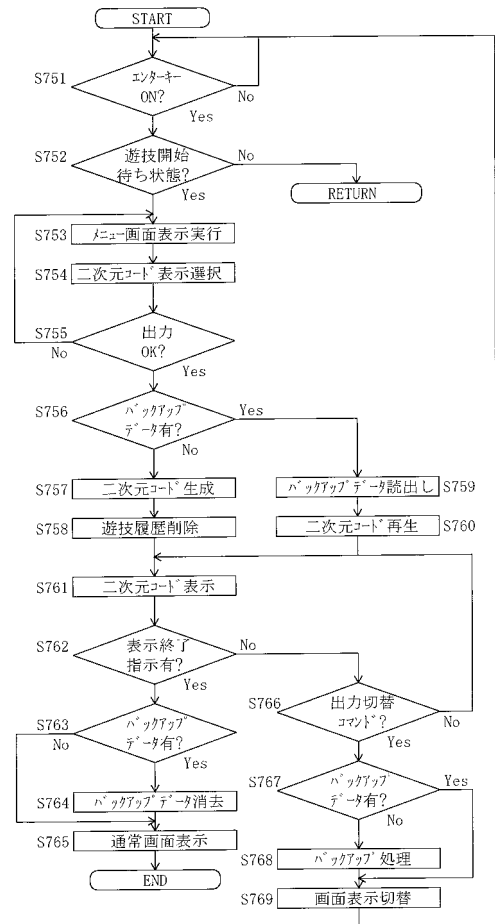
【図 4 6】



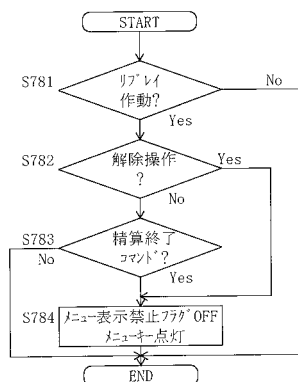
【図 47】



【図 48】



【図 49】



フロントページの続き

Fターム(参考) 2C082 AA02 AB03 AB08 AB12 AB17 AC12 AC14 AC32 AC52 AC74
AC82 BA03 BA07 BA12 BA22 BA32 BA35 BA38 BB02 BB03
BB13 BB16 BB42 BB46 BB74 BB83 BB93 BB96 CA02 CA23
CA24 CA25 CA27 CB01 CB21 CB33 CB49 CB50 CC01 CC13
CC35 CC51 CD11 CD12 CD13 CD23 CD55 DA52 DA54 DA58
DA63 DA64 DA65 DB07 DB16 DB23 EA35 EB03