

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7636256号
(P7636256)

(45)発行日 令和7年2月26日(2025.2.26)

(24)登録日 令和7年2月17日(2025.2.17)

(51)国際特許分類	F I			
A 6 3 F	5/04	(2006.01)	A 6 3 F	5/04 6 1 1 B
			A 6 3 F	5/04 6 9 9
			A 6 3 F	5/04 6 0 5 B
			A 6 3 F	5/04 6 9 1 A
請求項の数 1 (全359頁)				

(21)出願番号	特願2021-75217(P2021-75217)	(73)特許権者	000154679
(22)出願日	令和3年4月27日(2021.4.27)		株式会社平和
(65)公開番号	特開2022-169282(P2022-169282 A)	(74)代理人	東京都台東区東上野一丁目16番1号 110000936
(43)公開日	令和4年11月9日(2022.11.9)		弁理士法人青海国際特許事務所
審査請求日	令和6年1月22日(2024.1.22)	(72)発明者	杉山 純也
			東京都台東区東上野一丁目16番1号 株式会社平和内
		(72)発明者	久保田 智裕
			東京都台東区東上野一丁目16番1号 株式会社平和内
		(72)発明者	田村 純一
			東京都台東区東上野一丁目16番1号 株式会社平和内
		(72)発明者	山口 功太郎
			最終頁に続く

(54)【発明の名称】 遊技機

(57)【特許請求の範囲】

【請求項1】
遊技の進行を制御する第1CPUと、遊技に供する電子化遊技価値を管理する第2CPUとを有し、前記電子化遊技価値の貸し出しを行う特定ユニットと接続可能な遊技機であって、
前記第2CPUと前記特定ユニットとが通信を確立することで、前記第2CPUの情報が前記特定ユニットに通知され、
前記第2CPUの情報には、第1情報、第2情報、第3情報が含まれ、
前記第1情報、前記第2情報、および、前記第3情報には、前記第1情報、前記第2情報、および、前記第3情報のいずれかの通知タイミングが重なった場合に前記特定ユニットに通知される情報の優先順位が予め設定されており、
前記第3情報が所定の条件を満たす場合、前記第3情報よりも前記第1情報が優先して通知されるが、
前記第3情報が所定の条件を満たさない場合、前記第1情報よりも前記第3情報が優先して通知され、
前記電子化遊技価値の少なくとも一部を前記特定ユニットに転送するための計数処理を実行可能であり、
前記特定ユニットとの接続状態を示す特定信号が未接続を示している場合、前記遊技機における遊技の進行、および、前記計数処理の実行を制限し、
前記特定信号が未接続を示していなくとも、前記第1CPUと前記第2CPUとの通信

が確立されていなければ、前記遊技機における遊技の進行、および、前記計数処理の実行を制限する遊技機。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、遊技者に遊技上の利益を付与するか否かを抽選により決定する遊技機に関する。

【背景技術】

【0002】

遊技機としてのパチンコ機では、遊技者のハンドル操作により遊技盤内の遊技領域に向かって遊技球が発射され、遊技領域を流下した遊技球が始動口に入球したことを条件に特別図柄に係る抽選が実行される。そして、特別図柄表示器に大当たりであることを示す特定の特別図柄が停止表示されると、通常の遊技に比べて遊技者に有利な大役遊技が開始され、遊技者は多くの賞球（遊技媒体、遊技価値）の払出を受けることが可能となる。

10

【0003】

また、遊技機としてのスロットマシンでは、遊技者によるメダル（遊技媒体、遊技価値）のベットおよびスタートスイッチの操作に応じて、当選役の抽選を行うとともに、種々の図柄が記された複数のリールが回転する。そして、抽選結果と遊技者によるストップスイッチの操作に応じてリールが順次停止され、払い出しの対象となるライン上である有効ライン上に、当選役に対応する図柄組み合わせが表示されると、所定枚数のメダルが払い出されるなど、遊技上の利益（以下、単に遊技利益という）が遊技者に付与されることとなる。

20

【0004】

また、パチンコ機を封入循環式にして遊技者が遊技球に触れることなく遊技を進行することができる管理遊技機（例えば、特許文献1）や、メダルの介在なしに遊技を進行することができるメダルレス遊技機（例えば、特許文献2）の開発が進んでいる。

【先行技術文献】

【特許文献】

【0005】

【文献】特開2020-156551号公報

30

【文献】特開2015-134014号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

このような管理遊技機やメダルレス遊技機では、遊技機の外部に、遊技球やメダル等の遊技媒体を流通させるための経路を設ける必要がなくなり、メダルレス遊技機においては、物理的な遊技媒体自体が不要となる。このように、メダルレス遊技機では、遊技媒体自体が使用されず、管理遊技機においては非磁性の遊技球を用いることで、金属製の遊技媒体を使用することを前提とするゴト行為を防止することが可能となる。また、遊技機内に遊技媒体の投入や払い出しを行う機構を設ける必要がなくなるので、設計コストや製造コストを削減することができる。さらに、遊技者への遊技媒体の貸し出しや、獲得した遊技媒体の計数等を一元管理することで、不正防止や射幸性を抑制することが可能となる。一方、物理的な遊技媒体の代わりに、電子的な遊技媒体（遊技価値）や非磁性の遊技媒体を利用するための構成に対し、新たに生じ得る不正を未然に防止する仕組みが必要となる。

40

【0007】

本発明は、このような課題に鑑み、遊技を適切に進行することが可能な遊技機を提供することを目的としている。

【課題を解決するための手段】

【0008】

上記課題を解決するために、遊技の進行を制御する第1CPU（例えば、メインCPU

50

500a)と、遊技に供する電子化遊技価値を管理する第2CPU(例えば、メダル数制御CPU946)とを有し、前記電子化遊技価値の貸し出しを行う特定ユニット(例えば、専用ユニット904)と接続可能な本発明の遊技機は、前記第2CPUと前記特定ユニットとが通信を確立することで、前記第2CPUの情報が前記特定ユニットに通知され、前記第2CPUの情報には、第1情報(例えば、遊技機性能情報)、第2情報(例えば、遊技機設置情報)、第3情報(例えば、ホールコン・不正監視情報)が含まれ、前記第1情報、前記第2情報、および、前記第3情報には、前記第1情報、前記第2情報、および、前記第3情報のいずれかの通知タイミングが重なった場合に前記特定ユニットに通知される情報の優先順位が予め設定されており、前記第3情報が所定の条件を満たす(主制御状態の更新がなく、かつ、遊技情報がない)場合、前記第3情報よりも前記第1情報が優先して通知されるが、前記第3情報が所定の条件を満たさない(主制御状態の更新がある、または、遊技情報がある)場合、前記第1情報よりも前記第3情報が優先して通知され、前記電子化遊技価値の少なくとも一部を前記特定ユニットに転送するための計数処理を実行可能であり、前記特定ユニットとの接続状態を示す特定信号(例えば、VL接続信号)が未接続を示している場合、前記遊技機における遊技の進行、および、前記計数処理の実行を制限し(遊技停止処理)、前記特定信号が未接続を示していなくとも、前記第1CPUと前記第2CPUとの通信が確立されていなければ、前記遊技機における遊技の進行、および、前記計数処理の実行を制限する(遊技停止処理)。

10

【発明の効果】

【0011】

20

本発明によれば、遊技を適切に進行することが可能となる。

【図面の簡単な説明】

【0012】

【図1】同時回し参考例に係る扉が開放された状態を示す遊技機の斜視図である。

【図2】同時回し参考例に係る遊技機の正面図である。

【図3】同時回し参考例に係る第2大入賞口を説明する図である。

【図4】同時回し参考例に係る遊技の進行を制御する制御手段の内部構成を示すブロック図である。

【図5】同時回し参考例に係るメインCPUが用いるメモリ領域のアドレスマップである。

【図6】同時回し参考例に係る低確時大当たり決定乱数判定テーブルを説明する図である。

30

【図7】同時回し参考例に係る高確時大当たり決定乱数判定テーブルを説明する図である。

【図8】同時回し参考例に係る当たり図柄乱数判定テーブルおよび小当たり図柄乱数判定テーブルを説明する図である。

【図9】同時回し参考例に係るリーチグループ決定乱数判定テーブルを説明する図である。

【図10】同時回し参考例に係るリーチモード決定乱数判定テーブルを説明する図である。

【図11】同時回し参考例に係る変動パターン乱数判定テーブルを説明する図である。

【図12】同時回し参考例に係る変動時間決定テーブルを説明する図である。

【図13】同時回し参考例に係る遊技状態および変動時間を説明する図である。

【図14】同時回し参考例に係る特別電動役物作動ラムセットテーブルを説明する第1の図である。

40

【図15】同時回し参考例に係る特別電動役物作動ラムセットテーブルを説明する第2の図である。

【図16】同時回し参考例に係る遊技状態設定テーブルを説明する図である。

【図17】同時回し参考例に係る当たり決定乱数判定テーブルを説明する図である。

【図18】(a)は同時回し参考例に係る普通図柄変動時間データテーブルを説明する図であり、(b)は同時回し参考例に係る開閉制御パターンテーブルを説明する図である。

【図19】同時回し参考例に係る本来の遊技性に則った遊技状態の遷移を説明する図である。

【図20】同時回し参考例に係る適切に遊技が行われなかった場合の遊技状態の遷移を説明する図である。

50

- 【図 2 1】同時回し参考例に係る遊技機状態フラグを説明する図である。
- 【図 2 2】同時回し参考例に係る主制御基板における CPU 初期化処理を説明する第 1 のフローチャートである。
- 【図 2 3】同時回し参考例に係る主制御基板における CPU 初期化処理を説明する第 2 のフローチャートである。
- 【図 2 4】同時回し参考例に係る主制御基板におけるサブコマンド群セット処理を説明するフローチャートである。
- 【図 2 5】同時回し参考例に係る主制御基板における電源断時退避処理を説明するフローチャートである。
- 【図 2 6】同時回し参考例に係る主制御基板におけるタイマ割込み処理を説明するフローチャートである。 10
- 【図 2 7】同時回し参考例に係る主制御基板における設定関連処理を説明するフローチャートである。
- 【図 2 8】同時回し参考例に係る主制御基板におけるスイッチ管理処理を説明するフローチャートである。
- 【図 2 9】同時回し参考例に係る主制御基板におけるゲート通過処理を説明するフローチャートである。
- 【図 3 0】同時回し参考例に係る主制御基板における第 1 始動口通過処理を説明するフローチャートである。
- 【図 3 1】同時回し参考例に係る主制御基板における第 2 始動口通過処理を説明するフローチャートである。 20
- 【図 3 2】同時回し参考例に係る主制御基板における特別図柄乱数取得処理を説明するフローチャートである。
- 【図 3 3】同時回し参考例に係る主制御基板における取得時演出判定処理を説明するフローチャートである。
- 【図 3 4】同時回し参考例に係る主制御基板における大入賞口通過処理を説明するフローチャートである。
- 【図 3 5】同時回し参考例に係る特別遊技管理フェーズおよび特別電動役物遊技管理フェーズを説明する図である。
- 【図 3 6】同時回し参考例に係る主制御基板における特別遊技管理処理を説明するフローチャートである。 30
- 【図 3 7】同時回し参考例に係る主制御基板における特別図柄変動待ち処理を説明するフローチャートである。
- 【図 3 8】同時回し参考例に係る主制御基板における特別図柄当たり判定処理を説明するフローチャートである。
- 【図 3 9】同時回し参考例に係る主制御基板における特別図柄変動番号決定処理を説明するフローチャートである。
- 【図 4 0】同時回し参考例に係る主制御基板における回数切り管理処理を説明するフローチャートである。
- 【図 4 1】同時回し参考例に係る主制御基板における特別図柄変動中処理を説明するフローチャートである。 40
- 【図 4 2】同時回し参考例に係る主制御基板における図柄強制停止処理を説明するフローチャートである。
- 【図 4 3】同時回し参考例に係る主制御基板における特別図柄停止図柄表示処理を説明するフローチャートである。
- 【図 4 4】同時回し参考例に係る主制御基板における特別電動役物遊技管理処理を説明するフローチャートである。
- 【図 4 5】同時回し参考例に係る主制御基板における大入賞口開放前処理を説明するフローチャートである。
- 【図 4 6】同時回し参考例に係る主制御基板における大入賞口開閉切替処理を説明するフ 50

ローチャートである。

【図 4 7】同時回し参考例に係る主制御基板における大入賞口開放制御処理を説明するフローチャートである。

【図 4 8】同時回し参考例に係る主制御基板における大入賞口閉鎖有効処理を説明するフローチャートである。

【図 4 9】同時回し参考例に係る主制御基板における大入賞口終了ウェイト処理を説明するフローチャートである。

【図 5 0】同時回し参考例に係る普通遊技管理フェーズを説明する図である。

【図 5 1】同時回し参考例に係る主制御基板における普通遊技管理処理を説明するフローチャートである。

【図 5 2】同時回し参考例に係る主制御基板における普通図柄変動待ち処理を説明するフローチャートである。

【図 5 3】同時回し参考例に係る主制御基板における普通図柄変動中処理を説明するフローチャートである。

【図 5 4】同時回し参考例に係る主制御基板における普通図柄停止図柄表示処理を説明するフローチャートである。

【図 5 5】同時回し参考例に係る主制御基板における普通電動役物入賞口開放前処理を説明するフローチャートである。

【図 5 6】同時回し参考例に係る主制御基板における普通電動役物入賞口開閉切替処理を説明するフローチャートである。

【図 5 7】同時回し参考例に係る主制御基板における普通電動役物入賞口開放制御処理を説明するフローチャートである。

【図 5 8】同時回し参考例に係る主制御基板における普通電動役物入賞口閉鎖有効処理を説明するフローチャートである。

【図 5 9】同時回し参考例に係る主制御基板における普通電動役物入賞口終了ウェイト処理を説明するフローチャートである。

【図 6 0】演出参考例に係るリーチなし変動パターンの変動演出の一例を説明する図である。

【図 6 1】演出参考例に係るノーマルリーチ変動パターンの変動演出の一例を説明する図である。

【図 6 2】演出参考例に係るハズレ時の発展リーチ変動パターンの変動演出の一例を説明する図である。

【図 6 3】演出参考例に係る大当たり時の発展リーチ変動パターンの変動演出の一例を説明する図である。

【図 6 4】演出参考例に係るリーチ発展演出が 2 回実行される場合の変動演出の一例を説明する図である。

【図 6 5】演出参考例に係る擬似連続リーチ変動パターンの変動演出の一例を説明する図である。

【図 6 6】演出参考例に係る変動演出決定テーブルを説明する図である。

【図 6 7】演出参考例に係る保留表示演出の一例を説明する図である。

【図 6 8】(a) は演出参考例に係る最終保留表示パターン決定テーブルを説明する図であり、(b) は演出参考例に係る 1 つ前保留表示パターン決定テーブルを説明する図である。

【図 6 9】演出参考例に係る副制御基板におけるサブ CPU 初期化処理を説明するフローチャートである。

【図 7 0】演出参考例に係る副制御基板におけるサブタイマ割込み処理を説明するフローチャートである。

【図 7 1】演出参考例に係る副制御基板における先読み指定コマンド受信処理を説明するフローチャートである。

【図 7 2】演出参考例に係る副制御基板における変動コマンド受信処理を説明するフロー

10

20

30

40

50

チャートである。

【図 7 3】スロットマシンの概略的な機械的構成を説明するための外観図である。

【図 7 4】スロットマシンの概略的な機械的構成を説明するための前面扉を開いた状態での外観図である。

【図 7 5】リールの図柄配列および有効ラインを説明する図である。

【図 7 6】スロットマシンの概略的な電氣的構成を示したブロック図である。

【図 7 7】当選役を説明するための説明図である。

【図 7 8】当選種別抽選テーブルを示す図である。

【図 7 9】当選種別抽選テーブルを示す図である。

【図 8 0】遊技状態の遷移を説明するための説明図である。

10

【図 8 1】演出状態の遷移を説明するための説明図である。

【図 8 2】主制御基板における CPU 初期化処理を説明するフローチャートである。

【図 8 3】主制御基板におけるコールドスタート処理を説明するフローチャートである。

【図 8 4】主制御基板におけるエラー停止処理を説明するフローチャートである。

【図 8 5】主制御基板における設定値切り替え処理を説明するフローチャートである。

【図 8 6】主制御基板における初期化スタート処理を説明するフローチャートである。

【図 8 7】主制御基板における状態復帰処理を説明するフローチャートである。

【図 8 8】主制御基板における遊技開始処理を説明するフローチャートである。

【図 8 9】主制御基板における遊技メダル投入処理を説明するフローチャートである。

【図 9 0】主制御基板における内部抽選処理を説明するフローチャートである。

20

【図 9 1】主制御基板における図柄コード設定処理を説明するフローチャートである。

【図 9 2】主制御基板における回胴回転中処理を説明するフローチャートである。

【図 9 3】主制御基板における回胴停止処理を説明するフローチャートである。

【図 9 4】主制御基板における表示判定処理を説明するフローチャートである。

【図 9 5】主制御基板における払出処理を説明するフローチャートである。

【図 9 6】主制御基板における遊技移行処理を説明するフローチャートである。

【図 9 7】主制御基板における電源断時退避処理を説明するフローチャートである。

【図 9 8】主制御基板におけるタイマ割込み処理を説明するフローチャートである。

【図 9 9】メイン CPU 周辺の電氣的な接続を説明するための図である。

【図 1 0 0】CPU コアの内部構成を示したブロック図である。

30

【図 1 0 1】レジスタの構成を説明した図である。

【図 1 0 2】メモリマップを示す説明図である。

【図 1 0 3】主制御基板の外観を示した説明図である。

【図 1 0 4】比率表示部の表示態様を説明するための説明図である。

【図 1 0 5】役物比率の第 1 の演算例を示すフローチャートである。

【図 1 0 6】8 ビット MUL 命令のみを用いた第 1 の演算例を説明するための説明図である。

【図 1 0 7】第 1 の演算例を実現するための具体的な処理を示したフローチャートである。

【図 1 0 8】第 1 の演算例を実現するための具体的なコマンドの一例を示した図である。

【図 1 0 9】16 ビット乗算器を用いた第 2 の演算例を説明するための説明図である。

40

【図 1 1 0】第 2 の演算例を実現するための具体的な処理を示したフローチャートである。

【図 1 1 1】第 2 の演算例を実現するための具体的なコマンドの一例を示した図である。

【図 1 1 2】第 1 の演算例を実現するための具体的なコマンドの一例を示した図である。

【図 1 1 3】2 の累乗数を乗じた数の繰り返し減算を用いた第 3 の演算例を説明するための説明図である。

【図 1 1 4】第 3 の演算例を実現するための具体的な処理を示したフローチャートである。

【図 1 1 5】第 3 の演算例を実現するための具体的なコマンドの一例を示した図である。

【図 1 1 6】第 4 の演算例を実現するための具体的な処理を示したフローチャートである。

【図 1 1 7】第 4 の演算例を実現するための具体的なコマンドの一例を示した図である。

【図 1 1 8】第 5 の演算例を実現するための具体的な処理を示したフローチャートである。

50

- 【図 1 1 9】第 5 の演算例を実現するための具体的なコマンドの一例を示した図である。
- 【図 1 2 0】第 6 の演算例を実現するための具体的な処理を示したフローチャートである。
- 【図 1 2 1】第 6 の演算例を実現するための具体的なコマンドの一例を示した図である。
- 【図 1 2 2】BY T E S E L モジュールの具体的な処理を示したフローチャートである。
- 【図 1 2 3】BY T E S E L モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 2 4】BY T E S E L モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 2 5】BY T E S E L モジュールを実現するためのコマンドの他の例を説明するための説明図である。
- 【図 1 2 6】BY T E S E L モジュールを実現するためのコマンドのさらに他の例を説明するための説明図である。
- 【図 1 2 7】WORD S E L モジュールの具体的な処理を示したフローチャートである。
- 【図 1 2 8】WORD S E L モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 2 9】WORD S E L モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 3 0】WORD S E L モジュールを実現するためのコマンドの他の例を説明するための説明図である。
- 【図 1 3 1】WORD S E L モジュールを実現するためのコマンドのさらに他の例を説明するための説明図である。
- 【図 1 3 2】CAL _ M O D モジュールを説明するための説明図である。
- 【図 1 3 3】H I D _ J U G モジュールを説明するための説明図である。
- 【図 1 3 4】RAM S E T モジュールの具体的な処理を示したフローチャートである。
- 【図 1 3 5】RAM S E T モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 3 6】RAM S E T モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 3 7】RAM S E T モジュールを実現するためのコマンドの他の例を説明するための説明図である。
- 【図 1 3 8】TAB L S E T モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 3 9】TAB L S E T モジュールを実現するためのコマンドの他の例を説明するための説明図である。
- 【図 1 4 0】BY T E D E C モジュールの具体的な処理を示したフローチャートである。
- 【図 1 4 1】BY T E D E C モジュールにおけるデクリメント態様とゼロフラグおよびキャリーフラグの設定を説明するための説明図である。
- 【図 1 4 2】BY T E D E C モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 4 3】BY T E D E C モジュールを実現するためのコマンドの他の例を説明するための説明図である。
- 【図 1 4 4】RAM _ D E C モジュールを説明するための説明図である。
- 【図 1 4 5】WORD D E C モジュールの具体的な処理を示したフローチャートである。
- 【図 1 4 6】WORD D E C モジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図 1 4 7】WORD D E C モジュールを実現するためのコマンドの他の例を説明するための説明図である。
- 【図 1 4 8】サブルーチンから戻る処理の一例を説明するための説明図である。
- 【図 1 4 9】P Y _ C M D A モジュールを説明するための説明図である。
- 【図 1 5 0】G A T _ P A S モジュールを説明するための説明図である。

10

20

30

40

50

- 【図151】TDN__PASモジュールを説明するための説明図である。
- 【図152】FD__OPNモジュールを説明するための説明図である。
- 【図153】TZ__STAモジュールを説明するための説明図である。
- 【図154】TZ__RGETモジュールを説明するための説明図である。
- 【図155】TRSVSELモジュールを説明するための説明図である。
- 【図156】TDOVCHKモジュールを説明するための説明図である。
- 【図157】BER__CHKモジュールを説明するための説明図である。
- 【図158】TEF__SELモジュールを説明するための説明図である。
- 【図159】SET__RIGモジュールを説明するための説明図である。
- 【図160】PRE__LOTモジュールを説明するための説明図である。
- 【図161】REG__LOTモジュールを説明するための説明図である。
- 【図162】BIG__SLTモジュールを説明するための説明図である。
- 【図163】NAV__SETモジュールを説明するための説明図である。
- 【図164】TOK__PRCモジュールの具体的な処理を示したフローチャートである。
- 【図165】TOK__PRCモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図166】TEF__SELモジュールの具体的な処理を示したフローチャートである。
- 【図167】TEF__SELモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図168】SWI__PRCモジュールの具体的な処理を示したフローチャートである。
- 【図169】SWI__PRCモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図170】CPUINITモジュールの具体的な処理を示したフローチャートである。
- 【図171】CPUINITモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図172】TMR__IPTモジュールを説明するための説明図である。
- 【図173】FZ__SPNモジュールの具体的な処理を示したフローチャートである。
- 【図174】FZ__SPNモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図175】CPUINITモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図176】EXE__SETモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図177】HPT__GRPモジュールの具体的な処理を示したフローチャートである。
- 【図178】(a)、(b)は、HPT__GRPモジュールを実現するためのコマンドの一例を説明するための説明図であり、(c)は、リーチグループ決定乱数判定テーブルの一例を説明するための説明図である。
- 【図179】E__ILGERモジュールを説明するための説明図である。
- 【図180】E__LEVOTモジュールを説明するための説明図である。
- 【図181】SBC__OUTモジュールの具体的な処理を示したフローチャートである。
- 【図182】SBC__OUTモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図183】SBC__OUTモジュールを実現するための他のコマンドの一例を説明するための説明図である。
- 【図184】FZ__OPNモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図185】TD__OPNモジュールを実現するためのコマンドの一例を説明するための説明図である。
- 【図186】HSY__PRCモジュールを実現するためのコマンドの一例を説明するための説明図である。

10

20

30

40

50

【図187】FDN__CHKモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図188】FZ__STPモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図189】乱数発生器の構成を説明するためのブロック図である。

【図190】乱数生成部の組み合わせを説明する図である。

【図191】SMC__ROTモジュールの具体的な処理を示したフローチャートである。

【図192】SMC__ROTモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図193】SMC__ROTモジュールを実現するためのコマンドの他の例を説明するための説明図である。

10

【図194】INITIALモジュールの具体的な処理を示したフローチャートである。

【図195】INITIALモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図196】INITIALモジュールを実現するためのコマンドの他の例を説明するための説明図である。

【図197】RANKSETモジュールを説明するための説明図である。

【図198】PWRFAILモジュールを説明するための説明図である。

【図199】DYM__OUTモジュールを説明するための説明図である。

【図200】IPT__PDモジュールの具体的な処理を示したフローチャートである。

20

【図201】IPT__PDモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図202】IPT__PDモジュールを実現するためのコマンドの他の例を説明するための説明図である。

【図203】DYNMOUTモジュールを説明するための説明図である。

【図204】EXT__PRCモジュールを説明するための説明図である。

【図205】STOPDCTモジュールの具体的な処理を示したフローチャートである。

【図206】STOPDCTモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図207】STOPDCTモジュールを実現するためのコマンドの他の例を説明するための説明図である。

30

【図208】E__SETTMモジュールの具体的な処理を示したフローチャートである。

【図209】E__SETTMモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図210】E__SETTMモジュールを実現するためのコマンドの他の例を説明するための説明図である。

【図211】DYM__OUTモジュールを説明するための説明図である。

【図212】DYM__OUTモジュールを説明するための説明図である。

【図213】E__SETTMモジュールの具体的な処理を示したフローチャートである。

【図214】E__SETTMモジュールを実現するためのコマンドのさらに他の例を説明するための説明図である。

40

【図215】RAM__INCモジュールの具体的な処理を示したフローチャートである。

【図216】RAM__INCモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図217】RAM__INCモジュールを実現するためのコマンドの一例を説明するための説明図である。

【図218】RAM__INCモジュールを実現するためのコマンドの他の例を説明するための説明図である。

【図219】RAM__INCモジュールを実現するためのコマンドの他の例を説明するための説明図である。

50

【図 2 2 0】 T A B L S E T モジュールの具体的な処理を示したフローチャートである。

【図 2 2 1】 T A B L S E T モジュールを実現するためのコマンドの一例を説明するための説明図である。

【図 2 2 2】 T A B L S E T モジュールを実現するためのコマンドの一例を説明するための説明図である。

【図 2 2 3】 T A B L S E T モジュールを実現するためのコマンドの他の例を説明するための説明図である。

【図 2 2 4】 I P T _ P C モジュールの具体的な処理を示したフローチャートである。

【図 2 2 5】 I P T _ P C モジュールを実現するためのコマンドの一例を説明するための説明図である。

10

【図 2 2 6】 C M D P R O C モジュールの具体的な処理を示したフローチャートである。

【図 2 2 7】 C M D P R O C モジュールを実現するためのコマンドの一例を説明するための説明図である。

【図 2 2 8】 S E T _ P L S モジュールの具体的な処理を示したフローチャートである。

【図 2 2 9】 S E T _ P L S モジュールを実現するためのコマンドの一例を説明するための説明図である。

【図 2 3 0】 O T M _ A T K モジュールの具体的な処理を示したフローチャートである。

【図 2 3 1】 O T M _ A T K モジュールを実現するためのコマンドの一例を説明するための説明図である。

【図 2 3 2】 K R S _ J D G モジュールを実現するためのコマンドの一例を説明するための説明図である。

20

【図 2 3 3】 K R S _ J D G モジュールを実現するためのコマンドの他の例を説明するための説明図である。

【図 2 3 4】 Q ' レジスタに値を設定するコマンドの一例を示した図である。

【図 2 3 5】 Q ' レジスタに値を設定するコマンドの一例を示した図である。

【図 2 3 6】 Q ' レジスタに値を設定するコマンドの一例を示した図である。

【図 2 3 7】 Q ' レジスタに値を設定するコマンドの一例を示した図である。

【図 2 3 8】 レジスタバンクおよびレジスタグループの遷移を説明するための図である。

【図 2 3 9】 遊技システムを示した機能ブロック図である。

【図 2 4 0】 メダルレス遊技機および専用ユニットの概略的な機械的構成を説明するための外観図である。

30

【図 2 4 1】 メダルレス遊技機および専用ユニットの概略的な電氣的構成を示したブロック図である。

【図 2 4 2】 他の基板構成を説明するための説明図である。

【図 2 4 3】 ケースの封入態様を説明するための説明図である。

【図 2 4 4】 メダルレス遊技機の各機能を実行する C P U および領域を説明するための説明図である。

【図 2 4 5】 遊技機情報通知のフォーマットを説明するための説明図である。

【図 2 4 6】 遊技機情報通知のフォーマットを説明するための説明図である。

【図 2 4 7】 遊技機情報通知のフォーマットを説明するための説明図である。

40

【図 2 4 8】 遊技機情報通知のフォーマットを説明するための説明図である。

【図 2 4 9】 計数通知のフォーマットを説明するための説明図である。

【図 2 5 0】 貸出受領結果応答のフォーマットを説明するための説明図である。

【図 2 5 1】 貸出通知のフォーマットを説明するための説明図である。

【図 2 5 2】 遊技機情報通知、計数通知、貸出通知、貸出受領結果応答の通知タイミングを示したタイミングチャートである。

【図 2 5 3】 遊技機情報通知の送信タイミングを説明するための説明図である。

【図 2 5 4】 電源投入時の通信仕様を示したフローチャートである。

【図 2 5 5】 操作時の通信仕様を示したフローチャートである。

【図 2 5 6】 遊技終了時の通信仕様を示したフローチャートである。

50

【図 2 5 7】設定変更中信号の更新処理を示したフローチャートである。

【図 2 5 8】設定確認中信号の更新処理を示したフローチャートである。

【図 2 5 9】メイン CPU とメダル数制御 CPU との通信処理の比較例を示した説明図である。

【図 2 6 0】コマンドの 1 バイト送信の概念を示したフローチャートとコマンドを示した図である。

【図 2 6 1】メイン CPU とメダル数制御 CPU との通信処理を示した説明図である。

【発明を実施するための形態】

【0013】

以下に添付図面を参照しながら、本発明の好適な実施形態について詳細に説明する。かかる実施形態に示す寸法、材料、その他具体的な数値等は、発明の理解を容易とするための例示にすぎず、特に断る場合を除き、本発明を限定するものではない。なお、本明細書および図面において、実質的に同一の機能、構成を有する要素については、同一の符号を付することにより重複説明を省略し、また本発明に直接関係のない要素は図示を省略する。

10

【0014】

本発明の実施形態では、遊技機として、パチンコ機とスロットマシンとをその順に例示し、その後、具体的な処理を詳述する。

【0015】

〈パチンコ機〉

本発明の実施例の理解を容易にするため、まず、同時回し参考例として、所謂同時回し機の機械的構成および電氣的構成、および、各基板における具体的な処理を説明する。そして、演出参考例として、同時回し機において実行可能な具体的な演出や当該演出に係る具体的な処理を説明する。その後、本発明の実施例として、各参考例と異なる構成について具体的に説明する。

20

【0016】

〈同時回し参考例〉

図 1 は、同時回し参考例に係る遊技機 100 の斜視図であり、扉が開放された状態を示している。図示のように、遊技機 100 は、略矩形状に組まれた四辺によって囲繞空間が形成される外枠 102 と、この外枠 102 にヒンジ機構によって開閉自在に取り付けられた中枠 104 と、この中枠 104 に、ヒンジ機構によって開閉自在に取り付けられた前枠 106 と、を備えている。

30

【0017】

中枠 104 は、外枠 102 と同様に、略矩形状に組まれた四辺によって囲繞空間が形成されており、この囲繞空間に遊技盤 108 が保持されている。また、前枠 106 には、ガラス製または樹脂製の透過板 110 が保持されている。そして、これら中枠 104 および前枠 106 を外枠 102 に対して閉じると、遊技盤 108 と透過板 110 とが所定の間隔を維持して略平行に対面するとともに、遊技機 100 の正面側から、透過板 110 を介して遊技盤 108 が視認可能となる。

【0018】

図 2 は、同時回し参考例に係る遊技機 100 の正面図である。この図に示すように、前枠 106 の下部には、遊技機 100 の正面側に突出する操作ハンドル 112 が設けられている。この操作ハンドル 112 は、遊技者が回転操作可能に設けられており、遊技者が操作ハンドル 112 を回転させて発射操作を行うと、当該操作ハンドル 112 の回転角度に応じた強度で、不図示の発射機構によって遊技球が発射される。このようにして発射された遊技球は、遊技盤 108 に設けられたレール 114 a、114 b 間を上昇して遊技領域 116 に導かれることとなる。

40

【0019】

遊技領域 116 は、遊技盤 108 と透過板 110 との間隔に形成される空間であって、遊技球が流下または転動可能な領域である。遊技盤 108 には、多数の釘や風車（不図示）が設けられており、遊技領域 116 に導かれた遊技球が釘や風車に衝突して、不規則な

50

方向に流下、転動するようにしている。

【0020】

遊技領域116は、発射機構の発射強度に応じて遊技球の進入度合いを互いに異にし、遊技球の打ち分けが可能な第1遊技領域116aおよび第2遊技領域116bを備えている。第1遊技領域116aは、遊技機100に正対した遊技者から見て遊技領域116の左側に位置し、第2遊技領域116bは、遊技機100に正対した遊技者から見て遊技領域116の右側に位置している。レール114a、114bが遊技領域116の左側にあることから、発射機構によって所定の強度未満の発射強度で発射された遊技球は第1遊技領域116aに進入し、所定の強度以上の発射強度で発射された遊技球は第2遊技領域116bに進入することとなる。

10

【0021】

また、遊技領域116には、遊技球が入球可能な一般入賞口118、第1固定始動口120A、第1可変始動口120B、第2始動口122、普図作動口125が設けられており、これら一般入賞口118、第1固定始動口120A、第1可変始動口120B、第2始動口122、普図作動口125に遊技球が入球すると、それぞれ所定の賞球が遊技者に払い出される。なお、以下では、第1固定始動口120Aおよび第1可変始動口120Bを総称して第1始動口120と呼ぶ。

【0022】

詳しくは後述するが、第1始動口120内には第1始動領域が設けられ、また、第2始動口122内には第2始動領域が設けられている。そして、第1始動口120または第2始動口122に遊技球が入球して第1始動領域または第2始動領域に遊技球が進入すると、予め設けられた複数の特別図柄の中からいずれか1の特別図柄を決定するための抽選が行われる。各特別図柄には、遊技者にとって有利な大役遊技や小当たり遊技の実行可否が対応付けられている。したがって、遊技者は、第1始動口120または第2始動口122に遊技球が入球すると、所定の賞球を獲得すると同時に、種々の遊技利益を受ける権利獲得の機会を獲得することとなる。

20

【0023】

また、第1固定始動口120A、第2始動口122および普図作動口125は、遊技球が常時入球可能に開口した固定始動口で構成される。一方、第1可変始動口120Bには、可動片120bが開閉可能に設けられており、この可動片120bの状態に応じて、第1可変始動口120Bへの遊技球の進入容易性が変化する可変始動口で構成されている。具体的には、可動片120bは、通常、閉状態に維持されており、この間は、第1可変始動口120Bへの遊技球の入球が困難もしくは不可能となる。

30

【0024】

これに対して、遊技領域116（第2遊技領域116b）に設けられたゲート124を遊技球が通過するか、普図作動口125に遊技球が入球すると、後述する普通図柄の抽選が行われ、この抽選によって当たりに当選すると、可動片120bが所定時間、開状態に制御される。可動片120bが開状態になると、第1可変始動口120Bへの遊技球の入球が可能となる。このように、可動片120bは、第1可変始動口120Bへの遊技球の入球を可能とする開状態、および、開状態よりも第1可変始動口120Bへの遊技球の入球が困難もしくは不可能となる閉状態に変移する可動部材（始動可変入賞装置）として機能する。

40

【0025】

なお、第1固定始動口120Aは、第1遊技領域116aを流下する遊技球のみが入球可能となり、第1可変始動口120Bおよび第2始動口122は、第2遊技領域116bを流下する遊技球のみが入球可能となる位置に配されている。なお、第1固定始動口120Aは、第2遊技領域116bを流下する遊技球が入球してもよいが、この場合には、第1遊技領域116aを流下する遊技球の方が、第2遊技領域116bを流下する遊技球よりも入球しやすい位置に配することが望ましい。

【0026】

50

同様に、第1可変始動口120Bおよび第2始動口122は、第1遊技領域116aを流下する遊技球が入球してもよいが、この場合には、第2遊技領域116bを流下する遊技球の方が、第1遊技領域116aを流下する遊技球よりも入球しやすい位置に配することが望ましい。いずれにしても、第1固定始動口120Aは、少なくとも第1遊技領域116aを流下する遊技球が入球可能な位置に配され、第1可変始動口120Bおよび第2始動口122は、少なくとも第2遊技領域116bを流下する遊技球が入球可能な位置に配されるとよい。

【0027】

さらに、第2遊技領域116bには、第1大入賞口126および第2大入賞口128が設けられている。第1大入賞口126および第2大入賞口128は、第2遊技領域116bを流下する遊技球のみが入球可能な位置に配される。ただし、第1大入賞口126および第2大入賞口128は、第1遊技領域116aおよび第2遊技領域116bを流下するいずれの遊技球も入球可能に配されてもよい。

【0028】

第1大入賞口126には、開閉扉126bが開閉可能に設けられており、通常、開閉扉126bが第1大入賞口126を閉鎖して、第1大入賞口126への遊技球の入球が不可能となっている。具体的には、開閉扉126bは、閉鎖状態において、遊技盤108の盤面と面一の状態となり、第1大入賞口126の前を遊技球が流下する。これに対して、前述の大役遊技が実行されると、開閉扉126bが開放されて遊技球を第1大入賞口126に導く受け皿として機能し、第1大入賞口126への遊技球の入球が可能となる。そして、第1大入賞口126に遊技球が入球すると、所定の賞球が遊技者に払い出される。

【0029】

第2大入賞口128は、第2遊技領域116bにおいて、第1大入賞口126の下方に設けられる。第2大入賞口128は、可動片128bを備えており、通常、可動片128bが閉状態に維持されている。これに対して、後述の小当たり遊技が実行されると、可動片128bが開状態に制御され、第2大入賞口128への遊技球の入球が可能となる。なお、以下では、第1大入賞口126および第2大入賞口128をまとめて単に大入賞口ともよぶ。

【0030】

図3は、同時回し参考例に係る第2大入賞口128を説明する図である。第2遊技領域116bには、遊技盤108の正面側に突出する構造物129が設けられている。この構造物129は、遊技機100の左右方向と前後方向とに位置する四辺、および、底辺が圍繞されており、上部に開口が形成されている。この構造物129の上部に形成される開口が、第2大入賞口128となる。構造物129の上部には、可動片128bが設けられており、通常、図3(a)に示すように、可動片128bは、第2大入賞口128を閉鎖する閉状態に維持されている。

【0031】

可動片128bは、遊技機100の上方に臨むようにして、遊技球が転動、流下する遊技領域116に突出している。したがって、可動片128bが閉状態に維持されている場合には、遊技領域116（第2遊技領域116b）を流下する遊技球が、可動片128b上に落下することとなる。ここで、構造物129は、底辺が水平方向に略平行であり、遊技機100の右側の側面は、左側の側面よりも高さ方向に僅かに長い寸法関係となっている。したがって、第2大入賞口128は、遊技機100の左側が右側よりも僅かに低く、閉状態に維持された可動片128bは、遊技機100の左側が右側よりも僅かに低い位置になるように傾斜している。そのため、可動片128bが閉状態にあるときには、図3(a)に矢印で示すように、可動片128b上に落下した遊技球が、可動片128b上を右方から左方へとゆっくりと転動することになる。

【0032】

そして、後述の小当たり遊技が実行されると、可動片128bは、第2大入賞口128を開放する開状態に変移する。ここで、可動片128bは、図3(b)に示すように、遊

技盤１０８の背面側に向けてスライドすることで、閉状態から開状態へと変移する。その結果、閉状態から開状態に変移する際に、可動片１２８ｂ上を転動している遊技球が、第２大入賞口１２８内に自重で落下する。

【００３３】

このように、同時回し参考例では、可動片１２８ｂを僅かに傾斜させ、可動片１２８ｂ上を遊技球が転動する時間を長く確保する。そして、可動片１２８ｂが閉状態から開状態に変移することで、可動片１２８ｂ上を転動している遊技球を第２大入賞口１２８内に導く。上記の構成により、可動片１２８ｂを開状態に維持する時間を僅かに設定したとしても、第２大入賞口１２８内に所定数の遊技球を導くことができる。換言すれば、第２大入賞口１２８内に所定数の遊技球を入球させるために必要となる、可動片１２８ｂを開状態に維持する時間を短時間とすることができる。なお、構造物１２９の背面には、遊技盤１０８の背面側に連通する孔が形成されており、第２大入賞口１２８に入球した遊技球は、遊技盤１０８の背面側に排出される。そして、第２大入賞口１２８に遊技球が入球すると、所定の賞球が遊技者に払い出される。

【００３４】

なお、ここでは、第２大入賞口１２８の構成について説明したが、第１可変始動口１２０Ｂも、第２大入賞口１２８と同様の構成である。すなわち、第１可変始動口１２０Ｂの可動片１２０ｂは、閉状態において、遊技盤１０８の前面側に突出しており、可動片１２０ｂ上を遊技球が転動する。そして、可動片１２０ｂの開状態では、可動片１２０ｂが遊技盤１０８の背面側にスライドし、第１可変始動口１２０Ｂへの遊技球の入球が可能となる。ただし、図２に示すように、可動片１２８ｂは、遊技機１００の正面視で右側が左側よりも高い位置にあるのに対して、可動片１２０ｂは、遊技機１００の正面視で左側が右側よりも高い位置にある。

【００３５】

ここで、第２遊技領域１１６ｂの盤面構成について詳述する。同時回し参考例では、第２遊技領域１１６ｂの最上部に、第２始動口１２２とゲート１２４とが並列して配置されている。第２遊技領域１１６ｂに導かれた遊技球は、全て、第２始動口１２２に入球するか、ゲート１２４を通過して下方に流下する。同時回し参考例では、第２始動口１２２は、１個の遊技球の入球に対して１個の遊技球が賞球として払い出される。

【００３６】

第２始動口１２２に遊技球が入球すると、賞球として１個の遊技球が払い出されるとともに、大役遊技あるいは小当たり遊技の実行有無等を決定する抽選が行われる。また、ゲート１２４を遊技球が通過すると、第１可変始動口１２０Ｂ（可動片１２０ｂ）を開状態とするか否かを決定する抽選が行われる。

【００３７】

第２始動口１２２およびゲート１２４の直下には、第１大入賞口１２６が設けられている。第１大入賞口１２６が開状態にある場合には、ゲート１２４を通過して下方に流下した遊技球の全てが第１大入賞口１２６に入球するように配置されている。同時回し参考例では、第１大入賞口１２６は、大役遊技においてのみ開放される。つまり、第１大入賞口１２６は、大役遊技専用の大入賞口と言える。大役遊技中に第１大入賞口１２６に遊技球が入球すると、１個の遊技球の入球に対して２個以上の所定数（ここでは１５個）の遊技球が賞球として払い出される。

【００３８】

第１大入賞口１２６の下方には第１可変始動口１２０Ｂが設けられている。また、第１大入賞口１２６と第１可変始動口１２０Ｂとの間にはアウト口１３１が設けられている。アウト口１３１は、遊技領域１１６から遊技球を排出するための通路の入口であり、アウト口１３１に遊技球が入球したとしても、賞球が払い出されることはない。

【００３９】

第２遊技領域１１６ｂには、第１大入賞口１２６よりも下方に流下した遊技球の多く（例えば９割以上）が、可動片１２０ｂ上に落下するように釘が配されている。また、これ

10

20

30

40

50

らの釘により、第1大入賞口126よりも下方に流下した遊技球の一部（例えば1%～10%）が、アウト口131に導かれる。

【0040】

第1可変始動口120Bの閉状態では、可動片120b上を遊技球が転動する。可動片120b上を遊技球が転動しているときに可動片120bが開状態になると、可動片120b上の遊技球は全て第1可変始動口120B内に導かれる。第1可変始動口120Bに遊技球が入球すると、1個の遊技球の入球に対して1個の遊技球が賞球として払い出されるとともに、大役遊技あるいは小当たり遊技の実行有無等を決定する抽選が行われる。

【0041】

第1可変始動口120Bに入球しなかった遊技球は、可動片120b上から遊技機100の正面視で右側に落下する。第1可変始動口120Bの下方には、第2大入賞口128が設けられており、可動片120b上から落下した遊技球の殆どが、第2大入賞口128の可動片128b上を転動する。可動片128b上の遊技球は、可動片128bの傾斜により、遊技機100の正面視で右側から左側に向けてゆっくりと転動する。

10

【0042】

詳しくは後述するが、同時回し参考例では、第2大入賞口128は、小当たり遊技においてのみ開放される。つまり、第2大入賞口128は、小当たり遊技専用の大入賞口と言える。可動片128b上を遊技球が転動しているときに可動片128bが開状態になると、可動片128b上の遊技球は全て第2大入賞口128内に導かれる。小当たり遊技中に第2大入賞口128に遊技球が入球すると、1個の遊技球の入球に対して2個以上の所定数（ここでは15個）の遊技球が賞球として払い出される。

20

【0043】

第2大入賞口128の左下方には、普図作動口125が設けられている。ここでは、第2大入賞口128上から下方に落下した遊技球の殆ど全てが普図作動口125に入球するように、第2遊技領域116bに釘が配されている。同時回し参考例では、普図作動口125は、1個の遊技球の入球に対して1個の遊技球が賞球として払い出される「特定の入賞口」を構成している。また、普図作動口125に遊技球が入球すると、ゲート124を遊技球が通過した場合と同様に、第1可変始動口120B（可動片120b）を開状態とするか否かを決定する抽選が行われる。

【0044】

なお、遊技領域116の最下部には、一般入賞口118、第1始動口120、第2始動口122、普図作動口125、および、大入賞口のいずれにも入球しなかった遊技球を、遊技領域116から遊技盤108の背面側に排出する排出口130が設けられている。

30

【0045】

そして、遊技機100には、遊技の進行中等に演出を行う演出装置として、液晶表示装置からなる演出表示装置200、可動装置からなる演出役物装置202、さまざまな点灯態様や発光色に制御されるランプからなる演出照明装置204、スピーカからなる音声出力装置206、遊技者の操作を受け付ける演出操作装置208が設けられている。

【0046】

演出表示装置200は、画像を表示する画像表示部からなるメイン演出表示部200aを備えており、このメイン演出表示部200aを、遊技盤108の略中央部分において、遊技機100の正面側から視認可能に配置している。このメイン演出表示部200aには、図示のように、3つの演出図柄210a、210b、210cが変動表示される等、種々の演出が実行されることとなる。

40

【0047】

演出役物装置202は、メイン演出表示部200aよりも前面に配置され、通常、遊技盤108の背面側の原点位置において、複数の構成部材に分割された状態で退避しており、遊技者が視認できないようになっている。そして、上記の演出図柄210a、210b、210cの変動表示中などに、アクチュエータの駆動により、メイン演出表示部200aの前面にある可動位置まで各構成部材が移動すると、メイン演出表示部200aの前面

50

で各構成部材が合体して、遊技者に大当たりの期待感を付与する。

【0048】

演出照明装置204は、演出役物装置202や遊技盤108等に設けられており、メイン演出表示部200aに表示される画像等に合わせて、さまざまに点灯制御される。

【0049】

音声出力装置206は、前枠106の上部位置や外枠102の最下部位置に設けられ、メイン演出表示部200aに表示される画像等に合わせて、遊技機100の正面側に向けてさまざまな音声を出力する。

【0050】

演出操作装置208は、遊技者の押下操作を受け付けるボタンで構成され、遊技機100の幅方向略中央位置であって、かつ、透過板110よりも下方位置に設けられている。この演出操作装置208は、メイン演出表示部200aに表示される画像等に合わせて有効化されるものであり、操作有効時間内に遊技者の操作を受け付けると、当該操作に応じて、さまざまな演出が実行される。

【0051】

なお、図中符号132は、遊技機100から払い出される賞球や、遊技球貸出装置から貸し出される遊技球が導かれる上皿であり、この上皿132が遊技球で一杯になると、遊技球は下皿134に導かれることとなる。また、この下皿134の底面には、当該下皿134から遊技球を排出するための球抜き孔（不図示）が形成されている。この球抜き孔は、通常、開閉板（不図示）によって閉じられているが、球抜きつまみ134aを押下することにより、当該球抜きつまみ134aと一体となって開閉板がスライドし、球抜き孔から下皿134の下方に遊技球を排出することが可能となっている。

【0052】

また、遊技盤108には、遊技領域116の外方であって、かつ、遊技者が視認可能な位置に、第1特別図柄表示器160、第2特別図柄表示器162、第1特別図柄保留表示器164、第2特別図柄保留表示器166、普通図柄表示器168、普通図柄保留表示器170、右打ち報知表示器172が設けられている。これら各表示器160～172は、遊技に係る種々の状況を表示するための装置であるが、その詳細については後述する。

【0053】

（制御手段の内部構成）

図4は、同時回し参考例に係る遊技の進行を制御する制御手段の内部構成を示すブロック図である。

【0054】

主制御基板300は遊技の基本動作を制御する。この主制御基板300は、メインCPU300a、メインROM300b、メインRAM300cを備えている。メインCPU300aは、各検出スイッチやタイマからの入力信号に基づいて、メインROM300bに格納されたプログラムを読み出して演算処理を行うとともに、各装置や表示器を直接制御したり、あるいは演算処理の結果に応じて他の基板にコマンドを送信したりする。メインRAM300cは、メインCPU300aの演算処理時におけるデータのワークエリアとして機能する。

【0055】

上記主制御基板300には、一般入賞口118に遊技球が入球したことを検出する一般入賞口検出スイッチ118s、第1固定始動口120Aに遊技球が入球したことを検出する第1固定始動口検出スイッチ120As、第1可変始動口120Bに遊技球が入球したことを検出する第1可変始動口検出スイッチ120Bs、第2始動口122に遊技球が入球したことを検出する第2始動口検出スイッチ122s、ゲート124を遊技球が通過したことを検出するゲート検出スイッチ124s、普図作動口125に遊技球が入球したことを検出する普図作動口検出スイッチ125s、第1大入賞口126に遊技球が入球したことを検出する第1大入賞口検出スイッチ126s、第2大入賞口128に遊技球が入球したことを検出する第2大入賞口検出スイッチ128sが接続されており、これら各検出

スイッチから主制御基板 300 に検出信号が入力されるようになっている。

【0056】

また、主制御基板 300 には、第 1 可変始動口 120 B の可動片 120 b を作動する普通電動役物ソレノイド 120 c と、第 1 大入賞口 126 を開閉する開閉扉 126 b を作動する第 1 大入賞口ソレノイド 126 c と、第 2 大入賞口 128 を開閉する可動片 128 b を作動する第 2 大入賞口ソレノイド 128 c と、が接続されており、主制御基板 300 によって、第 1 可変始動口 120 B、第 1 大入賞口 126、第 2 大入賞口 128 の開閉制御がなされるようになっている。

【0057】

さらに、主制御基板 300 には、第 1 特別図柄表示器 160、第 2 特別図柄表示器 162、第 1 特別図柄保留表示器 164、第 2 特別図柄保留表示器 166、普通図柄表示器 168、普通図柄保留表示器 170、右打ち報知表示器 172 が接続されており、主制御基板 300 によって、これら各表示器の表示制御がなされるようになっている。

10

【0058】

さらに、遊技盤 108 の背面には、設定変更スイッチ 180 s が設けられている。設定変更スイッチ 180 s は、専用の鍵によってアクセス可能に構成されている。設定変更スイッチ 180 s がオンしていることを条件として、設定値を変更、確認する操作が可能となる。詳しくは後述するが、同時回し参考例の遊技機 100 は、有利度合いが異なる 6 段階の設定値のいずれかが設定値バッファに登録設定値として記憶され、記憶されている登録設定値に応じて遊技が進行する。なお、ここでは、設定値が 6 段階であることとするが、設定値は高設定および低設定の 2 段階のみが設けられてもよいし、他の複数段階で設けられてもよい。さらには、設定値は必須ではなく、有利度合いが変更されなくてもよい。

20

【0059】

また、遊技盤 108 の背面には、RAM クリアボタンが押下操作可能に設けられており、この RAM クリアボタンの押下操作が RAM クリアスイッチ 182 s によって検出される。RAM クリアスイッチ 182 s は主制御基板 300 に接続されており、RAM クリアスイッチ 182 s から主制御基板 300 に RAM クリア操作信号が入力される。電源投入時に RAM クリアスイッチ 182 s から RAM クリア操作信号が入力されている場合、メイン CPU 300 a は、メイン RAM 300 c をクリアする。

【0060】

また、遊技盤 108 の背面には、性能表示モニタ 184 が設けられている。主制御基板 300 により、性能表示モニタ 184 に登録設定値やベース比率が表示される。

30

【0061】

また、同時回し参考例の遊技機 100 は、主に第 1 始動口 120 または第 2 始動口 122 への遊技球の入球によって開始される特別遊技と、ゲート 124 への遊技球の通過、あるいは、普通図作動口 125 への遊技球の入球によって開始される普通遊技とに大別される。そして、主制御基板 300 のメイン ROM 300 b には、特別遊技および普通遊技を進行するための種々のプログラムや、各種の遊技に必要なデータ、テーブルが記憶されている。

【0062】

また、主制御基板 300 には、払出制御基板 310 および副制御基板 330 が接続されている。払出制御基板 310 は、遊技球を発射させるための制御、および、賞球を払い出すための制御を行う。この払出制御基板 310 も、CPU、ROM、RAM を備えており、主制御基板 300 に対して双方向に通信可能に接続されている。この払出制御基板 310 には遊技情報出力端子板 312 が接続されており、主制御基板 300 から出力される遊技進行上の種々の情報が、払出制御基板 310 および遊技情報出力端子板 312 を介して、遊技店のホールコンピュータ等に出力されることとなる。

40

【0063】

また、払出制御基板 310 には、貯留部に貯留された遊技球を賞球として遊技者に払い出すための払出モータ 314 が接続されている。払出制御基板 310 は、主制御基板 300

50

0 から送信された払出個数指定コマンドに基づいて払出モータ 3 1 4 を制御して所定の賞球を遊技者に払い出すように制御する。このとき、払い出された遊技球数が払出球計数スイッチ 3 1 6 s によって検出され、払い出すべき賞球が遊技者に払い出されたかが把握されるようになっている。

【0064】

また、払出制御基板 3 1 0 には、下皿 1 3 4 の満タン状態を検出する皿満タン検出スイッチ 3 1 8 s が接続されている。この皿満タン検出スイッチ 3 1 8 s は、賞球として払い出される遊技球を下皿 1 3 4 に導く通路に設けられており、遊技球検出信号が払出制御基板 3 1 0 に入力されるようになっている。

【0065】

そして、下皿 1 3 4 に所定量以上の遊技球が貯留されて満タン状態になると、下皿 1 3 4 に向かう通路内に遊技球が滞留し、皿満タン検出スイッチ 3 1 8 s から払出制御基板 3 1 0 に向けて、遊技球検出信号が連続的に入力される。払出制御基板 3 1 0 は、遊技球検出信号が所定時間連続して入力された場合に、下皿 1 3 4 が満タン状態であると判断し、皿満タンコマンドを主制御基板 3 0 0 に送信する。一方、皿満タンコマンドを送信した後、遊技球検出信号の連続入力が途絶えた場合には、満タン状態が解除されたと判断し、皿満タン解除コマンドを主制御基板 3 0 0 に送信する。

【0066】

また、払出制御基板 3 1 0 には、遊技球の発射制御を行う発射制御回路 3 2 0 が設けられている。払出制御基板 3 1 0 には、操作ハンドル 1 1 2 に設けられ、当該操作ハンドル 1 1 2 に遊技者が触れたことを検出するタッチセンサ 1 1 2 s と、操作ハンドル 1 1 2 の操作角度を検出する操作ボリューム 1 1 2 a と、が接続されている。そして、タッチセンサ 1 1 2 s および操作ボリューム 1 1 2 a から信号が入力されると、発射制御回路 3 2 0 において、遊技球発射装置に設けられた発射用ソレノイド 1 1 2 c を通電して遊技球を発射させる制御がなされる。

【0067】

副制御基板 3 3 0 は、主に遊技中や待機中等の各演出を制御する。この副制御基板 3 3 0 は、サブ CPU 3 3 0 a、サブ ROM 3 3 0 b、サブ RAM 3 3 0 c を備えており、主制御基板 3 0 0 に対して、当該主制御基板 3 0 0 から副制御基板 3 3 0 への一方向に通信可能に接続されている。サブ CPU 3 3 0 a は、主制御基板 3 0 0 から送信されたコマンドやタイマからの入力信号等に基づいて、サブ ROM 3 3 0 b に格納されたプログラムを読み出して演算処理を行うとともに、演出を実行制御する。このとき、サブ RAM 3 3 0 c は、サブ CPU 3 3 0 a の演算処理時におけるデータのワークエリアとして機能する。

【0068】

具体的には、副制御基板 3 3 0 では、サブ CPU 3 3 0 a、サブ ROM 3 3 0 b、サブ RAM 3 3 0 c が協働して、サブメイン、画像制御部、役物制御部、照明制御部、音声制御部として機能する。サブメインは、各種入力コマンドに応じて、実行する演出の内容を決定したり、演出の実行を管理、統括したりする。画像制御部は、上記メイン演出表示部 2 0 0 a に画像を表示させる画像表示制御を行う。サブ ROM 3 3 0 b には、メイン演出表示部 2 0 0 a に表示される図柄や背景、字幕等の画像データが多数格納されており、画像制御部が、画像データをサブ ROM 3 3 0 b から不図示の VRAM に読み出して、メイン演出表示部 2 0 0 a の画像表示を制御する。

【0069】

また、役物制御部は、サブメインによる演出の管理にしたがってアクチュエータを駆動し、演出役物装置 2 0 2 を可動制御する。照明制御部は演出照明装置 2 0 4 を点灯制御する。また、音声制御部は、上記音声出力装置 2 0 6 から音声を出力させる音声出力制御を行う。サブ ROM 3 3 0 b には、音声出力装置 2 0 6 から出力される音声や楽曲等の音声データが多数格納されており、音声制御部が、音声データをサブ ROM 3 3 0 b から読み出して、音声出力装置 2 0 6 の音声出力を制御する。

【0070】

さらに、副制御基板 330 では、演出操作装置 208 が押下操作または回転操作されたことを検出する演出操作装置検出スイッチ 208 s から操作検出信号が入力された際に、所定の演出を実行する。

【0071】

なお、各基板には、不図示の電源基板が接続されており、電源基板を介して商用電源から各基板に電力供給がなされている。また、電源基板にはコンデンサからなるバックアップ電源が設けられている。

【0072】

図 5 は、同時回し参考例に係るメイン CPU 300 a が用いるメモリ領域のアドレスマップである。なお、図 5 において、アドレスは 16 進数で示しており、「H」は 16 進数であることを示している。図 5 に示すように、メイン CPU 300 a が用いるメモリ領域は、メイン ROM 300 b に割り当てられたメモリ領域（0000H～2FFFFH）と、メイン RAM 300 c に割り当てられたメモリ領域（F000H～F3FFFH）とを含んでいる。

【0073】

メイン ROM 300 b のメモリ領域は、遊技の進行を制御するためのプログラムおよびデータを格納する使用領域（0000H～1BFFH）と、使用領域以外の領域であって、遊技機規則で定める試験を行うための処理や、性能表示モニタ 184 を表示するための処理（性能表示モニタ 184 に表示するベース比率を算出するための処理を含む）を実行するためのプログラムおよびデータを格納する使用外領域（2000H～2BFFFH）とが設けられている。

【0074】

メイン ROM 300 b の使用領域には、遊技の進行を制御するためのプログラムが格納されるプログラム領域（0000H～0A89H）、未使用領域（0A8AH～11FFFH）、プログラム以外のデータが格納されるデータ領域（1200H～1BFFH）が設けられている。なお、使用領域は、未使用領域（0A8AH～0FFFFH）を含めないようにしてもよい。

【0075】

メイン ROM 300 b の使用外領域には、遊技機規則で定める試験を行うための処理や、性能表示モニタ 184 を表示するための処理を実行するためのプログラムが格納されるプログラム領域（2000H～27FFFH）、これらのプログラム以外のデータが格納されるデータ領域（2800H～2BFFFH）が設けられている。

【0076】

また、メイン ROM 300 b のメモリ領域には、使用領域および使用外領域以外にも、未使用領域（1A7BH～1DFFFH）、プログラムのタイトル、バージョン等の任意のデータが格納される ROM コメント領域（1E00H～1EFFFH）、未使用領域（1F00H～1FFFFH）、未使用領域（2C00H～2FBFFFH）、メイン CPU 300 a がプログラムを実行するために必要な情報が格納されるプログラム管理領域（2FC0H～2FFFFH）が設けられている。

【0077】

メイン RAM 300 c のメモリ領域は、遊技の進行を制御するためのプログラムが実行されている際に一時的に用いられる使用領域（F000H～F1FFFH）と、使用領域以外の領域であって、遊技機規則で定める試験を行うための処理や、性能表示モニタ 184 を表示するための処理のプログラムが実行されている際に一時的に用いられる使用外領域（F210H～F228H）とが設けられている。

【0078】

メイン RAM 300 c の使用領域には、遊技の進行を制御するためのプログラムが実行されている際に一時的に用いられるワーク領域（F000H～F12AH）、未使用領域（F12BH～F1D7H）、遊技の進行を制御するためのプログラムの実行中にデータを一時的に退避させるスタック領域（F1D8H～F1FFFH）が設けられている。なお

10

20

30

40

50

、使用領域は、未使用領域（F 1 2 B H～F 1 D 7 H）を含めないようにしてもよい。

【0079】

メインRAM 300cの使用外領域には、遊技機規則で定める試験を行うための処理や、性能表示モニタ184を表示するための処理のプログラムが実行されている際に一時的に用いられるワーク領域（F 2 1 0 H～F 2 1 F H）、これらのプログラムが実行されている際にデータを一時的に退避させるスタック領域（F 2 2 0 H～F 2 2 8 H）が設けられている。

【0080】

また、メインRAM 300cのメモリ領域には、使用領域および使用外領域以外にも、未使用領域（F 2 0 0 H～F 2 0 F H）、未使用領域（F 2 2 9 H～F 3 F F H）が設けられている。

10

【0081】

このように、メインROM 300bおよびメインRAM 300cでは、遊技の進行を制御するために用いられる使用領域と、遊技機規則で定める試験を行うための処理や、性能表示モニタ184の表示制御をするための処理を実行するために用いられる使用外領域とが分かれて設けられている。

【0082】

そして、メインRAM 300cでは、使用領域と使用外領域との間に、16バイトの未使用領域（F 2 0 0 H～F 2 0 F H）が設けられている。この未使用領域（F 2 0 0 H～F 2 0 F H）は、使用領域および使用外領域を分ける境界領域として設定されており、使用領域と使用外領域との境界が明確となり、遊技の進行を制御するためのプログラムが実行されている際に使用外領域が用いられること、および、遊技機規則で定める試験を行うための処理や、性能表示モニタ184の表示制御をするための処理のプログラムが実行されている際に使用領域が用いられることを防止している。

20

【0083】

なお、使用領域と使用外領域との間に設けられる未使用領域は、少なくとも1バイト以上であればよく、不正防止の観点から、4バイト以上であることが望ましく、16バイト以上に設定されることがより望ましい。また、未使用領域は、データの書き込みおよび読み出しが禁止されているが、不正防止の観点から、所定のタイミングでクリアするようにしてもよい。

30

【0084】

また、F E 0 0 h～F E F F hのメモリ空間には入出力部が割り当てられている。かかる入出力部については後程詳述する。

【0085】

次に、同時回し参考例の遊技機100における遊技について、メインROM 300bに記憶されている各種テーブルと併せて説明する。

【0086】

前述したように、同時回し参考例の遊技機100は、特別遊技と普通遊技の2種類の遊技が並行して進行するものである。特別遊技は、低確率遊技状態および高確率遊技状態のいずれかの遊技状態にて遊技が進行し、普通遊技は、非時短遊技状態、中時短遊技状態、時短遊技状態のいずれかの遊技状態にて遊技が進行する。

40

【0087】

各遊技状態の詳細については後述するが、低確率遊技状態というのは、大入賞口が開放される大役遊技を実行する権利獲得の確率が低く設定された遊技状態であり、高確率遊技状態というのは、大役遊技を実行する権利獲得の確率が高く設定された遊技状態である。また、非時短遊技状態というのは、可動片120bが開状態になりにくく、第1可変始動口120Bに遊技球が入球し難い遊技状態であり、中時短遊技状態というのは、非時短遊技状態よりも可動片120bが開状態になりやすく、第1可変始動口120Bに遊技球が入球しやすい遊技状態である。また、時短遊技状態というのは、中時短遊技状態よりも、さらに、可動片120bが開状態になりやすく、第1可変始動口120Bに遊技球が最も

50

入球しやすい遊技状態である。なお、時短遊技状態では、遊技中に第2遊技領域116bに向けて遊技球を発射し続けている場合に、僅かに遊技球が減少するか、遊技球が殆ど減少しないように設定されている。

【0088】

上記のように、特別遊技と普通遊技とは同時並行して進行することから、同時回し参考例では、低確率遊技状態または高確率遊技状態と、非時短遊技状態、中時短遊技状態、時短遊技状態のいずれかが組み合わされた遊技状態となる。以下では、理解を容易とするため、特別遊技に係る遊技状態、すなわち、低確率遊技状態および高確率遊技状態を特別遊技状態と呼び、普通遊技に係る遊技状態、すなわち、非時短遊技状態、中時短遊技状態、時短遊技状態を普通遊技状態と呼ぶ場合がある。遊技機100の初期状態は、低確率遊技状態および非時短遊技状態に設定されている。

10

【0089】

遊技者が操作ハンドル112を操作して遊技領域116に遊技球を発射させるとともに、遊技領域116を流下する遊技球が第1始動口120または第2始動口122に入球すると、遊技者に遊技利益を付与するか否かの抽選（以下、「大役抽選」という）が行われる。この大役抽選において、大当たりに当選すると、大入賞口が開放されるとともに当該大入賞口への遊技球の入球が可能となる大役遊技が実行され、また、当該大役遊技の終了後の遊技状態が、上記のいずれかの遊技状態に設定される。以下では、大役抽選方法について説明する。

【0090】

20

なお、詳しくは後述するが、第1始動口120または第2始動口122に遊技球が入球すると、大役抽選に係る種々の乱数値（大当たり決定乱数、当たり図柄乱数、リーチグループ決定乱数、リーチモード決定乱数、変動パターン乱数）が取得されるとともに、これら各乱数値がメインRAM300cの特図保留記憶領域に記憶される。以下では、第1始動口120に遊技球が入球して特図保留記憶領域に記憶された種々の乱数を総称して特1保留とよび、第2始動口122に遊技球が入球して特図保留記憶領域に記憶された種々の乱数を総称して特2保留とよぶ。

【0091】

メインRAM300cの特図保留記憶領域は、第1特図保留記憶領域と第2特図保留記憶領域とを備えている。第1特図保留記憶領域および第2特図保留記憶領域は、それぞれ4つの記憶部（第1～第4記憶部）を有している。そして、第1始動口120に遊技球が入球すると、特1保留を第1特図保留記憶領域の第1記憶部から順に記憶し、第2始動口122に遊技球が入球すると、特2保留を第2特図保留記憶領域の第1記憶部から順に記憶する。

30

【0092】

例えば、第1始動口120に遊技球が入球したとき、第1特図保留記憶領域の第1～第4記憶部のいずれにも保留が記憶されていない場合には、第1記憶部に特1保留を記憶する。また、例えば、第1記憶部～第3記憶部に特1保留が記憶されている状態で、第1始動口120に遊技球が入球した場合には、特1保留を第4記憶部に記憶する。また、第2始動口122に遊技球が入球した場合にも、上記と同様に、第2特図保留記憶領域の第1記憶部～第4記憶部の中で、特2保留が記憶されていない、最も番号（序数）の小さい記憶部に特2保留が記憶される。

40

【0093】

ただし、第1特図保留記憶領域および第2特図保留記憶領域に記憶可能な特1保留数（X1）および特2保留数（X2）は、それぞれ4つに設定されている。したがって、例えば、第1始動口120に遊技球が入球したときに、第1特図保留記憶領域に既に4つの特1保留が記憶されている場合には、当該第1始動口120への遊技球の入球によって新たに特1保留が記憶されることはない。同様に、第2始動口122に遊技球が入球したときに、第2特図保留記憶領域に既に4つの特2保留が記憶されている場合には、当該第2始動口122への遊技球の入球によって新たに特2保留が記憶されることはない。

50

【0094】

図6は、同時回し参考例に係る低確時大当たり決定乱数判定テーブルを説明する図である。第1始動口120または第2始動口122に遊技球が入球すると、0～65535の範囲内から1つの大当たり決定乱数が取得される。そして、大役抽選を開始するとき、すなわち、大当たりの判定を行うときの遊技状態に応じて大当たり決定乱数判定テーブルが選択され、当該選択された大当たり決定乱数判定テーブルと取得された大当たり決定乱数とによって大役抽選が行われる。

【0095】

低確率遊技状態において、特1保留および特2保留について大役抽選を開始する場合には、低確時大当たり決定乱数判定テーブルが参照される。ここで、同時回し参考例では、有利度合いを異にする6段階の設定値が設けられており、低確時大当たり決定乱数判定テーブルは、設定値ごとに設けられている。遊技中は、設定値が6段階のうちのいずれかに設定されており、現在設定されている設定値（設定値バッファに記憶されている登録設定値）に対応する低確時大当たり決定乱数判定テーブルを参照して大役抽選が行われる。

【0096】

低確率遊技状態であって、設定値＝1に設定されている場合（登録設定値＝1）には、図6（a）に示す低確時大当たり決定乱数判定テーブルaを参照して大役抽選が行われる。この低確時大当たり決定乱数判定テーブルaによれば、大当たり決定乱数が10001～10218であった場合に大当たりと判定し、大当たり決定乱数が20001～38996であった場合に小当たりと判定し、その他の大当たり決定乱数であった場合にはハズレと判定する。したがって、この場合の大当たり確率は約 $1/300.6$ となり、小当たり確率は約 $1/3.45$ となる。

【0097】

低確率遊技状態であって、設定値＝2に設定されている場合（登録設定値＝2）には、図6（b）に示す低確時大当たり決定乱数判定テーブルbを参照して大役抽選が行われる。この低確時大当たり決定乱数判定テーブルbによれば、大当たり決定乱数が10001～10225であった場合に大当たりと判定し、大当たり決定乱数が20001～38996であった場合に小当たりと判定し、その他の大当たり決定乱数であった場合にはハズレと判定する。したがって、この場合の大当たり確率は約 $1/291.2$ となり、小当たり確率は約 $1/3.45$ となる。

【0098】

低確率遊技状態であって、設定値＝3に設定されている場合（登録設定値＝3）には、図6（c）に示す低確時大当たり決定乱数判定テーブルcを参照して大役抽選が行われる。この低確時大当たり決定乱数判定テーブルcによれば、大当たり決定乱数が10001～10232であった場合に大当たりと判定し、大当たり決定乱数が20001～38996であった場合に小当たりと判定し、その他の大当たり決定乱数であった場合にはハズレと判定する。したがって、この場合の大当たり確率は約 $1/282.4$ となり、小当たり確率は約 $1/3.45$ となる。

【0099】

低確率遊技状態であって、設定値＝4に設定されている場合（登録設定値＝4）には、図6（d）に示す低確時大当たり決定乱数判定テーブルdを参照して大役抽選が行われる。この低確時大当たり決定乱数判定テーブルdによれば、大当たり決定乱数が10001～10239であった場合に大当たりと判定し、大当たり決定乱数が20001～38996であった場合に小当たりと判定し、その他の大当たり決定乱数であった場合にはハズレと判定する。したがって、この場合の大当たり確率は約 $1/274.2$ となり、小当たり確率は約 $1/3.45$ となる。

【0100】

低確率遊技状態であって、設定値＝5に設定されている場合（登録設定値＝5）には、図6（e）に示す低確時大当たり決定乱数判定テーブルeを参照して大役抽選が行われる。この低確時大当たり決定乱数判定テーブルeによれば、大当たり決定乱数が10001

～10246であった場合に大当たりと判定し、大当たり決定乱数が20001～38996であった場合に小当たりと判定し、その他の大当たり決定乱数であった場合にはハズレと判定する。したがって、この場合の大当たり確率は約 $1/266.4$ となり、小当たり確率は約 $1/3.45$ となる。

【0101】

低確率遊技状態であって、設定値＝6に設定されている場合（登録設定値＝6）には、図6（f）に示す低確時大当たり決定乱数判定テーブルfを参照して大役抽選が行われる。この低確時大当たり決定乱数判定テーブルfによれば、大当たり決定乱数が10001～10253であった場合に大当たりと判定し、大当たり決定乱数が20001～38996であった場合に小当たりと判定し、その他の大当たり決定乱数であった場合にはハズレと判定する。したがって、この場合の大当たり確率は約 $1/259.0$ となり、小当たり確率は約 $1/3.45$ となる。

10

【0102】

図7は、同時回し参考例に係る高確時大当たり決定乱数判定テーブルを説明する図である。高確率遊技状態において、特1保留および特2保留について大役抽選を開始する場合には、高確時大当たり決定乱数判定テーブルが参照される。高確時大当たり決定乱数判定テーブルも、低確時大当たり決定乱数判定テーブルと同様に設定値ごとに設けられている。

【0103】

高確率遊技状態であって、設定値＝1に設定されている場合（登録設定値＝1）には、図7（a）に示す高確時大当たり決定乱数判定テーブルaを参照して大役抽選が行われる。この高確時大当たり決定乱数判定テーブルaによれば、大当たり決定乱数が10001～10620であった場合に大当たりと判定し、大当たり決定乱数が20001～38996であった場合に小当たりと判定し、その他の大当たり決定乱数であった場合にはハズレと判定する。したがって、この場合の大当たり確率は約 $1/105.7$ となり、小当たり確率は約 $1/3.45$ となる。

20

【0104】

同様に、高確率遊技状態であって、設定値＝2～6に設定されている場合（登録設定値＝2～6）には、図7（b）～（f）に示す高確時大当たり決定乱数判定テーブルb～fを参照して大役抽選が行われる。これら高確時大当たり決定乱数判定テーブルb～fによれば、それぞれ大当たり決定乱数が図示の値であった場合に大当たりと判定する。したがって、設定値＝2～6の場合の大当たり確率は、それぞれ約 $1/102.4$ ～ $1/91.0$ となり、小当たり確率は約 $1/3.45$ となる。

30

【0105】

以上のように、大役抽選は、登録設定値に応じて行われる。このとき、登録設定値に応じて大当たりの当選確率が異なっており、登録設定値が大きい場合の方が、小さい場合に比べて、大当たりに当選しやすくなっている。なお、ここでは、登録設定値が異なっても、小当たりの当選確率は変わらないこととしたが、登録設定値ごとに小当たりの当選確率を異ならせてもよい。また、小当たりは必須ではなく、大役抽選において、大当たりおよびハズレのいずれかのみが決定されてもよい。

【0106】

また、ここでは、低確率遊技状態および高確率遊技状態の双方における大当たりの当選確率が、登録設定値に応じて異なることとしたが、低確率遊技状態および高確率遊技状態のいずれか一方における大当たりの当選確率のみが、登録設定値に応じて異なることとしてもよい。

40

【0107】

図8は、同時回し参考例に係る当たり図柄乱数判定テーブルおよび小当たり図柄乱数判定テーブルを説明する図である。第1始動口120または第2始動口122に遊技球が入球すると、0～99の範囲内から1つの当たり図柄乱数が取得される。そして、上記の大役抽選により「大当たり」の判定結果が導出された場合に、取得している当たり図柄乱数と当たり図柄乱数判定テーブルとによって、特別図柄の種別が決定される。このとき、特

50

1 保留によって「大当たり」に当選した場合には、図 8 (a) に示すように、特 1 用当たり図柄乱数判定テーブル a が選択され、特 2 保留によって「大当たり」に当選した場合には、図 8 (b) に示すように、特 2 用当たり図柄乱数判定テーブル b が選択され、特 1 保留によって「小当たり」に当選した場合には、図 8 (c) に示すように、特 1 用小当たり図柄乱数判定テーブル a が選択され、特 2 保留によって「小当たり」に当選した場合には、図 8 (d) に示すように、特 2 用小当たり図柄乱数判定テーブル b が選択される。以下では、当たり図柄乱数によって決定される特別図柄、すなわち、大当たりの判定結果が得られた場合に決定される特別図柄を大当たり図柄と呼び、小当たりの判定結果が得られた場合に決定される特別図柄を小当たり図柄と呼び、ハズレの判定結果が得られた場合に決定される特別図柄をハズレ図柄と呼ぶ。

10

【0108】

図 8 (a) に示す特 1 用当たり図柄乱数判定テーブル a、および、図 8 (b) に示す特 2 用当たり図柄乱数判定テーブル b によれば、取得した当たり図柄乱数の値に応じて、図示のとおり、特別図柄として大当たり図柄（特別図柄 A～J）が決定される。また、図 8 (c) に示す特 1 用小当たり図柄乱数判定テーブル a、および、図 8 (d) に示す特 2 用小当たり図柄乱数判定テーブル b によれば、取得した当たり図柄乱数の値に応じて、図示のとおり、特別図柄として小当たり図柄（特別図柄 Z 1～Z 6）が決定される。

【0109】

なお、大役抽選結果が「ハズレ」であった場合に、当該抽選結果が特 1 保留によって導出されたときは、抽選を行うことなくハズレ図柄として特別図柄 X が決定され、当該抽選結果が特 2 保留によって導出されたときは、抽選を行うことなくハズレ図柄として特別図柄 Y が決定される。つまり、当たり図柄乱数判定テーブルは、大役抽選結果が「大当たり」であった場合にのみ参照され、大役抽選結果が「ハズレ」または「小当たり」であった場合に参照されることはない。また、小当たり図柄乱数判定テーブルは、大役抽選結果が「小当たり」であった場合にのみ参照され、大役抽選結果が「大当たり」または「ハズレ」であった場合に参照されることはない。なお、小当たり図柄である特別図柄 Z 1～Z 6 をまとめて単に特別図柄 Z ともよぶ。

20

【0110】

図 9 は、同時回し参考例に係るリーチグループ決定乱数判定テーブルを説明する図である。このリーチグループ決定乱数判定テーブルは複数設けられており、保留種別、保留数、遊技状態等に応じて、予め設定されたテーブルが選択される。第 1 始動口 1 2 0 または第 2 始動口 1 2 2 に遊技球が入球すると、0～1 0 0 0 6 の範囲内から 1 つのリーチグループ決定乱数が取得される。上記のように、大役抽選結果が導出されると、当該大役抽選結果を報知する変動演出パターンを決定する処理が行われる。同時回し参考例では、大役抽選結果が「ハズレ」であった場合に、変動演出パターンを決定するにあたって、まず、リーチグループ決定乱数とリーチグループ決定乱数判定テーブルとによってグループ種別が決定される。

30

【0111】

例えば、遊技状態が詳しくは後述する通常状態に設定されているときに、特 1 保留に基づいて「ハズレ」の大役抽選結果が導出された場合において、大役抽選を行うときの特 1 保留の保留数（以下、単に「保留数」という）が 0 個であれば、図 9 (a) に示すように、リーチグループ決定乱数判定テーブル 1 が選択される。同様に、通常状態に設定されているときに、特 1 保留に基づいて「ハズレ」の大役抽選結果が導出された場合において、大役抽選を行うときの保留数が 1～2 個であれば、図 9 (b) に示すように、リーチグループ決定乱数判定テーブル 2 が選択され、保留数が 3 個であれば、図 9 (c) に示すように、リーチグループ決定乱数判定テーブル 3 が選択される。なお、図 9 において、グループ種別の欄に記載しているグループ x は、任意のグループ番号を示している。したがって、取得したリーチグループ決定乱数と、参照するリーチグループ決定乱数判定テーブルの種類とに応じて、グループ種別として種々のグループ番号が決定されることとなる。

40

【0112】

50

なお、ここでは、通常状態において、特1保留に基づいて「ハズレ」の大役抽選結果が導出されたときに参照されるリーチグループ決定乱数判定テーブルについて説明したが、メインROM 300bには、この他にも多数のリーチグループ決定乱数判定テーブルが記憶されている。

【0113】

なお、大役抽選結果が「大当たり」または「小当たり」であった場合には、変動演出パターンを決定するにあたってグループ種別を決定することはない。つまり、リーチグループ決定乱数判定テーブルは、大役抽選結果が「ハズレ」であった場合にのみ参照され、大役抽選結果が「大当たり」または「小当たり」であった場合に参照されることはない。

【0114】

図10は、同時回し参考例に係るリーチモード決定乱数判定テーブルを説明する図である。このリーチモード決定乱数判定テーブルは、大役抽選結果が「ハズレ」であった場合に選択されるハズレ時リーチモード決定乱数判定テーブルと、大役抽選結果が「大当たり」であった場合に選択される大当たり時リーチモード決定乱数判定テーブルと、大役抽選結果が「小当たり」であった場合に選択される小当たり時リーチモード決定乱数判定テーブルとに大別される。なお、ハズレ時リーチモード決定乱数判定テーブルは、上記のように決定されたグループ種別ごとに設けられており、大当たり時リーチモード決定乱数判定テーブルおよび小当たり時リーチモード決定乱数判定テーブルは、遊技状態、保留種別ごとに設けられている。

【0115】

また、各リーチモード決定乱数判定テーブルは、遊技状態や図柄の種別ごとにも設けられている。ここでは、所定の遊技状態および図柄種別において参照されるグループx用ハズレ時リーチモード決定乱数判定テーブルの一例を図10(a)に示し、特1用大当たり時リーチモード決定乱数判定テーブルの一例を図10(b)に示し、特2用大当たり時リーチモード決定乱数判定テーブルの一例を図10(c)に示し、特1用小当たり時リーチモード決定乱数判定テーブルの一例を図10(d)に示し、特2用小当たり時リーチモード決定乱数判定テーブルの一例を図10(e)に示す。

【0116】

第1始動口120または第2始動口122に遊技球が入球すると、0~250の範囲内から1つのリーチモード決定乱数が取得される。そして、上記の大役抽選の結果が「ハズレ」であった場合には、図10(a)に示すように、上記のグループ種別の抽選により決定されたグループ種別に対応するハズレ時リーチモード決定乱数判定テーブルが選択され、選択されたハズレ時リーチモード決定乱数判定テーブルとリーチモード決定乱数とに基づいて、変動モード番号が決定される。また、上記の大役抽選の結果が「大当たり」であった場合には、図10(b)、(c)に示すように、大当たり当選時の遊技状態、および、読み出された保留種別に対応する大当たり時リーチモード決定乱数判定テーブルが選択され、選択された大当たり時リーチモード決定乱数判定テーブルとリーチモード決定乱数とに基づいて、変動モード番号が決定される。

【0117】

さらに、上記の大役抽選の結果が「小当たり」であった場合には、図10(d)、(e)に示すように、小当たり当選時の遊技状態、および、読み出された保留種別に対応する小当たり時リーチモード決定乱数判定テーブルが選択され、選択された小当たり時リーチモード決定乱数判定テーブルとリーチモード決定乱数とに基づいて、変動モード番号が決定される。

【0118】

また、各リーチモード決定乱数判定テーブルにおいては、リーチモード決定乱数に、変動モード番号とともに、後述する変動パターン乱数判定テーブルが対応付けられており、変動モード番号が決定されるのと同時に、変動パターン乱数判定テーブルが決定される。なお、図10において、変動パターン乱数判定テーブルの欄に記載しているテーブルxは、任意のテーブル番号を示している。したがって、取得したリーチグループ決定乱数と、

10

20

30

40

50

参照するリーチモード決定乱数判定テーブルの種類とに応じて、変動モード番号と、変動パターン乱数判定テーブルのテーブル番号とが決定されることとなる。また、同時回し参考例において、変動モード番号および後述する変動パターン番号は、16進数で設定されている。以下において、16進数を示す場合には「H」を付するが、図10～図12に○
○Hと記載しているのは、16進数で示される任意の値を示すものである。

【0119】

以上のように、大役抽選結果が「ハズレ」であった場合には、まず、図9に示すリーチグループ決定乱数判定テーブルとリーチグループ決定乱数とによってグループ種別が決定される。そして、決定されたグループ種別と遊技状態に応じ、図10(a)に示すハズレ時リーチモード決定乱数判定テーブルとリーチモード決定乱数とによって、変動モード番号および変動パターン乱数判定テーブルが決定される。

10

【0120】

一方、大役抽選結果が「大当たり」または「小当たり」であった場合には、決定された大当たり図柄または小当たり図柄（特別図柄の種別）、大当たり、または、小当たり当選時の遊技状態等に対応する、図10に示す大当たり時リーチモード決定乱数判定テーブルを参照し、リーチモード決定乱数を用いて、変動モード番号、変動パターン乱数判定テーブルが決定されることとなる。

【0121】

図11は、同時回し参考例に係る変動パターン乱数判定テーブルを説明する図である。ここでは、所定のテーブル番号xの変動パターン乱数判定テーブルxを示すが、変動パターン乱数判定テーブルは、この他にも、テーブル番号ごとに多数設けられている。

20

【0122】

第1始動口120または第2始動口122に遊技球が入球すると、0～238の範囲内から1つの変動パターン乱数が取得される。そして、上記の変動モード番号と同時に決定された変動パターン乱数判定テーブルと、取得した変動パターン乱数とに基づいて、図示のように変動パターン番号が決定される。

【0123】

このように、大役抽選が行われると、大役抽選結果、決定された図柄種別、遊技状態、保留数、保留種別等に応じて、変動モード番号、変動パターン番号が決定される。これら変動モード番号、変動パターン番号は、変動演出パターンを特定するものであり、そのそれぞれに、変動演出の態様および時間が対応付けられている。

30

【0124】

図12は、同時回し参考例に係る変動時間決定テーブルを説明する図である。上記のように、変動モード番号が決定されると、図12(a)に示す変動時間1決定テーブルにしたがって変動時間1が決定される。この変動時間1決定テーブルによれば、変動モード番号ごとに変動時間1が対応付けられており、決定された変動モード番号に応じて、対応する変動時間1が決定される。

【0125】

また、上記のように、変動パターン番号が決定されると、図12(b)に示す変動時間2決定テーブルにしたがって変動時間2が決定される。この変動時間2決定テーブルによれば、変動パターン番号ごとに変動時間2が対応付けられており、決定された変動パターン番号に応じて、対応する変動時間2が決定される。このようにして決定された変動時間1、2の合計時間が、大役抽選結果を報知する変動演出の時間、すなわち、変動時間となる。この変動時間というのは、決定された特別図柄を、第1特別図柄表示器160または第2特別図柄表示器162に停止表示させるまでの時間である。

40

【0126】

詳しくは後述するが、特1保留に基づいて特別図柄が決定されるとともに、変動モード番号および変動パターン番号すなわち変動時間が決定されると、当該決定された変動時間に亘って第1特別図柄表示器160において図柄の変動表示が行われ、変動時間が経過すると、第1特別図柄表示器160に、決定された特別図柄が停止表示される。また、特2

50

保留に基づいて特別図柄が決定されるとともに、変動パターン番号すなわち変動時間が決定されると、当該決定された変動時間に亘って第2特別図柄表示器162において図柄の変動表示が行われ、変動時間が経過すると、第2特別図柄表示器162に、決定された特別図柄が停止表示される。このとき、ハズレ図柄が第1特別図柄表示器160に停止表示されることで、大役抽選の結果としてハズレが確定し、次の特1保留に基づく大役抽選が実行可能となり、ハズレ図柄が第2特別図柄表示器162に停止表示されることで、大役抽選の結果としてハズレが確定し、次の特2保留に基づく大役抽選が実行可能となる。一方、大当たり図柄が第1特別図柄表示器160または第2特別図柄表示器162に停止表示されると、大役抽選の結果として大当たりが確定し、大役遊技が実行され、小当たり図柄が第1特別図柄表示器160または第2特別図柄表示器162に停止表示されると、大役抽選の結果として小当たりが確定し、小当たり遊技が実行されることとなる。

10

【0127】

このように、変動時間は、第1特別図柄表示器160または第2特別図柄表示器162における図柄の変動表示の時間、換言すれば、大役抽選の結果を確定させるまでの時間を規定するものとなる。

【0128】

以上のようにして変動モード番号が決定されると、当該決定された変動モード番号に対応する変動モードコマンドが副制御基板330に送信され、変動パターン番号が決定されると、当該決定された変動パターン番号に対応する変動パターンコマンドが副制御基板330に送信される。副制御基板330においては、受信した変動モードコマンドに基づいて、主に変動演出の前半の態様が決定され、受信した変動パターンコマンドに基づいて、主に変動演出の後半の態様が決定されることとなるが、その詳細については後述する。なお、以下では、変動モード番号および変動パターン番号を総称して変動情報と呼び、変動モードコマンドおよび変動パターンコマンドを総称して変動コマンドと呼ぶ場合がある。

20

【0129】

図13は、同時回し参考例に係る遊技状態および変動時間を説明する図である。上述のように、特別遊技状態と普通遊技状態とが組み合わせられて1の遊技状態となり、設定中の遊技状態に応じて遊技の進行制御がなされる。既に説明したように、特別遊技状態は、大当たりの当選確率を異にする低確率遊技状態および高確率遊技状態の2種類が設けられている。また、普通遊技状態は、互いに第1可変始動口120Bへの遊技球の入球容易性（入球頻度）を異にする非時短遊技状態、中時短遊技状態、時短遊技状態の3種類が設けられている。

30

【0130】

ここで、普通遊技において、第1可変始動口120Bへの遊技球の入球容易性は、当選確率、変動時間、開放時間の3つの要素によって決定される。詳しくは後述するが、普通遊技では、ゲート124を遊技球が通過するか、あるいは、普図作動口125に遊技球が入球すると普図保留が記憶される。そして、記憶された普図保留に基づいて、可動片120bを開放するか否かを決定する普図抽選が行われる。この普図抽選の結果は、所定の変動時間が経過したところで確定する。普図抽選の結果として当たりが確定すると、可動片120bが開放される。このとき、普図抽選における当選確率、変動時間、および、可動片120bを開放する際の開放時間が、それぞれ普通遊技状態ごとに設定されている。

40

【0131】

同時回し参考例では、図13(a)に示すように、特別遊技状態および普通遊技状態の組み合わせにより6種類の遊技状態が設けられる。遊技機100の初期状態は、低確率遊技状態および非時短遊技状態となっている。なお、非時短遊技状態では、普図抽選における当選確率が低く、変動時間が長く、可動片120bの開放時間が短い。同時回し参考例では、低確率遊技状態および非時短遊技状態が組み合わせられた遊技状態を通常状態と呼ぶ。

【0132】

また、同時回し参考例では、高確率遊技状態および非時短遊技状態に設定される場合があり、この両者が組み合わせられた遊技状態を最優位状態と呼ぶ。なお、この最優位状態は

50

、6つの遊技状態の中で最も有利度合いが高く、適切に遊技球を発射していると、大当たりに当選せずとも、遊技中に徐々に遊技球が増加していくように設定されている。

【0133】

また、同時回し参考例では、低確率遊技状態および時短遊技状態に設定される場合がある。なお、時短遊技状態では、普図抽選における当選確率が高く、変動時間が短く、可動片120bの開放時間が長い。以下では、低確率遊技状態および時短遊技状態が組み合わされた遊技状態を低確時短状態と呼ぶ。

【0134】

また、同時回し参考例では、高確率遊技状態および時短遊技状態に設定される場合がある。以下では、高確率遊技状態および時短遊技状態が組み合わされた遊技状態を高確時短状態または高確前兆状態と呼ぶ。なお、高確時短状態および高確前兆状態について、詳しくは後述する。

【0135】

また、同時回し参考例では、高確率遊技状態および中時短遊技状態に設定される場合がある。なお、中時短遊技状態では、普図抽選における当選確率が、非時短遊技状態よりも高く時短遊技状態よりも低く、変動時間が短く、可動片120bの開放時間が長い。この遊技状態は、適切に遊技球を発射させなかった場合等、不測の事態が生じた場合に設定され得るものである。以下では、高確率遊技状態および中時短遊技状態が組み合わされた遊技状態をペナルティ状態と呼ぶ。

【0136】

なお、副制御基板330においては、主制御基板300で設定されている遊技状態に対応する演出モードが設定される。演出モードというのは、メイン演出表示部200aに表示される背景画像やBGM等を規定するものであり、演出モードごとに、演出の内容が異なっている。つまり、遊技者は、演出モードによって、現在の遊技状態を識別することができる。

【0137】

以上のように、同時回し参考例では、6つの遊技状態が設けられている。そして、上記したように、大役抽選が行われたときの遊技状態、保留種別、当該遊技状態における変動回数、図柄の種別に応じて、変動モード番号および変動パターン番号、すなわち、変動時間が決定される。

【0138】

ここで、遊技機100では、遊技状態ごとに、実質変動対象が設定されている。実質変動対象というのは、本来、大役抽選を行うべき保留の種別を示しており、遊技状態ごとに、特1保留および特2保留のいずれかが実質変動対象に設定されている。通常状態では、特1保留が実質変動対象に設定されている。また、通常状態では、普通遊技状態が非時短遊技状態であるため、第1可変始動口120Bが殆ど開放されることがない。したがって、通常状態では、遊技者は、第1固定始動口120Aに遊技球を入球させるべく、第1遊技領域116aに向けて遊技球を発射させる必要がある。

【0139】

通常状態において、実質変動対象である特1保留によって大役抽選が行われ、ハズレ図柄もしくは小当たり図柄が決定されると、3～100秒の範囲内で変動時間が決定される。また、通常状態において、特1保留によって大役抽選が行われ、大当たり図柄が決定されると、40～100秒の範囲内で変動時間が決定される。

【0140】

一方、通常状態において、実質変動対象ではない特2保留によって大役抽選が行われた場合には、決定された図柄種別に拘わらず、変動時間が必ず10分に決定される。このように、変動時間を10分といった長時間に設定することで、通常状態では、仮に遊技者が第2始動口122に遊技球を入球させたとしても、特2保留に基づく大役抽選の実行機会が極めて少なくなる。

【0141】

具体的に説明すると、第2始動口122が第2遊技領域116bに設けられており、また、第2始動口122は、常に遊技球が入球可能となる固定始動口で構成されている。さらに、遊技機100の遊技性により、第2始動口122は、第1始動口120よりも、容易に遊技球が入球する位置に配されている。そのため、仮に、通常状態において、特2保留に係る変動時間を短時間としてしまうと、大役抽選の機会が必要以上に遊技者に与えられてしまう。そこで、通常状態における本来の遊技性に則り、適切に第1遊技領域116aに向けて遊技球を発射させるべく、変動時間を10分といった長時間に設定している。

【0142】

最優位状態では、特2保留が実質変動対象に設定されている。したがって、最優位状態では、遊技者は、第2始動口122に遊技球を入球させるべく、第2遊技領域116bに向けて遊技球を発射させる必要がある。最優位状態において、実質変動対象ではない特1保留によって大役抽選が行われた場合には、決定された図柄種別に拘わらず、変動時間が必ず10秒に決定される。なお、最優位状態において実質変動対象ではない特1保留によって大役抽選が行われる場合の方が、通常状態で実質変動対象ではない特2保留によって大役抽選が行われる場合に比べて、遊技性に与える影響が小さい。そのため、最優位状態では、実質変動対象ではない特1保留によって大役抽選が行われた場合の変動時間を、10秒と短く設定している。

【0143】

最優位状態において、実質変動対象である特2保留によって大役抽選が行われ、ハズレ図柄または小当たり図柄が決定されると、1秒～3秒の範囲内で変動時間が決定される。また、最優位状態において、特2保留によって大役抽選が行われ、大当たり図柄が決定されると、変動時間が3秒～10秒の範囲内で決定される。

【0144】

低確時短状態、高確時短状態では、いずれも特1保留が実質変動対象に設定されている。また、低確時短状態、高確時短状態では、普通遊技状態が時短遊技状態に設定されており、第1可変始動口120Bが頻繁に開状態に制御される。したがって、これら2つの遊技状態では、遊技者は、第1可変始動口120Bに遊技球を入球させるべく、第2遊技領域116bに向けて遊技球を発射させる必要がある。これら2つの遊技状態では、いずれも同一の変動パターン乱数判定テーブルが選択されるが、これら2つの遊技状態は、普通遊技状態が時短遊技状態である点で共通している。つまり、普通遊技状態が時短遊技状態である場合において、実質変動対象である特1保留によって大役抽選が行われ、ハズレ図柄もしくは小当たり図柄が決定されると、必ず1秒の変動時間が決定される。また、低確時短状態、高確時短状態である場合において、特1保留によって大役抽選が行われ、大当たり図柄が決定されると、変動時間が必ず30秒に決定される。

【0145】

高確前兆状態では、特1保留が実質変動対象に設定されている。また、高確前兆状態では、普通遊技状態が時短遊技状態に設定されており、第1可変始動口120Bが頻繁に開状態に制御される。したがって、高確前兆状態では、遊技者は、第1可変始動口120Bに遊技球を入球させるべく、第2遊技領域116bに向けて遊技球を発射させる必要がある。高確前兆状態において、実質変動対象である特1保留によって大役抽選が行われ、ハズレ図柄もしくは小当たり図柄が決定されると、3秒～10秒の範囲内で変動時間が決定される。また、高確前兆状態において、特1保留によって大役抽選が行われ、大当たり図柄が決定されると、変動時間が必ず30秒に決定される。

【0146】

一方、低確時短状態、高確時短状態および高確前兆状態、すなわち、普通遊技状態が時短遊技状態である場合において、実質変動対象ではない特2保留によって大役抽選が行われた場合には、決定された図柄種別に拘わらず、変動時間が必ず10分に決定される。

【0147】

ペナルティ状態では、特1保留が実質変動対象に設定されている。また、ペナルティ状態では、普通遊技状態が中時短遊技状態に設定されており、第1可変始動口120Bが一

10

20

30

40

50

定の頻度で開状態に制御される。したがって、ペナルティ状態では、遊技者は、第1可変始動口120Bに遊技球を入球させるべく、第2遊技領域116bに向けて遊技球を発射させる必要がある。ペナルティ状態において、実質変動対象である特1保留によって大役抽選が行われ、ハズレ図柄または小当たり図柄が決定されると、3～100秒の範囲内で変動時間が決定される。また、ペナルティ状態において、特1保留によって大役抽選が行われ、大当たり図柄が決定されると、40～100秒の範囲内で変動時間が決定される。

【0148】

一方、ペナルティ状態において、実質変動対象ではない特2保留によって大役抽選が行われた場合には、決定された図柄種別に拘わらず、変動時間が必ず10分に決定される。

【0149】

以上のように、遊技状態ごとに実質変動対象が設定されており、実質変動対象としての保留種別に基づく大役抽選が行われた場合には、最長でも変動時間が100秒となっている。一方、実質変動対象ではない保留種別に基づく大役抽選が行われた場合には、変動時間が概ね10分となり、本来の遊技性に反した遊技が行われないようにしている。

【0150】

なお、同時回し参考例では、高確時短状態の変動時間の平均が、高確前兆状態の変動時間の平均よりも短い場合について説明するが、高確時短状態の変動時間の平均が、高確前兆状態の変動時間の平均よりも長くてもよい。つまり、高確時短状態の変動時間の平均と、高確前兆状態の変動時間の平均とが異なってもよい。

【0151】

図14は、同時回し参考例に係る特別電動役物作動ラムセットテーブルを説明する第1の図であり、図15は、同時回し参考例に係る特別電動役物作動ラムセットテーブルを説明する第2の図である。なお、特別電動役物作動ラムセットテーブルは、大役遊技または小当たり遊技を制御するための各種データが記憶されたものであり、大役遊技中および小当たり遊技中は、特別電動役物作動ラムセットテーブルを参照して、第1大入賞口ソレノイド126cまたは第2大入賞口ソレノイド128cが通電制御される。なお、実際は、特別電動役物作動ラムセットテーブルは、特別図柄（大当たり図柄および小当たり図柄）の種別ごとに複数設けられており、決定された特別図柄の種別に応じて、対応するテーブルが大役遊技または小当たり遊技の開始時にセットされるが、ここでは、説明の都合上、図柄の種別ごとに特別図柄の制御データを示す。

【0152】

図14に示すように、大役遊技は、大入賞口が所定回数開閉される複数回のラウンド遊技で構成され、小当たり遊技は、ラウンド遊技が1回のみ実行される。この特別電動役物作動ラムセットテーブルによれば、オープニング時間（最初のラウンド遊技が開始されるまでの待機時間）、特別電動役物最大作動回数（1回の大役遊技または小当たり遊技中に実行されるラウンド遊技の回数）、特別電動役物開閉切替回数（1ラウンド中の大入賞口の開放回数）、ソレノイド通電時間（大入賞口の開放回数ごとの第1大入賞口ソレノイド126cまたは第2大入賞口ソレノイド128cの通電時間、すなわち、1回の大入賞口の開放時間）、規定数（1回のラウンド遊技における大入賞口への最大入賞可能数）、大入賞口閉鎖有効時間（ラウンド遊技間の大入賞口の閉鎖時間、すなわち、インターバル時間）、エンディング時間（最後のラウンド遊技が終了してから、通常の特別遊技（後述する図柄の変動表示）が再開されるまでの待機時間）が、制御データとして、特別図柄の種別ごとに、図示のように予め記憶されている。

【0153】

なお、特1保留によって大当たりに当選し、大当たり図柄として特別図柄A、B、Dが決定された場合には、4回のラウンド遊技で構成される大役遊技が実行される。この大役遊技においては、1～4回目のラウンド遊技において第1大入賞口126がそれぞれ1回のみ開放される。各ラウンド遊技では、第1大入賞口126が最大で29.0秒開放され、この間に規定数の遊技球が入球するか、もしくは、最大開放時間（29.0秒）が経過すると、第1大入賞口126が閉鎖されて1回のラウンド遊技が終了となる。

【0154】

また、特1保留によって大当たりに当選し、大当たり図柄として特別図柄C、Eが決定された場合には、10回のラウンド遊技で構成される大役遊技が実行される。これらの大役遊技においては、1～10回目のラウンド遊技において第1大入賞口126がそれぞれ1回のみ開放される。各ラウンド遊技では、第1大入賞口126が最大で29.0秒開放され、この間に規定数の遊技球が入球するか、もしくは、最大開放時間(29.0秒)が経過すると、第1大入賞口126が閉鎖されて1回のラウンド遊技が終了となる。

【0155】

また、特1保留によって小当たりに当選し、小当たり図柄として特別図柄Z1～Z3が決定された場合には、1回のラウンド遊技で構成される小当たり遊技が実行される。この小当たり遊技では、1回のラウンド遊技において、第2大入賞口128が0.1秒×1回開放される。

10

【0156】

また、図15に示すように、特2保留によって大当たりに当選し、大当たり図柄として特別図柄F、G、Iが決定された場合には、4回のラウンド遊技で構成される大役遊技が実行される。この大役遊技においては、1～4回目のラウンド遊技において第1大入賞口126がそれぞれ1回のみ開放される。各ラウンド遊技では、第1大入賞口126が最大で29.0秒開放され、この間に規定数の遊技球が入球するか、もしくは、最大開放時間(29.0秒)が経過すると、第1大入賞口126が閉鎖されて1回のラウンド遊技が終了となる。

20

【0157】

また、特2保留によって大当たりに当選し、大当たり図柄として特別図柄H、Jが決定された場合には、10回のラウンド遊技で構成される大役遊技が実行される。この大役遊技においては、1～10回目のラウンド遊技において第1大入賞口126がそれぞれ1回のみ開放される。各ラウンド遊技では、第1大入賞口126が最大で29.0秒開放され、この間に規定数の遊技球が入球するか、もしくは、最大開放時間(29.0秒)が経過すると、第1大入賞口126が閉鎖されて1回のラウンド遊技が終了となる。

【0158】

また、特2保留によって小当たりに当選し、小当たり図柄として特別図柄Z4～Z6が決定された場合にも、1回のラウンド遊技からなる小当たり遊技が実行される。ここで、小当たり図柄として特別図柄Z4が決定された場合の小当たり遊技では、1回のラウンド遊技において、第2大入賞口128が0.1秒×2回開放される。なお、第2大入賞口128の2回の開放の間の休止時間であるラウンド中インターバル時間は、1.78秒に設定されている。遊技球は最短で0.6秒間隔で発射されるため、第2大入賞口128の開放時間と、遊技球の発射間隔とから考えると、この小当たり遊技中に遊技球が第2大入賞口128に入球する確率は低い。

30

【0159】

しかしながら、同時回し参考例では、第2大入賞口128を閉状態に維持する可動片128b上に遊技球が滞留しやすい構造となっており、可動片128bが開状態に変移することで、可動片128b上に滞留する遊技球が第2大入賞口128内に導かれる。そのため、第2大入賞口128の0.1秒×2回の開放により、平均で2～3個の遊技球が第2大入賞口128に入球することとなる。

40

【0160】

また、小当たり図柄として特別図柄Z5が決定された場合の小当たり遊技では、1回のラウンド遊技において、第2大入賞口128が0.1秒×3回開放される。なお、この場合のラウンド中インターバル時間は、0.84秒に設定されている。この小当たり遊技においては、平均で3～4個の遊技球が第2大入賞口128に入球する。

【0161】

さらに、小当たり図柄として特別図柄Z6が決定された場合の小当たり遊技では、1回のラウンド遊技において、第2大入賞口128が0.1秒×12回開放される。なお、こ

50

の場合のラウンド中インターバル時間は、0.84秒に設定されている。この小当たり遊技においては、第2遊技領域116bに向けて遊技球を発射し続けることにより、ほぼ確実に規定数（例えば10個）の遊技球を第2大入賞口128に入球させることができる。

【0162】

なお、遊技機100の設計段階では、遊技球の発射球数と、払い出された賞球数との比率である発射賞球比率を厳密に管理、調整する必要がある。そのため、同時回し参考例では、小当たり遊技において0.1秒の開放が2回なされる特別図柄Z4と、小当たり遊技において0.1秒の開放が3回なされる特別図柄Z5とを設け、これらの選択比率を変更するだけで、発射賞球比率を容易に調整、変更することができるようにしている。

【0163】

図16は、同時回し参考例に係る大役遊技の終了後の遊技状態を設定するための遊技状態設定テーブルを説明する図である。同時回し参考例においては、大役遊技が実行された場合、大当たり当選時の遊技状態、保留種別、特別図柄（大当たり図柄）の種別に応じて、遊技状態設定テーブルを参照し、大役遊技の終了後の遊技状態を設定する。

【0164】

大当たり当選時の遊技状態が通常状態またはペナルティ状態である場合、実質変動対象である特1保留によって大当たりに当選すると、大当たり図柄の種別に応じて大役遊技後の遊技状態が設定される。具体的には、大当たり図柄として特別図柄Aが決定された場合には、低確時短状態（特別遊技状態が低確率遊技状態、普通遊技状態が時短遊技状態）に設定される。このとき、時短遊技状態の継続回数（以下、「時短回数」という）は100回に設定される。これは、大役抽選が100回行われるまで、時短遊技状態が継続することを意味している。ただし、上記した時短回数は1の時短遊技状態における最大継続回数を示すものであり、上記の継続回数に到達するまでの間に大当たりに当選した場合には、再度、遊技状態の設定が行われることとなる。したがって、大役遊技の終了後に時短遊技状態に設定された場合に、当該時短遊技状態において大当たりの抽選結果が導出されることなく、大当たり以外の抽選結果が100回導出されると、非時短遊技状態（通常状態）に遊技状態が変更されることとなる。

【0165】

また、大当たり図柄として特別図柄B、Dが決定された場合には、高確時短状態（特別遊技状態が高確率遊技状態、普通遊技状態が時短遊技状態）に設定される。このとき、高確回数として「次回」が設定され、高確率遊技状態は、次回の大当たりに当選するまで継続する。また、時短回数として「次回」が設定され、時短遊技状態は、次回の大当たりに当選するまで継続する。したがって、特別図柄B、Dが決定された場合には、大役遊技後、次回の大当たりに当選するまで、高確時短状態が継続することとなる。

【0166】

また、大当たり図柄として特別図柄C、Eが決定された場合には、高確前兆状態（特別遊技状態が高確率遊技状態、普通遊技状態が時短遊技状態）に設定される。このとき、高確回数として「次回」が設定され、時短回数が100回に設定される。特別図柄C、Eが決定された場合には、次回の大当たりの当選まで高確率遊技状態が継続する一方で、時短遊技状態は100回で終了となる。したがって、特別図柄C、Eが決定された場合には、大役遊技後、大役抽選の結果が100回導出されたところで、遊技状態が最優位状態に移行することとなる。

【0167】

また、大当たり当選時の遊技状態が通常状態またはペナルティ状態である場合、実質変動対象ではない特2保留によって大当たりに当選すると、大役遊技後の遊技状態が次のように設定される。すなわち、大当たり図柄として特別図柄Fが決定された場合には、通常状態（特別遊技状態が低確率遊技状態、普通遊技状態が非時短遊技状態）に設定される。また、大当たり図柄として特別図柄G～Jが決定された場合には、ペナルティ状態（特別遊技状態が高確率遊技状態、普通遊技状態が中時短遊技状態）に設定される。このとき、高確回数および時短回数は、いずれも「次回」に設定される。

【0168】

また、大当たり当選時の遊技状態が最優位状態である場合、実質変動対象ではない特1保留によって大当たりに当選すると、大役遊技後の遊技状態が次のように設定される。すなわち、大当たり図柄として特別図柄Aが決定された場合には、低確時短状態（特別遊技状態が低確率遊技状態、普通遊技状態が時短遊技状態）に設定される。このとき、時短回数は100回に設定される。また、大当たり図柄として特別図柄B～Eが決定された場合には、通常状態およびペナルティ状態と同様に、大役遊技後の遊技状態が設定される。

【0169】

一方、大当たり当選時の遊技状態が最優位状態である場合、実質変動対象である特2保留によって大当たりに当選すると、大役遊技後の遊技状態が次のように設定される。すなわち、大当たり図柄として特別図柄Fが決定された場合には、低確時短状態（特別遊技状態が低確率遊技状態、普通遊技状態が時短遊技状態）に設定される。このとき、時短回数は100回に設定される。また、大当たり図柄として特別図柄G、Iが決定された場合には、高確時短状態（特別遊技状態が高確率遊技状態、普通遊技状態が時短遊技状態）に設定される。このとき、高確回数および時短回数が「次回」に設定される。

【0170】

また、大当たり図柄として特別図柄H、Jが決定された場合には、高確前兆状態（特別遊技状態が高確率遊技状態、普通遊技状態が時短遊技状態）に設定される。このとき、高確回数が「次回」に設定され、時短回数が100回に設定される。

【0171】

また、大当たり当選時の遊技状態が低確時短状態、高確時短状態、高確前兆状態である場合、つまり、普通遊技状態が時短遊技状態である場合、大役遊技後の遊技状態は、通常状態およびペナルティ状態と同様に設定される。

【0172】

図17は、同時回し参考例に係る当たり決定乱数判定テーブルを説明する図である。遊技領域116を流下する遊技球がゲート124を通過するか、もしくは、普図作動口125に入球すると、第1可変始動口120Bの可動片120bを通電制御するか否かが対応付けられた普通図柄の判定処理（以下、「普図抽選」という）が行われる。

【0173】

なお、詳しくは後述するが、遊技球がゲート124を通過するか、もしくは、普図作動口125に入球すると、0～99の範囲内から1つの当たり決定乱数が取得されるとともに、この乱数値がメインRAM300cの普図保留記憶領域に4つを上限として記憶される。つまり、普図保留記憶領域は、当たり決定乱数をセーブする4つの記憶部を備えている。したがって、普図保留記憶領域の4つの記憶部全てに当たり決定乱数が記憶された状態で、遊技球がゲート124を通過するか、もしくは、普図作動口125に入球した場合には、当該遊技球の通過に基づいて当たり決定乱数が記憶されることはない。以下では、ゲート124を遊技球が通過するか、もしくは、普図作動口125に遊技球が入球して普図保留記憶領域に記憶された当たり決定乱数を普図保留とよぶ。

【0174】

普通遊技状態が非時短遊技状態であるときに普図抽選を開始する場合には、図17（a）に示すように、非時短遊技状態用当たり決定乱数判定テーブルが参照される。この非時短遊技状態用当たり決定乱数判定テーブルによれば、当たり決定乱数が0であった場合に、普通図柄の種別として当たり図柄が決定され、当たり決定乱数が1～99であった場合に、普通図柄の種別としてハズレ図柄が決定される。したがって、非時短遊技状態において当たり図柄が決定される確率、すなわち、当選確率は1/100となる。詳しくは後述するが、この普図抽選において当たり図柄が決定されると、第1可変始動口120Bの可動片120bが開状態に制御され、ハズレ図柄が決定された場合には、第1可変始動口120Bの可動片120bが閉状態に維持される。

【0175】

また、中時短遊技状態において普図抽選を開始する場合には、図17（b）に示すよう

に、中時短遊技状態用当たり決定乱数判定テーブルが参照される。この中時短遊技状態用当たり決定乱数判定テーブルによれば、当たり決定乱数が0～49であった場合に、普通図柄の種別として当たり図柄が決定され、当たり決定乱数が50～99であった場合に、普通図柄の種別としてハズレ図柄が決定される。したがって、中時短遊技状態において当たり図柄が決定される確率、すなわち、当選確率は50／100となる。

【0176】

また、時短遊技状態において普図抽選を開始する場合には、図17(c)に示すように、時短遊技状態用当たり決定乱数判定テーブルが参照される。この時短遊技状態用当たり決定乱数判定テーブルによれば、当たり決定乱数が0～98であった場合に、普通図柄の種別として当たり図柄が決定され、当たり決定乱数が99であった場合に、普通図柄の種別としてハズレ図柄が決定される。したがって、時短遊技状態において当たり図柄が決定される確率、すなわち、当選確率は99／100となる。

【0177】

図18(a)は、同時回し参考例に係る普通図柄変動時間データテーブルを説明する図であり、図18(b)は、同時回し参考例に係る開閉制御パターンテーブルを説明する図である。上記のように、普図抽選が行われると、普通図柄の変動時間が決定される。普通図柄変動時間データテーブルは、普図抽選によって当たり図柄もしくはハズレ図柄が決定されたときに、当該普通図柄の変動時間を決定する際に参照されるものである。この普通図柄変動時間データテーブルによれば、遊技状態が非時短遊技状態および中時短遊技状態に設定されている場合には変動時間が10秒に決定され、遊技状態が時短遊技状態に設定されている場合には変動時間が1秒に決定される。このようにして変動時間が決定されると、当該決定された時間にわたって普通図柄表示器168が変動表示(点滅表示)される。そして、当たり図柄が決定された場合には普通図柄表示器168が点灯し、ハズレ図柄が決定された場合には普通図柄表示器168が消灯する。

【0178】

そして、普図抽選によって当たり図柄が決定されるとともに、普通図柄表示器168が点灯した場合には、第1可変始動口120Bの可動片120bが、図18(b)に示すように、開閉制御パターンテーブルを参照して通電制御される。なお、実際は、開閉制御パターンテーブルは、遊技状態ごとに設けられており、普通図柄が決定されたときの遊技状態に応じて、対応するテーブルが普通電動役物ソレノイド120cの通電開始時にセットされる。

【0179】

当たり図柄が決定されると、図18(b)に示すように、開閉制御パターンテーブルを参照して第1可変始動口120Bが開閉制御される。この開閉制御パターンテーブルによれば、普電開放前時間(第1可変始動口120Bの開放が開始されるまでの待機時間)、普通電動役物最大開閉切替回数(第1可変始動口120Bの開放回数)、ソレノイド通電時間(第1可変始動口120Bの開放回数ごとの普通電動役物ソレノイド120cの通電時間、すなわち、1回の第1可変始動口120Bの開放時間)、規定数(第1可変始動口120Bの全開放中における第1可変始動口120Bへの最大入賞可能数)、普電閉鎖有効時間(第1可変始動口120Bの各開放間の閉鎖時間、すなわち、休止時間)、普電有効状態時間(第1可変始動口120Bの最後の開放終了からの待機時間)、普電終了ウェイト時間(普電有効状態時間の経過後、後述する普通図柄の変動表示が再開されるまでの待機時間)が、第1可変始動口120Bの制御データとして、遊技状態ごとに、図示のように予め記憶されている。

【0180】

このように、普通図柄の当選確率、変動時間および開放時間を設定することにより、図18(b)の下部に示すように、発射賞球比率(遊技領域116に発射される遊技球に対して第1可変始動口120B、第2始動口122、普図作動口125および大入賞口に遊技球が入球して遊技者に払い出される賞球数の比率)は、非時短遊技状態では発射数：賞球数＝100：20、中時短遊技状態では発射数：賞球数＝100：40、時短遊技状態

10

20

30

40

50

では発射数：賞球数＝１００：９９となる。

【０１８１】

なお、第１可変始動口１２０Ｂの開閉条件は、普通図柄の当選確率、普通図柄の変動表示の時間、第１可変始動口１２０Ｂの開放時間の３つの要素を規定するものである。つまり、普通図柄の当選確率、普通図柄の変動表示の時間、第１可変始動口１２０Ｂの開放時間の３つの要素を組み合わせることにより、非時短遊技状態、中時短遊技状態、時短遊技状態それぞれにおける、第１可変始動口１２０Ｂへの遊技球の入球頻度や、発射賞球比率を設定可能である。いずれにしても、ここに示した３つの要素の組み合わせは一例にすぎず、非時短遊技状態よりも時短遊技状態の方が、発射賞球比率が高くなるように、３つの要素を組み合わせればよい。

10

【０１８２】

図１９は、同時回し参考例に係る本来の遊技性に則った遊技状態の遷移を説明する図である。上記の構成により、遊技機１００は、次のような遊技性を実現する。なお、ここでは、登録設定値が「１」に設定されている場合について説明する。まず、遊技機１００の初期状態では、図１９の（ａ）に示す通常状態に設定されている。通常状態では、実質変動対象が特１保留に設定されているため、遊技者は第１固定始動口１２０Ａに遊技球を入球させるべく、第１遊技領域１１６ａに向けて遊技球を発射させる。第１遊技領域１１６ａは、遊技盤１０８の左側に位置することから、遊技者は、通常状態において、所謂「左打ち」を行うこととなる。

【０１８３】

第１固定始動口１２０Ａに遊技球が入球すると特１保留が第１特図保留記憶領域に記憶される。第１特図保留記憶領域に記憶された特１保留は、始動条件の成立により順次読み出され、読み出した特１保留に基づく大役抽選が行われる。このとき、大当たりの当選確率は約１／３００．６に設定されている。通常状態では、この特１保留に基づく大役抽選において、大当たりに当選することを目的として遊技を行うこととなる。なお、第１遊技領域１１６ａに向けて遊技球を発射した場合の発射賞球比率は１００：２０に設定されており、遊技中に遊技球が減少していくことになる。

20

【０１８４】

そして、通常状態において、特１保留による大役抽選で大当たりに当選すると、大役遊技が実行される。この大役遊技では、第１大入賞口１２６が開放されるラウンド遊技が４回または１０回実行され、遊技者は４ラウンドまたは１０ラウンド分の賞球を獲得することができる。そして、特１保留によって大当たりに当選した場合には、大当たり図柄として特別図柄Ａ～Ｅのいずれかが決定される。

30

【０１８５】

通常状態において、第１特別図柄表示器１６０に停止表示された大当たり図柄が特別図柄Ａであった場合、大役遊技後の遊技状態は、図１９の（ｂ）に示す低確時短状態となる。特１保留で大当たりに当選した場合に、大当たり図柄として特別図柄Ａが決定される確率は３０％である。したがって、通常状態で大当たりに当選した場合、３０％の確率で、遊技状態が低確時短状態に移行する。低確時短状態では、実質変動対象が特１保留に設定されているが、普通遊技状態が時短遊技状態であるため、遊技者は第１可変始動口１２０Ｂに遊技球を入球させるべく、第２遊技領域１１６ｂを狙った右打ちを行うこととなる。

40

【０１８６】

つまり、この低確時短状態では、通常状態と同様、特１保留に基づく大役抽選において、大当たりに当選することを目的として遊技を行うこととなる。なお、低確時短状態は、大当たりの当選確率が約１／３００．６であるが、普通遊技状態が時短遊技状態であることから、可動片１２０ｂが頻繁に開状態となる。そのため、発射賞球比率は１００：９９となり、遊技者は、遊技球の費消を低減しながら、大当たりの当選を狙うことができる。

【０１８７】

なお、低確時短状態に移行すると、時短回数が１００回に設定され、１００回の大役抽選で大当たりに当選しなかった場合には、遊技状態が再び通常状態に移行することとなる

50

(時短抜け)。

【0188】

また、通常状態において、第1特別図柄表示器160に停止表示された大当たり図柄が特別図柄B、Dであった場合、大役遊技後の遊技状態は、図19の(c)に示す高確時短状態となる。特1保留で大当たりに当選した場合に、大当たり図柄として特別図柄B、Dが決定される確率は35%である。したがって、通常状態で大当たりに当選した場合、35%の確率で、遊技状態が高確時短状態に移行する。高確時短状態では、実質変動対象が特1保留に設定されているが、普通遊技状態が時短遊技状態であるため、遊技者は第1可変始動口120Bに遊技球を入球させるべく、第2遊技領域116bを狙った右打ちを行うこととなる。

10

【0189】

つまり、この高確時短状態では、通常状態と同様、特1保留に基づく大役抽選において、大当たりに当選することを目的として遊技を行うこととなる。なお、高確時短状態は、大当たりの当選確率が約 $1/105.7$ であり、普通遊技状態が時短遊技状態であることから、可動片120bが頻繁に開状態となる。そのため、発射賞球比率は100:99となり、遊技者は、遊技球の費消を低減しながら、大役抽選を行うことができる。したがって、高確時短状態では、実質上、次回の大当たりの当選が確約された状態と言える。

【0190】

また、通常状態において、第1特別図柄表示器160に停止表示された大当たり図柄が特別図柄C、Eであった場合、大役遊技後の遊技状態は、図19の(d)に示す高確前兆状態となる。特1保留で大当たりに当選した場合に、大当たり図柄として特別図柄C、Eが決定される確率は35%である。したがって、通常状態で大当たりに当選した場合、35%の確率で、遊技状態が高確前兆状態に移行する。高確前兆状態では、実質変動対象が特1保留に設定されているが、普通遊技状態が時短遊技状態であるため、遊技者は第1可変始動口120Bに遊技球を入球させるべく、第2遊技領域116bを狙った右打ちを行うこととなる。

20

【0191】

そして、特別図柄C、Eが決定された場合には、高確前兆状態に設定されるとともに、時短回数が100回に設定される。このとき、大役遊技後の変動回数が100回に到達すると、時短遊技状態が終了し、普通遊技状態が非時短遊技状態となる。その結果、時短抜けにより、遊技状態が図19の(e)に示す最優位状態に移行することとなる。詳しくは後述するように、最優位状態では、第2遊技領域116bに向けて遊技球を発射し続けるだけで、遊技球を増加させることができる。したがって、高確前兆状態では、大当たりの当選が遊技の目的となるのではなく、大当たりに当選せずに、時短抜けすることが遊技の目的となる。

30

【0192】

上記の高確前兆状態、高確時短状態、低確時短状態における実質変動対象の特1保留によれば、大当たりに当選した場合に、大当たり図柄として特別図柄A～Eが決定される。特別図柄Aが決定されると、大役遊技において4回のラウンド遊技が実行されるとともに、大役遊技後の遊技状態が、図19の(b)に示す低確時短状態となる。また、特別図柄B、Dが決定された場合、大役遊技において4回または10回のラウンド遊技が実行されるとともに、大役遊技後の遊技状態が高確時短状態となる。また、特別図柄C、Eが決定された場合、大役遊技において4回または10回のラウンド遊技が実行されるとともに、大役遊技後の遊技状態が、高確前兆状態となる。

40

【0193】

最優位状態では、第2始動口122に遊技球が入球すると特2保留が第2特図保留記憶領域に記憶される。第2特図保留記憶領域に記憶された特2保留は、始動条件の成立により順次読み出され、読み出した特2保留に基づく大役抽選が行われる。このとき、大当たりの当選確率は約 $1/105.7$ に設定されており、また、小当たりの当選確率は約 $1/3.45$ に設定されている。最優位状態では、この特2保留に基づく大役抽選において、

50

小当たりに当選することが遊技の最大の目的となる。

【0194】

具体的には、第2遊技領域116bに遊技球を発射させた場合、発射球数に対する、第2始動口122への遊技球の入球によって払い出される賞球の比率は、100:60~80程度に設定されている。そして、最優位状態では、特2保留による大役抽選において、約1/3.45の確率で小当たりに当選するため、小当たり遊技が頻繁に行われる。ここで、特2保留によって小当たりに当選した場合には、小当たり図柄Z4~Z6が決定される。上記したように、小当たり図柄Z4が第2特別図柄表示器162に停止表示された場合の小当たり遊技では、平均2~3個の遊技球が第2大入賞口128に入球する。小当たり図柄Z5が第2特別図柄表示器162に停止表示された場合の小当たり遊技では、平均3~4個の遊技球が第2大入賞口128に入球する。さらに、小当たり図柄Z6が第2特別図柄表示器162に停止表示された場合の小当たり遊技では、ほぼ規定数の遊技球が第2大入賞口128に入球する。

10

【0195】

第2大入賞口128に遊技球が入球すると、例えば、1個の遊技球の入球に対して15個の賞球が払い出される。これにより、最優位状態では、発射球数に対する全ての賞球数の比率である発射賞球比率が100:120となり、第2遊技領域116bに向けて遊技球を発射し続けるだけで、遊技球を増加させることができる。

【0196】

なお、この最優位状態では、普通遊技状態が非時短遊技状態となっており、可動片120bが開状態となることは殆どない。また、最優位状態では、特別遊技状態が高確率遊技状態となっており、最優位状態における大当たりの当選確率が約1/105.7であることから、最優位状態では、実質上、次回の大当たりの当選が確約された状態と言える。

20

【0197】

この最優位状態における実質変動対象の特2保留によれば、大当たりに当選した場合に、大当たり図柄として特別図柄F~Jが決定される。特別図柄Fが決定された場合、大役遊技において4回のラウンド遊技が実行されるとともに、大役遊技後の遊技状態が、図19の(b)に示す低確時短状態となる。また、特別図柄G、Iが決定された場合、大役遊技において4回または10回のラウンド遊技が実行されるとともに、大役遊技後の遊技状態が高確時短状態となる。また、特別図柄H、Jが決定された場合、大役遊技において4回または10回のラウンド遊技が実行されるとともに、大役遊技後の遊技状態が、高確前兆状態となる。

30

【0198】

最優位状態は、他の遊技状態に比べて、有利度合いが極めて高いため、遊技機100における遊技の最大の目的は、遊技状態を最優位状態に移行させることとなる。上記のとおり、遊技はまず、通常状態で開始されるが、この通常状態から一気に最優位状態に移行することはない。そのため、高確前兆状態を経由して最優位状態へ移行するという移行ルートが、遊技機100における最優位状態への移行ルートとなる。

【0199】

さらに、同時回し参考例では、高確前兆状態における時短抜けとは別に、特定の小当たり図柄に当選することが、高確前兆状態から最優位状態への移行条件として設定されている。具体的には、高確前兆状態における実質変動対象である特1保留によって小当たりに当選した場合、小当たり図柄として、特別図柄Z1が1%、特別図柄Z2が69%、特別図柄Z3が30%の確率で決定される(図8(c)参照)。

40

【0200】

このとき、小当たり図柄として特別図柄Z1が決定されると、小当たり遊技の終了に伴って時短遊技状態が終了し、その結果、遊技状態が最優位状態へと移行することとなる。

【0201】

このように、特定の小当たりの当選によって最優位状態に移行することから、変動回数が規定回数(100回)に達したときにのみ最優位状態に移行する場合に比べて、遊技者

50

に対して、常時、期待感と緊張感が付与される。

【0202】

なお、上記の高確前兆状態、高確時短状態、低確時短状態においては、約1／3．45の確率で小当たりに当選する。そのため、高確前兆状態、高確時短状態、低確時短状態においても、最優位状態と同様に、頻繁に小当たり遊技が実行される。しかしながら、高確前兆状態、高確時短状態、低確時短状態は、いずれも普通遊技状態が時短遊技状態である。また、詳しくは後述するが、小当たり遊技中も、普通遊技状態は時短遊技状態に維持されている。そのため、小当たり遊技中に第2大入賞口128が開放されるものの、この間に、第1可変始動口120Bも開放されてしまう。

【0203】

上記したように、第1可変始動口120Bは、第2大入賞口128よりも上方に設けられており、しかも、時短遊技状態では、第1可変始動口120Bの開放時間が、第2大入賞口128の開放時間よりも極めて長い。そのため、高確前兆状態、高確時短状態、低確時短状態においては、第2遊技領域116bを流下する遊技球の殆どが第1可変始動口120Bに入球し、第2大入賞口128には殆ど遊技球が入球することはない。その結果、高確前兆状態、高確時短状態、低確時短状態においては、最優位状態とは異なり、遊技中に右打ちを行ったとしても、遊技球が徐々に減少することとなる。

【0204】

以上のように、本来の遊技性に則って実質変動対象によって遊技が進行すると、大当たり

【0205】

また、高確時短状態では、小当たり時およびハズレ時の変動時間が1秒に設定されるの

【0206】

したがって、高確時短状態では、小当たり時およびハズレ時の変動時間が相対的に短いため、大当たり

【0207】

一方、高確前兆状態では、小当たり時およびハズレ時の変動時間が相対的に長い

【0208】

このように、高確時短状態では、特定の小当たり（特別図柄Z1）に当選しても最優位

10

20

30

40

50

とを付与することができる。

【0209】

以上のように、特別遊技状態が高確率遊技状態であり、普通遊技状態が時短遊技状態であることが共通している高確時短状態および高確前兆状態が設けられ、高確時短状態の変動時間の平均を高確前兆状態の変動時間の平均よりも短くすることで、新たな遊技性を提供することができる。

【0210】

図20は、同時回し参考例に係る適切に遊技が行われなかった場合の遊技状態の遷移を説明する図である。上記のように、遊技機100では、第1特別図柄表示器160における図柄の変動表示と、第2特別図柄表示器162における図柄の変動表示とが、同時並行して行われる。このとき、実質変動対象以外の保留によって大役抽選が行われた結果、遊技者に不利益が生じる可能性がある場合については、変動時間を10分といった長時間に設定している。しかしながら、実質変動対象以外の保留によって大役抽選が行われた後、例えば、遊技を中断してしまう等した結果、実質変動対象以外の保留による大当たりが確定してしまうこともある。この場合には、図20に示すように遊技状態が遷移することとなる。

10

【0211】

以下に、上記の遊技性を実現するための、主制御基板300の主な処理について説明する。

【0212】

図21は、同時回し参考例に係る遊技機状態フラグを説明する図である。主制御基板300においては、遊技を進行可能な状態であるか否かが遊技機状態フラグにより管理される。遊技機状態フラグには、00H～05Hの6種類のフラグ値のいずれかがセットされる。遊技機状態フラグのフラグ値=00Hは遊技可能状態を示しており、遊技機状態フラグが00Hである場合に、遊技が進行制御され、遊技機状態フラグが00H以外である場合には、遊技が停止される。

20

【0213】

遊技機状態フラグのフラグ値=01Hは設定変更状態を示しており、遊技機状態フラグが01Hである場合には、登録設定値の変更操作が可能となる。遊技機状態フラグのフラグ値=02Hは設定確認状態を示しており、遊技機状態フラグが02Hである場合には、登録設定値が性能表示モニタ184に表示される等して、登録設定値を確認することが可能となる。遊技機状態フラグのフラグ値=03Hは設定異常状態を示しており、遊技機状態フラグが03Hである場合には、登録設定値が異常であるとして遊技が停止される。遊技機状態フラグのフラグ値=04HはRAM異常状態を示しており、遊技機状態フラグが04Hである場合には、遊技が停止される。遊技機状態フラグのフラグ値=05Hはチェックサム異常状態を示しており、遊技機状態フラグが05Hである場合には、遊技が停止される。電源が投入されると、遊技機状態フラグがいずれかのフラグ値にセットされ、遊技機状態フラグに応じた処理が行われる。

30

【0214】

(主制御基板300のCPU初期化処理)

図22は、同時回し参考例に係る主制御基板300におけるCPU初期化処理を説明する第1のフローチャートであり、図23は、同時回し参考例に係る主制御基板300におけるCPU初期化処理を説明する第2のフローチャートである。

40

【0215】

電源基板より電源が供給されると、メインCPU300aにシステムリセットが発生し、メインCPU300aは、以下のCPU初期化処理(S100)を行う。

【0216】

(ステップS100-1)

メインCPU300aは、電源投入に応じて、初期設定処理として、メインROM300bから起動プログラムを読み込むとともに、各種処理を実行するために必要な設定処理

50

を行う。

【0217】

(ステップS100-3)

メインCPU300aは、タイマカウンタにウェイト処理時間を設定する。

【0218】

(ステップS100-5)

メインCPU300aは、電源断予告信号を検出しているかを判定する。なお、主制御基板300には、電源断検知回路が設けられており、電源電圧が所定値以下になると、電源断検知回路から電源断予告信号が出力される。電源断予告信号を検出している場合には、上記ステップS100-3に処理を移し、電源断予告信号を検出していない場合には、

10

【0219】

(ステップS100-7)

メインCPU300aは、上記ステップS100-3で設定したウェイト時間が経過したか否かを判定する。その結果、ウェイト時間が経過したと判定した場合にはステップS100-9に処理を移し、ウェイト時間は経過していないと判定した場合には上記ステップS100-5に処理を移す。

【0220】

(ステップS100-9)

メインCPU300aは、メインRAM300cへのアクセスを許可するために必要な処理を実行する。

20

【0221】

(ステップS100-11)

メインCPU300aは、Dレジスタに電源断前の遊技機状態フラグのフラグ値をロードする。

【0222】

(ステップS100-13)

メインCPU300aは、チェックサムを算出するとともに、算出したチェックサムが、電源断時に保存されたチェックサムと一致する(正常である)か、ならびに、バックアップフラグが正常であることを判定する。その結果、バックアップフラグおよびチェックサムが正常であると判定した場合にはステップS100-15に処理を移し、いずれか一方または双方が正常ではないと判定した場合にはステップS100-25に処理を移す。

30

【0223】

(ステップS100-15)

メインCPU300aは、メインRAM300cのクリア対象の先頭アドレスに、設定値および遊技機状態フラグを含まない番地をセットする。

【0224】

(ステップS100-17)

メインCPU300aは、RAMクリアスイッチ182sからRAMクリア操作信号が入力されているか(RAMクリアボタンが押下操作されているか)を判定する。その結果、RAMクリア操作信号が入力されていると判定した場合にはステップS100-31に処理を移し、RAMクリア操作信号は入力されていないと判定した場合にはステップS100-19に処理を移す。

40

【0225】

(ステップS100-19)

メインCPU300aは、上記ステップS100-11でロードした遊技機状態フラグのフラグ値が00H(遊技可能状態)であり、設定変更スイッチ180sがオンしており、かつ、中枠104が開放しているかを判定する。その結果、3つの条件をすべて満たすと判定した場合にはステップS100-21に処理を移し、3つの条件の1つでも満たされないと判定した場合にはステップS100-23に処理を移す。

50

【0226】

(ステップS100-21)

メインCPU300aは、遊技機状態フラグに02H（設定確認状態）をセットする。すなわち、中枠104が開放されており、設定変更スイッチ180sがオンしており、RAMクリアボタンが押下されていない状態で正常に電源投入がなされると、設定確認状態となる。

【0227】

(ステップS100-23)

メインCPU300aは、メインRAM300cのうち、上記ステップS100-15でセットされた先頭アドレス以降の領域である電源復帰時のクリア対象をクリアする初期化処理を実行し、ステップS100-49に処理を移す。

10

【0228】

(ステップS100-25)

メインCPU300aは、Dレジスタに05H（チェックサム異常状態）をセットする。

【0229】

(ステップS100-27)

メインCPU300aは、使用外領域のリードライトメモリのチェックおよびクリアを行う領域外リードライトチェック処理を行う。

【0230】

(ステップS100-29)

メインCPU300aは、メインRAM300cのクリア対象の先頭アドレスに、設定値および遊技機状態フラグを含む番地をセットする。

20

【0231】

(ステップS100-31)

メインCPU300aは、使用領域のリードライトメモリのチェックおよびクリアを行う。

【0232】

(ステップS100-33)

メインCPU300aは、上記ステップS100-31におけるリードライトメモリのチェック結果が正常であるかを判定する。その結果、正常であると判定した場合にはステップS100-37に処理を移し、正常ではないと判定した場合にはステップS100-35に処理を移す。

30

【0233】

(ステップS100-35)

メインCPU300aは、Dレジスタに04H（RAM異常状態）をセットし、ステップS100-45に処理を移す。

【0234】

(ステップS100-37)

メインCPU300aは、Dレジスタに02H（設定確認状態）がセットされているかを判定する。その結果、02Hがセットされていると判定した場合にはステップS100-39に処理を移し、02Hはセットされていないと判定した場合にはステップS100-41に処理を移す。

40

【0235】

(ステップS100-39)

メインCPU300aは、Dレジスタに00H（遊技可能状態）をセットする。

【0236】

(ステップS100-41)

メインCPU300aは、設定変更条件を満たしているかを判定する。その結果、設定変更条件を満たしていると判定した場合にはステップS100-43に処理を移し、設定変更条件は満たしていないと判定した場合にはステップS100-45に処理を移す。な

50

お、ここでは、設定変更条件に、設定変更スイッチ180sがオンしていること、中枠104が開放していること、および、RAMクリアスイッチ182sからRAMクリア操作信号が入力されていることが少なくとも含まれる。

【0237】

(ステップS100-43)

メインCPU300aは、Dレジスタに01H（設定変更状態）をセットする。

【0238】

(ステップS100-45)

メインCPU300aは、Dレジスタにセットされている値を遊技機状態フラグにセーブする。

【0239】

(ステップS100-47)

メインCPU300aは、メインRAM300cのうち、RAMクリア時のクリア対象をクリアする初期化処理を実行し、ステップS100-49に処理を移す。

【0240】

(ステップS100-49)

メインCPU300aは、メインRAM300cがクリアされたことを払出制御基板310に伝達するための払出コマンド（RAMクリア指定コマンド）の送信処理（RAMクリア指定コマンドを送信バッファに格納）を行う。

【0241】

(ステップS100-51)

メインCPU300aは、遊技機状態フラグをロードする。

【0242】

(ステップS100-53)

メインCPU300aは、上記ステップS100-51でロードした遊技機状態フラグが00H（遊技可能状態）であるかを判定する。その結果、00Hであると判定した場合にはステップS110に処理を移し、00Hではないと判定した場合にはステップS100-55に処理を移す。

【0243】

(ステップS110)

メインCPU300aは、サブコマンド群セット処理を行う。なお、このサブコマンド群セット処理については後述する。

【0244】

(ステップS100-55)

メインCPU300aは、所定のコマンドを副制御基板330に送信するためのサブコマンドセット処理を行う。

【0245】

(ステップS100-57)

メインCPU300aは、タイマ割込みの周期を設定する。

【0246】

(ステップS100-59)

メインCPU300aは、割込みを禁止するための処理を行う。

【0247】

(ステップS100-61)

メインCPU300aは、当たり図柄乱数用初期値更新乱数を更新する。なお、当たり図柄乱数用初期値更新乱数は、当たり図柄乱数の初期値および終了値を決定するためのものである。つまり、後述する当たり図柄乱数の更新処理によって当たり図柄乱数が、当たり図柄乱数用初期値更新乱数から、当該当たり図柄乱数用初期値更新乱数-1まで1周すると、当たり図柄乱数は、そのときの当たり図柄乱数用初期値更新乱数に更新されることとなる。

10

20

30

40

50

【0248】

(ステップS100-63)

メインCPU300aは、払出制御基板310から受信した受信データ（主コマンド）を解析し、受信データに応じた種々の処理を実行する。

【0249】

(ステップS100-65)

メインCPU300aは、送信バッファに格納されているサブコマンドを副制御基板330に送信するための処理を行う。

【0250】

(ステップS100-67)

メインCPU300aは、割込みを許可するための処理を行う。

【0251】

(ステップS100-69)

メインCPU300aは、リーチグループ決定乱数、リーチモード決定乱数、変動パターン乱数を更新し、以後、上記ステップS100-59から処理を繰り返す。なお、以下では、変動演出パターンを決定するためのリーチグループ決定乱数、リーチモード決定乱数、変動パターン乱数を総称して変動演出用乱数と呼ぶ。

【0252】

図24は、同時回し参考例に係る主制御基板300におけるサブコマンド群セット処理（S110）を説明するフローチャートである。

【0253】

(ステップS110-1)

メインCPU300aは、遊技機状態フラグのフラグ値をロードする。

【0254】

(ステップS110-3)

メインCPU300aは、所定のコマンドを副制御基板330に送信するためのサブコマンドセット処理を行う。

【0255】

(ステップS110-5)

メインCPU300aは、遊技機100の機種情報を示す機種コマンドを送信バッファにセットする機種コマンド設定処理を行う。

【0256】

(ステップS110-7)

メインCPU300aは、登録設定値を示す設定値指定コマンドを送信バッファにセットする設定値指定コマンド設定処理を行う。

【0257】

(ステップS110-9)

メインCPU300aは、特1保留数を示す特図1保留指定コマンドを送信バッファにセットする特図1保留指定コマンド設定処理を行う。

【0258】

(ステップS110-11)

メインCPU300aは、特2保留数を示す特図2保留指定コマンドを送信バッファにセットする特図2保留指定コマンド設定処理を行う。

【0259】

(ステップS110-13)

メインCPU300aは、時短遊技状態の残り回数を示す回数コマンドを送信バッファにセットする回数コマンド設定処理を行う。

【0260】

(ステップS110-15)

メインCPU300aは、変動パターン選択状態を示す変動パターン選択状態指定コマ

10

20

30

40

50

ンドを送信バッファにセットする変動パターン選択状態指定コマンド設定処理を行う。

【0261】

(ステップS110-17)

メインCPU300aは、特別遊技管理フェーズを示す特図フェーズ指定コマンドを送信バッファにセットする特図フェーズ指定コマンド設定処理を行う。なお、特別遊技管理フェーズについては後述する。

【0262】

(ステップS110-19)

メインCPU300aは、特別遊技管理フェーズが特別図柄変動待ち状態であるかを判定する。その結果、特別図柄変動待ち状態であると判定した場合にはステップS110-21に処理を移し、特別図柄変動待ち状態ではないと判定した場合には当該サブコマンド群セット処理を終了する。

10

【0263】

(ステップS110-21)

メインCPU300aは、客待ち指定コマンドを送信バッファにセットし、当該サブコマンド群セット処理を終了する。

【0264】

次に、主制御基板300における割込み処理について説明する。ここでは、電源断時退避処理(XINT割込み処理)およびタイマ割込み処理について説明する。

【0265】

(主制御基板300の電源断時退避処理(XINT割込み処理))

図25は、同時回し参考例に係る主制御基板300における電源断時退避処理(XINT割込み処理)を説明するフローチャートである。メインCPU300aは、電源断検知回路を監視しており、電源電圧が所定値以下になると、CPU初期化処理に割り込んで電源断時退避処理を実行する。

20

【0266】

(ステップS300-1)

電源断予告信号が入力されると、メインCPU300aは、レジスタを退避する。

【0267】

(ステップS300-3)

メインCPU300aは、電源断予告信号をチェックする。

30

【0268】

(ステップS300-5)

メインCPU300aは、電源断予告信号を検出しているかを判定する。その結果、電源断予告信号を検出していると判定した場合にはステップS300-11に処理を移し、電源断予告信号を検出していないと判定した場合にはステップS300-7に処理を移す。

【0269】

(ステップS300-7)

メインCPU300aは、レジスタを復帰させる。

【0270】

(ステップS300-9)

メインCPU300aは、割込みを許可するための処理を行い、当該電源断時退避処理を終了する。

40

【0271】

(ステップS300-11)

メインCPU300aは、出力ポートの出力を停止する出力ポートクリア処理を実行する。

【0272】

(ステップS300-13)

メインCPU300aは、チェックサムを算出して保存するチェックサム設定処理を実

50

行する。

【0273】

(ステップS300-15)

メインCPU300aは、メインRAM300cへのアクセスを禁止するために必要なRAMプロテクト設定処理を実行する。

【0274】

(ステップS300-17)

メインCPU300aは、電源断発生監視時間を設定すべく、ループカウンタのカウント値に所定の電源断検出信号検出回数をセットする。

【0275】

(ステップS300-19)

メインCPU300aは、電源断予告信号をチェックする。

【0276】

(ステップS300-21)

メインCPU300aは、電源断予告信号を検出しているかを判定する。その結果、電源断予告信号を検出していると判定した場合にはステップS300-17に処理を移し、電源断予告信号を検出していないと判定した場合にはステップS300-23に処理を移す。

【0277】

(ステップS300-23)

メインCPU300aは、上記ステップS300-17でセットしたループカウンタの値を1減算する。

【0278】

(ステップS300-25)

メインCPU300aは、ループカウンタのカウント値が0でないかを判定する。その結果、カウンタ値が0ではないと判定した場合にはステップS300-19に処理を移し、カウンタ値が0であると判定した場合には上記したCPU初期化処理(ステップS100)に移行する。

【0279】

なお、実際に電源断が生じた場合には、ステップS300-17～ステップS300-25をループしている間に遊技機100の稼働が停止する。

【0280】

(主制御基板300のタイマ割込み処理)

図26は、同時回し参考例に係る主制御基板300におけるタイマ割込み処理を説明するフローチャートである。主制御基板300には、所定の周期(同時回し参考例では4ミリ秒、以下「4ms」という)毎にクロックパルスを発生させるリセット用クロックパルス発生回路が設けられている。そして、リセット用クロックパルス発生回路によって、クロックパルスが発生すると、CPU初期化処理(ステップS100)に割り込んで、以下のタイマ割込み処理が実行される。

【0281】

(ステップS400-1)

メインCPU300aは、レジスタを退避する。

【0282】

(ステップS400-3)

メインCPU300aは、割込みを許可するための処理を行う。

【0283】

(ステップS400-5)

メインCPU300aは、コモン出力バッファにセットされたコモンデータを出力ポートに出力し、第1特別図柄表示器160、第2特別図柄表示器162、第1特別図柄保留表示器164、第2特別図柄保留表示器166、普通図柄表示器168、普通図柄保留表

10

20

30

40

50

示器 170、右打ち報知表示器 172、性能表示モニタ 184 を点灯制御するダイナミックポート出力処理を実行する。

【0284】

(ステップ S400-7)

メイン CPU 300a は、各種の入力ポート情報を読み込み、最新のスイッチ状態を正確に取得するためのポート入力処理を実行する。

【0285】

(ステップ S400-9)

メイン CPU 300a は、遊技機状態フラグのフラグ値をロードする。

【0286】

(ステップ S400-11)

メイン CPU 300a は、上記ステップ S400-9 でロードしたフラグ値が 00H (遊技可能状態) であるかを判定する。その結果、00H であると判定した場合にはステップ S400-15 に処理を移し、00H ではないと判定した場合にはステップ S400-13 に処理を移す。

【0287】

(ステップ S400-13)

メイン CPU 300a は、上記ステップ S400-9 でロードしたフラグ値が 03H (設定異常状態) 以上であるかを判定する。その結果、03H 以上であると判定した場合にはステップ S400-29 に処理を移し、03H 以上ではないと判定した場合にはステップ S450 に処理を移す。

【0288】

(ステップ S450)

メイン CPU 300a は、設定関連処理を実行し、ステップ S400-29 に処理を移す。なお、設定関連処理については後述する。

【0289】

(ステップ S400-15)

メイン CPU 300a は、各種タイマカウンタを更新するタイマ更新処理を行う。ここで、各種タイマカウンタは、特に断る場合を除き、当該主制御基板 300 のタイマ割込み処理の度に減算され、0 になると減算を停止する。

【0290】

(ステップ S400-17)

メイン CPU 300a は、上記ステップ S100-61 と同様、当たり図柄乱数用初期値更新乱数の更新処理を実行する。

【0291】

(ステップ S400-19)

メイン CPU 300a は、当たり図柄乱数を更新する処理を行う。具体的には、乱数カウンタを 1 加算して更新し、加算した結果が乱数範囲の最大値を超えた場合には、乱数カウンタを 0 に戻し、乱数カウンタが 1 周した場合には、その時の当たり図柄乱数用初期値更新乱数の値から乱数を更新する。

【0292】

なお、詳しい説明は省略するが、同時回し参考例では、大当たり決定乱数および当たり決定乱数は、主制御基板 300 に内蔵されたハードウェア乱数生成部によって更新されるハードウェア乱数を用いている。ハードウェア乱数生成部は、大当たり決定乱数および当たり決定乱数を、いずれも一定の規則にしたがって更新し、乱数列が一巡するごとに自動的に乱数列を変更するとともに、システムリセット毎にスタート値を変更している。

【0293】

(ステップ S500)

メイン CPU 300a は、第 1 固定始動口検出スイッチ 120As、第 1 可変始動口検出スイッチ 120Bs、第 2 始動口検出スイッチ 122s、ゲート検出スイッチ 124s

10

20

30

40

50

、普図作動口検出スイッチ125s、第1大入賞口検出スイッチ126s、第2大入賞口検出スイッチ128sから信号の入力があったか否か判定するスイッチ管理処理を実行する。なお、このスイッチ管理処理の詳細については後述する。

【0294】

(ステップS600)

メインCPU300aは、特別遊技のうちの特2保留に基づく特別図柄の変動表示を進行制御するための特別遊技管理処理を実行する。なお、この特別遊技管理処理の詳細については後述する。

【0295】

(ステップS600)

メインCPU300aは、特別遊技のうちの特1保留に基づく特別図柄の変動表示を進行制御するための特別遊技管理処理を実行する。なお、ここでは、特2保留に基づく特別図柄の変動表示を進行制御するための特別遊技管理処理と同一のプログラム（モジュール）が読み出されて、特1保留に基づく特別図柄の変動表示を進行制御するための特別遊技管理処理が実行されることになる。

【0296】

(ステップS700)

メインCPU300aは、特別遊技における大役遊技および小当たり遊技を進行制御するための特別電動役物遊技管理処理を実行する。なお、この特別電動役物遊技管理処理の詳細については後述する。

【0297】

(ステップS800)

メインCPU300aは、上記の普通遊技を進行制御するための普通遊技管理処理を実行する。なお、この普通遊技管理処理の詳細については後述する。

【0298】

(ステップS400-21)

メインCPU300aは、各種エラーの判定およびエラー判定結果に応じた設定を行うためのエラー管理処理を実行する。

【0299】

(ステップS400-23)

メインCPU300aは、一般入賞口検出スイッチ118s、第1始動口検出スイッチ120s、第2始動口検出スイッチ122s、第1大入賞口検出スイッチ126s、第2大入賞口検出スイッチ128sのチェックを行い、該当する賞球制御用のカウンタ等を加算するための入賞口スイッチ処理を実行する。

【0300】

(ステップS400-25)

メインCPU300aは、上記ステップS400-23でセットされた賞球制御用のカウンタのカウンタ値等に基づく払出コマンドの作成および送信を行うための払出制御管理処理を実行する。

【0301】

(ステップS400-27)

メインCPU300aは、遊技球の発射位置、すなわち、第1遊技領域116aおよび第2遊技領域116bのどちらに遊技球を発射させるかを指示する発射位置指定コマンドを副制御基板330に送信するための発射位置指定管理処理を実行する。

【0302】

(ステップS400-29)

メインCPU300aは、遊技情報出力端子板312から外部へ出力する外部情報用の出力データをセットするための外部情報管理処理を実行する。

【0303】

(ステップS400-31)

10

20

30

40

50

メインCPU300aは、第1特別図柄表示器160、第2特別図柄表示器162、第1特別図柄保留表示器164、第2特別図柄保留表示器166、普通図柄表示器168、普通図柄保留表示器170、右打ち報知表示器172等の各種表示器(LED)を点灯制御するための表示データを、各コモンに対応した出力バッファにセットするLED表示設定処理を実行する。

【0304】

(ステップS400-33)

メインCPU300aは、普通電動役物ソレノイド120c、第1大入賞口ソレノイド126c、第2大入賞口ソレノイド128cのソレノイド出力イメージを合成し、出力ポートバッファに格納するためのソレノイド出力イメージ合成処理を実行する。

【0305】

(ステップS400-35)

メインCPU300aは、各出力ポートバッファに格納されたコモン出力バッファの値を出力ポートに出力するためのポート出力処理を実行する。

【0306】

(ステップS400-37)

メインCPU300aは、割込みを禁止するための処理を行う。

【0307】

(ステップS400-39)

メインCPU300aは、メインRAM300cの使用外領域を用いて、性能表示モニタ184に表示するベース比率を算出するための処理を行い、算出したベース比率を性能表示モニタ184に表示するためのコモンデータをコモン出力バッファにセットする性能表示モニタ制御処理を実行する。なお、性能表示モニタ制御処理においては、所定期間ごとにベース比率が算出される。ここで、性能表示モニタ184には、現在の期間のベース比率と、それ以前の期間のベース比率とが所定時間ごとに切り替え表示されてもよい。また、所定の操作に応じて、性能表示モニタ184に表示されるベース比率が切り替わってもよい。

【0308】

(ステップS400-41)

メインCPU300aは、レジスタを復帰してタイマ割込み処理を終了する。

【0309】

図27は、同時回し参考例に係る設定関連処理(S450)を説明するフローチャートである。

【0310】

(ステップS450-1)

メインCPU300aは、遊技機状態フラグのフラグ値が01H(設定変更状態)であるかを判定する。その結果、01Hであると判定した場合にはステップS450-3に処理を移し、01Hではないと判定した場合にはステップS450-15に処理を移す。

【0311】

(ステップS450-3)

メインCPU300aは、設定値バッファに記憶されている登録設定値を所定の処理領域にロードする。

【0312】

(ステップS450-5)

メインCPU300aは、RAMクリアスイッチ182sが押下されたか(RAMクリア操作信号が入力されているか)を判定する。その結果、RAMクリアスイッチ182sが押下されていると判定した場合にはステップS450-7に処理を移し、RAMクリアスイッチ182sが押下されていないと判定した場合にはステップS450-9に処理を移す。

【0313】

10

20

30

40

50

(ステップ S 4 5 0 - 7)

メイン CPU 3 0 0 a は、処理領域の設定値に 1 を加算する。

【 0 3 1 4 】

(ステップ S 4 5 0 - 9)

メイン CPU 3 0 0 a は、処理領域の設定値が 1 ～ 6 の範囲であるかを判定する。その結果、設定値が 1 ～ 6 の範囲であると判定した場合にはステップ S 4 5 0 - 1 3 に処理を移し、設定値が 1 ～ 6 の範囲ではないと判定した場合にはステップ S 4 5 0 - 1 1 に処理を移す。

【 0 3 1 5 】

(ステップ S 4 5 0 - 1 1)

メイン CPU 3 0 0 a は、処理領域の設定値を 1 にセットする。

【 0 3 1 6 】

(ステップ S 4 5 0 - 1 3)

メイン CPU 3 0 0 a は、処理領域の設定値を設定値バッファにセットする。

【 0 3 1 7 】

(ステップ S 4 5 0 - 1 5)

メイン CPU 3 0 0 a は、設定変更スイッチ 1 8 0 s がオンしているかを判定する。その結果、設定変更スイッチ 1 8 0 s がオンしていると判定した場合には当該設定関連処理を終了し、設定変更スイッチ 1 8 0 s はオンしていないと判定した場合にはステップ S 4 5 0 - 1 7 に処理を移す。

【 0 3 1 8 】

(ステップ S 4 5 0 - 1 7)

メイン CPU 3 0 0 a は、設定関連処理の終了を示す設定関連終了指定コマンドを送信バッファにセットする。

【 0 3 1 9 】

(ステップ S 1 1 0)

メイン CPU 3 0 0 a は、図 2 4 のサブコマンド群セット処理を実行する。すなわち、設定関連処理が実行された場合、その終了時に、機種コマンド、設定値指定コマンド、特図 1 保留指定コマンド、特図 2 保留指定コマンド、回数コマンド、変動パターン選択状態指定コマンド、特図フェーズ指定コマンド、客待ち指定コマンドが副制御基板 3 3 0 に送信されることとなる。

【 0 3 2 0 】

(ステップ S 4 5 0 - 1 9)

メイン CPU 3 0 0 a は、遊技機状態フラグに 0 0 H (遊技可能状態) をセットし、当該設定関連処理を終了する。

【 0 3 2 1 】

以上のように、同時回し参考例によれば、中枠 1 0 4 が開放され、設定変更スイッチ 1 8 0 s がオンされ、RAM クリアボタンが押下操作された状態で、正常に電源投入がなされると、CPU 初期化処理 (図 2 2) において、遊技機状態フラグに 0 1 H (設定変更状態) がセットされる。その後、タイマ割込み処理が実行されるが、遊技機状態フラグに 0 1 H (設定変更状態) がセットされているため、遊技の進行に係る全ての処理 (図 2 6 のステップ S 4 0 0 - 1 5 ～ステップ S 4 0 0 - 2 7) が停止され、設定関連処理が実行される。

【 0 3 2 2 】

設定関連処理は、設定変更スイッチ 1 8 0 s がオンしている間、繰り返し実行され、この設定関連処理中は、RAM クリアボタンの押下操作が、登録設定値の設定変更操作として受け付けられる。すなわち、設定変更操作を受け付ける設定変更処理 (S 4 5 0 - 1 ～S 4 5 0 - 1 3) 中は、設定変更操作に応じて、設定値バッファに記憶する登録設定値が、複数段階設けられた設定値のいずれかに切り替えられる。

【 0 3 2 3 】

10

20

30

40

50

そして、遊技機状態フラグに01H（設定変更状態）がセットされている状態で、設定変更スイッチ180sがオフに切り替わると、設定変更処理が終了となり、遊技機状態フラグに00H（遊技可能状態）がセットされる。これにより、次のタイマ割込み処理から、遊技の進行に係る処理が実行可能となる。

【0324】

ここで、同時回し参考例の設定関連処理では、RAMクリアボタンの押下操作、すなわち、登録設定値の設定変更操作の受け付け終了後に、サブコマンド群セット処理において、登録設定値に対応する設定値指定コマンドが副制御基板330に送信される。一方で、設定変更操作の受け付け中は、設定値指定コマンドが副制御基板330に送信されることはない。このように、設定変更操作の受け付け中は、設定値指定コマンドを送信せずに、設定変更操作の受け付けが終了し、遊技の進行が可能な状態に移行する場合に、設定値指定コマンドを送信することで、登録設定値が不正に取得されるリスクを低減することができる。

【0325】

また、同時回し参考例では、01H（設定変更状態）を少なくとも含む複数のフラグ値が切り替えられる。そして、遊技機状態フラグに01H（設定変更状態）がセットされている場合に設定関連処理が実行可能となり、かつ、遊技の進行が停止される。このように、遊技の進行中に設定関連処理が実行されることがないため、遊技の進行中に設定値指定コマンドが送信されることもなく、登録設定値が不正に取得されるリスクが低減される。

【0326】

次に、上記したタイマ割込み処理のうち、ステップS500のスイッチ管理処理、ステップS600の特別遊技管理処理、ステップS700の特別電動役物遊技管理処理、ステップS800の普通遊技管理処理について、詳細に説明する。

【0327】

図28は、同時回し参考例に係る主制御基板300におけるスイッチ管理処理（ステップS500）を説明するフローチャートである。

【0328】

（ステップS500-1）

メインCPU300aは、ゲート検出スイッチオン検出時であるか、すなわち、ゲート124を遊技球が通過してゲート検出スイッチ124sからの検出信号がオンされたかを判定する。その結果、ゲート検出スイッチオン検出時であると判定した場合にはステップS510に処理を移し、ゲート検出スイッチオン検出時ではないと判定した場合にはステップS500-7に処理を移す。

【0329】

（ステップS510）

メインCPU300aは、ゲート124への遊技球の通過に基づいてゲート通過処理を実行する。なお、このゲート通過処理の詳細については後述する。

【0330】

（ステップS500-3）

メインCPU300aは、普図作動口検出スイッチオン検出時であるか、すなわち、普図作動口125に遊技球が入球して普図作動口検出スイッチ125sからの検出信号がオンされたかを判定する。その結果、普図作動口検出スイッチオン検出時であると判定した場合にはステップS510に処理を移し、普図作動口検出スイッチオン検出時ではないと判定した場合にはステップS500-5に処理を移す。

【0331】

（ステップS510）

メインCPU300aは、普図作動口125への遊技球の入球に基づいてゲート通過処理を実行する。

【0332】

（ステップS500-5）

メインCPU300aは、第1固定始動口検出スイッチオン検出時であるか、すなわち、第1固定始動口120Aに遊技球が入球して第1固定始動口検出スイッチ120Asから検出信号が入力されたかを判定する。その結果、第1固定始動口検出スイッチオン検出時であると判定した場合にはステップS520に処理を移し、第1固定始動口検出スイッチオン検出時ではないと判定した場合にはステップS500-7に処理を移す。

【0333】

(ステップS520)

メインCPU300aは、第1固定始動口120Aへの遊技球の入球に基づいて第1始動口通過処理を実行する。なお、この第1始動口通過処理の詳細については後述する。

【0334】

(ステップS500-7)

メインCPU300aは、第1可変始動口検出スイッチオン検出時であるか、すなわち、第1可変始動口120Bに遊技球が入球して第1可変始動口検出スイッチ120Bsから検出信号が入力されたかを判定する。その結果、第1可変始動口検出スイッチオン検出時であると判定した場合にはステップS520に処理を移し、第1可変始動口検出スイッチオン検出時ではないと判定した場合にはステップS500-11に処理を移す。

【0335】

(ステップS520)

メインCPU300aは、第1可変始動口120Bへの遊技球の入球に基づいて第1始動口通過処理を実行する。なお、この第1始動口通過処理の詳細については後述する。

【0336】

(ステップS500-9)

メインCPU300aは、第1可変始動口120Bへの遊技球の入球が適正になされたものであるかを判定し、遊技球の入球が適正になされていないと判定した場合には、第1可変始動口120Bへの不正な遊技球の入球を示す普電不正入賞エラー発生指定コマンドを副制御基板330に送信するための普通電動役物入賞時確認処理を実行する。

【0337】

(ステップS500-11)

メインCPU300aは、第2始動口検出スイッチオン検出時であるか、すなわち、第2始動口122に遊技球が入球して第2始動口検出スイッチ122sから検出信号が入力されたかを判定する。その結果、第2始動口検出スイッチオン検出時であると判定した場合にはステップS530に処理を移し、第2始動口検出スイッチオン検出時ではないと判定した場合にはステップS500-13に処理を移す。

【0338】

(ステップS530)

メインCPU300aは、第2始動口122への遊技球の入球に基づいて第2始動口通過処理を実行する。なお、この第2始動口通過処理の詳細については後述する。

【0339】

(ステップS500-13)

メインCPU300aは、大入賞口検出スイッチオン検出時であるか、すなわち、第1大入賞口126または第2大入賞口128に遊技球が入球して第1大入賞口検出スイッチ126sまたは第2大入賞口検出スイッチ128sから検出信号が入力されたかを判定する。その結果、大入賞口検出スイッチオン検出時であると判定した場合にはステップS540に処理を移し、大入賞口検出スイッチオン検出時ではないと判定した場合には当該スイッチ管理処理を終了する。

【0340】

(ステップS540)

メインCPU300aは、第1大入賞口126または第2大入賞口128への遊技球の入球が適正になされたものであるかを判定し、遊技球の入球が適正になされた場合と判定した場合には、第1大入賞口126または第2大入賞口128への遊技球の入球を示す大入賞

10

20

30

40

50

口入球コマンドを副制御基板 330 に送信するための大入賞口通過処理を実行する。なお、この大入賞口通過処理の詳細については後述する。

【0341】

図 29 は、同時回し参考例に係る主制御基板 300 におけるゲート通過処理（ステップ S510）を説明するフローチャートである。

【0342】

（ステップ S510-1）

メイン CPU 300a は、ハードウェア乱数生成部によって更新された当たり決定乱数をロードする。

【0343】

（ステップ S510-3）

メイン CPU 300a は、普通図柄保留球数カウンタのカウンタ値が最大値以上であるか、つまり、普通図柄保留球数カウンタのカウンタ値が 4 以上であるかを判定する。その結果、普通図柄保留球数カウンタのカウンタ値が最大値以上であると判定した場合には当該ゲート通過処理を終了し、普通図柄保留球数カウンタは最大値以上ではないと判定した場合にはステップ S510-5 に処理を移す。

【0344】

（ステップ S510-5）

メイン CPU 300a は、普通図柄保留球数カウンタのカウンタ値を、現在のカウンタ値に「1」加算した値に更新する。

【0345】

（ステップ S510-7）

メイン CPU 300a は、普図保留記憶領域の 4 つの記憶部のうち、取得した当たり決定乱数をセーブする対象となる対象記憶部を算定する。

【0346】

（ステップ S510-9）

メイン CPU 300a は、上記ステップ S510-1 で取得した当たり決定乱数を、上記ステップ S510-7 で算定した対象記憶部にセーブする。

【0347】

（ステップ S510-11）

メイン CPU 300a は、普図保留記憶領域に記憶されている普図保留数を示す普図保留指定コマンドを送信バッファにセットし、当該ゲート通過処理を終了する。

【0348】

図 30 は、同時回し参考例に係る主制御基板 300 における第 1 始動口通過処理（ステップ S520）を説明するフローチャートである。

【0349】

（ステップ S520-1）

メイン CPU 300a は、特別図柄識別値として「00H」をセットする。なお、特別図柄識別値は、保留種別として特 1 保留および特 2 保留のいずれであるかを識別するためのもので、特別図柄識別値（00H）は特 1 保留を示し、特別図柄識別値（01H）は特 2 保留を示す。

【0350】

（ステップ S520-3）

メイン CPU 300a は、特別図柄 1 保留球数カウンタのアドレスをセットする。

【0351】

（ステップ S535）

メイン CPU 300a は、特別図柄乱数取得処理を実行し、当該第 1 始動口通過処理を終了する。なお、この特別図柄乱数取得処理は、第 2 始動口通過処理（ステップ S530）と共通のモジュールを利用して実行される。したがって、特別図柄乱数取得処理の詳細は、第 2 始動口通過処理の説明後に説明する。

10

20

30

40

50

【0352】

図31は、同時回し参考例に係る主制御基板300における第2始動口通過処理（ステップS530）を説明するフローチャートである。

【0353】

（ステップS530-1）

メインCPU300aは、特別図柄識別値として「01H」をセットする。

【0354】

（ステップS530-3）

メインCPU300aは、特別図柄2保留球数カウンタのアドレスをセットする。

【0355】

（ステップS535）

メインCPU300aは、後述する特別図柄乱数取得処理を実行し、当該第2始動口通過処理を終了する。

【0356】

図32は、同時回し参考例に係る主制御基板300における特別図柄乱数取得処理（ステップS535）を説明するフローチャートである。この特別図柄乱数取得処理は、上記した第1始動口通過処理（ステップS520）および第2始動口通過処理（ステップS530）において、共通のモジュールを用いて実行される。

【0357】

（ステップS535-1）

メインCPU300aは、上記ステップS520-1またはステップS530-1でセットした特別図柄識別値をロードする。

【0358】

（ステップS535-3）

メインCPU300aは、対象特別図柄保留球数をロードする。ここでは、上記ステップS535-1でロードした特別図柄識別値が「00H」であれば、特別図柄1保留球数カウンタのカウント値、すなわち、特1保留数をロードする。また、上記ステップS535-1でロードした特別図柄識別値が「01H」であれば、特別図柄2保留球数カウンタのカウント値、すなわち、特2保留数をロードする。

【0359】

（ステップS535-5）

メインCPU300aは、ハードウェア乱数生成部によって更新された大当たり決定乱数をロードする。

【0360】

（ステップS535-7）

メインCPU300aは、上記ステップS535-3でロードした対象特別図柄保留球数が上限値以上であるかを判定する。その結果、上限値以上であると判定した場合には、当該特別図柄乱数取得処理を終了し、上限値以上ではないと判定した場合には、ステップS535-9に処理を移す。

【0361】

（ステップS535-9）

メインCPU300aは、対象特別図柄保留球数カウンタのカウント値を、現在のカウンタ値に「1」加算した値に更新する。

【0362】

（ステップS535-11）

メインCPU300aは、特図保留記憶領域の記憶部のうち、取得した大当たり決定乱数をセーブする対象となる対象記憶部を算定する。

【0363】

（ステップS535-13）

メインCPU300aは、上記ステップS535-5でロードした大当たり決定乱数、

10

20

30

40

50

上記ステップ S 4 0 0 - 1 9 で更新された当たり図柄乱数、上記ステップ S 1 0 0 - 6 9 で更新された変動パターン乱数を取得し、上記ステップ S 5 3 5 - 1 1 で算定した対象記憶部に格納する。

【 0 3 6 4 】

(ステップ S 5 3 5 - 1 5)

メイン CPU 3 0 0 a は、特別図柄 1 保留球数カウンタおよび特別図柄 2 保留球数カウンタのカウント値をロードする。

【 0 3 6 5 】

(ステップ S 5 3 5 - 1 7)

メイン CPU 3 0 0 a は、上記ステップ S 5 3 5 - 1 5 でロードしたカウンタ値に基づいて、特図保留指定コマンドを送信バッファにセットする。ここでは、特別図柄 1 保留球数カウンタのカウント値（特 1 保留数）に基づいて特図 1 保留指定コマンドをセットし、特別図柄 2 保留球数カウンタのカウント値（特 2 保留数）に基づいて特図 2 保留指定コマンドをセットする。これにより、特 1 保留または特 2 保留が記憶されるたびに、特 1 保留数および特 2 保留数が副制御基板 3 3 0 に伝達されることとなる。

【 0 3 6 6 】

(ステップ S 5 3 6)

メイン CPU 3 0 0 a は、取得時演出判定処理を行い、当該特別図柄乱数取得処理を終了する。この取得時演出判定処理では、大役抽選の結果や、変動パターン番号等が仮判定され、仮判定の結果に応じた先読み指定コマンドを副制御基板 3 3 0 に送信する。この取得時演出判定処理について、図 3 3 を用いて説明する。

【 0 3 6 7 】

図 3 3 は、同時回し参考例に係る主制御基板 3 0 0 における取得時演出判定処理（ステップ S 5 3 6）を説明するフローチャートである。

【 0 3 6 8 】

(ステップ S 5 3 6 - 1)

メイン CPU 3 0 0 a は、設定中の設定値に基づいて、対応する大当たり決定乱数判定テーブルを選択する。具体的には、現在の遊技状態、および、設定中の設定値に基づいて、対応する大当たり決定乱数判定テーブルを選択する。そして、選択したテーブルと、上記ステップ S 5 3 5 - 1 3 で対象記憶部に記憶した大当たり決定乱数とに基づいて、大当たり、小当たり、ハズレのいずれかを仮判定する特別図柄当たり仮判定処理を行う。

【 0 3 6 9 】

(ステップ S 5 3 6 - 3)

メイン CPU 3 0 0 a は、特別図柄を仮決定するための特別図柄図柄仮判定処理を実行する。ここでは、上記ステップ S 5 3 6 - 1 の仮大役抽選の結果（特別図柄当たり仮判定処理によって導出された結果）が大当たりまたは小当たりであった場合には、上記ステップ S 5 3 5 - 1 3 で対象記憶部に記憶した当たり図柄乱数、当選種別（大当たりであるか小当たりであるか）、保留種別をロードし、対応する当たり図柄乱数判定テーブルを選択して特別図柄判定データを抽出し、抽出した特別図柄判定データ（大当たり図柄または小当たり図柄の種別）をセーブする。また、上記ステップ S 5 3 6 - 1 の仮大役抽選の結果がハズレであった場合には、所定のハズレ用の特別図柄判定データ（ハズレ図柄の種別）をセーブする。

【 0 3 7 0 】

(ステップ S 5 3 6 - 5)

メイン CPU 3 0 0 a は、ステップ S 5 3 6 - 3 でセーブした特別図柄判定データに対応する先読み図柄種別指定コマンド（先読み指定コマンド）を送信バッファにセットする。

【 0 3 7 1 】

(ステップ S 5 3 6 - 7)

メイン CPU 3 0 0 a は、上記ステップ S 5 3 6 - 1 の特別図柄当たり仮判定処理によって導出された結果が、大当たりまたは小当たりであるかを判定する。その結果、大当た

10

20

30

40

50

りまたは小当たりであると判定した場合にはステップS 5 3 6 - 9 に処理を移し、大当たりまたは小当たりではない（ハズレである）と判定した場合にはステップS 5 3 6 - 1 1 に処理を移す。

【0 3 7 2】

（ステップS 5 3 6 - 9）

メインCPU 3 0 0 a は、大当たり時リーチモード決定乱数判定テーブル（図1 0（b）、（c）参照）または小当たり時リーチモード決定乱数判定テーブル（図1 0（d）、（e））をセットし、ステップS 5 3 6 - 1 9 に処理を移す。

【0 3 7 3】

（ステップS 5 3 6 - 1 1）

メインCPU 3 0 0 a は、上記ステップS 5 3 5 - 1 3 で対象記憶部に記憶したリーチグループ決定乱数をロードする。

【0 3 7 4】

（ステップS 5 3 6 - 1 3）

メインCPU 3 0 0 a は、上記ステップS 5 3 6 - 1 1 でロードしたリーチグループ決定乱数が固定値（9 0 0 0 以上）であるかを判定する。ここで、グループ種別は、リーチグループ決定乱数判定テーブルを参照して決定されるが、このリーチグループ決定乱数判定テーブルは、記憶されている保留数に応じて選択される。このとき、リーチグループ決定乱数は、0 ~ 1 0 0 0 6 の範囲から取得され、リーチグループ決定乱数の値が9 0 0 0 以上であれば、保留数に拘わらず、同一のリーチグループ決定乱数判定テーブルが選択され、リーチグループ決定乱数の値が9 0 0 0 未満であれば、保留数に応じて異なるリーチグループ決定乱数判定テーブルが選択される。以下では、リーチグループ決定乱数のうち、保留数に応じて異なるリーチグループ決定乱数判定テーブルが選択される0 ~ 8 9 9 9 の範囲の値を不定値とし、保留数に拘わらず同一のリーチグループ決定乱数判定テーブルが選択される9 0 0 0 ~ 1 0 0 0 6 の範囲の値を固定値と呼ぶ。上記ステップS 5 3 6 - 1 1 でロードしたリーチグループ決定乱数が固定値（9 0 0 0 以上）であると判定した場合にはステップS 5 3 6 - 1 5 に処理を移し、上記ステップS 5 3 6 - 1 1 でロードしたリーチグループ決定乱数が固定値（9 0 0 0 以上）ではないと判定した場合にはステップS 5 3 6 - 2 7 に処理を移す。

【0 3 7 5】

（ステップS 5 3 6 - 1 5）

メインCPU 3 0 0 a は、リーチグループ決定乱数判定テーブル（図9 参照）をセットする。なお、リーチグループ決定乱数判定テーブルは、保留数に応じて複数種類設けられているが、ここでは、保留数が0 のときに用いられるテーブルが選択される。そして、セットしたリーチグループ決定乱数判定テーブルと、上記ステップS 5 3 5 - 1 3 で対象記憶部に記憶したリーチグループ決定乱数とに基づいて、リーチグループ（グループ種別）を仮決定する。

【0 3 7 6】

（ステップS 5 3 6 - 1 7）

メインCPU 3 0 0 a は、上記ステップS 5 3 6 - 1 5 で仮決定されたグループ種別に対応するハズレ時リーチモード決定乱数判定テーブル（図1 0（a）参照）をセットし、ステップS 5 3 6 - 1 9 に処理を移す。

【0 3 7 7】

（ステップS 5 3 6 - 1 9）

メインCPU 3 0 0 a は、上記ステップS 5 3 6 - 9 または上記ステップS 5 3 6 - 1 7 でセットしたリーチモード決定乱数判定テーブルと、上記ステップS 5 3 5 - 1 3 で対象記憶部に記憶したリーチモード決定乱数とに基づいて、変動モード番号を仮決定する。また、ここでは、変動モード番号とともに、変動パターン乱数判定テーブルが仮決定される。

【0 3 7 8】

10

20

30

40

50

(ステップ S 5 3 6 - 2 1)

メイン CPU 3 0 0 a は、上記ステップ S 5 3 6 - 1 9 で仮決定した変動モード番号に対応する先読み指定変動モードコマンド（先読み指定コマンド）を送信バッファにセットする。

【 0 3 7 9 】

(ステップ S 5 3 6 - 2 3)

メイン CPU 3 0 0 a は、上記ステップ S 5 3 6 - 1 9 で仮決定した変動パターン乱数判定テーブルと、上記ステップ S 5 3 5 - 1 3 で対象記憶部に記憶した変動パターン乱数とに基づいて、変動パターン番号を仮決定する。

【 0 3 8 0 】

(ステップ S 5 3 6 - 2 5)

メイン CPU 3 0 0 a は、上記ステップ S 5 3 6 - 2 3 で仮決定した変動パターン番号に対応する先読み指定変動パターンコマンド（先読み指定コマンド）を送信バッファにセットし、当該取得時演出判定処理を終了する。

【 0 3 8 1 】

(ステップ S 5 3 6 - 2 7)

メイン CPU 3 0 0 a は、対象記憶部に新たに記憶された保留について、当該保留が読み出されたときの保留数に応じて、グループ種別、すなわち、変動演出パターンが変化することを示す不定値コマンド（先読み指定変動モードコマンドおよび先読み指定変動パターンコマンド = 7 F H）を送信バッファにセットし、当該取得時演出判定処理を終了する。

【 0 3 8 2 】

図 3 4 は、同時回し参考例に係る主制御基板 3 0 0 における大入賞口通過処理（ステップ S 5 4 0）を説明するフローチャートである。

【 0 3 8 3 】

(ステップ S 5 4 0 - 1)

メイン CPU 3 0 0 a は、上記ステップ S 5 0 0 - 1 3 で大入賞口検出スイッチオン検出時であると判定した場合には、詳しくは後述する特別電動役物遊技管理フェーズをロードする。なお、詳しくは後述するが、特別電動役物遊技管理フェーズは、大役遊技または小当たり遊技の実行処理の段階、すなわち、大役遊技または小当たり遊技の進行状況を示すものであり、大役遊技または小当たり遊技の実行処理の段階に応じて更新される。

【 0 3 8 4 】

(ステップ S 5 4 0 - 3)

メイン CPU 3 0 0 a は、上記ステップ S 5 4 0 - 1 でロードした特別電動役物遊技管理フェーズが、大入賞口開放前処理以上の実行処理の段階を示すものであるかを判定する。なお、特別電動役物遊技管理フェーズは、0 0 H ~ 0 8 H の 9 段階が設けられており、このうち、0 1 H ~ 0 8 H が、大入賞口開放前処理以上の実行処理の段階に相当する。大役遊技または小当たり遊技は、特別電動役物遊技管理フェーズが 0 1 H ~ 0 8 H である場合に実行されることから、ここでは、現在、大役遊技中または小当たり遊技中であるかを判定することとなる。特別電動役物遊技管理フェーズが、大入賞口開放前処理以上の実行処理の段階を示すものであると判定した場合にはステップ S 5 4 0 - 5 に処理を移し、特別電動役物遊技管理フェーズは、大入賞口開放前処理以上の実行処理の段階を示すものではないと判定した場合にはステップ S 5 4 0 - 7 に処理を移す。

【 0 3 8 5 】

(ステップ S 5 4 0 - 5)

メイン CPU 3 0 0 a は、第 1 大入賞口 1 2 6 または第 2 大入賞口 1 2 8 に適正に遊技球が入球したことを示す大入賞口入球コマンドを送信バッファにセットし、大入賞口通過処理を終了する。

【 0 3 8 6 】

(ステップ S 5 4 0 - 7)

メイン CPU 3 0 0 a は、第 1 大入賞口 1 2 6 または第 2 大入賞口 1 2 8 への遊技球の

10

20

30

40

50

入球が不適切なものであるとして、所定のエラー処理を実行し、大入賞口通過処理を終了する。

【0387】

図35は、同時回し参考例に係る特別遊技管理フェーズを説明する図である。既に説明したとおり、同時回し参考例では、第1始動口120または第2始動口122への遊技球の入球を契機とする特別遊技と、ゲート124への遊技球の通過もしくは普図作動口125への遊技球の入球を契機とする普通遊技とが、同時並行して進行する。特別遊技に係る処理は、段階的に、かつ、繰り返し実行されるが、主制御基板300では、こうした特別遊技に係る各処理を特別遊技管理フェーズおよび特別電動役物遊技管理フェーズによって管理している。

10

【0388】

図35に示すように、メインROM300bには、特別遊技のうちの特別図柄の変動表示を実行制御するための複数の特別遊技制御モジュールが格納されており、これら特別遊技制御モジュールごとに、特別遊技管理フェーズが対応付けられている。具体的には、特別遊技管理フェーズが「00H」である場合には、「特別図柄変動待ち処理」を実行するためのモジュールがコールされ、特別遊技管理フェーズが「01H」である場合には、「特別図柄変動中処理」を実行するためのモジュールがコールされ、特別遊技管理フェーズが「02H」である場合には、「特別図柄停止図柄表示処理」を実行するためのモジュールがコールされる。

【0389】

20

また、メインROM300bには、特別遊技のうちの大役遊技および小当たり遊技を実行制御するための複数の特別電動役物遊技制御モジュールが格納されており、これら特別電動役物遊技制御モジュールごとに、特別電動役物遊技管理フェーズが対応付けられている。具体的には、特別電動役物遊技管理フェーズが「01H」または「05H」である場合には、「大入賞口開放前処理」を実行するためのモジュールがコールされ、特別電動役物遊技管理フェーズが「02H」または「06H」である場合には、「大入賞口開放制御処理」を実行するためのモジュールがコールされ、特別電動役物遊技管理フェーズが「03H」または「07H」である場合には、「大入賞口閉鎖有効処理」を実行するためのモジュールがコールされ、特別電動役物遊技管理フェーズが「04H」または「08H」である場合には、「大入賞口終了ウェイト処理」を実行するためのモジュールがコールされる。なお、特別電動役物遊技管理フェーズが「00H」である場合には、いずれの特別電動役物遊技制御モジュールもコールされることはない。

30

【0390】

図36は、同時回し参考例に係る主制御基板300における特別遊技管理処理（ステップS600）を説明するフローチャートである。

【0391】

（ステップS600-1）

メインCPU300aは、特別電動役物遊技管理フェーズをロードする。

【0392】

（ステップS600-3）

40

メインCPU300aは、上記ステップS600-1でロードした特別電動役物遊技管理フェーズが、「00H」以外であるかを判定する。すなわち、ここでは、現在、大役遊技中または小当たり遊技中であるかを判定することとなる。特別電動役物遊技管理フェーズが「00H」以外であると判定した場合には当該特別遊技管理処理を終了し、特別電動役物遊技管理フェーズが「00H」以外でないと判定した場合にはステップS600-5に処理を移す。

【0393】

（ステップS600-5）

メインCPU300aは、特別遊技特別図柄判定フラグをロードする。なお、特別遊技特別図柄判定フラグは、特別遊技管理処理の対象となる保留種別として特1保留および特

50

2 保留のいずれであるかを判定するためのものであり、特別遊技特別図柄判定フラグ（00H）は特1 保留を示し、特別遊技特別図柄判定フラグ（01H）は特2 保留を示す。

【0394】

（ステップS600-7）

メインCPU300aは、上記ステップS600-5でロードした特別遊技特別図柄判定フラグを反転する。ここでは、特別遊技特別図柄判定フラグが「00H」であった場合には「01H」に反転し、特別遊技特別図柄判定フラグが「01H」であった場合には「00H」に反転する。特別遊技特別図柄判定フラグは初期値が「00H」に設定されているため、図26に示した2回の特別遊技管理処理S600のうち、1回目の特別遊技管理処理S600で特別遊技特別図柄判定フラグが「01H」にされ、特2 保留について後続の処理が実行され、2回目の特別遊技管理処理S600で特別遊技特別図柄判定フラグが「00H」にされ、特1 保留について後続の処理が実行される。つまり、特2 保留が優先して処理されることになる。

10

【0395】

（ステップS600-9）

メインCPU300aは、上記ステップS600-7で反転した特別遊技特別図柄判定フラグをセーブする。

【0396】

（ステップS600-11）

メインCPU300aは、特別遊技管理フェーズをロードする。

20

【0397】

（ステップS600-13）

メインCPU300aは、上記ステップS600-11でロードした特別遊技管理フェーズに対応する特別遊技制御モジュールを選択する。

【0398】

（ステップS600-15）

メインCPU300aは、上記ステップS600-13で選択した特別遊技制御モジュールをコールして処理を開始する。

【0399】

（ステップS600-17）

メインCPU300aは、特別遊技の制御時間を管理する特別遊技タイマをロードし、当該特別遊技管理処理を終了する。

30

【0400】

図37は、同時回し参考例に係る主制御基板300における特別図柄変動待ち処理を説明するフローチャートである。この特別図柄変動待ち処理は、特別遊技管理フェーズが「00H」であった場合に実行される。

【0401】

（ステップS610-1）

メインCPU300aは、特別遊技管理処理の対象となる保留（特1 保留または特2 保留、以下、対象保留と呼ぶ）の特別図柄保留球数が1以上であるかを判定する。その結果、特別図柄保留球数が1以上であると判定した場合にはステップS610-3に処理を移し、特別図柄保留球数が1以上ではないと判定した場合には当該特別図柄変動待ち処理を終了する。

40

【0402】

（ステップS610-3）

メインCPU300aは、特別遊技管理処理の対象となっていない保留（特2 保留または特1 保留、以下、非対象保留と呼ぶ）に基づく特別図柄（以下、非対象特別図柄と呼ぶ）が確定中であるかを判定する。その結果、非対象特別図柄が確定中であると判定した場合にはステップS610-5に処理を移し、非対象特別図柄に基づく特別図柄が確定中ではないと判定した場合にはステップS610-9に処理を移す。

50

【0403】

(ステップS610-5)

メインCPU300aは、非対象特別図柄が大当たり図柄であるかを判定する。その結果、大当たり図柄であると判定した場合には当該特別図柄変動待ち処理を終了し、大当たり図柄でないと判定した場合にはステップS610-7に処理を移す。

【0404】

(ステップS610-7)

メインCPU300aは、非対象特別図柄が小当たり図柄であるかを判定する。その結果、小当たり図柄であると判定した場合には当該特別図柄変動待ち処理を終了し、小当たり図柄でないと判定した場合にはステップS610-9に処理を移す。

10

【0405】

(ステップS610-9)

メインCPU300aは、対象保留に対応する特図保留記憶領域の第1記憶部～第4記憶部に記憶されている対象保留を、1つ序数の小さい記憶部にブロック転送する。具体的には、第2記憶部～第4記憶部に記憶されている対象保留を、第1記憶部～第3記憶部に転送する。また、メインRAM300cには、処理対象となる第0記憶部が設けられており、第1記憶部に記憶されている対象保留を、第0記憶部にブロック転送する。なお、この特別図柄記憶エリアシフト処理においては、対象保留に対応する対象特別図柄保留球数カウンタのカウント値を「1」減算するとともに、対象保留が「1」減算されたことを示す、保留減指定コマンドを送信バッファにセットする。

20

【0406】

(ステップS611)

メインCPU300aは、大役抽選を行う特別図柄当たり判定処理を実行する。この特別図柄当たり判定処理については後述する。

【0407】

(ステップS610-11)

メインCPU300aは、特別図柄を決定するための特別図柄図柄判定処理を実行する。ここでは、上記ステップS611における大役抽選の結果が大当たりまたは小当たりであった場合、第0記憶部に転送された当たり図柄乱数、保留種別をロードし、対応する当たり図柄乱数判定テーブルまたは小当たり図柄乱数判定テーブルを選択して特別図柄判定データを抽出し、抽出した特別図柄判定データ（大当たり図柄の種別）をセーブする。また、上記ステップS611における大役抽選の結果がハズレであった場合には、ハズレ用の特別図柄判定データをセーブする。そして、特別図柄判定データをセーブしたら、当該特別図柄判定データに対応する図柄種別指定コマンドを送信バッファにセットする。

30

【0408】

(ステップS610-13)

メインCPU300aは、上記ステップS610-11で抽出した特別図柄判定データに対応する特別図柄停止図柄番号をセーブする。なお、第1特別図柄表示器160および第2特別図柄表示器162は、それぞれ7セグで構成されており、7セグを構成する各セグメントには番号（カウンタ値）が対応付けられている。ここで決定する特別図柄停止図柄番号は、最終的に点灯するセグメントの番号（カウンタ値）を示すものである。

40

【0409】

(ステップS612)

メインCPU300aは、変動モード番号および変動パターン番号を決定する特別図柄変動番号決定処理を実行する。この特別図柄変動番号決定処理の詳細は後述する。

【0410】

(ステップS610-15)

メインCPU300aは、上記ステップS612で決定された変動モード番号および変動パターン番号をロードするとともに、変動時間決定テーブルを参照して、変動時間1および変動時間2を決定する。そして、決定した変動時間1、2の合計時間を、特別図柄変

50

動タイマにセットする。

【0411】

(ステップS610-17)

メインCPU300aは、大役抽選の結果が大当たりであるか否かを判定し、大当たりであった場合には、上記ステップS610-11でセーブした特別図柄判定データをロードして、大当たり図柄の種別を確認する。そして、遊技状態設定テーブルおよび現在の遊技状態を参照して、大役遊技終了後に設定される遊技状態および高確回数、時短回数を判定し、その判定結果を特別図柄確率状態予備フラグ、時短状態予備フラグ、および、高確回数切り予備カウンタ、時短回数切り予備カウンタにセーブする。なお、ハズレ図柄がセーブされている場合には、当該処理を実行することなく、次の処理に移行する。

10

【0412】

(ステップS610-19)

メインCPU300aは、第1特別図柄表示器160または第2特別図柄表示器162において、特別図柄の変動表示を開始するために、特別図柄表示図柄カウンタを設定する処理を実行する。第1特別図柄表示器160および第2特別図柄表示器162を構成する7セグの各セグメントにはカウンタ値が対応付けられており、特別図柄表示図柄カウンタに設定されたカウンタ値に対応するセグメントが点灯制御される。ここでは、特別図柄の変動表示の開始時に点灯させるセグメントに対応するカウンタ値が特別図柄表示図柄カウンタに設定されることとなる。なお、特別図柄表示図柄カウンタは、第1特別図柄表示器160に対応する特別図柄1表示図柄カウンタと、第2特別図柄表示器162に対応する特別図柄2表示図柄カウンタとが別個に設けられており、ここでは、保留種別に対応するカウンタにカウンタ値が設定される。

20

【0413】

(ステップS613)

メインCPU300aは、回数切り管理処理を実行する。ここでは、変動回数に応じて時短遊技状態を終了させるための処理が行われる。この回数切り管理処理については後述する。

【0414】

(ステップS610-21)

メインCPU300aは、高確回数および時短回数が0になるまでの残り回数（実残り回数）を示す回数コマンドを送信バッファにセットする。

30

【0415】

(ステップS610-23)

メインCPU300aは、特別図柄の変動表示の開始時の遊技状態を示す遊技状態変化指定コマンドを送信バッファにセットする。

【0416】

(ステップS610-25)

メインCPU300aは、特別遊技管理フェーズを「01H」に更新し、当該特別図柄変動待ち処理を終了する。

【0417】

図38は、同時回し参考例に係る特別図柄当たり判定処理（S611）を説明するフローチャートである。

40

【0418】

(ステップS611-1)

メインCPU300aは、特別図柄確率状態フラグをロードする。

【0419】

(ステップS611-3)

メインCPU300aは、設定値バッファの登録設定値をロードする。

【0420】

(ステップS611-5)

50

メインCPU300aは、上記ステップS611-3でロードした登録設定値が正常範囲内の値であるかを判定する。その結果、正常範囲内の値であると判定した場合にはステップS611-11に処理を移し、正常範囲内の値ではないと判定した場合にはステップS611-7に処理を移す。

【0421】

(ステップS611-7)

メインCPU300aは、遊技機状態フラグに03H（設定異常状態）をセットする。

【0422】

(ステップS611-9)

メインCPU300aは、設定異常状態コマンド（サブコマンド）を送信バッファにセットし、当該特別図柄当たり判定処理を終了する。この設定異常状態コマンドが副制御基板330に送信されると、設定異常である旨の報知がなされる。

【0423】

(ステップS611-11)

メインCPU300aは、上記ステップS611-1およびステップS611-3でロードした情報に対応する大当たり決定乱数判定テーブルを参照し、大当たりまたは小当たりと判定する際の下限值および上限値をそれぞれセットする。

【0424】

(ステップS611-13)

メインCPU300aは、第0記憶部に転送された大当たり決定乱数と、上記の下限值および上限値とを比較し、大当たりまたは小当たりの当選有無を判定する判定処理（大役抽選）を行う。

【0425】

(ステップS611-15)

メインCPU300aは、非対象特別図柄が大当たり図柄であるかを判定する。その結果、大当たり図柄であると判定した場合にはステップS611-17に処理を移し、大当たり図柄でないと判定した場合にはステップS611-21に処理を移す。

【0426】

(ステップS611-17)

メインCPU300aは、上記ステップS611-13における大役抽選結果が小当たりまたはハズレであるかを判定する。その結果、小当たりまたはハズレであると判定した場合にはステップS611-21に処理を移し、小当たりおよびハズレでないと判定した場合にはステップS611-19に処理を移す。

【0427】

(ステップS611-19)

メインCPU300aは、上記ステップS611-13における大役抽選結果をハズレに変更する。

【0428】

(ステップS611-21)

メインCPU300aは、上記ステップS611-13の判定処理の結果、または、上記ステップS611-19で変更された結果を判定情報としてセットし、当該特別図柄当たり判定処理を終了する。

【0429】

図39は、同時回し参考例に係る主制御基板300における特別図柄変動番号決定処理（ステップS612）を説明するフローチャートである。

【0430】

(ステップS612-1)

メインCPU300aは、上記ステップS611における大役抽選の結果が大当たりまたは小当たりであるかを判定する。その結果、大当たりまたは小当たりであると判定した場合にはステップS612-3に処理を移し、大当たりおよび小当たりのいずれでもない

10

20

30

40

50

(ハズレである)と判定した場合にはステップS 6 1 2 - 5 に処理を移す。

【0 4 3 1】

(ステップS 6 1 2 - 3)

メインCPU 3 0 0 a は、現在の遊技状態および保留種別に対応するリーチモード決定乱数判定テーブルをセットする。

【0 4 3 2】

(ステップS 6 1 2 - 5)

メインCPU 3 0 0 a は、読み出した保留の保留種別が特2保留である場合には、特別図柄2保留球数カウンタのカウンタ値を確認し、読み出した保留の保留種別が特1保留である場合には、特別図柄1保留球数カウンタのカウンタ値を確認する。

10

【0 4 3 3】

(ステップS 6 1 2 - 7)

メインCPU 3 0 0 a は、現在の遊技状態、上記ステップS 6 1 2 - 5 で確認した保留数、保留種別に基づいて、対応するリーチグループ決定乱数判定テーブルをセットする。そして、セットしたリーチグループ決定乱数判定テーブルと、上記ステップS 6 1 0 - 9 で第0記憶部に転送したリーチグループ決定乱数とに基づいて、リーチグループ(グループ種別)を決定する。

【0 4 3 4】

(ステップS 6 1 2 - 9)

メインCPU 3 0 0 a は、上記ステップS 6 1 2 - 7 で決定されたグループ種別に対応するハズレ時リーチモード決定乱数判定テーブルをセットする。

20

【0 4 3 5】

(ステップS 6 1 2 - 1 1)

メインCPU 3 0 0 a は、上記ステップS 6 1 2 - 3 または上記ステップS 6 1 2 - 9 でセットしたリーチモード決定乱数判定テーブルと、上記ステップS 6 1 0 - 9 で第0記憶部に転送したリーチモード決定乱数とに基づいて、変動モード番号を決定する。また、ここでは、変動モード番号とともに、変動パターン乱数判定テーブルが決定される。

【0 4 3 6】

(ステップS 6 1 2 - 1 3)

メインCPU 3 0 0 a は、上記ステップS 6 1 2 - 1 1 で決定した変動モード番号に対応する変動モードコマンドを送信バッファにセットする。

30

【0 4 3 7】

(ステップS 6 1 2 - 1 5)

メインCPU 3 0 0 a は、上記ステップS 6 1 2 - 1 1 で決定した変動パターン乱数判定テーブルと、上記ステップS 6 1 0 - 9 で第0記憶部に転送した変動パターン乱数とに基づいて、変動パターン番号を決定する。

【0 4 3 8】

(ステップS 6 1 2 - 1 7)

メインCPU 3 0 0 a は、上記ステップS 6 1 2 - 1 5 で決定した変動パターン番号に対応する変動パターンコマンドを送信バッファにセットして、当該特別図柄変動番号決定処理を終了する。

40

【0 4 3 9】

図40は、同時回し参考例に係る主制御基板300における回数切り管理処理(ステップS 6 1 3)を説明するフローチャートである。

【0 4 4 0】

(ステップS 6 1 3 - 1)

メインCPU 3 0 0 a は、時短回数、すなわち、時短回数切りカウンタのカウンタ値が0よりも大きいかを判定する。その結果、時短回数が0よりも大きいと判定した場合にはステップS 6 1 3 - 3 に処理を移し、時短回数が0であると判定した場合には、当該回数切り管理処理を終了する。

50

【0441】

(ステップS613-3)

メインCPU300aは、時短回数切りカウンタをデクリメントする。

【0442】

(ステップS613-5)

メインCPU300aは、上記ステップS613-3において、カウンタ値（時短回数）が0に更新されたかを判定する。その結果、時短回数が0であると判定した場合にはステップS613-7に処理を移し、時短回数は0ではないと判定した場合には当該回数切り管理処理を終了する。

【0443】

(ステップS613-7)

メインCPU300aは、普通遊技状態を非時短遊技状態に設定するべく、時短状態フラグの設定を行う。これにより、普通遊技状態が時短遊技状態に設定された後、変動回数が時短回数（ここでは50回または100回）に到達したときの変動開始時に、普通遊技状態が非時短遊技状態に変更されることとなる。例えば、高確前兆状態に設定されていた場合には、最優位状態に設定されることとなる。

【0444】

(ステップS613-9)

メインCPU300aは、時短終了フラグをオンして、当該回数切り管理処理を終了する。

【0445】

図41は、同時回し参考例に係る主制御基板300における特別図柄変動中処理を説明するフローチャートである。

【0446】

(ステップS620-1)

メインCPU300aは、中断中フラグがオンしているかを判定する。なお、詳しくは後述するが、同時回し参考例では、第1特別図柄表示器160における図柄の変動表示中に、第2特別図柄表示器162に小当たり図柄が停止表示される場合がある。この場合、第2特別図柄表示器162に小当たり図柄が停止表示されると、小当たり遊技が実行されるが、この間、第1特別図柄表示器160における特別図柄の変動時間の減算が中断され、小当たり遊技の終了後に、第1特別図柄表示器160における図柄の変動表示が再開される。中断中フラグは、第2特別図柄表示器162に小当たり図柄が停止表示されたときに、第1特別図柄表示器160において図柄の変動表示中であつた場合にオンされる。ここでは、中断中フラグがオンであると判定した場合には当該特別図柄変動中処理を終了し、中断中フラグがオンではないと判定した場合にはステップS620-3に処理を移す。

【0447】

(ステップS620-3)

メインCPU300aは、特別図柄変動ベースカウンタを更新する処理を実行する。なお、特別図柄変動ベースカウンタは、所定周期（例えば100ms）で1周するようにカウンタ値が設定される。具体的には、特別図柄変動ベースカウンタのカウンタ値が「0」であつた場合には、所定のカウンタ値（例えば25）がセットされ、カウンタ値が「1」以上であつた場合には、現在のカウンタ値から「1」減算した値にカウンタ値を更新する。

【0448】

(ステップS620-5)

メインCPU300aは、上記ステップS620-3で更新した特別図柄変動ベースカウンタのカウンタ値が「0」であるかを判定する。その結果、カウンタ値が「0」であつた場合にはステップS620-7に処理を移し、カウンタ値が「0」ではなかつた場合にはステップS620-11に処理を移す。

【0449】

(ステップS620-7)

メインCPU300aは、上記ステップS610-15で設定された特別図柄変動タイマのタイマ値を所定値減算する特別図柄変動タイマ更新処理を行う。

【0450】

(ステップS620-9)

メインCPU300aは、上記ステップS620-7で更新した特別図柄変動タイマのタイマ値が「0」であるかを判定する。その結果、タイマ値が「0」であった場合にはステップS620-17に処理を移し、タイマ値が「0」ではなかった場合にはステップS620-11に処理を移す。

【0451】

(ステップS620-11)

メインCPU300aは、第1特別図柄表示器160および第2特別図柄表示器162を構成する7セグの各セグメントの点灯時間を計時する特別図柄表示タイマを更新する。具体的には、特別図柄表示タイマのタイマ値が「0」であった場合には、所定のタイマ値がセットされ、タイマ値が「1」以上であった場合には、現在のタイマ値から「1」減算した値にタイマ値を更新する。

【0452】

(ステップS620-13)

メインCPU300aは、特別図柄表示タイマのタイマ値が「0」であるかを判定する。その結果、特別図柄表示タイマのタイマ値が「0」であると判定した場合にはステップS620-15に処理を移し、特別図柄表示タイマのタイマ値が「0」でないと判定した場合には当該特別図柄変動中処理を終了する。

【0453】

(ステップS620-15)

メインCPU300aは、更新対象の特別図柄表示図柄カウンタのカウンタ値を更新し、当該特別図柄変動中処理を終了する。これにより、7セグを構成する各セグメントが、所定時間おきに順次点灯することとなる。

【0454】

(ステップS620-17)

メインCPU300aは、非対象特別図柄が変動表示中であるかを判定する。その結果、非対象特別図柄が変動表示中であると判定した場合にはステップS621に処理を移し、非対象特別図柄が変動表示中でないと判定した場合にはステップS620-19に処理を移す。

【0455】

(ステップS621)

メインCPU300aは、図柄強制停止処理を実行する。この図柄強制停止処理については、図42を用いて後述する。

【0456】

(ステップS620-19)

メインCPU300aは、特別遊技管理フェーズを「02H」に更新する。

【0457】

(ステップS620-21)

メインCPU300aは、対象の特別図柄表示図柄カウンタに、上記ステップS610-13で決定した特別図柄停止図柄番号(カウンタ値)をセーブする。これにより、第1特別図柄表示器160または第2特別図柄表示器162に、決定された特別図柄が停止表示されることとなる。

【0458】

(ステップS620-23)

メインCPU300aは、第1特別図柄表示器160または第2特別図柄表示器162に特別図柄が停止表示されたことを示す特図停止指定コマンドを送信バッファにセットする。

10

20

30

40

50

【0459】

(ステップS620-25)

メインCPU300aは、特別図柄を停止表示する時間である特別図柄変動停止時間を特別遊技タイマにセットし、当該特別図柄変動中処理を終了する。

【0460】

図42は、同時回し参考例に係る主制御基板300における図柄強制停止処理（ステップS621）を説明するフローチャートである。

【0461】

(ステップS621-1)

メインCPU300aは、停止表示中（当該）の特別図柄が小当たり図柄であるかを判定する。その結果、小当たり図柄であると判定した場合にはステップS621-3に処理を移し、小当たり図柄ではないと判定した場合にはステップS621-11に処理を移す。

10

【0462】

(ステップS621-3)

メインCPU300aは、特別遊技特別図柄判定フラグが00Hであるか、つまり、小当たり図柄が停止表示されたのが、第1特別図柄表示器160であるかを判定する。その結果、特別遊技特別図柄判定フラグが00Hであると判定した場合にはステップS621-11に処理を移し、特別遊技特別図柄判定フラグが00Hではないと判定した場合にはステップS621-5に処理を移す。

20

【0463】

(ステップS621-5)

メインCPU300aは、変動表示中（他方）の特別図柄が大当たり図柄であるかを判定する。その結果、大当たり図柄であると判定した場合にはステップS621-11に処理を移し、大当たり図柄ではないと判定した場合にはステップS621-7に処理を移す。

【0464】

(ステップS621-7)

メインCPU300aは、中断中フラグをオンする。

【0465】

(ステップS621-9)

メインCPU300aは、特別図柄の変動表示を中断する変動中断処理を実行し、当該図柄強制停止処理を終了する。ここでは、変動時間の残り時間や、特別図柄にかかる情報を一時的に所定の記憶領域に退避させる処理を行う。

30

【0466】

(ステップS621-11)

メインCPU300aは、図柄が変動表示されている第1特別図柄表示器160または第2特別図柄表示器162にハズレ図柄を強制停止させるとともに、残りの変動時間を強制的に終了させるための変動時間特殊停止フラグをオンにする処理を行い、当該図柄強制停止処理を終了する。

【0467】

上記の処理により、第1特別図柄表示器160に小当たり図柄が停止表示された場合には、第2特別図柄表示器162にハズレ図柄が強制停止表示される。また、第2特別図柄表示器162に小当たり図柄が停止表示された場合、大当たり図柄が第1特別図柄表示器160に最終的に停止表示される変動表示中であれば、第1特別図柄表示器160にハズレ図柄が強制停止表示される。一方、第2特別図柄表示器162に小当たり図柄が停止表示された場合、小当たり図柄またはハズレ図柄が第1特別図柄表示器160に最終的に停止表示される変動表示中であれば、第1特別図柄表示器160における変動表示が一時的に中断されることとなる。

40

【0468】

図43は、同時回し参考例に係る主制御基板300における特別図柄停止図柄表示処理を説明するフローチャートである。

50

【0469】

(ステップS630-1)

メインCPU300aは、上記ステップS620-25でセットした特別遊技タイマのタイマ値が「0」でないかを判定する。その結果、特別遊技タイマのタイマ値が「0」ではないと判定した場合には当該特別図柄停止図柄表示処理を終了し、特別遊技タイマのタイマ値が「0」であると判定した場合にはステップS630-3に処理を移す。

【0470】

(ステップS630-3)

メインCPU300aは、変動時間特殊停止フラグがオンであるかを判定する。その結果、変動時間特殊停止フラグがオンであると判定した場合にはステップS630-5に処理を移し、変動時間特殊停止フラグがオンでないと判定した場合には当該特別図柄停止図柄表示処理を終了する。

10

【0471】

(ステップS630-5)

メインCPU300aは、大役抽選の結果を確認する。

【0472】

(ステップS630-7)

メインCPU300aは、大役抽選の結果がハズレであるかを判定する。その結果、ハズレであると判定した場合にはステップS630-27に処理を移し、ハズレではないと判定した場合にはステップS630-9に処理を移す。

20

【0473】

(ステップS630-9)

メインCPU300aは、遊技状態を更新する遊技状態更新処理を行う。ここでは、停止表示中の特別図柄が大当たり図柄である場合に遊技状態を初期状態に設定し、停止表示中の特別図柄が小当たり図柄である場合には、そのまま次の処理に移る。

【0474】

(ステップS630-11)

メインCPU300aは、確定した特別図柄の種別に応じて、特別電動役物作動ラムセットテーブルのデータをセットする。なお、メインCPU300aは、変動時間特殊停止フラグがオンである場合にはオフにする。

30

【0475】

(ステップS630-13)

メインCPU300aは、特別電動役物最大作動回数設定処理を行う。具体的には、上記ステップS630-11でセットしたデータを参照し、特別電動役物最大作動回数カウンタに、カウンタ値として所定数（特別図柄の種別に対応するカウンタ値＝ラウンド数）をセットする。なお、この特別電動役物最大作動回数カウンタは、これから開始する大役遊技において実行可能なラウンド数を示すものである。一方、メインRAM300cには、特別電動役物連続作動回数カウンタが設けられており、各ラウンド遊技の開始時に、特別電動役物連続作動回数カウンタのカウンタ値を「1」加算することで、現在のラウンド遊技数が管理される。ここでは、大役遊技の開始に伴って、この特別電動役物連続作動回数カウンタのカウンタ値をリセット（「0」に更新）する処理が併せて実行される。

40

【0476】

(ステップS630-15)

メインCPU300aは、上記ステップS630-11でセットしたデータを参照し、特別電動役物遊技タイマに、タイマ値として所定のオープニング時間をセーブする。

【0477】

(ステップS630-17)

メインCPU300aは、大役遊技の開始を副制御基板330に伝達するためのオープニング指定コマンドを送信バッファにセットする。

【0478】

50

(ステップ S 6 3 0 - 1 9)

メイン CPU 3 0 0 a は、大役遊技を開始する場合には特別電動役物遊技管理フェーズを「0 1 H」に更新し、小当たり遊技を開始する場合には特別電動役物遊技管理フェーズを「0 5 H」に更新する。

【0 4 7 9】

(ステップ S 6 3 0 - 2 1)

メイン CPU 3 0 0 a は、特別遊技管理フェーズを「0 0 H」に更新する。

【0 4 8 0】

(ステップ S 6 3 0 - 2 3)

メイン CPU 3 0 0 a は、上記ステップ S 6 3 0 - 1 9 で更新した特別電動役物遊技管理フェーズが「0 1 H」であるか、すなわち、大当たりかを判定する。その結果、特別電動役物遊技管理フェーズが「0 1 H」であると判定した場合にはステップ S 6 3 0 - 2 5 に処理を移し、特別電動役物遊技管理フェーズは「0 1 H」ではないと判定した場合には、ステップ S 6 3 0 - 2 7 に処理を移す。

【0 4 8 1】

(ステップ S 6 3 0 - 2 5)

メイン CPU 3 0 0 a は、遊技情報出力端子板 3 1 2 から大当たり信号を出力するための大当たり信号出力開始処理を行い、当該特別図柄停止図柄表示処理を終了する。この処理により、大役遊技（オープニング）の開始に伴って大当たり信号が出力されることとなる。なお、遊技情報出力端子板 3 1 2 から出力される信号は複数設けられているが、ここでは、所定の大当たり信号についてのみ説明する。

【0 4 8 2】

(ステップ S 6 3 0 - 2 7)

メイン CPU 3 0 0 a は、特別図柄が確定したときの遊技状態を示す特図確定時遊技状態確認指定コマンドを送信バッファにセットする。

【0 4 8 3】

(ステップ S 6 3 0 - 2 9)

メイン CPU 3 0 0 a は、時短終了フラグがオンしているかを判定する。上記したように、時短終了フラグは、時短遊技状態から非時短遊技状態に変更される際の変動開始時に、図 4 0 のステップ S 6 1 3 - 9 においてオンされる。すなわち、ここで時短終了フラグがオンしている場合というのは、高確前兆状態における 5 0 回目もしくは 1 0 0 回目の変動開始時に、時短抜けによって、普通遊技状態が時短遊技状態から非時短遊技状態に変更された場合である。つまり、ここでは、時短抜け時の変動が終了するときに限り、時短終了フラグがオンしていると判定されることとなる。時短終了フラグがオンしていると判定した場合には、ステップ S 6 3 0 - 3 1 に処理を移し、時短終了フラグはオンしていないと判定した場合にはステップ S 6 3 0 - 3 5 に処理を移す。

【0 4 8 4】

(ステップ S 6 3 0 - 3 1)

メイン CPU 3 0 0 a は、遊技情報出力端子板 3 1 2 から出力されている大当たり信号を停止するための大当たり信号出力停止処理を行う。すなわち、大当たり信号は、大役遊技中もしくは時短遊技状態中に出力されることとなる。

【0 4 8 5】

(ステップ S 6 3 0 - 3 3)

メイン CPU 3 0 0 a は、時短終了フラグをオフする。

【0 4 8 6】

(ステップ S 6 3 0 - 3 5)

メイン CPU 3 0 0 a は、特別遊技管理フェーズを「0 0 H」に更新し、当該特別図柄停止図柄表示処理を終了する。

【0 4 8 7】

図 4 4 は、同時回し参考例に係る主制御基板 3 0 0 における特別電動役物遊技管理処理

10

20

30

40

50

(ステップ S 7 0 0) を説明するフローチャートである。

【 0 4 8 8 】

(ステップ S 7 0 0 - 1)

メイン CPU 3 0 0 a は、特別電動役物遊技管理フェーズをロードする。

【 0 4 8 9 】

(ステップ S 7 0 0 - 3)

メイン CPU 3 0 0 a は、上記ステップ S 7 0 0 - 1 でロードした特別電動役物遊技管理フェーズが、「 0 0 H」であるかを判定する。すなわち、ここでは、現在、大役遊技中および小当たり遊技中でないかを判定することとなる。特別電動役物遊技管理フェーズが「 0 0 H」であると判定した場合には当該特別電動役物遊技管理処理を終了し、特別電動役物遊技管理フェーズが「 0 0 H」でないと判定した場合にはステップ S 7 0 0 - 5 に処理を移す。

10

【 0 4 9 0 】

(ステップ S 7 0 0 - 5)

メイン CPU 3 0 0 a は、上記ステップ S 7 0 0 - 1 でロードした特別電動役物遊技管理フェーズに対応する特別電動役物遊技制御モジュールを選択する。

【 0 4 9 1 】

(ステップ S 7 0 0 - 7)

メイン CPU 3 0 0 a は、上記ステップ S 7 0 0 - 5 で選択した特別電動役物遊技制御モジュールをコールして処理を開始する。

20

【 0 4 9 2 】

(ステップ S 7 0 0 - 9)

メイン CPU 3 0 0 a は、特別電動役物遊技の制御時間を管理する特別電動役物遊技タイマをロードし、当該特別電動役物遊技管理処理を終了する。

【 0 4 9 3 】

図 4 5 は、同時回し参考例に係る主制御基板 3 0 0 における大入賞口開放前処理を説明するフローチャートである。この大入賞口開放前処理は、特別電動役物遊技管理フェーズが「 0 1 H」、「 0 5 H」であった場合に実行される。

【 0 4 9 4 】

(ステップ S 7 1 0 - 1)

メイン CPU 3 0 0 a は、上記ステップ S 6 3 0 - 1 5 等でセットした特別電動役物遊技タイマのタイマ値が「 0」でないかを判定する。その結果、特別電動役物遊技タイマのタイマ値が「 0」ではないと判定した場合には当該大入賞口開放前処理を終了し、特別電動役物遊技タイマのタイマ値が「 0」であると判定した場合にはステップ S 7 1 0 - 3 に処理を移す。

30

【 0 4 9 5 】

(ステップ S 7 1 0 - 3)

メイン CPU 3 0 0 a は、特別電動役物連続作動回数カウンタのカウンタ値を、現在のカウンタ値に「 1」加算した値に更新する。

【 0 4 9 6 】

40

(ステップ S 7 1 0 - 5)

メイン CPU 3 0 0 a は、大入賞口の開放開始（ラウンド遊技の開始）を副制御基板 3 3 0 に伝達するための大入賞口開放指定コマンドを送信バッファにセットする。

【 0 4 9 7 】

(ステップ S 7 1 1)

メイン CPU 3 0 0 a は、大入賞口開閉切替処理を実行する。この大入賞口開閉切替処理については後述する。

【 0 4 9 8 】

(ステップ S 7 1 0 - 7)

メイン CPU 3 0 0 a は、特別電動役物遊技管理フェーズを、現在の値に 0 1 H を加算

50

した値（「02H」または「06H」）に更新し、当該大入賞口開放前処理を終了する。

【0499】

図46は、同時回し参考例に係る主制御基板300における大入賞口開閉切替処理（S711）を説明するフローチャートである。

【0500】

（ステップS711-1）

メインCPU300aは、特別電動役物開閉切替回数カウンタのカウンタ値が、特別電動役物開閉切替回数（1回のラウンド遊技中における大入賞口の開閉回数）の上限値であるかを判定する。その結果、カウンタ値が上限値であると判定した場合には当該大入賞口開閉切替処理を終了し、カウンタ値は上限値ではないと判定した場合にはステップS711-3に処理を移す。

10

【0501】

（ステップS711-3）

メインCPU300aは、特別電動役物作動ラムセットテーブルのデータを参照し、特別電動役物開閉切替回数カウンタのカウンタ値に基づいて、第1大入賞口ソレノイド126c、第2大入賞口ソレノイド128cを通電制御するためのソレノイド制御データ、および、通電時間もしくは通電停止時間であるタイマデータを抽出する。

【0502】

（ステップS711-5）

メインCPU300aは、上記ステップS711-3で抽出したソレノイド制御データに基づいて、第1大入賞口ソレノイド126cまたは第2大入賞口ソレノイド128cの通電を開始するか、もしくは、通電を停止するための大入賞口ソレノイド通電制御処理を実行する。この大入賞口ソレノイド通電制御処理の実行により、上記ステップS400-33およびステップS400-35において、第1大入賞口ソレノイド126cまたは第2大入賞口ソレノイド128cの通電開始もしくは通電停止の制御がなされることとなる。

20

【0503】

（ステップS711-7）

メインCPU300aは、上記ステップS711-3で抽出したタイマデータに基づくタイマ値を、特別電動役物遊技タイマにセーブする。なお、ここで特別電動役物遊技タイマにセーブされるタイマ値は、大入賞口の1回の最大開放時間となる。

30

【0504】

（ステップS711-9）

メインCPU300aは、第1大入賞口ソレノイド126cまたは第2大入賞口ソレノイド128cの通電開始状態か、すなわち、上記ステップS711-5において、第1大入賞口ソレノイド126cまたは第2大入賞口ソレノイド128cの通電を開始する制御処理がなされたかを判定する。その結果、通電開始状態であると判定した場合にはステップS711-11に処理を移し、通電開始状態ではないと判定した場合には当該大入賞口開閉切替処理を終了する。

【0505】

（ステップS711-11）

メインCPU300aは、特別電動役物開閉切替回数カウンタのカウンタ値を、現在のカウンタ値に「1」加算した値に更新し、当該大入賞口開閉切替処理を終了する。

40

【0506】

図47は、同時回し参考例に係る主制御基板300における大入賞口開放制御処理を説明するフローチャートである。この大入賞口開放制御処理は、特別電動役物遊技管理フェーズが「02H」、「06H」であった場合に実行される。

【0507】

（ステップS720-1）

メインCPU300aは、上記ステップS711-7でセーブした特別電動役物遊技タイマのタイマ値が「0」でないかを判定する。その結果、特別電動役物遊技タイマのタイ

50

マ値が「0」ではないと判定した場合にはステップS 7 2 0 - 5 に処理を移し、特別電動役物遊技タイマのタイマ値が「0」であると判定した場合にはステップS 7 2 0 - 3 に処理を移す。

【0508】

(ステップS 7 2 0 - 3)

メインCPU 300 a は、特別電動役物開閉切替回数カウンタのカウンタ値が、特別電動役物開閉切替回数の上限値であるかを判定する。その結果、カウンタ値が上限値であると判定した場合にはステップS 7 2 0 - 7 に処理を移し、カウンタ値は上限値ではないと判定した場合にはステップS 7 1 1 に処理を移す。

【0509】

(ステップS 7 1 1)

上記ステップS 7 2 0 - 3 において、特別電動役物開閉切替回数カウンタのカウンタ値が、特別電動役物開閉切替回数の上限値ではないと判定した場合には、メインCPU 300 a は、上記ステップS 7 1 1 の処理を実行する。

【0510】

(ステップS 7 2 0 - 5)

メインCPU 300 a は、上記ステップS 5 0 0 - 9 で更新された大入賞口入賞球数カウンタのカウンタ値が規定数に到達していないか、すなわち、大入賞口に、1 ラウンド中の最大入賞可能数と同数の遊技球が入球していないかを判定する。その結果、規定数に到達していないと判定した場合には当該大入賞口開放制御処理を終了し、規定数に到達したと判定した場合にはステップS 7 2 0 - 7 に処理を移す。

【0511】

(ステップS 7 2 0 - 7)

メインCPU 300 a は、第1大入賞口ソレノイド126 c または第2大入賞口ソレノイド128 c の通電を停止して大入賞口を閉鎖するために必要な大入賞口閉鎖処理を実行する。これにより、大入賞口は閉鎖状態となる。

【0512】

(ステップS 7 2 0 - 9)

メインCPU 300 a は、大入賞口閉鎖有効時間（インターバル時間）を特別電動役物遊技タイマにセーブする。

【0513】

(ステップS 7 2 0 - 11)

メインCPU 300 a は、特別電動役物遊技管理フェーズを、現在の値に01Hを加算した値（「03H」または「07H」）に更新する。

【0514】

(ステップS 7 2 0 - 13)

メインCPU 300 a は、大入賞口が閉鎖されたことを示す大入賞口閉鎖指定コマンドを送信バッファにセットし、当該大入賞口開放制御処理を終了する。

【0515】

図48は、同時回し参考例に係る主制御基板300における大入賞口閉鎖有効処理を説明するフローチャートである。この大入賞口閉鎖有効処理は、特別電動役物遊技管理フェーズが「03H」、「07H」であった場合に実行される。

【0516】

(ステップS 7 3 0 - 1)

メインCPU 300 a は、上記ステップS 7 2 0 - 9 でセーブした特別電動役物遊技タイマのタイマ値が「0」でないかを判定する。その結果、特別電動役物遊技タイマのタイマ値が「0」でないと判定した場合には当該大入賞口閉鎖有効処理を終了し、特別電動役物遊技タイマのタイマ値が「0」であると判定した場合にはステップS 7 3 0 - 3 に処理を移す。

【0517】

(ステップ S 7 3 0 - 3)

メイン CPU 3 0 0 a は、特別電動役物連続作動回数カウンタのカウンタ値が、特別電動役物最大作動回数カウンタのカウンタ値と一致するか、すなわち、予め設定された回数のラウンド遊技が終了したかを判定する。その結果、特別電動役物連続作動回数カウンタのカウンタ値が、特別電動役物最大作動回数カウンタのカウンタ値と一致すると判定した場合にはステップ S 7 3 0 - 9 に処理を移し、一致しないと判定した場合にはステップ S 7 3 0 - 5 に処理を移す。

【 0 5 1 8 】

(ステップ S 7 3 0 - 5)

メイン CPU 3 0 0 a は、特別電動役物遊技管理フェーズを「 0 1 H」に更新する。なお、特別電動役物遊技管理フェーズが 0 7 H である場合、すなわち、小当たり遊技の制御中は、小当たり遊技のラウンド遊技回数が「 1」であることから、上記ステップ S 7 3 0 - 3 で必ず Y E S と判定され、当該ステップに処理が移行することはない。

10

【 0 5 1 9 】

(ステップ S 7 3 0 - 7)

メイン CPU 3 0 0 a は、所定の大入賞口閉鎖時間を特別遊技タイマにセーブし、当該大入賞口閉鎖有効処理を終了する。これにより、次のラウンド遊技が開始されることとなる。

【 0 5 2 0 】

(ステップ S 7 3 0 - 9)

メイン CPU 3 0 0 a は、エンディング時間を特別電動役物遊技タイマにセーブするエンディング時間設定処理を実行する。

20

【 0 5 2 1 】

(ステップ S 7 3 0 - 1 1)

メイン CPU 3 0 0 a は、特別電動役物遊技管理フェーズを、現在の値に 0 1 H を加算した値（「 0 4 H」または「 0 8 H」）に更新する。

【 0 5 2 2 】

(ステップ S 7 3 0 - 1 3)

メイン CPU 3 0 0 a は、エンディングの開始を示すエンディング指定コマンドを送信バッファにセットし、当該大入賞口閉鎖有効処理を終了する。

30

【 0 5 2 3 】

図 4 9 は、同時回し参考例に係る主制御基板 3 0 0 における大入賞口終了ウェイト処理を説明するフローチャートである。この大入賞口終了ウェイト処理は、特別電動役物遊技管理フェーズが「 0 4 H」、「 0 8 H」であった場合に実行される。

【 0 5 2 4 】

(ステップ S 7 4 0 - 1)

メイン CPU 3 0 0 a は、上記ステップ S 7 3 0 - 9 でセーブした特別電動役物遊技タイマのタイマ値が「 0」でないかを判定する。その結果、特別電動役物遊技タイマのタイマ値が「 0」ではないと判定した場合には当該大入賞口終了ウェイト処理を終了し、特別電動役物遊技タイマのタイマ値が「 0」であると判定した場合にはステップ S 7 4 0 - 3 に処理を移す。

40

【 0 5 2 5 】

(ステップ S 7 4 0 - 3)

メイン CPU 3 0 0 a は、特別電動役物遊技管理フェーズが「 0 8 H」であるか、すなわち、小当たり遊技の終了であるかを判定する。その結果、特別電動役物遊技管理フェーズが「 0 8 H」であると判定した場合にはステップ S 7 4 0 - 1 1 に処理を移し、特別電動役物遊技管理フェーズは「 0 8 H」ではないと判定した場合にはステップ S 7 4 0 - 5 に処理を移す。

【 0 5 2 6 】

(ステップ S 7 4 0 - 5)

50

メインCPU300aは、大役遊技終了後の遊技状態を設定するための状態設定処理を実行する。ここでは、上記ステップS610-17で予備領域に設定した遊技状態や高確回数、時短回数をロードし、大役遊技後の遊技状態として各フラグの設定やカウンタ値をセットする。

【0527】

(ステップS740-7)

メインCPU300aは、上記ステップS740-5において、普通遊技状態が非時短遊技状態に設定されたかを判定する。その結果、非時短遊技状態に設定されたと判定した場合にはステップS740-9に処理を移し、非時短遊技状態に設定されていないと判定した場合にはステップS740-21に処理を移す。

10

【0528】

(ステップS740-9)

メインCPU300aは、遊技情報出力端子板312から出力されている大当たり信号を停止するための大当たり信号出力停止処理を行う。すなわち、大役遊技後に最優位状態に設定される場合には、大役遊技の終了とともに、大当たり信号の出力が停止されることとなる。

【0529】

(ステップS740-11)

メインCPU300aは、現在の遊技状態が高確前兆状態（高確率遊技状態かつ時短遊技状態）であるかを判定する。その結果、高確前兆状態であると判定した場合にはステップS740-13に処理を移し、高確前兆状態ではないと判定した場合にはステップS740-21に処理を移す。

20

【0530】

(ステップS740-13)

メインCPU300aは、停止表示された小当たり図柄が特別図柄Z1であるかを判定する。その結果、特別図柄Z1であると判定した場合にはステップS740-15に処理を移し、特別図柄Z1ではないと判定した場合にはステップS740-21に処理を移す。

【0531】

(ステップS740-15)

メインCPU300aは、普通遊技状態を非時短遊技状態に変更するべく、時短状態フラグをセットする。これにより、高確前兆状態において特別図柄Z1が決定された場合には、その小当たり遊技の終了時に、遊技状態が最優位状態に変更されることとなる。

30

【0532】

(ステップS740-17)

メインCPU300aは、時短回数カウンタをリセットするカウンタリセット処理を行う。

【0533】

(ステップS740-19)

メインCPU300aは、遊技情報出力端子板312から出力されている大当たり信号を停止するための大当たり信号出力停止処理を行う。すなわち、特別図柄Z1に当選して小当たり遊技後に最優位状態に設定される場合には、小当たり遊技の終了とともに、大当たり信号の出力が停止されることとなる。

40

【0534】

(ステップS740-21)

メインCPU300aは、大役遊技の終了後に設定される遊技状態を伝達するための遊技状態変化指定コマンドを送信バッファにセットする。

【0535】

(ステップS740-23)

メインCPU300aは、高確回数、時短回数に対応する回数コマンドを送信バッファにセットする。

50

【0536】

(ステップS740-25)

メインCPU300aは、特別電動役物遊技管理フェーズを「00H」に更新し、当該大入賞口終了ウェイト処理を終了する。これにより、特1保留または特2保留が記憶されている場合には、図柄の変動表示が再開されることとなる。

【0537】

図50は、同時回し参考例に係る普通遊技管理フェーズを説明する図である。既に説明したとおり、同時回し参考例では、ゲート124への遊技球の通過もしくは普図作動口125への遊技球の入球を契機とする普通遊技に係る処理が、段階的に、かつ、繰り返し実行されるが、主制御基板300では、こうした普通遊技に係る各処理を普通遊技管理フェーズによって管理している。

10

【0538】

図50に示すように、メインROM300bには、普通遊技を実行制御するための複数の普通遊技制御モジュールが格納されており、これら普通遊技制御モジュールごとに、普通遊技管理フェーズが対応付けられている。具体的には、普通遊技管理フェーズが「00H」である場合には、「普通図柄変動待ち処理」を実行するためのモジュールがコールされ、普通遊技管理フェーズが「01H」である場合には、「普通図柄変動中処理」を実行するためのモジュールがコールされ、普通遊技管理フェーズが「02H」である場合には、「普通図柄停止図柄表示処理」を実行するためのモジュールがコールされ、普通遊技管理フェーズが「03H」である場合には、「普通電動役物入賞口開放前処理」を実行するためのモジュールがコールされ、普通遊技管理フェーズが「04H」である場合には、「普通電動役物入賞口開放制御処理」を実行するためのモジュールがコールされ、普通遊技管理フェーズが「05H」である場合には、「普通電動役物入賞口閉鎖有効処理」を実行するためのモジュールがコールされ、普通遊技管理フェーズが「06H」である場合には、「普通電動役物入賞口終了ウェイト処理」を実行するためのモジュールがコールされる。

20

【0539】

図51は、同時回し参考例に係る主制御基板300における普通遊技管理処理（ステップS800）を説明するフローチャートである。

【0540】

(ステップS800-1)

メインCPU300aは、普通遊技管理フェーズをロードする。

30

【0541】

(ステップS800-3)

メインCPU300aは、上記ステップS800-1でロードした普通遊技管理フェーズに対応する普通遊技制御モジュールを選択する。

【0542】

(ステップS800-5)

メインCPU300aは、上記ステップS800-3で選択した普通遊技制御モジュールをコールして処理を開始する。

【0543】

(ステップS800-7)

メインCPU300aは、普通遊技の制御時間を管理する普通遊技タイマをロードする。

40

【0544】

図52は、同時回し参考例に係る主制御基板300における普通図柄変動待ち処理を説明するフローチャートである。この普通図柄変動待ち処理は、普通遊技管理フェーズが「00H」であった場合に実行される。

【0545】

(ステップS810-1)

メインCPU300aは、普通図柄保留球数カウンタのカウンタ値をロードし、カウンタ値が「0」であるか、すなわち、普図保留が「0」であるかを判定する。その結果、カ

50

ウンタ値が「0」であると判定した場合には当該普通図柄変動待ち処理を終了し、カウンタ値は「0」ではないと判定した場合にはステップS810-3に処理を移す。

【0546】

(ステップS810-3)

メインCPU300aは、普図保留記憶領域の第1記憶部～第4記憶部に記憶されている普図保留(当たり決定乱数)を、1つ序数の小さい記憶部にブロック転送する。具体的には、第2記憶部～第4記憶部に記憶されている普図保留を、第1記憶部～第3記憶部に転送する。また、メインRAM300cには、処理対象となる第0記憶部が設けられており、第1記憶部に記憶されている普図保留を、第0記憶部に転送する。なお、この普通図柄記憶エリアシフト処理においては、普通図柄保留球数カウンタのカウンタ値を「1」減算するとともに、普図保留が「1」減算したことを示す、普図保留減指定コマンドを送信バッファにセットする。

10

【0547】

(ステップS810-5)

メインCPU300aは、第0記憶部に転送された当たり決定乱数をロードし、現在の遊技状態に対応する当たり決定乱数判定テーブルを選択して普図抽選を行い、その抽選結果を記憶する普通図柄当たり判定処理を実行する。

【0548】

(ステップS810-7)

メインCPU300aは、上記ステップS810-5の普図抽選の結果に対応する普通図柄停止図柄番号をセーブする。なお、同時回し参考例では、普通図柄表示器168は1つのLEDランプで構成されており、当たりの場合には普通図柄表示器168を点灯させ、ハズレの場合には普通図柄表示器168を消灯させる。ここで決定する普通図柄停止図柄番号は、最終的に普通図柄表示器168を点灯するか否かを示すものであり、例えば、当たりに当選した場合には、普通図柄停止図柄番号として「0」が決定され、ハズレの場合には、普通図柄停止図柄番号として「1」が決定される。

20

【0549】

(ステップS810-9)

メインCPU300aは、現在の遊技状態を確認し、対応する普通図柄変動時間データテーブルを選択してセットする。

30

【0550】

(ステップS810-11)

メインCPU300aは、上記ステップS810-3で第0記憶部に転送した当たり決定乱数と、上記ステップS810-9でセットした普通図柄変動時間データテーブルとに基づいて、普通図柄変動時間を決定する。

【0551】

(ステップS810-13)

メインCPU300aは、上記ステップS810-11で決定した普通図柄変動時間を普通遊技タイマにセーブする。

【0552】

(ステップS810-15)

メインCPU300aは、普通図柄表示器168において、普通図柄の変動表示を開始するために、普通図柄表示図柄カウンタを設定する処理を実行する。この普通図柄表示図柄カウンタに、カウンタ値として例えば「0」が設定されている場合には普通図柄表示器168が点灯制御され、カウンタ値として「1」が設定されている場合には普通図柄表示器168が消灯制御される。ここでは、普通図柄の変動表示の開始時に所定のカウンタ値が普通図柄表示図柄カウンタに設定されることとなる。

40

【0553】

(ステップS810-17)

メインCPU300aは、普図保留記憶領域に記憶されている普図保留数を示す普図保

50

留指定コマンドを送信バッファにセットする。

【0554】

(ステップS810-19)

メインCPU300aは、上記ステップS810-7で決定された普通図柄停止図柄番号、すなわち、普通図柄当たり判定処理によって決定された図柄種別（当たり図柄またはハズレ図柄）に基づいて、普通図柄指定コマンドを送信バッファにセットする。

【0555】

(ステップS810-21)

メインCPU300aは、普通遊技管理フェーズを「01H」に更新し、当該普通図柄変動待ち処理を終了する。

10

【0556】

図53は、同時回し参考例に係る主制御基板300における普通図柄変動中処理を説明するフローチャートである。この普通図柄変動中処理は、普通遊技管理フェーズが「01H」であった場合に実行される。

【0557】

(ステップS820-1)

メインCPU300aは、上記ステップS810-13でセーブした普通遊技タイマのタイマ値が「0」であるかを判定する。その結果、タイマ値が「0」であった場合にはステップS820-9に処理を移し、タイマ値が「0」ではなかった場合にはステップS820-3に処理を移す。

20

【0558】

(ステップS820-3)

メインCPU300aは、普通図柄表示器168の点灯時間および消灯時間を計時する普通図柄表示タイマを更新する。具体的には、普通図柄表示タイマのタイマ値が「0」であった場合には、所定のタイマ値がセットされ、タイマ値が「1」以上であった場合には、現在のタイマ値から「1」減算した値にタイマ値を更新する。

【0559】

(ステップS820-5)

メインCPU300aは、普通図柄表示タイマのタイマ値が「0」であるかを判定する。その結果、普通図柄表示タイマのタイマ値が「0」であると判定した場合にはステップS820-7に処理を移し、普通図柄表示タイマのタイマ値が「0」でないと判定した場合には当該普通図柄変動中処理を終了する。

30

【0560】

(ステップS820-7)

メインCPU300aは、普通図柄表示図柄カウンタのカウンタ値を更新する。ここでは、普通図柄表示図柄カウンタのカウンタ値が、普通図柄表示器168の消灯を示すカウンタ値であった場合には点灯を示すカウンタ値に更新し、普通図柄表示器168の点灯を示すカウンタ値であった場合には消灯を示すカウンタ値に更新し、当該普通図柄変動中処理を終了する。これにより、普通図柄表示器168は、普通図柄変動時間に亘って、所定時間おきに点灯、消灯を繰り返す（点滅する）こととなる。

40

【0561】

(ステップS820-9)

メインCPU300aは、普通図柄表示図柄カウンタに、上記ステップS810-7で決定した普通図柄停止図柄番号（カウンタ値）をセーブする。これにより、普通図柄表示器168が最終的に点灯もしくは消灯制御され、普図抽選の結果が報知されることとなる。

【0562】

(ステップS820-11)

メインCPU300aは、普通図柄を停止表示する時間である普通図柄変動停止時間を普通遊技タイマにセットする。

【0563】

50

(ステップ S 8 2 0 - 1 3)

メイン CPU 3 0 0 a は、普通図柄の停止表示が開始されたことを示す普図停止指定コマンドを送信バッファにセットする。

【 0 5 6 4 】

(ステップ S 8 2 0 - 1 5)

メイン CPU 3 0 0 a は、普通遊技管理フェーズを「 0 2 H」に更新し、当該普通図柄変動中処理を終了する。

【 0 5 6 5 】

図 5 4 は、同時回し参考例に係る主制御基板 3 0 0 における普通図柄停止図柄表示処理を説明するフローチャートである。この普通図柄停止図柄表示処理は、普通遊技管理フェーズが「 0 2 H」であった場合に実行される。

【 0 5 6 6 】

(ステップ S 8 3 0 - 1)

メイン CPU 3 0 0 a は、上記ステップ S 8 2 0 - 1 1 でセットした普通遊技タイマのタイマ値が「 0」でないかを判定する。その結果、普通遊技タイマのタイマ値が「 0」ではないと判定した場合には当該普通図柄停止図柄表示処理を終了し、普通遊技タイマのタイマ値が「 0」であると判定した場合にはステップ S 8 3 0 - 3 に処理を移す。

【 0 5 6 7 】

(ステップ S 8 3 0 - 3)

メイン CPU 3 0 0 a は、普図抽選の結果を確認する。

【 0 5 6 8 】

(ステップ S 8 3 0 - 5)

メイン CPU 3 0 0 a は、普図抽選の結果が当たりであるかを判定する。その結果、当たりであると判定した場合にはステップ S 8 3 0 - 9 に処理を移し、当たりではない（ハズレである）と判定した場合にはステップ S 8 3 0 - 7 に処理を移す。

【 0 5 6 9 】

(ステップ S 8 3 0 - 7)

メイン CPU 3 0 0 a は、普通遊技管理フェーズを「 0 0 H」に更新し、当該普通図柄停止図柄表示処理を終了する。これにより、1 の普図保留に基づく普通遊技管理処理が終了し、普図保留が記憶されている場合には、次の保留に基づく普通図柄の変動表示を開始するための処理が行われることとなる。

【 0 5 7 0 】

(ステップ S 8 3 0 - 9)

メイン CPU 3 0 0 a は、開閉制御パターンテーブルのデータを参照し、普通遊技タイマに、タイマ値として普電開放前時間をセーブする。

【 0 5 7 1 】

(ステップ S 8 3 0 - 1 1)

メイン CPU 3 0 0 a は、普通遊技管理フェーズを「 0 3 H」に更新し、当該普通図柄停止図柄表示処理を終了する。これにより、第 1 可変始動口 1 2 0 B の開閉制御が開始されることとなる。

【 0 5 7 2 】

図 5 5 は、同時回し参考例に係る主制御基板 3 0 0 における普通電動役物入賞口開放前処理を説明するフローチャートである。この普通電動役物入賞口開放前処理は、普通遊技管理フェーズが「 0 3 H」であった場合に実行される。

【 0 5 7 3 】

(ステップ S 8 4 0 - 1)

メイン CPU 3 0 0 a は、上記ステップ S 8 3 0 - 9 でセットした普通遊技タイマのタイマ値が「 0」でないかを判定する。その結果、普通遊技タイマのタイマ値が「 0」ではないと判定した場合には当該普通電動役物入賞口開放前処理を終了し、普通遊技タイマのタイマ値が「 0」であると判定した場合にはステップ S 8 4 1 に処理を移す。

10

20

30

40

50

【0574】

(ステップS841)

メインCPU300aは、普通電動役物入賞口開閉切替処理を実行する。この普通電動役物入賞口開閉切替処理については後述する。

【0575】

(ステップS840-3)

メインCPU300aは、普通遊技管理フェーズを「04H」に更新し、当該普通電動役物入賞口開放前処理を終了する。

【0576】

図56は、同時回し参考例に係る主制御基板300における普通電動役物入賞口開閉切替処理を説明するフローチャートである。

10

【0577】

(ステップS841-1)

メインCPU300aは、普通電動役物開閉切替回数カウンタのカウンタ値が、普通電動役物開閉切替回数（1回の開閉制御中における第1可変始動口120Bの可動片120bの開閉回数）の上限値であるかを判定する。その結果、カウンタ値が上限値であると判定した場合には当該普通電動役物入賞口開閉切替処理を終了し、カウンタ値は上限値ではないと判定した場合にはステップS841-3に処理を移す。

【0578】

(ステップS841-3)

メインCPU300aは、開閉制御パターンテーブルのデータを参照し、普通電動役物開閉切替回数カウンタのカウンタ値に基づいて、普通電動役物ソレノイド120cを通電制御するためのソレノイド制御データ（通電制御データまたは通電停止制御データ）、および、普通電動役物ソレノイド120cの通電時間（ソレノイド通電時間）もしくは通電停止時間（普通電動役物有効時間＝休止時間）であるタイマデータを抽出する。

20

【0579】

(ステップS841-5)

メインCPU300aは、上記ステップS841-3で抽出したソレノイド制御データに基づいて、普通電動役物ソレノイド120cの通電を開始するか、もしくは、普通電動役物ソレノイド120cの通電を停止するための普通電動役物ソレノイド通電制御処理を実行する。この普通電動役物ソレノイド通電制御処理の実行により、上記ステップS400-33およびステップS400-35において、普通電動役物ソレノイド120cの通電開始もしくは通電停止の制御がなされることとなる。

30

【0580】

(ステップS841-7)

メインCPU300aは、上記ステップS841-3で抽出したタイマデータに基づくタイマ値を、普通遊技タイマにセーブする。なお、ここで普通遊技タイマにセーブされるタイマ値は、第1可変始動口120Bの1回の最大開放時間となる。

【0581】

(ステップS841-9)

メインCPU300aは、普通電動役物ソレノイド120cの通電開始状態か、すなわち、上記ステップS841-5において、普通電動役物ソレノイド120cの通電を開始する制御処理がなされたかを判定する。その結果、通電開始状態であると判定した場合にはステップS841-11に処理を移し、通電開始状態ではないと判定した場合には当該普通電動役物入賞口開閉切替処理を終了する。

40

【0582】

(ステップS841-11)

メインCPU300aは、普通電動役物開閉切替回数カウンタのカウンタ値を、現在のカウンタ値に「1」加算した値に更新する。

【0583】

50

図57は、同時回し参考例に係る主制御基板300における普通電動役物入賞口開放制御処理を説明するフローチャートである。この普通電動役物入賞口開放制御処理は、普通遊技管理フェーズが「04H」であった場合に実行される。

【0584】

(ステップS850-1)

メインCPU300aは、上記ステップS841-7でセーブした普通遊技タイマのタイマ値が「0」でないかを判定する。その結果、普通遊技タイマのタイマ値が「0」ではないと判定した場合にはステップS850-5に処理を移し、普通遊技タイマのタイマ値が「0」であると判定した場合にはステップS850-3に処理を移す。

【0585】

(ステップS850-3)

メインCPU300aは、普通電動役物開閉切替回数カウンタのカウンタ値が、普通電動役物開閉切替回数の上限值であるかを判定する。その結果、カウンタ値が上限値であると判定した場合にはステップS850-7に処理を移し、カウンタ値は上限値ではないと判定した場合にはステップS841に処理を移す。

【0586】

(ステップS841)

上記ステップS850-3において、普通電動役物開閉切替回数カウンタのカウンタ値が、普通電動役物開閉切替回数の上限值ではないと判定した場合には、メインCPU300aは、上記ステップS841の処理を実行する。

【0587】

(ステップS850-5)

メインCPU300aは、上記ステップS530-9で更新された普通電動役物入賞球数カウンタのカウンタ値が規定数に到達していないか、すなわち、第1可変始動口120Bに、1回の開閉制御中の最大入賞可能数と同数の遊技球が入球していないかを判定する。その結果、規定数に到達していないと判定した場合には当該普通電動役物入賞口開放制御処理を終了し、規定数に到達したと判定した場合にはステップS850-7に処理を移す。

【0588】

(ステップS850-7)

メインCPU300aは、普通電動役物ソレノイド120cの通電を停止して第1可変始動口120Bを閉鎖するために必要な普通電動役物閉鎖処理を実行する。これにより、第1可変始動口120Bが閉鎖状態となる。

【0589】

(ステップS850-9)

メインCPU300aは、普電有効状態時間を普通遊技タイマにセーブする。

【0590】

(ステップS850-11)

メインCPU300aは、普通遊技管理フェーズを「05H」に更新し、当該普通電動役物入賞口開放制御処理を終了する。

【0591】

図58は、同時回し参考例に係る主制御基板300における普通電動役物入賞口閉鎖有効処理を説明するフローチャートである。この普通電動役物入賞口閉鎖有効処理は、普通遊技管理フェーズが「05H」であった場合に実行される。

【0592】

(ステップS860-1)

メインCPU300aは、上記ステップS850-9でセーブした普通遊技タイマのタイマ値が「0」でないかを判定する。その結果、普通遊技タイマのタイマ値が「0」ではないと判定した場合には当該普通電動役物入賞口閉鎖有効処理を終了し、普通遊技タイマのタイマ値が「0」であると判定した場合にはステップS860-3に処理を移す。

10

20

30

40

50

【0593】

(ステップS860-3)

メインCPU300aは、普電終了ウェイト時間を普通遊技タイマにセーブする。

【0594】

(ステップS860-5)

メインCPU300aは、普通遊技管理フェーズを「06H」に更新し、当該普通電動役物入賞口閉鎖有効処理を終了する。

【0595】

図59は、同時回し参考例に係る主制御基板300における普通電動役物入賞口終了ウェイト処理を説明するフローチャートである。この普通電動役物入賞口終了ウェイト処理は、普通遊技管理フェーズが「06H」であった場合に実行される。

10

【0596】

(ステップS870-1)

メインCPU300aは、上記ステップS860-3でセーブした普通遊技タイマのタイマ値が「0」でないかを判定する。その結果、普通遊技タイマのタイマ値が「0」ではないと判定した場合には当該普通電動役物入賞口終了ウェイト処理を終了し、普通遊技タイマのタイマ値が「0」とであると判定した場合にはステップS870-3に処理を移す。

【0597】

(ステップS870-3)

メインCPU300aは、普通遊技管理フェーズを「00H」に更新し、当該普通電動役物入賞口終了ウェイト処理を終了する。これにより、普図保留が記憶されている場合には、普通図柄の変動表示が再開されることとなる。次に、演出参考例として、上記した一種遊技機、一種二種遊技機、および、同時回し機において実行可能な具体的な演出や当該演出に係る具体的な処理を説明する。

20

【0598】

〈演出参考例〉

図60は、演出参考例に係るリーチなし変動パターンの変動演出の一例を説明する図である。上記のように、主制御基板300において大役抽選が行われると、特別図柄の変動表示中、すなわち、特別図柄の変動時間に亘って、大役抽選の結果を報知する変動演出が実行される。この変動演出においては、メイン演出表示部200aにおいて種々の背景画像が表示されるとともに、この背景画像に重畳して、演出図柄210a、210b、210cが表示される。なお、変動演出中は、メイン演出表示部200aに表示される画像に伴って、音声出力装置206から音声が出力されるとともに、演出照明装置204が点灯制御され、また、演出役物装置202が可動制御されるが、ここでは詳細な説明は省略する。

30

【0599】

演出参考例に係る変動演出は、リーチなし変動パターン、リーチ変動パターンに大別される。リーチなし変動パターンの変動演出では、メイン演出表示部200aに背景画像(図示を省略)が表示されるとともに、この背景画像に演出図柄210a、210b、210cが重畳して変動表示される。例えば、図60(a)に示すように、大役抽選結果がハズレであったことを示す組み合わせで演出図柄210a、210b、210cが停止表示されているとする。この状態で、新たに特別図柄の変動表示が行われると、当該特別図柄の変動表示の開始に伴って、図60(b)に示すように、3つの演出図柄210a、210b、210cが変動表示(スクロール表示)を開始する。なお、図中下向きの白抜き矢印は、演出図柄210a、210b、210cが高さ方向にスクロール表示されていることを示している。

40

【0600】

そして、図60(c)に示すように、まず、演出図柄210aが停止表示され、その後、図60(d)に示すように、演出図柄210aと異なる演出図柄210cが停止表示される。そして、特別図柄の変動表示が終了して、第1特別図柄表示器160または第2特

50

別図柄表示器 162 に特別図柄が停止表示するのとほぼ同じタイミングで、図 60 (e) に示すように、演出図柄 210b が停止表示され、このときの 3 つの演出図柄 210a、210b、210c の最終的な停止表示態様によって、大役抽選結果が遊技者に報知される。

【0601】

図 61 は、演出参考例に係るノーマルリーチ変動パターンの変動演出の一例を説明する図である。演出参考例では、リーチ変動パターンが、ノーマルリーチ変動パターン、発展リーチ変動パターン、擬似連続リーチ変動パターンに大別される。ノーマルリーチ変動パターンの変動演出は、リーチなし変動パターンの変動演出と同様に、特別図柄の変動表示の開始に伴って、演出図柄 210a、210b、210c の変動表示が開始され、図 61 (a) に示すように、演出図柄 210a がまず停止表示される。その後、図 61 (b) に示すように、演出図柄 210a と同一の演出図柄 210c が停止表示される。

10

【0602】

このように、メイン演出表示部 200a において、同一の演出図柄 210a、210c が停止表示されるリーチ態様で表示されると、図 61 (c) に示すように、メイン演出表示部 200a において、演出図柄 210a、210c に重畳して「リーチ」と表示される。なお、リーチ態様は複数種類設けられており、「1」～「9」のいずれかの数字が記された同一の演出図柄 210a、210c が停止表示される。その後、図 61 (d) に示すように、演出図柄 210a、210c の形状を、リーチ態様になる前と異にして変動表示が継続される。そして、図 61 (e) に示すように、最終的に、演出図柄 210a、210c と異なる演出図柄 210b が停止表示され、大役抽選の結果がハズレであったことが遊技者に報知される。

20

【0603】

図 62 は、演出参考例に係るハズレ時の発展リーチ変動パターンの変動演出の一例を説明する図であり、図 63 は、演出参考例に係る大当たり時の発展リーチ変動パターンの変動演出の一例を説明する図である。発展リーチ変動パターンの変動演出は、図 62 (a) ～ (d) および図 63 (a) ～ (d) に示すように、ノーマルリーチ変動パターンの変動演出と同様に、メイン演出表示部 200a において、演出図柄 210a、210c がリーチ態様で表示され、その後、所定の発展画像（動画）が再生表示されるリーチ発展演出が実行される。このリーチ発展演出では、例えば、図 62 (e) および図 63 (e) に示すように、メイン演出表示部 200a にミッションが表示されるとともに、図 62 (f)、(g) および図 63 (f)、(g) に示すように、ミッションの達成に向けた画像が表示される。

30

【0604】

ここで、リーチ発展演出用の発展画像は、ハズレパターンと大当たりパターンとに大別され、ハズレパターンの発展画像では、図 62 (h) に示すように、ミッションの失敗を示す画像が最終的に表示され、その後、図 62 (i) に示すように、演出図柄 210a、210b、210c がハズレを報知する組み合わせで停止表示される。一方、大当たりパターンの発展画像では、図 63 (h) に示すように、ミッションの成功を示す画像が最終的に表示され、その後、図 63 (i) に示すように、演出図柄 210a、210b、210c が大当たりを報知する組み合わせで停止表示される。

40

【0605】

なお、リーチ発展演出は、例えば、上記のように、ミッションに挑む内容の発展画像が表示されるミッション演出と、味方キャラクタと敵キャラクタとが対戦する発展画像が表示されるバトル演出と、が設けられている。そして、ミッション演出は、ミッションの内容を異にする複数の実行パターンが設けられており、バトル演出は、登場キャラクタや対戦方法を異にする複数の実行パターンが設けられている。また、上記のように、ミッション演出の実行パターンは、ミッションを達成する大当たりパターンと、ミッションに失敗するハズレパターンとに大別されるが、バトル演出の実行パターンも同様に、味方キャラクタが敵キャラクタに勝利する大当たりパターンと、味方キャラクタが敵キャラクタに敗

50

北するハズレパターンとに大別される。

【0606】

大当たりパターンとハズレパターンとは、演出の終盤までは同一内容で構成されており、最終的に味方キャラクタが勝利するか敗北するか、あるいは、ミッションを達成するか否かといった点を異にしている。したがって、リーチ発展演出中は、変動演出の終盤まで、大役抽選の結果を遊技者が識別することができず、遊技者に大当たりの期待感が付与されることとなる。

【0607】

なお、大当たりパターンは、大役抽選の結果が大当たりであった場合にのみ選択され、ハズレパターンは、大役抽選の結果がハズレであった場合にのみ選択される。ただし、1回の変動演出において、リーチ発展演出が2回実行されることもあり、この場合には、1回目のリーチ発展演出がハズレパターンで実行され、2回目のリーチ発展演出が、ハズレパターンまたは大当たりパターンで実行される。以下に、1回の変動演出において、リーチ発展演出が2回実行される場合の演出の流れについて説明する。

【0608】

図64は、演出参考例に係るリーチ発展演出が2回実行される場合の変動演出の一例を説明する図である。例えば、演出図柄210a、210cがリーチ態様で表示された後、図64(a)、(b)に示すように、ミッション演出が実行されたとする。ここまでは、1回の変動演出においてリーチ発展演出が1回のみ実行される場合と相違はないが、ミッションを達成できなかったことが報知された直後に、図64(c)に示すように、メイン演出表示部200aに「REACH UP」と表示される。

【0609】

その後、メイン演出表示部200aには、図64(d)に示すように、バトル演出用の発展画像が表示されており、2回目のリーチ発展演出が開始される。このバトル演出用の発展画像は、味方キャラクタと敵キャラクタとが対戦する内容となっており、大当たり当選時には、図64(e)に示すように、最終的に味方キャラクタが敵キャラクタに勝利するとともに、図64(f)に示すように、演出図柄210a、210b、210cが大当たりを報知する組み合わせで停止表示される。一方、ハズレ時には、図64(g)に示すように、最終的に味方キャラクタが敵キャラクタに敗北するとともに、図64(h)に示すように、演出図柄210a、210b、210cがハズレを報知する組み合わせで停止表示される。

【0610】

図65は、演出参考例に係る擬似連続リーチ変動パターンの変動演出の一例を説明する図である。擬似連続リーチ変動パターンの変動演出は、図65(a)に示すように、演出図柄210a、210b、210cの変動表示が開始されると、図65(b)に示すように、演出図柄210a、210b、210cが、予め設けられた複数種類の擬似態様のうちのいずれかで仮停止表示される。この擬似態様は、例えば、同一の演出図柄210a、210bと、これら演出図柄210a、210bよりも、「2」大きな数字が記された演出図柄210cとが仮停止表示されるものである。

【0611】

演出図柄210a、210b、210cが擬似態様で仮停止表示されると、図65(c)に示すように、演出図柄210a、210b、210cの変動表示が再開される。つまり、擬似態様は、演出図柄210a、210b、210cの再変動表示を示すものと言える。その後、図65(d)に示すように、演出図柄210a、210b、210cが、再び擬似態様で仮停止表示される。

【0612】

そして、図65(e)に示すように、演出図柄210a、210b、210cの変動表示が再開されると、図65(f)に示すように、演出図柄210a、210cがリーチ態様で表示され、以後、図65(g)～(i)に示すように、発展リーチ変動パターンと同様にリーチ発展演出が実行され、大役抽選の結果が遊技者に報知される。

【0613】

このように、擬似連続リーチ変動パターンの変動演出は、演出図柄210a、210cがリーチ態様となるまでの内容が、発展リーチ変動パターンの変動演出と異なっており、リーチ態様となった後は、発展リーチ変動パターンと同様に変動演出が進行されることとなる。

【0614】

なお、擬似連続リーチ変動パターンにおいて、リーチ態様となるまでの演出図柄210a、210b、210cの変動表示パターンは複数パターン設けられており、変動表示パターンごとに、演出図柄210a、210b、210cの仮停止表示の回数、換言すれば、演出図柄210a、210b、210cの変動表示回数が異なっている。この変動表示パターンは、変動モードコマンドによって決定され、演出図柄210a、210b、210cの仮停止表示（変動表示）の回数が多くなるほど、最終的に大当たりの当選が報知される可能性（以下「信頼度」という）が高くなるように、大当たり当選時およびハズレ時における変動モードコマンドの選択比率が設定されている。

10

【0615】

具体的には、大役抽選の結果が大当たりであった場合には、変動表示回数の多い変動モードコマンドの選択比率が、変動表示回数の少ない変動モードコマンドの選択比率よりも高く設定されており、大役抽選の結果がハズレであった場合には、変動表示回数の少ない変動モードコマンドの選択比率が、変動表示回数の多い変動モードコマンドの選択比率よりも高く設定されている。

20

【0616】

また、主制御基板300においては、擬似連続リーチ変動パターンの信頼度が、発展リーチ変動パターンの信頼度よりも高くなるように設定されている。したがって、演出図柄210a、210b、210cの仮停止表示（変動表示）の回数によって信頼度が示唆されることとなり、遊技者は、演出図柄210a、210b、210cがより多く仮停止表示（変動表示）されることを期待しながら、演出の行方を見守ることとなる。

【0617】

上記した変動演出の実行パターンは、主制御基板300で決定された変動コマンドに基づいて、副制御基板330において決定、実行制御される。つまり、変動演出の実行パターンは、主制御基板300と副制御基板330とで協働して決定されると言える。

30

【0618】

図66は、演出参考例に係る変動演出決定テーブルを説明する図であり、図66(a)には前半変動演出決定テーブルを示し、図66(b)には後半変動演出決定テーブルを示す。上記したように、主制御基板300において大役抽選が行われると、大役抽選の結果に基づいて、変動コマンドが決定され、決定された各コマンドが副制御基板330に送信される。副制御基板330においては、変動モードコマンドを受信すると、0～249の範囲から1の演出乱数を取得するとともに、前半変動演出決定テーブルを参照して、取得した演出乱数と、受信した変動モードコマンドとに基づいて、前半の変動演出の実行パターンを決定する。また、変動パターンコマンドを受信すると、0～249の範囲から1の演出乱数を取得するとともに、後半変動演出決定テーブルを参照して、取得した演出乱数と、受信した変動パターンコマンドとに基づいて、後半の変動演出の実行パターンを決定する。なお、図66においては、前半変動演出決定テーブルおよび後半変動演出決定テーブルの一部のみを抽出して示している。

40

【0619】

図66に示すように、前半変動演出決定テーブルによれば、変動モード番号（変動モードコマンド）ごとに、前半の変動演出の実行パターンについての選択比率がそれぞれ設定され、後半変動演出決定テーブルによれば、変動パターン番号（変動パターンコマンド）ごとに、後半の変動演出の実行パターンについての選択比率がそれぞれ設定されている。そして、決定された前半および後半の変動演出の実行パターンを組み合わせることで、1回の変動演出が実行されることとなる。

50

【0620】

リーチなし変動パターンの変動演出は、前半の実行パターンとして、前半の変動演出を実行しないことを示す「なし」が決定され、後半の実行パターンとして、リーチなし変動パターンに対応する「ノーマルハズレ1」、「ノーマルハズレ2」、「特殊ハズレ1」、「特殊ハズレ2」が決定された場合に実行される。例えば、前半の変動演出が実行されないことを示す「01H」の変動モード番号に対応する変動モードコマンドを受信すると、副制御基板330では、必ず、前半の実行パターンとして「なし」が決定される。また、このとき、同時に受信し得る変動パターンコマンドには、「ノーマルハズレ1」、「ノーマルハズレ2」、「特殊ハズレ1」、「特殊ハズレ2」のいずれかのみが決定されるように、後半変動演出決定テーブルにおいて選択比率の設定がなされている。したがって、前半の実行パターンとして「なし」が決定され、後半の実行パターンとして「ノーマルハズレ1」、「ノーマルハズレ2」、「特殊ハズレ1」、「特殊ハズレ2」が決定されることで、変動演出の実行パターンは、上記のリーチなし変動パターンに決定されることとなる。

10

【0621】

一方、リーチ変動パターンの変動演出は、前半の実行パターンとして「なし」以外が決定され、後半の実行パターンとして、いずれかのリーチ発展演出（図中、発展1～5で示す）が決定された場合に実行される。換言すれば、メイン演出表示部200aにおいて、リーチ変動パターンの変動演出が実行される場合には、必ず、変動モード番号＝01H以外の変動モード番号に対応する変動モードコマンドを受信しており、発展1～5のいずれかが決定される変動パターン番号に対応する変動パターンコマンドを受信していることとなる。

20

【0622】

ここで、図66(a)において、前半の実行パターンにおける「ノーマルリーチ1」や「ノーマルリーチ2」等は、それぞれ、ノーマルリーチ変動パターンの変動演出のうち、演出図柄210a、210b、210cがリーチ態様になるまで、より詳細には、リーチ発展演出が開始されるまでの、メイン演出表示部200aに表示される背景画像や演出図柄210a、210b、210cの変動表示パターンを示している。これらの画像パターンは、変動モード番号に対応付けられた特別図柄の変動表示の時間と一致するように予め設計されており、例えば、「ノーマルリーチ1」が決定されると、図61(a)～(d)に示す画像がメイン演出表示部200aに表示されることとなる。

30

【0623】

また、図66(a)において、前半の実行パターンにおける「擬似2a」等は、擬似連続リーチ変動パターンの変動演出のうち、リーチ発展演出が開始されるまでの、メイン演出表示部200aに表示される主変動演出画像の表示パターン、すなわち、演出図柄210a、210b、210cが変動表示される図柄表示演出の実行パターンを示している。例えば、「擬似2a」は、演出図柄210a、210b、210cの変動表示回数が2回である「擬似2」の擬似連続リーチ変動パターンであって、主変動演出画像が表示パターンaであることを示している。また、「擬似3b」は、演出図柄210a、210b、210cの変動表示回数が3回である「擬似3」の擬似連続リーチ変動パターンであって、主変動演出画像が表示パターンbであることを示している。

40

【0624】

なお、図66に示す前半変動演出決定テーブルおよび後半変動演出決定テーブルにおいては、リーチなし変動パターンおよびノーマルリーチ変動パターンの変動演出は、大役抽選の結果がハズレであった場合にのみ実行されるように、選択比率の設定がなされている。また、発展リーチ変動パターンおよび擬似連続リーチ変動パターンは、ハズレ時および大当たり時の双方で決定されるが、発展リーチ変動パターンは、擬似連続リーチ変動パターンよりも、ハズレ時の選択比率が高く、大当たり時の選択比率が低く設定されている。このように、ハズレ時と大当たり時とで選択比率を設定することにより、擬似連続リーチ変動パターンは、発展リーチ変動パターンよりも、信頼度が高く設定されることとなる。

【0625】

50

さらに、擬似連続リーチ変動パターンの中でも、擬似回数が多くなるほど、大当たり時の選択比率が高く、ハズレ時の選択比率が低く設定されており、擬似回数が多くなるほど、信頼度が高くなるように設定がなされている。

【0626】

以上のように、変動演出決定テーブルにより、変動演出の大まかな流れが決定されるが、変動演出の開始時には、変動モードコマンドまたは変動パターンコマンドに基づいて、変動演出を構成するさまざまな要素演出の実行可否および実行パターンがさらに決定される。ここで、要素演出というのは、例えば、上記のように、メイン演出表示部200aにおける演出図柄210a、210b、210cの変動表示や、リーチ発展演出でメイン演出表示部200aに表示される発展画像、さらには、演出役物装置202を可動させる演出等、変動演出を構成する全ての演出をいう。実施例では、変動演出を構成する要素演出として、変動演出中のさまざまなタイミングで予告演出（示唆演出）が実行される。

【0627】

この予告演出というのは、変動演出の開始時や、擬似連続リーチ変動パターンの変動演出における演出図柄210a、210b、210cの再変動表示時、さらには、リーチ発展演出中等に、メイン演出表示部200aに所定の画像を表示したり、所定のタイミングで演出役物装置202を可動したりする演出であり、予告演出ごとに、その実行可否や実行パターンが決定される。各予告演出には、それぞれ実行パターンが複数種類設けられ、複数種類の実行パターンそれぞれについて、変動パターンコマンドや変動モードコマンドごとに、換言すれば、大当たりの当選可否ごとに選択比率が設定され、この選択比率によって、実行パターンごとに期待値が設定されている。

【0628】

以上説明したように、副制御基板330においては、変動コマンドを受信すると、変動演出の実行パターンや各要素演出の実行可否、実行パターンが決定され、特別図柄の変動表示中に変動演出が実行されることとなる。このように、変動演出は、1回の特別図柄の変動表示に対して1回行われるが、実施例では、複数回の特別図柄の変動表示に跨る演出も実行される。

【0629】

図67は、演出参考例に係る保留表示演出の一例を説明する図である。メイン演出表示部200aの下部には保留表示領域211が設けられる。図60～図65では図示を省略したが、保留表示領域211は、変動演出中や遊技の待機中もメイン演出表示部200aに常時表示されている。そして、変動演出中は、この保留表示領域211において保留表示演出が行われる。保留表示演出では、大役抽選の際に処理領域（第0記憶部）に読み出された保留を示す当該保留表示212a、第1特図保留記憶領域の第1記憶部～第4記憶部に記憶されている保留をそれぞれ示す、第1保留表示212b、第2保留表示212c、第3保留表示212d、第4保留表示212eが保留表示領域211に表示される。なお、以下では、当該保留表示212aおよび第1保留表示212b～第4保留表示212eを総称して、保留表示212と呼ぶ。

【0630】

例えば、特別図柄の変動表示中であって、しかも、メインRAM300cに4つの特1保留が記憶されている場合には、図67(a)に示すように、当該保留表示212a、第1保留表示212b～第4保留表示212eの合計5つの保留表示212が保留表示領域211に表示される。そして、この状態から、特別図柄の変動表示が終了し、第1記憶部に記憶されている特1保留が処理領域（第0記憶部）に読み出されて大役抽選が行われるとともに、メインRAM300cの保留シフト処理が実行されると、図67(b)に示すように、当該保留表示212aが消去されるとともに、第1保留表示212b～第4保留表示212eが1つ左に移動表示される。また、さらにこの状態から次なる特1保留が読み出されると、図67(c)に示すように、さらに各保留表示212が移動表示される。このように、保留表示演出は、メインRAM300cに記憶されている特1保留数を遊技者に報知する演出となっている。

【0631】

また、保留表示212の表示パターンは複数設けられており、表示パターンごとに表示色を異ならせている。主制御基板300においては、保留が記憶されたときに取得時演出判定処理（ステップS536）が実行され、新たに記憶された保留が第0記憶部に読み出された際に決定される変動情報を示す先読み指定コマンドを副制御基板330に送信する。副制御基板330においては、先読み指定コマンドを受信すると、当該受信コマンドに基づいて、新たに記憶された保留に対応する保留表示212の表示パターンを決定する。このとき、先読み指定コマンドごとに、つまり、新たに記憶された保留が大役抽選で読み出された際に決定される変動情報ごとに、各表示パターンの選択比率が設定されている。つまり、大当たりの当選可否や、変動演出の実行パターンに応じて各表示パターンの選択比率が設定されていることから、保留表示212の表示パターンによって、大当たりの信頼度（期待値）が示唆されることとなる。

10

【0632】

図68（a）は最終保留表示パターン決定テーブルを説明する図であり、図68（b）は1つ前保留表示パターン決定テーブルを説明する図である。上記したように、主制御基板300における取得時演出判定処理では、新たに記憶された保留が読み出された際に決定される変動モード番号および変動パターン番号を示す先読み指定コマンドを副制御基板330に送信する。つまり、先読み指定コマンドは、保留が読み出されたときに決定される変動モード番号および変動パターン番号を副制御基板330に伝達するコマンドである。最終保留表示パターン決定テーブルによれば、先読み指定コマンド（変動パターン番号）ごとに、保留表示212の表示パターンの選択比率がそれぞれ設定されており、先読み指定コマンドを受信すると、保留表示212の最終の表示パターン、すなわち、当該保留表示212aの最終的な表示パターンが決定される。

20

【0633】

図68（a）に示す最終保留表示パターン決定テーブルによれば、「デフォルト（白）」、「点滅」、「青」、「黄」、「緑」、「黒」、「赤」、「プレミアム（虹）」の8種類の表示パターンのいずれかが決定される。そして、当該保留表示212aの最終的な表示パターンが決定されると、それ以前に表示される保留表示212の表示パターンが、図68（b）に示す1つ前保留表示パターン決定テーブルを参照して決定される。この1つ前保留表示パターン決定テーブルによれば、保留表示212の表示パターンごとに、移動表示前に表示する保留表示212の表示パターンの選択比率が設定されている。

30

【0634】

例えば、主制御基板300において、第1特図保留記憶領域の第2記憶部に保留が記憶された場合に、最終保留表示パターン決定テーブルを参照して、当該保留表示212aの最終的な表示パターンが決定されたとする。この場合には、次に、第1保留表示212bの表示パターンを、1つ前保留表示パターン決定テーブルを参照して決定する。このとき、第1保留表示212bの表示パターンは、先に決定された当該保留表示212aの最終的な表示パターンに基づいて決定される。例えば、当該保留表示212aの最終的な表示パターンが「青」であった場合、1つ前保留表示パターン決定テーブルによれば、第1保留表示212bの表示パターンとして、「点滅」が200／250の確率で決定され、「青」が50／250の確率で決定され。

40

【0635】

このようにして、第1保留表示212bの表示パターンが決定されると、次に、先に決定された第1保留表示212bの表示パターンに基づいて、再び、1つ前保留表示パターン決定テーブルを参照して、第2保留表示212cの表示パターンが決定される。

【0636】

以上のように、保留が記憶されると、まず、当該保留表示212aの最終的な表示パターンが決定され、その後、決定された当該保留表示212aの最終的な表示パターンに基づいて、第1保留表示212bの表示パターンが決定される等、表示順序を逆方向に遡るように、表示パターンが順次決定されることとなる。なお、1つ前保留表示パターン決定

50

テーブルによれば、先に決定されている保留表示 2 1 2 の表示パターンと同じか、もしくは、信頼度の低い表示パターンのみが決定されるように、選択比率が設定されている。

【0637】

上記のように、保留表示演出では、保留表示 2 1 2 について、所定の遊技利益の付与に対する期待値が異なる複数の表示パターンが設けられている。そして、保留表示 2 1 2 は、メイン演出表示部 2 0 0 a に最初に表示されてから最終的に消去されるまでの間、1 の表示パターンで表示される場合もあれば、表示期間中に表示パターンが変化する場合もある。

【0638】

演出参考例において、保留表示 2 1 2 の表示パターンの変化が生じるタイミングは、新たに記憶された特 1 保留（以下、対象保留ともいう）が、第 1 保留表示 2 1 2 b ~ 第 3 保留表示 2 1 2 d に移動表示されたタイミングと、対象保留に係る対象変動演出中とに大別される。

10

【0639】

次に、上記の変動演出を実行するための副制御基板 3 3 0 における処理について説明する。なお、以下では、副制御基板 3 3 0 における処理のうち、変動演出と関係のない処理については説明を省略する。

【0640】

（副制御基板 3 3 0 のサブ CPU 初期化处理）

図 6 9 は、演出参考例に係る副制御基板 3 3 0 のサブ CPU 初期化处理（S 1 0 0 0）を説明するフローチャートである。

20

【0641】

（ステップ S 1 0 0 0 - 1）

サブ CPU 3 3 0 a は、電源投入に応じて、サブ ROM 3 3 0 b から CPU 初期化处理プログラムを読み込むとともに、サブ RAM 3 3 0 c に記憶されるフラグ等の初期化、設定処理を行う。

【0642】

（ステップ S 1 0 0 0 - 3）

次に、サブ CPU 3 3 0 a は、各演出乱数を更新する処理を行うとともに、以後は、割込み処理が行われるまで当該ステップ S 1 0 0 0 - 3 の処理を繰り返し行う。なお、演出乱数は複数種類設けられており、ここでは、それぞれの演出乱数が非同期的に更新されている。

30

【0643】

（副制御基板 3 3 0 のサブタイマ割込み処理）

図 7 0 は、演出参考例に係る副制御基板 3 3 0 のサブタイマ割込み処理（S 1 1 0 0）を説明するフローチャートである。副制御基板 3 3 0 には、所定の周期（1 秒間に 3 0 回）でクロックパルスを発生するリセット用クロックパルス発生回路（不図示）が設けられている。そして、このリセット用クロックパルス発生回路によるクロックパルスの発生により、サブ CPU 3 3 0 a はタイマ割込み処理プログラムを読み込んで当該サブタイマ割込み処理を開始する。

40

【0644】

（ステップ S 1 1 0 0 - 1）

サブ CPU 3 3 0 a は、レジスタを退避する。

【0645】

（ステップ S 1 1 0 0 - 3）

サブ CPU 3 3 0 a は、割込みを許可するための処理を行う。

【0646】

（ステップ S 1 1 0 0 - 5）

サブ CPU 3 3 0 a は、副制御基板 3 3 0 で用いられる各種タイマカウンタの更新処理を行う。ここで、各種タイマカウンタは、特に断る場合を除き、当該副制御基板 3 3 0 の

50

サブタイマ割込み処理の度に1ずつ減算され、0になると減算を停止する。

【0647】

(ステップS1200)

サブCPU330aは、サブRAM330cの受信バッファに格納されているコマンドを解析するとともに、受信したコマンドに応じた種々の処理を行う。副制御基板330においては、主制御基板300からコマンドが送信されると、コマンド受信割込み処理が行われ、主制御基板300から送信されたコマンドが受信バッファに格納される。ここでは、コマンド受信割込み処理によって受信バッファに格納されたコマンドを解析することとなる。

【0648】

(ステップS1100-7)

サブCPU330aは、タイムテーブルを参照して、当該タイムテーブルに記憶された該当時間に対応する処理を実行するタイムスケジュール管理処理を行う。ここでは、タイムテーブルにセットされたタイムデータに基づいて、各種のフラグをオン、オフしたり、あるいは、各演出デバイスにコマンドを送信したりすることで、変動演出や大役演出をはじめとする各演出の実行を制御することとなる。

【0649】

(ステップS1100-9)

サブCPU330aは、レジスタを復帰して当該サブタイマ割込み処理を終了する。

【0650】

図71は、上記コマンド解析処理のうち、先読み指定コマンドを受信した際に実行される演出参考例に係る先読み指定コマンド受信処理を説明するフローチャートである。上記したとおり、先読み指定コマンド(先読み指定変動パターンコマンド)は、主制御基板300において、取得時演出判定処理(図33のステップS536-21、ステップS536-25、ステップS536-27)でセットされた後、サブコマンド送信処理(図23のステップS100-65)によって副制御基板330に送信される。

【0651】

(ステップS1210-1)

サブCPU330aは、まず、受信した先読み指定コマンドを解析する。

【0652】

(ステップS1210-3)

サブCPU330aは、上記ステップS1210-1の解析結果に基づいて、事前判定情報を記憶する。なお、副制御基板330のサブRAM330cには、主制御基板300の第1特図保留記憶領域に対応する第1事前判定情報記憶部と、第2特図保留記憶領域に対応する第2事前判定情報記憶部とが設けられている。第1事前判定情報記憶部は第1記憶部～第4記憶部の4つの記憶部を備えている。これら第1事前判定情報記憶部の第1記憶部～第4記憶部は、第1特図保留記憶領域の第1記憶部～第4記憶部にそれぞれ対応している。同様に、第2事前判定情報記憶部は第1記憶部～第4記憶部の4つの記憶部を備えており、これら第2事前判定情報記憶部の第1記憶部～第4記憶部は、第2特図保留記憶領域の第1記憶部～第4記憶部にそれぞれ対応している。ここでは、主制御基板300の第1特図保留記憶領域または第2特図保留記憶領域の第1記憶部～第4記憶部のうち、新たに保留が記憶された記憶部に対応する記憶部に事前判定情報が記憶される。

【0653】

(ステップS1210-5)

サブCPU330aは、保留表示212の最終の表示パターンを決定する最終保留表示パターン決定処理を行う。ここでは、受信した先読み指定コマンドに基づき、最終保留表示パターン決定テーブル(図68(a))を参照し、当該保留表示212aの最終の表示パターンを決定して記憶する。

【0654】

(ステップS1210-7)

10

20

30

40

50

サブCPU330aは、保留が記憶された記憶部に基づいて、保留表示212の表示パターンを決定する回数、すなわち保留表示212の変化タイミングを導出し、導出した回数だけ、1つ前保留表示パターン決定テーブル（図68（b））を参照して、保留表示212の表示パターンを決定する。そして、決定した保留表示212の表示パターン情報を所定の記憶部に記憶し、ステップS1210-9に処理を移す。

【0655】

（ステップS1210-9）

サブCPU330aは、上記ステップS1210-5およびステップS1210-7の決定に基づいて、保留表示212の表示を開始させる保留表示開始処理を行い、当該先読み指定コマンド受信処理を終了する。これにより、保留が記憶されたときに、対応する保留表示212の表示が開始されることとなる。

10

【0656】

図72は、演出参考例に係る上記コマンド解析処理のうち、変動コマンドを受信した際に実行される変動コマンド受信処理を説明するフローチャートである。上記したとおり、変動コマンドは、主制御基板300において、特別図柄変動番号決定処理（図39のステップS612-13、S612-17）でセットされた後、サブコマンド送信処理（図23のステップS100-65）によって副制御基板330に送信される。

【0657】

（ステップS1220-1）

変動コマンドを受信すると、サブCPU330aは、まず、受信した変動パターンコマンドを解析して、記憶する。

20

【0658】

（ステップS1220-3）

サブCPU330aは、上記ステップS1000-3で更新された演出乱数（0～249）を取得し、当該取得した演出乱数および上記ステップS1220-1における解析結果に基づいて、後半の変動演出の実行パターンを決定、記憶する。

【0659】

（ステップS1220-5）

サブCPU330aは、受信した変動モードコマンドを解析して、記憶する。

【0660】

（ステップS1220-7）

サブCPU330aは、上記ステップS1000-3で更新された演出乱数（0～249）を取得し、当該取得した演出乱数および上記ステップS1220-5における解析結果に基づいて、前半の変動演出の実行パターンを決定、記憶する。

30

【0661】

（ステップS1220-9）

サブCPU330aは、上記ステップS1000-3で更新された演出乱数（0～249）を予告演出ごとに取得し、当該取得した演出乱数および上記ステップS1220-1、ステップS1220-5における解析結果に基づいて、各予告演出決定テーブルを参照して、各予告演出の実行有無ならびに実行パターンを決定、記憶する。

40

【0662】

（ステップS1220-11）

サブCPU330aは、事前判定情報記憶部に記憶されている事前判定情報をシフトするシフト処理を実行する。ここでは、特1保留に基づく変動演出を開始する場合には、第1事前判定情報記憶部の第4記憶部～第2記憶部に記憶されている事前判定情報を、それぞれ第1事前判定情報記憶部の第3記憶部～第1記憶部にシフトし、特2保留に基づく変動演出を開始する場合には、第2事前判定情報記憶部の第4記憶部～第2記憶部に記憶されている事前判定情報を、それぞれ第2事前判定情報記憶部の第3記憶部～第1記憶部にシフトする。

【0663】

50

(ステップ S 1 2 2 0 - 1 3)

サブ C P U 3 3 0 a は、保留表示 2 1 2 を移動表示させる保留表示シフト処理を行う。また、ここでは、保留表示 2 1 2 の表示パターンが変化する場合には、所定のタイミングで表示パターンを変化させるための実行データをセットする。

【 0 6 6 4 】

(ステップ S 1 2 2 0 - 1 5)

サブ C P U 3 3 0 a は、上記各ステップの決定に基づいてタイムテーブルのタイムデータをセットして、当該変動コマンド受信処理を終了する。なお、ここでセットされたタイムテーブルに基づき、上記ステップ S 1 1 0 0 - 7 において、変動演出用の画像をメイン演出表示部 2 0 0 a に表示する処理や、音声出力処理、演出照明装置 2 0 4 の点灯制御処理等の演出実行制御がなされることとなる。

【 0 6 6 5 】

〈スロットマシン 4 0 0〉

図 7 3 および図 7 4 の外観図に示すように、遊技機としてのスロットマシン 4 0 0 は、前面が開口した筐体 4 0 2 と、筐体 4 0 2 の前面一端に回転可能に上下に並んで配置される前面上扉 4 0 4 および前面下扉 4 0 6 とが設けられている。前面上扉 4 0 4 の下部略中央には、ガラス板や透明樹脂板等で構成された無色透明の図柄表示窓 4 0 8 が設けられており、筐体 4 0 2 内の図柄表示窓 4 0 8 に対応する位置には、3つのリール 4 1 0 (左リール 4 1 0 a、中リール 4 1 0 b、右リール 4 1 0 c) が、それぞれ独立して回転可能に設けられている。左リール 4 1 0 a、中リール 4 1 0 b、右リール 4 1 0 c の外周面には、図 7 5 (a) の図柄配列に示すように、20 に等分された各領域に複数種類の図柄がそれぞれ配列されており、遊技者は、図柄表示窓 4 0 8 を通じて、上段、中段、下段に位置する、左リール 4 1 0 a、中リール 4 1 0 b、右リール 4 1 0 c それぞれの3つの連続する合計9個の図柄を視認することができる。

【 0 6 6 6 】

前面下扉 4 0 6 の上部には操作部設置台 4 1 2 が形成され、操作部設置台 4 1 2 には、メダル投入部 4 1 4、ベットスイッチ 4 1 6、スタートスイッチ 4 1 8、ストップスイッチ 4 2 0、演出スイッチ 4 2 2、精算スイッチ 4 3 7 等が設けられている。メダル投入部 4 1 4 は、メダル投入口 4 1 4 a を通じて遊技価値としてのメダルの投入を受け付ける。ベットスイッチ 4 1 6 は、スロットマシン 4 0 0 の内部に電氣的に貯留 (以下、単にクレジットという) されているメダルのうち、1 遊技で必要とされる規定数のメダルを投入 (ベット) する。

【 0 6 6 7 】

スタートスイッチ 4 1 8 は、例えば傾倒操作を検出可能なレバーで構成され、遊技者による遊技の開始操作を検出する。ストップスイッチ 4 2 0 (ストップスイッチ 4 2 0 a、ストップスイッチ 4 2 0 b、ストップスイッチ 4 2 0 c) は、左リール 4 1 0 a、中リール 4 1 0 b、右リール 4 1 0 c それぞれに対応して設けられており、遊技者の停止操作を検出する。なお、ストップスイッチ 4 2 0 の停止操作が可能な状態で、遊技者が、ストップスイッチ 4 2 0 a、ストップスイッチ 4 2 0 b、ストップスイッチ 4 2 0 c のいずれかを最初に停止操作することを第1停止といい、第1停止の後、停止操作されていない2つのストップスイッチ 4 2 0 のいずれかを停止操作することを第2停止といい、第2停止の後、最後に残ったストップスイッチ 4 2 0 を停止操作することを第3停止という。演出スイッチ 4 2 2 は、例えば、押圧スイッチと、その周囲に回転自在に配されたジョグダイヤルスイッチとから構成され、遊技者の押圧操作や回転操作を検出する。精算スイッチ 4 3 7 は、例えば、押圧スイッチで構成され、クレジットされたメダルをメダル排出口 4 4 0 a から受け皿部 4 4 0 に払い出す (精算する)。

【 0 6 6 8 】

前面上扉 4 0 4 の上部略中央には、演出に伴う様々な画像を表示する液晶表示部 4 2 4 が設けられている。また、前面上扉 4 0 4 の上部や左右には、例えば高輝度の発光ダイオード (L E D) によって構成される演出用ランプ 4 2 6 が設けられる。また、前面上扉 4

10

20

30

40

50

04の裏面における液晶表示部424の左右位置や前面下扉406の裏面における左右位置には、効果音や楽音等による聴覚的な演出を行うスピーカ428が設けられている。

【0669】

操作部設置台412には、メインクレジット表示部430およびメイン払出表示部432が設けられている。また、図柄表示窓408と操作部設置台412との間には、サブクレジット表示部434およびサブ払出表示部436が設けられている。これらメインクレジット表示部430およびサブクレジット表示部434にはクレジットされているメダルの枚数（クレジット枚数）が表示され、メイン払出表示部432およびサブ払出表示部436にはメダルの払出枚数が表示される。

【0670】

筐体402内におけるリール410の下方には、メダル排出口440aからメダルを払い出すためのメダル払出装置（メダルホッパー）442が設けられている。また、前面下扉406の前面下部には、メダル排出口440aから払い出されたメダルを貯留するための受け皿部440が設けられている。また、筐体402内には、電源スイッチ444が設けられている。電源スイッチ444は、スロットマシン400を管理する管理者が操作し、電源の切断状態と電源の投入状態の2つの状態を切り換えるために用いられる。

【0671】

また、筐体402内には、後述する主制御基板500に、設定ドアキー438および図示しない設定変更スイッチ（これらを合わせて設定値設定手段という）が設けられている。スロットマシン400では、設定ドアキー438に所定の鍵（操作キー）が挿入されてOFFの位置からONの位置へ回転された状態で電源スイッチ444を介して電源が投入されると設定変更モードに移行し、設定値の変更（単に設定変更ともいう）が可能な状態となる。設定値は、遊技者の有利度合（機械割）を段階的に示したものであり、例えば、1～6の6段階で表され、一般に、設定値の数値が大きいほど遊技全体として有利度合が高い（期待獲得枚数が高い）ように設定されている。そして、設定変更が可能な状態において設定変更スイッチが押下される度に設定値が1ずつ加算され、例えば、6段階の設定値のうちのいずれかの設定値に変更され、スタートスイッチ418が操作されると、設定値が確定し、設定ドアキー438を元の位置（OFFの位置）に戻すことで設定変更モードが終了して遊技が可能となる。なお、設定変更は、電源スイッチ444が操作されて電源の投入状態となってから一定期間のみ可能となっている。

【0672】

スロットマシン400では、遊技が開始可能となり、規定数のメダルがベットされると、有効ラインAが有効化するとともに、スタートスイッチ418に対する操作が有効となる。ここで、ベットは、ベットスイッチ416の操作を通じてクレジットされているメダルを投入する場合と、メダル投入部414を通じてメダルを投入する場合と、詳しくは後述するリプレイ役が有効ラインA上に表示されたことに基づいてメダルを自動投入する場合のいずれも含む。また、有効ラインAは、当選役の入賞を判定するためのラインであり、本実施形態では1本である。有効ラインAは、図75（b）に示すように、図柄表示窓408に臨む9つの図柄（3リール×上中下の3段）のうち、左リール410aの中段、中リール410bの中段、右リール410cの上段に停止する図柄に対応する位置を結んだラインに設定されている。無効ラインは、有効ラインA上に表示された図柄組み合わせのみでは当選役を把握しにくい場合に、当選役の把握を容易にする他の図柄組み合わせを表示する、当選役の入賞判定には用いられない有効ラインA以外のラインであり、本実施形態では、図75（b）に示す5つの無効ラインB1、B2、B3、C1、C2を想定している。

【0673】

そして、遊技者によりスタートスイッチ418が操作されると、遊技が開始され、左リール410a、中リール410b、右リール410cが回転されるとともに、当選種別抽選等が実行される。その後、ストップスイッチ420a、420b、420cの操作に応じて、対応する左リール410a、中リール410b、右リール410cをそれぞれ停止

10

20

30

40

50

させる。そして、当選種別抽選の抽選結果および有効ラインAに表示された図柄の組み合わせによって、メダルの払い出しを受け得る当選役が入賞した場合にはメダルの払い出しが実行され、メダルの払い出しを受け得る当選種別に不当選であった場合または当選したが入賞しなかった場合には左リール410a、中リール410b、右リール410cが全て停止したことをもって、遊技が終了する。

【0674】

なお、本実施形態において、上記1遊技は、メダル投入部414を通じたメダルの投入、ベットスイッチ416の操作を通じたクレジットされているメダルの投入、または、リプレイ役が有効ラインA上に表示されたことに基づくメダルの自動投入のいずれかが行われてから、遊技者によるスタートスイッチ418の操作に応じて、左リール410a、中リール410b、右リール410cが回転制御されるとともに当選種別抽選が実行され、当選種別抽選の抽選結果および遊技者による複数のストップスイッチ420a、420b、420cの操作に応じて、操作されたストップスイッチ420a、420b、420cに対応する左リール410a、中リール410b、右リール410cがそれぞれ停止制御され、メダルの払い出しを受け得る当選役が入賞した場合、そのメダルの払い出しが実行されるまでの遊技をいう。また、メダルの払い出しを受け得る当選種別に不当選であった場合または当選したが入賞しなかった場合、左リール410a、中リール410b、右リール410cが全て停止したことをもって1遊技が終了する。ただし、1遊技の開始を、上記のメダルの投入、または、リプレイ役の当選の代わりに、遊技者によるスタートスイッチ418の操作と読み替えてもよい。また、かかる1遊技が繰り返される数を遊技数とする。

【0675】

図76は、スロットマシン400の概略的な電氣的構成を示したブロック図である。図76に示すように、スロットマシン400は、遊技の進行を制御する主制御基板500（主制御部）と、遊技の進行に応じた演出を制御する副制御基板502（副制御部）とを含む制御基板が設けられている。また、主制御基板500と副制御基板502との間の電氣的な信号の伝達は、不正防止等の観点から、主制御基板500から副制御基板502への一方向のみに制限される。

【0676】

（主制御基板500）

主制御基板500は、中央処理装置であるメインCPU500a、プログラム等が格納されたメインROM500b、ワークエリアとして機能するメインRAM500c等を含む半導体集積回路を有し、スロットマシン400全体を統括的に制御する。なお、メインRAM500cは、電源が切断された場合においても、設定変更が行われてRAMクリアが実行されない限り、データが消去されることなく保持される。

【0677】

また、主制御基板500は、メインCPU500aが、メインROM500bに格納されたプログラムに基づきメインRAM500cと協働することで機能する、初期化手段600、ベット手段602、当選種別抽選手段604、リール制御手段606、判定手段608、払出制御手段610、遊技状態制御手段612、演出状態制御手段614、コマンド送信手段616等の機能部を有する。

【0678】

主制御基板500では、メダル投入口414aへのメダルの投入を検出する投入メダル検出部414b、ベットスイッチ416、スタートスイッチ418およびストップスイッチ420a、420b、420cから各種の検出信号を受信しており、受信した検出信号に基づいて、メインCPU500aが種々の処理を実行する。

【0679】

初期化手段600は、主制御基板500における初期化処理を実行する。ベット手段602は、遊技に使用するためのメダルをベットする。当選種別抽選手段604は、スタートスイッチ418の操作に基づき、詳しくは後述するように、当選役の当否、より詳しく

は、当選役が含まれる当選種別の当否を決定する当選種別抽選を行う。

【0680】

リール制御手段606は、スタートスイッチ418の操作に応じて、左リール410a、中リール410b、右リール410cを回転制御し、回転している左リール410a、中リール410b、右リール410cにそれぞれ対応したストップスイッチ420a、420b、420cの操作に応じて、対応する左リール410a、中リール410b、右リール410cを停止制御する。また、リール制御手段606は、スタートスイッチ418の操作に応じて、前回の遊技においてストップスイッチ420a、420b、420cの操作を有効化してから、当選種別抽選の抽選結果を表示するために遊技者によるストップスイッチ420a、420b、420cの操作を有効化するまで（前回の遊技におけるストップスイッチ420a、420b、420cの操作完了により無効化されている）の時間を規定の時間より延長し、その間、リール410a、410b、410cを多彩な態様で回転させるリール演出（フリーズ演出）を行う場合がある。リール演出は、本来有効となるべき任意のスイッチを所定時間有効にしなかったり、本来実行されるべき処理を所定時間保留したり、本来送受信されるべき任意のスイッチの信号を所定時間送信または受信させなかったりすることで実現できる。

10

【0681】

また、主制御基板500には、リール駆動制御部450が接続されている。このリール駆動制御部450は、スタートスイッチ418の操作信号に応じ、リール制御手段606から送信される、左リール410a、中リール410b、右リール410cの回転開始信号に基づいて、ステッピングモータ452を駆動する。また、リール駆動制御部450は、ストップスイッチ420の操作信号に応じ、リール制御手段606から送信される、左リール410a、中リール410b、右リール410cそれぞれの停止信号および回転位置検出回路454の検出信号に基づいて、ステッピングモータ452の駆動を停止する。

20

【0682】

判定手段608は、当選役に対応する図柄組み合わせが有効ラインA上に表示されたか否かを判定する。ここで、当選役に対応する図柄組み合わせが有効ラインA上に表示されることを単に入賞という場合がある。払出制御手段610は、当選役に対応する図柄組み合わせが有効ラインA上に表示されたこと（入賞したこと）に基づいて、当該当選役に対応する数（価値量）だけメダルを払い出す。また、主制御基板500には、メダル払出装置442が接続されており、払出制御手段610は、メダルの払出枚数を計数しながらメダルを排出する。

30

【0683】

遊技状態制御手段612は、当選種別抽選の結果や判定手段608の判定結果を参照し、複数種類の遊技状態のいずれかに遊技状態を移行させる。また、演出状態制御手段614は、当選種別抽選の結果、判定手段608の判定結果、遊技状態の遷移情報を参照し、複数種類の演出状態のいずれかに演出状態を移行させる。

【0684】

コマンド送信手段616は、ベット手段602、当選種別抽選手段604、リール制御手段606、判定手段608、払出制御手段610、遊技状態制御手段612、演出状態制御手段614等の動作に伴う、遊技に関するコマンドを順次決定し、決定したコマンドを副制御基板502に順次送信する。

40

【0685】

また、主制御基板500には、乱数発生器（乱数生成手段）500dが設けられる。乱数発生器500dは、計数値を順次インクリメントし、所定の数値範囲内でループさせ、所定の時点における計数値を抽出することで乱数を得る。主制御基板500の乱数発生器500dによって生成される乱数（以下、当選種別抽選乱数という）は、遊技者に付与する遊技利益、例えば、当選種別抽選手段604が当選種別を決定するために用いられる。

【0686】

（副制御基板502）

50

また、副制御基板 502 は、主制御基板 500 と同様に、中央処理装置であるサブ CPU 502 a、プログラム等が格納されたサブ ROM 502 b、ワークエリアとして機能するサブ RAM 502 c 等を含む各種半導体集積回路を有し、主制御基板 500 からのコマンドに基づき、特に演出を制御する。また、サブ RAM 502 c にもメイン RAM 500 c 同様、不図示のバックアップ電源が接続されており、電源が切断された場合においても、データが消去されることなく保持される。なお、副制御基板 502 にも、主制御基板 500 同様、乱数発生器（乱数生成手段）502 d が設けられており、乱数発生器 502 d によって生成される乱数（以下、演出抽選乱数という）は、主に演出の態様を決定するために用いられる。

【0687】

また、副制御基板 502 では、サブ CPU 502 a が、サブ ROM 502 b に格納されたプログラムに基づき、サブ RAM 502 c と協働することで機能する、初期化決定手段 630、コマンド受信手段 632、演出制御手段 634 等の機能部を有する。

【0688】

初期化決定手段 630 は、副制御基板 502 における初期化処理を実行する。コマンド受信手段 632 は、主制御基板 500 等、他の制御基板からのコマンドを受信し、コマンドに対する処理を行う。演出制御手段 634 は、演出スイッチ 422 から検出信号を受信するとともに、受信されたコマンドに基づいて液晶表示部 424、スピーカ 428、演出用ランプ 426 の各デバイスで行われる遊技の演出を決定する。具体的に、演出制御手段 634 は、液晶表示部 424 に表示される画像データや、演出用ランプ 426、サブクレジット表示部 434、サブ払出表示部 436 等の電飾機器を通じた演出のための電飾データを決定するとともに、スピーカ 428 から出力すべき音声を構成する音声データを決定する。そして、演出制御手段 634 は、決定した遊技の演出を実行する。なお、演出には、補助演出も含まれる。補助演出は、当選種別抽選において、正解役（特定の役）と不正解役とが重複した選択当選種別に当選したときに、その正解役の入賞条件となるストップスイッチ 420 a、420 b、420 c の正解操作態様を報知する演出である。かかる補助演出により、遊技者は、正解役に対応する図柄組み合わせを、遊技者が有効ライン A 上に容易に表示させることができる。かかる補助演出を実行する演出状態を A T（アシストタイム）演出状態という。また、A T 演出状態とリプレイ役の当選確率が高い R T（リプレイタイム）遊技状態が並行して進行される所謂 A R T 遊技状態を用いることもある。

【0689】

なお、以下では、液晶表示部 424、演出用ランプ 426、スピーカ 428、サブクレジット表示部 434、サブ払出表示部 436 といった、副制御基板 502 を含む、主制御基板 500 以外の基板で管理される報知手段を他報知手段という場合がある。これに対し、メインクレジット表示部 430、メイン払出表示部 432 といった、主制御基板 500 で管理される報知手段を主報知手段（指示モニタ）という場合がある。また、補助演出を実行可能な主報知手段および他報知手段を合わせて補助演出実行手段という場合もある。演出状態制御手段 614 は、A T 演出状態において、補助演出を補助演出実行手段に実行させる。

【0690】

（主制御基板 500 で用いられるテーブル）

図 77 は、当選役を説明するための説明図であり、図 78 および図 79 は、当選種別抽選テーブルを説明するための説明図である。

【0691】

スロットマシン 400 においては、詳しくは後述するように、複数種類の遊技状態および演出状態が設けられており、遊技の進行に応じて遊技状態および演出状態が移行される。そして、主制御基板 500 では、遊技状態制御手段 612 により管理、制御される遊技状態に対応する複数の当選種別抽選テーブル等がメイン ROM 500 b に格納されている。当選種別抽選手段 604 は、メイン RAM 500 c に記憶された現在の設定値（遊技利益を得る容易性を段階的に示したもの）と現在の遊技状態に応じて、対応する当選種別抽

10

20

30

40

50

選テーブルをメインROM500bから抽出し、抽出した当選種別抽選テーブルに基づき、スタートスイッチ418の操作信号に応じて取得された当選種別抽選乱数が当選種別抽選テーブル内のいずれの当選種別に対応するか判定する。

【0692】

ここで、当選種別抽選テーブルで抽出される当選種別を構成する当選役には、リプレイ役、小役、ボーナス役が含まれる。リプレイ役は、リプレイ役に対応する図柄組み合わせが有効ラインA上に表示されると、遊技者によるメダルの新たなベットを行わずして再度遊技を実行できる役である。小役は、その小役に対応する図柄組み合わせが有効ラインA上に表示されることにより、図柄組み合わせに応じて所定枚数のメダルの払い出しを受けることができる役である。また、ボーナス役は、そのボーナス役に対応する図柄組み合わせが有効ラインA上に表示されることにより、遊技状態制御手段612により管理される遊技状態をボーナス遊技状態（後述するRBB作動中遊技状態）に移行させることができる役である。

【0693】

本実施形態における当選役は、図77に示すように、リプレイ役として、当選役「リプレイ1」～「リプレイ7」が設けられている。また、小役として、当選役「小役1」～「小役39」が設けられている。また、ボーナス役として、当選役「RBB」が設けられている。図77では、左リール410a、中リール410b、右リール410cそれぞれに、各当選役を構成する1または複数の図柄が対応付けられている。

【0694】

ここで、本実施形態においては、遊技者によってストップスイッチ420が操作されたときに、入賞可能な当選役に対応する図柄組み合わせを構成する図柄が有効ラインA上にある場合には、リール制御手段606によって、当該図柄が有効ラインA上に停止するように停止制御がなされる。また、ストップスイッチ420が操作されたときに、入賞可能な当選役に対応する図柄組み合わせを構成する図柄が、有効ラインA上にはないが、リール410の回転方向と反対の方向の図柄4コマ分に相当する範囲（引込範囲）内に存在している場合には、リール制御手段606によって、離れている図柄数が滑りコマ数となり、当該当選役に対応する図柄組み合わせを構成する図柄を有効ラインA上に引き込むように滑りコマ数分回転を維持した後に停止するように停止制御がなされる。また、入賞可能な当選役に対応する図柄がリール410中に複数あり、いずれもリール410の引込範囲内に存在している場合には、予め定められた優先順位に従っていずれの図柄を有効ラインA上に引き込むか決定され、当該優先された図柄を有効ラインA上に引き込むように滑りコマ数分回転を維持した後に停止するように停止制御がなされる。なお、ストップスイッチ420が押圧操作されたときに、入賞可能な当選役以外の当選役に対応する図柄組み合わせを構成する図柄が有効ラインA上にある場合には、リール制御手段606によって、その図柄を有効ラインA上に停止させないようにする、所謂蹴飛ばし処理も並行して実行される。また、後述するように、当選種別に含まれる当選役に操作態様（操作順や操作タイミング）が入賞条件として設定されている場合、リール制御手段606は、遊技者の操作態様に依拠して当選役に対応する図柄組み合わせを有効ラインA上に表示可能に停止制御する。

【0695】

そして、例えば、当選役「リプレイ1」～「リプレイ4」、当選役「小役1」～「小役6」、「小役35」～「小役39」に対応する図柄組み合わせを構成する図柄は、各リール410において、上記の停止制御によって、必ず有効ラインA上に表示可能なように配列されている。このような当選役をPB=1と表す場合がある。一方、例えば、当選役「リプレイ5」～「リプレイ7」、当選役「小役7」～「小役34」、当選役「RBB」に対応する図柄組み合わせを構成する図柄は、各リール410において、上記の停止制御によって、必ずしも有効ラインA上に表示可能なように配列されていないので、所謂とりこぼしが発生する場合がある。このような当選役をPB≠1と表す場合がある。

【0696】

図78および図79に示すように、当選種別抽選テーブルでは、複数の当選領域が区画されており、各遊技状態によって抽選の対象となる当選種別が異なったり、不当選（ハズレ）の有無が異なったりする。図78および図79では、各遊技状態（非内部遊技状態（非内部）、RBB内部中遊技状態（RBB内部中）、RBB作動中遊技状態（RBB作動中））毎に割り当てられた当選領域（当選種別）を「◎」や「○」で表しているが、実際には、複数の遊技状態それぞれに対応する当選種別抽選テーブルがメインROM500bに記憶されている。なお、「◎」は有利区間に移行させる抽選を行うことが可能な有利区間抽選可当選種別であることを示し、「○」は有利区間に移行させる抽選を行うことが不可な有利区間抽選不可当選種別であることを示している。

【0697】

当選種別抽選テーブルでは、区画化された各当選領域にはそれぞれ当選範囲を示す数値である所定の置数（当選範囲値）と当選種別が対応付けられており、遊技状態毎に割り当てられた全ての当選領域の置数を合計すると当選種別抽選乱数の総数（65536）となる。したがって、当選種別それぞれが決定される確率は、当選領域に対応付けられた置数を当選種別抽選乱数の総数で除算した値となる。当選種別抽選手段604は、その時点の遊技状態に基づいて、当該当選種別抽選テーブルにおける複数の当選領域のうち番号の高い方から、順次、置数を取得し、その置数を当選種別抽選乱数から減算して、減算後の値が0未満となると、その時点の当選領域に対応付けられた当選種別を当選種別抽選の抽選結果としている。また、当選領域1以上の全ての当選領域の置数を当選種別抽選乱数から減算して、減算後の値が0以上となっていれば、当選領域0の当選種別「ハズレ」が当選種別抽選の抽選結果となる。

【0698】

ここで、当選種別「RBB」を構成する当選役「RBB」について補足する。所定の第1種特別役物RBは、規定数ごとの入賞に係る図柄の組み合わせの数を増加させ、または規定数ごとの入賞に係る条件装置が作動する確率を上昇させる役物で、あらかじめ定められた場合に作動し、12回を超えない回数 of 遊技の結果が得られるまで作動を継続することができるものをいう。ここで、条件装置は、その作動が入賞、再遊技、役物または役物連続作動装置の作動に係る図柄の組み合わせが表示されるために必要な条件とされている装置で、当選種別抽選（遊技機内で行われる電子計算機によるくじ）に当選した場合に作動するもの、すなわち、当選フラグを意味する。そして、当選種別「RBB」を構成する第1種特別役物に係る役物連続作動装置（当選役「RBB」）は、第1種特別役物RBを連続して作動させることができる装置であり、特定の図柄の組み合わせが表示された場合に作動し、あらかじめ定められた場合に作動を終了するものをいう。

【0699】

図78の当選種別抽選テーブルによれば、例えば当選領域0には、当選種別「ハズレ」が対応付けられており、かかる当選種別に当選すると、図77に示したいずれの当選役に対応する図柄組み合わせも有効ラインA上に表示されることはなく、メダルの払い出し等が行われることはない。ただし、後述するように、RBB内部当選フラグが次遊技に持ち越されている場合、当選種別「ハズレ」の当選により、当選役「RBB」に対応する図柄組み合わせを有効ラインA上に表示させることが可能となる。

【0700】

また、当選領域1には、当選役「小役1」～「小役39」が重複して含まれる当選種別「小役ALL」が対応付けられており、当選領域2には、当選役「小役11」～「小役39」が重複して含まれる当選種別「1枚ALLA」が対応付けられており、当選領域3には、当選役「小役12」～「小役27」が重複して含まれる当選種別「1枚ALLB」が対応付けられている。また、当選領域4には、当選役「小役12」、「小役20」が重複して含まれる当選種別「折1枚1」が対応付けられ、当選領域5には、当選役「小役13」、「小役21」が重複して含まれる当選種別「折1枚2」が対応付けられ、当選領域6には、当選役「小役14」、「小役22」が重複して含まれる当選種別「折1枚3」が対応付けられ、当選領域7には、当選役「小役15」、「小役23」が重複して含まれる当

10

20

30

40

50

選種別「択１枚４」が対応付けられている。また、当選領域８には、当選役「小役３５」、「小役３６」が重複して含まれる当選種別「弱チェリー」が対応付けられ、当選領域９には、当選役「小役３７」～「小役３９」が重複して含まれる当選種別「強チェリー」が対応付けられ、当選領域１０には、当選役「小役７」、「小役８」、「小役１１」が重複して含まれる当選種別「スイカＡ」が対応付けられ、当選領域１１には、当選役「小役７」～「小役１０」が重複して含まれる当選種別「スイカＢ」が対応付けられ、当選領域１２には、当選役「小役１」、「小役３」、「小役５」が重複して含まれる当選種別「強ベル」が対応付けられている。また、当選領域１３には、当選役「小役１」～「小役６」が重複して含まれる当選種別「共通ベルＡ」が対応付けられ、当選領域１４には、当選役「小役１」～「小役６」、「小役１１」が重複して含まれる当選種別「共通ベルＢ」が対応付けられている。

10

【０７０１】

また、当選領域１５～２６には、払出枚数が１１枚となる正解役（当選役「小役１」～「小役６」）と、払出枚数が１枚の不正解役（当選役「小役１２」～「小役３４」）が重複して含まれる選択当選種別（当選種別「打順ベル１Ａ」～「打順ベル６Ａ」、「打順ベル１Ｂ」～「打順ベル６Ｂ」）がそれぞれ対応付けられている。

【０７０２】

また、図７９の当選種別抽選テーブルによれば、当選領域２７～３９には、当選役「リプレイ１」～「リプレイ７」が重複して含まれる当選種別「シンボルリプレイＡ１」～「シンボルリプレイＡ７」、「シンボルリプレイＢ１」～「シンボルリプレイＢ６」が対応付けられている。また、当選領域４０には、当選役「リプレイ１」、「リプレイ２」が重複して含まれる当選種別「通常リプレイ」が対応付けられている。

20

【０７０３】

また、当選領域４１～４６には、当選役「ＲＢＢ」が単独、または、他の小役と重複して含まれる当選種別「ＲＢＢ１」～「ＲＢＢ６」が対応付けられている。

【０７０４】

そして、複数の当選役が重複して含まれる当選種別に当選した場合には、いずれの当選役に対応する図柄組み合わせを有効ラインＡ上に優先的に表示させるかについての入賞条件、例えば、ストップスイッチ４２０ａ、４２０ｂ、４２０ｃが操作される順番が設定されている。

30

【０７０５】

以下の説明において、左リール４１０ａ、中リール４１０ｂ、右リール４１０ｃの順にリールを停止させるストップスイッチ４２０ａ、４２０ｂ、４２０ｃの操作を「打順１」とし、左リール４１０ａ、右リール４１０ｃ、中リール４１０ｂの順にリールを停止させるストップスイッチ４２０ａ、４２０ｂ、４２０ｃの操作を「打順２」とし、中リール４１０ｂ、左リール４１０ａ、右リール４１０ｃの順にリールを停止させるストップスイッチ４２０ａ、４２０ｂ、４２０ｃの操作を「打順３」とし、中リール４１０ｂ、右リール４１０ｃ、左リール４１０ａの順にリールを停止させるストップスイッチ４２０ａ、４２０ｂ、４２０ｃの操作を「打順４」とし、右リール４１０ｃ、左リール４１０ａ、中リール４１０ｂの順にリールを停止させるストップスイッチ４２０ａ、４２０ｂ、４２０ｃの操作を「打順５」とし、右リール４１０ｃ、中リール４１０ｂ、左リール４１０ａの順にリールを停止させるストップスイッチ４２０ａ、４２０ｂ、４２０ｃの操作を「打順６」とする。

40

【０７０６】

例えば、非内部遊技状態において、当選領域１５の当選種別「打順ベル１Ａ」に当選し、正解操作態様（打順１）による操作が行われた場合、払出枚数が１１枚の正解役である当選役「小役１」に対応する図柄組み合わせが有効ラインＡ上に優先的に表示されるように停止制御がなされる。また、打順２による操作が行われた場合、払出枚数１枚の不正解役である当選役「小役２８」に対応する図柄組み合わせが有効ラインＡ上に優先的に表示されるように停止制御がなされ、打順３、４による操作が行われた場合、払出枚数１枚の

50

不正解役である当選役「小役１２」に対応する図柄組み合わせが有効ラインＡ上に優先的に１／４の確率で表示されるように停止制御がなされ、打順５、６による操作が行われた場合、払出枚数１枚の不正解役である当選役「小役１６」に対応する図柄組み合わせが有効ラインＡ上に優先的に１／４の確率で表示されるように停止制御がなされる。

【０７０７】

なお、当選領域１５～２６の各当選種別の当選確率（置数）は等しくなるように設定されている。遊技者は、通常、いずれの当選種別に当選しているのかわかることができないため、上記のような当選領域１５～２６を設けることにより、正解役を入賞させにくくしている。また、上記のように、不正解役が優先的に表示される打順でストップスイッチ４２０ａ、４２０ｂ、４２０ｃが操作されても、必ずしも不正解役に対応する図柄組み合わせを有効ラインＡ上に表示させられるとは限らないので、その操作態様によっては、とりこぼしが発生することがある（ＰＢ≠１）。

【０７０８】

なお、上述したいずれかの当選種別に当選すると、それぞれの当選種別に対応する内部当選フラグが成立（ＯＮ）するとともに、この内部当選フラグの成立状況に応じて、各ルール４１０の停止制御がなされることとなる。このとき、小役が含まれる当選種別に当選したものの、これら当選役に対応する図柄組み合わせを、その遊技内で有効ラインＡ上に表示させることができなかった場合には、当該遊技の終了後に内部当選フラグがＯＦＦされる。つまり、小役の当選の権利は小役が含まれる当選種別に当選した遊技内のみに限られ、当該権利を次遊技に持ち越すことはできない。これに対して、当選役「ＲＢＢ」が含まれる当選種別に当選した場合には、ＲＢＢ内部当選フラグが成立（ＯＮ）するとともに、当選役「ＲＢＢ」に対応する図柄組み合わせが有効ラインＡ上に表示されるまで、ＲＢＢ内部当選フラグが遊技を跨いで持ち越される。なお、リプレイ役が含まれる当選種別に対応する内部当選フラグが成立した場合には、その当選種別に含まれるリプレイ役のうちのいずれかのリプレイ役に対応する図柄組み合わせが必ず有効ラインＡ上に表示され、メダルを要することなく次遊技を行うために必要となる処理が行われた後に、当該内部当選フラグがＯＦＦされる。

【０７０９】

（遊技状態の遷移）

ここで、図８０を用い、遊技状態の遷移について説明する。ここでは、非内部遊技状態、ＲＢＢ内部中遊技状態、ＲＢＢ作動中遊技状態といった複数の遊技状態が準備されている。各遊技状態は、後述するように、ボーナス役の当選、入賞（作動）、終了に応じて遷移させる。

【０７１０】

非内部遊技状態は、複数の遊技状態における初期状態に相当する遊技状態である。かかる非内部遊技状態では、リプレイ役の当選確率が約１／７．３に設定されている。また、非内部遊技状態では、当選役「ＲＢＢ」が所定の確率（例えば約１／３０）で決定されている。

【０７１１】

遊技状態制御手段６１２は、当選役「ＲＢＢ」の当選に応じて遊技状態を遷移させる。例えば、当選役「ＲＢＢ」が当選した遊技において、当選役「ＲＢＢ」に対応する図柄組み合わせが有効ラインＡ上に表示されると、遊技状態制御手段６１２は、遊技状態をＲＢＢ作動中遊技状態に移行させる（１）。

【０７１２】

ＲＢＢ作動中遊技状態では、リプレイ役の当選確率が０に設定されている。なお、かかるＲＢＢ作動中遊技状態では、当選可能な当選種別として、当選領域１に当選種別「小役ＡＬＬ」が、当選領域２、３に当選種別「１枚ＡＬＬＡ」、「１枚ＡＬＬＢ」が設定されている。当選種別「小役ＡＬＬ」に当選すると、当選役「小役１」～「小役３９」のいずれかに対応する図柄組み合わせが有効ラインＡ上に表示され、当選種別「１枚ＡＬＬ」に当選すると、当選役「小役１１」～「小役３９」のいずれかに対応する図柄組み合わせが

有効ラインA上に表示されるように停止制御される。ここでは、かかる小役の構成によりRBB作動中遊技状態での単位遊技当たりの期待獲得枚数を低くしている。

【0713】

RBB作動中遊技状態の終了条件を満たすと、すなわち、獲得枚数が所定枚数に到達すると、遊技状態制御手段612は、遊技状態を非内部遊技状態に移行させる(2)。

【0714】

一方、当選役「RBB」が当選した遊技において、当選役「RBB」に対応する図柄組み合わせを有効ラインA上に表示することができなかった場合、遊技状態制御手段612は、遊技状態をRBB内部中遊技状態(特別遊技状態)に移行させる(3)。

【0715】

RBB内部中遊技状態では、リプレイ役の当選確率が約 $1/5.9$ に設定されている。また、RBB内部中遊技状態では当選種別「ハズレ」に当選することはない。換言すれば、当選役「RBB」の当選遊技で当選役「RBB」に対応する図柄組み合わせを有効ラインA上に表示することができなかった場合、その後は、当選役「RBB」より小役やリプレイ役の方が優先して有効ラインA上に停止制御されるので、当選役「RBB」に対応する図柄組み合わせを有効ラインA上に表示することができない。したがって、一旦、遊技状態がRBB内部中遊技状態に移行すると、その後、遊技状態が遷移することなく、RBB内部中遊技状態が維持されることとなる。ここでは、かかるRBB内部中遊技状態を維持しつつ、そのRBB内部中遊技状態においてAT演出状態を実現する。

【0716】

ここでは、RBB内部中遊技状態において、複数種類の正解役が互いに重複せずに当選するため、正解役を入賞させることができる機会を多くすることができ、その結果、例えば、RBB内部中遊技状態におけるAT演出状態において補助演出が行われることで、メダルを獲得しやすくなる。一方、RBB作動中遊技状態では、複数種類の正解役が重複して当選するため、正解役を入賞させることができる機会が少ないので、他の遊技状態におけるAT演出状態よりも正解役を入賞させることができる機会が減り、遊技者が所有するメダルを増やしにくくしている。したがって、RBB内部中遊技状態よりも入賞に係る当選役の当選確率が高いというRBB作動中遊技状態の機能を備えつつ、メダルの獲得性能の面ではRBB作動中遊技状態がRBB内部中遊技状態に劣るという仕様(アクセルRBB)を実現することができる。

【0717】

(演出状態の遷移)

図81は、演出状態の遷移を説明するための説明図である。以下、主制御基板500において演出状態制御手段614により遷移される演出状態(非AT演出状態、AT演出状態)について詳述する。なお、以下では、遊技状態がRBB内部中遊技状態である場合について説明する。

【0718】

ここで、メダルの獲得性能が高い遊技状態が偏っているか否かを統括的かつ画一的に判定すべく、指示機能に係る性能を有する遊技区間、すなわち、補助演出(指示機能)を実行する遊技区間等を含む、遊技者にとって有利な遊技区間を有利区間(特定区間)として定義する。なお、有利区間は、主制御基板500で補助演出の作動に係る抽選等を行った結果、補助演出が作動した場合には、主制御基板500において指示の内容が識別できるよう、例えば、主報知手段に表示したときに限り、指示情報を、副制御基板502等の周辺基板に送信してもよい遊技区間である。また、有利区間と異なる遊技区間を非有利区間とする。したがって、複数の演出状態(非AT演出状態、AT演出状態)は、遊技区間である有利区間および非有利区間のいずれかに属することとなる。本実施形態では、ほぼ全ての演出状態が有利区間に属し、非AT演出状態の一部の演出状態(ここでは非有利演出状態)で非有利区間を実現している。

【0719】

なお、有利区間において、補助演出がないと正解役を取りこぼしてしまう当選態様のう

10

20

30

40

50

ち、正解役の配当が最大（ここでは、11枚）となる選択当選種別において、正解役の入賞を補助する補助演出（最大払出枚数を獲得できる補助演出）を行う場合、例えば、区間表示器160を点灯させることによって、その旨を報知しなければならない。

【0720】

また、非有利区間においては、当選種別の当選確率を設定値毎に異ならせることは可能であるが、同一の当選種別において補助演出を伴う演出状態（AT演出状態）への移行を決定する確率は設定値毎に異ならせてはならない。一方、有利区間においては、当選種別の当選確率、および、同一の当選種別における補助演出を伴う演出状態（AT演出状態）への移行（または追加）を決定する確率のいずれも設定値毎に異ならせることは可能である。

10

【0721】

したがって、演出状態制御手段614は、演出状態の移行の管理に加え、非有利区間と有利区間との移行も管理することとなる。また、有利区間は、このような管理に拘わらず、以下の終了条件を満たすことで強制的に終了する。例えば、有利区間において計数される値が所定値に達したこと（例えば、滞在遊技数（遊技実行回数）が1500遊技に達したり、獲得枚数（純増枚数）が2400枚を超えたこと）に基づいて強制的に終了する。いずれの場合においても、演出状態制御手段614は、有利区間から非有利区間に移行することで、有利区間で更新された情報（指示機能に係る性能に影響を及ぼす全ての変数）を全てリセットする。

【0722】

20

（非AT演出状態、AT演出状態）

非AT演出状態においては、AT演出状態より、補助演出の実行頻度が極めて低く、補助演出がほぼ行われないので、獲得できるメダルの枚数が制限される。ここでは、非AT演出状態として通常演出状態、非有利演出状態、準備演出状態、第1インターバル演出状態、継続抽選演出状態、第2インターバル演出状態といった6つの演出状態が設けられている。

【0723】

AT演出状態においては、選択当選種別全ての当選時において補助演出実行手段に補助演出を実行させることで、メダルの消費を抑えつつ、多くのメダルを獲得することが可能となる。したがって、遊技者は、AT演出状態に移行することで、非AT演出状態と比べ、遊技を有利に進行することができる。ここでは、AT演出状態として、擬似ボーナス演出状態が設けられている。以下、各演出状態（通常演出状態、擬似ボーナス演出状態、非有利演出状態、準備演出状態、第1インターバル演出状態、継続抽選演出状態、第2インターバル演出状態）について個々に説明する。

30

【0724】

（各演出状態）

通常演出状態は、複数の演出状態のうち、初期状態に相当する演出状態である。演出状態制御手段614は、通常演出状態においてAT抽選を行う。AT抽選は、AT演出状態への移行を決定する抽選であり、演出状態制御手段614は、当選種別抽選により決定された当選種別ごとに異なる確率でAT抽選を行う。そして、AT抽選に当選した場合、演出状態制御手段614は、所定の前兆遊技数が経過した後に、演出状態をAT演出状態である擬似ボーナス演出状態に移行させる（1）。また、擬似ボーナス演出状態に移行することなく通常演出状態で所定の天井条件（例えば、通常演出状態を連続して所定遊技数消化）が満たされると（所謂、天井到達）、演出状態制御手段614は、演出状態を擬似ボーナス演出状態へ移行させる（1）。

40

【0725】

擬似ボーナス演出状態では、所定の終了条件（例えば、消化した遊技数が所定遊技数に到達すること）を満たすまで、補助演出が実行され、遊技者は、擬似ボーナス演出状態において、例えば、期待獲得枚数3.5枚を得ることができる。なお、擬似ボーナス演出状態では、主として2つの終了条件が準備されている。例えば、終了条件が、消化した遊技

50

数が所定遊技数に到達することであった場合、所定遊技数として、30と80の2段階設けられ、いずれかが抽選により決定される。擬似ボーナス演出状態では、当然、消化した遊技数が多いほど、遊技者は、多くの遊技利益を得ることになる。したがって、遊技者は、所定遊技数が30ではなく80であることを望むこととなる。ただし、所定遊技数として30が決定された場合であっても、その消化中に補助演出の所定遊技数を30から80に増加させる昇格抽選も行われる。なお、ここでは、擬似ボーナス演出状態の終了条件として、消化した遊技数が所定遊技数に到達したことを例示しているが、これに代えて、最大払出枚数を獲得できる補助演出の回数が所定数に到達することや、獲得枚数（純増枚数）が所定数に到達することを採用してもよい。

【0726】

そして、擬似ボーナス演出状態において、所定の終了条件を満たすと、演出状態制御手段614は、演出状態を非有利演出状態に移行する場合がある（2）。このとき、演出状態制御手段614は、有利区間を、一旦、非有利区間に移行し、有利区間で更新された情報を全てリセットする。例えば、前段の擬似ボーナス演出状態が初回の擬似ボーナス演出状態であった場合、すなわち、通常演出状態から移行した直後の擬似ボーナス演出状態であれば、演出状態制御手段614は、演出状態を非有利演出状態に移行するとともに、有利区間を、一旦、非有利区間に移行する。また、前段の擬似ボーナス演出状態が、通常演出状態を経由しない3の倍数（3、6、…）回目の擬似ボーナス演出状態であった場合も、演出状態制御手段614は、有利区間を、一旦、非有利区間に移行するとともに、有利区間を、一旦、非有利区間に移行する。

【0727】

そして、演出状態制御手段614は、有利区間への移行を高確率（例えば、1／2）で抽選し、数遊技を消化後、有利区間に戻すとともに、演出状態を準備演出状態に移行する（3）。これは、有利区間において計数される値が所定値に達することにより、擬似ボーナス演出状態が強制的に終了することを回避するためである。

【0728】

一方、前段の擬似ボーナス演出状態が初回ではない、もしくは、通常演出状態を経由しない3の倍数回目の擬似ボーナス演出状態ではない場合、演出状態制御手段614は、演出状態を非有利演出状態に移行することなく、直接、準備演出状態に移行する（4）。これは、有利区間において計数される値が所定値に達することにより擬似ボーナス演出状態が強制的に終了することを回避しつつ、不要に有利区間を終了させないためである。

【0729】

準備演出状態において、演出状態制御手段614は、第1インターバル演出状態への移行を高確率（例えば、1／2）で抽選し、第1インターバル演出状態への移行の決定に基づいて、演出状態を第1インターバル演出状態に移行する（5）。

【0730】

第1インターバル演出状態は、所定の終了条件（例えば、40遊技の消化）を満たすまで継続し、その間、擬似ボーナス演出状態の継続抽選が実行される。かかる第1インターバル演出状態は、擬似ボーナス演出状態と異なり、単位遊技当たりの期待獲得枚数が少なく、負の値となる場合もある。なお、本実施形態では、非有利区間において有利区間に当選した場合、演出状態が必ず第1インターバル演出状態となるように設定されている。したがって、擬似ボーナス演出状態の終了後は、有利区間が継続した場合であっても、上記のように、一旦、非有利区間に移行した場合であっても、第1インターバル演出状態に移行することとなる。そして、第1インターバル演出状態における所定の終了条件を満たすと、演出状態制御手段614は、演出状態を継続抽選演出状態に移行する（6）。

【0731】

継続抽選演出状態は、所定の終了条件（例えば、5遊技の消化）を満たすまで継続し、最終的に、第1インターバル演出状態における、擬似ボーナス演出状態の継続抽選の結果が報知される。かかる継続抽選演出状態も、擬似ボーナス演出状態と異なり、単位遊技当たりの期待獲得枚数が少なく、負の値となる場合もある。第1インターバル演出状態にお

10

20

30

40

50

いて、擬似ボーナス演出状態への移行（継続）が決定されていれば、演出状態制御手段 6 1 4 は、演出状態を擬似ボーナス演出状態への移行し（7）、擬似ボーナス演出状態への移行が決定されていなければ、演出状態制御手段 6 1 4 は、演出状態を通常演出状態へ戻す（8）。

【0732】

このように、本実施形態では、一旦、擬似ボーナス演出状態に移行すると、継続抽選演出状態の継続抽選に漏れない限り、擬似ボーナス演出状態→準備演出状態→第1インターバル演出状態→継続抽選演出状態の移行を繰り返すことで、遊技者は、メダルを蓄積できる。

【0733】

なお、準備演出状態において、演出状態制御手段 6 1 4 は、第1インターバル演出状態への移行抽選に加え、第2インターバル演出状態への移行抽選も行っている。ただし、第1インターバル演出状態への移行確率（例えば、 $1/2$ ）より、第2インターバル演出状態への移行確率（例えば、 $1/20$ ）の方が小さく設定されている。準備演出状態において、第1インターバル演出状態への移行が決定する前に、第2インターバル演出状態への移行が決定されると、演出状態制御手段 6 1 4 は、第2インターバル演出状態への移行の決定に基づいて、演出状態を第2インターバル演出状態に移行する（9）。なお、ここでは、準備演出状態において第2インターバル演出状態への移行抽選を行う例を挙げているが、同様に、非有利演出状態において第2インターバル演出状態への移行抽選を行ってもよい。

【0734】

第2インターバル演出状態は、第1インターバル演出状態同様、所定の終了条件（例えば、40遊技の消化）を満たすまで継続し、その間、擬似ボーナス演出状態の継続抽選が実行される。かかる第2インターバル演出状態は、第1インターバル演出状態同様、擬似ボーナス演出状態と異なり、単位遊技当たりの期待獲得枚数が少なく、負の値となる場合もある。

【0735】

ただし、第2インターバル演出状態は、第1インターバル演出状態より高確率で擬似ボーナス演出状態が決定される。例えば、第1インターバル演出状態においては、40遊技間、擬似ボーナス演出状態を毎遊技 $1/58$ で決定し（当選確率50%）、第2インターバル演出状態においては、40遊技間、擬似ボーナス演出状態を毎遊技 $1/25$ で決定する（当選確率80%）。そして、所定の終了条件を満たすと、演出状態制御手段 6 1 4 は、演出状態を継続抽選演出状態に移行する（10）。なお、一旦、第2インターバル演出状態への移行が決定されると、その後は、継続抽選演出状態に漏れるまで、非有利演出状態、準備演出状態、第1インターバル演出状態に移行することはなく、擬似ボーナス演出状態からは第2インターバル演出状態にしか移行しない（11）。したがって、遊技者は、常に、高確率で継続抽選を受けることができる。また、一旦、第2インターバル演出状態への移行が決定されると、継続抽選演出状態において擬似ボーナス演出状態への移行が決定されなかった場合でも、演出状態制御手段 6 1 4 は、直ちに、演出状態を通常演出状態に戻すことなく、非有利演出状態に移行させる（12）。したがって、継続抽選演出状態において擬似ボーナス演出状態への移行が決定されなかった場合、再度、継続抽選演出状態の継続抽選に漏れるまで、擬似ボーナス演出状態→準備演出状態→第1インターバル演出状態→継続抽選演出状態の移行を繰り返すことができる。このように、一旦、第2インターバル演出状態に移行すると、高確率で、擬似ボーナス演出状態→第2インターバル演出状態→継続抽選演出状態の移行を繰り返すことができ、さらに、継続抽選演出状態において擬似ボーナス演出状態への移行が決定されなくとも、少なくとも、再度、擬似ボーナス演出状態→準備演出状態→第1インターバル演出状態→継続抽選演出状態の移行を繰り返すことができるので、遊技者は、より多くのメダルを蓄積することが可能となる。つまり、第2インターバル演出状態は、第1インターバル演出状態よりも有利な状態といえる。

10

20

30

40

50

【0736】

また、上記のように、演出状態制御手段614は、擬似ボーナス演出状態から第2インターバル演出状態に移行する際には、第1インターバル演出状態への移行と異なり、非有利区間に移行させることがない。本実施形態においては、擬似ボーナス演出状態が初回の擬似ボーナス演出状態であった場合、演出状態制御手段614は、第2インターバル演出状態に移行させることはなく、有利区間を、一旦、非有利区間に移行して、必ず、第1インターバル演出状態へ移行させる。そうすると、有利区間が一旦リセットされていることになる。したがって、第2インターバル演出状態へ移行する際に有利区間をリセットしなくても、第2インターバル演出状態を経由する擬似ボーナス演出状態が有利区間の制限により短期に終了することではなく、遊技者は、擬似ボーナス演出状態→第2インターバル演出状態→継続抽選演出状態のループにより、多くのメダルを得ることができる。

10

【0737】

以下、主制御基板500、副制御基板502における具体的処理をフローチャートに基づいて説明する。

【0738】

(主制御基板500のCPU初期化处理)

図82は、主制御基板500におけるCPU初期化处理を説明するフローチャートである。電源基板より電源が供給されると、メインCPU500aにシステムリセットが発生し、メインCPU500aは、以下のCPU初期化处理(S2000)を行う。

【0739】

(ステップS2000-1)

メインCPU500aは、電源投入に応じて、初期設定処理として、メインROM500bから起動プログラムを読み込むとともに、各種処理を実行するために必要な設定処理を行う。

【0740】

(ステップS2000-3)

メインCPU500aは、タイマカウンタにウェイト処理時間を設定する。

【0741】

(ステップS2000-5)

メインCPU500aは、電源断予告信号を検出しているかを判定する。なお、主制御基板500には、電源断検知回路が設けられており、電源電圧が所定値以下になると、電源断検知回路から電源断予告信号が出力される。電源断予告信号を検出している場合には、上記ステップS2000-3に処理を移し、電源断予告信号を検出していない場合には、ステップS2000-7に処理を移す。

30

【0742】

(ステップS2000-7)

メインCPU500aは、上記ステップS2000-3で設定したウェイト処理時間が経過したか否かを判定する。その結果、ウェイト処理時間が経過したと判定した場合にはステップS2000-9に処理を移し、ウェイト時間は経過していないと判定した場合には上記ステップS2000-5に処理を移す。

40

【0743】

(ステップS2000-9)

メインCPU500aは、メインRAM500cへのアクセスを許可するために必要な処理を実行する。

【0744】

(ステップS2000-11)

メインCPU500aは、チェックサム確認処理を実行する。ここでは、メインCPU500aは、チェックサムを算出し、算出したチェックサムが、電源断時に保存されたチェックサムと一致しない(異常である)か、ならびに、バックアップが異常であるかを判定する。そして、メインCPU500aは、バックアップおよびチェックサムのいずれか

50

一方または双方が異常であると判定した場合、バックアップ異常フラグをオンにし、バックアップおよびチェックサムの双方が異常でないと判定した場合、バックアップ異常フラグをオフにする。

【0745】

(ステップS2000-13)

メインCPU500aは、バックアップ異常フラグがオンであるかを判定する。その結果、バックアップ異常フラグがオンであると判定した場合にはステップS2010に処理を移し、バックアップ異常フラグがオンでないと判定した場合にはステップS2020に処理を移す。

【0746】

(ステップS2010)

メインCPU500aは、コールドスタート処理を実行する。なお、このコールドスタート処理については後述する。

【0747】

(ステップS2020)

メインCPU500aは、設定値を切り替える設定値切り替え処理を実行する。なお、この設定値切り替え処理については後述する。

【0748】

(ステップS2030)

メインCPU500aは、電源断直前の状態に戻す状態復帰処理を実行する。なお、この状態復帰処理については後述する。

【0749】

図83は、主制御基板500におけるコールドスタート処理(S2010)を説明するフローチャートである。

【0750】

(ステップS2010-1)

メインCPU500aは、メインRAM500cにおける使用領域をクリアするとともに、使用領域の異常を検出する使用領域RAMチェック処理を実行する。

【0751】

(ステップS2010-3)

メインCPU500aは、メインRAM500cにおける別領域(使用外領域)をクリアするとともに、別領域の異常を検出する別領域RAMチェック処理を実行する。なお、別領域RAMチェック処理において別領域に異常が検出された場合、メインCPU500aは、RAMリードライトエラーフラグをオンにする。

【0752】

(ステップS2010-5)

メインCPU500aは、メインRAM500cの異常を示すエラーコード「EA」をセットする。

【0753】

(ステップS2010-7)

メインCPU500aは、上記ステップS2010-1において異常が検出されたかを判定する。その結果、上記ステップS2010-1において異常が検出されたと判定した場合にはステップS2011に処理を移し、上記ステップS2010-1において異常が検出されていないと判定された場合にはステップS2010-9に処理を移す。

【0754】

(ステップS2010-9)

メインCPU500aは、上記ステップS2010-3において異常が検出されたときにオンになるRAMリードライトエラーフラグを取得する。

【0755】

(ステップS2010-11)

10

20

30

40

50

メインCPU500aは、RAMリードライトエラーフラグがオンであることを判定する。その結果、RAMリードライトエラーフラグがオンであると判定した場合にはステップS2011に処理を移し、RAMリードライトエラーフラグがオンでないと判定された場合にはステップS2020に処理を移す。

【0756】

(ステップS2020)

メインCPU500aは、設定値を切り替える設定値切り替え処理を実行する。なお、この設定値切り替え処理については後述する。

【0757】

(ステップS2010-13)

メインCPU500aは、バックアップエラーであることを示すエラーコード「E7」をセットする。

【0758】

(ステップS2011)

メインCPU500aは、エラーにより遊技の進行を停止させるためのエラー停止処理を実行する。なお、このエラー停止処理については後述する。

【0759】

図84は、主制御基板500におけるエラー停止処理(S2011)を説明するフローチャートである。

【0760】

(ステップS2011-1)

メインCPU500aは、スタックポインタのアドレスとして、初期スタックポインタ値をセットする。

【0761】

(ステップS2011-3)

メインCPU500aは、エラー表示および警告音設定を行うエラー設定処理を実行する。

【0762】

(ステップS2011-5)

メインCPU500aは、外部信号1~3に対応するビットの出力イメージをオフにする外部信号1~3出力ビットオフをセットする。

【0763】

(ステップS2011-7)

メインCPU500aは、上記ステップS2011-5でセットしたビットについて、出力イメージを更新する出力ポートイメージセット処理を実行する。

【0764】

(ステップS2011-9)

メインCPU500aは、永久ループに移行する。これにより、遊技の進行が停止することになる。

【0765】

図85は、主制御基板500における設定値切り替え処理(S2020)を説明するフローチャートである。

【0766】

(ステップS2020-1)

メインCPU500aは、入力ポート1の信号を取得し、取得した入力ポート1の信号に基づいて、設定値切り替え条件が成立していないかを判定する。その結果、設定値切り替え条件が成立していないと判定した場合には当該設定値切り替え処理を終了し、設定値切り替え条件が成立していると判定した場合にはステップS2020-3に処理を移す。ここで、入力ポート1の信号には、前面上扉404および前面下扉406が開放されているか否かを示す信号、および、設定ドアキー438がオンにされているか否かを示す信号

10

20

30

40

50

が含まれる。そして、ここでは、前面上扉 404 および前面下扉 406 が開放されていることを示す信号、ならびに、設定ドアキー 438 がオンにされていることを示す信号を取得した場合に、設定値切り替え条件が成立していると判定している。

【0767】

(ステップ S2020-3)

メイン CPU 500a は、メイン RAM 500c において設定変更時にクリアすべき使用領域をクリアする RAM クリア処理を実行する。

【0768】

(ステップ S2020-5)

メイン CPU 500a は、設定値切り替え時データテーブルのテーブルデータをメイン RAM 500c に転送するテーブル内容セット処理を実行する。

10

【0769】

(ステップ S2020-7)

メイン CPU 500a は、設定値の変更を開始することを示す設定変更開始コマンドを送信バッファにセットする。

【0770】

(ステップ S2020-9)

メイン CPU 500a は、入力ポートの信号の立ち上がりエッジ（オンエッジ）、および、立ち下がりエッジ（オフエッジ）を検出するエッジチェック処理を実行する。

20

【0771】

(ステップ S2020-11)

メイン CPU 500a は、現在の設定値を示す設定値データを取得する。

【0772】

(ステップ S2020-13)

メイン CPU 500a は、上記ステップ S2020-9 において設定変更スイッチのオンエッジを検出していないかを判定する。その結果、設定変更スイッチのオンエッジを検出していないと判定した場合にはステップ S2020-17 に処理を移し、設定変更スイッチのオンエッジを検出したと判定した場合にはステップ S2020-15 に処理を移す。

【0773】

(ステップ S2020-15)

メイン CPU 500a は、設定値データを 1 インクリメントする。

30

【0774】

(ステップ S2020-17)

メイン CPU 500a は、設定値データが、設定値として設定可能な範囲（1～6）内であるかを判定する。その結果、設定値データが範囲内であると判定した場合にはステップ S2020-21 に処理を移し、設定値データが範囲内でないと判定した場合にはステップ S2020-19 に処理を移す。

【0775】

(ステップ S2020-19)

メイン CPU 500a は、設定値データを 0 にセットする。

40

【0776】

(ステップ S2020-21)

メイン CPU 500a は、上記ステップ S2020-15 または上記ステップ S2020-19 でインクリメントまたはセットされた値に設定値データを更新する。

【0777】

(ステップ S2020-23)

メイン CPU 500a は、設定値をメインクレジット表示部 430 に表示する表示データ変換処理を実行する。

【0778】

(ステップ S2020-25)

50

メインCPU500aは、設定変更スイッチのオンエッジを検出していないかを判定する。その結果、設定変更スイッチのオンエッジを検出していないと判定した場合にはステップS2020-31に処理を移し、設定変更スイッチのオンエッジを検出していると判定した場合にはステップS2020-27に処理を移す。

【0779】

(ステップS2020-27)

メインCPU500aは、設定変更スイッチがオンであるかを判定する。その結果、設定変更スイッチがオンであると判定した場合にはステップS2020-27に処理を移し、設定変更スイッチがオンでないと判定した場合にはステップS2020-29に処理を移す。

【0780】

(ステップS2020-29)

メインCPU500aは、設定値変更スイッチ間隔タイマをセットする。

【0781】

(ステップS2020-31)

メインCPU500aは、設定変更スイッチ間隔タイマが0になるまで待つタイマウェイト処理を実行する。

【0782】

(ステップS2020-33)

メインCPU500aは、スタートスイッチ418のオンエッジを検出していないかを判定する。その結果、スタートスイッチ418のオンエッジを検出していないと判定した場合にはステップS2020-9に処理を移し、スタートスイッチ418のオンエッジを検出していると判定した場合にはステップS2020-35に処理を移す。

【0783】

(ステップS2020-35)

メインCPU500aは、設定ドアキー438がオフであるかを判定する。その結果、設定ドアキー438がオフであると判定した場合にはステップS2020-35に処理を移し、設定ドアキー438がオフでないと判定した場合にはステップS2020-37に処理を移す。

【0784】

(ステップS2020-37)

メインCPU500aは、設定ドアキー438がオンであるかを判定する。その結果、設定ドアキー438がオンであると判定した場合にはステップS2020-37に処理を移し、設定ドアキー438がオンでないと判定した場合にはステップS2021に処理を移す。

【0785】

(ステップS2021)

メインCPU500aは、初期化スタートを開始する初期化スタート処理を実行する。なお、この初期化スタート処理については後述する。

【0786】

図86は、主制御基板500における初期化スタート処理(S2021)を説明するフローチャートである。

【0787】

(ステップS2021-1)

メインCPU500aは、設定値の変更が終了したことを示す設定変更終了コマンドを送信バッファにセットする。

【0788】

(ステップS2021-3)

メインCPU500aは、設定値の変更が終了したときの状態を示す設定変更状態コマンドを送信バッファにセットする。

10

20

30

40

50

【0789】

(ステップS2021-5)

メインCPU500aは、初期化スタート時ウェイトタイマをセットする。

【0790】

(ステップS2021-7)

メインCPU500aは、初期化スタート時ウェイトタイマが0になるまで待つタイマウェイト処理を実行する。

【0791】

(ステップS2021-9)

メインCPU500aは、メインRAM500cのうちの別領域をクリアする設定変更時RAMクリア処理を実行する。

10

【0792】

(ステップS2021-11)

メインCPU500aは、メインRAM500cにおいて設定変更時にクリアすべき使用領域をクリアするRAMクリア処理を実行する。

【0793】

(ステップS2021-13)

メインCPU500aは、現在の遊技状態を示す遊技状態コマンドを送信バッファにセットする。

【0794】

(ステップS2100)

メインCPU500aは、遊技を開始するための遊技開始処理を実行する。なお、この遊技開始処理については後述する。

20

【0795】

図87は、主制御基板500における状態復帰処理(S2030)を説明するフローチャートである。

【0796】

(ステップS2030-1)

メインCPU500aは、スタックポインタを復帰させる。

【0797】

(ステップS2030-3)

メインCPU500aは、メインRAM500cのうちの未使用領域をクリアする未使用領域クリア処理を実行する。

30

【0798】

(ステップS2030-5)

メインCPU500aは、スタックポインタ保存バッファをクリアする。

【0799】

(ステップS2030-7)

メインCPU500aは、電源断復帰後フラグを設定(オン)する。

【0800】

(ステップS2030-9)

メインCPU500aは、入力ポートのイメージを更新するポート入力処理を実行する。

40

【0801】

(ステップS2030-11)

メインCPU500aは、上記ステップS2030-9で更新された入力ポートのイメージに基づいて、操作対象ビットの情報を抽出する操作対象ビット抽出処理を実行する。

【0802】

(ステップS2030-13)

メインCPU500aは、上記ステップS2030-11で抽出した操作対象ビットを、前回状態の操作対象ビットとしてセットする。

50

【0803】

(ステップS2030-15)

メインCPU500aは、リール410a、410b、410cのモータフェーズを取得する。ここで、リール410a、410b、410cの状態として、モータフェーズが設定されている。モータフェーズは、リール410a、410b、410cの動作状態、すなわち、加速中、定常回転中、停止中、待機中を示す。具体的に、モータフェーズに割り当てられた1バイト（記憶単位）の変数が、そのステッピングモータ452の動作状態に応じて、加速中=3、定常回転中=2、停止中=1、待機中=0といった値に変化する。

【0804】

(ステップS2030-17)

メインCPU500aは、上記ステップS2030-15で取得したモータフェーズに基づいて、リール410a、410b、410cのいずれもが定常回転中および加速中でないかを判定する。その結果、リール410a、410b、410cのいずれもが定常回転中および加速中でないと判定した場合にはステップS2030-21に処理を移し、リール410a、410b、410cのいずれかが定常回転中または加速中であると判定した場合にはステップS2030-19に処理を移す。

【0805】

(ステップS2030-19)

メインCPU500aは、リール410a、410b、410cのエラー検出時の設定を行う回転エラー処理を実行する。

【0806】

(ステップS2030-21)

メインCPU500aは、退避していたレジスタ群を復帰させる。

【0807】

(ステップS2030-23)

メインCPU500aは、割込みを許可し、当該状態復帰処理を終了する。これにより、メインCPU500aは、電源断直前の状態に復帰する。

【0808】

図88は、主制御基板500における遊技開始処理（S2100）を説明するフローチャートである。

【0809】

(ステップS2100-1)

メインCPU500aは、再遊技であるか否かを示す再遊技状態識別信号を出力するための再遊技状態識別信号出力設定処理を実行する。

【0810】

(ステップS2100-3)

メインCPU500aは、メダルの投入枚数（ベット枚数）を表示する投入枚数表示器に対応するビットをオフ（消灯）するための投入枚数表示器出力ビットオフをセットする。

【0811】

(ステップS2100-5)

メインCPU500aは、上記ステップS2100-3でセットしたビットについて、出力イメージを更新する出力ポートイメージセット処理を実行する。

【0812】

(ステップS2100-7)

メインCPU500aは、遊技開始ウェイトタイマをセットする。

【0813】

(ステップS2100-9)

メインCPU500aは、遊技開始ウェイトタイマが0になるまで待つタイマウェイト処理を実行する。

【0814】

10

20

30

40

50

(ステップ S 2 1 0 0 - 1 1)

メイン CPU 5 0 0 a は、メイン RAM 5 0 0 c における使用領域のうち、1 遊技毎にクリアすべき領域をクリアする 1 遊技 RAM クリア処理を実行する。

【 0 8 1 5 】

(ステップ S 2 1 0 0 - 1 3)

メイン CPU 5 0 0 a は、ボーナス信号を設定するボーナス信号設定処理を実行する。

【 0 8 1 6 】

(ステップ S 2 1 0 0 - 1 5)

メイン CPU 5 0 0 a は、入力ポートイメージのエッジ情報をクリアするエッジクリア処理を実行する。

【 0 8 1 7 】

(ステップ S 2 2 0 0)

メイン CPU 5 0 0 a は、メダルの投入を受け付ける遊技メダル投入処理を実行する。なお、この遊技メダル投入処理については後述する。

【 0 8 1 8 】

図 8 9 は、主制御基板 5 0 0 における遊技メダル投入処理 (S 2 2 0 0) を説明するフローチャートである。

【 0 8 1 9 】

(ステップ S 2 2 0 0 - 1)

メイン CPU 5 0 0 a は、各種エラーの検出結果の確認を行うエラー確認処理を実行する。

【 0 8 2 0 】

(ステップ S 2 2 0 0 - 3)

メイン CPU 5 0 0 a は、入力ポートの信号の立ち上がりエッジ (オンエッジ) 、および、立ち下がりエッジ (オフエッジ) を検出するエッジチェック処理を実行する。

【 0 8 2 1 】

(ステップ S 2 2 0 0 - 5)

メイン CPU 5 0 0 a は、前面上扉 4 0 4 または前面下扉 4 0 6 が開放されているときに 1 が立つドア開放エラー検出フラグを取得する。

【 0 8 2 2 】

(ステップ S 2 2 0 0 - 7)

メイン CPU 5 0 0 a は、上記ステップ S 2 2 0 0 - 5 で取得したドア開放エラー検出フラグに基づき、前面上扉 4 0 4 および前面下扉 4 0 6 が閉鎖されているかを判定する。その結果、前面上扉 4 0 4 および前面下扉 4 0 6 が閉鎖されていると判定した場合にはステップ S 2 2 0 0 - 1 7 に処理を移し、前面上扉 4 0 4 または前面下扉 4 0 6 の少なくとも一方が閉鎖されていないと判定した場合にはステップ S 2 2 0 0 - 9 に処理を移す。

【 0 8 2 3 】

(ステップ S 2 2 0 0 - 9)

メイン CPU 5 0 0 a は、前面上扉 4 0 4 または前面下扉 4 0 6 の少なくとも一方が開放されていることを示すエラーコード「 E 8 」をセットする。

【 0 8 2 4 】

(ステップ S 2 2 0 0 - 1 1)

メイン CPU 5 0 0 a は、エラー表示、警告音の要求、ならびに、エラー復帰待ちを行うエラーウェイト処理を実行する。

【 0 8 2 5 】

(ステップ S 2 2 0 0 - 1 3)

メイン CPU 5 0 0 a は、設定値を確認する設定値確認処理を実行する。

【 0 8 2 6 】

(ステップ S 2 2 0 0 - 1 5)

メイン CPU 5 0 0 a は、入力ポートイメージのエッジ情報をクリアするエッジクリア

10

20

30

40

50

処理を実行する。

【0827】

(ステップS2200-17)

メインCPU500aは、貯留（クレジット）されているメダルを払い戻すためのクレジットスイッチ（不図示）が押下されている場合に、貯留されているメダルを払い戻すクレジットボタンチェック処理を実行する。

【0828】

(ステップS2200-19)

メインCPU500aは、メダルをベットする遊技メダル投入ボタン関連処理を実行する。ここでは、ベットスイッチ416が押下された場合に、貯留（クレジット）されているメダルを規定数までベットするとともに、ベットした枚数分だけ貯留枚数を減算する。また、メダル投入口414aを通じてメダルが投入された場合、規定数までメダルをベットし、規定数よりも多くメダルが投入された場合、その分だけ貯留枚数に加算する。

【0829】

(ステップS2200-21)

メインCPU500aは、投入枚数が規定数であるかを確認する遊技メダル取得処理を実行する。

【0830】

(ステップS2200-23)

メインCPU500aは、上記ステップS2200-21の確認結果に基づき、投入枚数が規定数でないかを判定する。その結果、投入枚数が規定数でないとは判定した場合にはステップS2200-1に処理を移し、投入枚数が規定数であると判定した場合にはステップS2200-25に処理を移す。

【0831】

(ステップS2200-25)

メインCPU500aは、スタートスイッチ418の操作が有効になったか否かを示すスタート表示器（不図示）をオン（点灯）するためのスタート表示器出力ビットをセットする。

【0832】

(ステップS2200-27)

メインCPU500aは、スタートスイッチ418の立ち下がりエッジ（押下）を検出していなかを判定する。その結果、スタートスイッチ418の立ち下がりエッジを検出していないとは判定した場合にはステップS2200-1に処理を移し、スタートスイッチ418の立ち下がりエッジを検出していると判定した場合にはステップS2200-29に処理を移す。

【0833】

(ステップS2200-29)

メインCPU500aは、メイン払出表示部432の表示をクリアするためにメイン払出表示部バッファをクリアする。

【0834】

(ステップS2200-31)

メインCPU500aは、再遊技状態識別信号をクリアする再遊技状態識別信号クリア処理を実行する。

【0835】

(ステップS2200-33)

メインCPU500aは、スタート表示器をオフ（消灯）するためのブロッカー閉塞前処理を実行する。

【0836】

(ステップS2200-35)

メインCPU500aは、スタートスイッチ418が押下されたことを示すレバー押下

10

20

30

40

50

コマンドを送信バッファにセットする。

【0837】

(ステップS2300)

メインCPU500aは、当選種別抽選を行う内部抽選処理を実行する。なお、この内部抽選処理については後述する。

【0838】

図90は、主制御基板500における内部抽選処理(S2300)を説明するフローチャートである。

【0839】

(ステップS2300-1)

メインCPU500aは、設定値データを取得する。

【0840】

(ステップS2300-3)

メインCPU500aは、設定値異常エラーを示すエラーコード「EC」をセットする。

【0841】

(ステップS2300-5)

メインCPU500aは、上記ステップS2300-1で取得した設定値データが異常であるかを判定する。その結果、設定値データが異常であると判定した場合にはステップS2011に処理を移し、設定値データが異常でないと判定した場合にはステップS2300-7に処理を移す。

【0842】

(ステップS2300-7)

メインCPU500aは、乱数発生器500dによって更新された当選種別抽選乱数を取得する。

【0843】

(ステップS2300-9)

メインCPU500aは、遊技状態に係るオフセット値を取得する状態オフセット取得処理を実行する。

【0844】

(ステップS2300-11)

メインCPU500aは、内部抽選エリア定義テーブル(当選種別抽選テーブル)のアドレスをセットする。

【0845】

(ステップS2300-13)

メインCPU500aは、上記ステップS2300-11でセットしたアドレスに対して、上記ステップS2300-9で取得したオフセット値を加算したアドレスに示される値を当選領域の初期値としてセットする。ここでは、現在の遊技状態の当選種別抽選テーブルにおける最初の当選領域が初期値としてセットされることになる。

【0846】

(ステップS2300-15)

メインCPU500aは、その当選領域の当選範囲を示す数値である抽選データを取得するとともに、当選領域を1ずらす抽選データ取得処理を実行する。

【0847】

(ステップS2300-17)

メインCPU500aは、当選種別抽選を行わないかを判定する。その結果、当選種別抽選を行わないと判定した場合にはステップS2300-21に処理を移し、当選種別抽選を行うと判定した場合にはステップS2300-19に処理を移す。

【0848】

(ステップS2300-19)

メインCPU500aは、乱数値から抽選データを減算する。

10

20

30

40

50

【0849】

(ステップS2300-21)

メインCPU500aは、上記ステップS2300-19の減算結果が負であるか、すなわち、当選種別抽選によって、その当選領域に当選しているかを判定する。その結果、当選種別抽選に当選していると判定した場合にはステップS2400に処理を移し、当選種別抽選に当選していないと判定した場合にはステップS2300-23に処理を移す。

【0850】

(ステップS2300-23)

メインCPU500aは、当選種別抽選が終了でないかを判定する。その結果、当選種別抽選が終了でないと判定した場合にはステップS2300-15に処理を移し、当選種別抽選が終了であると判定した場合にはステップS2300-25に処理を移す。

10

【0851】

(ステップS2300-25)

メインCPU500aは、トリガー役種別をクリアする。

【0852】

(ステップS2400)

メインCPU500aは、当選領域および遊技状態に基づいて、図柄コードを設定する図柄コード設定処理を実行する。なお、この図柄コード設定処理については後述する。

【0853】

図91は、主制御基板500における図柄コード設定処理(S2400)を説明するフローチャートである。

20

【0854】

(ステップS2400-1)

メインCPU500aは、上記ステップS2300で当選した当選領域を取得し、取得した当選領域にボーナス役が含まれる場合には遊技状態を内部中遊技状態に設定する遊技状態設定処理を実行する。

【0855】

(ステップS2400-3)

メインCPU500aは、上記ステップS2400-1で取得した当選領域を停止制御番号として設定する。

30

【0856】

(ステップS2400-5)

メインCPU500aは、上記ステップS2400-1で取得した当選領域に基づいて、当選種別を決定(設定)する。

【0857】

(ステップS2400-7)

メインCPU500aは、上記ステップS2400-3で設定した停止制御番号に基づいて、表示可能な図柄、および、引き込み対象の図柄を示す図柄コードを設定する図柄コード初期設定処理を実行する。

【0858】

(ステップS2400-9)

メインCPU500aは、表示図柄ビットを設定する表示図柄ビット初期値設定処理を実行する。

40

【0859】

(ステップS2400-11)

メインCPU500aは、実行フラグの設定、演出状態に関する各種処理、補助演出に関する処理等を行う実行フラグ設定処理を実行する。

【0860】

(ステップS2400-13)

メインCPU500aは、有利区間に関するコマンドである演出コマンドを送信バッフ

50

ァにセットする。

【0861】

(ステップS2400-15)

メインCPU500aは、当選種別を示す当選情報コマンドを送信バッファにセットする。

【0862】

(ステップS2400-17)

メインCPU500aは、1遊技間タイマを確認する。

【0863】

(ステップS2400-19)

メインCPU500aは、リール410a、410b、410cが回転前であることを示す回胴回転前コマンドを送信バッファにセットする。

【0864】

(ステップS2400-21)

メインCPU500aは、ステッピングモータ452の励磁解放を待つ励磁解放待ち処理を実行する。

【0865】

(ステップS2400-23)

メインCPU500aは、1遊技間タイマが0でないかを判定する。その結果、1遊技間タイマが0でないと判定した場合にはステップS2400-23に処理を移し、1遊技間タイマが0であると判定した場合にはステップS2400-25に処理を移す。

【0866】

(ステップS2400-25)

メインCPU500aは、リール410a、410b、410cの回転を開始させるための回胴開始処理を実行する。ここでは、リール410a、410b、410cのモータフェーズを加速中に設定して各リールの回転を開始させたり、1遊技間タイマを4.1秒に相当する値にセットしたりする。

【0867】

(ステップS2400-27)

メインCPU500aは、リール410a、410b、410cの回転が開始したことを示す回胴開始コマンドを送信バッファにセットする。

【0868】

(ステップS2500)

メインCPU500aは、リール410a、410b、410cの回転中の処理である回胴回転中処理を実行する。なお、この回胴回転中処理については後述する。

【0869】

図92は、主制御基板500における回胴回転中処理(S2500)を説明するフローチャートである。

【0870】

(ステップS2500-1)

メインCPU500aは、ストップスイッチ420a、420b、420cの表示器(不図示)に対応するビットをオフ(消灯)するために停止表示器出力ビットオフ(出力イメージ)をセットする。ここで、停止表示器出力ビットは、3ビットのビット列で構成され、各ビットがそれぞれ3つのストップスイッチ420a、420b、420cの発光色に対応付けられており、青色=1、赤色=0で表される。

【0871】

(ステップS2500-3)

メインCPU500aは、上記ステップS2500-1でセットしたビットについて、出力イメージを更新する出力ポートイメージセット処理を実行する。

【0872】

10

20

30

40

50

(ステップ S 2 5 0 0 - 5)

メイン CPU 5 0 0 a は、各種エラーの検出結果の確認を行うエラー確認処理を実行する。

【 0 8 7 3 】

(ステップ S 2 5 0 0 - 7)

メイン CPU 5 0 0 a は、インデックスフラグを参照し、回転しているリール 4 1 0 a、4 1 0 b、4 1 0 c のインデックスを取得する。なお、インデックスフラグは、リール 4 1 0 a、4 1 0 b、4 1 0 c が定速回転速度に到達した後にしか立たないので、換言すれば、インデックスフラグが立っているということは、リール 4 1 0 a、4 1 0 b、4 1 0 c が定速回転速度に到達していることも示すこととなる。

10

【 0 8 7 4 】

(ステップ S 2 5 0 0 - 9)

メイン CPU 5 0 0 a は、リール 4 1 0 a、4 1 0 b、4 1 0 c 全てのインデックスフラグを検出済みでないかを判定する。その結果、全てのインデックスフラグを検出済みでない判定した場合にはステップ S 2 5 0 0 - 1 に処理を移し、全てのインデックスフラグを検出済みであると判定した場合にはステップ S 2 5 0 0 - 1 1 に処理を移す。

【 0 8 7 5 】

(ステップ S 2 5 0 0 - 1 1)

メイン CPU 5 0 0 a は、停止または停止開始しているリール 4 1 0 a、4 1 0 b、4 1 0 c を示す停止回胴ビットを取得する。ここで、停止回胴ビットは、3 ビットのビット列で構成され、各ビットがそれぞれ 3 つのリール 4 1 0 a、4 1 0 b、4 1 0 c のいずれかに対応付けられており、定速状態 = 1、加速状態、減速状態または停止状態 = 0 で表される。

20

【 0 8 7 6 】

(ステップ S 2 5 0 0 - 1 3)

メイン CPU 5 0 0 a は、上記ステップ S 2 5 0 0 - 1 1 で取得した停止回胴ビットを回胴回転中フラグとして保存する。

【 0 8 7 7 】

(ステップ S 2 5 0 0 - 1 5)

メイン CPU 5 0 0 a は、ストップスイッチ 4 2 0 a、4 2 0 b、4 2 0 c の表示器（不図示）に対応するビットをオン（消灯）するために停止表示器出力ビットオン（出力イメージ）をセットする。

30

【 0 8 7 8 】

(ステップ S 2 5 0 0 - 1 7)

メイン CPU 5 0 0 a は、入力ポート 0 のイメージを取得し、取得したイメージから、操作対象ビットを抽出する操作対象ビット抽出処理を実行する。ここで、操作対象ビットは、3 ビットのビット列で構成され、各ビットがそれぞれ 3 つのストップスイッチ 4 2 0 a、4 2 0 b、4 2 0 c のいずれかに対応付けられており、操作されている = 1、操作されていない = 0 で表される。

【 0 8 7 9 】

40

(ステップ S 2 5 0 0 - 1 9)

メイン CPU 5 0 0 a は、上記ステップ S 2 5 0 0 - 1 3 で取得した回胴回転中フラグと、上記ステップ S 2 5 0 0 - 1 7 で抽出した操作対象ビットとの論理積を演算する。ここで、リール 4 1 0 が回転中であり、かつ、そのリールに対応するストップスイッチ 4 2 0 が操作されていれば、すなわち、操作したストップスイッチ 4 2 0 が有効に回転しているリール 4 1 0 に対応していれば、論理積は 1 となる。

【 0 8 8 0 】

(ステップ S 2 5 0 0 - 2 1)

メイン CPU 5 0 0 a は、上記ステップ S 2 5 0 0 - 1 9 で演算した論理積が 0 である、すなわち、回転しているリール 4 1 0 に対応したストップスイッチ 4 2 0 が操作されて

50

いないかを判定する。その結果、回転しているリール 4 1 0 に対応したストップスイッチ 4 2 0 が操作されていないと判定した場合にはステップ S 2 5 0 0 - 3 に処理を移し、回転しているリール 4 1 0 に対応したストップスイッチ 4 2 0 が操作されていると判定した場合にはステップ S 2 5 0 0 - 2 3 に処理を移す。

【0881】

(ステップ S 2 5 0 0 - 2 3)

メイン CPU 5 0 0 a は、停止表示器出力ビットが含まれる出力イメージを取得し、取得した出力イメージと、上記ステップ S 2 5 0 0 - 1 9 で演算した論理積との論理積を演算する。ここでは、操作されたストップスイッチ 4 2 0 が、赤色点灯中である場合に論理積のビットが 0 となり、青色点灯中である場合に論理積のビットが 1 となる。

10

【0882】

(ステップ S 2 5 0 0 - 2 5)

メイン CPU 5 0 0 a は、上記ステップ S 2 5 0 0 - 2 3 で演算した論理積が 0 であるか、すなわち、操作されたストップスイッチ 4 2 0 が赤色点灯中であるかを判定する。その結果、操作されたストップスイッチ 4 2 0 が赤色点灯中であると判定した場合にはステップ S 2 5 0 0 - 1 に処理を移し、操作されたストップスイッチ 4 2 0 が赤色点灯中でないと判定した場合にはステップ S 2 5 0 0 - 2 7 に処理を移す。

【0883】

(ステップ S 2 5 0 0 - 2 7)

メイン CPU 5 0 0 a は、操作されたストップスイッチ 4 2 0 が有効でないかを判定する。その結果、操作されたストップスイッチ 4 2 0 が有効でないと判定した場合にはステップ S 2 5 0 0 - 1 に処理を移し、操作されたストップスイッチ 4 2 0 が有効であると判定した場合にはステップ S 2 5 0 0 - 2 9 に処理を移す。なお、ここでは、操作されたストップスイッチ 4 2 0 が 1 つであるか否かを判定している。そして、操作されたストップスイッチ 4 2 0 が 1 つであると判定した場合にはステップ S 2 5 0 0 - 2 9 に処理を移し、操作されたストップスイッチ 4 2 0 が 1 つでない、すなわち、2 つ以上であると判定した場合にはステップ S 2 5 0 0 - 1 に処理を移す。

20

【0884】

(ステップ S 2 5 0 0 - 2 9)

メイン CPU 5 0 0 a は、操作されたストップスイッチ 4 2 0 に対応するリール 4 1 0 を停止させるための各種パラメータを取得する停止制御回胴設定処理を実行する。

30

【0885】

(ステップ S 2 5 0 0 - 3 1)

メイン CPU 5 0 0 a は、割込みを禁止する。

【0886】

(ステップ S 2 5 0 0 - 3 3)

メイン CPU 5 0 0 a は、有効ライン A 上に位置する図柄の図柄番号を押下基準位置として導出する押下基準位置取得処理を実行する。

【0887】

(ステップ S 2 5 0 0 - 3 5)

メイン CPU 5 0 0 a は、リール 4 1 0 の滑りコマ数を決定する滑りコマ数取得処理を実行する。

40

【0888】

(ステップ S 2 6 0 0)

メイン CPU 5 0 0 a は、操作されたストップスイッチ 4 2 0 に対応するリール 4 1 0 を停止させる回胴停止処理を実行する。なお、この回胴停止処理については後述する。

【0889】

図 9 3 は、主制御基板 5 0 0 における回胴停止処理 (S 2 6 0 0) を説明するフローチャートである。

【0890】

50

(ステップ S 2 6 0 0 - 1)

メイン CPU 5 0 0 a は、上記ステップ S 2 5 0 0 - 3 5 で導出した押下基準位置を取得する。

【 0 8 9 1】

(ステップ S 2 6 0 0 - 3)

メイン CPU 5 0 0 a は、上記ステップ S 2 6 0 0 - 1 で取得した押下基準位置に対して、上記ステップ S 2 5 0 0 - 3 7 で決定した滑りコマ数を補正することにより、停止要求番号を算定する。

【 0 8 9 2】

(ステップ S 2 6 0 0 - 5)

メイン CPU 5 0 0 a は、停止要求フラグを設定する (1 にする)。停止要求フラグは、並行して動作するプログラムに対し、対象となるリール 4 1 0 の停止処理を要求するためのフラグであり、停止要求フラグを 1 とすることで、停止要求番号に対応する図柄を有効ライン A 上に停止することが可能となる。かかる停止要求フラグおよび上記の停止要求番号は、並行して動作するプログラムにより読み出され、リール 4 1 0 の停止処理が行われる。なお、停止処理が完了すると、そのプログラムによって、停止要求フラグは 0 (O F F) にリセットされる。

【 0 8 9 3】

(ステップ S 2 6 0 0 - 7)

メイン CPU 5 0 0 a は、割込みを許可する。

【 0 8 9 4】

(ステップ S 2 6 0 0 - 9)

メイン CPU 5 0 0 a は、リール 4 1 0 の停止順序を示す停止情報コマンドを送信バッファにセットする。

【 0 8 9 5】

(ステップ S 2 6 0 0 - 1 1)

メイン CPU 5 0 0 a は、ストップスイッチ 4 2 0 の表示器 (不図示) に対応するビットをオフ (消灯) するために停止表示器出力ビットオフ (出力イメージ) をセットする。

【 0 8 9 6】

(ステップ S 2 6 0 0 - 1 3)

メイン CPU 5 0 0 a は、上記ステップ S 2 6 0 0 - 1 1 でセットしたビットについて、出力イメージを更新する出力ポートイメージセット処理を実行する。

【 0 8 9 7】

(ステップ S 2 6 0 0 - 1 5)

メイン CPU 5 0 0 a は、表示図柄ビットを設定する表示図柄ビット設定処理を実行する。

【 0 8 9 8】

(ステップ S 2 6 0 0 - 1 7)

メイン CPU 5 0 0 a は、次のリール 4 1 0 を停止させるための次回胴設定前処理を実行する。

【 0 8 9 9】

(ステップ S 2 6 0 0 - 1 9)

メイン CPU 5 0 0 a は、全てのリール 4 1 0 の停止処理が終了済みでないかを判定する。その結果、全てのリール 4 1 0 の停止処理が終了済みでないとは判定した場合にはステップ S 2 5 0 0 に処理を移し、全てのリール 4 1 0 の停止処理が終了済みであると判定した場合にはステップ S 2 6 0 0 - 2 1 に処理を移す。

【 0 9 0 0】

(ステップ S 2 6 0 0 - 2 1)

メイン CPU 5 0 0 a は、いずれかのリール 4 1 0 について停止要求フラグがオンである、すなわち、全てのリール 4 1 0 が停止済みでないかを判定する。その結果、全てのリ

10

20

30

40

50

ール410が停止済みでないと判定した場合にはステップS2600-21に処理を移し、全てのリール410が停止済みであると判定した場合にはステップS2600-23に処理を移す。

【0901】

(ステップS2600-23)

メインCPU500aは、各種エラーの検出結果の確認を行うエラー確認処理を実行する。

【0902】

(ステップS2600-25)

メインCPU500aは、操作対象ビットの情報を抽出する操作対象ビット抽出処理を実行する。

10

【0903】

(ステップS2600-27)

メインCPU500aは、上記ステップS2600-25で取得した操作対象ビットに基づいて、ストップスイッチ420が押下されているかを判定する。その結果、ストップスイッチ420が押下されていると判定した場合にはステップS2600-23に処理を移し、ストップスイッチ420が押下されていないと判定した場合にはステップS2700に処理を移す。

【0904】

(ステップS2700)

メインCPU500aは、入賞した当選役を判定する表示判定処理を実行する。なお、この表示判定処理については後述する。

20

【0905】

図94は、主制御基板500における表示判定処理(S2700)を説明するフローチャートである。

【0906】

(ステップS2700-1)

メインCPU500aは、メイン払出表示部432のバッファをクリアする。

【0907】

(ステップS2700-3)

メインCPU500aは、有効ラインA上に表示された図柄組み合わせと、有効ラインA上に表示許可された図柄組み合わせとが一致するか否かによって、表示判定異常が発生しているかを判定する表示判定異常検出処理を実行する。

30

【0908】

(ステップS2700-5)

メインCPU500aは、表示判定異常(エラー)であることを示すエラーコード「E」をセットする。

【0909】

(ステップS2700-7)

メインCPU500aは、上記ステップS2700-3の判定結果に基づき、表示判定異常であるかを判定する。その結果、表示判定異常であると判定した場合にはステップS2011に処理を移し、表示判定異常でないと判定した場合にはステップS2700-9に処理を移す。

40

【0910】

(ステップS2700-9)

メインCPU500aは、有効ラインA上に停止(表示)された図柄組み合わせに基づいて、入賞した当選役を決定する表示図柄識別生成処理を実行する。

【0911】

(ステップS2700-11)

メインCPU500aは、払出枚数の初期値として0をセットする。

50

【0912】

(ステップS2700-13)

メインCPU500aは、小役が入賞したかを判定する。その結果、小役が入賞したと判定した場合にはステップS2700-15に処理を移し、小役が入賞していないと判定した場合にはステップS2700-35に処理を移す。

【0913】

(ステップS2700-15)

メインCPU500aは、小役が入賞したことを示す入賞フラグをオンにする。

【0914】

(ステップS2700-17)

メインCPU500aは、入賞した小役に応じた払出枚数を設定する払出枚数設定処理を実行する。

【0915】

(ステップS2700-19)

メインCPU500aは、有利区間でないかを判定する。その結果、有利区間でないと判定した場合にはステップS2800に処理を移し、有利区間であると判定した場合にはステップS2700-21に処理を移す。

【0916】

(ステップS2700-21)

メインCPU500aは、有利区間中の純増枚数をカウントする有利区間MYカウンタの値を取得する。

【0917】

(ステップS2700-23)

メインCPU500aは、上記ステップS2700-23で取得した有利区間MYカウンタの値に払出枚数を加算する。

【0918】

(ステップS2700-25)

メインCPU500aは、当該遊技の投入枚数を取得する。

【0919】

(ステップS2700-27)

メインCPU500aは、上記ステップS2700-23で加算した値から投入枚数を減算する。

【0920】

(ステップS2700-29)

メインCPU500aは、上記ステップS2700-27の減算結果が負でないかを判定する。その結果、減算結果が負でないとは判定した場合にはステップS2700-33に処理を移し、減算結果が負であると判定した場合にはステップS2700-31に処理を移す。

【0921】

(ステップS2700-31)

メインCPU500aは、有利区間MYカウンタの値をクリアする(0にする)。

【0922】

(ステップS2700-33)

メインCPU500aは、上記ステップS2700-27で減算した値、または、上記ステップS2700-31でクリアした値に、有利区間MYカウンタの値を更新する。

【0923】

(ステップS2700-35)

メインCPU500aは、リプレイ役が入賞していなかを判定する。その結果、リプレイ役が入賞していないとは判定した場合にはステップS2800に処理を移し、リプレイ役が入賞していると判定した場合にはステップS2700-37に処理を移す。

10

20

30

40

50

【0924】

(ステップS2700-37)

メインCPU500aは、払出枚数に投入枚数をセットする。

【0925】

(ステップS2700-39)

メインCPU500aは、再遊技作動中フラグをオンにする。

【0926】

(ステップS2700-41)

メインCPU500aは、自動投入枚数をセットする。

【0927】

(ステップS2800)

メインCPU500aは、メダルを払い出す払出処理を実行する。なお、この払出処理については後述する。

【0928】

図95は、主制御基板500における払出処理(S2800)を説明するフローチャートである。

【0929】

(ステップS2800-1)

メインCPU500aは、再遊技作動中フラグを取得する。

【0930】

(ステップS2800-3)

メインCPU500aは、メダルの払い出しが開始されたことを示す払出開始コマンドを送信バッファにセットする。

【0931】

(ステップS2800-5)

メインCPU500aは、上記ステップS2800-1で取得した再遊技作動中フラグに基づいて、リプレイ役が入賞したかを判定する。その結果、リプレイ役が入賞したと判定した場合にはステップS2800-41に処理を移し、リプレイ役が入賞していないと判定した場合にはステップS2800-7に処理を移す。

【0932】

(ステップS2800-7)

メインCPU500aは、メイン払出表示部432に0を表示するためのメイン表示器表示処理を実行する。

【0933】

(ステップS2800-9)

メインCPU500aは、払い出しがない(払出枚数が0枚)を判定する。その結果、払い出しがないと判定した場合にはステップS2800-35に処理を移し、払い出しがあると判定した場合にはステップS2800-11に処理を移す。

【0934】

(ステップS2800-11)

メインCPU500aは、貯留枚数が50枚以上であるかを判定する。その結果、貯留枚数が50枚以上であると判定した場合にはステップS2800-13に処理を移し、貯留枚数が50枚以上でないと判定した場合にはステップS2800-15に処理を移す。

【0935】

(ステップS2800-13)

メインCPU500aは、メダル払出装置442からメダルを1枚払い出させるメダル払出装置制御処理を実行し、ステップS2800-23に処理を移す。

【0936】

(ステップS2800-15)

メインCPU500aは、払出開始間隔タイマをセットする。

10

20

30

40

50

【0937】

(ステップS2800-17)

メインCPU500aは、払出開始タイマが0でない、すなわち、初回払出時であるかを判定する。その結果、初回払出時であると判定した場合にはステップS2800-21に処理を移し、初回払出時でないと判定した場合にはステップS2800-19に処理を移す。

【0938】

(ステップS2800-19)

メインCPU500aは、払出開始間隔タイマが0になるまで待つタイマウェイト処理を実行する。

【0939】

(ステップS2800-21)

メインCPU500aは、貯留枚数を1インクリメントする。

【0940】

(ステップS2800-23)

メインCPU500aは、1枚のメダルが払い出されたことを示す払出実行コマンドを送信バッファにセットする。

【0941】

(ステップS2800-25)

メインCPU500aは、既に払い出された払出枚数をメイン払出表示部432に表示するためのメイン表示器表示前処理を実行する。

【0942】

(ステップS2800-27)

メインCPU500aは、ボーナス遊技状態でないかを判定する。その結果、ボーナス遊技状態でないと判定した場合にはステップS2800-31に処理を移し、ボーナス遊技状態であると判定した場合にはステップS2800-29に処理を移す。

【0943】

(ステップS2800-29)

メインCPU500aは、ボーナス遊技状態において払い出されたメダルの枚数であるボーナス作動中獲得枚数を1インクリメントする。

【0944】

(ステップS2800-31)

メインCPU500aは、払出枚数のメダルの払い出しが終了していないかを判定する。その結果、払い出しが終了していないと判定した場合にはステップS2800-11に処理を移し、払い出しが終了していると判定した場合にはステップS2800-33に処理を移す。

【0945】

(ステップS2800-33)

メインCPU500aは、メダルの払い出しを終了するための払出終了処理を実行する。

【0946】

(ステップS2800-35)

メインCPU500aは、オーバーエラーが検出されていないかを判定する。その結果、オーバーエラーが検出されていないと判定した場合にはステップS2800-41に処理を移し、オーバーエラーが検出されていると判定した場合にはステップS2800-37に処理を移す。

【0947】

(ステップS2800-37)

メインCPU500aは、オーバーエラーを示すエラーコード「E5」をセットする。

【0948】

(ステップS2800-39)

10

20

30

40

50

メインCPU500aは、エラー表示、警告音の要求、ならびに、エラー復帰待ちを行うエラーウェイト処理を実行する。

【0949】

(ステップS2800-41)

メインCPU500aは、メダルの払い出しが終了したことを示す払出終了コマンドを送信バッファにセットする。

【0950】

(ステップS2900)

メインCPU500aは、遊技状態の移行、有利区間を管理する処理等を行う遊技移行処理を実行する。なお、この遊技移行処理については後述する。

10

【0951】

図96は、主制御基板500における遊技移行処理(S2900)を説明するフローチャートである。

【0952】

(ステップS2900-1)

メインCPU500aは、再遊技作動中フラグを取得し、取得した再遊技作動中フラグに基づいて、次遊技が再遊技であることを示すリプレイ表示器(不図示)に対応するビットをオンまたはオフするために停止表示器出力ビットオフ(出力イメージ)をセットし、セットした出力イメージの出力ビットを更新するリプレイ表示器制御処理を実行する。

【0953】

20

(ステップS2900-3)

メインCPU500aは、ボーナス役が入賞した場合に、ボーナス遊技状態を制御するための各種パラメータを設定する役物作動図柄表示処理を実行する。

【0954】

(ステップS2900-5)

メインCPU500aは、ボーナス遊技状態において、ボーナス作動中獲得枚数が所定枚数に到達した場合に、遊技状態を非内部遊技状態に移行させるボーナス作動終了処理を実行する。

【0955】

(ステップS2900-7)

30

メインCPU500aは、有利区間を管理する有利区間更新処理を実行する。

【0956】

(ステップS2900-9)

メインCPU500aは、次遊技がAT演出状態でないかを判定する。その結果、次遊技がAT演出状態でないと判定した場合にはステップS2900-15に処理を移し、次遊技がAT演出状態であると判定した場合にはステップS2900-11に処理を移す。

【0957】

(ステップS2900-11)

メインCPU500aは、ボーナス遊技状態でないかを判定する。その結果、ボーナス遊技状態でないと判定した場合にはステップS2900-15に処理を移し、ボーナス遊技状態であると判定した場合にはステップS2900-13に処理を移す。

40

【0958】

(ステップS2900-13)

メインCPU500aは、区間表示器160を点灯させるための有利ランプフラグをオンにセットする。

【0959】

(ステップS2900-15)

メインCPU500aは、有利区間に関するコマンドである演出コマンドを送信バッファにセットする演出コマンド設定処理を実行する。

【0960】

50

(ステップ S 2 9 0 0 - 1 7)

メイン CPU 5 0 0 a は、1 遊技が終了したことを示す遊技終了コマンドを送信バッファにセットする。

【 0 9 6 1】

(ステップ S 2 9 0 0 - 1 9)

メイン CPU 5 0 0 a は、外部信号を出力するための端子板信号出力処理を実行する。

【 0 9 6 2】

(ステップ S 2 9 0 0 - 2 1)

メイン CPU 5 0 0 a は、上記ステップ S 2 9 0 0 - 7 において有利区間を終了させるときに設定される演出用ウェイトタイマが 0 でないかを判定する。その結果、演出用ウェイトタイマが 0 でないと判定した場合にはステップ S 2 9 0 0 - 2 1 に処理を移し、演出用ウェイトタイマが 0 であると判定した場合にはステップ S 2 9 0 0 - 2 3 に処理を移す。

【 0 9 6 3】

(ステップ S 2 9 0 0 - 2 3)

メイン CPU 5 0 0 a は、遊技状態を示す遊技状態コマンドを送信バッファにセットする。

【 0 9 6 4】

(ステップ S 2 9 0 0 - 2 5)

メイン CPU 5 0 0 a は、次遊技の開始を示す遊技開始コマンドを送信バッファにセットし、ステップ S 2 1 0 0 に処理を移す。

【 0 9 6 5】

ステップ S 2 1 0 0 からステップ S 2 9 0 0 までの一連の処理を通じて 1 遊技が実行される。以後は、ステップ S 2 1 0 0 からステップ S 2 9 0 0 までを繰り返すこととなる。

【 0 9 6 6】

次に、主制御基板 5 0 0 における電源断時退避処理およびタイマ割込み処理について説明する。

【 0 9 6 7】

(主制御基板 5 0 0 の電源断時退避処理)

図 9 7 は、主制御基板 5 0 0 における電源断時退避処理を説明するフローチャートである。メイン CPU 5 0 0 a は、電源断検知回路を監視しており、電源電圧が所定値以下になると、割り込んで電源断時退避処理を実行する。

【 0 9 6 8】

(ステップ S 3 0 0 0 - 1)

電源断予告信号が入力されると、メイン CPU 5 0 0 a は、レジスタを退避する。

【 0 9 6 9】

(ステップ S 3 0 0 0 - 3)

メイン CPU 5 0 0 a は、電源断予告信号をチェックする。

【 0 9 7 0】

(ステップ S 3 0 0 0 - 5)

メイン CPU 5 0 0 a は、電源断予告信号を検出しているかを判定する。その結果、電源断予告信号を検出していると判定した場合にはステップ S 3 0 0 0 - 1 1 に処理を移し、電源断予告信号を検出していないと判定した場合にはステップ S 3 0 0 0 - 7 に処理を移す。

【 0 9 7 1】

(ステップ S 3 0 0 0 - 7)

メイン CPU 5 0 0 a は、レジスタを復帰させる。

【 0 9 7 2】

(ステップ S 3 0 0 0 - 9)

メイン CPU 5 0 0 a は、割込みを許可するための処理を行い、当該電源断時退避処理を終了する。

10

20

30

40

50

【0973】

(ステップS3000-11)

メインCPU500aは、出力ポートの出力を停止する出力ポートクリア処理を実行する。

【0974】

(ステップS3000-13)

メインCPU500aは、別領域についての電源断時の退避処理を実行する。

【0975】

(ステップS3000-15)

メインCPU500aは、メインRAM300cへのアクセスを禁止するために必要なRAMプロテクト設定処理を実行する。

10

【0976】

(ステップS3000-17)

メインCPU500aは、電源断発生監視時間を設定すべく、ループカウンタのカウンタ値に所定の電源断検出信号検出回数をセットする。

【0977】

(ステップS3000-19)

メインCPU500aは、上記ステップS3000-17でセットしたループカウンタの値を1減算する。

20

【0978】

(ステップS3000-21)

メインCPU500aは、ループカウンタのカウンタ値が0でないかを判定する。その結果、カウンタ値が0ではないと判定した場合にはステップS3000-19に処理を移し、カウンタ値が0であると判定した場合には上記したCPU初期化処理（ステップS1000）に移行する。

【0979】

なお、実際に電源断が生じた場合には、ステップS3000-19～ステップS3000-21をループしている間にスロットマシン400の稼働が停止する。

【0980】

(主制御基板500のタイマ割込み処理)

30

図98は、主制御基板500におけるタイマ割込み処理を説明するフローチャートである。主制御基板500には、所定の周期（同時回し参考例では1.49ミリ秒、以下「1.49ms」という）毎にクロックパルスが発生させるリセット用クロックパルス発生回路が設けられている。そして、リセット用クロックパルス発生回路によって、クロックパルスが発生すると、割り込んで、以下のタイマ割込み処理が実行される。

【0981】

(ステップS3100-1)

メインCPU500aは、レジスタを退避する。

【0982】

(ステップS3100-3)

メインCPU500aは、割込みフラグをクリアする。

40

【0983】

(ステップS3100-5)

メインCPU500aは、各種の入力ポートイメージを読み込み、最新のスイッチ状態を正確に取得するためのポート入力処理を実行する。

【0984】

(ステップS3100-7)

メインCPU500aは、セットされた出カイメージを出力ポートに出力し、メインクレジット表示部430、メイン払出表示部432、投入枚数表示器、スタート表示器、ストップスイッチ420a、420b、420cの表示器、リプレイ表示器、区間表示器1

50

60を点灯制御するダイナミックポート出力処理を実行する。

【0985】

(ステップS3100-9)

メインCPU500aは、タイマ割込み用フェーズを更新する。なお、タイマ割込み用フェーズは、0～3のいずれかであり、ここでは、タイマ割込み用フェーズが0、1、2の場合には1加算され、タイマ割込み用フェーズが3の場合には0に変更される。

【0986】

(ステップS3100-11)

メインCPU500aは、送信バッファに格納されたコマンドを副制御基板502に送信するためのサブコマンド送信処理を行う。

【0987】

(ステップS3100-13)

メインCPU500aは、ステッピングモータ452を制御するステッピングモータ制御処理を実行する。

【0988】

(ステップS3100-15)

メインCPU500aは、メダル払出装置442へ出力する出力イメージを出力する出力ポートイメージ出力処理を実行する。

【0989】

(ステップS3100-17)

メインCPU500aは、各種乱数を更新する乱数更新処理を実行する。

【0990】

(ステップS3100-19)

メインCPU500aは、エラーに対応する外部信号（外部信号4、5）を外部に出力するためにエラーを検出する不正監視処理を実行する。

【0991】

(ステップS3100-21)

メインCPU500aは、上記ステップS3100-9で更新したタイマ割込み処理用フェーズに対応するモジュール（サブルーチン）を実行する。ここで、タイマ割込み処理用フェーズは0～3のいずれかに設定されており、タイマ割込み処理用フェーズ0～3それぞれに対応するモジュールが1つずつ設けられているため（合計4つ）、1つのモジュールは、タイマ割込み処理の4回に1回（5.96ms毎に）実行されることになる。例えば、各種タイマを減算する時間監視処理を実行するモジュールが1つのタイマ割込み処理用フェーズに対応付けられている。

【0992】

(ステップS3100-23)

メインCPU500aは、試験信号を外部に出力する試験信号出力処理を実行する。

【0993】

(ステップS3100-25)

メインCPU500aは、各種の入力ポートイメージを読み込み、最新のスイッチ状態を正確に取得するためのポート入力処理を実行する。

【0994】

(ステップS3100-27)

メインCPU500aは、レジスタを復帰する。

【0995】

(ステップS3100-29)

メインCPU300aは、割込みを許可し、当該タイマ割込み処理を終了する。

【0996】

また、上述した実施形態では、主制御基板500と副制御基板502とが、遊技を進行するための機能部を分担するように配したが、主制御基板500の機能部を副制御基板5

10

20

30

40

50

02に配しても、副制御基板502の機能部を主制御基板500に配してもよく、また、全ての機能部を1の制御基板に纏めて配することもできる。

【0997】

また、上述した実施形態では、AT演出状態が1種類のみ設けられるようにしたが、例えば、AT演出状態の継続遊技数の上乘せ特化ゾーンなど、複数種類のAT演出状態が設けられるようにしてもよい。

【0998】

また、上記した実施形態では、遊技価値としてのメダルを用いて遊技を行うようにしたが、遊技価値は電氣的な情報であってもよい（所謂メダルレスであってもよい）。この場合、当選役が入賞したときに、当選役に対応する価値量を遊技者に電氣的な情報で付与すればよい。

【0999】

また、上述した主制御基板500および副制御基板502が行う各処理は、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいはサブルーチンによる処理を含んでもよい。

【1000】

また、上述した主制御基板500および副制御基板502が行う各処理は、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいはサブルーチンによる処理を含んでもよい。

【1001】

〈主制御基板のCPU周辺の構成〉

図99は、メインCPU300a周辺の電氣的な接続を説明するための図である。メインCPU300aは、CPUコア700とバスコントローラ702とを含む。CPUコア700は、BC端子から出力されるバス制御信号（Bus Cont）を通じてバスコントローラ702を制御し、メインROM300b、メインRAM300c、もしくは、入出力部704からデータを読み出し、または、メインRAM300cにデータを書き込む。なお、ここでは、CPU300aとして、Z80系CPUをベースとするエルイーテック（LEEtech）社が販売するマイクロプロセッサを用いている。なお、ここでは、パチンコ機のメインCPU300a、メイン、ROM300b、メインRAM300cを挙げて説明するが、スロットマシン400のメインCPU500a、メイン、ROM500b、メインRAM500cに置き換え可能なのは言うまでもない。

【1002】

例えば、メインROM300b、メインRAM300c、または、入出力部704からデータを読み出す場合、バスコントローラ702は、16ビットのアドレス（A[16]）信号を出力し、デコーダ706a、706b、706cを通じてメインROM300b、メインRAM300c、または、入出力部704のいずれかを特定するとともに、リード（RD）信号を制御して、メインROM300b、メインRAM300c、または、入出力部704からデータ（D[8]）信号を読み出す。また、メインRAM300c、または、入出力部704にデータを書き込む場合、バスコントローラ702は、アドレス（A[16]）信号およびデータ（D[8]）信号を出力し、デコーダ706b、706cを通じてメインRAM300c、または、入出力部704のいずれかを特定するとともに、ライト（WR）信号を制御して、メインRAM300c、または、入出力部704にデータ（D[8]）信号を書き込む。

【1003】

ここでは、後述するように、入出力部704のアドレス空間を、メインROM300bおよびメインRAM300cのアドレス空間と一体化している。したがって、従来、メモリとI/Oのいずれをアクセスするかを特定するための信号を出力するメモリリクエスト（MREQ）端子およびI/Oリクエスト（IORQ）端子を設けていない。かかる2端子を任意の他の信号に割り当て直すことで、プログラム開発の自由度を高めることができる。

10

20

30

40

50

【1004】

また、CPUコア700には、割込処理の開始トリガとなる割り込み／待ち（INT／WAIT）信号、最優先で割込処理を実行できるマスク不可割込（NMI）信号、バス信号をハイインピーダンスに遷移可能なバスリクエスト（BUSREQ）信号等の外部信号も入力される。

【1005】

図100は、CPUコア700の内部構成を示したブロック図である。CPUコア700は、外部入力ユニット710、状態制御ユニット712、中央制御ユニット714、レジスタユニット716、算術論理演算装置（ALU）718を含む。外部入力ユニット710は、外部信号を受信し、その外部信号に基づいた制御情報を状態制御ユニット712および中央制御ユニット714に出力する。

10

【1006】

状態制御ユニット712は、内部状態（RESET、命令フェッチ、命令デコード、演算、メモリロード、メモリストア、HALT等）を管理および遷移させてCPUコア700の動作状態を決定するとともに、その内部状態に基づいた制御情報を中央制御ユニット714に出力する。

【1007】

中央制御ユニット714は、バスコントローラ702を経由して入力された入力データ（DI[8]）からオペコード（命令）を抽出し、命令デコーダによってデコードしたコマンドに基づいてALU718を制御する。また、中央制御ユニット714は、デコードしたコマンドによりレジスタユニット716の各レジスタから必要な情報を取得したり、各レジスタを更新したりする。

20

【1008】

レジスタユニット716は、セレクトポート722a、722b、722c、入力側バンクセクタ724、第1レジスタバンク726、第2レジスタバンク728、出力側バンクセクタ730、アドレスポート732、個別レジスタ734を含む。なお、個別レジスタ734には、次に実行すべきプログラムのアドレスを示す16ビットのプログラムカウンタ（PC）、割込モード時に使用される8ビットのインタラプト（I）レジスタ、オペコードフェッチサイクルを計数する8ビットのリフレッシュ（R）レジスタ、割込の許可／禁止を制御する8ビットの割込許可（IFF）レジスタが含まれる。

30

【1009】

また、レジスタユニット716には、大役抽選に係る種々の乱数値（大当たり決定乱数、当たり図柄乱数、リーチグループ決定乱数、リーチモード決定乱数、変動パターン乱数、当たり決定乱数）を取得するための乱数発生器（図示せず）が対応付け、入力ポート（FE73h～FE9Ch）を介してラッチされた乱数値が取得される。

【1010】

乱数発生器は、システムクロック（外部入力を2分周したクロック）で動作し、所定の最大値未満の乱数を発生する。なお、乱数発生器は、乱数の最大値を設定可能な乱数発生器である最大値設定乱数発生器として、16ビットの最大値を設定可能な乱数発生器が4チャンネル、8ビットの最大値を設定可能な乱数発生器が8チャンネル準備されている。ここで、16ビットの最大値設定乱数発生器は、乱数更新周期が32～47クロックの範囲で選択でき、最大値の設定範囲が256～65535の範囲で設定できる。また、8ビットの最大値設定乱数発生器は、乱数更新周期が16～31クロックの範囲で選択でき、最大値の設定範囲が4チャンネルで16～255の範囲で設定でき、他の4チャンネルで64～255の範囲で設定できる。また、乱数の最大値が固定された乱数発生器である最大値固定乱数発生器として、16ビットの最大値を設定可能な乱数発生器が4チャンネル、8ビットの最大値を設定可能な乱数発生器が8チャンネル準備されている。ここで、16ビットの最大値固定乱数発生器は、乱数更新周期が1クロックに、最大値が65535に固定されている。また、8ビットの最大値固定乱数発生器は、乱数更新周期が1クロックに、最大値が255に固定されている。

40

50

【1011】

なお、乱数の種類が足りない場合、ハードウェア乱数生成部（乱数発生器）から取得した乱数値に、プログラム内において所定の数値を乗じ、また、除算することで他の乱数を生成する（ソフトウェア乱数生成部）ことも可能である。

【1012】

図101は、レジスタの構成を説明した図である。第1レジスタバンク（バンク0）726および第2レジスタバンク（バンク1）728は、いずれも、8ビットのレジスタ（Q、U、A、F、B、C、D、E、H、L）と、16ビットのレジスタ（IX、IY、SP）が設けられる。また、第1レジスタバンク726および第2レジスタバンク728のレジスタには、メインレジスタ（表レジスタ）グループ726a、728aと、サブレジスタ（裏レジスタ）グループ726b、728bとがある。CPU300a、500aは、Fレジスタにおけるレジスタバンク指定レジスタRBの示すレジスタバンクのみにアクセスでき、対になるレジスタグループにはアクセスできない。

【1013】

図101に示すレジスタのうち、Qレジスタは、遊技機用拡張仕様の8ビット専用レジスタである。かかるQレジスタは、例えば、F0h固定で、F000h～F0FFhのメインRAM300cのアクセスに利用する。Uレジスタは、遊技機用拡張仕様の8ビット専用レジスタである。かかるUレジスタは、例えば、FEh固定で、FE00h～FEFFhの入出力部704に接続された内蔵デバイス（タイマ、乱数発生器、外部入出力回路等へのアクセスに利用する。Aレジスタは、演算処理やデータ転送に使う8ビットのアクキュムレータである。Fレジスタは、各種演算結果を保持する8ビットのフラグレジスタである。ここで、Fレジスタの各ビットは、図101に示すように、最上位ビット（MSB：Most Significant Bit）から最下位ビット（LSB：Least Significant Bit）にかけて、Sは、演算結果が負のとき1にセットされるサインフラグであり、Zは、演算の結果、全ビットが0のとき1にセットされるゼロフラグ（第1ゼロフラグ）であり、TZは、データ転送命令（LD；ロード）の実行により、全ビットが0のとき1にセットされる（値の変わる）遊技機用拡張仕様の特定ビットフラグ（第2ゼロフラグ）であり、ティーゼットフラグと呼ぶ場合もある。Hは、プログラマーが関与できないハーフキャリーフラグであり、RBは、現在のレジスタバンク（第1レジスタバンク726＝0、第2レジスタバンク728＝1）を示すレジスタバンクモニタであり、P/Vは、パリティオーバフローフラグであり、Nは、プログラマーが関与できない加減算フラグであり、Cは、演算の結果、桁上げまたはボロー発生時に1がセットされるキャリーフラグである。なお、Fレジスタは、AレジスタとペアレジスタAFを構成する。

【1014】

また、Bレジスタ、Cレジスタ、Dレジスタ、Eレジスタ、Hレジスタ、Lレジスタは、8ビットの汎用レジスタであり、それぞれ予め組合せが定められている16ビットのペアレジスタBC、DE、HLを構成する。IXレジスタ、IYレジスタは、インデックスアドレッシング用16ビット専用レジスタである。SP（スタックポインタ）レジスタは、16ビットであり、スタックポインタとなるアドレスを格納する。Q'レジスタ、A'レジスタ、F'レジスタ、B'レジスタ、C'レジスタ、D'レジスタ、E'レジスタ、H'レジスタ、L'レジスタ、IX'レジスタ、IY'レジスタは、Qレジスタ、Aレジスタ、Fレジスタ、Bレジスタ、Cレジスタ、Dレジスタ、Eレジスタ、Hレジスタ、Lレジスタ、IXレジスタ、IYレジスタのメインレジスタグループ726a、728aとの交換命令によりデータ（内容）交換可能なサブレジスタグループ726b、728bであり、A'レジスタとF'レジスタでペアレジスタAF'を構成し、B'レジスタとC'レジスタでペアレジスタBC'を構成し、D'レジスタとE'レジスタでペアレジスタDE'を構成し、H'レジスタとL'レジスタでペアレジスタHL'を構成する。サブレジスタグループ726b、728bは、メインレジスタグループ726a、728aと、入れ替え命令などによって互いにどちらかのレジスタを選択して入れ替えて使用することができる。一方、レジスタU、レジスタSPについては、サブレジスタグループ726b、728bを持たない単一のレジス

タである。

【1015】

ところで、上述したように、主制御基板300、500においては、メインCPU300aが、メインROM300bに格納されたプログラムに基づきメインRAM300cと協働して遊技の進行を制御する。これらの機能部を実行するためのプログラムは、メインROM300bおよびメインRAM300cの所定の領域（使用領域）に配される。

【1016】

図102は、メモリマップを示す説明図である。なお、パチンコ機におけるメモリマップについては、図5を用いて既に説明されているので、ここでは、スロットマシン400のメモリマップを挙げて説明する。メインROM500bには、0000h～3FFFh（12kbyte）のメモリ空間が割り当てられ、メインRAM500cには、F000h～F3FFFh（1kbyte）のメモリ空間が割り当てられ、入出力部704には、FE00h～FEFFFh（256byte）のメモリ空間が割り当てられている。なお、プログラムの命令コードはアセンブラ言語で記述されている。ここで、プログラムは、命令コードで構成されたものであり、コンピュータに読み出され、データやワークエリアと協働して所定の処理を実現することができる。

【1017】

メインROM500bの0000h～1DF3hのメモリ空間には使用領域が割り当てられている。使用領域は、遊技の進行を制御する遊技制御処理を実行するためのプログラムやデータを格納する領域である。具体的に、0000h～11FFFh（4.5kbyte）に制限されたメモリ空間（制御領域）に、初期化手段600、ベット手段602、当選種別抽選手段604、リール制御手段606、判定手段608、払出制御手段610、遊技状態制御手段612、演出状態制御手段614、コマンド送信手段616を機能させて遊技の進行を制御する遊技制御処理を実行するためのプログラムの命令コードが格納され、1200h～1DF3h（3.0kbyte）に制限されたメモリ空間（データ領域）に、遊技制御処理のプログラムに用いられるデータが格納されている。また、1E00h～1FFFFhのメモリ空間にはコメント領域が割り当てられ、3FC0h～3FFFFhのメモリ空間にはプログラム管理領域が割り当てられている。また、2000h～3FBFFhのメモリ空間には別領域（使用外領域）が割り当てられている。別領域は、後述するように、使用領域に格納することが定められていないプログラムやデータを格納する領域である。具体的に、2000h～3FBFFhのメモリ空間には、遊技の進行に影響を及ぼさない、遊技機用試験処理やセキュリティ関連処理のうち一部または全部の処理（以降では、単に遊技制御外処理という場合がある）を遂行するプログラムの命令コードおよびプログラムデータが格納されている。ここで、遊技機用試験処理は、回胴式遊技機用試験機の接続仕様書（第四版）に記載されているスロットマシン400の試験処理である。セキュリティ関連処理は、第三者の不正防止や不具合発見を目的とした異常状態を特定する処理であり、例えば、上述したバックアップフラグの判定やチェックサムの実行も含まれる。なお、別領域に記憶容量の制限はなく、図102の例では、使用領域、コメント領域、および、プログラム管理領域以外の記憶領域に、自由に割り当てることができる。

【1018】

上記のように、メインCPU500aは、遊技制御処理のみならず、遊技機用試験処理、セキュリティ関連処理（遊技制御外処理）も遂行する場合がある。しかし、使用領域の記憶容量は予め定められており、例えば、図102に示したように、制御領域が4.5kbyteに制限され、データ領域が3.0kbyteに制限されている。したがって、遊技制御処理のみならず、遊技機用試験処理、セキュリティ関連処理のプログラムやデータまでも使用領域に配すると、その分、遊技制御処理を行うための記憶領域が制限されてしまう。ここで、遊技制御処理を実行するためのプログラム（使用プログラム）やデータは、必ず使用領域に格納しなければならないが、遊技制御処理以外の遊技の進行に影響を及ぼさない遊技制御外処理（遊技機用試験処理、セキュリティ関連処理等）を実行するためのプログラム（別プログラム）やデータは使用領域および別領域のいずれにも格納するこ

10

20

30

40

50

とができる。そこで、セキュリティ関連処理に該当する処理であるバックアップフラグの判定処理やチェックサムの実行処理を実行するためのプログラムやデータの少なくとも一部を、使用領域とは異なる（使用領域以外の）記憶領域のうちの一部である別領域に記述している。

【1019】

このように使用領域で遂行される処理（ここでは、遊技制御処理）と、必ずしも使用領域で行わなくてよい処理（ここでは、セキュリティ関連処理）とが混在している場合には、遊技制御処理を実行するためのプログラム（使用プログラム）やデータを使用領域に格納し、使用領域で行わなくてよい、遊技制御処理以外の遊技の進行に影響を及ぼさない遊技制御外処理（セキュリティ関連処理）を実行するためのプログラム（別プログラム）やデータを別領域に格納することが望ましい。このように記憶領域を複数に区分することで、別領域に移動させたプログラムの分だけ使用領域の記憶領域（容量）に余裕が生じる。したがって、その分、使用領域を遊技制御処理（使用プログラム）に割り当てることが可能となる。

【1020】

ただし、上記のように記憶領域を、使用領域、別領域とで役割分担した場合においても、遊技機の公正さは担保されなくてはならない。そこで、遊技機の公正さを担保しつつ、使用領域と別領域とで適切に役割分担するために、以下の（１）～（６）の条件を規定する。

【1021】

条件（１）、別領域に配置するプログラムについては、遊技機の試験に必要な信号の出力（遊技機用試験処理）および不正防止（セキュリティ関連処理）を目的として使用され、遊技の公正を害さない（損なわない）ものであること。条件（２）、使用領域と別領域の制御領域およびデータ領域については、それぞれを明示的に区別された領域に配置すること。条件（３）、別領域に配置するプログラム（別プログラム）は、使用領域のプログラム（使用プログラム）から静的に呼び出された上で実行されること。また、その際のプログラムリストにおいては、呼び出し先アドレスが明らかに記載されていること。条件（４）、別領域に配置するプログラムは機能ごとにモジュール化し、呼び出された際には、使用領域で利用している全レジスタの内容を保護すること。条件（５）、使用領域または別領域から互いの領域にあるＲＡＭへのアクセスは参照のみ可能とし、更新は不可とすること。条件（６）、別領域の制御領域から使用領域の制御領域にあるサブルーチンを呼び出すことは不可とすること。なお、使用領域に割込処理を行うサブルーチンを設けることから、別領域の制御領域を使用する際には、割込禁止にする必要が生じる。なお、遊技制御処理を適切に遂行するために、割込禁止を行ってから割込禁止を解除するまでの時間は、遊技制御処理における割込処理の間隔（例えば１．４９ｍｓｅｃ）以内とならなければならない。したがって、別領域の制御領域を使用するサブルーチンを呼び出す場合、その１回の呼び出しにかかる総時間は、遊技制御処理の割込処理の間隔以内となるように設定することとなる。

【1022】

また、メインＲＡＭ５００ｃのＦ０００ｈ～Ｆ１ＦＦｈのメモリ空間には使用領域が割り当てられている。具体的に、Ｆ０００ｈ～Ｆ１３Ｆｈのメモリ空間には、上記遊技制御処理のワークエリアが割り当てられ、タイマ、カウンタ、フラグ等の変数管理に用いられる。Ｆ１Ｃ０ｈ～Ｆ１ＦＦｈのメモリ空間には、上記遊技制御処理のスタック領域が割り当てられている。また、メインＲＡＭ５００ｃのＦ２００ｈ～Ｆ３ＦＦｈのメモリ空間には別領域が割り当てられている。具体的に、Ｆ２１０ｈ～Ｆ２２Ｆｈのメモリ空間には、上記セキュリティ関連処理のうち一部または全部の処理のワークエリアが割り当てられ、タイマ、カウンタ、フラグ等の変数管理に用いられる。Ｆ２３０ｈ～Ｆ２４６ｈのメモリ空間には、上記セキュリティ関連処理のうち一部または全部の処理のスタック領域が割り当てられている。

【1023】

また、F E 0 0 h ~ F E F F h のメモリ空間には入出力部 7 0 4 が割り当てられている。従来、入出力部 7 0 4 に対応するデバイスにアクセスするため、メモリ空間と独立して 2 5 6 バイトの I / O 空間が設けられていた。これに対し、本実施形態では、M R E Q、I O R Q の信号をなくし、メモリ、入出力部 7 0 4 へのアクセスを共通して R D、W R 信号で行うこととした。また、入出力部 7 0 4 に接続されたデバイスにアクセスするための上位 8 ビットのアドレスを指定するハードウェアとしての U レジスタを設け、ここに予め 8 ビットの上位アドレスを指定しておく。これにより、メモリ空間とは独立して設けられていた I / O 空間を、メモリ空間に統合して一つのアドレス空間とし、I N 命令、O U T 命令を実行するとメモリ空間に割り当てられた入出力部 7 0 4 に対し、上位 8 ビットを U レジスタで指定し下位 8 ビットは I N 命令、O U T 命令のオペランドで指定した下位 8 ビットを用いてアクセス可能とした。

10

【1024】

本実施形態では、L D Q 命令では Q レジスタの値を用いてメモリ空間（主にデータエリア、ワークエリア）をアクセスし、I N 命令、O U T 命令では U レジスタを用いてデバイス（タイマ、乱数発生器、外部入出力回路等）の I / O をアクセスするようにプログラムを記述できるようになる。かかる構成により、設計時にプログラムを把握し易くなる。また、メモリおよび I / O を、1 6 ビットのアドレスで特定してアクセスしていたものを下位 8 ビットのオペランドでアクセスすることが可能になり、プログラム容量を圧縮することができる。さらに Q レジスタ、Q' レジスタ、U レジスタと複数の上位指定レジスタを持つことにより、上位レジスタが 1 つだけの時よりも上位レジスタの使い回しによる入れ替えの回数が少なくなり、プログラム容量をさらに圧縮することができる。

20

【1025】

上記の例では I N 命令、O U T 命令で I / O 空間に対応するメモリ空間にアクセスしたが、I N 命令、O U T 命令で直接メモリ空間にアクセスしてもよい。このことは、例えばメモリ上の 3 つの 2 5 6 バイト領域をアクセスする場合に Q レジスタ、Q' レジスタ、U レジスタにそれぞれの上位 8 ビットを指定しておき、L D Q 命令と I N 命令 O U T 命令でそれぞれの領域をアクセスすることで実現できる。

【1026】

（遊技情報の表示）

ところで、遊技機、例えば、スロットマシン 4 0 0 では、上述したようなボーナス遊技状態や A T 演出状態への移行頻度が偏ると射倖性が過度に高まってしまう懸念がある。そこで、メダルの獲得性能が高い遊技状態が偏っているか否かを統括的かつ画一的に判定すべく、有利区間比率、指示込役物比率、連続役物比率、役物比率、役物等状態比率といった判断基準を設ける。有利区間比率は、初回起動時から、有利区間に滞在している遊技数である滞在遊技数を、遊技数の総数である総遊技数で除算し、% で示した値を言い、指示込役物比率は、初回起動時から、所定の基準集計期間における、ボーナス遊技等、役物の作動および指示機能作動に基づいて払出制御手段 6 1 0 が払い出したメダルの枚数である指示込役物払出数を、払出制御手段 6 1 0 が払い出したメダルの総数である総払出数で除算し、% で示した値を言い、連続役物比率は、初回起動時から、所定の基準集計期間における、レギュラーボーナス遊技等、第 1 種特別役物の作動によって払出制御手段 6 1 0 が払い出したメダルの枚数である連続役物払出数を、払出制御手段 6 1 0 が払い出したメダルの総数である総払出数で除算し、% で示した値を言い、役物比率は、初回起動時から、所定の基準集計期間における、ボーナス遊技等、役物の作動に基づいて払出制御手段 6 1 0 が払い出したメダルの枚数である役物払出数を、払出制御手段 6 1 0 が払い出したメダルの総数である総払出数で除算し、% で示した値を言い、役物等状態比率は、初回起動時から、所定の基準集計期間における、ボーナス遊技等、役物が作動した遊技数である作動遊技数を、遊技数の総数である総遊技数で除算し、% で示した値を言う。なお、有利区間比率、指示込役物比率、連続役物比率、役物比率、役物等状態比率は、設定変更によってはクリアされない。

30

40

【1027】

50

このような、有利区間比率、指示込役物比率、連続役物比率、役物比率、役物等状態比率の表示は、図 9 8 に示したタイマ割込み処理 S 3 1 0 0 のダイナミックポート出力処理 (SYNMOU T モジュール) S 3 1 0 0 - 7 における使用領域該役比表示取得処理 (E _ S D P S T モジュール) において実行される。

【1 0 2 8】

ここで、有利区間は、非 A T 演出状態のうちの一部の演出状態（ここでは非有利演出状態）を除く全ての演出状態に滞在している区間を言う。なお、有利区間に対し、非有利演出状態を非有利区間という場合もある。ただし、有利区間として、A T 演出状態や非 A T 演出状態に限らず、遊技者に対し有利に遊技が進行する遊技状態および演出状態のいずれかを任意に設定してもよい。

【1 0 2 9】

また、役物は、入賞を容易にするための装置であり、レギュラーボーナス (R B)、チャレンジボーナス (C B)、シングルボーナス (S B) 等が該当する。また、かかる役物のうち、レギュラーボーナス (R B) が第 1 種特別役物に対応する。なお、レギュラービッグボーナス (R B B) は、第 1 種特別役物であるレギュラーボーナス (R B) の入賞形態を示す図柄組合せの表示によらずにレギュラーボーナス (R B) が連続で作動するものであり、チャレンジビッグボーナス (C B B) は、第 2 種特別役物（役物のうちの一つ）であるチャレンジボーナス (C B) の入賞形態を示す図柄組合せの表示によらずにチャレンジボーナス (C B) が連続で作動するものである。したがって、連続役物比率は、所定の基準集計期間において払い出された総払出数に対する第 1 種特別役物の作動によって払い出された連続役物払出数の比率を表すこととなり、役物比率は、所定の基準集計期間において払い出された総払出数に対する役物の作動によって払い出された役物払出数の比率を表すこととなる。

【1 0 3 0】

なお、基準集計期間は、払出制御手段 6 1 0 が払い出したメダルの枚数を累積する対象期間である。本実施形態では、例えば、有限の値である 6 0 0 0 遊技や、当該スロットマシン 4 0 0 を遊技場（ホール）に設置してから全ての期間（以下、「総累計」という）の 2 通りを挙げる。したがって、連続役物比率としては、6 0 0 0 遊技の連続役物比率と、総累計の連続役物比率とを導出し、役物比率としては、6 0 0 0 遊技の役物比率と、総累計の役物比率とを導出する。ただし、基準集計期間として、任意の遊技数を設定できるのは言うまでもない。

【1 0 3 1】

このような、有利区間比率、指示込役物比率、連続役物比率、役物比率、役物等状態比率を求めるべく、累積手段（図示せず）は、総遊技数、滞在遊技数、作動遊技数、総払出数、指示込役物払出数、連続役物払出数、役物払出数を累積する。このとき、総遊技数、滞在遊技数、作動遊技数、総払出数、指示込役物払出数、連続役物払出数、役物払出数それぞれには、カウンタが対応付けられている。具体的に、累積手段は、1 遊技が開始されると、総遊技数を示す総遊技数カウンタを 1 だけインクリメントする。そして、現在の遊技が有利区間における遊技であるか否か判定し、有利区間における遊技であれば、滞在遊技数を示す滞在遊技数カウンタを 1 だけインクリメントする。なお、滞在遊技数カウンタの更新は、非有利区間である場合において有利区間に当選した遊技（有利区間への移行処理が行われた遊技）の次遊技から開始される。ただし、レギュラービッグボーナス (R B B) またはチャレンジビッグボーナス (C B B) の当選に基づいて有利区間に当選した場合には当該ボーナスが入賞した遊技（有利区間への移行処理が行われた遊技）の次遊技から開始される。また、現在の遊技が、役物が作動した遊技であるか否か判定し、役物が作動した遊技であれば、作動遊技数を示す作動遊技数カウンタを 1 だけインクリメントする。

【1 0 3 2】

また、累積手段は、その遊技でメダルが払い出されると、総払出数を示す総払出数カウンタを、その払出枚数分だけインクリメントする。そして、当該メダルの払い出しが役物の作動および指示機能作動に基づいていれば、指示込役物払出数を示す指示込役物払出数

10

20

30

40

50

カウンタを、その払出枚数分だけインクリメントし、当該メダルの払い出しが役物に基づいていれば、役物払出数を示す役物払出数カウンタを、その払出枚数分だけインクリメントし、さらに、当該メダルの払い出しが第1種特別役物に基づいていれば、連続役物払出数を示す連続役物払出数カウンタを、その払出枚数分だけインクリメントする。

【1033】

なお、総累計については、かかるカウンタのみで計数できるが、連続役物比率および役物比率に関しては、総累計のみならず、基準集計期間、ここでは、6000遊技といった限られた期間も対象にしなければならない。そこで、6000遊技を、例えば、400遊技毎に均等に区分けし、400遊技ずつ計数する複数（例えば15個）のカウンタを設け、リングバッファとして利用する。なお、ここでは、6000遊技を400遊技毎に区分けして計数する例を挙げているが、6000の因数（約数）であれば、区分けする遊技数を任意に設定可能なのは言うまでもない。

【1034】

具体的に、累積手段は、その遊技でメダルが払い出されると、総払出数カウンタ、連続役物払出数カウンタ、役物払出数カウンタを更新する処理を400遊技まで行い、400遊技に到達すると、過去に計数した14個のカウンタと合算して、総払出数、連続役物払出数、役物払出数を導出する。そして、14個のカウンタのうち最も古い遊技で計数したカウンタを初期化し、その初期化したカウンタによって、新たに400遊技分、総払出数カウンタ、連続役物払出数カウンタ、役物払出数カウンタを更新する。こうして、400遊技毎に、新たな400遊技分の遊技情報と、過去の5600遊技分の遊技情報とを参照することができるので、直近の6000遊技の総払出数、連続役物払出数、役物払出数を導出することが可能となる。なお、かかるカウンタによる計数は、規定数や設定値に拘わらず常に実行される。このようにして、総遊技数、滞在遊技数、作動遊技数、総払出数、指示込役物払出数、連続役物払出数、役物払出数がそれぞれ累積される。

【1035】

続いて、比率導出手段（図示せず）は、累積手段が累積した、総遊技数、滞在遊技数、作動遊技数、総払出数、指示込役物払出数、連続役物払出数、役物払出数に基づいて有利区間比率、指示込役物比率、連続役物比率、役物比率、役物等状態比率を導出する。具体的に、比率導出手段は、滞在遊技数を総遊技数で除算し、%換算して有利区間比率を導出し、総累計の指示込役物払出数を総累計の総払出数で除算し、%換算して総累計の指示込役物比率を導出し、総累計の連続役物払出数を総累計の総払出数で除算し、%換算して総累計の連続役物比率を導出し、総累計の役物払出数を総累計の総払出数で除算し、%換算して総累計の役物比率を導出し、作動遊技数を総遊技数で除算し、%換算して役物等状態比率を導出する。また、比率導出手段は、400遊技毎に、6000遊技の連続役物払出数を6000遊技の総払出数で除算し、%換算して6000遊技の連続役物比率を導出し、6000遊技の役物払出数を6000遊技の総払出数で除算し、%換算して6000遊技の役物比率を導出する。

【1036】

次に、比率表示手段（図示せず）は、所定の表示指令に基づいて、比率導出手段が導出した有利区間比率、指示込役物比率、6000遊技の連続役物比率、6000遊技の役物比率、総累計の連続役物比率、総累計の役物比率、役物等状態比率を比率表示部860に表示する。

【1037】

図103は、主制御基板500の外観を示した説明図である。図103(a)に示すように、主制御基板500の端側には複数のコネクタ870が設けられ、また、主制御基板500の中央近傍には比率表示部860が設けられている。比率表示部860は、十進のアラビア数字を表すことができる7つのセグメントと、その右下に位置するドットセグメント(DP)とからなる7セグメントを4つ並置して構成される。そして、主制御基板500をスロットマシン400に組み込む際には、図103(b)のように、主制御基板500を基板ケース872に収容し、基板ケース872の前面部位872aに主制御基板5

00の背面からビス874で固定する。基板ケース872の前面部位872aにおける、比率表示部860に対応する位置には、比率表示部860を視認可能な透過窓872bが設けられている。したがって、管理者は、図74に示したように、前面上扉404を開放した状態で、当該透過窓872bを通じ、比率表示部860の表示内容を視認することができる。なお、かかる透過窓872bの代わりに、基板ケース872全体を透過または半透過な材質で形成してもよい。

【1038】

ここでは、比率表示部860を構成する4つの7セグメントのみで、有利区間比率、6000遊技の連続役物比率、6000遊技の役物比率、総累計の連続役物比率、総累計の役物比率といった5つの遊技情報を表示するため、各比率に識別子に対応付け、比率表示部860には、その識別子と各比率とを順次表示することとする。なお、各比率は、0～99の範囲で示される。したがって、4つの7セグメントのうち、左側に位置する2つの7セグメントで識別子を示し、右側に位置する2つの7セグメントで比率を示す。

【1039】

図104は、比率表示部860の表示態様を説明するための説明図である。電源スイッチ444をONすると、図104(a)のように、比率表示手段は、有利区間比率に対応する識別子である「7U」と、その比率である、例えば「70」%とを一度に表示する（所謂7Uタイプ）。また、スロットマシン100によっては、有利区間比率の代わりに指示込役物比率を表示することもできる。この場合、比率表示手段は、指示込役物比率に対応する識別子である「7P」と、その比率である、例えば「70」%とを一度に表示する（所謂7Pタイプ）。ここで、7Uタイプでは、設定差のある当選種別を参照して有利区間に移行させる抽選を行うことは可能であるが、設定値を直接参照して有利区間に移行させる抽選を行うことはできない。一方、7Pタイプでは、設定差のある当選種別、および、設定値のいずれも参照して有利区間に移行させる抽選を行うことが可能である。なお、比率表示手段は、比率表示部860の2つめの7セグメントのドットセグメントを点灯することで識別子と比率とを区切っている。

【1040】

この状態で、所定時間（例えば5.0sec）が経過すると、図104(b)のように、比率表示手段は、6000遊技の連続役物比率に対応する識別子である「6Y」と、その比率である「60」%とを一度に表示する。そして、所定時間が経過する度に、図104(c)のような、6000遊技の役物比率に対応する識別子である「7Y」と、その比率である「70」%との組み合わせ、図104(d)のような、総累計の連続役物比率に対応する識別子である「6A」と、その比率である「60」%との組み合わせ、図104(e)のような、総累計の役物比率に対応する識別子である「7A」と、その比率である「70」%との組み合わせ、図104(f)のような、役物等状態比率に対応する識別子である「5H」と、その比率である「50」%との組み合わせが順次表示される。また、図104(f)のように、役物等状態比率に対応する識別子である「5H」と、その比率である「50」%との組み合わせが表示されている状態で、さらに所定時間が経過すると、図104(a)に示した、有利区間比率に対応する識別子である「7U」と、その比率である「70」%とが再度表示され、以後、所定時間が経過する度に、有利区間比率（指示込役物比率）、6000遊技の連続役物比率、6000遊技の役物比率、総累計の連続役物比率、総累計の役物比率、役物等状態比率の順で表示が切り換わる。

【1041】

なお、ここでは、4つの7セグメントで識別子と比率を同時に示す例を挙げて説明したが、2つの7セグメントのみを用い、識別子と比率とを切り換えて表示することもできる。この場合、比率表示手段は、比率表示部860において、所定時間が経過する度に、例えば、「7U」→「70」→「6Y」→「60」→「7Y」→「70」→「6A」→「60」→「7A」→「70」→「5H」→「50」といった順に表示を切り替える。

【1042】

なお、比率表示手段は、主制御基板500の比率表示部860に代えて、4つの7セグ

メントに対応する既存の表示部（メインクレジット表示部430およびメイン払出表示部432）に識別子と比率とを切り換えて表示することもできる。この場合、1遊技の結果が報知された後、次の遊技の開始までに、前面上扉404を開放し、その状態で、既存のスイッチ、例えば、設定値を設定または変更する操作を実行するスイッチを操作することで、有利区間比率に対応する識別子と、その比率が表示され、以後、所定時間が経過する度に、有利区間比率（指示込役物比率）→6000遊技の連続役物比率→6000遊技の役物比率→総累計の連続役物比率→総累計の役物比率→指示込役物比率の順で表示が繰り返し切り換わることとなる。なお、比率表示手段が既存の表示部に、有利区間比率（指示込役物比率）、6000遊技の連続役物比率、6000遊技の役物比率、総累計の連続役物比率、総累計の役物比率、役物等状態比率を表示している間であっても、メダルの投入等、スロットマシン400の状態に変化があった場合には、その表示を削除する。

10

【1043】

管理者は、このような比率表示部860または既存の表示部を通じて、有利区間比率、指示込役物比率、6000遊技の連続役物比率、6000遊技の役物比率、総累計の連続役物比率、総累計の役物比率、役物等状態比率といった遊技情報を視認することができる。そして、遊技情報が適切な範囲に収まっているか否かを判定し、スロットマシン400が適正に動作しているかを判断することが可能となる。

【1044】

かかる適切な範囲は、例えば、有利区間比率<70%、指示込役物比率<70%、6000遊技の連続役物比率<60%、6000遊技の役物比率<70%、総累計の連続役物比率<60%、総累計の役物比率<70%、役物等状態比率<50%と設定している。かかる適切な範囲を逸脱している場合、比率表示手段は、その対象となる比率に関し、比率表示部860の各セグメントを例えば0.6sec周期で点滅させるとしてもよい。かかる点滅表示により、管理者は、適切な範囲を逸脱している識別子およびその比率を迅速に把握することが可能となる。

20

【1045】

ところで、上述したように、比率導出手段は、滞在遊技数を総遊技数で除算して有利区間比率を導出し、総累計の指示込役物払出数を総累計の総払出数で除算して総累計の指示込役物払出数を導出し、総累計の連続役物払出数を総累計の総払出数で除算して総累計の連続役物払出数を導出し、総累計の役物払出数を総累計の総払出数で除算して総累計の役物払出数を導出し、作動遊技数を総遊技数で除算して役物等状態比率を導出している。しかし、主制御基板500におけるメインCPU500aは処理能力が低く、かかるメインCPU500aで利用可能な乗算命令や除算命令は、被乗数、乗数、被除数、および、除数となる数値の範囲が制限されている。例えば、メインCPU500aでの演算命令として予め準備されている、除算を実行するDIV命令には、被除数および除数のいずれも数値範囲が2バイトに制限される場合があり、つまり、0~65535までの範囲までしか取り扱うことができない場合がある。そうすると、総遊技数が、例えば175000遊技以上となる場合、数値範囲を超えてしまい、単純に2バイトのDIV命令を利用することができない。この場合、人が筆算で除算を行うのと同様、加算および減算を繰り返すプログラムを別途準備しなければならず、処理負荷の増大を招くこととなる。そこで、本実施形態では、処理負荷の増大を抑えつつ、遊技情報を適切に得ることを目的とする。

30

40

【1046】

なお、上記では、スロットマシン400において、例えば、有利区間比率、指示込役物比率、6000遊技の連続役物比率、6000遊技の役物比率、総累計の連続役物比率、総累計の役物比率、役物等状態比率といった遊技情報を比率表示部860に表示する例を挙げて説明した。パチンコ機においては、例えば、所定の基準集計期間における、賞球数（低確時かつ時短無し時の払出個数である低確払出個数）を、アウト数（低確時かつ時短無し時のアウト個数である低確アウト個数）で除算し、%で示したベース値を性能表示モニタ（比率表示部860）に表示するとしてもよい。なお、パチンコ機におけるベース値では、スロットマシン400のように、小数点以下を切り捨てしないで、四捨五入する。

50

【1047】

(各比率の導出処理)

ここでは、各比率（有利区間比率、指示込役物比率、連続役物比率、役物比率、役物等状態比率）を導出する処理例を複数挙げて説明するが、まず、その複数の処理に共通する部分を詳述する。本実施形態では、有利区間比率、指示込役物比率、連続役物比率、役物比率、役物等状態比率として、最終的に％に換算された値を求めることとなる。これは、除算した結果に対し、小数点以下2桁のみが必要であることを示す。また、スロットマシン400では、％に換算した場合の小数点以下の値は切り捨てとしている。これは、除算した結果に対し、小数点以下2桁のみを導出できれば、小数点以下3桁以降の演算を要さないことを意味する。したがって、本実施形態では、除算の結果を2桁繰上すべく、所定の乗数（ここでは100）を乗じる。なお、ここでは、小数点2桁のみが必要なので乗数として「100」を乗じる例を挙げたが、小数点3桁が必要な場合は乗数として「1000」を乗じる等、必要な桁数に応じて乗数を変更することができる。

10

【1048】

また、ここでは、導出処理の対象として「役物比率」を挙げて説明するが、有利区間比率、指示込役物比率、連続役物比率、役物等状態比率にも同様の処理が適用できるのは言うまでもない。役物比率は、上述したように、役物払出数を総払出数で除算し、％換算して導出される。かかる「％換算する」点が乗数「100」を乗じることに相当する。したがって、役物比率＝役物払出数÷総払出数×100といった式で表すことができる。しかし、役物払出数≤総払出数となるので、役物払出数＝総払出数ではない限り、この式の順では役物払出数÷総払出数の時点で必ず小数点以下の値となってしまう、例えば、メインCPU500aでは演算不能となる。

20

【1049】

そこで、演算順を変更し、役物比率＝役物払出数×100÷総払出数といった順、すなわち、まず、役物払出数に乗数（100）を乗じ、その後、乗数を乗じた結果（被除数）を総払出数で除算する順で演算を行う。こうすることで、小数点以下の演算を行うことなく、整数のみで演算結果を完結することが可能となり、結果的に、小数点以下の切り捨てが簡易に実現される。

【1050】

図105は、役物比率の演算例を示すフローチャートである。図105の例では、仮に、役物払出数が「7123」、総払出数が「10000」として役物比率を導出する。ここでは、例えば、除算を減算に置き換えることで、数値範囲（0～65535）を超えた数の除算を実現している。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

30

【1051】

まず、比率導出手段は、役物払出数が総払出数より大きいかな否かを判定する（S1）。その結果、役物払出数が総払出数より大きいと判定されれば（S1におけるYES）、役物払出数または総払出数が適正な数値ではないとして、役物比率が異常であるという結果を導出し、当該演算処理を終了する（S2）。役物払出数が総払出数以下であると判定されれば（S1におけるNO）、比率導出手段は、総払出数が0であるかな否かを判定する（S3）。その結果、総払出数が0、すなわち、まだメダルの払い出しが行われていない状態であれば（S3におけるYES）、役物比率が0％であると決定して当該演算処理を終了する（S4）。総払出数が0ではないと判定されれば（S3におけるNO）、比率導出手段は、役物払出数と総払出数とが等しいかな否かを判定する（S5）。その結果、役物払出数と総払出数とが等しいと判定されれば（S5におけるYES）、役物比率が99％であると決定して当該演算処理を終了する（S6）。このように、本実施形態では、役物払出数と総払出数とが等しい場合、役物比率として本来100％と示すところ、99％と表している。こうすることで、役物比率を全て2桁で表すことが可能となる。

40

【1052】

役物払出数と総払出数とが等しくないと判定されれば（S5におけるNO）、比率導出

50

手段は、役物払出数「7 1 2 3」に乗数「1 0 0」を乗じ（＝「7 1 2 3 0 0」）、被除数（変数）として保持する（S 7）。続いて、比率導出手段は、被除数「7 1 2 3 0 0」を除数である総払出数「1 0 0 0 0」で除算して商（＝「7 1」）を求める（S 8）。かかる商「7 1」が役物比率となる。これは、役物払出数「7 1 2 3」を総払出数「1 0 0 0 0」で除算した結果を%換算した7 1 . 2 3 %の整数部分に相当するので、かかる演算が正しいことが理解できる。

【1 0 5 3】

このような役物比率の演算例により、加算、減算、乗算といった簡易な演算のみで、また、整数のみにより、除算を実現することが可能となる。

【1 0 5 4】

なお、上記では、役物比率の小数点以下を切り捨てることを前提としたが、かかる場合に限らず、役物比率の小数点以下を四捨五入する場合、負となる最終的な被除数の値（ここでは「- 7 7 0 0」）の絶対値が、除数（総払出数、ここでは「1 0 0 0 0」）の1 / 2より大きければ切り捨て、1 / 2以下であれば切り上げるとすることで実現できる。

【1 0 5 5】

ここでは、さらに、役物比率を算出する際の2段階の演算手順、すなわち、役物払出数×乗数の演算（図1 0 5のS 7）と、乗算結果÷総払出数（図1 0 5のS 8）とに分けて、有効な演算手順を模索する。具体的に、まず、役物払出数×乗数を実行する複数の演算手順の有効性を検討し、次に、乗算結果÷総払出数を実行する複数の演算手順の有効性を検討する。

【1 0 5 6】

（役物払出数×乗数）

役物払出数を数値として表す場合、その数値範囲から、最大値は3バイトとなり、メモリ容量として3バイト（24ビット）を要す。また、乗数は、ここでは1 0 0なので、1バイト（8ビット）で表すことができる。そうすると、役物払出数×乗数の演算は、24ビット×8ビットの演算となる。ここで、メインCPU 5 0 0 aは、8ビット×8ビットの乗算で16ビットの演算結果を得られる8ビットMUL命令を用いて24ビット×8ビットの演算を実行することができる。

【1 0 5 7】

また、メインCPU 5 0 0 aには、上述したように、内蔵デバイスとしてメインCPU 5 0 0 aと独立かつ並行して演算を実行可能な乗算器および除算器が内蔵されている。ここで、乗算器は、コマンドの処理と並行して、例えば16ビット×16ビットの乗算を行うものであり、内蔵レジスタの指定のアドレス（被乗数設定レジスタ）に被乗数を書き込み、他の指定のアドレス（乗数設定レジスタ）に乗数を書き込むと、乗算器内で乗算が実行され、所定の時間後、異なる指定のアドレス（乗算結果レジスタ）に、例えば32ビットの乗算結果が設定される機能部である。また、除算器は、コマンドの処理と並行して、例えば32ビット÷32ビットの除算を行うものであり、内蔵レジスタの指定のアドレス（被除数設定レジスタ）に被除数を書き込み、他の指定のアドレス（除数設定レジスタ）に除数を書き込むと、除算器内で除算が実行され、所定の時間後、異なる指定のアドレス（除算結果レジスタ）に、例えば32ビットの除算結果が設定される機能部である。したがって、メインCPU 5 0 0 aでは、上記の8ビットMUL命令に加え、16ビット×16ビットの乗算器（以下、単に16ビット乗算器という）を採用することができる。

【1 0 5 8】

ここでは、8ビットMUL命令のみを用いた第1の演算例と、16ビット乗算器を用いた第2の演算例を挙げ、その有効性を比較する。

【1 0 5 9】

（役物払出数×乗数を実行する第1の演算例）

図1 0 6は、8ビットMUL命令のみを用いた第1の演算例を説明するための説明図である。8ビットMUL命令のみを用いた第1の演算例では、被乗数（3バイト）×乗数（1バイト）の演算をするため、図1 0 6に示すように、被乗数をBCDレジスタ（BCレ

10

20

30

40

50

ジスタ、および、Dレジスタ）に格納し、8ビットMUL命令を用い、1バイト単位に分離されたDレジスタの値、Cレジスタの値、Bレジスタの値それぞれと乗数（100）とを、その順に乗算する。そして、乗算した結果である、Dレジスタの値×100、Cレジスタの値×100、Bレジスタの値×100それぞれを加算（積算）することで、BCDレジスタの値×100の演算結果（4バイト）を得ることができる。

【1060】

図107は、第1の演算例を実現するための具体的な処理を示したフローチャートであり、図108は、第1の演算例を実現するための具体的なコマンドの一例を示した図である。かかる図107の説明におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1061】

図107に示すように、まず、累計役物カウンタの値をBCDレジスタに読み出す（S1）。具体的に、図108の2行目のコマンド「LD A, (__EX_YAKR)」によって、アドレス「__EX_YAKR」に格納されている1バイト値がAレジスタに読み出され、3行目のコマンド「LD D, A」によって、Aレジスタの値がDレジスタに複製され、4行目のコマンド「LD BC, (__EX_YAKR+1)」によって、アドレス「__EX_YAKR+1」に格納されている2バイト値がBCレジスタに読み出される。

【1062】

次に、図107に示すように、BCDレジスタに格納された累計役物カウンタの値が0であるか否か判定され（S2）、0であれば（S2におけるYES）、乗数を乗じても結果が変わらないので（0なので）、当該第1の演算例を終了する。具体的に、図108の5行目のコマンド「OR A, B」によって、Dレジスタの値と等しいAレジスタの値とBレジスタの値との論理和を演算し、図108の6行目のコマンド「OR A, C」によって、5行目の演算結果であるAレジスタの値とCレジスタの値との論理和を演算し、7行目のコマンド「JR Z, E_PYS AV 28」によって、BCDレジスタの論理和が0であれば、他の処理を示す指標「E_PYS AV 28」（図示せず）に移動する。なお、コマンド「JR Z, E_PYS AV 28」等、Fレジスタの内容によって処理が異なる場合、サイクル数も異なることとなる。例えば、コマンド「JR Z, E_PYS AV 28」は、移動を伴う場合、図108のサイクル数のスラッシュより左に示すように3サイクルとなり、移動を伴わない場合、サイクル数のスラッシュより右に示すように2サイクルとなる。以下のコマンドについても、Fレジスタの内容によって処理が異なる場合は同様の表示をしている。

【1063】

続いて、図107に示すように、被乗数の最下位バイトであるDレジスタの値と乗数（100）とを乗じ（S3）、その演算結果をDEレジスタに設定する（S4）。具体的に、図108の8行目のコマンド「LD A, D」によって、Dレジスタの値をAレジスタに複製し、9行目のコマンド「MUL HL, A, 100」によって、Aレジスタの値と100とを乗じた結果をHLレジスタに保持し、10行目のコマンド「EX DE, HL」によって、HLレジスタの値をDEレジスタに設定する。

【1064】

次に、図107に示すように、被乗数の中位バイトであるCレジスタの値と乗数（100）とを乗じ（S5）、その演算結果にDレジスタの値を加算した結果をDレジスタに戻し（S6）、メインRAM500cに設定する（S7）。具体的に、図108の11行目のコマンド「LD A, C」によって、Cレジスタの値をAレジスタに複製し、12行目のコマンド「MUL HL, A, 100」によって、Aレジスタの値と100とを乗じた結果をHLレジスタに保持し、13行目のコマンド「ADDWB HL, D」によって、乗算結果のHLレジスタに、10行目のコマンド「EX DE, HL」によって設定されたDEレジスタのうちの上位1バイト（Dレジスタの値）のみを加算し、14行目のコマンド「LD D, L」によって、加算結果のうち下位1バイトをDレジスタに戻し、15行目のコマンド「EX DE, HL」によって、乗算結果の下位2バイトに相当するDE

10

20

30

40

50

レジスタの値をHLレジスタに移動し、16行目のコマンド「LD (__EX_WRK2), HL」によって、HLレジスタの2バイト値をアドレス「__EX_WRK2」に格納する。

【1065】

続いて、図107に示すように、被乗数の最上位バイトであるBレジスタの値と乗数(100)とを乗じ(S8)、その演算結果に、ステップS6の演算結果のうち上位1バイト(Hレジスタの値)とを加算し(S9)、メインRAM500cに設定する(S10)。具体的に、図108の17行目のコマンド「LD A, B」によって、Bレジスタの値をAレジスタに複製し、18行目のコマンド「MUL HL, A, 100」によって、Aレジスタの値と100とを乗じた結果をHLレジスタに保持し、19行目のコマンド「ADDWB HL, D」によって、乗算結果のHLレジスタに、15行目のコマンド「EXDE, HL」によって設定されたDEレジスタ(13行目のコマンド「ADDWB HL, D」の演算結果)のうちの上位1バイト(Dレジスタの値)のみを加算し、20行目のコマンド「LD (__EX_WRK2+2), HL」によって、HLレジスタの2バイト値をアドレス「__EX_WRK2+2」に格納する。こうして、アドレス「__EX_WRK2」に、4バイトのBCDレジスタの値×100の演算結果を得ることができる。

【1066】

ここで、当該コマンド例では、図108に示すように、コマンドサイズの合計は38バイトとなり、サイクル数の単純合計(移動命令を考慮することなく単純にサイクル数を合計したもの)は49/48サイクルとなる。

【1067】

(役物払出数×乗数を実行する第2の演算例)

図109は、16ビット乗算器を用いた第2の演算例を説明するための説明図である。16ビット乗算器を用いた第2の演算では、被乗数(3バイト)×乗数(1バイト)の演算をするため、図109に示すように、被乗数をBCDレジスタ(BCレジスタおよびDレジスタ)に格納し、BCレジスタの値(2バイト値)と乗数(100)とを16ビット乗算器を用いて乗算するとともに、Dレジスタの値それぞれと乗数(100)とを8ビットMUL命令を用いて乗算する。そして、乗算した結果、BCレジスタの値×100、Dレジスタの値×100を加算することで、4バイトのBCDレジスタの値×100の演算結果を得ることができる。

【1068】

図110は、第2の演算例を実現するための具体的な処理を示したフローチャートであり、図111は、第2の演算例を実現するための具体的なコマンドの一例を示した図である。かかる図110の説明におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1069】

図110に示すように、まず、累計役物カウンタの値をBCDレジスタに読み出す(S1)。具体的に、図111の2行目のコマンド「LD A, (__EX_YAKR)」によって、アドレス「__EX_YAKR」に格納されている1バイト値がAレジスタに読み出され、3行目のコマンド「LD D, A」によって、Aレジスタの値がDレジスタに複製され、4行目のコマンド「LD BC, (__EX_YAKR+1)」によって、アドレス「__EX_YAKR+1」に格納されている2バイト値がBCレジスタに読み出される。

【1070】

次に、図110に示すように、BCDレジスタに格納された累計役物カウンタの値が0であるか否か判定され(S2)、0であれば(S2におけるYES)、乗数を乗じても結果が変わらないので(0なので)、当該第2の演算例を終了する。具体的に、図111の5行目のコマンド「OR A, B」によって、Dレジスタの値と等しいAレジスタの値とBレジスタの値との論理和を演算し、図111の6行目のコマンド「OR A, C」によって、5行目の演算結果であるAレジスタの値とCレジスタの値との論理和を演算し、7行目のコマンド「JR Z, E_PYS28」によって、BCDレジスタの論理和が

10

20

30

40

50

0であれば、他の処理を示す指標「E_PYSAV28」（図示せず）に移動する。

【1071】

続いて、図110に示すように、被乗数の上位2バイトを被乗数設定レジスタに設定し（S3）、乗数（100）を乗数設定レジスタに設定して、乗算を実行する（S4）。具体的に、図111の8行目のコマンド「LD (@MULA16__), BC」によって、BCレジスタの値を被乗数設定レジスタ「@MULA16__」に設定し、9行目のコマンド「LD A, 100」によって、Aレジスタに乗数「100」を設定し、10行目のコマンド「OUT (LOW @MULB16__), A」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@MULB16__」を下位1バイトとし、そのアドレスで示される乗数設定レジスタにAレジスタの値を出力する。なお、乗数設定レジスタ「@MULB16__」の上位1バイトにはデフォルト値として「00H」が設定されている。こうして、16ビット乗算器を動作させ、乗算結果レジスタに乗算結果が格納される。

10

【1072】

次に、図110に示すように、乗数の最下位バイトであるDレジスタの値と乗数（100）とを乗じる（S5）。具体的に、図111の11行目のコマンド「LD A, D」によって、Dレジスタの値をAレジスタに複製し、12行目のコマンド「MUL HL, A, 100」によって、Aレジスタの値と100とを乗じた結果をHLレジスタに保持する。

【1073】

続いて、図110に示すように、乗算結果レジスタに格納された乗算結果の下位3バイトをDEレジスタに読み出し（S6）、Aレジスタの値と、ステップS5の乗算結果のうち上位1バイト（Hレジスタの値）とを加算し（S7）、メインRAM500cに設定する（S8）。具体的に、図111の13行目のコマンド「IN A, (LOW @MUL32__+0)」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@MULB32__+0」を下位1バイトとし、そのアドレスで示される乗算結果レジスタの値をAレジスタに読み出し、14行目のコマンド「LD ED, (LOW @MUL32__+1)」によって、乗算結果レジスタの最下位から2バイト目と3バイト目をDEレジスタに読み出し、15行目のコマンド「ADD A, H」によって、Aレジスタの値に、12行目のコマンド「MUL HL, A, 100」によって設定された乗算結果の上位1バイト（Hレジスタの値）を加算し、16行目のコマンド「LD H, A」によって、加算結果（Aレジスタの値）をHレジスタに戻し、17行目のコマンド「OUT (LOW @DIVA32__+0), HL」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@DIVA32__」を下位1バイトとし、そのアドレスで示される被除数設定レジスタ「@DIVA32__」にHLレジスタの値を出力する。

20

30

【1074】

次に、図110に示すように、ステップS6で設定したDEレジスタの値をHLレジスタに複製し（S9）、ステップS7の加算において値が繰り上げされたか否か判定され（S10）、繰り上げされていれば（S10におけるYES）、HLレジスタに繰り上げ分を加算し（S11）、メインRAM500cに設定する（S12）。具体的に、図111の18行目のコマンド「EX DE, HL」によって、DEレジスタの値をHLレジスタに設定し、19行目のコマンド「JR NC, E_PYSAVX1」によって、15行目のコマンド「ADD A, H」によりキャリーフラグが立っているか否か判断され、キャリーフラグが立っていなかったら、21行目の指標「E_PYSAVX1」に移動し、キャリーフラグが立っていれば、20行目のコマンド「INC HL」によって、HLレジスタの値が1だけインクリメントされ、22行目のコマンド「OUT (LOW @DIVA32__+2), HL」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@DIVA32__+2」を下位1バイトとし、そのアドレスで示される除数設定レジスタにHLレジスタの値を出力する。

40

【1075】

ここで、当該コマンド例では、図111に示すように、コマンドサイズの合計は43バイトとなり、サイクル数の単純合計は59／57サイクルとなる。

50

【1076】

以上説明したように、役物払出数×乗数を実行する第1の演算例と、役物払出数×乗数を実行する第2の演算例とを比較すると、コマンドサイズの合計およびサイクル数の単純合計のいずれも第1の演算例の方が小さくなる。したがって、スロットマシン400で役物払出数×乗数を実行する場合、図106～図108を用いて説明したように、8ビットMUL命令のみによって実行する方が好ましい。

【1077】

なお、上述した第1の演算例では、図108の16行目および20行目のコマンドとしてコマンド「LD」を用いたが、それ以後の処理態様に応じ、図112のように、これに代えて、16行目および20行目のコマンドとしてコマンド「OUT」を用いてもよい。図112の場合でも、図108同様、コマンドサイズの合計は38バイトとなり、サイクル数の単純合計は49／48サイクルとなる。

【1078】

(乗算結果÷総払出数)

次に、役物比率を算出する際の複数段階の演算手順のうち、(役物払出数×乗数)の乗算結果÷総払出数について有効性を検討する。上述したように、役物払出数(3バイト)×乗数(1バイト)の乗算結果は、最大4バイトとなるため、メモリの容量として4バイト以上を要する。したがって、ここでは、乗算結果を4バイト(32ビット)で表す。また、総払出数も、最大値が3バイトとなるため、メモリ容量として3バイト以上を要する。したがって、総払出数も4バイトで表す。そうすると、乗算結果÷総払出数の演算は32ビット÷32ビットの演算となる。メインCPU500aは、乗算結果から総払出数を繰り返し減算し、その減算回数で32ビット÷32ビットの演算を実現することができる。

【1079】

しかし、単純に、乗算結果から総払出数を繰り返し減算していく方法では、役物比率に相当する回数分、繰り返し乗算結果から総払出数を減算しなければならない。そうすると、減算処理が最大で100回実行されることになる。そこで、除数である総払出数を整数倍して減算することで減算処理の回数を削減することが考えられる。例えば、役物払出数が「7123」、総払出数が「10000」であった場合に、被除数である乗算結果「712300」から総払出数「10000」を単純に減算せず、総払出数「10000」に整数「10」を乗じた「100000」を減算することで、役物比率の2桁目(10の桁)に相当する「7」を導出することができる。また、更新された(減算された)乗算結果から元の総払出数「10000」を減算することで、役物比率の1桁目(1の桁)に相当する「1」を導出することができる。このように、総払出数(除数)を整数倍することで、ここでは、減算処理を多くとも20回に削減することが可能となる。

【1080】

ただし、メインCPU500aでは、2進数で演算が行われるので、ここでは、10進数の桁上がりに相当する数「10」の代わりに、2進数の桁上がりに対応する数「2」を用いる。したがって、ここでは、乗算結果「712300」から、総払出数「10000」に、2の累乗数である所定の整数「64(2の6乗)」を乗じた数を減算し、以降、整数を「32(2の5乗)」→「16(2の4乗)」→「8(2の3乗)」→「4(2の2乗)」→「2(2の1乗)」→「1(2の0乗)」の順に変更して総払出数を乗じ、乗算結果から減算する。ここで、総払出数に乘じる整数を、2の累乗数である64から開始しているのは、役物比率が0～99の範囲に収まるため、「128(2の7乗)」以上の数値を減算する必要がないからである。

【1081】

また、上述したように、メインCPU500aには、例えば、32ビット÷32ビットの除算器(以下、単に32ビット除算器という)が内蔵されている。したがって、メインCPU500aでは、上記の乗算結果から、総払出数に2の累乗数を乗じた数を繰り返し減算する演算(以下、単に2の累乗数を乗じた数の繰り返し減算という)の代わりに、32ビット除算器を採用することができる。

【1082】

ここでは、2の累乗数を乗じた数の繰り返し減算を用いた第3の演算例と、32ビット除算器を用いた第4の演算例を挙げ、その有効性を比較する。

【1083】

(乗算結果÷総払出数を実行する第3の演算例)

図113は、2の累乗数を乗じた数の繰り返し減算を用いた第3の演算例を説明するための説明図である。2の累乗数を乗じた数の繰り返し減算を用いた第3の演算例では、乗算結果である被乗数(4バイト)÷総払出数である除数(4バイト)の演算例を示す。ここで、被除数は「712300」、除数は本来の除数「10000」に64を乗じた「64×10000」と仮定する。

【1084】

図113に示すように、1回目の判定において、被除数 $\geq$ 除数と判定されれば、被乗数から除数を減算できるので、比率導出手段は、被除数「712300」から除数「64×10000」を減算して被除数を「72300」に更新するとともに、商「0」に「64」を加算して商を「64(2進数で表すと1000000B)」に更新する。2回目～4回目の判定では、被除数<除数なので、除数の減算を行わない。5回目の判定において、被除数 $\geq$ 除数と判定されれば、比率導出手段は、被除数「72300」から除数「4×10000」を減算して被除数を「32300」に更新するとともに、商「64」に「4」を加えて商を「68(2進数で表すと1000100B)」に更新する。同様に、6回目の判定において、被除数 $\geq$ 除数と判定されれば、比率導出手段は、被除数「32300」から除数「2×10000」を減算して被除数を「12300」に更新するとともに、商「68」に「2」を加えて商を「70(2進数で表すと1000110B)」に更新する。7回目の判定において、被除数 $\geq$ 除数と判定されれば、比率導出手段は、被除数「12300」から除数「1×10000」を減算して被除数を「2300」に更新するとともに、商「70」に「1」を加えて商を「71(2進数で表すと1000111B)」に更新する。

【1085】

整数＝「64」から、減算ステップおよび加算ステップを7回繰り返すと、整数は小数点以下が切り捨てられて0(<1)となるので、演算処理が終了し、役物比率は、その時点の商「71」となる。これは、役物払出数「7123」を総払出数「10000」で除算した結果を%換算した71.23%の整数部分に相当するので、かかる演算が正しいことが理解できる。

【1086】

図114は、第3の演算例を実現するための具体的な処理を示したフローチャートであり、図115は、第3の演算例を実現するための具体的なコマンドの一例を示した図である。かかる図114の説明におけるステップSの数値は、本図の説明においてのみ用いることとする。なお、ここでは、4バイトの除数を格納しているアドレス「__EX__WRK1」と、4バイトの被除数を格納しているアドレス「__EX__WRK2」とが連続して配置されているとする。

【1087】

図114に示すように、まず、商を0に、整数(2の累乗数)を64に初期化し(S1)、被除数の上位2バイトと除数の上位2バイトを読み出し(S2)、被除数の上位2バイトが除数の上位2バイト未満であるか否か判定する(S3)。そして被除数の上位2バイトが除数の上位2バイト未満であれば(S3におけるYES)、ステップS10に移動する。具体的に、図115の1行目のコマンド「XOR A, A」によって商を示すAレジスタに「0」を設定し、2行目のコマンド「LD C, 64」によって整数を示すCレジスタに「64」を設定し、4行目のコマンド「LD HL, __EX__WRK2+2」によって、被除数(役物払出数×乗数の乗算結果)を格納しているアドレス「__EX__WRK2」の値に2を加算した値(上位2バイトの値)をHLレジスタに読み出し、5行目のコマンド「LD DE, (HL-4)」によって、被除数を格納しているアドレス「__E

10

20

30

40

50

X_WRK 2」の4バイト下位に位置するアドレス「_EX_WRK 1」に格納された除数の上位2バイトをDEレジスタに読み出し、6行目のコマンド「JCP C, (HL), DE, E_PYS AV 2 7」によって、HLレジスタで示されるアドレスの値（被除数の上位2バイト）と、DEレジスタの値（除数の上位2バイト）とが比較され、キャリーフラグが立っていれば、すなわち、被除数の上位2バイトが除数の上位2バイト未満であれば、20行目の指標「E_PYS AV 2 7」に移動する。

【1088】

次に、図114に示すように、被除数の上位2バイトが除数の上位2バイト以上であれば（S3におけるNO）、後段の計算のため、被除数の下位2バイトと除数の下位2バイトを読み出し（S4）、被除数の上位2バイトと除数の上位2バイトとが等しくないか否か、すなわち、被除数の上位2バイトが除数の上位2バイトより大きいと判定する（S5）。そして被除数の上位2バイトと除数の上位2バイトとが等しくなければ（S5におけるYES）、ステップS8に移動する。具体的に、図115の7行目のコマンド「LD HL, _EX_WRK 1」によって、除数を格納しているアドレス「_EX_WRK 1」の下位2バイトをHLレジスタに読み出し、8行目のコマンド「LD DE, (HL + 4)」によって、除数を格納しているアドレス「_EX_WRK 1」の4バイト上位に位置するアドレス「_EX_WRK 2」に格納された被除数の下位2バイトをDEレジスタに読み出し、9行目のコマンド「JR NZ, E_PYS AV 2 6」によって、6行目のコマンドの実行によってゼロフラグが立っていなければ、12行目の指標「E_PYS AV 2 6」に移動する。

【1089】

続いて、図114に示すように、ステップS4で読み出した被除数の下位2バイトが除数の下位2バイトより大きいと判定し（S6）、被除数の下位2バイトが除数の下位2バイトより大きい場合（S6におけるYES）、ステップS8に移動し、除数の下位2バイトが除数の下位2バイト以下の場合（S6におけるNO）、ステップS6において、被除数の下位2バイトと除数の下位2バイトとが等しくないか否か、すなわち、被除数の下位2バイトが除数の下位2バイト未満であるか否か判定する（S7）。そして被除数の下位2バイトと除数の下位2バイトとが等しくなければ（S7におけるYES）、ステップS10に移動する。具体的に、図115の10行目のコマンド「JCP C, (HL), DE, E_PYS AV 2 6」によって、HLレジスタで示されるアドレスの値（除数の下位2バイト）と、DEレジスタの値（被除数の下位2バイト）とが比較され、キャリーフラグが立っていれば、すなわち、被除数の下位2バイトが被除数の下位2バイトより大きいと、12行目の指標「E_PYS AV 2 6」に移動し、11行目のコマンド「JR NZ, E_PYS AV 2 7」によって、10行目のコマンドの実行によってゼロフラグが立っていなければ、20行目の指標「E_PYS AV 2 7」に移動する。

【1090】

次に、図114に示すように、被除数が除数以上であれば、商に、現在の整数を加算し（S8）、被除数から除数を減算して、その減算結果を新たな被除数として更新する（S9）。具体的に、図115の13行目のコマンド「ADD A, C」によって、商を示すAレジスタに、整数を示すCレジスタの値を加算し、14行目のコマンド「SUB DE, (HL)」によって、被除数の下位2バイトが格納されたDEレジスタの値から、HLレジスタで示されるアドレスに格納された除数の下位2バイトの値を減算してDEレジスタを更新し、15行目のコマンド「LD (HL + 4), DE」によって、HLレジスタで示される除数を格納しているアドレスの4バイト上位に位置する被除数を示すアドレスに、減算結果である更新された被除数の下位2バイトを格納し、16行目のコマンド「LD DE, (HL + 2)」によって、HLレジスタで示されるアドレスに格納された除数の上位2バイトをDEレジスタに読み出し、17行目のコマンド「LD HL, (HL + 6)」によって、HLレジスタで示される除数を格納しているアドレスの6バイト上位に位置する被除数の上位2バイトをHLレジスタに読み出し、18行目のコマンド「SBC HL, DE」によって、被除数の上位2バイトが格納されたHLレジスタの値から、除数

10

20

30

40

50

の上位2バイトが格納されたDEレジスタの値を減算してHLレジスタを更新し、19行目のコマンド「LD (__EX_WRK2+2), HL」によって、アドレス「__EX_WRK2」が示す被除数の上位2バイトに、減算結果である更新された被除数の上位2バイトを格納する。

【1091】

続いて、図114に示すように、除数を2で除算（右にビットシフト）し、新たな除数として更新し（S10）、整数を2で除算（右にビットシフト）し、新たな整数として更新し（S11）、整数が0ではないか否か判定し（S12）、0でなければ（S12におけるYES）、ステップS2からの処理を繰り返し、0であれば（S12におけるNO）、役物比率をメインRAM500cに設定し（S13）、当該2の累乗数を乗じた数を繰り返し減算する処理を終了する。具体的に、図115の21行目のコマンド「LD HL, __EX_WRK1+3」によって、除数の最上位1バイトを示すアドレス「__EX_WRK1+3」をHLレジスタに読み出し、22行目のコマンド「SRL (HL)」によってHLレジスタで示されるアドレスの値（除数の4バイト目）が右に1だけビットシフトされ、23行目のコマンド「DEC HL」によって、HLレジスタの値が1だけデクリメントされ、24行目のコマンド「RR (HL)」によってHLレジスタで示されるアドレスの値（除数の下位から3バイト目）が右に1だけビットシフトされ、25行目のコマンド「DEC HL」によって、HLレジスタの値が1だけデクリメントされ、26行目のコマンド「RR (HL)」によってHLレジスタで示されるアドレスの値（除数の下位から2バイト目）が右に1だけビットシフトされ、27行目のコマンド「DEC HL」によって、HLレジスタの値が1だけデクリメントされ、28行目のコマンド「RR (HL)」によってHLレジスタで示されるアドレスの値（除数の最下位1バイト）が右に1だけビットシフトされ、29行目のコマンド「SRL C」によって、整数を示すCレジスタの値が右に1だけビットシフトされ、30行目のコマンド「JR NZ, E__PYSAV25」によって、29行目のコマンドの実行によってゼロフラグが立っていないければ、3行目の指標「E__PYSAV25」に移動し、32行目の「LD (__EX_DPA7), A」によって、商であるAレジスタの値をアドレス「__EX_DPA7」に格納する。

【1092】

かかる第3の演算例により、加算、減算、乗算といった簡易な演算のみで、また、整数のみにより、除算を実現することが可能となる。また、ここでは、減算および加算処理を7回繰り返せば足りるので、演算処理を迅速に行うことができる。さらに、役物比率の小数点以下を切り捨てることを前提とすると、誤差が生じないので、役物比率を正確に導出することが可能となる。

【1093】

ここで、当該コマンド例では、図115に示すように、コマンドサイズの合計は59バイトとなり、サイクル数の単純合計は95/90サイクルとなる。また、実際のサイクル数は、以下ようになる。すなわち、当該処理では、指標「E__PYSAV25」の次のコマンド（4行目）から指標「E__PYSAV28」の直前のコマンド（30行目）までを7回ループすることとなる。このとき、7回全てのループにおいて、6行目のコマンド「JCP C, (HL), DE, E__PYSAV27」で指標「E__PYSAV27」に移動した場合、7行目～19行目のコマンドを実行しないので、サイクル数は最小の321（3+45（最初の6ループ）×6+44（7ループ目）+4）サイクルとなる。一方、全ループにおいて、6行目のコマンド「JCP C, (HL), DE, E__PYSAV27」で指標「E__PYSAV27」に移動しなかった場合、サイクル数は最大となる。ただし、役物比率は100%を超えることはなく、95%および63%のときの処理負荷が最大になることを勘案すると、サイクル数の最大は566（3+61（初回ループ）+83（残りの6ループ）×6+4）サイクルとなる。したがって、サイクル数の実質的な合計は、321～566サイクルの範囲となる。

【1094】

なお、ここでも、役物比率の小数点以下を切り捨てることを前提としたが、かかる場合に限らず、役物比率の小数点以下を四捨五入する場合、減算および加算処理を8回繰り返して、8回目の判定において、被除数 $\geq$ 除数と判定されれば、商に1を加えることで実現できる。

【1095】

(乗算結果 $\div$ 総払出数を実行する第4の演算例)

図116は、第4の演算例を実現するための具体的な処理を示したフローチャートであり、図117は、第4の演算例を実現するための具体的なコマンドの一例を示した図である。かかる図116の説明におけるステップSの数値は、本図の説明においてのみ用いることとする。なお、ここでは、図112を用いて説明したように、役物払出数 $\times$ 乗数の乗算結果である被除数は、すでに被除数設定レジスタ「DIVA32」に設定されているとする。

【1096】

図116に示すように、まず、累計役物カウンタの値をレジスタに呼び出し、除数設定レジスタ「DIVB32」に設定する(S1)。具体的に、図117の1行目のコマンド「LD HL, (EXPRZR)」によって、累計役物カウンタを示すアドレス「EXPRZR」に格納されている値がHLレジスタに読み出され、2行目のコマンド「LD A, (EXPRZR+2)」によって、累計役物カウンタの上位1バイトを示すアドレス「EXPRZR+2」に格納されている値がAレジスタに読み出され、3行目のコマンド「OUT (LOW @DIVB32_+0), HL」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@DIVB32_」を下位1バイトとし、そのアドレスに、HLレジスタの値を出力し、4行目のコマンド「OUT (LOW @DIVB32_+2), A」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@DIVB32_+2」を下位1バイトとし、そのアドレスに、Aレジスタの値を出力する。なお、除数設定レジスタ「@DIVB32_」の上位1バイトにはデフォルト値として「00H」が設定されている。

【1097】

次に、図116に示すように、除算の時間待ちを行い(S2)、所定時間経過後、演算結果をメインRAM500cに設定する(S3)。具体的に、図117の5~8行目のコマンド「NOP」によって4サイクルの時間を消費させ、9行目のコマンド「IN A, (LOW @DIV32_+0)」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@DIV32_+0」を下位1バイトとし、そのアドレスで示される除算結果レジスタの値をAレジスタに読み出し、11行目の「LD (EXDPA7), A」によって、商であるAレジスタの値をアドレス「EXDPA7」に格納する。

【1098】

ここで、当該コマンド例では、図117に示すように、コマンドサイズの合計は20バイトとなり、サイクル数の合計は28サイクルとなる。

【1099】

以上説明したように、乗算結果 $\div$ 総払出数を実行する第3の演算例と、乗算結果 $\div$ 総払出数を実行する第4の演算例とを比較すると、コマンドサイズの合計およびサイクル数の合計のいずれも第4の演算例の方が小さくなる。したがって、スロットマシン400で乗算結果 $\div$ 総払出数を実行する場合、図116、図117を用いて説明したように、32ビット除算器を用いて実行する方が好ましい。

【1100】

以上、第1の演算例~第4の演算例を挙げて、スロットマシン400における役物比率の演算態様を説明した。以下、パチンコ機におけるベース値について説明する。なお、スロットマシン400における有利区間比率、指示込役物比率、連続役物比率、役物比率、役物等状態比率については、全て小数点以下を切り捨てるとしているが、パチンコ機におけるベース値(賞球数/アウト数)は小数点以下を四捨五入する。そこで、パチンコ機においては、賞球数の乗数として「100」ではなく「200」を採用し、小数点以下が0

10

20

30

40

50

・ 5 以上であるか否かを整数（最下位ビット）に置き換えて判断する。したがって、ベース値は、賞球数に乗数（200）を乗じ、その後、乗数を乗じた結果（被除数）をアウト数で除算し、最下位ビット（MLB）に応じて切り上げまたは切り捨てを判断する。

【1101】

ここでは、8ビットMUL命令および2の累乗数を乗じた数の繰り返し減算を用いた第5の演算例と、16ビット乗算器および32ビット除算器を用いた第6の演算例を挙げ、その有効性を比較する。

【1102】

（賞球数×乗数÷アウト数を実行する第5の演算例）

図118は、8ビットMUL命令および2の累乗数を乗じた数の繰り返し減算を用いた第5の演算例を実現するための具体的な処理を示したフローチャートであり、図119は、第5の演算例を実現するための具体的なコマンドの一例を示した図である。かかる図118の説明におけるステップSの数値は、本図の説明においてのみ用いることとする。なお、ここでは、3バイトの被除数（賞球数に乗数を乗じた演算結果）を格納しているアドレス「R_ERW_BF1」と、3バイトの除数（アウト数）を格納しているアドレス「R_ERW_BF2」とが連続して配置されているとする。

【1103】

図118に示すように、まず、通常賞球カウンタの値をBCレジスタに読み出し（S1）、通常アウトカウンタの値をHLレジスタに読み出し（S2）、HLレジスタが0であるか否か判定され（S3）、0であれば（S3におけるYES）、ステップS23に移行する。具体的に、図119の2行目のコマンド「XOR A, A」によって、後程、除数の計算に用いられるAレジスタの値を初期化し、3行目の「LD BC, (R_ERW_TJS)」によって、アドレス「R_ERW_TJS」で示される通常賞球カウンタの値（賞球数）がBCレジスタに読み出され、4行目のコマンド「LD HL, (R_ERW_TJO)」によって、アドレス「R_ERW_TJO」で示される通常アウトカウンタの値（アウト数）がHLレジスタに読み出され、5行目のコマンド「JR TZ, E_SHYMT_60」によって、HLレジスタの値が0であれば、49行目の指標「E_SHYMT_60」に移動する。

【1104】

次に、図118に示すように、アウト数に整数（2の累乗数、ここでは128）を乗じ（S4）、その演算結果を除数バッファとしてメインRAM500cに格納する（S5）。具体的に、図119の6行目のコマンド「SRL H」によって、Hレジスタの値が右に1だけビットシフトされ、7行目のコマンド「RRL」によって、Hレジスタからのキャリアも含めLレジスタの値が右に1だけビットシフトされ、8行目のコマンド「RRA」によって、Lレジスタからのキャリアも含めAレジスタの値が右に1だけビットシフトされる。ここで、HLAレジスタという単位で数値を判断すると、HLレジスタの値に256を乗じ、その後右に1だけビットシフトされていると見做せるので、アウト数に128（256／2）を乗じたこととなり、これが除数となる。そして、図119の9行目のコマンド「LD (R_ERW_BF2), A」によって、除数の下位バイトであるAレジスタの値をアドレス「R_ERW_BF2」に格納し、10行目のコマンド「LD (R_ERW_BF2+1), HL」によって、除数の上位2バイトであるHLレジスタの値をアドレス「R_ERW_BF2+1」に格納する。

【1105】

続いて、図118に示すように、被乗数（賞球数）の下位バイトであるCレジスタの値と乗数（200）とを乗じ（S6）、その演算結果をDEレジスタに設定する（S7）。具体的に、図119の11行目のコマンド「LD A, C」によって、Cレジスタの値をAレジスタに複製し、12行目のコマンド「MUL HL, A, 200」によって、Aレジスタの値と200とを乗じた結果をHLレジスタに保持し、13行目のコマンド「EX DE, HL」によって、HLレジスタの値をDEレジスタに設定する。

【1106】

10

20

30

40

50

次に、図118に示すように、被乗数の上位バイトであるBレジスタの値と乗数（200）とを乗じ（S8）、その演算結果にDレジスタの値を加算して演算結果を更新し（S9）、その演算結果を被除数バッファとしてメインRAM500cに格納する（S10）。具体的に、図119の14行目のコマンド「LD A, B」によって、Bレジスタの値をAレジスタに複製し、15行目のコマンド「MUL HL, A, 200」によって、Aレジスタの値と200とを乗じた結果をHLレジスタに保持し、16行目のコマンド「ADDWB HL, D」によって、乗算結果のHLレジスタに、13行目のコマンド「EXDE, HL」によって設定されたDEレジスタのうちの上位1バイト（Dレジスタの値）のみを加算し、17行目のコマンド「LD A, E」によって、13行目のコマンド「EXDE, HL」によって設定されたDEレジスタのうちの下位1バイトをAレジスタに設定し、18行目のコマンド「LD (R__ERW__BF1), A」によって、被除数の下位バイトであるAレジスタの値をアドレス「R__ERW__BF1」に格納し、19行目のコマンド「LD (R__ERW__BF1+1), HL」によって、被除数の上位2バイトであるHLレジスタの値をアドレス「R__ERW__BF1+1」に格納する。

【1107】

続いて、図118に示すように、ループ回数を8に、商を0に初期化し（S11）、商を2倍し（S12）、被除数の上位2バイトと除数の上位2バイトを読み出し（S13）、被除数の上位2バイトが除数の上位2バイト未満であるか否か判定する（S14）。そして被除数の上位2バイトが除数の上位2バイト未満であれば（S14におけるYES）、ステップS20に移動する。具体的に、図119の20行目のコマンド「LD BC, 8*256+0」によって、ループ回数を示すBレジスタに8を、商を示すCレジスタに0を設定し、22行目のコマンド「SLA C」によって、Cレジスタの値を左に1ビットシフトし（MSBには0が入る）、23行目のコマンド「LD HL, R__ERW__BF1+1」によって、被除数（賞球数×乗数の乗算結果）を格納しているアドレス「R__ERW__BF1」の値に1を加算した値をHLレジスタに読み出し、24行目のコマンド「LD DE, (HL+3)」によって、被除数を格納しているアドレス「R__ERW__BF1」の3バイト上位に位置するアドレス「R__ERW__BF2」に格納された除数の上位2バイトをDEレジスタに読み出し、25行目のコマンド「JCP C, (HL), DE, E__SHYMT__55」によって、HLレジスタで示されるアドレスの値（被除数の上位2バイト）と、DEレジスタの値（除数の上位2バイト）とが比較され、キャリーフラグが立っていれば、すなわち、被除数の上位2バイトが除数の上位2バイト未満であれば、38行目の指標「E__SHYMT__55」に移動する。

【1108】

次に、図118に示すように、被除数の上位2バイトが除数の上位2バイト以上であれば（S14におけるNO）、後段の計算のため、除数の下位1バイトと被除数の下位1バイトを読み出し（S15）、ステップS14において、被除数の上位2バイトと除数の上位2バイトとが等しくないか否か、すなわち、被除数の上位2バイトが除数の上位2バイトより大きいのか否か判定する（S16）。そして被除数の上位2バイトと除数の上位2バイトとが等しければ（S16におけるYES）、ステップS18に移動し、被除数の上位2バイトと除数の上位2バイトとが等しければ（S16におけるNO）、被除数の下位1バイトが除数の下位1バイト未満であるか否か判定する（S17）。そして被除数の下位1バイトが除数の下位1バイト未満であれば（S17におけるYES）、ステップS20に移動し、被除数の下位1バイトが除数の下位1バイト以上であれば（S17におけるNO）、ステップS18に移動する。具体的に、図119の26行目のコマンド「LD L, LOW R__ERW__BF2」によって、除数を格納しているアドレス「R__ERW__BF2」の下位1バイトをLレジスタに読み出し（なお、Hレジスタには、23行目のコマンド「LD HL, R__ERW__BF1+1」によって、既に「R__ERW__BF2」の上位1バイトが保持されている）、27行目のコマンド「LD A, (HL-3)」によって、除数を格納しているアドレス「R__ERW__BF2」の3バイト下位に位置するアドレス「R__ERW__BF1」に格納された被除数の下位1バイトをAレジスタに読

み出し、28行目のコマンド「JR NZ, E__SHYMT__50」によって、25行目のコマンドの実行によってゼロフラグが立っていなければ、31行目の指標「E__SHYMT__50」に移動し、29行目のコマンド「CP A, (HL)」によって、Aレジスタの値（被除数の下位1バイト）と、HLレジスタで示されるアドレスの値（除数の下位1バイト）とが比較され、30行目のコマンド「JR C, E__SHYMT__55」によって、キャリーフラグが立っていれば、すなわち、被除数の下位1バイトが除数の下位1バイト未満であれば、38行目の指標「E__SHYMT__55」に移動する。

【1109】

続いて、図118に示すように、被除数が除数以上であれば、被除数から除数を減算して、その減算結果を新たな被除数として更新し（S18）、商に1を加算する（S19）。具体的に、図119の32行目のコマンド「SUB A, (HL)」によって、被除数の下位1バイトが格納されたAレジスタの値から、HLレジスタで示される除数の下位1バイトの値を減算し、33行目のコマンド「LD (HL-3), A」によって、HLレジスタで示される除数を格納しているアドレスの3バイト下位に位置する被除数を示すアドレスに、減算結果である更新された被除数の下位1バイトを格納し、34行目のコマンド「LD HL, (HL-2)」によって、HLレジスタで示される除数を格納しているアドレスの2バイト下位に位置する被除数の上位2バイトを示すアドレスをHLレジスタに読み出し、35行目のコマンド「SBC HL, DE」によって、被除数の上位2バイトが格納されたHLレジスタの値から、除数の上位2バイトが格納されたDEレジスタの値を減算し、36行目のコマンド「LD (R__ERW__BF1+1), HL」によって、アドレス「R__ERW__BF1」が示す被除数の上位2バイトに、減算結果である更新された被除数の上位2バイトを格納し、37行目のコマンド「INC C」によって、商を示すCレジスタの値を1だけインクリメントする。

【1110】

次に、図118に示すように、除数を2で除算（右にビットシフト）し、新たな除数として更新し（S20）、ループ回数を減算し、そのループ回数が0ではないか否か判定し（S21）、0でなければ（S21におけるYES）、ステップS12からの処理を繰り返し、0であれば（S21におけるNO）、四捨五入を実行し（S22）、ベース値をメインRAM500cに設定して（S23）、当該処理を終了する。具体的に、図119の39行目のコマンド「LD HL, R__ERW__BF2+2」によって、除数の上位1バイトを示すアドレス「R__ERW__BF2+2」をHLレジスタに読み出し、40行目のコマンド「SRL (HL)」によってHLレジスタで示されるアドレスの値（除数の3バイト目）が右に1だけビットシフトされ、41行目のコマンド「DEC HL」によって、HLレジスタの値が1だけデクリメントされ、42行目のコマンド「RR (HL)」によってHLレジスタで示されるアドレスの値（除数の2バイト目）が右に1だけビットシフトされ、43行目のコマンド「DEC HL」によって、HLレジスタの値が1だけデクリメントされ、44行目のコマンド「RR (HL)」によってHLレジスタで示されるアドレスの値（除数の1バイト目）が右に1だけビットシフトされ、45行目のコマンド「DJNZ E__SHYMT__45」によって、ループ回数を示すBレジスタの値が1だけデクリメントされ、その値が0でなければ、21行目の指標「E__SHYMT__45」へ移動する。一方、ループ回数が0となると、図119の46行目のコマンド「LD A, C」によって、商を示すCレジスタの値がAレジスタに複製され、47行目のコマンド「SRL A」によってAレジスタで示されるアドレスの値が右に1だけビットシフトされ、48行目のコマンド「ADC A, 0」によって、47行目のコマンド「SRL A」によってキャリーフラグが立っていれば、すなわち、小数点以下が0.5以上であれば、Aレジスタに1が加算され、50行目のコマンド「LD HL, R__ERW__BAS」によって、ベース値を示すアドレス「R__ERW__BAS」がHLレジスタに設定され、51行目のコマンド「LD (HL), A」によって、ベース値を示すAレジスタの値がアドレス「R__ERW__BAS」に格納される。

【1111】

10

20

30

40

50

かかる第5の演算例により、加算、減算、乗算といった簡易な演算のみで、また、2の累乗数を乗じた数の繰り返し減算を用いてベース値を導出することが可能となる。

【1112】

ここで、当該コマンド例では、図119に示すように、コマンドサイズの合計は97バイトとなり、サイクル数の単純合計は136／131サイクルとなる。また、実際のサイクル数は、以下ようになる。すなわち、当該処理では、指標「E__SHYMT__45」の次のコマンド（22行目）から45行目のコマンド「DJNZ E__SHYMT__45」までを8回ループすることとなる。このとき、8回全てのループにおいて、25行目のコマンド「JCP C, (HL), DE, E__SHYMT__55」で指標「E__SHYMT__55」に移動した場合、26行目～37行目のコマンドを実行しないので、サイクル数は最小の367（54+38（最初の7ループ）×7+37（8回目のループ）+10）サイクルとなる。一方、全ループにおいて、25行目のコマンド「JCP C, (HL), DE, E__SHYMT__55」で指標「E__SHYMT__55」に移動しなかった場合、サイクル数は最大となる。ただし、取り得る数値の関係で、1、2回目のループでは、25行目のコマンド「JCP C, (HL), DE, E__SHYMT__55」で移動せず、28行目のコマンド「JR NZ, E__SHYMT__50」で移動し、3～8回目のループでは、25行目のコマンド「JCP C, (HL), DE, E__SHYMT__55」、28行目のコマンド「JR NZ, E__SHYMT__50」、30行目のコマンド「JR C, E__SHYMT__55」のいずれにおいても移動しない場合が最大となることを勘案すると、サイクル数の最大は601（54+65（最初の2ループ）×2+68（その後の5ループ）×5+67（8回目のループ）+10）サイクルとなる。したがって、サイクル数の実質的な合計は、367～601サイクルの範囲となる。

【1113】

（賞球数×乗数÷アウト数を実行する第6の演算例）

図120は、16ビット乗算器および32ビット除算器を用いた第6の演算例を実現するための具体的な処理を示したフローチャートであり、図121は、第6の演算例を実現するための具体的なコマンドの一例を示した図である。かかる図120の説明におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1114】

図120に示すように、まず、賞球数を被乗数レジスタに設定し（S1）、乗数「200」を乗数レジスタに設定する（S2）。具体的に、図121の1行目のコマンド「LD HL, (R__ERW__TJS)」によって、アドレス「R__ERW__TJS」で示される通常賞球カウンタの値（賞球数）がHLレジスタに読み出され、2行目のコマンド「OUT (LOW @MULA16__）」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@MULA16__」を下位1バイトとし、そのアドレスで示される乗数設定レジスタにHLレジスタの値を出力し、3行目のコマンド「EX DE, HL」によって、HLレジスタの値をDEレジスタに複製し、4行目のコマンド「LDA, 200」によって、Aレジスタに乗数「200」を設定し、5行目のコマンド「OUT (LOW @MULB16__), A」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@MULB16__」を下位1バイトとし、そのアドレスで示される乗数設定レジスタにAレジスタの値を出力する。なお、乗数設定レジスタ「@MULB16__」の上位1バイトにはデフォルト値として「00H」が設定されている。

【1115】

次に、図120に示すように、通常賞球カウンタが0である場合を想定し、ベース値の初期値として0を設定し（S3）、アウト数が0であるか否か判定され（S4）、0であれば（S4におけるYES）、乗数を乗じても結果が変わらないので（0なので）、ステップS13に移動し、0でなければ（S4におけるNO）、ベース値を100に設定し（S5）、アウト数が賞球数未満であるか否か判定し（S6）、アウト数が賞球数未満であれば（S6におけるYES）、ステップS13に移動する。具体的に、図121の6行目のコマンド「XOR A, A」によってベース値を示すAレジスタの値を0に設定し、7

行目のコマンド「LD HL, (R__ERW__TJO)」によって、アドレス「R__ERW__TJO」で示される通常アウトカウンタの値（アウト数）がHLレジスタに読み出され、8行目のコマンド「JR TZ, E__SHYMT__60」によって、HLレジスタの値が0であれば、23行目の指標「E__SHYMT__60」に移動し、9行目のコマンド「LD A, 100」によって、ベース値を示すAレジスタに初期値として100を設定し、10行目のコマンド「JCP C, HL, DE, E__SHYMT__60」によって、HLレジスタの値（アウト数）と、DEレジスタの値（賞球数）とが比較され、キャリーフラグが立っていれば、すなわち、アウト数が賞球数未満であれば、23行目の指標「E__SHYMT__60」に移動する。

【1116】

続いて、図120に示すように、アウト数を除数設定レジスタに設定し（S7）、16ビット乗算器による乗算結果をHLAレジスタ（HLレジスタおよびAレジスタ）に読み出し（S8）、それを被除数設定レジスタに設定する（S9）。具体的に、図121の11行目のコマンド「OUT (LOW @DIVB32_), HL」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@DIVB32_」を下位1バイトとし、そのアドレスで示される除数設定レジスタ「@DIVB32_」にHLレジスタの値を出力し、12行目のコマンド「IN A, (LOW @MUL32_」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@MULB32_」を下位1バイトとし、そのアドレスで示される乗算結果レジスタの1バイト値をAレジスタに読み出し、13行目のコマンド「IN HL, (LOW @MUL32_+1」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@MULB32_+1」を下位1バイトとし、そのアドレスで示される乗算結果レジスタの2バイト値をHLレジスタに読み出し、14行目のコマンド「OUT (LOW @DIVA32_+0), A」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@DIVA32_」を下位1バイトとし、そのアドレスで示される被除数設定レジスタ「@DIVA32_」にAレジスタの値を出力し、15行目のコマンド「OUT (LOW @DIVA32_+1), HL」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@DIVA32_+1」を下位1バイトとし、そのアドレスで示される被除数設定レジスタ「@DIVA32_+1」にHLレジスタの値を出力する。こうして、16ビット乗算器を動作させ、乗算結果レジスタに乗算結果が格納される。

【1117】

次に、図120に示すように、除算の時間待ちを行い（S10）、所定時間経過後、演算結果をAレジスタに読み出し（S11）、四捨五入を実行し（S12）、ベース値をメインRAM500cに設定して（S13）、当該処理を終了する。具体的に、図121の16～19行目のコマンド「NOP」によって4サイクルの時間を消費させ、20行目のコマンド「IN A, (LOW @DIV32_+0)によって、除算器から演算結果をAレジスタに読み出し、21行目のコマンド「SRL A」によってAレジスタで示されるアドレスの値が右に1だけビットシフトされ、22行目のコマンド「ADC A, 0」によって、21行目のコマンド「SRL A」によってキャリーフラグが立っていれば、すなわち、小数点以下が0.5以上であれば、Aレジスタに1が加算され、24行目のコマンド「LD HL, R__ERW__BAS」によって、ベース値を示すアドレス「R__ERW__BAS」がHLレジスタに設定され、25行目のコマンド「LD (HL), A」によって、ベース値を示すAレジスタの値がアドレス「R__ERW__BAS」に格納される。

【1118】

ここで、当該コマンド例では、図121に示すように、コマンドサイズの合計は49バイトとなり、サイクル数の単純合計は68／66サイクルとなる。

【1119】

以上説明したように、ベース値を導出する第5の演算例と、ベース値を導出する第6の演算例とを比較すると、コマンドサイズの合計およびサイクル数の単純合計のいずれも、

16ビット乗算器および32ビット除算器を用いた第6の演算例の方が小さくなる。したがって、パチンコ機でベース値を実行する場合、図120、図121を用いて説明したように、16ビット乗算器および32ビット除算器を用いて実行する方が好ましい。

【1120】

以上、説明したように、本実施形態によると、処理負荷の増大を抑えることで、タイマ割込みの禁止区間を短縮化することができ、遊技情報を適切に得つつ、遊技機としての動作を安定化することが可能となる。

【1121】

なお、上述した実施形態においては、第4の演算例や第6の演算例において、32ビット除算器による除算の時間待ちに1サイクルの時間を消費する「NOP」を4回連続して利用している。しかし、かかる場合に限らず、4サイクルを確保しさえすれば、他のコマンドでもよい。例えば、コマンドサイズ1バイトで2サイクルのコマンドを2回連続して利用するか、コマンドサイズ1バイトで4サイクルのコマンドや、コマンドサイズ2バイトで4サイクルのコマンドを利用するとしてもよい。こうして、コマンドサイズを削減することができる。

【1122】

また、上述した実施形態においては、役物払出数×乗数を実行する演算例として8ビットMUL命令のみを用いた第1の演算例と、16ビット乗算器を用いた第2の演算例を挙げ、また、乗算結果÷総払出数を実行する演算例として2の累乗数を乗じた数の繰り返し減算を用いた第3の演算例と、32ビット除算器を用いた第4の演算例を挙げた。上述した実施形態においては、役物払出数×乗数を行った後、その乗算結果を用いて乗算結果÷総払出数を実行するので、第1の演算例または第2の演算例と、第3の演算例または第4の演算例とを組み合わせることで演算することになる。かかる演算例の組み合わせは任意に設定できる。例えば、第1の演算例と第3の演算例との組み合わせ、第1の演算例と第4の演算例との組み合わせ、第2の演算例と第3の演算例との組み合わせ、第2の演算例と第4の演算例との組み合わせを採用することができる。また、第5の演算例および第6の演算例は、いずれも、役物払出数×乗数と、乗算結果÷総払出数とに分けることができ、かかる演算例の組み合わせは任意に設定できる。例えば、上述した第5の演算例の役物払出数×乗数と、第5の演算例の乗算結果÷総払出数との組み合わせ、第6の演算例の役物払出数×乗数と、第6の演算例の乗算結果÷総払出数との組み合わせに加え、第5の演算例の役物払出数×乗数と、第6の演算例の乗算結果÷総払出数との組み合わせ、第6の演算例の役物払出数×乗数と、第5の演算例の乗算結果÷総払出数との組み合わせを採用することができる。

【1123】

また、上述した実施形態においては、スロットマシン400の役物比率を例に挙げて、様々な演算（第1の演算例～第4の演算例）を実行したが、かかる場合に限らず、スロットマシン400の各比率（有利区間比率、指示込役物比率、6000遊技の連続役物比率、6000遊技の役物比率、総累計の連続役物比率、総累計の役物比率、役物等状態比率）に適用できるのは言うまでもない。また、スロットマシン400に限らず、第5の演算例や第6の演算例を用いて説明したように、パチンコ機のベース値にも適用可能である。

【1124】

〈モジュールおよびコマンドの説明〉

以下、上述した各処理（モジュール）について、それを実現するための具体的な命令コード（コマンド群）を、モジュール内で主として用いられるコマンド単位で説明する。

【1125】

〈コマンド「ADDA LD」〉

BYTESELモジュールは、バイトデータ選択処理、すなわち、選択アドレスをオフセットさせ、そのオフセットされたアドレスに格納された1バイト値を読み出すための汎用モジュールである。汎用モジュールは、プログラムを実行する上で用いられる同一の処理をサブルーチン化したもので、特に他のモジュールから呼び出される頻度が高いモジュ

10

20

30

40

50

ールを示す。汎用モジュールは、図5（または図102）で示したメモリマップ上の最も低いアドレス0000H近傍に配置されることが多い。これは、呼び出される頻度の高い汎用モジュールを0000H近傍に配置することで、呼び出しに用いられるコマンド「CALL」のみならず、他のコマンド「RST」を利用して呼び出せるからである。なお、ここでは、BYTESELモジュールをメモリマップ上の0008Hに配置する例を挙げて説明する。

【1126】

ここで、コマンド「RST」は、メモリマップのうち0000H近傍の低い数値のアドレスであり、8バイトずつ離れた複数のアドレス（0008H、0010H、0018H、0020H、0028H、0030H、0038H、0040H）のいずれかを呼び出すことができるコマンドである。コマンド「RST」は、呼び出しに用いられる通常のコマンド「CALL」と実行サイクルは「4」で等しいが、通常のコマンド「CALL」のコマンドサイズが「3」であるのに対し、コマンド「RST」のコマンドサイズは「1」である。したがって、コマンド「RST」を用いることでプログラムの短縮化を図ることができる。

【1127】

メインCPU300aは、メインROM300bからプログラムを読み出し、読み出したプログラムを遂行し、任意の処理において、サブルーチンとしてBYTESELモジュールを呼び出し、BYTESELモジュールを遂行する。BYTESELモジュールでは、選択アドレスを所定値だけオフセットさせ、オフセットされたアドレスに格納された1バイト値を保持する。こうして、オフセットに応じた値を設定することができる。

【1128】

図122は、BYTESELモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図26のS400-31で示したLED表示設定処理を実行するLED_PRCモジュールの一部を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該BYTESELモジュールの説明中、第1レジスタはHLレジスタであり、第2レジスタはAレジスタである。

【1129】

メインCPU300aは、図122（a）のように、任意の処理において、オフセットをAレジスタに設定し（S1）、読み出し対象となるアドレス群の先頭アドレスをHLレジスタに設定する（S2）。そして、サブルーチンとしてBYTESELモジュールを呼び出す（S3）。

【1130】

メインCPU300aは、図122（b）のように、BYTESELモジュールにおいて、HLレジスタの値にAレジスタの値を加算してHLレジスタを更新するとともに、HLレジスタで示されるアドレスに格納された1バイト値をAレジスタに読み出す（S4）。続いて、メインCPU300aは、Aレジスタの値が0（ゼロ）か否かに応じて、ゼロフラグ（第1ゼロフラグ）を設定する（S5）。そして、当該BYTESELモジュールを終了して1段上のルーチンに戻る（S6）。

【1131】

図123、図124は、BYTESELモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図123（a）は、BYTESELモジュールを呼び出す任意の処理のコマンド群を示し、図123（b）は、BYTESELモジュールのコマンド群を示し、図123（c）は、メインROM300bのプログラムデータにおける1バイトデータ群の配置を示す。図122で示したフローチャートは、例えば、図123に示したプログラムによって実現される。

【1132】

図123（a）の1行目のコマンド「LDQ A, (LOW R_TZ1_DSP)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_TZ1_DSP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特

10

20

30

40

50

別図柄 1 表示図柄カウンタのカウンタ値) を A レジスタに読み出す。かかる 1 行目のコマンドが、図 1 2 2 (a) のステップ S 1 に対応する。2 行目のコマンド「LD HL, D__DSP__TZ」によって、読み出し元となる特別図柄表示用 LED 出力データテーブルのデータ群の先頭アドレス「D__DSP__TZ」を HL レジスタに設定する。かかる 2 行目のコマンドが、図 1 2 2 (a) のステップ S 2 に対応する。そして、3 行目のコマンド「RST BYTESEL」によって、サブルーチンとして BYTESEL モジュールが呼び出される。かかる 3 行目のコマンドが、図 1 2 2 (a) のステップ S 3 に対応する。

【1133】

図 1 2 3 (b) の 1 行目の指標「BYTESEL:」は、当該 BYTESEL モジュールの先頭アドレスを示す。2 行目のコマンド「ADDWB HL, A」によって、HL レジスタの値に A レジスタの値が加算され、HL レジスタの値が更新される。そして、3 行目のコマンド「LD A, (HL)」によって、HL レジスタで示されるアドレスに格納された 1 バイト値が A レジスタに読み出される。かかる 2、3 行目のコマンドが、図 1 2 2 (b) のステップ S 4 に対応する。4 行目のコマンド「OR A, A」によって、A レジスタの値と A レジスタの値との論理和が計算される。かかるコマンド「OR A, A」では、A レジスタの値同士の論理和なので、A レジスタの値は変化しない。一方、論理和によって、ゼロフラグ (第 1 ゼロフラグおよび第 2 ゼロフラグ) を変化させることができる。すなわち、かかる BYTESEL モジュールでは、読み出された A レジスタの値がゼロであるか否かをゼロフラグによって判定することが可能となる。かかる 4 行目のコマンドが、図 1 2 2 (b) のステップ S 5 に対応する。そして、5 行目のコマンド「RET」によって、1 段上のルーチンに戻る。かかる 5 行目のコマンドが、図 1 2 2 (b) のステップ S 6 に対応する。

【1134】

なお、図 1 2 3 (c) における指標「D__DSP__TZ」を起点としたテーブルは、例えば、特 1 大当たり図柄および特 2 大当たり図柄に関する情報を示す。例えば、変数「R__TZ1__DSP」、すなわち、特別図柄 1 表示図柄カウンタのカウンタ値が「02H」であった場合、図 1 2 4 に示すように、メイン ROM 3 0 0 b のプログラムデータに記述されたテーブル (2AH、A2H、A8H、36H…5DH、FDH) において選択アドレス (例えば、1250H) に A レジスタの値 (例えば、02H) を加算したアドレス (低い方から 3 番目のアドレス) に格納された値 (A8H) を A レジスタに読み出すことができる。このように、ここでは、同一バイト長の複数の値を連続して並置したテーブル中の所望する 1 バイト値を取得することが可能となる。

【1135】

BYTESEL モジュールは、複数のモジュールからサブルーチンとして呼び出される。例えば、図 2 3 のステップ S 1 0 0 - 5 5 で示したサブコマンドセット処理を実行する SBCMDST モジュール、図 2 6 のステップ S 4 0 0 - 3 1 で示した LED 表示設定処理を実行する LED__PRC モジュール、図 2 6 のステップ S 4 0 0 - 3 3 で示したソレノイドデータ設定処理 (ソレノイド出力イメージ合成処理) を実行する SOL__SET モジュール、図 2 6 のステップ S 4 0 0 - 3 1 で示した LED 表示設定処理のサブルーチンである保留球数ビットデータ合成処理を実行する MEM__DAT モジュール、図 2 8 のステップ S 5 3 0、S 5 4 0 で示したカウントスイッチ通過処理 (第 2 始動口通過処理、大入賞口通過処理) を実行する TDN__PAS モジュール、図 4 3 のステップ S 6 3 0 で示した特別図柄停止図柄表示処理を実行する TZ__STP モジュール、図 4 7 のステップ S 7 2 0 で示した大入賞口開放制御処理を実行する TD__OPN モジュール、図 5 2 のステップ S 8 1 0 - 1 1 で示した普通図柄変動時間設定処理 (普通図柄変動時間の決定処理) を実行する FSPNTMR モジュール、図 3 9 のステップ S 6 1 2 で示した特別図柄変動パターン決定処理 (特別図柄変動番号決定処理) を実行する THPTSEL モジュール、図 3 7 のステップ S 6 1 0 - 1 7 で示される予備領域設定処理を実行する TRSVSEL モジュール、図 4 8 のステップ S 7 3 0 - 9 で示したエンディング時間設定処理を実行する TENDTMR モジュール、図 4 3 のステップ S 6 3 0 - 1 3 (特別電動役物最大作動

回数設定処理) 中で実行される図柄オフセット取得処理(図示せず)を実行するZUGOFFFSモジュール、図33のステップS536で示した取得時演出判定処理を実行するRGETCHKモジュール等から呼び出される。

【1136】

このように、図123(b)に示したBYTESELモジュールの命令の総命令サイズは、2行目の命令「ADDWB HL, A」1バイト+3行目の命令「LD A, (HL)」1バイト+4行目の命令「OR A, A」1バイト+5行目の命令「RET」1バイト=4バイトとなり、総実行サイクルは、2行目1サイクル+3行目2サイクル+4行目1サイクル+5行目3サイクル=7サイクルとなる。かかるBYTESELモジュールを設けることによって、上述した各モジュール内でバイトデータ選択処理を行うことなく、命令サイズ1バイトの命令「RST BYTESEL」で賄うことができるので、命令の短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1137】

図125は、BYTESELモジュールを実現するための命令の他の例を説明するための説明図である。ここでは、図123(b)のBYTESELモジュールの命令群を図125(b)のBYTESELモジュールの命令群に置き換えており、他の、図123(a)のBYTESELモジュールを呼び出す任意の処理の命令群、および、図123(c)のメインROM300bのプログラムデータにおける1バイトデータ群は、図125(a)、図125(c)として、そのまま用いている。ここでは、図125(a)、図125(c)のように、図123(a)、図123(c)と実質的に等しい処理についてはその説明を省略し、図125(b)の異なる処理のみを説明する。

【1138】

図125(b)の1行目の指標「BYTESEL:」は、当該BYTESELモジュールの先頭アドレスを示す。2行目の命令「ADDALD A, (HL)」によって、HLレジスタの値にAレジスタの値が加算され、加算後のHLレジスタで示されるアドレスに格納された1バイト値がAレジスタに読み出される。かかる2行目の命令が、図122(b)のステップS4に対応する。3行目の命令「OR A, A」によって、Aレジスタの値とAレジスタの値との論理和が計算される。かかる論理和により、ゼロフラグ(第1ゼロフラグおよび第2ゼロフラグ)を変化させることができる。すなわち、BYTESELモジュールでは、読み出されたAレジスタの値がゼロであるか否かをゼロフラグによって判定することが可能となる。かかる3行目の命令が、図122(b)のステップS5に対応する。そして、4行目の命令「RET」によって、1段上のルーチンに戻る。かかる4行目の命令が、図122(b)のステップS6に対応する。

【1139】

ここで、命令「ADDALD A, (HL)」は、HLレジスタの値にAレジスタの値を加算してHLレジスタを更新するとともに、HLレジスタで示されるアドレスに格納された値をAレジスタに読み出す命令である。かかる命令の命令サイズは「1」であり、実行サイクルは「3」である。

【1140】

図125(b)のBYTESELモジュールの命令の総命令サイズは、2行目の命令「ADDALD A, (HL)」1バイト+3行目の命令「OR A, A」1バイト+4行目の命令「RET」1バイト=3バイトとなり、総実行サイクルは、2行目3サイクル+3行目1サイクル+4行目3サイクル=7サイクルとなる。したがって、図123(b)の場合と比べ、総命令サイズが1バイト削減されていることが理解できる。かかるBYTESELモジュールによって、命令の更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1141】

また、図123(b)と図125(b)とを比較して理解できるように、図123(b)において2行(2、3行目)を占有していた命令群を、図125(b)においては

1行（2行目）で表すことができ、コマンド数自体の削減、および、設計負荷の軽減を図ることが可能となる。

【1142】

図126は、BYTESELモジュールを実現するためのコマンドのさらに他の例を説明するための説明図である。ここでは、図125（a）のBYTESELモジュールを呼び出す任意の処理のコマンド群、および、図125（b）のBYTESELモジュールのコマンド群を、図126（a）の任意の処理のコマンド群に置き換えており、他の、図125（c）のメインROM300bのプログラムデータにおける1バイトデータ群は、図126（b）として、そのまま用いている。ここでは、図126（b）のように、図125（c）と実質的に等しい処理についてはその説明を省略し、図126（a）の異なる処理のみを説明する。

10

【1143】

図126（a）の1行目のコマンド「LDQ A, (LOW R_TZ1_DSP)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_TZ1_DSP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値をAレジスタに読み出す。かかる1行目のコマンドが、図122（a）のステップS1に対応する。2行目のコマンド「LD HL, D_DSP_TZ」によって、読み出し元となる1バイトデータ群の先頭アドレス「D_DSP_TZ」をHLレジスタに設定する。かかる2行目のコマンドが、図122（a）のステップS2に対応する。そして、3行目のコマンド「ADDA LD A, (HL)」によって、HLレジスタの値にAレジスタの値が加算され、加算後のHLレジスタで示されるアドレスに格納された1バイト値がAレジスタに読み出される。かかる3行目のコマンドが、図122（b）のステップS4に対応する。

20

【1144】

図126の例では、図122（a）のステップS3に対応するコマンド「RST BYTESEL」、図122（b）のステップS5に対応するコマンド「OR A, A」、および、図122（b）のステップS6に対応するコマンド「RET」が省略されている。

【1145】

図126（a）の任意の処理のコマンド群の総コマンドサイズは、1行目のコマンド「LDQ A, (LOW R_TZ1_DSP)」2バイト+2行目のコマンド「LD HL, D_DSP_TZ」3バイト+3行目のコマンド「ADDA LD A, (HL)」1バイト=6バイトとなり、総実行サイクルは、1行目2サイクル+2行目3サイクル+3行目3サイクル=8サイクルとなる。

30

【1146】

これに対し、図125（a）、図125（b）の任意の処理およびBYTESELモジュールのコマンドの総コマンドサイズは、図125（a）の1行目のコマンド「LDQ A, (LOW R_TZ1_DSP)」2バイト+2行目のコマンド「LD HL, D_DSP_TZ」3バイト+3行目のコマンド「RST BYTESEL」1バイト+図125（b）の2行目のコマンド「ADDA LD A, (HL)」1バイト+3行目のコマンド「OR A, A」1バイト+4行目のコマンド「RET」1バイト=9バイトとなり、総実行サイクルは、図125（a）の1行目2サイクル+2行目3サイクル+3行目4サイクル+図125（b）の2行目3サイクル+3行目1サイクル+4行目3サイクル=16サイクルとなる。したがって、図126の例では、図125の場合と比べ、総コマンドサイズが3バイト削減され、総実行サイクルが8サイクル削減されている。このようにBYTESELモジュール自体をコマンドとして任意の処理に埋め込むことで、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

40

【1147】

また、図125と図126とを比較して理解できるように、図125（b）において4行（1～4行目）を占有していたコマンド群を、図126においては記述する必要がなく

50

なるので、コマンド数自体の削減、および、設計負荷の軽減を図ることが可能となる。

【1148】

また、BYTESELモジュールは、コマンド「RST」の対象としての汎用モジュールであるが、かかる汎用モジュールを省略することで、貴重な汎用モジュールの領域を空け、その領域に他のモジュールを配置することができる。こうしてリソースの有効活用が可能となる。

【1149】

ただし、図126の例では、コマンド「OR A, A」を省略することで、第1ゼロフラグを変化させることができなくなる。すなわち、読み出されたAレジスタの値がゼロであるか否かを第1ゼロフラグによって判定することができなくなる。しかし、バイトデータ選択処理後にAレジスタの値がゼロか否か判定する必要がある場合、そもそも、この機能を必要としない。また、コマンド「ADDA LD A, (HL)」は、そのコマンドを実行しただけで、第2ゼロフラグを変化させることができる。したがって、Aレジスタの値がゼロか否か判定する必要がある場合、第2ゼロフラグを参照することによってAレジスタの値がゼロであるか否かを判定することが可能となる。

【1150】

なお、かかるBYTESELモジュールは、パチンコ機のみならず、スロットマシンにおける、例えば、レジスタ加算処理を実行するためのモジュールに利用される。

【1151】

〈コマンド「ADDA LDW」〉

WORDSELモジュールは、ワードデータ選択処理、すなわち、選択アドレスをオフセットさせ、そのオフセットされたアドレスに格納された2バイト値を設定するための汎用モジュールである。ここでは、WORDSELモジュールをメモリマップ上の0018Hに配置する例を挙げて説明する。

【1152】

メインCPU300aは、メインROM300bからプログラムを読み出し、読み出したプログラムを遂行し、任意の処理において、サブルーチンとしてWORDSELモジュールを呼び出し、WORDSELモジュールを遂行する。WORDSELモジュールでは、選択アドレスを所定値だけオフセットさせ、オフセットされたアドレスに格納された2バイト値を保持する。こうして、オフセットに応じた値を設定することができる。

【1153】

図127は、WORDSELモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図51のステップS800で示した普通遊技管理処理を実行するFUT_PRCモジュールの一部を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該WORDSELモジュールの説明中、第1レジスタはHLレジスタであり、第2レジスタはAレジスタである。

【1154】

メインCPU300aは、図127(a)のように、任意の処理において、オフセットをAレジスタに設定し(S1)、読み出し対象となるアドレス群の先頭アドレスをHLレジスタに設定する(S2)。そして、サブルーチンとしてWORDSELモジュールを呼び出す(S3)。

【1155】

メインCPU300aは、図127(b)のように、WORDSELモジュールにおいて、HLレジスタの値にAレジスタの値を2回加算し、もしくは、Aレジスタの値を2倍した値を加算してHLレジスタを更新し(S4)、HLレジスタで示されるアドレスに格納された2バイト値をHLレジスタに読み出す(S5)。続いて、メインCPU300aは、Lレジスタの値をAレジスタに設定する(S6)。そして、当該WORDSELモジュールを終了して1段上のルーチンに戻る(S7)。

【1156】

WORDSELモジュールによって、HLレジスタに2バイト値が設定されると、メイ

10

20

30

40

50

ンCPU300aは、そのHLレジスタの値を用いて様々な処理を遂行する。例えば、ここでは、図127(a)のように、HLレジスタが示すアドレスをサブルーチンとして呼び出している(S8)。

【1157】

図128、図129は、WORDSELモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図128(a)は、WORDSELモジュールを呼び出す任意の処理のコマンド群を示し、図128(b)は、WORDSELモジュールのコマンド群を示し、図128(c)は、メインROM300bのプログラムデータにおける2バイトデータ群の配置を示す。図127で示したフローチャートは、例えば、図128に示したプログラムによって実現される。

【1158】

図128(a)の1行目のコマンド「LDQ A, (LOW R_FUT_PHS)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_TZ1_DSP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値をAレジスタに読み出す。かかる1行目のコマンドが、図127(a)のステップS1に対応する。2行目のコマンド「LD HL, J_JMP_FUT」によって、読み出し元となる2バイトデータ群の先頭アドレス「J_JMP_FUT」をHLレジスタに設定する。かかる2行目のコマンドが、図127(a)のステップS2に対応する。そして、3行目のコマンド「RST WORDSEL」によって、サブルーチンとしてWORDSELモジュールが呼び出される。かかる3行目のコマンドが、図127(a)のステップS3に対応する。

【1159】

図128(b)の1行目の指標「WORDSEL:」は、当該WORDSELモジュールの先頭アドレスを示す。2行目のコマンド「ADDWB HL, A」によって、HLレジスタの値にAレジスタの値が加算され、HLレジスタが更新される。さらに、3行目のコマンド「ADDWB HL, A」によって、HLレジスタの値にAレジスタの値が加算され、HLレジスタが更新される。ここで、Aレジスタの値をHLレジスタに2回加算しているのは、読み出し元の値が2バイト値だからである。すなわち、テーブルに2バイト単位でデータが割り当てられているので、データを読み出すためには、アドレスを2ずつオフセットする必要があるからである。なお、ここでは、コマンド「ADDWB HL, A」を2回実行することで、HLレジスタにAレジスタの値を2回加算しているが、2行目のコマンド「ADDWB HL, A」に代え、コマンド「ADD A, A」を用い、Aレジスタの値を予め2倍してから、HLレジスタに加算することでも実現できる。かかる2、3行目のコマンドが、図127(b)のステップS4に対応する。

【1160】

そして、図128(b)の4行目のコマンド「LD HL, (HL)」によって、HLレジスタで示されるアドレスに格納された2バイト値がHLレジスタに読み出される。かかる4行目のコマンドが、図127(b)のステップS5に対応する。5行目のコマンド「LD A, L」によって、HLレジスタのうち、Lレジスタの値がAレジスタに設定される。こうして、HLレジスタの値の下位1バイトが予めAレジスタに設定される。かかる5行目のコマンドが、図127(b)のステップS6に対応する。そして、6行目のコマンド「RET」によって、1段上のルーチンに戻る。かかる6行目のコマンドが、図127(b)のステップS7に対応する。

【1161】

なお、図128(c)における指標「J_JMP_FUT」を起点としたテーブルは、例えば、普通遊技の状態を示すアドレスを示す。例えば、変数「R_FUT_PHS」、すなわち、普通遊技管理フェーズの値が「02H」であった場合、図129に示すように、メインROM300bのプログラムデータに記述されたテーブル(FZ_STA、FZ_SPN、FZ_STP、FD_PRE、FD_OPN、FD_END)において選択アドレス(例えば、1270H)にAレジスタの値を2倍した値(例えば、04H)を加算

10

20

30

40

50

したアドレス（低い方から5番目および6番目のアドレス）に格納された値、ここでは、0832H（FZ__STP）をHLレジスタに保持することができる。このように、ここでは、同一バイト長の複数の値を連続して並置したテーブル中の所望する2バイト値を取得することが可能となる。

【1162】

ここで、FZ__STAは、図52のステップS810で示した普通図柄変動待ち処理のアドレス値を示し、FZ__SPNは、図53のステップS820で示した普通図柄変動中処理のアドレス値を示し、FZ__STPは、図54のステップS830で示した普通図柄停止図柄表示処理のアドレス値を示し、FD__PREは、図55のステップS840で示した普通電動役物入賞口開放前処理のアドレス値を示し、FD__OPNは、図57のステップS850で示した普通電動役物入賞口開放制御処理のアドレス値を示し、FD__CLSは、図58のステップS860で示した普通電動役物入賞口閉鎖有効処理のアドレス値を示し、FD__ENDは、図59のステップS870で示した普通電動役物入賞口終了ウェイト処理のアドレス値を示す。

【1163】

続いて、図128（a）の4行目のコマンド「CALL（HL）」によって、取得された2バイト値、すなわち、普通遊技の状態に応じたアドレス値を呼び出す。例えば、上記のように、テーブルにおいて0832H（FZ__STP）がHLレジスタに読み出されると、かかるコマンド「CALL（HL）」によって「FZ__STP」が示すアドレスを呼び出し、普通図柄停止図柄表示処理が実行されることとなる。かかる4行目のコマンドが、図127（a）のステップS8に対応する。

【1164】

WORDSELモジュールは、複数のモジュールからサブルーチンとして呼び出される。例えば、図43のステップS630で示した特別図柄停止図柄表示処理の中で実行されるダブルバイト選択処理（図示せず）を実行するDBLBYTEモジュール、図23のステップS100-65で示したサブコマンド送信処理を実行するSBC__OUTモジュール、図26のステップS400-5で示したダイナミックポート出力処理を実行するDYM__OUTモジュール、図51のステップS800で示した普通遊技管理処理を実行するFUT__PRCモジュール、図36のステップS600で示した特別遊技管理処理を実行するTOK__PRCモジュール、図44のステップS700で特別電動役物遊技管理処理を実行するTDN__PRCモジュール、図31のステップS530で示した第2始動口通過処理を実行するSTA__PASモジュール、図34のステップS540で示したカウントスイッチ通過処理（大入賞口通過処理）を実行するTDN__PASモジュール、図32のステップS535で示した特別図柄乱数取得処理を実行するTZ__RGETモジュール、普通図柄変動時間設定処理を実行するFSPNTMRモジュール、図56のステップS841で示した普通電動役物入賞口開閉切替処理を実行するFDNCHGモジュール、図37のステップS610-9で示した特別図柄記憶エリアシフト処理を実行するTMEMSFTモジュール、図39のステップS612で示した特別図柄変動パターン決定処理を実行するTHPTSELモジュール、図37のステップS610で示した特別図柄変動待ち処理のステップS610-15で示した特別図柄変動時間設定処理を実行するTSPNTMTMモジュール、図43のステップS630-15で示したオープニング時間設定処理を実行するTSTATMRモジュール、図46のステップS711で示した大入賞口開閉切替処理を実行するTDNCHGモジュール、図48のステップS730-7で示した大入賞口閉鎖時間設定処理を実行するTCLSTMTRモジュール、図48のステップS730-9で示したエンディング時間設定処理を実行するTENDTMRモジュール、図39のS612-15で示した変動パターン選択2処理（変動パターン番号決定処理）を実行するHPT__MODモジュール、図39のS612-15で示した変動パターン選択3処理（変動パターン番号決定処理）を実行するHPT__PATモジュール、図52のステップS810-5の中で実行されるワードデータ判定処理（図示せず）を実行するWORDJDGモジュール、図33のステップS536で示した取得時演出判定処理を実行するRG

ETCHKモジュール、図47のステップS720-9で示した大入賞口閉鎖有効時間設定処理のサブルーチンである大入賞口閉鎖有効時間選択処理を実行するTEF__SELモジュール等から呼び出される。

【1165】

このように、図128(b)に示したWORDSELモジュールの命令の総命令サイズは、2行目の命令「ADDWB HL, A」1バイト+3行目の命令「ADDWB HL, A」1バイト+4行目の命令「LD HL, (HL)」2バイト+5行目の命令「LD A, L」1バイト+6行目の命令「RET」1バイト=6バイトとなり、総実行サイクルは、2行目1サイクル+3行目1サイクル+4行目4サイクル+5行目1サイクル+6行目3サイクル=10サイクルとなる。かかるWORDSELモジュールを設けることによって、上述した各モジュール内でワードデータ選択処理を行うことなく、命令サイズ1バイトの命令「RST WORDSEL」で賄うことができるので、命令の短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1166】

図130は、WORDSELモジュールを実現するための命令の他の例を説明するための説明図である。ここでは、図128(b)のWORDSELモジュールの命令群を図130(b)のWORDSELモジュールの命令群に置き換えており、他の、図128(a)のWORDSELモジュールを呼び出す任意の処理の命令群、および、図128(c)のメインROM300bのプログラムデータにおける2バイトデータ群は、図130(a)、図130(c)として、そのまま用いている。ここでは、図130(a)、図130(c)のように、図128(a)、図128(c)と実質的に等しい処理についてはその説明を省略し、図130(b)の異なる処理のみを説明する。

20

【1167】

図130(b)の1行目の指標「WORDSEL:」は、当該WORDSELモジュールの先頭アドレスを示す。2行目の命令「ADDALDW HL, (HL)」によって、HLレジスタの値にAレジスタの値が2回加算され、加算後のHLレジスタで示されるアドレスに格納された2バイト値がHLレジスタに読み出される。かかる2行目の命令が、図127(b)のステップS4、S5に対応する。3行目の命令「LD A, L」によって、HLレジスタのうち、Lレジスタの値がAレジスタに設定される。こうして、HLレジスタの値の下位1バイトが予めAレジスタに設定される。かかる3行目の命令が、図127(b)のステップS6に対応する。そして、4行目の命令「RET」によって、1段上のルーチンに戻る。かかる6行目の命令が、図127(b)のステップS7に対応する。

30

【1168】

ここで、命令「ADDALDW HL, (HL)」は、HLレジスタの値にAレジスタの値が2回加算されHLレジスタを更新するとともに、加算後のHLレジスタで示されるアドレスに格納された値をHLレジスタに読み出す命令である。かかる命令の命令サイズは「2」であり、実行サイクルは「6」である。

【1169】

続いて、図130(a)の4行目の命令「CALL (HL)」によって、取得された2バイト値、すなわち、普通遊技の状態に応じたアドレス値を呼び出す。かかる4行目の命令が、図127(a)のステップS8に対応する。

40

【1170】

図130(b)のWORDSELモジュールの命令の総命令サイズは、2行目の命令「ADDALDW HL, (HL)」2バイト+3行目の命令「LD A, L」1バイト+4行目の命令「RET」1バイト=4バイトとなり、総実行サイクルは、2行目6サイクル+3行目1サイクル+4行目3サイクル=10サイクルとなる。したがって、図128(b)の場合と比べ、総命令サイズが2バイト削減されている。かかるWORDSELモジュールによって、命令の更なる短縮化を図り、遊技制御処

50

理を行うための制御領域の容量を確保することが可能となる。

【1171】

また、図128(b)と図130(b)とを比較して理解できるように、図128(b)において3行(2、3、4行目)を占有していたコマンド群を、図130(b)においては1行(2行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1172】

図131は、WORDSELモジュールを実現するためのコマンドのさらに他の例を説明するための説明図である。ここでは、図130(a)のWORDSELモジュールを呼び出す任意の処理のコマンド群、および、図130(b)のWORDSELモジュールのコマンド群を、図131(a)の任意の処理のコマンド群に置き換えており、他の、図130(c)のメインROM300bのプログラムデータにおける2バイトデータ群は、図131(b)として、そのまま用いている。ここでは、図131(b)のように、図130(c)と実質的に等しい処理についてはその説明を省略し、図131(a)の異なる処理のみを説明する。

【1173】

図131(a)の1行目のコマンド「LDQ A, (LOW R_FUT_PHS)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FUT_PHS」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値をAレジスタに読み出す。かかる1行目のコマンドが、図127(a)のステップS1に対応する。2行目のコマンド「LD HL, J_JMP_FUT」によって、読み出し元となる1バイトデータ群の先頭アドレス「J_JMP_FUT」をHLレジスタに設定する。かかる2行目のコマンドが、図127(a)のステップS2に対応する。そして、3行目のコマンド「ADDALDW HL, (HL)」によって、HLレジスタの値にAレジスタの値が2回加算され、加算後のHLレジスタで示されるアドレスに格納された2バイト値がHLレジスタに読み出される。かかる3行目のコマンドが、図127(b)のステップS4、5に対応する。4行目のコマンド「CALL (HL)」によって、取得された2バイト値、すなわち、普通遊技の状態に応じたアドレス値を呼び出す。かかる4行目のコマンドが、図127(a)のステップS8に対応する。

【1174】

図131の例では、図127(a)のステップS3に対応するコマンド「RST WORDSEL」、図127(b)のステップS6に対応するコマンド「LD A, L」、および、図127(b)のステップS7に対応するコマンド「RET」が省略されている。

【1175】

図131(a)の任意の処理およびWORDSELモジュールのコマンドの総コマンドサイズは、1行目のコマンド「LDQ A, (LOW R_FUT_PHS)」2バイト+2行目のコマンド「LD HL, J_JMP_FUT」3バイト+3行目のコマンド「ADDALDW HL, (HL)」2バイト+4行目のコマンド「CALL (HL)」2バイト=9バイトとなり、総実行サイクルは、1行目2サイクル+2行目3サイクル+3行目6サイクル+4行目5サイクル=16サイクルとなる。

【1176】

これに対し、図130(a)、図130(b)の任意の処理およびWORDSELモジュールのコマンドの総コマンドサイズは、図130(a)の1行目のコマンド「LDQ A, (LOW R_FUT_PHS)」2バイト+2行目のコマンド「LD HL, J_JMP_FUT」3バイト+3行目のコマンド「RST WORDSEL」1バイト+4行目のコマンド「CALL (HL)」2バイト+図130(b)の2行目のコマンド「ADDALDW HL, (HL)」2バイト+3行目のコマンド「LD A, L」1バイト+4行目のコマンド「RET」1バイト=12バイトとなり、総実行サイクルは、図130(a)の1行目2サイクル+2行目3サイクル+3行目4サイクル+4行目5サイクル+

図130(b)の2行目6サイクル+3行目1サイクル+4行目3サイクル=24サイクルとなる。したがって、図131の例では、図130の場合と比べ、総コマンドサイズが3バイト削減され、総実行サイクルが8サイクル削減されている。このようにWORDSELモジュール自体をコマンドとして任意の処理に埋め込むことで、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1177】

また、図130と図131とを比較して理解できるように、図130(b)において4行(1~4行目)を占有していたコマンド群を、図131においては記述する必要がなくなるので、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

10

【1178】

また、WORDSELモジュールは、コマンド「RST」の対象としての汎用モジュールであるが、かかる汎用モジュールを省略することで、貴重な汎用モジュールの領域を空け、その領域に他のモジュールを配置することができる。こうしてリソースの有効活用が可能となる。

【1179】

ただし、図131の例では、コマンド「LD A, L」を省略している。したがって、Lレジスタの値をAレジスタに設定する必要がある場合、別途、コマンド「LD A, L」を追加する必要がある。しかし、コマンド「LD A, L」は、コマンドサイズが「1」であり、実行サイクルが「1」なので、かかるコマンドの追加の影響は小さい。

20

【1180】

なお、かかるWORDSELモジュールは、パチンコ機のみならず、スロットマシンにおける、例えば、図98のステップS3100-21で示した状態別モジュール実行処理を実行するためのCAL_MODモジュールや、そのCAL_MODモジュールから移行し、本前兆開始判定処理を実行するためのHID_JUGモジュールに利用される。

【1181】

図132は、CAL_MODモジュールを説明するための説明図である。図132に示したCAL_MODモジュールは、状態別モジュール実行処理、すなわち、状態に対応したモジュールを実行する。

【1182】

かかるCAL_MODモジュールを実行する段階では、既に、オフセット値が所定のレジスタ(例えば、Aレジスタ)に保持されている。図132(a)の1行目の指標「CAL_MOD:」は、当該CAL_MODモジュールの先頭アドレスを示す。2行目のコマンド「LD HL, T_INT_PHASE」によって、読み出し元となる2バイトデータ群の先頭アドレス「T_INT_PHASE」がHLレジスタに設定される。3行目のコマンド「ADD A, A」によって、Aレジスタの値にAレジスタの値が加算され、Aレジスタの値が2倍になる。ここで、Aレジスタの値を2倍にしているのは、読み出し元の値が2バイト値だからである。すなわち、テーブルに2バイト単位でデータが割り当てられているので、データを読み出すためには、アドレスを2ずつオフセットする必要があるからである。4行目のコマンド「ADDWB HL, A」によって、HLレジスタの値にAレジスタの値が加算され、HLレジスタが更新される。

40

【1183】

そして、図132(a)の5行目のコマンド「LD HL, (HL)」によって、HLレジスタで示されるアドレスに格納された2バイト値がHLレジスタに読み出される。6行目のコマンド「JP (HL)」によって、HLアドレスで示されるアドレスに移動する。

【1184】

ここで、コマンド「ADDALDW」への置き換えを行うと、図132(a)のコマンド群を図132(b)のように変更することができる。図132(b)の1行目の指標「CAL_MOD:」は、当該CAL_MODモジュールの先頭アドレスを示す。2行目の

50

コマンド「LD HL, T_INT_PHASE」によって、読み出し元となる2バイトデータ群の先頭アドレス「T_INT_PHASE」がHLレジスタに設定される。3行目のコマンド「ADDALDW HL, (HL)」によって、HLレジスタの値にAレジスタの値が2回加算され、加算後のHLレジスタで示されるアドレスに格納された2バイト値がHLレジスタに読み出される。そして、4行目のコマンド「JP (HL)」によって、HLアドレスで示されるアドレスに移動する。

【1185】

ここで、図132(a)に示したCAL_MODモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD HL, T_INT_PHASE」3バイト+3行目のコマンド「ADD A, A」1バイト+4行目のコマンド「ADDWB HL, A」1バイト+5行目のコマンド「LD HL, (HL)」2バイト+6行目のコマンド「JP (HL)」2バイト=9バイトとなり、総実行サイクルは、2行目3サイクル+3行目1サイクル+4行目1サイクル+5行目4サイクル+6行目2サイクル=11サイクルとなる。一方、図132(b)に示したCAL_MODモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD HL, T_INT_PHASE」3バイト+3行目のコマンド「ADDALDW HL, (HL)」2バイト+4行目のコマンド「JP (HL)」2バイト=7バイトとなり、総実行サイクルは、2行目3サイクル+3行目6サイクル+4行目2サイクル=11サイクルとなる。したがって、図132(a)のコマンド群を図132(b)のコマンド群に置き換えることで、総コマンドサイズが2バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1186】

図133は、HID_JUGモジュールを説明するための説明図である。図133に示したHID_JUGモジュールは、本前兆開始判定処理を実行する。

【1187】

図133(a)の1行目の指標「HID_JUG:」は、当該HID_JUGモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW_NMD_KND)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「_NMD_KND」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(通常モード種別)をAレジスタに読み出す。3行目のコマンド「LD HL, T_DAT_PGM-2」によって、読み出し元となる2バイトデータ群の先頭アドレス(「T_DAT_PGM」から2を減算した値)がHLレジスタに設定される。4行目のコマンド「ADD A, A」によって、Aレジスタの値にAレジスタの値が加算され、Aレジスタの値が2倍になる。ここで、Aレジスタの値を2倍にしているのは、読み出し元の値が2バイト値だからである。すなわち、テーブルに2バイト単位でデータが割り当てられているので、データを読み出すためには、アドレスを2ずつオフセットする必要があるからである。5行目のコマンド「ADDWB HL, A」によって、HLレジスタの値にAレジスタの値が加算され、HLレジスタが更新される。そして、図133(a)の6行目のコマンド「LD HL, (HL)」によって、HLレジスタで示されるアドレスに格納された2バイト値がHLレジスタに読み出される。

【1188】

ここで、コマンド「ADDALDW」への置き換えを行うと、図133(a)のコマンド群を図133(b)のように変更することができる。図133(b)の1行目の指標「HID_JUG:」は、当該HID_JUGモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW_NMD_KND)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「_NMD_KND」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(通常モード種別)をAレジスタに読み出す。3行目のコマンド「LD HL, T_DAT_PGM-2」によって、読み出し元となる2バイトデータ群の先頭アドレス(「T_DAT_PGM」から2を減算した値)がHLレジスタに設定される。4行目のコマンド「ADDALDW HL, (HL)」

によって、HLレジスタの値にAレジスタの値が2回加算され、加算後のHLレジスタで示されるアドレスに格納された2バイト値がHLレジスタに読み出される。

【1189】

ここで、図133(a)に示したHID__JUGモジュールの命令の総命令サイズは、2行目の命令「LDQ A, (LOW __NMD__KND)」2バイト+3行目の命令「LD HL, T__DAT__PGM-2」3バイト+4行目の命令「ADD A, A」1バイト+5行目の命令「ADDWB HL, A」1バイト+6行目の命令「LD HL, (HL)」2バイト=9バイトとなり、総実行サイクルは、2行目2サイクル+3行目3サイクル+4行目1サイクル+5行目1サイクル+6行目4サイクル=11サイクルとなる。一方、図133(b)に示したHID__JUGモジュールの命令の総命令サイズは、2行目の命令「LDQ A, (LOW __NMD__KND)」2バイト+3行目の命令「LD HL, T__DAT__PGM-2」3バイト+4行目の命令「ADDALDW HL, (HL)」2バイト=7バイトとなり、総実行サイクルは、2行目2サイクル+3行目3サイクル+4行目6サイクル=11サイクルとなる。したがって、図133(a)の命令群を図133(b)の命令群に置き換えることで、総命令サイズが2バイト削減される。かかる置き換えによって、命令の更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1190】

〈命令「INLD」、「INLDTQR」〉

20

RAMSETモジュールは、ラムセット処理、すなわち、メインRAM300cの変数に所定の値(初期値)を設定するための汎用モジュールである。ここでは、RAMSETモジュールをメモリマップ上の0020Hに配置する例を挙げて説明する。

【1191】

メインCPU300aは、メインROM300bからプログラムを読み出し、読み出したプログラムを遂行し、任意の処理において、サブルーチンとしてRAMSETモジュールを呼び出し、RAMSETモジュールを遂行する。RAMSETモジュールでは、メインROM300bのプログラムデータに記述された複数の1バイト値を、メインRAM300cのワークエリアにおいて変数として扱われる複数のデータを保持する領域に転送する。こうして、複数の変数の値が設定される。ここで、転送するデータの数を単にデータ数と言う。

30

【1192】

図134は、RAMSETモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図22のステップS100で示したCPU初期化処理を実行するCPUINITモジュールの一部を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該RAMSETモジュールの説明中、第1レジスタはBレジスタであり、第2レジスタはHLレジスタであり、第1値は「1」であり、第2値は「2」である。なお、第1値および第2値を任意に設定できるのは言うまでもない。

【1193】

40

メインCPU300aは、図134(a)のように、任意の処理において、転送元となる1バイトデータ群の先頭アドレスをHLレジスタに設定する(S1)。そして、サブルーチンとしてRAMSETモジュールを呼び出す(S2)。

【1194】

メインCPU300aは、図134(b)のように、RAMSETモジュールにおいて、HLレジスタで示されるアドレスに格納された1バイト値をBレジスタに読み出す(S3)。かかる1バイト値はデータ数を示す。続いて、メインCPU300aは、HLレジスタに「1」を加えた値で示されるアドレスに格納された値で特定されるアドレスに、HLレジスタに「2」を加えた値で示されるアドレスに格納された値を転送し、HLレジスタの値に「2」を加えて次に転送するアドレスを設定する(S4)。続いて、メインCP

50

U300aは、Bレジスタの値をデクリメント（「1」減算）し、デクリメントした結果が0であるか否か判定する（S5）。ここで、デクリメントした結果が0でなければ（S5におけるNO）、ステップS4からの処理を繰り返し、デクリメントした結果が0であれば（S5におけるYES）、当該RAMSETモジュールを終了して1段上のルーチンに戻る（S6）。

【1195】

図135、図136は、RAMSETモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図135（a）は、RAMSETモジュールを呼び出す任意の処理のコマンド群を示し、図135（b）は、RAMSETモジュールのコマンド群を示し、図135（c）は、メインROM300bのプログラムデータにおける1バイトデータ群の配置を示し、図135（d）は、メインRAM300cのワークエリアにおける1バイトデータ群の配置を示す。図134で示したフローチャートは、例えば、図135に示したプログラムによって実現される。

【1196】

図135（a）の1行目のコマンド「LD HL, D_RAM_RST_INI」によって、転送元となる1バイトデータ群の先頭アドレス「D_RAM_RST_INI」をHLレジスタに設定する。かかる1行目のコマンドが、図134（a）のステップS1に対応する。そして、2行目のコマンド「RST RAMSET」によって、サブルーチンとしてRAMSETモジュールが呼び出される。かかる2行目のコマンドが、図134（a）のステップS2に対応する。

【1197】

図135（b）の1行目の指標「RAMSET:」は、当該RAMSETモジュールの先頭アドレスを示す。2行目のコマンド「LD B, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値がBレジスタに読み出される。このとき、HLレジスタには、図135（a）で示したように、アドレス「D_RAM_RST_INI」が設定されている。したがって、図135（c）の2行目の1バイト値「(D_RAM_RST_INI - D_RAM_RST_INI - 1) / 2」がデータ数（転送の繰り返し数）としてBレジスタに読み出されることとなる。なお、「D_RAM_RST_INI」は、転送元となる1バイトデータ群の最終アドレスの次のアドレスであり、D_RAM_RST_INIは転送元となる1バイトデータ群の先頭アドレスなので、D_RAM_RST_INI - D_RAM_RST_INI - 1の値（データ数を示す1バイト値）の値が、1バイト値の総バイト数となり、その値を、アドレスと値との組み合わせ数である2で除算すると、データ数が導出される。図135（c）の例では、データ数は、(11 - 1) / 2 = 5となる。かかる図135（c）の2行目のコマンドが、図134（b）のステップS3に対応する。

【1198】

図135（b）の3行目の指標「RAMSET_10:」は繰り返し処理の先頭アドレスを示す。4行目のコマンド「INLD AC, (HL)」および5行目のコマンド「LDQ (C), A」によって、HLレジスタに「1」を加えた値で示されるアドレスに格納された値で特定されるアドレスに、HLレジスタに「2」を加えた値で示されるアドレスに格納された値が格納され、HLレジスタの値に「2」が加えられて次に転送するアドレスが設定される。かかる4、5行目のコマンドが、図134（b）のステップS4に対応する。

【1199】

ここで、コマンド「INLD AC, (HL)」は、HLレジスタに「1」を加えた値で示されるアドレスに格納された値をCレジスタに格納し、HLレジスタに「2」を加えた値で示されるアドレスに格納された値をAレジスタに格納し、HLレジスタの値に「2」を加えてHLレジスタの値を更新するコマンドである。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「4」である。

【1200】

10

20

30

40

50

例えば、HLレジスタがアドレス「D__RAM__RST__INI」の値であった場合、図135(c)の3行目における2バイト変数「R__TZ1__STP」の下位1バイト値がCレジスタに格納され、図135(c)の3行目における「@DSP__HAZ」の値がAレジスタに格納される。

【1201】

また、図135(b)の5行目のコマンド「LDQ (C), A」は、Qレジスタの値をアドレスの上位1バイトとし、Cレジスタの値をアドレスの下位1バイトとし、そのアドレスに、Aレジスタの値を格納するコマンドである。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「3」である。

【1202】

例えば、Cレジスタの値、すなわち「R__TZ1__STP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに、Aレジスタの値、すなわち、「@DSP__HAZ」の値を格納する。

【1203】

ここで、図135(c)、図135(d)において、変数「R__TZ1__STP」は、特別図柄1停止図柄番号を示し、変数「R__TZ2__STP」は、特別図柄2停止図柄番号を示し、変数「R__TZ1__DSP」は、特別図柄1表示図柄カウンタのカウント値を示し、変数「R__TZ2__DSP」は、特別図柄2表示図柄カウンタのカウント値を示し、変数「R__EXT__SEC__TMR」はセキュリティ外部信号タイマ値を示す。なお、図135(d)に示す変数の配置は図のように連続している必要はなく、離隔していてもよい。

【1204】

続いて、図135(b)の6行目のコマンド「DJNZ RAMSET__10」によって、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0でなければ、アドレス「RAMSET__10」に移動し、デクリメントした結果が0であれば、当該コマンドの次のコマンドに処理を移す。ここでは、データ数が「5」なので、「RAMSET__10」からの処理を5回繰り返すとBレジスタの値が0となる。かかる6行目のコマンドが、図134(b)のステップS5に対応する。そして、7行目のコマンド「RET」によって、1段上のルーチンに戻る。かかる7行目のコマンドが、図134(b)のステップS6に対応する。

【1205】

こうして、図136に示すように、メインROM300bのプログラムデータに記述された複数の1バイト値（55H、77H、66H、22H、AAH）を、メインRAM300cのワークエリアにおける変数（「R__TZ1__STP」、「R__TZ2__STP」、「R__TZ1__DSP」、「R__TZ2__DSP」、「R__EXT__SEC__TMR」）に転送することが可能となる。

【1206】

RAMSETモジュールは、複数のモジュールからサブルーチンとして呼び出される。例えば、図22のステップS100で示したCPU初期化処理を実行するCPUINITモジュール、セキュリティ設定処理を実行するSEC__SETモジュール、図26のステップS400-21で示した状態管理処理（エラー管理処理）を実行するSTC__PRCモジュール、図52のステップS810で示した普通図柄変動待ち処理を実行するFZ__STAモジュール、図53のステップS820で示した普通図柄変動中処理を実行するFZ__SPNモジュール、図54のステップS830で示した普通図柄停止図柄表示処理を実行するFZ__STPモジュール、図55のステップS840で示した普通電動役物入賞口開放前処理を実行するFD__PREモジュール、図57のステップS850で示した普通電動役物入賞口開放制御処理を実行するFD__OPNモジュール、図37のステップS610で示した特別図柄変動待ち処理を実行するTZ__STAモジュール、図41のステップS620で示した特別図柄変動中処理を実行するTZ__SPNモジュール、図43のステップS630で示した特別図柄停止図柄表示処理を実行するTZ__STPモジュール

10

20

30

40

50

、図45のステップS710で示した大入賞口開放前処理を実行するTD__PREモジュール、図47のステップS720で示した大入賞口開放制御処理を実行するTD__OPNモジュール、図48のステップS730で示した大入賞口閉鎖有効処理を実行するTD__EFFモジュール、図49のステップS740で示した大入賞口終了ウェイト処理を実行するTD__ENDモジュール、図43のステップS630の中で実行される特別図柄変動待ち状態移行処理（図示せず）を実行するTOKZEROモジュール、図40のステップS613で示した回数切り管理処理を実行するCHGSTSモジュール、図27のステップS450で示した設定関連処理を実行するRNK__PRCモジュール等から呼び出される。
【1207】

このように、図135（b）に示したRAMSETモジュールの命令の総命令サイズは、2行目の命令「LD B, (HL)」1バイト+4行目の命令「IN LD AC, (HL)」2バイト+5行目の命令「LDQ (C), A」2バイト+6行目の命令「DJNZ RAMSET__10」2バイト+7行目の命令「RET」1バイト=8バイトとなり、総実行サイクルは、2行目2サイクル+4行目4サイクル+5行目3サイクル+6行目3サイクル（または2サイクル）+7行目3サイクル=15サイクル（または14サイクル）となる。なお、括弧内のサイクル数は、命令「DJNZ RAMSET__10」によって移動しなかった場合の実行サイクルを示している。かかるRAMSETモジュールを設けることによって、上述した各モジュール内でラムセット処理を行うことなく、命令サイズ1バイトの命令「RST RAMSET」で賄うことができるので、命令の短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1208】

図137は、RAMSETモジュールを実現するための命令の他の例を説明するための説明図である。ここでは、図135（b）のRAMSETモジュールの命令群を図137（b）のRAMSETモジュールの命令群に置き換えており、他の、図135（a）のRAMSETモジュールを呼び出す任意の処理の命令群、図135（c）のメインROM300bのプログラムデータにおける1バイトデータ群、および、図135（d）のメインRAM300cのワークエリアにおける1バイトデータ群は、図137（a）、図137（c）、図137（d）として、そのまま用いている。ここでは、図137（a）、図137（c）、図137（d）のように、図135（a）、図135（c）、図135（d）と実質的に等しい処理についてはその説明を省略し、図137（b）の異なる処理のみを説明する。

【1209】

図137（b）の1行目の指標「RAMSET:」は、当該RAMSETモジュールの先頭アドレスを示す。2行目の命令「LD B, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値がBレジスタに読み出される。このとき、HLレジスタには、図137（a）で示したように、アドレス「D__RAM__RST__INI」が設定されている。したがって、図137（c）の2行目の1バイト値「(@D__RAM__RST__INI-D__RAM__RST__INI-1)/2」がデータ数としてBレジスタに読み出される。かかる図137（b）の2行目の命令が、図134（b）のステップS3に対応する。

【1210】

図137（b）の3行目の命令「INLDTQR AC, (HL)」によって、HLレジスタに「1」を加えた値で示されるアドレスに格納された値で特定されるアドレスに、HLレジスタに「2」を加えた値で示されるアドレスに格納された値が転送され、HLレジスタの値に「2」を加えて次に転送するアドレスが設定され、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0になるまで、当該処理が繰り返される。かかる3行目の命令が、図134（b）のステップS4、S5に対応する。

【1211】

ここで、命令「INLDTQR AC, (HL)」は、Qレジスタの値をアドレス

10

20

30

40

50

の上位1バイトとし、HLレジスタに「1」を加えた値で示されるアドレスに格納された値をアドレスの下位1バイトとし、そのアドレスに、HLレジスタに「2」を加えた値で示されるアドレスに格納された値を転送し、HLレジスタの値に「2」を加えてHLレジスタの値を更新し、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0になるまで、値の転送を繰り返すコマンドである。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「5」である。

【1212】

例えば、HLレジスタがアドレス「D__RAM__RST__INI」の値であった場合、Qレジスタの値をアドレスの上位1バイトとし、図137(c)の3行目における「R__TZ1__STP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに、図137(c)の3行目における「@DSP__HAZ」の値が転送され、HLレジスタの値に「2」が加えられ、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0になるまで、すなわち、5回転送が繰り返される。

【1213】

そして、図137(b)の4行目のコマンド「RET」によって、1段上のルーチンに戻る。かかる4行目のコマンドが、図134(b)のステップS6に対応する。

【1214】

図137(b)のRAMSETモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD B, (HL)」1バイト+3行目のコマンド「INLDTQR AC, (HL)」2バイト+4行目のコマンド「RET」1バイト=4バイトとなり、総実行サイクルは、2行目2サイクル+3行目5サイクル+4行目3サイクル=10サイクルとなる。したがって、図135(b)の場合と比べ、総コマンドサイズが4バイト削減され、総実行サイクルも少なくとも4サイクル削減される。特に、図135(b)の例では、コマンド「DJNZ RAMSET__10」におけるBレジスタの値が2以上であれば、その値に10サイクル（4行目4サイクル+5行目3サイクル+6行目3サイクル）を乗じた分だけ総実行サイクルが増えるので、図137(b)の総実行サイクルの削減量も多くなる。かかるRAMSETモジュールによって、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1215】

また、ここでは、図135の例と比較して、AレジスタおよびCレジスタを利用していない。したがって、Aレジスタの値やCレジスタの値を意図せず更新してしまうことはない。また、図135の例では、AレジスタやCレジスタが既に利用されている場合、スタックしてAレジスタやCレジスタの値を退避する必要があったが、ここでは、AレジスタやCレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1216】

また、図135(b)と図137(b)とを比較して理解できるように、図135(b)において4行（3行目～6行目）を占有していたコマンド群を、図137(b)においては1行（3行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1217】

なお、かかるRAMSETモジュールは、パチンコ機のみならず、スロットマシンにおいても、メインRAM500cの変数に所定の値（初期値）を設定するためのテーブルセット処理を実行するTABLSETモジュール（汎用モジュール）として利用されている。

【1218】

図138は、TABLSETモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図138(a)は、TABLSETモジュールを呼び出す任意の処理のコマンド群を示し、図138(b)は、TABLSETモジュールのコマンド群を示し、図138(c)は、メインROM500bのプログラムデータにおける1バイトデータ群の配置を示し、図138(d)は、メインRAM500cのワークエリアに

おける1バイトデータ群の配置を示す。

【1219】

なお、ここでは、任意の処理として、図89のステップS2200-11や図95のステップS2800-39で示したエラーウェイト処理、すなわち、エラー表示、警告音要求およびエラー復帰待ちを実行するERRWAITモジュールの一部を挙げている。

【1220】

図138(a)の1行目のコマンド「LD HL, T__ERR__RCV」によって、転送元となる1バイトデータ群の先頭アドレス「T__ERR__RCV」をHLレジスタに設定する。そして、2行目のコマンド「RST TABLSET」によって、サブルーチンとしてTABLSETモジュールが呼び出される。

【1221】

図138(b)の1行目の指標「TABLSET:」は、当該TABLSETモジュールの先頭アドレスを示す。2行目のコマンド「PUSH BC」によって、BCレジスタの値がスタック領域に退避される。3行目のコマンド「DI」によって割込が禁止される。

【1222】

図138(b)の4行目のコマンド「LD B, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値がBレジスタに読み出される。このとき、HLレジスタには、図138(a)で示したように、アドレス「T__ERR__RCV」が設定されている。したがって、図138(c)の2行目の1バイト値「(T__ERR__RCV - T__ERR__RCV) / 2」がデータ数(転送の繰り返し数)としてBレジスタに読み出されることとなる。なお、「T__ERR__RCV」は、転送元となる1バイトデータ群の最終アドレスの次のアドレスであり、T__ERR__RCVは転送元となる1バイトデータ群の先頭アドレスなので、T__ERR__RCV - T__ERR__RCVの値(データ数を示す1バイト値)の値が、1バイト値の総バイト数となり、その値を、アドレスと値との組み合わせ数である2で除算すると、データ数が導出される。図138(c)の例では、データ数は、 $9 / 2 = 4$ となる。

【1223】

図138(b)の5行目の指標「TABLSET10:」は繰り返し処理の先頭アドレスを示す。6行目のコマンド「INLD AC, (HL)」および7行目のコマンド「LDQ (C), A」によって、HLレジスタに「1」を加えた値で示されるアドレスに格納された値で特定されるアドレスに、HLレジスタに「2」を加えた値で示されるアドレスに格納された値が格納され、HLレジスタの値に「2」が加えられて次に転送するアドレスが設定される。

【1224】

例えば、HLレジスタがアドレス「T__ERR__RCV」の値であった場合、図138(c)の3行目における2バイト変数「__ERR__NUM」の下位1バイト値がCレジスタに格納され、図138(c)の3行目における「0」の値がAレジスタに格納される。

【1225】

また、図138(b)の7行目のコマンド「LDQ (C), A」は、Qレジスタの値をアドレスの上位1バイトとし、Cレジスタの値をアドレスの下位1バイトとし、そのアドレスに、Aレジスタの値を格納するコマンドである。

【1226】

例えば、Cレジスタの値、すなわち「__ERR__NUM」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに、Aレジスタの値、すなわち、「0」の値を格納する。

【1227】

ここで、図138(c)、図138(d)において、変数「__ERR__NUM」は、エラー番号を示し、変数「__CRE__TMR」は、クレジットボタン検出タイマを示し、変数「__CRE__FLG」は、クレジットボタン検出フラグを示し、変数「__SNS__OLD」はメダル通過センサービット前回状態を示す。なお、図138(d)に示す変数の配

10

20

30

40

50

置は図のように連続している必要はなく、離隔していてもよい。

【1228】

続いて、図138(b)の8行目のコマンド「DJNZ TABLSET10」によって、Bレジスタの値がデクリメント(「1」減算)され、デクリメントした結果が0でなければ、アドレス「TABLSET10」に移動し、デクリメントした結果が0であれば、当該コマンドの次のコマンドに処理を移す。ここでは、データ数が「4」なので、「TABLSET10」からの処理を4回繰り返すとBレジスタの値が0となる。

【1229】

図138(b)の9行目のコマンド「EI」によって割込が許可される。10行目のコマンド「POP BC」によってスタック領域に退避されていたデータがBCレジスタに復帰される。そして、11行目のコマンド「RET」によって、1段上のルーチンに戻る。

【1230】

ここで、コマンド「INLDTQR」への置き換えを行うと、コマンド群を以下のように記述できる。

【1231】

図139は、TABLSETモジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図138(b)のTABLSETモジュールのコマンド群を、図139(b)のTABLSETモジュールのコマンド群に置き換えており、他の、図138(a)のTABLSETモジュールを呼び出す任意の処理のコマンド群、図138(c)のメインROM500bのプログラムデータにおける1バイトデータ群、および、図138(d)のメインRAM500cのワークエリアにおける1バイトデータ群は、図139(a)、図139(c)、図139(d)として、そのまま用いている。ここでは、図139(a)、図139(c)、図139(d)のように、図138(a)、図138(c)、図138(d)と実質的に等しい処理についてはその説明を省略し、図139(b)の異なる処理のみを説明する。

【1232】

図139(b)の1行目の指標「TABLSET:」は、当該TABLSETモジュールの先頭アドレスを示す。2行目のコマンド「PUSH BC」によって、BCレジスタの値がスタック領域に退避される。3行目のコマンド「DI」によって割込が禁止される。

【1233】

図139(b)の4行目のコマンド「LD B, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値がBレジスタに読み出される。このとき、HLレジスタには、図139(a)で示したように、アドレス「T__ERR__RCV」が設定されている。したがって、図139(c)の2行目の1バイト値「(T__ERR__RCV - T__ERR__RCV) / 2」がデータ数(転送の繰り返し数)としてBレジスタに読み出されることとなる。図139(c)の例では、データ数は4となる。

【1234】

図139(b)の5行目のコマンド「INLDTQR (HL)」によって、HLレジスタに「1」を加えた値で示されるアドレスに格納された値で特定されるアドレスに、HLレジスタに「2」を加えた値で示されるアドレスに格納された値が転送され、HLレジスタの値に「2」を加えて次に転送するアドレスが設定され、Bレジスタの値がデクリメント(「1」減算)され、デクリメントした結果が0になるまで、当該処理が繰り返される。

【1235】

続いて、図139(b)の6行目のコマンド「EI」によって割込が許可される。7行目のコマンド「POP BC」によってスタック領域に退避されていたデータがBCレジスタに復帰される。そして、8行目のコマンド「RET」によって、1段上のルーチンに戻る。

【1236】

ここで、図138(b)に示したTABLSETモジュールのコマンドの総コマンドサ

10

20

30

40

50

イズは、2行目のコマンド「PUSH BC」1バイト+3行目のコマンド「DI」1バイト+4行目のコマンド「LD B, (HL)」1バイト+6行目のコマンド「INLD AC, (HL)」2バイト+7行目のコマンド「LDQ (C), A」2バイト+8行目のコマンド「DJNZ TABLSET10」2バイト+9行目のコマンド「EI」1バイト+10行目のコマンド「POP BC」1バイト+11行目のコマンド「RET」1バイト=12バイトとなり、総実行サイクルは、2行目3サイクル+3行目1サイクル+4行目2サイクル+6行目4サイクル+7行目3サイクル+8行目3サイクル（または2サイクル）+9行目1サイクル+10行目3サイクル+11行目3サイクル=23サイクル（または22サイクル）となる。一方、図139（b）に示したTABLSETモジュールのコマンドの総コマンドサイズは、2行目のコマンド「PUSH BC」1バイト+3行目のコマンド「DI」1バイト+4行目のコマンド「LD B, (HL)」1バイト+5行目のコマンド「INLDTQR (HL)」2バイト+6行目のコマンド「EI」1バイト+7行目のコマンド「POP BC」1バイト+8行目のコマンド「RET」1バイト=8バイトとなり、総実行サイクルは、2行目3サイクル+3行目1サイクル+4行目2サイクル+5行目5サイクル+6行目1サイクル+7行目3サイクル+8行目3サイクル=18サイクルとなる。したがって、図138（b）の場合と比べ、総コマンドサイズが4バイト削減され、総実行サイクルも少なくとも5サイクル削減される。特に、図138（b）の例では、コマンド「DJNZ TABLSET10」におけるBレジスタの値が2以上であれば、その値に10サイクル（6行目4サイクル+7行目3サイクル+8行目3サイクル）を乗じた分だけ総実行サイクルが増えるので、図139（b）の総実行サイクルの削減量も多くなる。かかるTABLSETモジュールによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1237】

また、ここでは、図138の例と比較して、AレジスタおよびCレジスタを利用していない。したがって、Aレジスタの値やCレジスタの値を意図せず更新してしまうことはない。また、図138の例では、AレジスタやCレジスタが既に利用されている場合、スタックしてAレジスタやCレジスタの値を退避する必要があったが、ここでは、AレジスタやCレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1238】

また、図138（b）と図139（b）とを比較して理解できるように、図138（b）において4行（5行目～8行目）を占有していたコマンド群を、図139（b）においては1行（5行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1239】

なお、TABLSETモジュールは、ERRWAITモジュールのみならず、複数のモジュールからサブルーチンとして呼び出される。例えば、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、本前兆中処理（AT状態=「3」）を実行するHID__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、終了画面中処理（AT状態=「4」）を実行するFIN__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、準備中処理（AT状態=「5」）を実行するPRE__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、REG中処理（AT状態=「6」）を実行するREG__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊

技状態および演出状態に応じて選択的に移行し、BIG中処理（AT状態＝「7」）を実行するBIG_LOTモジュール、図85のステップS2020で示した設定値切り替え処理を実行するRANKSETモジュール、図91のステップS2400で示した図柄コード設定処理を実行するFGSETUPモジュール、図96のステップS2900で示した遊技移行処理を実行するGAMESETモジュール、図96のステップS2900-3で示した役物作動図柄表示処理を実行するJCGMSETモジュール等から呼び出される。

【1240】

〈コマンド「DECM」、「DECWM」〉

BYTEDECモジュールは、バイトカウンタ減算処理、すなわち、メインRAM300cの1バイト変数を1だけデクリメントするための汎用モジュールである。なお、BYTEDECモジュールでは、デクリメントのみならず、デクリメントした結果、ゼロフラグおよびキャリーフラグを設定している。

【1241】

メインCPU300aは、メインROM300bからプログラムを読み出し、読み出したプログラムを遂行し、任意の処理において、サブルーチンとしてBYTEDECモジュールを呼び出し、BYTEDECモジュールを遂行する。BYTEDECモジュールでは、メインRAM300cに保持された1バイト値をデクリメントする。

【1242】

図140は、BYTEDECモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図43のステップS630で示した特別図柄停止図柄表示処理を実行するTZ_STPモジュールの一部を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該BYTEDECモジュールの説明中、所定のレジスタはHLレジスタである。

【1243】

メインCPU300aは、図140（a）のように、任意の処理を実行する。メインCPU300aは、まず、HLレジスタに、デクリメントの対象となる変数のアドレスを設定する（S1）。そして、サブルーチンとしてBYTEDECモジュールを呼び出す（S2）。

【1244】

メインCPU300aは、図140（b）のように、BYTEDECモジュールにおいて、HLレジスタで示されるアドレスに格納された1バイト値を1だけデクリメントし、かつ、ゼロフラグおよびキャリーフラグを設定する（S3）。そして、当該BYTEDECモジュールを終了して1段上のルーチンに戻る（S4）。以下、デクリメント態様とフラグの設定について詳述する。

【1245】

図141は、BYTEDECモジュールにおけるデクリメント態様とゼロフラグおよびキャリーフラグの設定を説明するための説明図である。ここでは、デクリメント前が1以上（1または2以上）であれば、デクリメント後は、1だけデクリメントされた値になる。ただし、デクリメント前が0であった場合、デクリメント後は-1とならず、0となる。すなわち、下限値は0であり、負の値となることはない。このような0の維持は、デクリメント前が0であった場合、デクリメントしてさらにインクリメント、または、強制的に0を読み出して実現してもよいし、デクリメントを行わないことでも実現できる。

【1246】

また、デクリメント前が2以上であれば、デクリメント後は1以上となるので、ゼロフラグは立たず0となるが、デクリメント前が1または0であれば、デクリメント後は0となるので、ゼロフラグが立って1となる。また、当該BYTEDECモジュールでは、デクリメント前が1であれば、デクリメント後は0となり、その場合にキャリーフラグを立てて1とする。そして、デクリメント前が2以上または0であればキャリーフラグを立てず0とする。

【1247】

10

20

30

40

50

図140(a)に戻り、BYTEDECモジュールによって、ゼロフラグおよびキャリーフラグが設定されると、メインCPU300aは、そのフラグの値を用いて様々な処理を遂行する。例えば、ここでは、デクリメントが1を減算して0とする処理ではないことを判定し(S5)、その結果、1を減算して0とする処理でなければ(S5におけるYES)、特別図柄当たりフラグチェック処理を行う所定のアドレス(TZ__STP__10)に移動し(S6)、1を減算して0とする処理であれば(S5におけるNO)、移動することなく次の処理を遂行する。

【1248】

図142は、BYTEDECモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図142(a)は、BYTEDECモジュールを呼び出す任意の処理のコマンド群を示し、図142(b)は、BYTEDECモジュールのコマンド群を示す。図140で示したフローチャートは、例えば、図142に示したプログラムによって実現される。

【1249】

図142(a)の1行目のコマンド「LDQ HL, LOW R__EXT__HCT」によって、Qレジスタの値をHレジスタに読み出し、アドレス「R__EXT__HCT」の下位1バイトの値をLレジスタに読み出す。かかる1行目のコマンドが、図140(a)のステップS1に対応する。そして、2行目のコマンド「CALLF BYTEDEC」によって、サブルーチンとしてBYTEDECモジュールが呼び出される。かかる2行目のコマンドが、図140(a)のステップS2に対応する。

【1250】

ここで、コマンド「CALLF BYTEDEC」は、メモリマップのうち0000H～11FFHの範囲のみ呼び出しできるコマンドである。呼び出しに用いられるコマンド「CALLF」は通常のコマンド「CALL」と実行サイクルは「4」で等しいが、通常のコマンド「CALL」のコマンドサイズが「3」であるのに対し、コマンド「CALLF」のコマンドサイズは「2」である。したがって、プログラムの短縮化を図ることができる。なお、コマンド「CALLF BYTEDEC」の代わりに、上述した汎用モジュール同様「RST BYTEDEC」によってBYTEDECモジュールを呼び出すとしてもよい。そうすることで、コマンドサイズを「1」にすることができる。

【1251】

図142(b)の1行目の指標「BYTEDEC:」は、当該BYTEDECモジュールの先頭アドレスを示す。2行目のコマンド「LD A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値がAレジスタに読み出される。かかるコマンド「LD A, (HL)」では、その実行により第2ゼロフラグが変化する。そして、3行目のコマンド「RTZ, A」によって、Aレジスタの値が0であれば(第2ゼロフラグが1であれば)、1段上のルーチンに戻る。このとき、Aレジスタの値が0であれば、コマンド「RTZ, A」によってゼロフラグ(第1ゼロフラグ)が1となり、キャリーフラグは0となる。こうして、HLレジスタで示されるアドレスに格納された値が0であれば、デクリメントを行うことなく、ゼロフラグ=1、キャリーフラグ=0とすることができる。

【1252】

図142(b)の4行目のコマンド「CP A, 2」によって、Aレジスタの値と2が比較され、Aレジスタが2以上であれば、キャリーフラグは立たず0となり、1であれば、キャリーフラグが立って1となる。なお、コマンド「CP A, 2」は、Aレジスタが2未満の場合にキャリーフラグが立つコマンドであるが、かかるコマンド「CP A, 2」の実行段階では、Aレジスタの値は1か2以上のいずれかとなっているので(0ではないので)、2未満でキャリーフラグを立たせることで、結果的に1の場合にキャリーフラグが立つこととなる。こうして、Aレジスタの値が1であれば、キャリーフラグ=1とすることができる。5行目のコマンド「DEC (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値を1だけデクリメントして、HLレジスタで示され

10

20

30

40

50

るアドレスに格納された値を更新する。ここで、HLレジスタで示されるアドレスに格納された値が2以上であれば、デクリメントした値は1以上なので、ゼロフラグは立たず0となり、1であれば、デクリメントした値が0となるので、ゼロフラグが立って1となる。なお、コマンド「DEC (HL)」によってはキャリーフラグは変化しない。こうして、HLレジスタで示されるアドレスに格納された値が2以上であれば、ゼロフラグ=0、キャリーフラグ=0とし、1であれば、ゼロフラグ=1、キャリーフラグ=1とすることができる。かかる2～5行目のコマンドが図140(b)のステップS3に対応する。そして、6行目のコマンド「RET」によって、1段上のルーチンに戻る。かかる6行目のコマンドが、図140(b)のステップS4に対応する。こうして、HLレジスタで示されるアドレスに格納された値を1だけデクリメントし、かつ、ゼロフラグおよびキャリーフラグを設定することが可能となる。

10

【1253】

続いて、図142(a)の3行目のコマンド「JR NC, TZ__STP__10」によって、キャリーフラグが0(NC)であれば、TZ__STP__10に移動する。かかる3行目のコマンドが、図140(a)のステップS5、6に対応する。こうして、BYTEDECモジュールの結果に基づいた処理が可能となる。

【1254】

BYTEDECモジュールは、複数のモジュールからサブルーチンとして呼び出される。例えば、図26のステップS400-15で示したタイマ更新処理を実行するTMR__NEWモジュール、図26のステップS400-21で示した状態管理処理(エラー管理処理)を実行するSTC__PRCモジュール、図43のステップS630で示した特別図柄停止図柄表示処理を実行するTZ__STPモジュール、図40のステップS613で示した回数切り管理処理を実行するCHGSTSモジュール、図34のステップS540で示したカウントスイッチ通過処理(大入賞口通過処理)において大入賞口入球コマンドをセット(S540-3)した後に実行する大入賞口過剰入賞監視処理を実行するTDOVCHKモジュール、図26のステップS400-21で示した状態管理処理(エラー管理処理)のサブルーチンであるベース異常エラー監視処理を実行するBER__CHKモジュール等から呼び出される。

20

【1255】

このように、図142(b)に示したBYTEDECモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD A, (HL)」1バイト+3行目のコマンド「RTZ, A」1バイト+4行目のコマンド「CP A, 2」2バイト+5行目のコマンド「DEC (HL)」1バイト+6行目のコマンド「RET」1バイト=6バイトとなり、総実行サイクルは、2行目2サイクル+3行目4サイクル(または2サイクル)+4行目2サイクル+5行目4サイクル+6行目3サイクル=15サイクル(または13サイクル)となる。なお、括弧内のサイクル数は、コマンド「RTZ, A」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。かかるBYTEDECモジュールを設けることによって、上述した各モジュール内でバイトカウンタ減算処理を行うことなく、コマンドサイズ2バイトのコマンド「CALLF BYTEDEC」で賄うことができるので、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

30

40

【1256】

図143は、BYTEDECモジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図142(b)のBYTEDECモジュールのコマンド群を図143(b)のBYTEDECモジュールのコマンド群に置き換えており、他の、図142(a)のBYTEDECモジュールを呼び出す任意の処理のコマンド群は、図143(a)として、そのまま用いている。ここでは、図143(a)のように、図142(a)と実質的に等しい処理についてはその説明を省略し、図143(b)の異なる処理のみを説明する。

【1257】

50

図143 (b) の1行目の指標「BYTEDEC:」は、当該BYTEDECモジュールの先頭アドレスを示す。2行目のコマンド「DECM (HL)」によって、HLレジスタで示されるアドレスに格納された値が1以上であれば1だけデクリメントし、0であれば0を維持する。

【1258】

ここで、コマンド「DECM (HL)」は、HLレジスタで示されるアドレスに格納された1バイト値をデクリメントし、キャリーフラグが立っていれば(−1となれば)、HLレジスタで示されるアドレスに格納された1バイト値に強制的に0を格納するコマンドである。そうすると、HLレジスタで示されるアドレスに格納された値が1以上であれば1だけデクリメントし、0であれば0を維持することとなる。かかるコマンド「DECM (HL)」を実行することで、デクリメント後の値がゼロであれば(デクリメント前の値が1または0であれば)、ゼロフラグ(第1ゼロフラグおよび第2ゼロフラグ)を立てて1とし、デクリメント前の値がゼロであれば、キャリーフラグを立てて1とする。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「5」である。

【1259】

図143 (b) の3行目のコマンド「RET NZ」によって、ゼロフラグが1でなければ(デクリメント後の値がゼロでなければ)、1段上のルーチンに戻る。ここでは、コマンド「DECM (HL)」によって既にデクリメントが行われているので、ゼロフラグが1ではないということは、デクリメント前が2以上であることを示す。その場合、ゼロフラグは立たず0となり、キャリーフラグも立たずに0となるので、1段上のルーチンに戻ってもよい。なお、ここでは「NZ」を挙げ、第1ゼロフラグを参照しているが、通常、第1ゼロフラグを変化させた場合には併せて第2ゼロフラグも変化するので、全ての処理において「NZ」の代わりに「NTZ」を用いることもできる。

【1260】

なお、コマンド「DECM (HL)」では、デクリメント前のHLレジスタで示されるアドレスに格納された値が1であれば、デクリメント後に0となるとともに、ゼロフラグ=1、キャリーフラグ=0となり、0であれば、デクリメントはされず0を維持するとともに、ゼロフラグ=1、キャリーフラグ=1となる。すなわち、キャリーフラグについて、図141に示した仕様と逆になる。そこで、図143 (b) の4行目のコマンド「CCF」によって、キャリーフラグを反転(1→0、0→1)させる。なお、コマンド「CCF」によってゼロフラグは変化しない。こうして、デクリメント前にHLレジスタで示されるアドレスに格納された値が1であれば、ゼロフラグ=0、キャリーフラグ=1とし、0であれば、ゼロフラグ=1、キャリーフラグ=0とすることができる。かかる2~4行目のコマンドが図140 (b) のステップS3に対応する。そして、5行目のコマンド「RET」によって、1段上のルーチンに戻る。かかる5行目のコマンドが、図140 (b) のステップS4に対応する。こうして、HLレジスタで示されるアドレスに格納された値を1だけデクリメントし、かつ、ゼロフラグおよびキャリーフラグを設定することが可能となる。

【1261】

続いて、図143 (a) の3行目のコマンド「JR NC, TZ_STP_10」によって、キャリーフラグが0(NC)であれば、TZ_STP_10に移動する。かかる3行目のコマンドが、図140 (a) のステップS5、6に対応する。こうして、BYTEDECモジュールの結果に基づいた処理が可能となる。

【1262】

図143 (b) のBYTEDECモジュールのコマンドの総コマンドサイズは、2行目のコマンド「DECM (HL)」2バイト+3行目のコマンド「RET NZ」1バイト+4行目のコマンド「CCF」2バイト+5行目のコマンド「RET」1バイト=6バイトとなり、総実行サイクルは、2行目5サイクル+3行目3サイクル(1サイクル)+4行目2サイクル+6行目3サイクル=13サイクル(11サイクル)となる。なお、括弧内のサイクル数は、コマンド「RET NZ」によって1つ上のルーチンに移動しなかつ

10

20

30

40

50

た場合の実行サイクルを示している。したがって、図142(b)の場合と比べ、総実行サイクルが少なくとも2サイクル削減される。かかるBYTEDECモジュールによって、処理負荷の軽減を図りつつ、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1263】

また、ここでは、図142の例と比較して、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図142の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1264】

また、図142(b)と図143(b)とを比較して理解できるように、図142(b)において4行(2行目~5行目)を占有していたコマンド群を、図143(b)においては3行(2行目~4行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1265】

なお、かかるBYTEDECモジュールは、パチンコ機のみならず、スロットマシンにおいても、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE_SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、REG中処理(AT状態=「6」)を実行するREG_LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE_SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、BIG中処理(AT状態=「7」)を実行するBIG_LOTモジュール、図96のステップS2900-3で示した役物作動図柄表示処理を実行するJCGMSETモジュール、図98のステップS3100で示したタイマ割込み処理を実行するTMR_IPTモジュール、図98のステップS3100-21で示した状態別モジュール実行処理を実行するためのCAL_MODモジュールにおいて、タイマ割込み処理用フェーズ(0~3)に応じて4回に1回(5.96msec毎に)選択的に移行し、端子板出力制御処理を実行するIPT_PAモジュール、図98のステップS3100-21で示した状態別モジュール実行処理を実行するためのCAL_MODモジュールにおいて、タイマ割込み処理用フェーズ(0~3)に応じて4回に1回(5.96msec毎に)選択的に移行し、時間監視処理を実行するIPT_PCモジュール、および、図98のステップS3100-21で示した状態別モジュール実行処理を実行するためのCAL_MODモジュールにおいて、タイマ割込み処理用フェーズ(0~3)に応じて4回に1回(5.96msec毎に)選択的に移行し、外部信号出力制御処理を実行するIPT_PDモジュール等からコマンド「RST」によって呼び出される、カウンタ減算処理を実行するRAM_DECモジュール(汎用モジュール)として利用されている。

【1266】

図144は、RAM_DECモジュールを説明するための説明図である。図144に示したRAM_DECモジュールは、上述したBYTEDECモジュール同様、カウンタ減算処理、すなわち、メインRAM500cの1バイト変数を1だけデクリメントするための汎用モジュールである。なお、RAM_DECモジュールでは、BYTEDECモジュール同様、デクリメントのみならず、デクリメントした結果、図141に示したゼロフラグおよびキャリーフラグが設定される。

【1267】

図144(a)の1行目の指標「RAM_DEC:」は、当該RAM_DECモジュールの先頭アドレスを示す。2行目のコマンド「LD A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値がAレジスタに読み出される。かかるコマンド「LD A, (HL)」では、その実行により第2ゼロフラグが変化する。そして、3行目のコマンド「RTZ, A」によって、Aレジスタの値が0であれば(第2ゼロ

10

20

30

40

50

フラグが1であれば)、1段上のルーチンに戻る。このとき、Aレジスタの値が0であれば、コマンド「RTZ, A」によってゼロフラグ(第1ゼロフラグ)が1となり、キャリーフラグは0となる。こうして、HLレジスタで示されるアドレスに格納された値が0であれば、デクリメントを行うことなく、ゼロフラグ=1、キャリーフラグ=0とすることができる。

【1268】

図144(a)の4行目のコマンド「CP A, 2」によって、Aレジスタの値と2が比較され、Aレジスタが2以上であれば、キャリーフラグは立たず0となり、1であれば、キャリーフラグが立って1となる。なお、コマンド「CP A, 2」は、Aレジスタが2未満の場合にキャリーフラグが立つコマンドであるが、かかるコマンド「CP A, 2」の実行段階では、Aレジスタの値は1か2以上のいずれかとなっているので(0ではないので)、2未満でキャリーフラグを立たせることで、結果的に1の場合にキャリーフラグが立つこととなる。こうして、Aレジスタの値が1であれば、キャリーフラグ=1とすることができる。5行目のコマンド「DEC A」によって、Aレジスタで示されるアドレスに格納された1バイト値が1だけデクリメントされる。6行目のコマンド「LD (HL), A」によって、Aレジスタの値がHLレジスタで示されるアドレスに格納される。ここで、HLレジスタで示されるアドレスに格納された値が2以上であれば、デクリメントした値は1以上なので、ゼロフラグは立たず0となり、1であれば、デクリメントした値が0となるので、ゼロフラグが立って1となる。なお、コマンド「DEC A」によってキャリーフラグは変化しない。こうして、HLレジスタで示されるアドレスに格納された値が2以上であれば、ゼロフラグ=0、キャリーフラグ=0とし、1であれば、ゼロフラグ=1、キャリーフラグ=1とすることができる。そして、7行目のコマンド「RET」によって、1段上のルーチンに戻る。こうして、HLレジスタで示されるアドレスに格納された値を1だけデクリメントし、かつ、ゼロフラグおよびキャリーフラグを設定することが可能となる。

【1269】

ここで、コマンド「DECM」への置き換えを行うと、図144(a)のコマンド群を図144(b)のように変更することができる。図144(b)の1行目の指標「RAM__DEC:」は、当該RAM__DECモジュールの先頭アドレスを示す。2行目のコマンド「DECM (HL)」によって、HLレジスタで示されるアドレスに格納された値が1以上であれば1だけデクリメントし、0であれば0を維持する。

【1270】

図144(b)の3行目のコマンド「RET NZ」によって、ゼロフラグが1でなければ(デクリメント後の値がゼロでなければ)、1段上のルーチンに戻る。ここでは、コマンド「DECM (HL)」によって既にデクリメントが行われているので、ゼロフラグが1ではないということは、デクリメント前が2以上であることを示す。その場合、ゼロフラグは立たず0となり、キャリーフラグも立たずに0となるので、1段上のルーチンに戻ってもよい。続いて、4行目のコマンド「CCF」によって、キャリーフラグを反転(1→0、0→1)させる。なお、コマンド「CCF」によってゼロフラグは変化しない。こうして、デクリメント前にHLレジスタで示されるアドレスに格納された値が1であれば、ゼロフラグ=0、キャリーフラグ=1とし、0であれば、ゼロフラグ=1、キャリーフラグ=0とすることができる。そして、5行目のコマンド「RET」によって、1段上のルーチンに戻る。こうして、HLレジスタで示されるアドレスに格納された値を1だけデクリメントし、かつ、ゼロフラグおよびキャリーフラグを設定することが可能となる。

【1271】

ここで、図144(a)に示したRAM__DECモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD A, (HL)」1バイト+3行目のコマンド「RTZ, A」1バイト+4行目のコマンド「CP A, 2」2バイト+5行目のコマンド「DEC A」1バイト+6行目のコマンド「LD (HL), A」1バイト+7行目のコマンド「

RET」1バイト＝7バイトとなり、総実行サイクルは、2行目2サイクル＋3行目4サイクル（または2サイクル）＋4行目2サイクル＋5行目1サイクル＋6行目2サイクル＋7行目3サイクル＝14サイクル（または12サイクル）となる。なお、括弧内のサイクル数は、コマンド「RTZ, A」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。一方、図144（b）に示したRAM_DECモジュールのコマンドの総コマンドサイズは、2行目のコマンド「DECM（HL）」2バイト＋3行目のコマンド「RET NZ」1バイト＋4行目のコマンド「CCF」2バイト＋5行目のコマンド「RET」1バイト＝6バイトとなり、総実行サイクルは、2行目5サイクル＋3行目3サイクル（1サイクル）＋4行目2サイクル＋6行目3サイクル＝13サイクル（11サイクル）となる。なお、括弧内のサイクル数は、コマンド「RET NZ」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。したがって、図144（a）のコマンド群を図144（b）のコマンド群に置き換えることで、総コマンドサイズが1バイト削減され、総実行サイクルが少なくとも1サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1272】

また、WORDDECモジュールは、ワードカウンタ減算処理、すなわち、メインRAM300cの2バイト変数を1だけデクリメントするための汎用モジュールである。なお、WORDDECモジュールでは、デクリメントのみならず、デクリメントした結果、ゼロフラグおよびキャリーフラグを設定している。

【1273】

メインCPU300aは、メインROM300bからプログラムを読み出し、読み出したプログラムを遂行し、任意の処理において、サブルーチンとしてWORDDECモジュールを呼び出し、WORDDECモジュールを遂行する。WORDDECモジュールでは、メインRAM300cに保持された2バイトデータをデクリメントする。

【1274】

図145は、WORDDECモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図26のステップS400－21で示した状態管理処理（エラー管理処理）のサブルーチンである入賞頻度異常エラー判定処理を実行するNHICHKモジュールの一部を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該WORDDECモジュールの説明中、所定のレジスタはHLレジスタである。

【1275】

メインCPU300aは、図145（a）のように、任意の処理を実行する。メインCPU300aは、まず、HLレジスタに、デクリメントの対象となる変数のアドレスを設定する（S1）。そして、サブルーチンとしてWORDDECモジュールを呼び出す（S2）。

【1276】

メインCPU300aは、図145（b）のように、WORDDECモジュールにおいて、HLレジスタで示されるアドレスに格納された2バイト値を1だけデクリメントし、かつ、ゼロフラグおよびキャリーフラグを設定する（S3）。そして、当該WORDDECモジュールを終了して1段上のサブルーチンに戻る（S4）。なお、デクリメント態様とフラグの設定については、BYTEDECモジュール同様、図141で説明されている。

【1277】

WORDDECモジュールによって、ゼロフラグおよびキャリーフラグが設定されると、メインCPU300aは、そのHLレジスタの値を用いて様々な処理を遂行する。例えば、ここでは、デクリメント結果が0ではないことを判定し（S5）、その結果、0でなければ（S5におけるYES）、当該サブルーチンから戻り（S6）、0であれば（S5におけるNO）、移動することなく次の処理を遂行する。

【1278】

図146は、WORDDECモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図146(a)は、任意の処理のコマンド群を示し、図146(b)は、WORDDECモジュールのコマンド群を示す。図145で示したフローチャートは、例えば、図146に示したプログラムによって実現される。

【1279】

図146(a)の1行目のコマンド「POP HL」によって、スタック領域に退避されていた入賞頻度異常エラー監視タイマのアドレス値をHLレジスタに復帰させる。かかる1行目のコマンドが、図145(a)のステップS1に対応する。そして、2行目のコマンド「CALLF WORDDEC」によって、サブルーチンとしてWORDDECモジュールが呼び出される。かかる2行目のコマンドが、図145(a)のステップS2に対応する。

【1280】

図146(b)の1行目の指標「WORDDEC:」は、当該WORDDECモジュールの先頭アドレスを示す。2行目のコマンド「JTWZ, (HL), WORDDEC__99」によって、HLレジスタで示されるアドレスに格納された2バイト値が0かどうか判断され、0であれば、指標「WORDDEC__99:」に移動する。かかるコマンド「JTWZ, (HL), WORDDEC__99」では、その実行によりゼロフラグが変化する。例えば、HLレジスタで示されるアドレスに格納された値が0であれば、ゼロフラグが1となり、キャリーフラグは0となる。また、HLレジスタで示されるアドレスに格納された値が0であれば、指標「WORDDEC__99:」に移動し、7行目のコマンド「RET」によって、1段上のサブルーチンに戻る。こうして、HLレジスタで示されるアドレスに格納された値が0であれば、デクリメントを行うことなく、ゼロフラグ=1、キャリーフラグ=0とすることができる。

【1281】

図146(b)の3行目のコマンド「DECW (HL)」によって、HLレジスタで示されるアドレスに格納された2バイト値を1だけデクリメントして、HLレジスタで示されるアドレスに格納された値を更新する。ここで、HLレジスタで示されるアドレスに格納された値が2以上であれば、デクリメントした値は1以上なので、ゼロフラグは立たず0となり、1であれば、デクリメントした値が0となるので、ゼロフラグ(第2ゼロフラグ)が立って1となる。そして、4行目のコマンド「RET NTZ」によって、ゼロフラグが1でなければ、1段上のサブルーチンに戻る。ここで「NTZ」としているのは、コマンド「DECW (HL)」によって第2ゼロフラグは変化するが第1ゼロフラグは変化しないからである。なお、コマンド「RET NTZ」によっては、ゼロフラグもキャリーフラグも変化しない。このとき、ゼロフラグが1とならない(1段上のサブルーチンに戻る)のは、デクリメント前のHLレジスタで示されるアドレスに格納された値が2以上の場合である。したがって、デクリメント前が2以上であれば、ゼロフラグ=0、キャリーフラグ=0となっているので、1段上のサブルーチンに戻っても問題ない。

【1282】

また、ゼロフラグが1であれば、デクリメント前のHLレジスタで示されるアドレスに格納された値が1であり、ゼロフラグ=1、キャリーフラグ=0となっている。すなわち、キャリーフラグについて、図141に示した仕様と逆になる。そこで、5行目のコマンド「SCF」によって、キャリーフラグを強制的に立てて1とする。こうして、デクリメント前にHLレジスタで示されるアドレスに格納された値が1であれば、ゼロフラグ=1、キャリーフラグ=1となる。なお、コマンド「SCF」によってはゼロフラグは変化しない。かかる2~6行目のコマンドが図145(b)のステップS3に対応する。そして、7行目のコマンド「RET」によって、1段上のサブルーチンに戻る。かかる7行目のコマンドが、図145(b)のステップS4に対応する。こうして、HLレジスタで示されるアドレスに格納された値を1だけデクリメントし、かつ、ゼロフラグおよびキャリーフラグを設定することが可能となる。

【1283】

続いて、図146(a)の3行目のコマンド「RET NTZ」によって、ゼロフラグ(第2ゼロフラグ)が1でなければ(0であれば)、さらに1段上のサブルーチンに戻る。かかる3行目のコマンドが、図145(a)のステップS5、6に対応する。こうして、WORDDECモジュールの結果に基づいた処理が可能となる。

【1284】

WORDDECモジュールは、複数のモジュールからサブルーチンとして呼び出される。例えば、図26のステップS400-15で示したタイマ更新処理を実行するTMR__NEWモジュール、図41のステップS620で示した特別図柄変動中処理を実行するTZ__SPNモジュール、図26のステップS400-21で示した状態管理処理(エラー管理処理)のサブルーチンである入賞頻度異常エラー判定処理を実行するNHIE__CHKモジュール等から呼び出される。

10

【1285】

このように、図146(b)に示したWORDDECモジュールのコマンドの総コマンドサイズは、2行目のコマンド「JTW Z, (HL), WORDDEC__99」3バイト+3行目のコマンド「DECW (HL)」2バイト+4行目のコマンド「RET NTZ」1バイト+5行目のコマンド「SCF」2バイト+7行目のコマンド「RET」1バイト=9バイトとなり、総実行サイクルは、2行目6サイクル(または5サイクル)+3行目7サイクル+4行目3サイクル(または1サイクル)+5行目2サイクル+7行目3サイクル=21サイクル(または18サイクル)となる。なお、括弧内のサイクル数は、コマンド「JTW Z, (HL), WORDDEC__99」やコマンド「RET NTZ」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。かかるWORDDECモジュールを設けることによって、上述した各モジュール内でワードカウンタ減算処理を行うことなく、コマンドサイズ2バイトのコマンド「CALLF WORDDEC」で賄うことができるので、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

20

【1286】

図147は、WORDDECモジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図146(b)のWORDDECモジュールのコマンド群を図147(b)のWORDDECモジュールのコマンド群に置き換えており、他の、図146(a)のWORDDECモジュールを呼び出す任意の処理のコマンド群は、図147(a)として、そのまま用いている。ここでは、図147(a)のように、図146(a)と実質的に等しい処理についてはその説明を省略し、図147(b)の異なる処理のみを説明する。

30

【1287】

図147(b)の1行目の指標「WORDDEC:」は、当該WORDDECモジュールの先頭アドレスを示す。2行目のコマンド「DECWM (HL)」によって、HLレジスタで示されるアドレスに格納された値が1以上であれば1だけデクリメントし、0であれば0を維持する。

【1288】

ここで、コマンド「DECWM (HL)」は、HLレジスタで示されるアドレスに格納された2バイト値をデクリメントし、キャリーフラグが立っていれば(-1となれば)、HLレジスタで示されるアドレスに格納された2バイト値に強制的に0を格納するコマンドである。そうすると、HLレジスタで示されるアドレスに格納された値が1以上であれば1だけデクリメントし、0であれば0を維持することとなる。かかるコマンド「DECWM (HL)」を実行することで、デクリメント後の値がゼロであれば(デクリメント前の値が1または0であれば)、ゼロフラグ(第1ゼロフラグおよび第2ゼロフラグ)を立てて1とし、デクリメント前の値がゼロであれば、キャリーフラグを立てて1とする。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「7」である。

40

【1289】

図147(b)の3行目のコマンド「RET NZ」によって、ゼロフラグが1でなけ

50

れば（デクリメント後の値がゼロでなければ）、1 段上のサブルーチンに戻る。ここでは、コマンド「DECWM (HL)」によって既にデクリメントが行われているので、ゼロフラグが1ではないということは、デクリメント前が2以上であることを示す。その場合、ゼロフラグは立たず0となり、キャリーフラグも立たずに0となるので、1 段上のサブルーチンに戻ってもよい。

【1290】

なお、コマンド「DECWM (HL)」では、デクリメント前のHLレジスタで示されるアドレスに格納された値が1であれば、デクリメント後に0となるとともに、ゼロフラグ=1、キャリーフラグ=0となり、0であれば、デクリメントはされず0を維持するとともに、ゼロフラグ=1、キャリーフラグ=1となる。すなわち、キャリーフラグにつ

いて、図141に示した仕様と逆になる。そこで、図147 (b) の4 行目のコマンド「CCF」によって、キャリーフラグを反転(1→0、0→1)させる。なお、コマンド「CCF」によってはゼロフラグは変化しない。こうして、デクリメント前にHLレジスタで示されるアドレスに格納された値が1であれば、ゼロフラグ=1、キャリーフラグ=1とし、0であれば、ゼロフラグ=1、キャリーフラグ=0とすることができる。かかる2～4 行目のコマンドが図145 (b) のステップS3に対応する。そして、5 行目のコマンド「RET」によって、1 段上のサブルーチンに戻る。かかる5 行目のコマンドが、図145 (b) のステップS4に対応する。こうして、HLレジスタで示されるアドレスに格納された値を1だけデクリメントし、かつ、ゼロフラグおよびキャリーフラグを設定することが可能となる。

【1291】

続いて、図147 (a) の3 行目のコマンド「RET NTZ」によって、ゼロフラグ（第2ゼロフラグ）が1でなければ（0であれば）、さらに1 段上のサブルーチンに戻る。かかる3 行目のコマンドが、図145 (a) のステップS5、6に対応する。こうして、WORDDECモジュールの結果に基づいた処理が可能となる。

【1292】

図147 (b) のWORDDECモジュールのコマンドの総コマンドサイズは、2 行目2 バイト+3 行目1 バイト+4 行目2 バイト+5 行目1 バイト=6 バイトとなり、総実行サイクルは、2 行目7 サイクル+3 行目3 サイクル(1 サイクル)+4 行目2 サイクル+6 行目3 サイクル=15 サイクル(13 サイクル)となる。なお、括弧内のサイクル数は、コマンド「RET NZ」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。したがって、図146 (b) の場合と比べ、総コマンドサイズが3 バイト削減され、総実行サイクルが少なくとも6 サイクル削減される。かかるWORDDECモジュールによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1293】

また、図146 (b) と図147 (b) とを比較して理解できるように、図146 (b) において5 行(2 行目～6 行目)を占有していたコマンド群を、図147 (b) においては3 行(2 行目～4 行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1294】

〈コマンド「RCP」〉

図148は、サブルーチンから戻る処理の一例を説明するための説明図である。図148 (a) におけるコマンド「CP A, n」は、Aレジスタの値と、所定の値nとを比較するコマンドであり、A=nの場合、ゼロフラグが立って1となり、A<nの場合、キャリーフラグが立って1となる。かかるコマンド「CP A, n」のコマンドサイズは「2」であり、実行サイクルは「2」である。

【1295】

図148 (a) におけるコマンド「RET cc」は、ゼロフラグまたはキャリーフラグを参照し、対象となるフラグが1となっていれば、サブルーチンから1 段上のルーチン

10

20

30

40

50

に戻るコマンドである。かかるコマンド「RET cc」のコマンドサイズは「1」であり、実行サイクルは「3」（または「1」）である。なお、括弧内のサイクル数は、コマンド「RET cc」によって1つ上のルーチンに移動しなかった（サブルーチンから戻らなかった）場合の実行サイクルを示している。

【1296】

ここでは、図148（a）のように、1行目のコマンド「CP A, n」によって、Aレジスタの値と、所定の値nとを比較し、2行目のコマンド「RET cc」によって、その比較結果に応じてサブルーチンから戻るか否かが決定する。ここで、かかる2つのコマンドを、図148（b）のように、1つのコマンド「RCP cc, A, n」に纏める（置き換える）ことができる。

【1297】

図148（b）におけるコマンド「RCP cc, A, n」は、Aレジスタの値と、所定の値nとを比較し、その比較結果に応じてサブルーチンから1段上のルーチンに戻るコマンドである。かかるコマンド「RCP cc, A, n」のコマンドサイズは「2」であり、実行サイクルは「5」（または「3」）である。なお、括弧内のサイクル数は、コマンド「RCP cc, A, n」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1298】

ここで、図148（a）に示したコマンド群の総コマンドサイズは、1行目のコマンド「CP A, n」2バイト+2行目の「RET cc」1バイト=3バイトとなり、総実行サイクルは、1行目2サイクル+2行目3サイクル（1サイクル）=5サイクル（3サイクル）となる。一方、図148（b）に示したコマンド「RCP cc, A, n」のコマンドサイズは、2バイトとなり、実行サイクルは、5サイクル（3サイクル）となる。したがって、図148（a）のコマンド群を図148（b）のコマンドに置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。以下、このような置き換えが可能な複数のモジュールをそれぞれ説明する。

【1299】

図149は、PY_CMDAモジュールを説明するための説明図である。図149に示したPY_CMDAモジュールは、図23におけるステップS100-63に示した主コマンド解析処理、すなわち、主コマンドを解析し、主コマンドの異常を検出する処理を実行する。

【1300】

図149（a）の1行目の指標「PY_CMDA:」は、当該PY_CMDAモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW R_BCR_BUF)」によって、Qレジスタの値をアドレスの上位1バイトとし、主コマンドのバッファ値を格納するアドレス「R_BCR_BUF」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（主コマンド）をAレジスタに読み出す。

【1301】

そして、バッファをクリアし、払出起動指定コマンド確認処理を実行すると、受信データ確認処理として、図149（a）の3行目のコマンド「SUB A, @BCR_PAY_CLR」によって、Aレジスタの値から「@BCR_PAY_CLR」で示される固定値、ここでは、20Hを減算する。そして、4行目のコマンド「CP A, 00CH」によって、減算されたAレジスタの値と0CHとを比較する。ここでは、主コマンドが20H~2BHの範囲に含まれることを確認している。すなわち、Aレジスタの値が20H未満であれば、コマンド「SUB A, @BCR_PAY_CLR」によってAレジスタの値は負の値になり、1バイト値としてみると0CHより大きくなる。したがって、減算されたAレジスタの値が0CH未満であれば、コマンド「CP A, 00CH」によってキャリーフラグが立って1となる。そして、5行目の「RET NC」によって、キャリーフラグが1でなければ、1段上のルーチンに戻る。なお、キャリーフラグが1であれば（

10

20

30

40

50

20H～2BHの範囲に含まれれば)、払出エラー指定コマンドセット処理および払出電波エラーフラグ設定処理を実行して、1段上のルーチンに戻る。

【1302】

ここで、図148の置き換えを行うと、図149(a)のコマンド群を図149(b)のように変更することができる。図149(b)の1行目の指標「PY__CMDA:」は、当該PY__CMDAモジュールの先頭アドレスを示す。1行目のコマンド「LDQ A, (LOW R__BCR__BUF)」によって、Qレジスタの値をアドレスの上位1バイトとし、主コマンドのバッファ値を格納するアドレス「R__BCR__BUF」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(主コマンド)をAレジスタに読み出す。

10

【1303】

そして、バッファをクリアし、払出起動指定コマンドに対する所定の処理を実行すると、図149(b)の3行目のコマンド「SUB A, @BCR__PAY__CLR」によって、Aレジスタの値から「@BCR__PAY__CLR」で示される固定値、ここでは、20Hを減算する。4行目のコマンド「RCP NC, A, 00CH」によって、減算されたAレジスタの値と0CHとを比較し、キャリーフラグが1でなければ、1段上のルーチンに戻る。ここでは、主コマンドが20H～2BHの範囲に含まれることを確認し、その範囲に入っていないければ、それ以降の処理を行わず1段上のルーチンに戻っている。なお、キャリーフラグが1であれば、払出エラー指定コマンドセット処理および払出電波エラーフラグ設定処理を実行して、1段上のルーチンに戻る。

20

【1304】

ここで、図149(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__BCR__BUF)」2バイト+3行目のコマンド「SUB A, @BCR__PAY__CLR」2バイト+4行目のコマンド「RCP A, 00CH」2バイト+5行目の「RET NC」1バイト=7バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目2サイクル+5行目3サイクル(1サイクル)=10サイクル(8サイクル)となる。一方、図149(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__BCR__BUF)」2バイト+3行目のコマンド「SUB A, @BCR__PAY__CLR」2バイト+4行目のコマンド「RCP NC, A, 00CH」2バイト=6バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目5サイクル(3サイクル)=10サイクル(8サイクル)となる。したがって、図149(a)のコマンド群を図149(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

30

【1305】

図150は、GAT__PASモジュールを説明するための説明図である。図150に示したGAT__PASモジュールは、図29のステップS510で示したゲートスイッチ通過処理(ゲート通過処理)、すなわち、ゲート検出スイッチ124sでの入賞球の通過に応じて普通図柄保留球数を加算し、普通図柄に関する処理を実行する。

40

【1306】

記憶カウントチェック処理として、図150(a)の1行目の指標「GAT__PAS:」は、当該GAT__PASモジュールの先頭アドレスを示す。2行目のコマンド「LDQ DE, LOW R__FZ__MEM」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__FZ__MEM」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された2バイト値(普通図柄保留球数カウンタのアドレス)をDEレジスタに読み出す。3行目のコマンド「LD A, (DE)」によって、DEレジスタで示されるアドレス(すなわちR__FZ__MEM)に格納された値(普通図柄保留球数カウンタのカウント値)をAレジスタに読み出す。

【1307】

50

そして、図150(a)の4行目のコマンド「CP A, @FZ__MEM__MAX」によって、Aレジスタの値と固定値「@FZ__MEM__MAX」、ここでは、「4」とを比較する。すなわち、Aレジスタの値が4未満であれば、キャリーフラグが立って1となる。そして、5行目の「RET NC」によって、キャリーフラグが1でなければ、1段上のルーチンに戻る。すなわち、普通図柄保留球数カウンタのカウント値が4以上であれば、それ以上、普通図柄保留球数カウンタを計数することはないので、当該GAT__PASモジュールを終了する。なお、キャリーフラグが1であれば、乱数値取得処理、記憶数加算処理、当たり決定乱数取得処理、転送先アドレス算定処理、および、記憶エリア格納処理を実行して、1段上のルーチンに戻る。

【1308】

ここで、図148の置き換えを行うと、図150(a)のコマンド群を図150(b)のように変更することができる。図150(b)の1行目の指標「GAT__PAS:」は、当該GAT__PASモジュールの先頭アドレスを示す。1行目のコマンド「LDQ DE, LOW R__FZ__MEM」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__FZ__MEM」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された2バイト値（普通図柄保留球数カウンタのアドレス）をDEレジスタに読み出す。3行目のコマンド「LD A, (DE)」によって、DEレジスタで示されるアドレス（すなわちR__FZ__MEM）に格納された値（普通図柄保留球数カウンタのカウント値）をAレジスタに読み出す。

【1309】

そして、図150(b)の4行目のコマンド「RCP NC, A, @FZ__MEM__MAX」によって、Aレジスタの値と@FZ__MEM__MAXとを比較し、Aレジスタの値が@FZ__MEM__MAX以上であれば、1段上のルーチンに戻る。すなわち、普通図柄保留球数カウンタのカウント値が4以上であれば、それ以上、普通図柄保留球数カウンタを計数することはないので、当該GAT__PASモジュールを終了する。なお、キャリーフラグが1であれば、記憶数加算処理、当たり決定乱数取得処理、転送先アドレス算定処理、および、記憶エリア格納処理を実行して、1段上のルーチンに戻る。

【1310】

ここで、図150(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ DE, LOW R__FZ__MEM」2バイト+3行目のコマンド「LD A, (DE)」1バイト+4行目のコマンド「CP A, @FZ__MEM__MAX」2バイト+5行目の「RET NC」1バイト=6バイトとなり、総実行サイクルは、2行目2サイクル+3行目2サイクル+4行目2サイクル+5行目3サイクル（1サイクル）=9サイクル（7サイクル）となる。一方、図150(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ DE, LOW R__FZ__MEM」2バイト+3行目のコマンド「LD A, (DE)」1バイト+4行目のコマンド「RCP NC, A, @FZ__MEM__MAX」2バイト=5バイトとなり、総実行サイクルは、2行目2サイクル+3行目2サイクル+4行目5サイクル（3サイクル）=9サイクル（7サイクル）となる。したがって、図150(a)のコマンド群を図150(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1311】

図151は、TDN__PASモジュールを説明するための説明図である。図151に示したTDN__PASモジュールは、図34のステップS540で示したカウントスイッチ通過処理（大入賞口通過処理）、すなわち、カウントスイッチ判定値に基づいて大入賞口の入賞を管理する処理を実行する。

【1312】

図151(a)の1行目の指標「TDN__PAS:」は、当該TDN__PASモジュールの先頭アドレスを示す。そして、カウントスイッチ判定値確認処理、特別電動役物連続

10

20

30

40

50

作動回数判定処理、大入賞口入賞指定コマンドセット処理、大入賞口過剰入賞監視処理を実行すると、特別電動役物作動チェック処理として、2行目のコマンド「LDQ A, (LOW R_TDN_PHS)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_TDN_PHS」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別電動役物遊技管理フェーズ）をAレジスタに読み出す。

【1313】

図151(a)の3行目のコマンド「JCP Z, A, @TD_OPN_BHT, TDN_PAS_10」によって、Aレジスタの値と固定値「@TD_OPN_BHT」、ここでは、2と比較し、その結果が等しければ、アドレス「TDN_PAS_10」に移動する。こうして、Aレジスタの値が大当たり大入賞口開放制御状態指定値（特別電動役物遊技管理フェーズが02H）であれば、以降の処理を省略して6行目の指標「TDN_PAS_10:」に移動することができる。4行目のコマンド「CP A, @TD_OPN_SHT」によって、Aレジスタの値と、小当たり大入賞口開放制御状態指定値を示す固定値「@TD_OPN_SHT」、ここでは、6とを比較し、等しければ（特別電動役物遊技管理フェーズが06H）、ゼロフラグが立って1となる。そして、5行目の「RET NZ」によって、ゼロフラグが1でなければ、1段上のルーチンに戻る。すなわち、Aレジスタの値が大当たり大入賞口開放制御状態指定値および小当たり大入賞口開放制御状態指定値でなければ、当該TDN_PASモジュールを終了する。なお、ゼロフラグが1であれば、特別電動役物入賞球数カウント処理、特電不正入賞検出条件確認処理、および、特電不正入賞エラー時処理を実行して、1段上のルーチンに戻る。

【1314】

ここで、図148の置き換えを行うと、図151(a)のコマンド群を図151(b)のように変更することができる。図151(b)の1行目の指標「TDN_PAS:」は、当該TDN_PASモジュールの先頭アドレスを示す。そして、カウントスイッチ判定値確認処理、特別電動役物連続作動回数判定処理、大入賞口入賞指定コマンドセット処理、大入賞口過剰入賞監視処理を実行すると、2行目のコマンド「LDQ A, (LOW R_TDN_PHS)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_TDN_PHS」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別電動役物遊技管理フェーズ）をAレジスタに読み出す。

【1315】

図151(b)の3行目のコマンド「JCP Z, A, @TD_OPN_BHT, TDN_PAS_10」によって、Aレジスタの値と固定値「@TD_OPN_BHT」、ここでは、2と比較し、その結果が等しければ、アドレス「TDN_PAS_10」に移動する。こうして、Aレジスタの値が大当たり大入賞口開放制御状態指定値であれば、以降の処理を省略して5行目の指標「TDN_PAS_10:」に移動することができる。4行目のコマンド「RCP NZ, A, @TD_OPN_SHT」によって、Aレジスタの値と、小当たり大入賞口開放制御状態指定値を示す固定値「@TD_OPN_SHT」、ここでは、6とを比較し、等しければ、1段上のルーチンに戻る。すなわち、Aレジスタの値が大当たり大入賞口開放制御状態指定値および小当たり大入賞口開放制御状態指定値でなければ、当該TDN_PASモジュールを終了する。なお、ゼロフラグが0であれば、特別電動役物入賞球数カウント処理、特電不正入賞検出条件確認処理、および、特電不正入賞エラー時処理を実行して、1段上のルーチンに戻る。

【1316】

ここで、図151(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R_TDN_PHS)」2バイト+3行目のコマンド「JCP Z, A, @TD_OPN_BHT, TDN_PAS_10」3バイト+4行目のコマンド「CP A, @TD_OPN_SHT」2バイト+5行目の「RET NZ」1バイト=8バイトとなり、総実行サイクルは、2行目3サイクル+3行目4サイクル（または3サイクル）+4行目2サイクル+5行目3サイクル（1サイクル）=12サイクル（9サイク

10

20

30

40

50

ル)となる。一方、図151(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R_TDN_PHS)」2バイト+3行目のコマンド「JCP Z, A, @TD_OPN_BHT, TDN_PAS_10」3バイト+4行目のコマンド「RCP NZ, A, @TD_OPN_SHT」2バイト=7バイトとなり、総実行サイクルは、2行目3サイクル+3行目4サイクル(または3サイクル)+4行目5サイクル(3サイクル)=12サイクル(9サイクル)となる。したがって、図151(a)のコマンド群を図151(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1317】

図152は、FD_OPNモジュールを説明するための説明図である。図152に示したFD_OPNモジュールは、図57のステップ850で示した普通電動役物入賞口開放制御処理、すなわち、普通電動役物に関する入賞数および作動時間に対する第1可変始動口120Bの開閉制御処理を実行する。

【1318】

図152(a)の1行目の指標「FD_OPN:」は、当該FD_OPNモジュールの先頭アドレスを示す。そして、普通電動役物入賞口開閉動作切替処理を実行すると、規定入賞数確認処理として、2行目のコマンド「LDQ A, (LOW R_FDNCNT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FDNCNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(普通電動役物入賞球数カウンタのカウンタ値)をAレジスタに読み出す。

【1319】

図152(a)の3行目のコマンド「CP A, @FDNCNT」によって、Aレジスタの値と、普通電動役物に係る入賞口入賞球数の最大値を示す固定値「@FDNCNT」、ここでは、8とを比較し、8未満であれば、キャリーフラグが立って1となる。そして、4行目の「RET C」によって、キャリーフラグが1であれば、1段上のルーチンに戻る。すなわち、Aレジスタの値が普通電動役物に係る入賞口入賞球数未満であれば、当該FD_OPNモジュールを終了する。なお、キャリーフラグが0であれば、普通電動役物作動終了時設定処理を実行して、1段上のルーチンに戻る。

【1320】

ここで、図148の置き換えを行うと、図152(a)のコマンド群を図152(b)のように変更することができる。図152(b)の1行目の指標「FD_OPN:」は、当該FD_OPNモジュールの先頭アドレスを示す。そして、普通電動役物入賞口開閉動作切替処理を実行すると、規定入賞数確認処理として、2行目のコマンド「LDQ A, (LOW R_FDNCNT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FDNCNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(普通電動役物入賞球数カウンタのカウンタ値)をAレジスタに読み出す。

【1321】

図152(b)の3行目のコマンド「RCP C, A, @FDNCNT」によって、Aレジスタの値と、普通電動役物に係る入賞口入賞球数の最大値を示す固定値「@FDNCNT」、ここでは、8とを比較し、8未満であれば、1段上のルーチンに戻る。すなわち、Aレジスタの値が普通電動役物に係る入賞口入賞球数未満であれば、当該FD_OPNモジュールを終了する。なお、キャリーフラグが0であれば、普通電動役物作動終了時設定処理を実行して、1段上のルーチンに戻る。

【1322】

ここで、図152(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R_FDNCNT)」2バイト+3行目のコマンド「CP A, @FDNCNT」2バイト+4行目の「RET C」1バイト=5バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目3サイクル(1サイクル

10

20

30

40

50

) = 8 サイクル (6 サイクル) となる。一方、図 152 (b) に示したコマンド群の総コマンドサイズは、2 行目のコマンド「LDQ A, (LOW R_FDN_CNT)」2 バイト + 3 行目のコマンド「RCP C, A, @FDN_CNT」2 バイト = 4 バイトとなり、総実行サイクルは、2 行目 3 サイクル + 3 行目 5 サイクル (3 サイクル) = 8 サイクル (6 サイクル) となる。したがって、図 152 (a) のコマンド群を図 152 (b) のコマンド群に置き換えることで、総コマンドサイズが 1 バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1323】

図 153 は、TZ__STA モジュールを説明するための説明図である。図 153 に示した TZ__STA モジュールは、図 37 のステップ S610 で示した特別図柄変動待ち処理、すなわち、特別図柄保留球数に基づいて特別図柄変動の準備処理を実行する。

10

【1324】

図 153 (a) の 1 行目の指標「TZ__STA:」は、当該 TZ__STA モジュールの先頭アドレスを示す。そして、特別図柄保留球数確認処理を実行すると、処理非対象特図状態確認処理として、2 行目のコマンド「RST DATSEL」によって、汎用モジュールである DATSEL モジュールが呼び出され、制御データの選択結果が A レジスタに格納される。

【1325】

図 153 (a) の 3 行目のコマンド「CP A, @TZ__BHT」によって、A レジスタの値と、特別図柄大当たり情報を示す固定値「@TZ__BHT」、ここでは、01H とを比較し、01H と等しければ、ゼロフラグが立って 1 となる。そして、4 行目の「RET Z」によって、ゼロフラグが 1 であれば、1 段上のルーチンに戻る。すなわち、大当たり確定中であれば、当該 TZ__STA モジュールを終了する。5 行目のコマンド「CP A, @TZ__SHT」によって、A レジスタの値と、特別図柄小当たり情報を示す固定値「@TZ__SHT」、ここでは、02H とを比較し、02H と等しければ、ゼロフラグが立って 1 となる。そして、6 行目の「RET Z」によって、ゼロフラグが 1 であれば、1 段上のルーチンに戻る。すなわち、小当たり確定中であれば、当該 TZ__STA モジュールを終了する。なお、ゼロフラグが 1 でなければ、特別図柄変動開始時設定処理、回数コマンドセット処理、および、特別図柄変動準備処理を実行して、1 段上のルーチンに戻る。

20

30

【1326】

ここで、図 148 の置き換えを行うと、図 153 (a) のコマンド群を図 153 (b) のように変更することができる。図 153 (b) の 1 行目の指標「TZ__STA:」は、当該 TZ__STA モジュールの先頭アドレスを示す。そして、特別図柄保留球数確認処理を実行すると、処理非対象特図状態確認処理として、2 行目のコマンド「RST DATSEL」によって、DATSEL モジュールが呼び出され、制御データの選択結果が A レジスタに格納される。

【1327】

図 153 (b) の 3 行目のコマンド「RCP Z, A, @TZ__BHT」によって、A レジスタの値と、特別図柄大当たり情報を示す固定値「@TZ__BHT」、ここでは、01H とが比較され、01H と等しければ、1 段上のルーチンに戻る。すなわち、大当たり確定中であれば、当該 TZ__STA モジュールを終了する。4 行目のコマンド「RCP Z, A, @TZ__SHT」によって、A レジスタの値と、特別図柄小当たり情報を示す固定値「@TZ__SHT」、ここでは、02H とが比較され、02H と等しければ、1 段上のルーチンに戻る。すなわち、小当たり確定中であれば、当該 TZ__STA モジュールを終了する。なお、ゼロフラグが 1 でなければ、特別図柄変動開始時設定処理、回数コマンドセット処理、および、特別図柄変動準備処理を実行して、1 段上のルーチンに戻る。

40

【1328】

ここで、図 153 (a) に示したコマンド群の総コマンドサイズは、2 行目のコマンド

50

「RST DATSEL」1バイト+3行目のコマンド「CP A, @TZ__BHT」2バイト+4行目の「RET Z」1バイト+5行目のコマンド「CP A, @TZ__SHT」2バイト+6行目の「RET Z」1バイト=7バイトとなり、総実行サイクルは、2行目4サイクル+3行目2サイクル+4行目3サイクル(1サイクル)+5行目2サイクル+6行目3サイクル(1サイクル)=14サイクル(10サイクル)となる。一方、図153(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「RST DATSEL」1バイト+3行目のコマンド「RCP Z, A, @TZ__BHT」2バイト+4行目のコマンド「RCP Z, A, @TZ__SHT」2バイト=5バイトとなり、総実行サイクルは、2行目4サイクル+3行目5サイクル(3サイクル)+4行目5サイクル(3サイクル)=14サイクル(10サイクル)となる。したがって、図153(a)のコマンド群を図153(b)のコマンド群に置き換えることで、総コマンドサイズが2バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1329】

図154は、TZ__RGETモジュールを説明するための説明図である。図154に示したTZ__RGETモジュールは、図32のステップS535で示した特別図柄乱数取得処理、すなわち、乱数値をレジスタに入力し、保留球数が上限数ではない場合のみ転送用バッファにレジスタの乱数値を格納する処理を実行する。

【1330】

図154(a)の1行目の指標「TZ__RGET:」は、当該TZ__RGETモジュールの先頭アドレスを示す。大当たり決定乱数取得処理を実行し、特別図柄保留球数更新処理として、2行目のコマンド「LD A, (DE)」によって、DEレジスタで示されるアドレスに格納された値(対象特別図柄保留球数カウンタのカウンタ値)をAレジスタに読み出す。なお、当該TZ__RGETモジュールが呼び出されるときにDEレジスタには予め対象特別図柄保留球数カウンタのカウンタ値を示すアドレスが格納されているとする。

20

【1331】

そして、図154(a)の3行目のコマンド「CP A, @TZ__MEM__MAX」によって、Aレジスタの値と、特別図柄保留球数上限値を示す固定値「@TZ__MEM__MAX」、ここでは、「4」とを比較する。すなわち、Aレジスタの値が4未満であれば、キャリーフラグが立って1となる。そして、4行目の「RET NC」によって、キャリーフラグが1でなければ、1段上のルーチンに戻る。すなわち、対象特別図柄保留球数カウンタのカウンタ値が4以上であれば、それ以上、対象特別図柄保留球数カウンタを計数することはないので、当該TZ__RGETモジュールを終了する。なお、キャリーフラグが1であれば、変動パターン乱数取得処理、リーチグループ決定乱数取得処理、リーチモード決定乱数取得処理、転送先アドレス算定処理、記憶エリア格納処理、取得時演出判定処理、および、特図保留指定コマンドセット処理を実行して、1段上のルーチンに戻る。

30

【1332】

ここで、図148の置き換えを行うと、図154(a)のコマンド群を図154(b)のように変更することができる。図154(b)の1行目の指標「TZ__RGET:」は、当該TZ__RGETモジュールの先頭アドレスを示す。大当たり決定乱数取得処理を実行し、特別図柄保留球数更新処理として、2行目のコマンド「LD A, (DE)」によって、DEレジスタで示されるアドレス(すなわちR__FZ__MEM)に格納された値(対象特別図柄保留球数カウンタのカウンタ値)をAレジスタに読み出す。

40

【1333】

そして、図154(b)の3行目のコマンド「RCP NC, A, @TZ__MEM__MAX」によって、Aレジスタの値と、特別図柄保留球数上限値を示す固定値「@TZ__MEM__MAX」、ここでは、「4」とを比較し、Aレジスタの値が4以上であれば、1段上のルーチンに戻る。すなわち、対象特別図柄保留球数カウンタのカウンタ値が4以上であれば、それ以上、対象特別図柄保留球数カウンタを計数することはないので、当該TZ__RGETモジュールを終了する。なお、Aレジスタの値が4未満であれば、変動パター

50

ン乱数取得処理、リーチグループ決定乱数取得処理、リーチモード決定乱数取得処理、転送先アドレス算定処理、記憶エリア格納処理、取得時演出判定処理、および、特図保留指定コマンドセット処理を実行して、1段上のルーチンに戻る。

【1334】

ここで、図154(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD A, (DE)」1バイト+3行目のコマンド「CP A, @TZ__MEM__MAX」2バイト+4行目の「RET NC」1バイト=4バイトとなり、総実行サイクルは、2行目2サイクル+3行目2サイクル+4行目3サイクル(1サイクル)=7サイクル(5サイクル)となる。一方、図154(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD A, (DE)」1バイト+3行目のコマンド「RCP NC, A, @TZ__MEM__MAX」2バイト=3バイトとなり、総実行サイクルは、2行目2サイクル+3行目5サイクル(3サイクル)=7サイクル(5サイクル)となる。したがって、図154(a)のコマンド群を図154(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1335】

図155は、TRSVSELモジュールを説明するための説明図である。図155に示したTRSVSELモジュールは、図37のステップS610-17で示される予備領域設定処理、すなわち、特別図柄当たりを確認し、特別図柄確率状態や特別図柄状態の予備設定処理を実行する。

【1336】

図155(a)の1行目の指標「TRSVSEL:」は、当該TRSVSELモジュールの先頭アドレスを示す。そして、状態オフセットチェックフラグ設定処理を実行すると、特別図柄当たり確認処理として、2行目のコマンド「RST DATSEL」によって、汎用モジュールであるDATSELモジュールが呼び出され、制御データの選択結果がAレジスタに格納される。

【1337】

図155(a)の3行目のコマンド「CP A, @TZ__BHT」によって、Aレジスタの値と、特別図柄大当たり情報を示す固定値「@TZ__BHT」、ここでは、01Hとを比較し、01Hと等しければ、ゼロフラグが立って1となる。そして、4行目の「RET NZ」によって、ゼロフラグが1でなければ、1段上のルーチンに戻る。すなわち、大当たり確定中でなければ、当該TRSVSELモジュールを終了する。なお、ゼロフラグが1であれば、特別図柄確率状態予備設定処理、および、特別図柄状態予備設定処理を実行して、1段上のルーチンに戻る。

【1338】

ここで、図148の置き換えを行うと、図155(a)のコマンド群を図155(b)のように変更することができる。図155(b)の1行目の指標「TRSVSEL:」は、当該TRSVSELモジュールの先頭アドレスを示す。そして、状態オフセットチェックフラグ設定処理を実行すると、特別図柄当たり確認処理として、2行目のコマンド「RST DATSEL」によって、DATSELモジュールが呼び出され、制御データの選択結果がAレジスタに格納される。

【1339】

図155(b)の3行目のコマンド「RCP NZ, A, @TZ__BHT」によって、Aレジスタの値と、特別図柄大当たり情報を示す固定値「@TZ__BHT」、ここでは、01Hとを比較し、01Hと等しくなければ、1段上のルーチンに戻る。すなわち、大当たり確定中でなければ、当該TRSVSELモジュールを終了する。なお、01Hと等しければ、特別図柄確率状態予備設定処理、および、特別図柄状態予備設定処理を実行して、1段上のルーチンに戻る。

【1340】

ここで、図155(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド

「RST DATSEL」1バイト+3行目のコマンド「C P A, @TZ__BHT」2バイト+4行目の「RET NZ」1バイト=4バイトとなり、総実行サイクルは、2行目4サイクル+3行目2サイクル+4行目3サイクル(1サイクル)=9サイクル(7サイクル)となる。一方、図155(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「RST DATSEL」1バイト+3行目のコマンド「RCP NZ, A, @TZ__BHT」2バイト=3バイトとなり、総実行サイクルは、2行目4サイクル+3行目5サイクル(3サイクル)=9サイクル(7サイクル)となる。したがって、図155(a)のコマンド群を図155(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1341】

図156は、TDOVCHKモジュールを説明するための説明図である。図156に示したTDOVCHKモジュールは、図34のステップS540で示したカウントスイッチ通過処理(大入賞口通過処理)において大入賞口入球コマンドをセット(S540-5)した後に実行する大入賞口過剰入賞監視処理、すなわち、大入賞口入賞球数カウンタや大入賞口過剰入賞回数カウンタの更新処理を実行する。

【1342】

図156(a)の1行目の指標「TDOVCHK:」は、当該TDOVCHKモジュールの先頭アドレスを示す。そして、特別電動役物遊技管理フェーズ確認処理として、2行目のコマンド「LDQ A, (LOW R__TDN__PHS)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__TDN__PHS」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(特別電動役物遊技管理フェーズ)をAレジスタに読み出す。

20

【1343】

図156(a)の3行目のコマンド「JCP NC, A, @TD__PRE__SHT, TDOVCHK__10」によって、Aレジスタの値と固定値「@TD__PRE__SHT」、ここでは、05Hと比較し、その結果、05H以上であれば、アドレス「TDOVCHK__10」に移動する。こうして、Aレジスタの値が小当たり大入賞口開放前状態指定値より後の指定値であれば、以降の処理を省略して6行目の指標「TDOVCHK__10」に移動することができる。4行目のコマンド「C P A, @TD__OPN__BHT」によって、Aレジスタの値と、大当たり大入賞口開放制御状態指定値を示す固定値「@TD__OPN__BHT」、ここでは、2とを比較し、等しければ、ゼロフラグが立って1となる。そして、5行目の「RET Z」によって、ゼロフラグが1であれば、1段上のルーチンに戻る。すなわち、Aレジスタの値が大当たり大入賞口開放制御状態指定値であれば、当該TDOVCHKモジュールを終了する。なお、ゼロフラグが1でなければ、大入賞口入賞球数カウンタ更新処理、大入賞口過剰入賞回数カウンタ更新処理、および、大入賞口過剰入賞エラー時処理を実行して、1段上のルーチンに戻る。

30

【1344】

ここで、図148の置き換えを行うと、図156(a)のコマンド群を図156(b)のように変更することができる。図156(b)の1行目の指標「TDOVCHK:」は、当該TDOVCHKモジュールの先頭アドレスを示す。そして、特別電動役物遊技管理フェーズ確認処理として、2行目のコマンド「LDQ A, (LOW R__TDN__PHS)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__TDN__PHS」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(特別電動役物遊技管理フェーズ)をAレジスタに読み出す。

40

【1345】

図156(b)の3行目のコマンド「JCP NC, A, @TD__PRE__SHT, TDOVCHK__10」によって、Aレジスタの値と固定値「@TD__PRE__SHT」、ここでは、05Hと比較し、その結果、05H以上であれば、アドレス「TDOVCHK__10」に移動する。こうして、Aレジスタの値が小当たり大入賞口開放前状態指定値よ

50

り後の指定値であれば、以降の処理を省略して5行目の指標「TDOVCHK__10」に移動することができる。4行目のコマンド「RCP Z, A, @TD__OPN__BHT」によって、Aレジスタの値と、大当たり大入賞口開放制御状態指定値を示す固定値「@TD__OPN__BHT」、ここでは、2とを比較し、等しければ、1段上のルーチンに戻る。すなわち、Aレジスタの値が大当たり大入賞口開放制御状態指定値であれば、当該TDOVCHKモジュールを終了する。なお、ゼロフラグが1でなければ、大入賞口入賞球数カウンタ更新処理、大入賞口過剰入賞回数カウンタ更新処理、および、大入賞口過剰入賞エラー時処理を実行して、1段上のルーチンに戻る。

【1346】

ここで、図156(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__TDN__PHS)」2バイト+3行目のコマンド「JCP NC, A, @TD__PRE__SHT, TDOVCHK__10」3バイト+4行目のコマンド「CP A, @TD__OPN__BHT」2バイト+5行目の「RET Z」1バイト=8バイトとなり、総実行サイクルは、2行目3サイクル+3行目4サイクル(または3サイクル)+4行目2サイクル+5行目3サイクル(1サイクル)=12サイクル(9サイクル)となる。一方、図156(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__TDN__PHS)」2バイト+3行目のコマンド「JCP NC, A, @TD__PRE__SHT, TDOVCHK__10」3バイト+4行目のコマンド「RCP Z, A, @TD__OPN__BHT」2バイト=7バイトとなり、総実行サイクルは、2行目3サイクル+3行目4サイクル(または3サイクル)+4行目5サイクル(3サイクル)=12サイクル(9サイクル)となる。したがって、図156(a)のコマンド群を図156(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1347】

図157は、BER__CHKモジュールを説明するための説明図である。図157に示したBER__CHKモジュールは、図26のステップS400-21で示した状態管理処理(エラー管理処理)のサブルーチンであるベース異常エラー監視処理、すなわち、ベース異常確認カウンタのカウンタ値を監視する処理を実行する。

【1348】

アウトスイッチ確認処理として、図157(a)の1行目の指標「BER__CHK:」は、当該BER__CHKモジュールの先頭アドレスを示す。2行目のコマンド「LDQ HL, LOW R__BAS__CNT」によって、Qレジスタの値をHレジスタに読み出し、アドレス「R__BAS__CNT」の下位1バイトの値をLレジスタに読み出す。3行目のコマンド「LD A, (HL)」によって、HLレジスタで示されるアドレス(すなわちR__BAS__CNT)に格納された値(ベース異常確認カウンタのカウンタ値)をAレジスタに読み出す。

【1349】

そして、図157(a)の4行目のコマンド「CP A, @BAS__CHK__CNT」によって、Aレジスタの値と、ベース異常検出球数を示す固定値「@BAS__CHK__CNT」ここでは、20とを比較する。すなわち、Aレジスタの値が20未満であれば、キャリーフラグが立って1となる。そして、5行目の「RET C」によって、キャリーフラグが1であれば、1段上のルーチンに戻る。すなわち、ベース異常確認カウンタのカウンタ値が20未満であれば、当該BER__CHKモジュールを終了する。なお、キャリーフラグが1でなければ、セキュリティ設定処理を実行して、1段上のルーチンに戻る。

【1350】

ここで、図148の置き換えを行うと、図157(a)のコマンド群を図157(b)のように変更することができる。図157(b)の1行目の指標「BER__CHK:」は、当該BER__CHKモジュールの先頭アドレスを示す。2行目のコマンド「LDQ HL, LOW R__BAS__CNT」によって、Qレジスタの値をHレジスタに読み出し、

アドレス「R__BAS__CNT」（ベース異常確認カウンタを格納するアドレス）の下位1バイトの値をLレジスタに読み出す。3行目のコマンド「LD A, (HL)」によって、HLレジスタで示されるアドレス（すなわちR__BAS__CNT）に格納された値（ベース異常確認カウンタのカウント値）をAレジスタに読み出す。

【1351】

そして、図157（b）の4行目のコマンド「RCP C, A, @BAS__CHK__CNT」によって、Aレジスタの値と、ベース異常検出球数を示す固定値「@BAS__CHK__CNT」ここでは、20とを比較し、Aレジスタの値が20未満であれば、1段上のルーチンに戻る。なお、Aレジスタの値が20以上であれば、セキュリティ設定処理を実行して、1段上のルーチンに戻る。

【1352】

ここで、図157（a）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ HL, LOW R__BAS__CNT」2バイト+3行目のコマンド「LD A, (HL)」1バイト+4行目のコマンド「CP A, @BAS__CHK__CNT」2バイト+5行目の「RET C」1バイト=6バイトとなり、総実行サイクルは、2行目の2サイクル+3行目2サイクル+4行目2サイクル+5行目3サイクル（1サイクル）=9サイクル（7サイクル）となる。一方、図157（b）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ HL, LOW R__BAS__CNT」2バイト+3行目のコマンド「LD A, (HL)」1バイト+4行目のコマンド「RCP C, A, @BAS__CHK__CNT」2バイト=5バイトとなり、総実行サイクルは、2行目2サイクル+3行目2サイクル+4行目5サイクル（3サイクル）=9サイクル（7サイクル）となる。したがって、図157（a）のコマンド群を図157（b）のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1353】

図158は、TEF__SELモジュールを説明するための説明図である。図158に示したTEF__SELモジュールは、図47のステップS720-9で示した大入賞口閉鎖有効時間設定処理のサブルーチンである大入賞口閉鎖有効時間選択処理、すなわち、大入賞口を閉鎖するまでの有効時間をテーブルから選択する処理を実行する。

【1354】

図158（a）の1行目の指標「TEF__SEL:」は、当該TEF__SELモジュールの先頭アドレスを示す。そして、大入賞口閉鎖有効時間データ選択処理として、2行目のコマンド「LDQ A, (LOW R__ZUG__CHK__FIX)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__ZUG__CHK__FIX」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別図柄判定フラグ）をAレジスタに読み出す。

【1355】

図158（a）の3行目のコマンド「CP A, @ZUG__SML1」によって、Aレジスタの値と、小当たり図柄1指定データを示す固定値「@ZUG__SML1」、ここでは、07Hとを比較し、07H未満であれば、キャリーフラグが立って1となる。そして、4行目の「RET NC」によって、キャリーフラグが1でなければ、1段上のルーチンに戻る。すなわち、Aレジスタの値が小当たり図柄1指定データ以上であれば、当該TEF__SELモジュールを終了する。なお、キャリーフラグが1であれば、ワードデータ選択処理を実行して、1段上のルーチンに戻る。

【1356】

ここで、図148の置き換えを行うと、図158（a）のコマンド群を図158（b）のように変更することができる。図158（b）の1行目の指標「TEF__SEL:」は、当該TEF__SELモジュールの先頭アドレスを示す。そして、大入賞口閉鎖有効時間データ選択処理として、2行目のコマンド「LDQ A, (LOW R__ZUG__CHK__

10

20

30

40

50

FIX)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_ZUG_CHK_FIX」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別図柄判定フラグ）をAレジスタに読み出す。

【1357】

図158(b)の3行目のコマンド「RCP NC, A, @ZUG_SML1」によって、Aレジスタの値と、小当たり図柄1指定データを示す固定値「@ZUG_SML1」、ここでは、07Hとを比較し、07H以上であれば、1段上のルーチンに戻る。すなわち、Aレジスタの値が小当たり図柄1指定データ以上であれば、当該TEF_SELモジュールを終了する。なお、Aレジスタの値が小当たり図柄1指定データ未満であれば、ワードデータ選択処理を実行して、1段上のルーチンに戻る。

10

【1358】

ここで、図158(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R_ZUG_CHK_FIX)」2バイト+3行目のコマンド「CP A, @ZUG_SML1」2バイト+4行目の「RET NC」1バイト=5バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目3サイクル(1サイクル)=8サイクル(6サイクル)となる。一方、図158(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R_ZUG_CHK_FIX)」2バイト+3行目のコマンド「RCP NC, A, @ZUG_SML1」2バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目5サイクル(3サイクル)=8サイクル(6サイクル)となる。したがって、図158(a)のコマンド群を図158(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

20

【1359】

なお、かかる図148で示した置き換えは、パチンコ機のみならず、スロットマシンにおいても、複数のモジュールからサブルーチンとして呼び出される。例えば、図92のステップS2500-35の滑りコマ数取得処理における逆押しデータ設定処理を実行するSET_RIGモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE_SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、準備中処理(AT状態=「5」)を実行するPRE_LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE_SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、REG中処理(AT状態=「6」)を実行するREG_LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE_SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、BIG中処理(AT状態=「7」)を実行するBIT_LOTモジュールにおけるBIGストック数抽せん処理を実行するBIG_SLTモジュール、図91のステップS2400-11で示した実行フラグ設定処理において指示情報設定処理を実行するNAV_SETモジュール等から呼び出される。

30

【1360】

図159は、SET_RIGモジュールを説明するための説明図である。図159に示したSET_RIGモジュールは、逆押しデータ設定処理、すなわち、停止制御に必要なデータを設定する処理を実行する。

40

【1361】

図159(a)の1行目の指標「SET_RIG:」は、当該SET_RIGモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW __HIT_NUM)」によって、Qレジスタの値をアドレスの上位1バイトとし、停止制御番号を格納するアドレス「__HIT_NUM」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（停止制御番号）をAレジスタに読み出す。

【1362】

50

そして、第2停止操作時使用ビットデータ更新処理として、図159(a)の3行目のコマンド「SUB A, 10」によって、Aレジスタの値から固定値「10」を減算する。そして、4行目のコマンド「CP A, 57-10+1」によって、減算されたAレジスタの値と48(57-10+1)とを比較する。ここでは、停止制御番号が10~57の範囲に含まれることを確認している。すなわち、Aレジスタの値が10未満であれば、コマンド「SUB A, 10」によってAレジスタの値は負の値になり、1バイト値としてみると48より大きくなる。したがって、減算されたAレジスタの値が10未満であれば、コマンド「CP A, 57-10+1」によってキャリーフラグが立って1となる。そして、5行目の「RET NC」によって、キャリーフラグが1でなければ、1段上のルーチンに戻る。なお、キャリーフラグが1であれば(停止制御番号が10~57の範囲に含まれれば)、次の処理を実行する。

10

【1363】

続いて、図159(a)の6行目のコマンド「LDQ A, (LOW __HIT__NUM)」によって、Qレジスタの値をアドレスの上位1バイトとし、停止制御番号を格納するアドレス「__HIT__NUM」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(停止制御番号)をAレジスタに読み出す。7行目のコマンド「CP A, 34」によって、Aレジスタの値と固定値「34」を比較している。ここでは、停止制御番号が34未満であることを確認している。Aレジスタの値が34未満であれば、コマンド「CP A, 34」によってキャリーフラグが立って1となる。そして、8行目の「RET NC」によって、キャリーフラグが1でなければ、1段上のルーチンに戻る。なお、キャリーフラグが1であれば(停止制御番号が34未満であれば)、残りの処理を実行して1段上のルーチンに戻る。

20

【1364】

ここで、図148の置き換えを行うと、図159(a)のコマンド群を図159(b)のように変更することができる。図159(b)の1行目の指標「SET__RIG:」は、当該SET__RIGモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW __HIT__NUM)」によって、Qレジスタの値をアドレスの上位1バイトとし、停止制御番号を格納するアドレス「__HIT__NUM」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(停止制御番号)をAレジスタに読み出す。

30

【1365】

そして、第2停止操作時使用ビットデータ更新処理として、図159(b)の3行目のコマンド「SUB A, 10」によって、Aレジスタの値から固定値「10」を減算する。4行目のコマンド「RCP NC, A, 57-10+1」によって、減算されたAレジスタの値と48(57-10+1)とを比較し、キャリーフラグが1でなければ、1段上のルーチンに戻る。ここでは、停止制御番号が10~57の範囲に含まれることを確認し、その範囲に入っていなければ、それ以降の処理を行わず1段上のルーチンに戻っている。なお、キャリーフラグが1であれば(停止制御番号が10~57の範囲に含まれれば)、次の処理を実行する。

【1366】

40

続いて、図159(b)の5行目のコマンド「LDQ A, (LOW __HIT__NUM)」によって、Qレジスタの値をアドレスの上位1バイトとし、停止制御番号を格納するアドレス「__HIT__NUM」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(停止制御番号)をAレジスタに読み出す。6行目のコマンド「RCP NC, A, 34」によって、Aレジスタの値と固定値「34」を比較し、キャリーフラグが1でなければ、1段上のルーチンに戻る。ここでは、停止制御番号が34未満であることを確認し、34以上であれば、それ以降の処理を行わず1段上のルーチンに戻っている。なお、キャリーフラグが1であれば(停止制御番号が34未満であれば)、残りの処理を実行して1段上のルーチンに戻る。

【1367】

50

ここで、図159(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW __HIT__NUM)」2バイト+3行目のコマンド「SUB A, 10」2バイト+4行目のコマンド「CP A, 57-10+1」2バイト+5行目の「RET NC」1バイト+6行目のコマンド「LDQ A, (LOW __HIT__NUM)」2バイト+7行目のコマンド「CP A, 34」2バイト+8行目の「RET NC」1バイト=12バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目2サイクル+5行目3サイクル(1サイクル)+6行目3サイクル+7行目2サイクル+8行目3サイクル(1サイクル)=18サイクル(14サイクル)となる。一方、図159(b)に示したコマンド群の総コマンドサイズは、2行目のコマンドコマンド「LDQ A, (LOW __HIT__NUM)」2バイト+3行目のコマンド「SUB A, 10」2バイト+4行目のコマンド「RCP NC, A, 57-10+1」2バイト+5行目のコマンドコマンド「LDQ A, (LOW __HIT__NUM)」2バイト+6行目のコマンド「RCP NC, A, 34」2バイト=10バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目5サイクル(3サイクル)+5行目3サイクル+6行目5サイクル(3サイクル)=18サイクル(14サイクル)となる。したがって、図159(a)のコマンド群を図159(b)のコマンド群に置き換えることで、総コマンドサイズが2バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1368】

10

図160は、PRE__LOTモジュールを説明するための説明図である。図160に示したPRE__LOTモジュールは、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、準備中処理、すなわち、AT状態が所定の値(例えば5)の場合の抽せん処理を実行する。

20

【1369】

図160(a)の1行目の指標「PRE__LOT:」は、当該PRE__LOTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW __TRG__KND)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__TRG__KND」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(トリガー役種別)をAレジスタに読み出す。

30

【1370】

図160(a)の3行目のコマンド「CP A, @LOT__OFS__1FK」によって、Aレジスタの値と、トリガー役種別として「フェイク1枚」を示す固定値「@LOT__OFS__1FK」、ここでは、3とを比較し、3未満であれば、キャリーフラグが立って1となる。そして、4行目の「RET C」によって、キャリーフラグが1であれば、1段上のルーチンに戻る。すなわち、Aレジスタの値が「フェイク1枚」を示す固定値未満であれば、当該PRE__LOTモジュールを終了する。なお、キャリーフラグが0であれば、REG開始時設定処理、BIG開始時BIGストック抽せん処理、BIG開始時設定処理を実行して、1段上のルーチンに戻る。

【1371】

40

ここで、図148の置き換えを行うと、図160(a)のコマンド群を図160(b)のように変更することができる。図160(b)の1行目の指標「PRE__LOT:」は、当該PRE__LOTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW __TRG__KND)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__TRG__KND」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(トリガー役種別)をAレジスタに読み出す。

【1372】

図160(b)の3行目のコマンド「RCP C, A, @LOT__OFS__1FK」によって、Aレジスタの値と、トリガー役種別として「フェイク1枚」を示す固定値「@LOT__OFS__1FK」、ここでは、3とを比較し、3未満であれば、1段上のルーチン

50

に戻る。すなわち、Aレジスタの値が「フェイク1枚」を示す固定値未満であれば、当該PRE__LOTモジュールを終了する。

【1373】

ここで、図160(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW __TRG__KND)」2バイト+3行目のコマンド「CP A, @LOT__OFS__1FK」2バイト+3行目の「RET C」1バイト=5バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目3サイクル(1サイクル)=8サイクル(6サイクル)となる。一方、図160(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW __TRG__KND)」2バイト+3行目のコマンド「RCP C, A, @LOT__OFS__1FK」2バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目5サイクル(3サイクル)=8サイクル(6サイクル)となる。したがって、図160(a)のコマンド群を図160(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1374】

図161は、REG__LOTモジュールを説明するための説明図である。図161に示したREG__LOTモジュールは、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、REG中処理、すなわち、AT状態が所定の値(例えば6)の場合の抽せん処理を実行する。

【1375】

図161(a)の1行目の指標「REG__LOT:」は、当該REG__LOTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW __TRG__KND)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__TRG__KND」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(トリガー役種別)をAレジスタに読み出す。

【1376】

図161(a)の3行目のコマンド「CP A, @LOT__OFS__PDJ」によって、Aレジスタの値と、トリガー役種別として「打順ベル」を示す固定値「@LOT__OFS__PDJ」、ここでは、5とを比較し、5であれば、ゼロフラグが立って1となる。そして、4行目の「RET NZ」によって、ゼロフラグが1でなければ、1段上のルーチンに戻る。すなわち、Aレジスタの値が「打順ベル」を示す固定値でなければ、当該REG__LOTモジュールを終了する。なお、ゼロフラグが0であれば、REG残りナビ回数減算処理、REG終了時設定処理を実行して、1段上のルーチンに戻る。

【1377】

ここで、図148の置き換えを行うと、図161(a)のコマンド群を図161(b)のように変更することができる。図161(b)の1行目の指標「REG__LOT:」は、当該REG__LOTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW __TRG__KND)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__TRG__KND」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(トリガー役種別)をAレジスタに読み出す。

【1378】

図161(b)の3行目のコマンド「RCP NZ, A, @LOT__OFS__PDJ」によって、Aレジスタの値と、トリガー役種別として「打順ベル」を示す固定値「@LOT__OFS__PDJ」、ここでは、5とを比較し、5でなければ、1段上のルーチンに戻る。すなわち、Aレジスタの値が「打順ベル」を示す固定値でなければ、当該REG__LOTモジュールを終了する。

【1379】

ここで、図161(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド

「LDQ A, (LOW __TRG__KND)」2バイト+3行目のコマンド「CP A, @LOT__OFS__PDJ」2バイト+4行目の「RET NZ」1バイト=5バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目3サイクル(1サイクル)=8サイクル(6サイクル)となる。一方、図161(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW __TRG__KND)」2バイト+3行目のコマンド「RCP NZ, A, @LOT__OFS__PDJ」2バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目5サイクル(3サイクル)=8サイクル(6サイクル)となる。したがって、図161(a)のコマンド群を図161(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1380】

図162は、BIG__SLTモジュールを説明するための説明図である。図162に示したBIG__SLTモジュールは、BIGストック数抽せん処理を実行する。

【1381】

図162(a)の1行目の指標「BIG__SLT:」は、当該BIG__SLTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW __TRG__KND)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__TRG__KND」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(トリガー役種別)をAレジスタに読み出す。

20

【1382】

図162(a)の3行目のコマンド「CP A, @LOT__OFS__LCY」によって、Aレジスタの値と、トリガー役種別として「弱チェリー」を示す固定値「@LOT__OFS__LCY」、ここでは、6とを比較し、6未満であれば、キャリーフラグが立って1となる。そして、4行目の「RET C」によって、キャリーフラグが1であれば、1段上のルーチンに戻る。すなわち、Aレジスタの値が「弱チェリー」を示す固定値未満であれば、当該BIG__SLTモジュールを終了する。なお、キャリーフラグが0であれば、BIG中ストック抽せん処理を実行して、1段上のルーチンに戻る。

【1383】

ここで、図148の置き換えを行うと、図162(a)のコマンド群を図162(b)のように変更することができる。図162(b)の1行目の指標「BIG__SLT:」は、当該BIG__SLTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ A, (LOW __TRG__KND)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__TRG__KND」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(トリガー役種別)をAレジスタに読み出す。

30

【1384】

図162(b)の3行目のコマンド「RCP C, A, @LOT__OFS__LCY」によって、Aレジスタの値と、トリガー役種別として「弱チェリー」を示す固定値「@LOT__OFS__LCY」、ここでは、6とを比較し、6未満であれば、1段上のルーチンに戻る。すなわち、Aレジスタの値が「弱チェリー」を示す固定値未満であれば、当該BIG__SLTモジュールを終了する。

40

【1385】

ここで、図162(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW __TRG__KND)」2バイト+3行目のコマンド「CP A, @LOT__OFS__LCY」2バイト+4行目の「RET C」1バイト=5バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目3サイクル(1サイクル)=8サイクル(6サイクル)となる。一方、図162(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW __TRG__KND)」2バイト+3行目のコマンド「RCP C, A, @LOT__OFS__LCY」2バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目5サイクル(3サイクル

50

) = 8 サイクル (6 サイクル) となる。したがって、図 1 6 2 (a) のコマンド群を図 1 6 2 (b) のコマンド群に置き換えることで、総コマンドサイズが 1 バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1 3 8 6】

図 1 6 3 は、NAV__SET モジュールを説明するための説明図である。図 1 6 3 に示した NAV__SET モジュールは、指示情報設定処理、すなわち、指示情報種別の設定処理を実行する。

【1 3 8 7】

図 1 6 3 (a) の 1 行目の指標「NAV__SET:」は、当該 NAV__SET モジュールの先頭アドレスを示す。2 行目のコマンド「LDQ A, (LOW __YRI__LMP)」によって、Q レジスタの値をアドレスの上位 1 バイトとし、アドレス「__YRI__LMP」の下位 1 バイトの値をアドレスの下位 1 バイトとし、そのアドレスに格納された値 (有効ランプフラグ) を A レジスタに読み出す。

【1 3 8 8】

図 1 6 3 (a) の 3 行目のコマンド「CP A, @AT__MOD__PRE」によって、A レジスタの値と、AT 状態定義が「準備中」を示す固定値「@AT__MOD__PRE」、ここでは、5 とを比較し、5 未満であれば、キャリーフラグが立って 1 となる。そして、4 行目の「RET C」によって、キャリーフラグが 1 であれば、1 段上のルーチンに戻る。すなわち、A レジスタの値が「準備中」を示す固定値未満であれば、当該 NAV__SET モジュールを終了する。なお、キャリーフラグが 0 であれば、残りの処理を実行して、1 段上のルーチンに戻る。

【1 3 8 9】

ここで、図 1 4 8 の置き換えを行うと、図 1 6 3 (a) のコマンド群を図 1 6 3 (b) のように変更することができる。図 1 6 3 (b) の 1 行目の指標「NAV__SET:」は、当該 NAV__SET モジュールの先頭アドレスを示す。2 行目のコマンド「LDQ A, (LOW __YRI__LMP)」によって、Q レジスタの値をアドレスの上位 1 バイトとし、アドレス「__YRI__LMP」の下位 1 バイトの値をアドレスの下位 1 バイトとし、そのアドレスに格納された値 (有効ランプフラグ) を A レジスタに読み出す。

【1 3 9 0】

図 1 6 3 (b) の 3 行目のコマンド「RCP C, A, @AT__MOD__PRE」によって、A レジスタの値と、AT 状態定義が「準備中」を示す固定値「@AT__MOD__PRE」、ここでは、5 とを比較し、5 未満であれば、1 段上のルーチンに戻る。すなわち、A レジスタの値が「準備中」を示す固定値未満であれば、当該 NAV__SET モジュールを終了する。

【1 3 9 1】

ここで、図 1 6 3 (a) に示したコマンド群の総コマンドサイズは、2 行目のコマンド「LDQ A, (LOW __YRI__LMP)」2 バイト + 3 行目のコマンド「CP A, @AT__MOD__PRE」2 バイト + 4 行目の「RET C」1 バイト = 5 バイトとなり、総実行サイクルは、2 行目 3 サイクル + 3 行目 2 サイクル + 4 行目 3 サイクル (1 サイクル) = 8 サイクル (6 サイクル) となる。一方、図 1 6 3 (b) に示したコマンド群の総コマンドサイズは、2 行目のコマンド「LDQ A, (LOW __YRI__LMP)」2 バイト + 3 行目のコマンド「RCP C, A, @AT__MOD__PRE」2 バイト = 4 バイトとなり、総実行サイクルは、2 行目 3 サイクル + 3 行目 5 サイクル (3 サイクル) = 8 サイクル (6 サイクル) となる。したがって、図 1 6 3 (a) のコマンド群を図 1 6 3 (b) のコマンド群に置き換えることで、総コマンドサイズが 1 バイト削減される。かかる置き換えによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1 3 9 2】

〈XORQ〉

図164は、TOK_PRCモジュールの具体的な処理を示したフローチャートである。TOK_PRCモジュールは、上記ステップS600の特別遊技管理処理（図36参照）を実行する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1393】

メインCPU300aは、図164のように、特別遊技特別図柄判定フラグをロードし（S1）、ロードした特別遊技特別図柄判定フラグを反転し（S2）、反転した特別遊技特別図柄判定フラグをセーブする（S3）。なお、ステップS1は、図36におけるステップS600-5に相当し、ステップS2は、図36におけるステップS600-7に相当し、ステップS3は、図36におけるステップS600-9に相当する。

10

【1394】

図165は、TOK_PRCモジュールを実現するためのコマンドの一例を説明するための説明図である。図164で示したフローチャートは、例えば、図165に示したプログラムによって実現される。

【1395】

図165（a）の1行目の指標「TOK_PRC:」は、当該TOK_PRCモジュールの先頭アドレスを示す。そして、特別遊技管理処理として、図165（a）の2行目のコマンド「LDQ A, (LOW R_STA_TZ)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_STA_TZ」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別遊技特別図柄判定フラグ）をAレジスタに読み出す。かかる2行目のコマンドが、図164のステップS1に対応する。3行目のコマンド「XOR 001H」によって、読み出した値（特別遊技特別図柄判定フラグ）と固定値（ここでは001H）との排他的論理和を計算する。これにより、特別遊技特別図柄判定フラグが反転される。かかる2行目のコマンドが、図164のステップS2に対応する。そして、4行目のコマンド「LDQ (LOW R_STA_TZ), A」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_STA_TZ」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに、Aレジスタの値を格納する。かかる4行目のコマンドが、図164のステップS3に対応する。

20

【1396】

ここで、図165（a）のコマンド群を図165（b）のように変更することができる。図165（b）の1行目の指標「TOK_PRC:」は、当該TOK_PRCモジュールの先頭アドレスを示す。そして、特別遊技管理処理を実行すると、2行目のコマンド「XORQ (LOW R_STA_TZ), 001H」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_STA_TZ」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別遊技特別図柄判定フラグ）と固定値（ここでは001H）との排他的論理和を計算し、計算結果を同アドレスに格納する。

30

【1397】

ここで、図165（a）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R_STA_TZ)」2バイト+3行目コマンド「XOR 001H」2バイト+4行目のコマンド「LDQ (LOW R_STA_TZ), A」2バイト=6バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目3サイクル=8サイクルとなる。一方、図165（b）に示したコマンド群の総コマンドサイズは、「XORQ (LOW R_STA_TZ), 001H」4バイト=4バイトとなり、総実行サイクルは、2行目7サイクル=7サイクルとなる。したがって、図165（a）のコマンド群を図165（b）のコマンド群に置き換えることで、総コマンドサイズが2バイト削減され、総実行サイクルも少なくとも1サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

40

【1398】

また、図165（a）と図165（b）とを比較して理解できるように、図165（a）

50

）において3行（2行目～4行目）を占有していたコマンド群を、図165（b）においては1行（2行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1399】

また、ここでは、図165（b）の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図165（a）の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があるが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1400】

〈CPQ〉

図166は、TEF__SELモジュールの具体的な処理を示したフローチャートである。TOK__PRCモジュールは、大入賞口開放制御処理（図47参照）において、大入賞口閉鎖有効時間を特別電動役物遊技タイマに設定する際（ステップS720-9）に実行される。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1401】

メインCPU300aは、図166のように、小当たり遊技の大入賞口閉鎖有効時間をセットし（S1）、特別図柄判定フラグをロードし（S2）、ロードした特別図柄判定フラグと、小当たり図柄であることを示す所定の値（ここでは、007H）とを比較する（S3）。なお、特別図柄判定フラグは、ハズレ図柄である場合に00H、大当たり図柄である場合に01H～06Hのいずれか、小当たり図柄である場合に07H～09Hのいずれかが設定される。

【1402】

そして、比較結果として小当たり図柄であれば（S4におけるYES）、当該TEF__SELモジュールを終了し（S6）、比較結果として小当たり図柄でなければ（S4におけるNO）、すなわち、大当たり図柄であれば、大役遊技の大入賞口閉鎖有効時間テーブルをセットし（S5）、当該TEF__SELモジュールを終了する（S6）。

【1403】

図167は、TEF__SELモジュールを実現するためのコマンドの一例を説明するための説明図である。図166で示したフローチャートは、例えば、図167に示したプログラムによって実現される。

【1404】

図167（a）の1行目の指標「TEF__SEL:」は、当該TEF__SELモジュールの先頭アドレスを示す。そして、図167（a）の2行目のコマンド「LD HL, @TMR__TDN__EF3」によって、「@TMR__TDN__EF3」の値がHLレジスタに格納される。なお、「@TMR__TDN__EF3」には、小当たり遊技の大入賞口閉鎖有効時間が設定されている。かかる1行目のコマンドが、図166のステップS1に対応する。

【1405】

図167（a）の3行目のコマンド「LDQ A, (LOW R__ZUG__CHK__FIX)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__ZUG__CHK__FIX」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別図柄判定フラグ）をAレジスタに読み出す。かかる3行目のコマンドが、図166のステップS2に対応する。

【1406】

図167（a）の4行目のコマンド「CP A, @ZUG__SML1」によって、Aレジスタの値と、小当たり図柄1指定データ（小当たり図柄）を示す固定値「@ZUG__SML1」、ここでは、07Hとを比較し、Aレジスタの値が07H未満であれば、キャリーフラグが立って1となる。かかる4行目のコマンドが、図166のステップS3に対

10

20

30

40

50

応する。

【1407】

そして、図167(a)の5行目のコマンド「RET NC」によって、キャリーフラグが1でなければ、1段上のルーチンに戻る。すなわち、Aレジスタの値が小当たり図柄1指定データ以上であれば、当該TEF__SELモジュールを終了する。かかる5行目のコマンドが、図166のステップS4におけるYES、および、ステップS6に対応する。

【1408】

一方、キャリーフラグが1であれば、図167(a)の6行目のコマンド「LDQ A, (LOW R__TDN__FLG)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__TDN__FLG」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別電動役物指定フラグ）をAレジスタに読み出し、図167(a)の7行目のコマンド「LD HL,D__EFF__TMR-2」によって、転送元となる1バイトデータ群の先頭アドレス「D__EFF__TMR-2」をHLレジスタに設定する。

【1409】

図167(a)の8行目のコマンド「RST WORDSEL」によって、サブルーチンとしてWORDSELモジュールが呼び出され、HLレジスタにAレジスタの値を2回加算し、HLレジスタに示されるアドレスのうち、下位バイトの値がAレジスタに読み出された後、図167(a)の9行目のコマンド「RET」によって、1段上のルーチンに戻る。なお、その後の1段上のルーチンにおいて（その後の処理において）、大役遊技の大入賞口閉鎖有効時間がタイマにセットされることになる。かかる6行目から8行目が図166のステップS5に対応する。

【1410】

ここで、図167(a)のコマンド群を図167(b)のように変更することができる。ここでは、図167(a)と実質的に等しい処理についてはその説明を省略し、図167(b)の異なる処理のみを説明する。

【1411】

図167(b)の3行目のコマンド「CPQ (LOW R__ZUG__CHK__FIX),@ZUG__SML1」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__ZUG__CHK__FIX」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別図柄判定フラグ）と、小当たり図柄1指定データ（小当たり図柄）を示す固定値「@ZUG__SML1」、ここでは、07Hとを比較し、07H未満であれば、キャリーフラグが立って1となる。かかる3行目のコマンドが、図166のステップS2およびステップS3に対応する。かかるコマンドのコマンドサイズは「3」であり、実行サイクルは「4」である。

【1412】

ここで、図167(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD HL,@TMR__TDN__EF3」3バイト+3行目コマンド「LDQ A,(LOW R__ZUG__CHK__FIX)」2バイト+4行目のコマンド「CP A,@ZUG__SML1」2バイト+5行目の「RET NC」1バイト+6行目のコマンド「LDQ A,(LOW R__TDN__FLG)」2バイト+7行目のコマンド「LD HL,D__EFF__TMR-2」3バイト+8行目のコマンド「RST WORDSEL」1バイト+9行目「RET」1バイト=15バイトとなり、実行サイクルは、2行目3サイクル+3行目2サイクル+4行目2サイクル+5行目3サイクル（1サイクル）+6行目2サイクル+7行目3サイクル+8行目4サイクル+9行目3サイクル=22サイクル（20サイクル）となる。なお、括弧内のサイクル数は、コマンド「RET NC」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1413】

一方、図167(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD HL,@TMR__TDN__EF3」3バイト+3行目コマンド「CPQ (LOW

10

20

30

40

50

R_ZUG_CHK_FIX),@ZUG_SML1」3バイト+4行目の「RET NC」1バイト+5行目のコマンド「LDQ A,(LOW R_TDN_FLG)」2バイト+6行目のコマンド「LD HL,D_EFF_TMR-2」3バイト+7行目のコマンド「RST WORDSEL」1バイト+8行目「RET」1バイト=14バイトとなり、実行サイクルは、2行目3サイクル+3行目4サイクル+4行目3サイクル(1サイクル)+5行目2サイクル+6行目3サイクル+7行目4サイクル+8行目3サイクル=22サイクル(20サイクル)となる。なお、括弧内のサイクル数は、コマンド「RET NC」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1414】

したがって、図167(a)のコマンド群を図167(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1415】

また、図167(a)と図167(b)とを比較して理解できるように、図167(a)において2行(3行目、4行目)を占有していたコマンド群を、図167(b)においては1行(3行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1416】

また、ここでは、図167(b)の例では、3行目のコマンドにおいてAレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図167(a)の例では、3行目のコマンドを実行する際にAレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1417】

〈JTANDQ〉

図168は、SWI_PRCモジュールの具体的な処理を示したフローチャートである。SWI_PRCモジュールは、上記ステップS500のスイッチ管理処理(図28参照)を実行する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1418】

メインCPU300aは、図168のように、第1可変始動口検出スイッチ120Bsの検出時であるかを判定し(S1)、第1可変始動口検出スイッチ120Bsの検出時であれば(S1におけるYES)、第1始動口通過処理(S2)および普通電動役物入賞時確認処理(S3)を実行する。一方、第1可変始動口検出スイッチ120Bsの検出時でなければ(S1におけるNO)、第1始動口通過処理(S2)および普通電動役物入賞時確認処理(S3)がスキップされる。なお、ステップS1は、図28におけるステップS500-7に相当し、ステップS2は、図28におけるステップS520に相当し、ステップS3は、図28におけるステップS500-9に相当する。

【1419】

図169は、SWI_PRCモジュールを実現するためのコマンドの一例を説明するための説明図である。図168で示したフローチャートは、例えば、図169に示したプログラムによって実現される。

【1420】

図169(a)の1行目の指標「SWI_PRC:」は、当該SWI_PRCモジュールの先頭アドレスを示す。そして、図169(a)の2行目のコマンド「LDQ A,(LOW R_IN3_PON)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_IN3_PON」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値をAレジスタに読み出す。なお、ここで、同アドレスに格納された値には、例えば、最下位ビットに、第1固定始動口検出スイッチ120As(入

10

20

30

40

50

力部)の検出結果が保持され、下位2ビット目に、第2始動口検出スイッチ122s(入力部)の検出結果が保持され、下位3ビット目に、第1可変始動口検出スイッチ120Bs(入力部)の検出結果が保持される。これら各ビットは、検出スイッチによって遊技球が入球したことが検出された場合に「1」となり、検出スイッチによって遊技球が入球したことが検出されていない場合に「0」となる。

【1421】

3行目のコマンド「AND A,@IN3__ST2__BIT」によって、読み出した値と固定値「@IN3__ST2__BIT(ここでは00000100b)との論理積を計算する。これにより、読み出した値の下位3ビット目以外が0にされ(マスクされ)、読み出した値の下位3ビット目が0であればゼロフラグ=1となり、読み出した値の下位3ビット目が1であればゼロフラグ=0となる。つまり、第1可変始動口検出スイッチ120Bsの検出結果が1であれば(遊技球の入球が検出されていれば)ゼロフラグ=0となり、第1可変始動口検出スイッチ120Bsの検出結果が0であれば(遊技球の入球が検出されていなければ)ゼロフラグ=0となる。

【1422】

そして、4行目のコマンド「JR Z,SWI__PRC__20」によって、ゼロフラグが1(Z)であれば、SWI__PRC__20(7行目)に移動する。かかる2行目~4行目のコマンドが、図168のステップS1に対応する。

【1423】

一方、4行目のコマンド「JR Z,SWI__PRC__20」によって、ゼロフラグが1でなければ、5行目のコマンド「CALLF STA__PAS」によって、サブルーチンとしてSTA__PASモジュールが呼び出され、STA__PASモジュールにより第1始動口通過処理が実行される。かかる5行目のコマンドが、図168のステップS2に対応する。また、6行目のコマンド「CALLF FDN__CHK」によって、サブルーチンとしてFDN__CHKモジュールが呼び出され、FDN__CHKモジュールにより普通電動役物入賞時確認処理が実行される。かかる6行目のコマンドが、図168のステップS3に対応する。

【1424】

ここで、図169(a)のコマンド群を図169(b)のように変更することができる。ここでは、図169(a)と実質的に等しい処理についてはその説明を省略し、図169(b)の異なる処理のみを説明する。

【1425】

図169(b)の2行目のコマンド「JTANDQ Z,(LOW R__IN3__PON),@IN3__ST2__BIT,SWI__PRC__20」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__IN3__PON」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値と、固定値「@IN3__ST2__BIT」との論理積を計算し、ゼロフラグが1であればSWI__PRC__20に移動する。かかる2行目のコマンドが、図168のステップS1に対応する。かかるコマンドのコマンドサイズは「4」であり、実行サイクルは「5(6)」である。なお、括弧内のサイクル数は、ゼロフラグが1でありSWI__PRC__20に移動した場合の実行サイクルを示している。

【1426】

ここで、図169(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A,(LOW R__IN3__PON)」2バイト+3行目コマンド「AND A,@IN3__ST2__BIT」2バイト+4行目のコマンド「JR Z,SWI__PRC__20」2バイト+5行目のコマンド「CALLF STA__PAS」2バイト+6行目のコマンド「CALLF FDN__CHK」2バイト=10バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目2(3)サイクル+5行目4サイクル+6行目4サイクル=15(16)サイクルとなる。なお、括弧内のサイクル数は、ゼロフラグが1でありSWI__PRC__20に移動した場合の実行サイクルを示している。

10

20

30

40

50

【1427】

一方、図169 (b) に示したコマンド群の総コマンドサイズは、2行目のコマンド「JTANDQ Z, (LOW R__IN3__PON), @IN3__ST2__BIT, SWI__RC__20」4バイト+3行目のコマンド「CALLF STA__PAS」2バイト+4行目のコマンド「CALLF FDN__CHK」2バイト=8バイトとなり、総実行サイクルは、2行目5 (6) サイクル+3行目4サイクル+4行目4サイクル=13 (14) サイクルとなる。

【1428】

したがって、図169 (a) のコマンド群を図169 (b) のコマンド群に置き換えることで、総コマンドサイズが2バイト削減され、総実行サイクルも少なくとも2サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1429】

また、図169 (a) と図169 (b) とを比較して理解できるように、図169 (a) において3行 (2行目~4行目) を占有していたコマンド群を、図169 (b) においては1行 (2行目) で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1430】

また、図169 (b) の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図169 (a) の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

20

【1431】

〈OUT〉

図170は、CPUINITモジュールの具体的な処理を示したフローチャートである。CPUINITモジュールは、上記CPU初期化処理 (図22参照) を実行する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1432】

メインCPU300aは、図170のように、RAM300cへのアクセスを許可するアクセス許可処理を実行する (S1)。なお、ステップS1は、図22におけるステップS100-9に相当する。

30

【1433】

図171は、CPUINITモジュールを実現するためのコマンドの一例を説明するための説明図である。図170で示したフローチャートは、例えば、図171に示したプログラムによって実現される。

【1434】

図171 (a) の1行目の指標「CPUINIT:」は、当該CPUINITモジュールの先頭アドレスを示す。そして、図171 (a) の2行目のコマンド「LD A, (@RAMENBL)」によって、アクセスを許可することを示す固定値「@RAMENBL」 (ここでは、1) をAレジスタに読み出す。そして、図171 (a) の3行目のコマンド「OUT (@RAP____), A」によって、Uレジスタの値をアドレスの上位1バイトとし、RAMアクセスプロテクトレジスタポート (内部レジスタI/Oポート) のアドレスの下位1バイトを示す固定値「@RAP____」を下位1バイトとし、そのアドレスに、Aレジスタの値 (1) を出力する。なお、Uレジスタには、RAMアクセスプロテクトレジスタポート (内部レジスタI/Oポート) のアドレスの上位1バイト (例えば「FEH」) が予め設定されている。

40

【1435】

ここで、図171 (a) のコマンド群を図171 (b) のように変更することができる。ここでは、図171 (a) と実質的に等しい処理についてはその説明を省略し、図17

50

1 (b) の異なる処理のみを説明する。

【1436】

図171 (b) の2行目のコマンド「OUT (@RAP____), @RAMENBL」によって、Uレジスタの値をアドレスの上位1バイトとし、RAMアクセスプロテクトレジスタポート (内部レジスタI/Oポート) のアドレスの下位1バイトを示す固定値「@RAP____」を下位1バイトとし、そのアドレスに、固定値「@RAMENBL」、すなわち、1を出力する。かかるコマンドのコマンドサイズは「3」であり、実行サイクルは「4」である。

【1437】

ここで、図171 (a) に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD A, (@RAMENBL)」2バイト+3行目コマンド「OUT (@RAP____), A」2バイト=4バイトとなり、総実行サイクルは、2行目2サイクル+3行目3サイクル=5サイクルとなる。

10

【1438】

一方、図171 (b) に示したコマンド群の総コマンドサイズは、2行目のコマンド「OUT (@RAP____), @RAMENBL」3バイト=3バイトとなり、総実行サイクルは、2行目4サイクル=4サイクルとなる。

【1439】

したがって、図171 (a) のコマンド群を図171 (b) のコマンド群に置き換えることで、総コマンドサイズが1バイト削減され、総実行サイクルも少なくとも1サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

20

【1440】

また、図171 (a) と図171 (b) とを比較して理解できるように、図171 (a) において2行 (2行目~3行目) を占有していたコマンド群を、図171 (b) においては1行 (2行目) で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1441】

また、図171 (b) の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図171 (a) の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

30

【1442】

図172 は、TMR_IPTモジュールを説明するための説明図である。図172 に示したTMR_IPTモジュールは、図98 に示したタイマ割込み処理を実行する。

【1443】

図172 (a) の1行目の指標「TMR_IPT:」は、当該TMR_IPTモジュールの先頭アドレスを示す。そして、図172 (a) の2行目のコマンド「LD A, @TOCRVAL」によって、アクセスを許可することを示す固定値「@TOCRVAL」 (ここでは、1) をAレジスタに読み出す。そして、図172 (a) の3行目のコマンド「OUT (@PTOCR____), A」によって、Uレジスタの値をアドレスの上位1バイトとし、RAMアクセスプロテクトレジスタポート (内部レジスタI/Oポート) のアドレスの下位1バイトを示す固定値「@PTOCR____」をアドレスの下位1バイトとし、そのアドレスに、Aレジスタの値 (1) を出力する。なお、Uレジスタには、RAMアクセスプロテクトレジスタポート (内部レジスタI/Oポート) のアドレスの上位1バイト (例えば「FEH」) が予め設定されている。

40

【1444】

ここで、図172 (a) のコマンド群を図172 (b) のように変更することができる。ここでは、図172 (a) と実質的に等しい処理についてはその説明を省略し、図17

50

2 (b) の異なる処理のみを説明する。

【1445】

図172 (b) の2行目のコマンド「OUT (@PTOCR___),@TOCRVAL」によって、Uレジスタの値をアドレスの上位1バイトとし、RAMアクセスプロテクトレジスタポート (内部レジスタE/Oポート) のアドレスの下位1バイトを示す固定値「@PTOCR___」をアドレスの下位1バイトとし、そのアドレスに、固定値「@TOCRVAL」、すなわち、1を出力する。かかるコマンドのコマンドサイズは「3」であり、実行サイクルは「4」である。

【1446】

ここで、図172 (a) に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD A, @TOCRVAL」2バイト+3行目のコマンド「OUT (@PTOCR___),A」2バイト=4バイトとなり、総実行サイクルは、2行目2サイクル+3行目3サイクル=5サイクルとなる。

10

【1447】

一方、図172 (b) に示したコマンド群の総コマンドサイズは、2行目のコマンド「OUT (@PTOCR___),@TOCRVAL」3バイト=3バイトとなり、総実行サイクルは、2行目4サイクル=4サイクルとなる。

【1448】

したがって、図172 (a) のコマンド群を図172 (b) のコマンド群に置き換えることで、総コマンドサイズが1バイト削減され、総実行サイクルも少なくとも1サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

20

【1449】

また、図172 (a) と図172 (b) とを比較して理解できるように、図172 (a) において2行 (2行目~3行目) を占有していたコマンド群を、図172 (b) においては1行 (2行目) で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1450】

また、図172 (b) の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図172 (a) の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

30

【1451】

〈LDQ〉

図173は、FZ__SPNモジュールの具体的な処理を示したフローチャートである。FZ__SPNモジュールは、上記普通図柄変動中処理 (図53参照) を実行する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1452】

メインCPU300aは、図173のように、普通図柄表示図柄カウンタに、普通図柄停止図柄番号 (カウンタ値) をセーブする。(S1)。なお、ステップS1は、図53におけるステップS820-9に相当する。

40

【1453】

図174は、FZ__SPNモジュールを実現するためのコマンドの一例を説明するための説明図である。図173で示したフローチャートは、例えば、図174に示したプログラムによって実現される。

【1454】

図174 (a) の1行目の指標「FZ__SPN:」は、当該FZ__SPNモジュールの先頭アドレスを示す。そして、図174 (a) の2行目のコマンド「LDQ A, (LOWR__FZ__STP)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレ

50

ス「R_FZ_STP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（普通図柄停止図柄番号）をAレジスタに読み出す。

【1455】

そして、図174(a)の2行目のコマンド「LDQ (LOW R_FZ_DSP), A」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FZ_DSP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレス（普通図柄表示図柄カウンタのアドレス）にAレジスタの値を格納する。

【1456】

ここで、図174(a)のコマンド群を図174(b)のように変更することができる。ここでは、図174(a)と実質的に等しい処理についてはその説明を省略し、図174(b)の異なる処理のみを説明する。

【1457】

図174(b)の2行目のコマンド「LDQ (LOW R_FZ_DSP), (LOW R_FZ_STP)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FZ_STP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値を、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FZ_DSP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納する。

【1458】

ここで、図174(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R_FZ_STP)」2バイト+3行目コマンド「LDQ (LOW R_FZ_DSP), A」2バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目3サイクル=6サイクルとなる。

【1459】

一方、図174(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ (LOW R_FZ_DSP), (LOW R_FZ_STP)」4バイト=4バイトとなり、総実行サイクルは、2行目6サイクル=6サイクルとなる。

【1460】

したがって、図174(a)と図174(b)とを比較して理解できるように、図174(a)において2行（2行目～3行目）を占有していたコマンド群を、総コマンドサイズおよび総実行サイクルを変更することなく、図174(b)においては1行（2行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1461】

また、図174(b)の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図174(a)の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1462】

図175は、CPUINITモジュールを実現するためのコマンドの一例を説明するための説明図である。ここで、特別遊技は、低確率遊技状態および高確率遊技状態のいずれかの遊技状態にて遊技が進行する。そして、遊技機100には、遊技状態が高確率遊技状態であることを示す表示器が設けられており、遊技状態が高確率遊技状態である場合には、当該表示器が点灯する。そして、電断復帰時において高確率遊技状態であることを遊技者に報知しない場合（所謂潜伏中）には、当該表示器を非表示にすることができる。このとき、実際の遊技状態に基づいて当該表示器の点灯有無を判定してしまうと、電断復帰時の遊技状態が高確率遊技状態であれば、当該表示器を点灯させてしまう。そこで、電断復帰時から初めて大役遊技が開始されるまでの間、潜伏中であることを示すフラグを用意し、そのフラグに基づいて当該表示器の点灯有無を判定するようになされている。

【1463】

10

20

30

40

50

図175 (a) の1行目の指標「CPUINIT:」は、当該CPUINITモジュールの先頭アドレスを示す。そして、図175 (a) の2行目のコマンド「LDQ A, (LOW R_KAK_FLG)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_KAK_FLG」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレス（高確率遊技状態であるか否かを示すフラグのアドレス）に格納された値（高確率遊技状態であれば1、高確率遊技状態でなければ0）をAレジスタに読み出す。

【1464】

そして、図175 (a) の3行目のコマンド「LDQ (LOW R_KAK_HOT), A」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_KAK_HOT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレス（潜伏中であることを示すフラグのアドレス）にAレジスタの値を格納する。

10

【1465】

ここで、図175 (a) のコマンド群を図175 (b) のように変更することができる。ここでは、図175 (a) と実質的に等しい処理についてはその説明を省略し、図175 (b) の異なる処理のみを説明する。

【1466】

図175 (b) の2行目のコマンド「LDQ (LOW R_KAK_HOT), (LOW R_KAK_FLG)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_KAK_FLG」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値を、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_KAK_HOT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納する。

20

【1467】

ここで、図175 (a) に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R_KAK_FLG)」2バイト+3行目コマンド「LDQ (LOW R_KAK_HOT), A」2バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目3サイクル=6サイクルとなる。

【1468】

一方、図175 (b) に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ (LOW R_KAK_HOT), (LOW R_KAK_FLG)」4バイト=4バイトとなり、総実行サイクルは、2行目6サイクル=6サイクルとなる。

30

【1469】

したがって、図175 (a) と図175 (b) とを比較して理解できるように、図175 (a) において2行（2行目～3行目）を占有していたコマンド群を、図175 (b) においては1行（2行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1470】

また、図175 (b) の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図175 (a) の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

40

【1471】

図176 は、EXE_SETモジュールを実現するためのコマンドの一例を説明するための説明図である。図176 に示したEXE_SETモジュールは、図91におけるステップS2400-11に示した実行フラグ設定処理を実行する。

【1472】

図176 (a) の1行目の指標「EXE_SET:」は、当該EXE_SETモジュールの先頭アドレスを示す。そして、図176 (a) の2行目のコマンド「LDQ A, (LOW AT_NXT)」によって、Qレジスタの値をアドレスの上位1バイトとし、ア

50

ドレス「AT__NXT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレス（次遊技の演出状態を示す値のアドレス）に格納された値（次遊技の演出状態を示す値）をAレジスタに読み出す。

【1473】

そして、図176（a）の3行目のコマンド「LDQ（LOW AT__MOD），A」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「AT__MOD」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレス（当該遊技の演出状態を示す値のアドレス）にAレジスタの値を格納する。

【1474】

ここで、図176（a）のコマンド群を図176（b）のように変更することができる。ここでは、図176（a）と実質的に等しい処理についてはその説明を省略し、図176（b）の異なる処理のみを説明する。

【1475】

図176（b）の2行目のコマンド「LDQ（LOW AT__MOD），（LOW AT__NXT）」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「AT__NXT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値を、Qレジスタの値をアドレスの上位1バイトとし、アドレス「AT__MOD」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納する。

【1476】

ここで、図176（a）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A，（LOW AT__NXT）」2バイト+3行目コマンド「LDQ（LOW AT__MOD），A」2バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目3サイクル=6サイクルとなる。

【1477】

一方、図176（b）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ（LOW AT__MOD），（LOW AT__NXT）」4バイト=4バイトとなり、総実行サイクルは、2行目6サイクル=6サイクルとなる。

【1478】

したがって、図176（a）と図176（b）とを比較して理解できるように、図176（a）において2行（2行目～3行目）を占有していたコマンド群を、図176（b）においては1行（2行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1479】

また、図176（b）の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図176（a）の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1480】

〈LDIN〉

図177は、HPT__GRPモジュールの具体的な処理を示したフローチャートである。HPT__GRPモジュールは、上記特別図柄変動番号決定処理（図39参照）におけるステップS612～7のリーチグループ決定処理を実行する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該HPT__GRPモジュールの説明中、第1レジスタはHLレジスタであり、他のレジスタはBCレジスタまたはAレジスタである。

【1481】

リーチグループ決定処理では、保留種別、保留数、遊技状態等に応じて、予め決定されたリーチグループ決定乱数判定テーブルのアドレスが、HLレジスタに事前に格納されている。

10

20

30

40

50

【1482】

メインCPU300aは、図177のように、HLレジスタに示されるアドレス（テーブルアドレス）に格納された比較値をBCレジスタにロードし、HLレジスタの値を2加算する（S1）。なお、詳しくは後述するように、リーチグループ決定乱数判定テーブルは、2バイト長の比較値と、1バイト長のグループ番号とが1セットとして、これら1セットが複数順に連続して格納されている。したがって、ここでは、比較値をロードした後にHLレジスタの値を2加算することで、HLレジスタは、ロードした比較値の次に格納されたグループ番号が格納されたアドレス値となる。

【1483】

その後、メインCPU300aは、BCレジスタにロードした比較値と、DEレジスタに事前にロードされたリーチグループ決定乱数（乱数値）とを比較し（S2）、HLレジスタに示されるアドレスに格納されたリーチグループ番号をロードする（S3）。 10

【1484】

その後、メインCPU300aは、HLレジスタの値（テーブルアドレス）を1加算する（S4）。ここでは、グループ番号をロードした後にHLレジスタの値を1加算することで、HLレジスタは、ロードしたグループ番号の次に格納された比較値が格納されたアドレス値となる。

【1485】

そして、メインCPU300aは、上記ステップS2の比較結果として、比較値がリーチグループ決定乱数（乱数値）以下であれば（S5におけるYES）、上記ステップS1に処理を戻し、比較値がリーチグループ決定乱数以下でなければ（S5におけるNO）、当該HPT__GRPモジュールを終了する（S6）。 20

【1486】

図178（a）、（b）は、HPT__GRPモジュールを実現するためのコマンドの一例を説明するための説明図である。図178（c）は、リーチグループ決定乱数判定テーブルの一例を説明するための説明図である。図177で示したフローチャートは、例えば、図178（a）、（b）に示したプログラムによって実現される。まず、図178（c）のリーチグループ決定乱数判定テーブルを説明した後に、図178（a）、（b）に示すHPT__GRPモジュールを実現するためのコマンドを説明する。

【1487】

図178（c）における指標「D__GRP__SEL__00:」を起点としたテーブルは、いずれかのリーチグループ決定乱数判定テーブルの先頭アドレスを示す。そして、2行目の固定値「8999」、4行目の固定値「9099」、6行目の固定値「9299」は、それぞれ2バイト長の比較値であり、3行目の固定値「@D__MOD__SEL__00」、5行目の固定値「@D__MOD__SEL__01」、7行目の固定値「@D__MOD__SEL__02」は、それぞれ1バイト長のグループ番号である。そして、リーチグループ決定乱数判定テーブルでは、2バイト長の比較値、当該比較値に対応する1バイト長のグループ番号が1セットとなって連続して順に格納されている。 30

【1488】

図178（a）の1行目の指標「HPT__GRP:」は、当該HPT__GRPモジュールの先頭アドレスを示す。そして、図178（a）の2行目のコマンド「LDIN BC, (HL)」によって、HLレジスタで示されるアドレスに格納された2バイト長の値（比較値）をBCレジスタに読み出し（ロードし）、HLレジスタの値を2加算する。かかる2行目のコマンドが、図177のステップS1に対応する。なお、HLレジスタには、事前に、予め決定されたリーチグループ決定乱数判定テーブルのアドレスがロードされている。 40

【1489】

そして、図178（a）の3行目のコマンド「CP BC, DE」によって、BCレジスタに格納された値（比較値）と、DEレジスタに格納された値（リーチグループ決定乱数）とを比較する。ここで、BCレジスタに格納された値が、DEレジスタに格納された 50

値以下であれば、ゼロフラグ（等しい場合）またはキャリーフラグ（未満の場合）に1が立つことになる。なお、DEレジスタには、事前に、リーチグループ決定乱数が格納されている。かかる3行目のコマンドが、図177のステップS2に対応する。

【1490】

その後、図178(a)の4行目のコマンド「LD A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト長の値（グループ番号）をAレジスタに読み出す。かかる4行目のコマンドが、図177のステップS3に対応する。そして、図178(a)の5行目のコマンド「INC HL」によって、HLレジスタの値を1加算する。かかる5行目のコマンドが、図177のステップS4に対応する。

【1491】

続いて、図178(a)の6行目のコマンド「JLS HPT_GRP」によって、3行目のコマンド「CP BC, DE」によってゼロフラグまたはキャリーフラグに1が立っていた場合、「HPT_GRP」が示すアドレスに移動し、ゼロフラグおよびキャリーフラグに1が立っていなかった場合、図178(a)の7行目のコマンド「RET」によって、当該HPT_GRPモジュールが終了する。かかる6行目、7行目のコマンドが、図177のステップS5、S6に対応する。そして、HPT_GRPモジュールが終了した際には、図9に示す範囲に応じたグループ番号がAレジスタに格納されることになる。

【1492】

ここで、図178(a)のコマンド群を図178(b)のように変更することができる。ここでは、図178(a)と実質的に等しい処理についてはその説明を省略し、図178(b)の異なる処理のみを説明する。

【1493】

図178(b)の4行目のコマンド「LDIN A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト長の値（グループ番号）をAレジスタに読み出し、HLレジスタの値を1加算する。かかる4行目のコマンドが、図177のステップS3、S4に対応する。かかるコマンドのコマンドサイズは「1」であり、実行サイクルは「2」である。

【1494】

ここで、図178(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDIN BC, (HL)」2バイト+3行目コマンド「CP BC, DE」2バイト+4行目のコマンド「LD A, (HL)」1バイト+5行目のコマンド「INC HL」1バイト+6行目のコマンド「JLS HPT_GRP」3バイト+7行目のコマンド「RET」1バイト=10バイトとなり、総実行サイクルは、2行目4サイクル+3行目2サイクル+4行目2サイクル+5行目1サイクル+6行目4(3)サイクル+7行目3サイクル=16(15)サイクルとなる。なお、括弧内のサイクル数は、コマンド「JLS HPT_GRP」によって移動しなかった場合の実行サイクルを示している。

【1495】

一方、図178(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDIN BC, (HL)」2バイト+3行目コマンド「CP BC, DE」2バイト+4行目のコマンド「LDIN A, (HL)」1バイト+5行目のコマンド「JLS HPT_GRP」3バイト+6行目のコマンド「RET」1バイト=9バイトとなり、総実行サイクルは、2行目4サイクル+3行目2サイクル+4行目2サイクル+5行目4(3)サイクル+6行目3サイクル=15(14)サイクルとなる。なお、括弧内のサイクル数は、コマンド「JLS HPT_GRP」によって移動しなかった場合の実行サイクルを示している。

【1496】

したがって、図178(a)のコマンド群を図178(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減され、総実行サイクルも少なくとも1サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

20

30

40

50

【1497】

また、図178(a)と図178(b)とを比較して理解できるように、図178(a)において2行(4行目~5行目)を占有していたコマンド群を、図178(b)においては1行(4行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1498】

また、図178(b)の例では、バイト長が異なる複数の値が交互に並置したテーブル中の所望する2バイト長の値または1バイト長の値を取得する際に、HLレジスタの値に2加算および1加算を同一のコマンド「LDIN」で実行することができ、設計負荷の軽減を図ることが可能となる。

【1499】

図179は、E_ILGERモジュールを説明するための説明図である。図179に示したE_ILGERモジュールは、図98におけるステップS3100-19に示した不正監視処理を実行する。

【1500】

図179(a)の1行目の指標「E_ILGER:」は、当該E_ILGERモジュールの先頭アドレスを示す。2行目のコマンド「LD HL, _EX_ILTM」によって、不正投入監視タイマを格納するアドレス「EX_ILTM」をHLレジスタに読み出す。

【1501】

その後、図179(a)の3行目のコマンド「LD A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト長の値(不正投入監視タイマ)をAレジスタに読み出す。そして、図179(a)の4行目のコマンド「INC HL」によって、HLレジスタの値を1加算する。続いて、図179(a)の5行目のコマンド「JT Z, A, E_ILGER01」によって、Aレジスタの値が0である場合(不正投入監視タイマが0である場合)、以降の処理を省略して6行目の指標「E_ILGER01:」に移動し、Aレジスタの値が0でない場合、当該コマンドの次のコマンドに処理を移す。

【1502】

ここで、図179(a)のコマンド群を図179(b)のように変更することができる。ここでは、図179(a)と実質的に等しい処理についてはその説明を省略し、図179(b)の異なる処理のみを説明する。

【1503】

図179(b)の3行目のコマンド「LDIN A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト長の値(不正投入監視タイマ)をAレジスタに読み出し、HLレジスタの値を1加算する。かかるコマンドのコマンドサイズは「1」であり、実行サイクルは「2」である。

【1504】

ここで、図179(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD HL, _EX_ILTM」3バイト+3行目のコマンド「LD A, (HL)」1バイト+4行目のコマンド「INC HL」1バイト+5行目のコマンド「JT Z, A, E_ILGER01」2バイト=7バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目1サイクル+5行目3(2)サイクル=9(8)サイクルとなる。なお、括弧内のサイクル数は、コマンド「JT Z, A, E_ILGER01」によって移動しなかった場合の実行サイクルを示している。

【1505】

一方、図179(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD HL, _EX_ILTM」3バイト+3行目のコマンド「LDIN A, (HL)」1バイト+4行目のコマンド「JT Z, A, E_ILGER01」2バイト=6バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+4行目3(2)サイクル=8(7)サイクルとなる。なお、括弧内のサイクル数は、コマンド「JT Z, A, E_ILGER01」によって移動しなかった場合の実行サイクルを示している。

10

20

30

40

50

【1506】

したがって、図179(a)のコマンド群を図179(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減され、総実行サイクルも少なくとも1サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1507】

また、図179(a)と図179(b)とを比較して理解できるように、図179(a)において2行(3行目~4行目)を占有していたコマンド群を、図179(b)においては1行(3行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

10

【1508】

図180は、E__LEVOTモジュールを説明するための説明図である。図180に示したE__LEVOTモジュールは、図91におけるステップS2400-11に示した実行フラグ設定処理において呼び出されるサブルーチンであり、レバー押下時試験信号送信処理を実行する。

【1509】

図180(a)の1行目の指標「E__LEVOT:」は、当該E__LEVOTモジュールの先頭アドレスを示す。2行目のコマンド「LD HL, T__EXM__STP」によって、試験用停止情報送信テーブルを格納するアドレス「T__EXM__STP」をHLレジスタに読み出す。この試験用停止情報送信テーブルには、1バイト長の6つの試験信号の値が1セットとして連続して格納されているとともに、複数セットが連続して格納されている。

20

【1510】

その後、図180(a)の3行目のコマンド「ADDWB HL, A」によって、HLレジスタの値にAレジスタの値が加算され、HLレジスタの値が更新される。なお、Aレジスタには、指示情報種別に6を乗算した値が格納されている。したがって、ここでは、HLレジスタの値が、試験用停止情報送信テーブルにおいて、指示情報種別に対応する1セットの先頭アドレスを示すこととなる。

【1511】

そして、4行目のコマンド「LD B, 6」によって、固定値「6」がBレジスタに読み出される。図180(a)の5行目の指標「E__LEVOT01:」は、指標「E__LEVOT01」のアドレスを示す。

30

【1512】

図180(a)の6行目のコマンド「LD A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト長の値(試験信号の値)をAレジスタに読み出す。そして、図180(a)の7行目のコマンド「OUT (@S2DT__), A」によって、Uレジスタの値をアドレスの上位1バイトとし、RAMアクセスプロテクトレジスタポート(内部レジスタI/Oポート)のアドレスの下位1バイトを示す固定値「@S2DT__」を下位1バイトとし、そのアドレスに、Aレジスタの値を出力する。

【1513】

図180(a)の8行目のコマンド「INC HL」によって、HLレジスタの値を1加算する。続いて、図180(a)の9行目のコマンド「DJNZ E__LEVOT01」によって、Bレジスタの値がデクリメント(「1」減算)され、デクリメントした結果が0でなければ、アドレス「E__LEVOT01」に移動し、デクリメントした結果が0であれば、当該コマンドの次のコマンドに処理を移す。

40

【1514】

ここで、図180(a)のコマンド群を図180(b)のように変更することができる。ここでは、図180(a)と実質的に等しい処理についてはその説明を省略し、図180(b)の異なる処理のみを説明する。

【1515】

50

図180 (b) の6行目のコマンド「LDIN A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト長の値（試験信号の値）をAレジスタに読み出し、HLレジスタの値を1加算する。かかるコマンドのコマンドサイズは「1」であり、実行サイクルは「2」である。

【1516】

ここで、図180 (a) に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD HL, T_EXM_STP」3バイト+3行目のコマンド「ADDWB HL, A」1バイト+4行目のコマンド「LD B, 6」2バイト+6行目のコマンド「LD A, (HL)」1バイト+7行目のコマンド「OUT (@S2DT___), A」2バイト+8行目のコマンド「INC HL」1バイト+9行目のコマンド「DJNZ E_LEVOT01」2バイト=12バイトとなり、総実行サイクルは、2行目3サイクル+3行目1サイクル+4行目2サイクル+6行目2サイクル+7行目3サイクル+8行目1サイクル+9行目3(2)サイクル=15(14)サイクルとなる。なお、括弧内のサイクル数は、コマンド「DJNZ E_LEVOT01」によって移動しなかった場合の実行サイクルを示している。

10

【1517】

一方、図180 (b) に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD HL, T_EXM_STP」3バイト+3行目のコマンド「ADDWB HL, A」1バイト+4行目のコマンド「LD B, 6」2バイト+6行目のコマンド「LDIN A, (HL)」1バイト+7行目のコマンド「OUT (@S2DT___), A」2バイト+5行目のコマンド「DJNZ E_LEVOT01」2バイト=11バイトとなり、総実行サイクルは、2行目3サイクル+3行目1サイクル+4行目2サイクル+6行目2サイクル+7行目3サイクル+9行目3(2)サイクル=14(13)サイクルとなる。なお、括弧内のサイクル数は、コマンド「DJNZ E_LEVOT01」によって移動しなかった場合の実行サイクルを示している。

20

【1518】

したがって、図180 (a) のコマンド群を図180 (b) のコマンド群に置き換えることで、総コマンドサイズが1バイト削減され、総実行サイクルも少なくとも1サイクル削減される。特に、図180 (a) の例では、コマンド「DJNZ E_LEVOT01」におけるBレジスタの値が2以上であれば、その値に1サイクルを乗じた分だけ総実行サイクルの差分が増えるので、図180 (b) の総実行サイクルの削減量も多くなる。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

30

【1519】

また、図180 (a) と図180 (b) とを比較して理解できるように、図180 (a) において2行(6行目、8行目)を占有していたコマンド群を、図180 (b) においては1行(6行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1520】

〈LDQP〉

40

図181は、SBC_OUTモジュールの具体的な処理を示したフローチャートである。SBC_OUTモジュールは、上記ステップS100-65のサブコマンド送信処理(図23参照)を実行する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1521】

ここで、サブコマンドは、割込み処理における複数の処理において送信バッファにセットされ、サブコマンド送信処理で送信される。ここで、送信バッファは、例えば96バイトのリングバッファであり、複数の処理においてサブコマンドが順にセットされ、サブコマンド送信処理において、先にセットされたサブコマンドから順に副制御基板330に送信されるFIFOである。

50

【1522】

メインCPU300aは、図181のように、サブコマンドライトポインタの値をAレジスタに読み出し（S1）、サブコマンドライトポインタの値と、サブコマンドリードポインタの値とを比較する（S2）。ここで、サブコマンドライトポインタの値と、サブコマンドリードポインタの値とが同じ場合、すなわち、未送信のサブコマンドがない場合にはゼロフラグに1が立ち、異なる場合、すなわち、未送信のサブコマンドがある場合にはゼロフラグに1が立たない。

【1523】

メインCPU300aは、ゼロフラグが1であれば（S3におけるYES）、当該SBC__OUTモジュールを終了し、ゼロフラグが1でなければ（S3におけるNO）、サブコマンドポインタ上限値、および、サブコマンドリードポインタのアドレスの下位バイトをDレジスタにそれぞれ読み出し（S4）、汎用モジュールであるカウントアップ処理によって、サブコマンドリードポインタの値を、サブコマンドポインタ上限値未満なら1加算し、サブコマンドポインタ上限値以上ならゼロクリアする（S5）。

10

【1524】

その後、メインCPU300aは、サブコマンドバッファ（送信バッファ）のアドレスをHレジスタに読み出し（S6）、上記のワードデータ選択処理を実行した後（S7）、サブコマンドを送信するためのSCU1データレジスタに先行コマンドを出力し（S8）、後続コマンドをAレジスタに読み出し（S9）、SCU1データレジスタに後続コマンドを出力し（S10）、当該SBC__OUTモジュールを終了する（S11）。

20

【1525】

図182は、SBC__OUTモジュールを実現するためのコマンドの一例を説明するための説明図である。

【1526】

図182（a）の1行目の指標「SBC__OUT:」は、当該SBC__OUTモジュールの先頭アドレスを示す。そして、図182（a）の2行目のコマンド「LDQ A, (LOW R__SBC__WPT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__SBC__WPT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（サブコマンドライトポインタの値）をAレジスタに読み出す。かかる2行目のコマンドが、図181のステップS1に対応する。

30

【1527】

図182（a）の3行目のコマンド「CPQ A, (LOW R__SBC__RPT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__SBC__RPT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（サブコマンドリードポインタの値）と、Aレジスタの値（サブコマンドライトポインタの値）とを比較する。ここで、同アドレスに格納された値と、Aレジスタの値とが一致していれば、ゼロフラグに1が立つ。かかる3行目のコマンドが、図181のステップS2に対応する。

【1528】

図182（a）の4行目のコマンド「RET Z」によって、ゼロフラグに1が立っていれば、当該SBC__OUTモジュールを終了する。かかる4行目のコマンドが、図181のステップS3に対応する。

40

【1529】

図182（a）の5行目のコマンド「LD DE, @LMT__CMD__BFP*256+LOW R__SBC__RPT」によって、サブコマンドポインタ上限値を示す固定値「@LMT__CMD__BFP」がDレジスタに読み出され、サブコマンドリードポインタのアドレスの下位バイトを示す「R__SBC__RPT」がEレジスタに読み出される。かかる5行目のコマンドが、図181のステップS4に対応する。

【1530】

図182（a）の6行目のコマンド「RST COUNTUP」によって、汎用モジュ

50

ールであるCOUNTUPモジュールが読み出されて実行される。これにより、サブコマンドリードポインタの値は、サブコマンドポインタ上限値未満なら1加算され、サブコマンドポインタ上限値以上ならゼロクリアされる。かかる6行目のコマンドが、図181のステップS5に対応する。

【1531】

図182(a)の7行目のコマンド「LD HL,R__SBC__SBF」によって、サブコマンドバッファ（送信バッファ）のアドレスを示す「R__SBC__SBF」がHLレジスタに読み出される。かかる7行目のコマンドが、図181のステップS6に対応する。ここで、「R__SBC__SBF」は、2バイト長のデータである。サブコマンドバッファのアドレスの上位1バイトが「F0H」であれば、コマンド「LDQ」によって読み出すことが可能であるが、サブコマンドバッファの上位1バイトが「F1H」になっていることもあるため、ここでは、コマンド「LDQ」を使用せずに2バイト長のデータを読み出している。

10

【1532】

図182(a)の8行目のコマンド「RST WORDSEL」によって、汎用モジュールであるWORDSELモジュールが読み出され、HLレジスタにAレジスタの値を2回加算し、HLレジスタに示されるアドレスのうち、下位バイトの値（Lレジスタの値、すなわち、先行コマンド）がAレジスタに読み出される。かかる8行目のコマンドが、図181のステップS7に対応する。

【1533】

図182(a)の9行目のコマンド「OUT (@S1DT___),A」によって、Aレジスタの値（先行コマンド）が、固定値「@S1DT___」で示されるアドレスに出力される。かかる9行目のコマンドが、図181のステップS8に対応する。

20

【1534】

図182(a)の10行目のコマンド「LD A,H」によって、Hレジスタの値（後続コマンド）がAレジスタに読み出される。かかる10行目のコマンドが、図181のS9に対応する。

【1535】

図182(a)の11行目のコマンド「OUT (@S1DT___),A」によって、Aレジスタの値（後続コマンド）が、固定値「@S1DT___」で示されるアドレスに出力され、当該SBC__OUTモジュールが終了する。かかる11行目のコマンドが、図181のステップS10に対応し、12行目のコマンドが、図181のステップS11に対応する。

30

【1536】

ここで、図182(a)のコマンド群を図182(b)のように変更することができる。ここでは、サブコマンドバッファ（送信バッファ）の上位1バイトが「F1H」である場合について説明し、図182(a)と実質的に等しい処理についてはその説明を省略し、図182(b)の異なる処理のみを説明する。

【1537】

図182(b)の7行目のコマンド「LDQP HL,LOW R__SBC__SBF」によって、Qレジスタの値に1加算された値（すなわち、「F1H」）が上位に、「R__SBC__SBF」が下位となる値（サブコマンドバッファのアドレス）がHLレジスタに読み出される。かかる7行目のコマンドが、図181のステップS6に対応する。かかるコマンドのコマンドサイズは「3」であり、実行サイクルは「7(5)」である。

40

【1538】

ここで、図182(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A,(LOW R__SBC__WPT)」2バイト+3行目コマンド「CPQ A,(LOW R__SBC__RPT)」3バイト+4行目のコマンド「RET Z」1バイト+5行目のコマンド「LD DE,@LMT__CMD__BFP*256+LOW R__SBC__RPT」2バイト+6行目のコマンド「RST COUNTUP」1バイト+7行目

50

のコマンド「LD HL,R__SBC__SBF」3バイト+8行目のコマンド「RST WORDSEL」1バイト+9行目のコマンド「OUT (@S1DT__),A」2バイト+10行目のコマンド「LD A,H」1バイト+11行目のコマンド「OUT (@S1DT__),A」2バイト+12行目のコマンド「RET」1バイト=19バイトとなり、総実行サイクルは、2行目2サイクル+3行目4サイクル+4行目3(1)サイクル+5行目2サイクル+6行目4サイクル+7行目3サイクル+8行目4サイクル+9行目3サイクル+10行目1サイクル+11行目3サイクル+12行目3サイクル=32(30)サイクルとなる。なお、括弧内のサイクル数は、コマンド「RET Z」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1539】

一方、図182(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A,(LOW R__SBC__WPT)」2バイト+3行目コマンド「CPQ A,(LOW R__SBC__RPT)」3バイト+4行目のコマンド「RET Z」1バイト+5行目のコマンド「LD DE,@LMT__CMD__BFP*256+LOW R__SBC__RPT」2バイト+6行目のコマンド「RST COUNTUP」1バイト+7行目のコマンド「LDQP HL,LOW R__SBC__SBF」3バイト+8行目のコマンド「RST WORDSEL」1バイト+9行目のコマンド「OUT (@S1DT__),A」2バイト+10行目のコマンド「LD A,H」1バイト+11行目のコマンド「OUT (@S1DT__),A」2バイト+12行目のコマンド「RET」1バイト=19バイトとなり、総実行サイクルは、2行目2サイクル+3行目4サイクル+4行目3(1)サイクル+5行目2サイクル+6行目4サイクル+7行目5サイクル+8行目4サイクル+9行目3サイクル+10行目1サイクル+11行目3サイクル+12行目3サイクル=34(32)サイクルとなる。なお、括弧内のサイクル数は、コマンド「RET Z」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1540】

したがって、図182(a)のコマンド群を図182(b)のコマンド群に置き換えることで、Qレジスタに「F0」が設定されている状態で、F100H以降のアドレスにアクセスする場合(上位の1バイトが「F1H」である場合)に、設計負荷の軽減を図ることが可能となる。なお、図37のステップS610-9で示した特別図柄記憶エリアシフト処理において、特別遊技特別図柄判別フラグを読み出す場合、特別遊技特別図柄判別フラグが格納されたアドレスの上位1バイトを「F1H」とし、コマンド「LDQP」を用いるようにしてもよい。さらに、図26のステップS400-29で示した外部情報管理処理において、出力ポート2バッファに合成ビットデータをセーブする場合に、出力ポート2バッファの上位1バイトを「F1H」とし、コマンド「LDQP」を用いるようにしてもよい。さらに、図25の電源断時退避処理において、バックアップ有効判定フラグにバックアップ有効判定値をセーブする場合に、バックアップ有効判定フラグが格納されたアドレスの上位1バイトを「F1H」とし、コマンド「LDQP」を用いるようにしてもよい。

【1541】

〈RCPQ〉

図183は、SBC__OUTモジュールを実現するための他のコマンドの一例を説明するための説明図である。なお、図183(a)は、図182(a)と同一であるため、説明は省略する。また、図183(b)について、図183(a)と実質的に等しい処理についてはその説明を省略し、図183(b)の異なる処理のみを説明する。

【1542】

図183(b)の3行目のコマンド「RCPQ Z,A,(LOW R__SBC__RPT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__SBC__RPT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(サブコマンドリードポインタの値)と、Aレジスタの値(サブコマンドライトポインタの値)とを比較し、ゼロフラグに1が立っていれば、コマンド「RET」を実行するよう

10

20

30

40

50

に、当該SBC_OUTモジュールを終了する。かかる3行目のコマンドが、図181のステップS2～S4に対応する。かかるコマンドのコマンドサイズは「3」であり、実行サイクルは「7（5）」である。なお、括弧内のサイクル数は、かかるコマンドによって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1543】

ここで、図183（b）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__SBC__WPT)」2バイト+3行目コマンド「RCPQZ, A, (LOW R__SBC__RPT)」3バイト+4行目のコマンド「LD DE, @LMT__CMD__BFP*256+LOW R__SBC__RPT」2バイト+5行目のコマンド「RST COUNTUP」1バイト+6行目のコマンド「LD HL, R__SBC__S

BF」3バイト+7行目のコマンド「RST WORDSEL」1バイト+8行目のコマンド「OUT (@S1DT__), A」2バイト+9行目のコマンド「LD A, H」1バイト+10行目のコマンド「OUT (@S1DT__), A」2バイト+11行目のコマンド「RET」1バイト=18バイトとなり、総実行サイクルは、2行目2サイクル+3行目7（5）サイクル+4行目2サイクル+5行目4サイクル+6行目3サイクル+7行目4サイクル+8行目3サイクル+9行目1サイクル+10行目3サイクル+11行目3サイクル=32（30）サイクルとなる。なお、括弧内のサイクル数は、コマンド「RCPQZ, A, (LOW R__SBC__RPT)」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1544】

したがって、図183（a）のコマンド群を図183（b）のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1545】

また、図183（a）と図183（b）とを比較して理解できるように、図183（a）において2行（3行目～4行目）を占有していたコマンド群を、図183（b）においては1行（3行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1546】

図184は、FZ__OPNモジュールを実現するためのコマンドの一例を説明するための説明図である。図184に示したFZ__OPNモジュールは、普通電動役物入賞口開放制御処理（図57参照）を実行する。

【1547】

普通電動役物入賞口開放制御処理では、上記ステップS850-5において、普通電動役物入賞球数カウンタのカウンタ値が規定数に到達していないか、すなわち、第1可変始動口120Bに、1回の開閉制御中の最大入賞可能数と同数の遊技球が入球していないかを判定する。その結果、規定数に到達していないと判定した場合には当該普通電動役物入賞口開放制御処理を終了し、規定数に到達したと判定した場合にはステップS850-7に処理を移す。

【1548】

この普通電動役物入賞口開放制御処理として、図184（a）の1行目の指標「FZ__OPN:」は、当該FZ__OPNモジュールの先頭アドレスを示す。そして、ステップS850-5として、2行目のコマンド「LDQ A, (LOW R__FDN__CNT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__FDN__CNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（普通電動役物入賞球数カウンタのカウンタ値）をAレジスタに読み出す。3行目のコマンド「CP A, @FDN__CNT」によって、Aレジスタの値と、規定数を示す固定値「@FDN__CNT」、ここでは、8とを比較する。すなわち、Aレジスタの値が8未満であれば、キャリーフラグが立って1となる。そして、4行目の「RET C」によって、キャリーフラグが1であれば、当該FZ__OPNモジュールを終了する。すなわち、普通電

動役物入賞球数カウンタのカウンタ値が8未満であれば、当該FZ__OPNモジュールを終了する。なお、キャリーフラグが1でなければ、以降の処理を実行する。

【1549】

ここで、図184(a)のコマンド群を図184(b)のように変更することができる。ここでは、図184(a)と実質的に等しい処理についてはその説明を省略し、図184(b)の異なる処理のみを説明する。

【1550】

図184(b)の2行目のコマンド「RCPQ C, (LOW R__FDN__CNT), @FDN__CNT」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__FDN__CNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（普通電動役物入賞球数カウンタのカウンタ値）と、規定数を示す固定値「@FDN__CNT」、ここでは、8とを比較し、キャリーフラグが1であれば、コマンド「RET」を実行するように、当該FZ__OPNモジュールを終了する。かかるコマンドのコマンドサイズは「4」であり、実行サイクルは「8(6)」である。なお、括弧内のサイクル数は、かかるコマンドによって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1551】

ここで、図184(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__FDN__CNT)」2バイト+3行目コマンド「CP A, @FDN__CNT」3バイト+4行目のコマンド「RET C」1バイト=6バイトとなり、総実行サイクルは、2行目2サイクル+3行目4サイクル+4行目3(1)サイクル=9(7)サイクルとなる。なお、括弧内のサイクル数は、コマンド「RET C」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1552】

一方、図184(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「RCPQ C, (LOW R__FDN__CNT), @FDN__CNT」4バイト=4バイトとなり、総実行サイクルは、2行目8(6)サイクル=8(6)サイクルとなる。なお、括弧内のサイクル数は、コマンド「RCPQ C, (LOW R__FDN__CNT), @FDN__CNT」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

【1553】

したがって、図184(a)のコマンド群を図184(b)のコマンド群に置き換えることで、総コマンドサイズが2バイト削減され、総実行サイクルも少なくとも1サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1554】

また、図184(a)と図184(b)とを比較して理解できるように、図184(a)において3行(2行目~4行目)を占有していたコマンド群を、図184(b)においては1行(2行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1555】

また、図184(b)の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図184(a)の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1556】

図185は、TD__OPNモジュールを実現するためのコマンドの一例を説明するための説明図である。図185に示したTD__OPNモジュールは、大入賞口開放制御処理(図47参照)を実行する。

【1557】

大入賞口開放制御処理では、上記ステップS720-5において、大入賞口入賞球数カウンタのカウンタ値が規定数に到達していないか、すなわち、大入賞口に、1ラウンド中の最大入賞可能数と同数の遊技球が入球していないかを判定する。その結果、規定数に到達していないと判定した場合には当該大入賞口開放制御処理を終了し、規定数に到達したと判定した場合にはステップS720-7に処理を移す。

【1558】

この大入賞口開放制御処理として、図185(a)の1行目の指標「TD__OPN:」は、当該TD__OPNモジュールの先頭アドレスを示す。そして、ステップS720-5として、2行目のコマンド「LDQ A, (LOW R__TDN__FLG)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__TDN__FLG」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（特別電動役物指定フラグ）をAレジスタに読み出す。なお、特別電動役物指定フラグは、どの大入賞口が当該大入賞口開放制御処理の対象となっているかを示すものである。

10

【1559】

3行目のコマンド「LD HL, D__TDC__MAX-1」によって、大入賞口規定入賞数データテーブル（図14における特別電動役物作動ラムセットテーブルの規定数が示されるテーブル）のアドレスから1減算した値をHLレジスタに読み出す。

【1560】

図185(a)の4行目のコマンド「RST BYTESEL」によって、汎用モジュールであるBYTESELモジュールが読み出され、HLレジスタにAレジスタの値を加算し、HLレジスタに示されるアドレスに格納された値（規定数）がAレジスタに読み出される。

20

【1561】

図185(a)の5行目のコマンド「CPQ A, (LOW R__TDN__CNT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__TDN__CNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（大入賞口入賞球数カウンタのカウンタ値）と、Aレジスタの値、ここでは、10とを比較する。すなわち、同アドレスに格納された値が10未満であれば、キャリーフラグが立たずに0となる。そして、6行目の「RET NC」によって、キャリーフラグが0であれば、当該TD__OPNモジュールを終了する。すなわち、大入賞口入賞球数カウンタのカウンタ値が10未満であれば、当該TD__OPNモジュールを終了する。なお、キャリーフラグが1であれば、以降の処理を実行する。

30

【1562】

ここで、図185(a)のコマンド群を図185(b)のように変更することができる。ここでは、図185(a)と実質的に等しい処理についてはその説明を省略し、図185(b)の異なる処理のみを説明する。

【1563】

図185(b)の5行目のコマンド「RCPQ NC, A, (LOW R__TDN__CNT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__TDN__CNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（大入賞口入賞球数カウンタのカウンタ値）と、Aレジスタの値（既定値）、ここでは、10とを比較し、キャリーフラグが0であれば、コマンド「RET」を実行するように、当該TD__OPNモジュールを終了する。かかるコマンドのコマンドサイズは「3」であり、実行サイクルは「7(5)」である。なお、括弧内のサイクル数は、かかるコマンドによって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

40

【1564】

ここで、図185(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__TDN__FLG)」2バイト+3行目コマンド「LD HL, D__TDC__MAX-1」3バイト+4行目のコマンド「RST BYTESEL」1バイト+5行目コマンド「CPQ A, (LOW R__TDN__CNT)」3バイト+4行

50

目のコマンド「RET NC」1バイト＝10バイトとなり、総実行サイクルは、2行目2サイクル＋3行目3サイクル＋4行目3サイクル＋5行目4サイクル＋4行目3（1）サイクル＝15（13）サイクルとなる。なお、括弧内のサイクル数は、コマンド「RET NC」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。
【1565】

一方、図185（b）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A,（LOW R_TDN_FLG）」2バイト＋3行目コマンド「LD HL, D_TDC_MAX－1」3バイト＋4行目のコマンド「RST BYTESEL」1バイト＋5行目コマンド「RCPQ NC,A,（LOW R_TDN_CNT）」3バイト＝9バイトとなり、総実行サイクルは、2行目2サイクル＋3行目3サイクル＋4行目3サイクル＋5行目7（5）サイクル＝15（13）サイクルとなる。なお、括弧内のサイクル数は、コマンド「RCPQ NC,A,（LOW R_TDN_CNT）」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

10

【1566】

したがって、図185（a）のコマンド群を図185（b）のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1567】

また、図185（a）と図185（b）とを比較して理解できるように、図185（a）において2行（5行目～6行目）を占有していたコマンド群を、図185（b）においては1行（5行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

20

【1568】

〈JCPQ〉

図186は、HSY_PRCモジュールを実現するためのコマンドの一例を説明するための説明図である。図186に示したHSY_PRCモジュールは、発射位置指定管理処理（図26のステップS400-27参照）を実行する。

【1569】

この発射位置指定管理処理として、図186（a）の1行目の指標「HSY_PRC:」は、当該HSY_PRCモジュールの先頭アドレスを示す。2行目のコマンド「XOR A,A」によって、Aレジスタの値と、Aレジスタの値との排他的論理和を計算し、Aレジスタを初期化する。

30

【1570】

そして、3行目のコマンド「LDQ HL, LOW R_TDN_FLG」によって、Qレジスタの値をHレジスタに読み出し、アドレス「R_TDN_FLG」の下位1バイトの値をLレジスタに読み出す。

【1571】

4行目のコマンド「JCP NZ,（HL）,@TDK_AT2,HSY_PRC_10」によって、HLレジスタに示される値（特別電動役物指定フラグ）と、第2大入賞口を示す固定値「@TDK_AT2」とを比較し、ゼロフラグに1が立っていなければ、7行目の指標「HSY_PRC_10」に示されるアドレスに移動する（処理を分岐させる）。

40

【1572】

一方、ゼロフラグに1が立っていれば、5行目のコマンド「LDQ HL, LOW R_ZUG_CHK_FIX」によって、Qレジスタの値をHレジスタに読み出し、アドレス「R_ZUG_CHK_FIX」の下位1バイトの値をLレジスタに読み出す。

【1573】

6行目のコマンド「JCP NZ,（HL）,@ZUG_SML1,HSY_PRC_20」によって、HLレジスタに示される値（特別図柄判定フラグ）と、小当たり図柄1指定データを示す固定値「@ZUG_SML1」、ここでは、07Hとを比較し、ゼロフラグに1が立っていなければ、8行目の指標「HSY_PRC_20」に示されるアドレスに

50

移動する（処理を分岐させる）。

【1574】

ここで、図186（a）のコマンド群を図186（b）のように変更することができる。ここでは、図186（a）と実質的に等しい処理についてはその説明を省略し、図186（b）の異なる処理のみを説明する。

【1575】

図186（b）の3行目のコマンド「JCPQ NZ, (LOW R__TDN__FLG), @TDK__AT2, HSY__PRC__10」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__TDN__FLG」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに示される値（特別電動役物指定フラグ）と、第2大入賞口を示す固定値「@TDK__AT2」とを比較し、ゼロフラグに1が立っていなければ、5行目の指標「HSY__PRC__10」に示されるアドレスに移動する（処理を分岐させる）。

【1576】

図186（b）の4行目のコマンド「JCPQ NZ, (LOW R__ZUF__CHK__FIX, @ZUG__SML1, HSY__PRC__20」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R__ZUF__CHK__FIX」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに示される値（特別図柄判定フラグ）と、第2大入賞口を示す固定値「@TDK__AT2」とを比較し、小当たり図柄1指定データを示す固定値「@ZUG__SML1」、ここでは、07Hとを比較し、ゼロフラグに1が立っていなければ、6行目の指標「HSY__PRC__20」に示されるアドレスに移動する（処理を分岐させる）。

【1577】

ここで、図186（a）に示したコマンド群の総コマンドサイズは、2行目のコマンド「XOR A,A」1バイト+3行目のコマンド「LDQ HL, LOW R__TDN__FLG」2バイト+4行目のコマンド「JCP NZ, (HL), @TDK__AT2, HSY__PRC__10」4バイト+5行目のコマンド「LDQ HL, LOW R__ZUG__CHK__FIX」2バイト+6行目のコマンド「JCP NZ, (HL), @ZUG__SML1, HSY__PRC__20」4バイト=13バイトとなり、総実行サイクルは、2行目1サイクル+3行目2サイクル+4行目6（5）サイクル+5行目2サイクル+6行目2サイクル6（5）=17（15）サイクルとなる。なお、括弧内のサイクル数は、コマンド「JCP NZ, ...」によって移動しなかった場合の実行サイクルを示している。

【1578】

一方、図186（b）に示したコマンド群の総コマンドサイズは、2行目のコマンド「XOR A,A」1バイト+3行目のコマンド「JCPQ NZ, (LOW R__TDN__FLG), @TDK__AT2, HSY__PRC__10」4バイト+4行目のコマンド「JCPQ NZ, (LOW R__ZUG__CHK__FIX, @ZUG__SML1, HSY__PRC__20」4バイト=9バイトとなり、総実行サイクルは、2行目1サイクル+3行目6（5）サイクル+4行目6（5）=13（11）サイクルとなる。なお、括弧内のサイクル数は、コマンド「JCP NZ, ...」によって移動しなかった場合の実行サイクルを示している。

【1579】

したがって、図186（a）のコマンド群を図186（b）のコマンド群に置き換えることで、総コマンドサイズが4バイト削減され、総実行サイクルも少なくとも4サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1580】

また、図186（a）と図186（b）とを比較して理解できるように、図186（a）において4行（3行目～6行目）を占有していたコマンド群を、図186（b）においては2行（3行目～4行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1581】

また、図186(b)の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図186(a)の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1582】

図187は、FDN_CHKモジュールを実現するためのコマンドの一例を説明するための説明図である。図187に示したFDN_CHKモジュールは、普通電動役物入賞時確認処理(図28のステップS500-9参照)を実行する。

10

【1583】

この普通電動役物入賞時確認処理として、図187(a)の1行目の指標「FDN_CHK:」は、当該FDN_CHKモジュールの先頭アドレスを示す。

【1584】

そして、2行目のコマンド「LDQ HL, (LOW R_FDN_EFF)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FDN_EFF」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスの値(普通電動役物閉鎖有効タイマの値)をHLレジスタに読み出す。

【1585】

3行目のコマンド「RET NTZ」によって、ゼロフラグが1でなければ、当該FDN_CHKモジュールを終了する。ここで「NTZ」としているのは、コマンド「LDQ HL, (LOW R_FDN_EFF)」によって第2ゼロフラグは変化するが第1ゼロフラグは変化しないからである。

20

【1586】

4行目のコマンド「LDQ A, (LOW R_FUT_PHS)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FUT_PHS」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスの値(普通遊技管理フェーズ)をAレジスタに読み出す。

【1587】

5行目のコマンド「JCP NZ, A, @FD_OPN, FDN_CHK_10」によって、Aレジスタに示される値(普通遊技管理フェーズ)と、普通電動役物入賞口開放制御処理を示す固定値「@FD_OPN」、ここでは04Hとを比較し、ゼロフラグに1が立っていなければ、8行目の指標「FDN_CHK_10」に示されるアドレスに移動する(処理を分岐させる)。

30

【1588】

一方、ゼロフラグに1が立っていれば、6行目のコマンド「INCQ (LOW R_FDN_CNT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FDN_CNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(普通電動役物入賞球数カウンタのカウンタ値)に1加算し、同アドレスに、加算した値を格納し、7行目のコマンド「RET」によって、当該FDN_CHKモジュールを終了する。

40

【1589】

ここで、図187(a)のコマンド群を図187(b)のように変更することができる。ここでは、図187(a)と実質的に等しい処理についてはその説明を省略し、図187(b)の異なる処理のみを説明する。

【1590】

図187(b)の4行目のコマンド「JCPQ NZ, (LOW R_FUT_PHS), @FD_OPN, FDN_CHK_10」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FUT_PHS」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスの値(普通遊技管理フェーズ)と、普通電動役物入賞口開放制御処

50

理を示す固定値「@FD__OPN」、ここでは04Hとを比較し、ゼロフラグに1が立っていないければ、7行目の指標「FDN__CHK__10」に示されるアドレスに移動する（処理を分岐させる）。

【1591】

ここで、図187（a）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ HL, (LOW R__FDN__EFF)」2バイト+3行目のコマンド「RET NTZ」1バイト+4行目のコマンド「LDQ A, (LOW R__FUT__PHS)」2バイト+5行目のコマンド「JCP NZ, A, @FD__OPN, FDN__CHK__10」3バイト+6行目のコマンド「INCQ (LOW R__FDN__CNT)」2バイト+7行目のコマンド「RET」1バイト=11バイトとなり、総実行サイクルは、2行目4サイ

10

【1592】

一方、図187（b）に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ HL, (LOW R__FDN__EFF)」2バイト+3行目のコマンド「RET NTZ」1バイト+4行目のコマンド「JCPQ NZ, (LOW R__FUT__PHS), @FD__OPN, FDN__CHK__10」4バイト+5行目のコマンド「INCQ (LOW R__FDN__CNT)」2バイト+6行目のコマンド「RET」1バイト=10バ

20

【1593】

したがって、図187（a）のコマンド群を図187（b）のコマンド群に置き換えることで、総コマンドサイズが1バイト削減され、総実行サイクルも少なくとも2サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1594】

また、図187（a）と図187（b）とを比較して理解できるように、図187（a）において2行（4行目～5行目）を占有していたコマンド群を、図187（b）においては1行（4行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

30

【1595】

また、図187（b）の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図187（a）の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

40

【1596】

図188は、FZ__STPモジュールを実現するためのコマンドの一例を説明するための説明図である。図188に示したFZ__STPモジュールは、普通図柄停止図柄表示処理（図54参照）を実行する。

【1597】

この普通図柄停止図柄表示処理として、図188（a）の1行目の指標「FZ__STP:」は、当該FZ__STPモジュールの先頭アドレスを示す。

【1598】

そして、2行目のコマンド「RET NTZ」によって、第2ゼロフラグに1が立っていないければ、当該FZ__STPモジュールを終了する。

50

【1599】

3行目のコマンド「LDQ A, (LOW R_FZ_ATA)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FZ_ATA」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスの値（普通図柄当たりフラグ）をAレジスタに読み出す。なお、普通図柄当たりフラグは、普図抽選で当たりが決定されると01Hになり、ハズレが決定されると00Hとなる。

【1600】

4行目のコマンド「JCP Z,A,@FZ_ATA,FZ_STP_10」によって、Aレジスタに示される値（普通図柄当たりフラグ）と、当たりを示す固定値「@FZ_ATA」、ここでは01Hとを比較し、ゼロフラグに1が立っていれば、8行目の指標「FZ_STP_10」に示されるアドレスに移動する（処理を分岐させる）。

10

【1601】

一方、ゼロフラグに1が立っていなければ、5行目のコマンド「CLRQ (LOW R_FUT_PHS)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FUT_PHS」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（普通遊技管理フェーズ）に0を代入する（クリアする）。

【1602】

6行目のコマンド「CLRQ (LOW R_FZ_STP)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FZ_STP」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（普通図柄停止図柄番号）に0を代入し（クリアし）、7行目のコマンド「RET」によって、当該FZ_STPモジュールを終了する。

20

【1603】

ここで、図188（a）のコマンド群を図188（b）のように変更することができる。ここでは、図188（a）と実質的に等しい処理についてはその説明を省略し、図188（b）の異なる処理のみを説明する。

【1604】

図188（b）の3行目のコマンド「JCPQ Z, (LOW R_FZ_ATA),@FZ_ATA,FZ_STP_10」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_FZ_ATA」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスの値（普通図柄当たりフラグ）と、当たりを示す固定値「@FZ_ATA」、ここでは01Hとを比較し、ゼロフラグに1が立っていれば、7行目の指標「FZ_STP_10」に示されるアドレスに移動する（処理を分岐させる）。

30

【1605】

ここで、図188（a）に示したコマンド群の総コマンドサイズは、2行目のコマンド「RET NTZ」1バイト+3行目のコマンド「LDQ A, (LOW R_FZ_ATA)」2バイト+4行目のコマンド「JCP Z,A,@FZ_ATA,FZ_STP_10」3バイト+5行目のコマンド「CLRQ (LOW R_FUT_PHS)」2バイト+6行目のコマンド「CLRQ (LOW R_FZ_STP)」2バイト+7行目のコマンド「RET」1バイト=11バイトとなり、総実行サイクルは、2行目3（1）サイクル+3行目2サイクル+4行目4（3）サイクル+5行目2サイクル+6行目2サイクル+7行目3サイクル=16（13）サイクルとなる。なお、括弧内のサイクル数は、コマンド「RET NTZ」「JCP Z,・・・」によって1つ上のルーチンに移動しなかった場合の実行サイクルを示している。

40

【1606】

一方、図188（b）に示したコマンド群の総コマンドサイズは、2行目のコマンド「RET NTZ」1バイト+3行目のコマンド「JCPQ Z, (LOW R_FZ_ATA),@FZ_ATA,FZ_STP_10」4バイト+4行目のコマンド「CLRQ (LOW R_FUT_PHS)」2バイト+5行目のコマンド「CLRQ (LOW R_FZ_STP)」2バイト+6行目のコマンド「RET」1バイト=10バイトとなり、総

50

実行サイクルは、2行目3(1)サイクル+3行目6(5)サイクル+4行目2サイクル+5行目2サイクル+6行目3サイクル=16(13)サイクルとなる。なお、括弧内のサイクル数は、コマンド「RET NTZ」、「JCPQ Z,・・・」によって移動しなかった場合の実行サイクルを示している。

【1607】

したがって、図188(a)のコマンド群を図188(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減される。かかる置き換えによって、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1608】

また、図188(a)と図188(b)とを比較して理解できるように、図188(a)において2行(3行目~4行目)を占有していたコマンド群を、図188(b)においては1行(3行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1609】

また、図188(b)の例では、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図188(a)の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1610】

〈乱数発生器〉

図189は、乱数発生器740~746の構成を説明するためのブロック図である。上記したように、メインCPU300a、500aには、複数の乱数発生器が設けられている。

【1611】

メインCPU300a、500aには、例えば、図189に示すように、16ビットの最大値を設定可能な乱数発生器である最大値設定乱数発生器740が4チャンネル、16ビットの最大値が65536に固定された乱数発生器である最大値固定乱数発生器742が4チャンネル、8ビットの最大値を設定可能な乱数発生器である最大値設定乱数発生器744が8チャンネル、8ビットの最大値が255に固定された乱数発生器である最大値固定乱数発生器746が8チャンネル設けられている。これらの乱数発生器740~746は、ハードウェア乱数発生器(ハードウェア乱数生成部)である。

【1612】

最大値設定乱数発生器740は、乱数更新周期を32~47クロックのいずれかに設定可能であり、スタート値を、0001h、IDナンバーを基にした値、システムリセット毎に変更される値のいずれかに設定可能である。最大値固定乱数発生器742は、乱数更新周期が1クロックであり、スタート値がシステムリセット毎に変更される値である。なお、システムリセットは、ハード的に状態をリセットするものであり、電源投入時、所定の異常が発生したときに実行される。

【1613】

最大値設定乱数発生器744は、乱数更新周期を16~31クロックのいずれかに設定可能であり、スタート値を、01h、IDナンバーを基にした値、システムリセット毎に変更される値のいずれかに設定可能である。最大値固定乱数発生器746は、乱数更新周期が1クロックであり、スタート値がシステムリセット毎に変更される値である。上記ステップS100-1の初期設定処理において、スタート値の設定および保存、最大値設定乱数発生器740、744の乱数更新周期の設定および保存が行われる。

【1614】

また、これらの乱数発生器740~746は、システムリセット毎にスタート値が更新される。そして、最大値設定乱数発生器740は、ハードウェアにより、内部システムクロックの32クロック~47クロックのいずれかで1回、乱数(乱数カウンタ)を更新す

10

20

30

40

50

る。最大値設定乱数発生器 744 は、ハードウェアにより、内部システムクロックの 16 クロック～31 クロックのいずれかで 1 回、乱数（乱数カウンタ）を更新する。最大値固定乱数発生器 742、746 は、ハードウェアにより、内部システムクロックの 1 クロックで 1 回、乱数（乱数カウンタ）を更新する。また、これらの乱数発生器 740～746 は、一定の規則に従って乱数を更新し、乱数列を一巡するごとに、自動的に乱数列が更新される。そして、ソフトウェアにより、遊技球が所定領域を通過したと判定されたときに、乱数値レジスタから乱数（ハードウェア乱数カウント値）が取得される。

【1615】

また、メイン CPU 300a、500a では、乱数発生器 740～746 から取得した乱数値に、プログラム内において所定の数値を乗じ、また、除算することでソフトウェア乱数を生成する（ソフトウェア乱数生成部）ことも可能である。ソフトウェア乱数生成部は、4ms 毎に発生する割込み処理が実行されていない間に乱数を更新する。具体的には、ソフトウェア乱数生成部は、前回の乱数と最大値とを比較し、前回の乱数が最大値に達していた場合には乱数を 0 にし、前回の乱数が最大値に達していない場合には乱数を 1 加算する。そして、遊技球が所定領域を通過したと判定されたときに、乱数（ソフトウェア乱数カウント値）が取得される。

【1616】

ここで、遊技機 100 では、乱数として、大当たり決定乱数、当たり図柄乱数、リーチグループ決定乱数、リーチモード決定乱数、変動パターン乱数、当たり決定乱数が設けられている。

【1617】

大当たり決定乱数は 0～65535 の範囲であり、当たり図柄乱数は 0～99 の範囲であり、リーチグループ決定乱数は 0～10006 の範囲であり、リーチモード決定乱数は 0～250 の範囲であり、変動パターン乱数は 0～238 の範囲であり、当たり決定乱数は 0～99 の範囲である。

【1618】

また、普通図柄の当たり図柄を複数設ける場合には、普図図柄乱数を設ける必要があり、例えば、0～99 の範囲となる。さらに、普通図柄の変動パターン（変動時間）を複数設ける場合には、普図変動パターン乱数を設ける必要があり、例えば、0～99 の範囲となる。

【1619】

図 190 は、乱数生成部の組み合わせを説明する図である。例えば、大当たり決定乱数は、0～65535 までの乱数を発生させる必要があるため、最大値を 65535 に設定した最大値設定乱数発生器 740 により生成された 16 ビットの乱数を用いる。当たり図柄乱数は、0～99 までの乱数を発生させる必要があるため、最大値を 99 に設定したソフトウェア乱数生成部により生成された 8 ビットの乱数を用いる。

【1620】

リーチグループ決定乱数は、0～10006 までの乱数を発生させる必要があるため、最大値を 10006 に設定した最大値設定乱数発生器 740 により生成された 16 ビットの乱数を用いる。リーチモード決定乱数は、0～250 までの乱数を発生させる必要があるため、最大値を 250 に設定したソフトウェア乱数生成部により生成された 8 ビットの乱数を用いる。変動パターン乱数は、0～238 までの乱数を発生させる必要があるため、最大値を 238 に設定したソフトウェア乱数生成部により生成された 8 ビットの乱数を用いる。

【1621】

当たり決定乱数は、0～99 までの乱数を発生させる必要があるため、最大値を 99 に設定した最大値設定乱数発生器 744 により生成された 8 ビットの乱数を用いる。普図図柄乱数は、0～99 までの乱数を発生させる必要があるため、最大値を 99 に設定したソフトウェア乱数生成部により生成された 8 ビットの乱数を用いる。普図変動パターン乱数は、0～99 までの乱数を発生させる必要があるため、最大値を 99 に設定したソフトウ

10

20

30

40

50

エア乱数生成部により生成された 8 ビットの乱数を用いる。

【1622】

ここで、ソフトウェア乱数生成部により生成された乱数を用いる場合、図 23 のステップ S100-61、図 26 のステップ S400-17、図 23 のステップ S100-69 において、乱数の初期値を決定したり、乱数を更新する処理を実行する必要がある。そうすると、遊技制御処理を行うための制御領域の容量が圧迫されるおそれがある。

【1623】

そこで、図 190 (b)、(c) に示すように、例えば、大当たり決定乱数は 0 ~ 65535 の範囲とし、当たり図柄乱数は 0 ~ 99 の範囲とし、リーチグループ決定乱数は 0 ~ 10006 の範囲とし、リーチモード決定乱数は 0 ~ 1000 の範囲とし、変動パターン乱数は 0 ~ 255 の範囲とし、当たり決定乱数は 0 ~ 65535 の範囲とし、普図図柄乱数は 0 ~ 99 の範囲とし、普図変動パターン乱数は 0 ~ 255 の範囲とする。なお、これらの乱数の範囲は一例であり、最大値固定乱数発生器 742 の最大値である 65535 を最大値とした乱数、最大値固定乱数発生器 746 の最大値である 255 を最大値とした乱数をどの乱数に設定するようにしてもよい。

【1624】

そして、図 190 (b) に示すように、大当たり決定乱数は、最大値を 65535 に設定した最大値設定乱数発生器 740 により生成された 16 ビットの乱数を用いる。当たり図柄決定乱数は、最大値が 99 に設定された最大値設定乱数発生器 740 により生成された 8 ビットの乱数を用いる。リーチグループ決定乱数は、最大値が 10006 に設定された最大値設定乱数発生器 740 により生成された 16 ビットの乱数を用いる。リーチモード決定乱数は、最大値を 1000 に設定した最大値設定乱数発生器により生成された 16 ビットの乱数を用いる。変動パターン乱数は、最大値が 255 に固定された最大値固定乱数発生器 746 により生成された 8 ビットの乱数を用いる。

【1625】

当たり決定乱数は、最大値を 65535 に設定した最大値設定乱数発生器 740 により生成された 16 ビットの乱数を用いる。普図図柄乱数は、最大値を 99 に設定した最大値設定乱数発生器 744 により生成された 8 ビットの乱数を用いる。普図変動パターン乱数は、最大値が 255 に固定された最大値固定乱数発生器 746 により生成された 8 ビットの乱数を用いる。

【1626】

また、図 190 (c) に示すように、大当たり決定乱数および当たり決定乱数は、最大値が 65535 に固定された最大値固定乱数発生器 742 により生成された 16 ビットの乱数を用いてもよい。最大値設定乱数発生器 740 は、スタート値、乱数更新周期、乱数の最大値等を適宜設定可能であるが、最大値固定乱数発生器 742 よりも乱数更新周期が遅い。そのため、遊技者の利益に直結する当否乱数（大当たり決定乱数および当たり決定乱数）については、乱数更新周期が早く、乱数の読み取りがされ辛い最大値固定乱数発生器 742 を用いるとよい。

【1627】

このように、遊技の進行に用いられる複数の乱数に、乱数発生器 740 ~ 746（ハードウェア乱数生成部）により生成された乱数を用いることにより、上記した図 23 のステップ S100-61、図 26 のステップ S400-17、図 23 のステップ S100-69 において、乱数の初期値を決定したり、乱数を更新する処理を実行する必要がなくなる。これにより、約 35 バイトのメイン ROM 300b の容量を削減することができるとともに、約 4 バイトのメイン RAM 300c の容量を削減することができる。かくして、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1628】

〈コマンド「OR」〉

図 191 は、SMC__ROT モジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、定常回転中処理を実行する SMC__ROT モジュールの

10

20

30

40

50

一部である、インデックスフラグに、リールの位置を示すフェーズフラグを合成する処理を挙げて説明する。ここで、インデックスフラグは、リール410a、410b、410cが定常回転の速度となった後、インデックス信号を取得したときにONされる1バイト値で表されるフラグである。例えば、左リール410aのインデックス信号を取得するとインデックスフラグのビット0が1となり、中リール410bのインデックス信号を取得するとインデックスフラグのビット1が1となり、右リール410cのインデックス信号を取得するとインデックスフラグのビット2が1となる。かかるインデックスフラグを参照することで、いずれのリール410a、410b、410cがインデックス信号を取得済みか確認することが可能となる。また、フェーズフラグは、処理中であるリール410a、410b、410cを特定する1バイトで示されるフラグである。例えば、左リール410aの処理中はフェーズフラグのビット0が1となり(00000001B)、中リール410bの処理中はフェーズフラグのビット1が1となり(00000010B)、右リール410cの処理中はフェーズフラグのビット2が1となる(00000100B)。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該SMC__ROTモジュールの説明中、第1レジスタはCレジスタであり、第2レジスタはHLレジスタであり、第1レジスタおよび第2レジスタと異なるレジスタはAレジスタである。

【1629】

かかるSMC__ROTモジュールを実行する段階では、既に、所定のレジスタ(例えば、Cレジスタ)にフェーズフラグの値が保持されているとする。そして、メインCPU500aは、図191のように、任意の処理を実行する。メインCPU500aは、まず、HLレジスタに、論理和の対象となるインデックスフラグ(変数)のアドレスを設定する(S1)。そして、メインCPU500aは、インデックスフラグにフェーズフラグを合成するため、既に設定されているCレジスタの値(フェーズフラグ)と、HLレジスタで示されるアドレスに格納された1バイト値との論理和を計算し、論理和の結果を、インデックスフラグに反映するために、HLレジスタで示されるアドレスに格納されている値に上書きする(S2)。ここで、合成は、変数であるフェーズフラグのいずれかのビットが立っていると(1であれば)、インデックスフラグの同ビットも立たせる(1にする)ことを言う。したがって、処理中のリール410a、410b、410cについてはフェーズフラグが立っているので、インデックス信号の取得有無に拘わらず、合成により、インデックスフラグの該当ビットが立つこととなる。また、全てのインデックス信号が取得されると(全リール410a、410b、410cが定常回転速度となると)、フェーズフラグに拘わらず、インデックスフラグの各リールに相当するビットが全て立つ(00000111B)こととなる。そして、励磁パターン更新前処理を実行するPRE__PLSモジュールに移動することで当該SMC__ROTモジュールを終了する(S3)。こうして、インデックスフラグにフェーズフラグを合成することが可能となる。

【1630】

図192は、SMC__ROTモジュールを実現するためのコマンドの一例を説明するための説明図である。図191で示したフローチャートは、例えば、図192に示したプログラムによって実現される。

【1631】

図192の1行目の指標「SMC__ROT:」は、当該SMC__ROTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ HL, LOW __IDX__FND」によって、Qレジスタの値をHレジスタに読み出し、アドレス「__IDX__FND」の下位1バイトの値をLレジスタに読み出す。かかる2行目のコマンドが、図191のステップS1に対応する。

【1632】

図192の3行目のコマンド「LD A, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値(インデックスフラグの値)がAレジスタに読み出される。4行目のコマンド「OR A, C」によって、Aレジスタの値とCレジスタの値の

10

20

30

40

50

論理和が計算され、その計算結果によってAレジスタが更新される。そして、5行目のコマンド「LD (HL), A」によって、Aレジスタの値をHLレジスタで示されるアドレスに格納する。すなわち、Aレジスタの値を、HLレジスタで示されるアドレスに格納された1バイト値に上書きし、インデックスフラグの値を更新する。かかる3～5行目のコマンドが、図191のステップS2に対応する。

【1633】

図192の6行目のコマンド「JP PRE_PLS」によって、PRE_PLSモジュールの先頭アドレスを示す指標「PRE_PLS:」に移動する。かかる6行目のコマンドが、図191のステップS3に対応する。

【1634】

このように、図192に示したSMC_ROTモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LDQ HL, LOW_IDX_FND」2バイト+3行目のコマンド「LD A, (HL)」1バイト+4行目のコマンド「OR A, C」1バイト+5行目のコマンド「LD (HL), A」1バイト+6行目のコマンド「JP PRE_PLS」3バイト=8バイトとなり、総実行サイクルは、2行目2サイクル+3行目2サイクル+4行目1サイクル+5行目2サイクル+6行目3サイクル=10サイクルとなる。かかるSMC_ROTモジュールにおいて3～5行目の処理を設けることによって、変数として格納されているインデックスフラグとフェーズフラグとを容易に合成（論理和）することが可能となる。

【1635】

図193は、SMC_ROTモジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図192のSMC_ROTモジュールのコマンド群を、図193のSMC_ROTモジュールのコマンド群に置き換えている。

【1636】

図193の1行目の指標「SMC_ROT:」は、当該SMC_ROTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ HL, LOW_IDX_FND」によって、Qレジスタの値をHレジスタに読み出し、アドレス「_IDX_FND」の下位1バイトの値をLレジスタに読み出す。

【1637】

図193の3行目のコマンド「OR (HL), C」によって、HLレジスタで示されるアドレスに格納された1バイト値と、Cレジスタの値との論理和が計算され、その計算結果をHLレジスタで示されるアドレスに格納する。すなわち、計算結果を、HLレジスタで示されるアドレスに格納された1バイト値に上書きし、インデックスフラグの値を更新する。かかる3行目のコマンドが、図191のステップS2に対応する。

【1638】

図193の4行目のコマンド「JP PRE_PLS」によって、PRE_PLSモジュールの先頭アドレスを示す指標「PRE_PLS:」に移動する。かかる4行目のコマンドが、図191のステップS3に対応する。

【1639】

図193に示したSMC_ROTモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LDQ HL, LOW_IDX_FND」2バイト+3行目のコマンド「OR (HL), C」2バイト+4行目のコマンド「JP PRE_PLS」3バイト=7バイトとなり、総実行サイクルは、2行目2サイクル+3行目5サイクル+4行目3サイクル=10サイクルとなる。したがって、図192の場合と比べ、総コマンドサイズが1バイト削減される。かかるSMC_ROTモジュールによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1640】

また、図192と図193とを比較して理解できるように、図192において3行（3行目～5行目）を占有していたコマンド群を、図193においては1行（3行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

10

20

30

40

50

【1641】

なお、ここでは、論理和を実行するコマンド「OR (HL), C」を挙げて説明したが、かかる場合に限らず、論理積を実行するコマンド「AND (HL), C」や排他的論理和「XOR (HL), C」であっても、論理和同様の置き換えができ、総コマンドサイズを1バイト削減することができる。また、変数のアドレスを格納するレジスタはHLレジスタに限らずDEレジスタでもよい。さらに、論理和の対象となるレジスタはCレジスタに限らず、Aレジスタ、Bレジスタ、Dレジスタ、Eレジスタ、Hレジスタ、Lレジスタを採用することもできる。

【1642】

また、ここでは、図192の例と比較して、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図192の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1643】

〈コマンド「LDF」〉

図194は、INITIALモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図82のステップS2000で示したCPU初期化処理を実行するINITIALモジュールの一部である内蔵レジスタエリア設定処理を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該INITIALモジュールの説明中、第1レジスタはHLレジスタである。

【1644】

メインCPU500aは、図194のように、任意の処理を実行する。メインCPU500aは、まず、HLレジスタに、CPU内蔵レジスタ設定データテーブル（データテーブル）の先頭アドレスを設定する（S1）。そして、メインCPU500aは設定データ数を設定する（S2）。

【1645】

続いて、メインCPU500aは、入出力部を通じてアクセス可能な内蔵レジスタのアドレスおよび設定データをそれぞれCレジスタおよびAレジスタに設定する（S3）。次に、Cレジスタで示される内蔵レジスタに、Aレジスタの値（設定データ）を出力する（S4）。続いて、メインCPU500aは、Bレジスタの値（設定データ数）をデクリメント（「1」減算）し、デクリメントした結果が0であるか否か判定する（S5）。ここで、デクリメントした結果が0でなければ（S5におけるNO）、ステップS3からの処理を繰り返し、デクリメントした結果が0であれば（S5におけるYES）、次の処理に移行する。

【1646】

図195は、INITIALモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図195（a）は、INITIALモジュールのコマンド群を示し、図195（b）は、メインROM500bのプログラムデータにおける1バイトデータ群の配置を示す。図194で示したフローチャートは、例えば、図195に示したプログラムによって実現される。

【1647】

図195（a）の1行目の指標「INITIAL:」は、当該INITIALモジュールの先頭アドレスを示す。2行目のコマンド「LD HL, T_CPU_REG」によって、CPU内蔵レジスタ設定データテーブルの先頭アドレス「T_CPU_REG」がHLレジスタに読み出される。かかる2行目のコマンドが、図194のステップS1に対応する。3行目のコマンド「LD B, @NUM_CPU_REG」によって、アドレス「@NUM_CPU_REG」に格納されている設定データ数がBレジスタに読み出される。かかる3行目のコマンドが、図194のステップS2に対応する。

【1648】

なお、図195 (b) の22行目を参照して理解できるように、アドレス「@NUM_CPU_REG」には、コマンド「`EQ ( $-T_CPU_REG ) / 2`」が記載されている。ここで、「\$」は、転送元となる1バイトデータ群の最終アドレスの次のアドレス、すなわち、アドレス「@NUM_CPU_REG」自体であり、「T_CPU_REG」は転送元となる1バイトデータ群の先頭アドレスなので、`$-T_CPU_REG`の値（データ数を示す1バイト値）の値が、1バイト値の総バイト数となり、その値を、アドレスと値との組み合わせ数である2で除算すると、設定データ数が導出される。例えば、図195 (b) の例では、設定データ数は $20 / 2 = 10$ となる。

【1649】

図195 (a) の4行目の指標「INITIAL01:」は繰り返し処理の先頭アドレスを示す。5行目のコマンド「`LD IN AC, (HL)`」によって、HLレジスタで示されるアドレスに格納された値がCレジスタに格納され、HLレジスタに「1」を加えた値で示されるアドレスに格納された値がAレジスタに格納される。かかる5行目のコマンドが、図194のステップS3に対応する。6行目のコマンド「`OUT (C), A`」によって、Cレジスタで示される値（内蔵レジスタ）に、Aレジスタの値（設定データ）が出力される。かかる6行目のコマンドが、図194のステップS4に対応する。

【1650】

例えば、HLレジスタがアドレス「T_CPU_REG」の値であった場合、図195 (b) の2行目における1バイト値「@PTOPR_」がCレジスタに格納され、図195 (b) の3行目における「120」の値がAレジスタに格納される。そして、Cレジスタの値、すなわち「@PTOPR_」で示されるアドレスに、Aレジスタの値、すなわち、「120」の値を出力する。

【1651】

続いて、図195 (a) の7行目のコマンド「`DJNZ INITIAL01`」によって、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0でなければ、アドレス「INITIAL01」に移動し、デクリメントした結果が0であれば、当該コマンドの次のコマンドに処理を移す。ここでは、データ数が「10」なので、「INITIAL01」からの処理を10回繰り返すとBレジスタの値が0となる。かかる7行目のコマンドが、図194のステップS5に対応する。

【1652】

このように、図195 (a) に示したINITIALモジュールのコマンドの総コマンドサイズは、2行目のコマンド「`LD HL, T_CPU_REG`」3バイト+3行目のコマンド「`LD B, @NUM_CPU_REG`」2バイト+5行目のコマンド「`LD IN AC, (HL)`」2バイト+6行目のコマンド「`OUT (C), A`」2バイト+7行目のコマンド「`DJNZ INITIAL01`」2バイト=11バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+5行目4サイクル+6行目3サイクル+7行目3サイクル（2サイクル）=15サイクル（14サイクル）となる。なお、括弧内のサイクル数は、コマンド「`DJNZ INITIAL01`」によって移動しなかった場合の実行サイクルを示している。なお、ここでは、設定データ数=10なので、5～7行目のコマンドが10回繰り返されることとなる。したがって、実際の総実行サイクルは、2行目3サイクル+3行目2サイクル+（5行目4サイクル+6行目3サイクル+7行目3サイクル）×9ループ+（5行目4サイクル+6行目3サイクル+7行目2サイクル）=104サイクルとなる。かかるINITIALモジュールによって、内部レジスタに設定データを設定することができる。

【1653】

図196は、INITIALモジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図195 (a) のINITIALモジュールのコマンド群を、図196 (a) のINITIALモジュールのコマンド群に置き換えており、他の、図195 (b) のメインROM500bのプログラムデータにおける1バイトデータ群

10

20

30

40

50

は、図196 (b) として、そのまま用いている。

【1654】

図196 (a) の1行目の指標「INITIAL:」は、当該INITIALモジュールの先頭アドレスを示す。2行目のコマンド「LDF HL, T_CPU_REG」によって、CPU内蔵レジスタ設定データテーブルの先頭アドレス「T_CPU_REG」がHLレジスタに読み出される。かかる2行目のコマンドが、図194のステップS1に対応する。

【1655】

ここで、コマンド「LDF HL, mn」は、「mn」の値をHLレジスタに格納するコマンドである。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「2」である。一方、一般的に用いられるロードコマンド「LD HL, mn」も「mn」の値をHLレジスタに格納するコマンドであるが、そのコマンドのコマンドサイズは「3」であり、実行サイクルは「3」である。したがって、mnがメモリマップ上の1200H～1DFFHの範囲であり、かつ、HLレジスタにロードすることを条件に、「LD」を「LDF」に変更することで、コマンドサイズが1バイト削減され、実行サイクルも1サイクル削減される。

【1656】

このように、コマンド「LDF HL, mn」におけるmnの値は、メモリマップにおける1200H～1DFFHのいずれかでなければならない。そこで、本実施形態では、メインROM500bの使用領域における1200H～1DFFHの範囲にプログラムデータを配置し、かかる範囲のデータを読み出す場合にのみコマンド「LDF HL, mn」を用いることとする。なお、「LDF HL, 1200H」のアセンブルコードは、対象となるアドレス「1200H」に5800Hを加算した6A00Hと表される。したがって、「LDF HL, 1200H」～「LDF HL, 1DFFH」のアセンブルコードは、6A00H～75FFHとなる。

【1657】

図196 (a) の3行目のコマンド「LD B, @NUM_CPU_REG」によって、アドレス「@NUM_CPU_REG」に格納されている設定データ数がBレジスタに読み出される。かかる3行目のコマンドが、図194のステップS2に対応する。

【1658】

図196 (a) の4行目の指標「INITIAL01:」は繰り返し処理の先頭アドレスを示す。5行目のコマンド「LDIN AC, (HL)」によって、HLレジスタで示されるアドレスに格納された値がCレジスタに格納され、HLレジスタに「1」を加えた値で示されるアドレスに格納された値がAレジスタに格納される。かかる5行目のコマンドが、図194のステップS3に対応する。6行目のコマンド「OUT (C), A」によって、Cレジスタで示される値(内蔵レジスタ)に、Aレジスタの値(設定データ)が出力される。かかる6行目のコマンドが、図194のステップS4に対応する。

【1659】

続いて、図196 (a) の7行目のコマンド「DJNZ INITIAL01」によって、Bレジスタの値がデクリメント(「1」減算)され、デクリメントした結果が0でなければ、アドレス「INITIAL01」に移動し、デクリメントした結果が0であれば、当該コマンドの次のコマンドに処理を移す。ここでは、データ数が「10」なので、「INITIAL01」からの処理を10回繰り返すとBレジスタの値が0となる。かかる7行目のコマンドが、図194のステップS5に対応する。

【1660】

このように、図196 (a) に示したINITIALモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LDF HL, T_CPU_REG」2バイト+3行目のコマンド「LD B, @NUM_CPU_REG」2バイト+5行目のコマンド「LDIN AC, (HL)」2バイト+6行目のコマンド「OUT (C), A」2バイト+7行目のコマンド「DJNZ INITIAL01」2バイト=10バイトとなり、総実行

10

20

30

40

50

サイクルは、2行目2サイクル+3行目2サイクル+5行目4サイクル+6行目3サイクル+7行目3サイクル(2サイクル)=14サイクル(13サイクル)となる。なお、括弧内のサイクル数は、コマンド「DJNZ INITIAL01」によって移動しなかった場合の実行サイクルを示している。なお、ここでは、設定データ数=10なので、5~7行目のコマンドが10回繰り返されることとなる。したがって、実際の総実行サイクルは、2行目2サイクル+3行目2サイクル+(5行目4サイクル+6行目3サイクル+7行目3サイクル)×9ループ+(5行目4サイクル+6行目3サイクル+7行目2サイクル)=103サイクルとなる。したがって、図195(a)の場合と比べ、総コマンドサイズが1バイト削減され、総実行サイクルが1サイクル削減されていることが理解できる。かかるINITIALモジュールによって、コマンドの更なる短縮化、および、処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1661】

なお、上述した実施形態では、INITIALモジュールを挙げ、内部レジスタに設定データを設定する処理において、コマンド「LD HL, T_CPU_REG」をコマンド「LDF HL, T_CPU_REG」に置き換え、コマンドの更なる短縮化、および、処理負荷の軽減を図る例を挙げて説明した。しかし、かかるコマンドは、INITIALモジュールに限らず、様々なモジュールにも適用できる。

【1662】

例えば、表示データ変換処理を実行するDGEXTRNモジュールにおいても、コマンド「LD HL, T_SEG_PTN」が存在し、INITIALモジュール同様、メイン表示器表示パターンデータテーブルの先頭アドレス「T_SEG_PTN」がHLレジスタに読み出されている。また、先頭アドレス「T_SEG_PTN」は、1200H~1DFFHの範囲に含まれる。したがって、DGEXTRNモジュールにおいても、コマンド「LD HL, T_SEG_PTN」をコマンド「LDF HL, T_SEG_PTN」に変更することで、コマンドの更なる短縮化、および、処理負荷の軽減を図ることができる。

20

【1663】

また、図柄配列データ取得処理を実行するGET_ARRAYモジュールにおいても、コマンド「LD HL, T_FIG_LIN」が存在し、INITIALモジュール同様、有効ライン別図柄配列オフセットテーブルの先頭アドレス「T_FIG_LIN」がHLレジスタに読み出されている。また、先頭アドレス「T_FIG_LIN」は、1200H~1DFFHの範囲に含まれる。したがって、GET_ARRAYモジュールにおいても、コマンド「LD HL, T_FIG_LIN」をコマンド「LDF HL, T_FIG_LIN」に変更することで、コマンドの更なる短縮化、および、処理負荷の軽減を図ることができる。

30

【1664】

このように、転送元のアドレスを1200H~1DFFHの範囲に収めることで、ほとんどのコマンド「LD HL, mn」をコマンド「LDF HL, mn」に置き換えることができる。かかる置き換えを本実施形態における全ての処理に適用することで、例えば、総コマンドサイズを80~90バイト短縮化することも可能となる。

40

【1665】

かかるコマンド「LDF」は、上述したINITIALモジュールのみならず、様々なモジュール、例えば、RANKSETモジュール等に採用することができる。また、スロットマシンのみならず、パチンコ機における、例えば、PWRFAILモジュールやDYM_OUTモジュール等にも採用することができる。

【1666】

図197は、RANKSETモジュールを説明するための説明図である。図85のステップS2020に示したRANKSETモジュールは、設定値切り替え処理、すなわち、設定値を切り替え、設定値データを更新する。なお、ここでは、RANKSETモジュール中のステップS2020-5の設定値切り替え時データテーブルセット処理を挙げて説

50

明する。

【1667】

図197(a)の1行目の指標「RANKSET:」は、当該RANKSETモジュールの先頭アドレスを示す。2行目のコマンド「LD HL, T_RNK_SET」によって、設定値切り替え時データテーブルの先頭アドレス「T_RNK_SET」がHLレジスタに読み出される。3行目のコマンド「RST TABLSET」によって、サブルーチンとして、図138を用いて説明したTABLSETモジュールが呼び出される。かかるTABLSETモジュールについては既に説明されているので、ここでは、その詳細な説明を省略する。ここでは、設定値切り替え時データテーブルにおける2つの変数（__WIN_DAT、__EXT_REQ）が所定値に設定される。

10

【1668】

ここで、コマンド「LDF」への置き換えを行うと、図197(a)のコマンド群を図197(b)のように変更することができる。図197(b)の1行目の指標「RANKSET:」は、当該RANKSETモジュールの先頭アドレスを示す。2行目のコマンド「LDF HL, T_RNK_SET」によって、設定値切り替え時データテーブルの先頭アドレス「T_RNK_SET」がHLレジスタに読み出される。なお、設定値切り替え時データテーブルの先頭アドレス「T_RNK_SET」はメインROM500bの使用領域における1200H~1DFFHの範囲に配置される。3行目のコマンド「RST TABLSET」によって、サブルーチンとして、図138を用いて説明したTABLSETモジュールが呼び出される。こうして、図197(a)同様、設定値切り替え時データテーブルにおける2つの変数（__WIN_DAT、__EXT_REQ）が所定値に設定される。

20

【1669】

ここで、図197(a)に示したRANKSETモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD HL, T_RNK_SET」3バイト+3行目のコマンド「RST TABLSET」1バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目4サイクル=7サイクルとなる。一方、図197(b)に示したRANKSETモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LDF HL, T_RNK_SET」2バイト+3行目のコマンド「RST TABLSET」1バイト=3バイトとなり、総実行サイクルは、2行目2サイクル+3行目4サイクル=6サイクルとなる。したがって、図197(a)の場合と比べ、総コマンドサイズが1バイト削減され、総実行サイクルが1サイクル削減されていることが理解できる。かかるRANKSETモジュールによって、コマンドの更なる短縮化、および、処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

30

【1670】

図198は、PWRFAILモジュールを説明するための説明図である。図25のステップS300に示したPWRFAILモジュールは、電源断時退避処理、すなわち、設定値を切り替え、設定値データを更新する。なお、ここでは、PWRFAILモジュール中のステップS300-11の出力ポートクリア処理を挙げて説明する。

【1671】

ここでは、出力ポートの初期値として予め00HがAレジスタに設定されているとする。図198(a)の1行目の指標「PWRFAIL:」は、当該PWRFAILモジュールの先頭アドレスを示す。2行目のコマンド「LD HL, D_OUT_PRT」によって、出力ポートアドレステーブルの先頭アドレス「D_OUT_PRT」がHLレジスタに読み出される。3行目のコマンド「LD B, @PRT_CLR_LOP」によって、アドレス「@PRT_CLR_LOP」に格納されている出力ポート初期化ループカウンタ値（ここでは13）がループ回数としてBレジスタに読み出される。

40

【1672】

図198(a)の4行目の指標「PWRFAIL_20:」は繰り返し処理の先頭アドレスを示す。5行目のコマンド「LD C, (HL)」によって、HLレジスタで示され

50

るアドレスに格納された値（出力ポートアドレス）がCレジスタに読み出される。6行目のコマンド「OUT（C），A」によって、Cレジスタで示される値（内蔵レジスタ）に、Aレジスタの値（ここでは00H）が出力され、出力ポートがクリアされる。

【1673】

続いて、図198（a）の7行目のコマンド「INC HL」によってHLレジスタの値が1だけインクリメントされる。8行目のコマンド「DJNZ PWRFAIL__20」によって、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0でなければ、アドレス「PWRFAIL__20」に移動し、デクリメントした結果が0であれば、当該コマンドの次のコマンドに処理を移す。ここでは、データ数が「13」なので、「PWRFAIL__20」からの処理を13回繰り返すとBレジスタの値が0となる。こうして、出力ポートがクリアされる。

【1674】

ここで、コマンド「LDF」への置き換えを行うと、図198（a）のコマンド群を図198（b）のように変更することができる。ここでも、出力ポートの初期値として予め00HがAレジスタに設定されているとする。図198（b）の1行目の指標「PWRFAIL：」は、当該PWRFAILモジュールの先頭アドレスを示す。2行目のコマンド「LDF HL，D__OUT__PRT」によって、出力ポートアドレステーブルの先頭アドレス「D__OUT__PRT」がHLレジスタに読み出される。なお、出力ポートアドレステーブルの先頭アドレス「D__OUT__PRT」はメインROM500bの使用領域における1200H～1DFFHの範囲に配置される。3行目のコマンド「LD B，@PRT__CLR__LOP」によって、アドレス「@PRT__CLR__LOP」に格納されている出力ポート初期化ループカウンタ値（ここでは13）がループ回数としてBレジスタに読み出される。

【1675】

図198（b）の4行目の指標「PWRFAIL__20：」は繰り返し処理の先頭アドレスを示す。5行目のコマンド「LD C，（HL）」によって、HLレジスタで示されるアドレスに格納された値（出力ポートアドレス）がCレジスタに読み出される。6行目のコマンド「OUT（C），A」によって、Cレジスタで示される値（内蔵レジスタ）に、Aレジスタの値（ここでは00H）が出力され、出力ポートがクリアされる。

【1676】

続いて、図198（b）の7行目のコマンド「INC HL」によってHLレジスタの値が1だけインクリメントされる。8行目のコマンド「DJNZ PWRFAIL__20」によって、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0でなければ、アドレス「PWRFAIL__20」に移動し、デクリメントした結果が0であれば、当該コマンドの次のコマンドに処理を移す。ここでは、データ数が「13」なので、「PWRFAIL__20」からの処理を13回繰り返すとBレジスタの値が0となる。こうして、出力ポートがクリアされる。

【1677】

ここで、図198（a）に示したPWRFAILモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD HL，D__OUT__PRT」3バイト+3行目のコマンド「LD B，@PRT__CLR__LOP」2バイト+5行目のコマンド「LD C，（HL）」1バイト+6行目のコマンド「OUT（C），A」2バイト+7行目のコマンド「INC HL」1バイト+8行目のコマンド「DJNZ PWRFAIL__20」2バイト＝11バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+5行目2サイクル+6行目3サイクル+7行目1サイクル+8行目3サイクル（2サイクル）＝14サイクル（13サイクル）となる。なお、ここでは、出力ポート初期化ループカウンタ値＝13なので、5～8行目のコマンドが13回繰り返されることとなる。したがって、実際の総実行サイクルは、2行目3サイクル+3行目2サイクル+（5行目2サイクル+6行目3サイクル+7行目1サイクル+8行目3サイクル）×12ループ+（5行目2サイクル+6行目3サイクル+7行目1サイクル+8行目2サイクル）＝121サイ

10

20

30

40

50

クルとなる。

【1678】

一方、図198(b)に示したPWRFAILモジュールの命令の総命令サイズは、2行目の命令「LDF HL, D_OUT_PRT」2バイト+3行目の命令「LD B, @PRT_CLR_LOP」2バイト+5行目の命令「LD C, (HL)」1バイト+6行目の命令「OUT (C), A」2バイト+7行目の命令「INC HL」1バイト+8行目の命令「DJNZ PWRFAIL_20」2バイト=10バイトとなり、総実行サイクルは、2行目2サイクル+3行目2サイクル+5行目2サイクル+6行目3サイクル+7行目1サイクル+8行目3サイクル(2サイクル)=13サイクル(12サイクル)となる。なお、ここでは、出力ポート初期化ループカウンタ値=13なので、5~8行目の命令が13回繰り返されることとなる。したがって、実際の総実行サイクルは、2行目2サイクル+3行目2サイクル+(5行目2サイクル+6行目3サイクル+7行目1サイクル+8行目3サイクル)×12ループ+(5行目2サイクル+6行目3サイクル+7行目1サイクル+8行目2サイクル)=120サイクルとなる。したがって、図198(a)の場合と比べ、総命令サイズが1バイト削減され、総実行サイクルが1サイクル削減されていることが理解できる。かかるPWRFAILモジュールによって、命令の更なる短縮化、および、処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1679】

図199は、DYM_OUTモジュールを説明するための説明図である。図26のステップS400-5で示したDYM_OUTモジュールは、第1特別図柄表示器160、第2特別図柄表示器162、第1特別図柄保留表示器164、第2特別図柄保留表示器166、普通図柄表示器168、普通図柄保留表示器170、右打ち報知表示器172、性能表示モニタ184を点灯制御するダイナミックポート出力処理を実行する。

20

【1680】

図199(a)の1行目の指標「DYM_OUT:」は、当該DYM_OUTモジュールの先頭アドレスを示す。2行目の命令「LD HL, D_DYM_SEL」によって、ダイナミック表示選択テーブルの先頭アドレス「D_DYM_SEL」がHLレジスタに読み出される。3行目の命令「RST WORDSEL」によって、サブルーチンとして、図127~図130を用いて説明したWORDSELモジュールが呼び出される。かかるWORDSELモジュールについては既に説明されているので、ここでは、その詳細な説明を省略する。こうして、ダイナミック表示選択テーブルにおける4つの2バイトデータが選択される。

30

【1681】

ここで、命令「LDF」への置き換えを行うと、図199(a)の命令群を図199(b)のように変更することができる。図199(b)の1行目の指標「DYM_OUT:」は、当該DYM_OUTモジュールの先頭アドレスを示す。2行目の命令「LDF HL, D_DYM_SEL」によって、ダイナミック表示選択テーブルの先頭アドレス「D_DYM_SEL」がHLレジスタに読み出される。なお、ダイナミック表示選択テーブルの先頭アドレス「D_DYM_SEL」はメインROM500bの使用領域における1200H~1DFFHの範囲に配置される。3行目の命令「RST WORDSEL」によって、サブルーチンとして、図127~図130を用いて説明したWORDSELモジュールが呼び出される。こうして、ダイナミック表示選択テーブルにおける4つの2バイトデータが選択される。

40

【1682】

ここで、図199(a)に示したDYM_OUTモジュールの命令の総命令サイズは、2行目の命令「LD HL, D_DYM_SEL」3バイト+3行目の命令「RST WORDSEL」1バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目4サイクル=7サイクルとなる。一方、図199(b)に示したDYM_OUTモジュールの命令の総命令サイズは、2行目の命令「LDF H

50

L, D_DYM_SEL」2バイト+3行目のコマンド「RST WORDSEL」1バイト=3バイトとなり、総実行サイクルは、2行目2サイクル+3行目4サイクル=6サイクルとなる。したがって、図199(a)の場合と比べ、総コマンドサイズが1バイト削減され、総実行サイクルが1サイクル削減されていることが理解できる。かかるDYM_OUTモジュールによって、コマンドの更なる短縮化、および、処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1683】

〈コマンド「SET」〉

図200は、IPT_PDモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図98のステップS3100-21で示した状態別モジュール実行処理を実行するためのCAL_MODモジュールにおいて、タイマ割込み処理用フェーズ(0~3)に応じて4回に1回(5.96ms毎に)選択的に移行し、外部信号出力制御処理を実行するIPT_PDモジュールの一部である外部信号4を設定する処理を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該IPT_PDモジュールの説明中、所定のレジスタはHLレジスタである。

【1684】

メインCPU500aは、図200のように、任意の処理を実行する。メインCPU500aは、まず、外部信号出力保持タイマに関するカウンタの減算処理を行う(S1)。そして、メインCPU500aは、HLレジスタに、出力ポート3イメージのアドレスを設定する(S2)。続いて、メインCPU500aは、ステップS1の減算処理によりセキュリティ信号ON条件を満たすか否か判定し、ON条件を満たしていれば、外部信号4、すなわち、HLレジスタで示されるアドレスの値の所定のビットを立たせる(1にする)(S3)。そして、当該IPT_PDモジュールを終了して1段上のルーチンに戻る(S4)。こうして、外部信号4を設定することが可能となる。

【1685】

図201は、IPT_PDモジュールを実現するためのコマンドの一例を説明するための説明図である。図200で示したフローチャートは、例えば、図201に示したプログラムによって実現される。

【1686】

図201の1行目の指標「IPT_PD:」は、当該IPT_PDモジュールの先頭アドレスを示す。2行目のコマンド「RST RAM_DEC」は、変数を1デクリメントする汎用モジュールを呼び出すコマンドであり、このコマンドによって外部信号出力保持タイマに関するカウンタが減算される。ここでは、コマンド「RST RAM_DEC」によって少なくともゼロフラグが更新される。かかる2行目のコマンドが、図200のステップS1に対応する。3行目のコマンド「LDQ HL, LOW_OUT_PT3」によって、Qレジスタの値をHレジスタに読み出し、アドレス「_OUT_PT3」の下位1バイトの値をLレジスタに読み出す。コマンド「LDQ HL, LOW_OUT_PT3」によってはゼロフラグが変化しない。かかる3行目のコマンドが、図200のステップS2に対応する。

【1687】

図201の4行目のコマンド「JR Z, IPT_PD03」によって、2行目のコマンド「RST RAM_DEC」の減算結果がゼロであるか否か判定され、ゼロであれば(ゼロフラグが立っていれば)、すなわち、セキュリティ信号ON条件を満たしていなければ、6行目の指標「IPT_PD03:」に移動し、ゼロでなければ(ゼロフラグが立っていなければ)、次の処理を行う。5行目のコマンド「SET @EXT_SG4_POS, (HL)」は、出力ポート3イメージ、すなわち、HLレジスタが示すアドレスに格納されている値における、@EXT_SG4_POSで定義される値(ここでは、2)のビットを立たせる。6行目の指標「IPT_PD03:」は4行目のコマンド「JR Z, IPT_PD03」の移動先を示す。かかる4~6行目のコマンドが、図200のス

10

20

30

40

50

テップ S 3 に対応する。そして、図 2 0 1 の 7 行目のコマンド「RET」によって、1 段上のルーチンに戻る。かかる 7 行目のコマンドが、図 2 0 0 のステップ S 4 に対応する。

【1 6 8 8】

このように、図 2 0 1 に示した IPT_PD モジュールのコマンドの総コマンドサイズは、2 行目のコマンド「RST RAM_DEC」1 バイト + 3 行目のコマンド「LDQ HL, LOW _OUT_PT3」2 バイト + 4 行目のコマンド「JR Z, IPT_PD03」2 バイト + 5 行目のコマンド「SET @EXT_SG4_POS, (HL)」2 バイト + 7 行目のコマンド「RET」1 バイト = 8 バイトとなり、総実行サイクルは、コマンド「JR Z, IPT_PD03」によって移動しなかった場合、2 行目 4 サイクル + 3 行目 2 サイクル + 4 行目 2 サイクル + 5 行目 5 サイクル + 7 行目 3 サイクル = 1 6 サイクルとなり、コマンド「JR Z, IPT_PD03」によって移動した場合、2 行目 4 サイクル + 3 行目 2 サイクル + 4 行目 3 サイクル + 7 行目 3 サイクル = 1 2 サイクルとなる。かかる IPT_PD モジュールにおいて 4 ~ 6 行目の処理を設けることによって、減算結果に応じた、出力ポート 3 イメージのビット操作が可能となる。

【1 6 8 9】

図 2 0 2 は、IPT_PD モジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図 2 0 1 の IPT_PD モジュールのコマンド群を、図 2 0 2 の IPT_PD モジュールのコマンド群に置き換えている。

【1 6 9 0】

図 2 0 2 の 1 行目の指標「IPT_PD:」は、当該 IPT_PD モジュールの先頭アドレスを示す。2 行目のコマンド「RST RAM_DEC」は、変数を 1 デクリメントする汎用モジュールを呼び出すコマンドであり、このコマンドによって外部信号出力保持タイマに関するカウンタが減算される。ここでは、コマンド「RST RAM_DEC」によって少なくともゼロフラグが更新される。かかる 2 行目のコマンドが、図 2 0 0 のステップ S 1 に対応する。3 行目のコマンド「LDQ HL, LOW _OUT_PT3」によって、Q レジスタの値を H レジスタに読み出し、アドレス「_OUT_PT3」の下位 1 バイトの値を L レジスタに読み出す。コマンド「LDQ HL, LOW _OUT_PT3」によってはゼロフラグが変化しない。かかる 3 行目のコマンドが、図 2 0 0 のステップ S 2 に対応する。

【1 6 9 1】

図 2 0 2 の 4 行目のコマンド「SET NZ, @EXT_SG4_POS, (HL)」によって、2 行目のコマンド「RST RAM_DEC」の減算結果がゼロでなければ（ゼロフラグが立っていなければ）、すなわち、セキュリティ信号 ON 条件を満たしていれば、出力ポート 3 イメージ、すなわち、HL レジスタが示すアドレスの値における、@EXT_SG4_POS で定義される値（ここでは、2）のビットを立たせる。かかる 4 行目のコマンドが、図 2 0 0 のステップ S 3 に対応する。そして、図 2 0 2 の 5 行目のコマンド「RET」によって、1 段上のルーチンに戻る。かかる 5 行目のコマンドが、図 2 0 0 のステップ S 4 に対応する。

【1 6 9 2】

ここで、コマンド「SET cc, b, (HL)」は、cc に対応するフラグ（ゼロフラグやキャリーフラグ）が真であれば、HL レジスタで示されるアドレスの値（変数）の b ビット目を立てるコマンドである。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「5」（「2」）である。なお、括弧内のサイクル数は、コマンド「SET cc, b, (HL)」においてフラグが真ではなかった場合の実行サイクルを示している。

【1 6 9 3】

このように、図 2 0 2 に示した IPT_PD モジュールのコマンドの総コマンドサイズは、2 行目のコマンド「RST RAM_DEC」1 バイト + 3 行目のコマンド「LDQ HL, LOW _OUT_PT3」2 バイト + 4 行目のコマンド「SET NZ, @EXT_SG4_POS, (HL)」2 バイト + 5 行目のコマンド「RET」1 バイト = 6 バイ

10

20

30

40

50

トとなり、総実行サイクルは、2行目4サイクル+3行目2サイクル+4行目5サイクル(2サイクル)+5行目3サイクル=14サイクル(11サイクル)となる。したがって、図201の場合と比べ、総コマンドサイズが2バイト削減され、総実行サイクルが少なくとも1サイクル削減される。かかるIPT_PDモジュールによって、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1694】

また、図201と図202とを比較して理解できるように、図201において3行(4行目~6行目)を占有していたコマンド群を、図202においては1行(4行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

10

【1695】

なお、ここでは、ビットを設定するコマンド「SET cc, b, (HL)」を挙げて説明したが、かかる場合に限らず、ビットを設定するコマンド「SET cc, b, r」を用いることもできる。ここで、コマンド「SET cc, b, r」は、ccに対応するフラグ(ゼロフラグやキャリーフラグ)が真であれば、r(Aレジスタ、Bレジスタ、Dレジスタ、Eレジスタ、Hレジスタ、Lレジスタ)の値のbビット目を立てるコマンドである。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「2」である。

【1696】

かかるコマンド「SET」は、上述したIPT_PDモジュールのみならず、様々なモジュール、例えば、DYNMOUTモジュール等に採用することができる。また、スロットマシンのみならず、パチンコ機における、例えば、EXT_PRCモジュール等にも採用することができる。

20

【1697】

図203は、DYNMOUTモジュールを説明するための説明図である。図98のステップS3100-7で示したDYNMOUTモジュールでは、メインCPU500aが、セットされた出カイメージを出力ポートに出力し、メインクレジット表示部430、メイン払出表示部432、投入枚数表示器、スタート表示器、ストップスイッチ420a、420b、420cの表示器、リプレイ表示器、区間表示器160を点灯制御するダイナミックポート出力処理(ダイナミック点灯制御処理)が実行される。

【1698】

30

図203(a)の1行目の指標「DYNMOUT:」は、当該DYNMOUTモジュールの先頭アドレスを示す。2行目のコマンド「RESQ @CHN_DSP_BIT, (LOW_WIN_DAT+1)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「_WIN_DAT」の下位1バイトに1を加算した値をアドレスの下位1バイトとし、そのアドレスに格納された値(メイン表示器データバッファ2)における@CHN_DSP_BITが示すビット(ここではビット7)に0を設定する(リセットする)。3行目のコマンド「LDQ A, (LOW_OUT_PT2)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「_OUT_PT2」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(出力ポート2イメージ)をAレジスタに読み出す。4行目のコマンド「JAND Z, A, 00000011B, DYNMOUT02」によって、Aレジスタの値と00000011Bとの論理積がゼロであれば、アドレス「DYNMOUT02」に移動し、ゼロでなければ5行目のコマンドを処理する。5行目のコマンド「SETQ @CHN_DSP_BIT, (LOW_WIN_DAT+1)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「_WIN_DAT」の下位1バイトに1を加算した値をアドレスの下位1バイトとし、そのアドレスに格納された値(メイン表示器データバッファ2)の@CHN_DSP_BITが示すビット(ここではビット7)に1を設定する(セットする)。6行目の指標「DYNMOUT02:」は、4行目のコマンド「JAND Z, A, 00000011B, DYNMOUT02」の移動先を示す。こうして、出力ポート2イメージの下位2ビットの値に応じて、メイン表示器データバッファ2のビット7を設定することが可能

40

50

となる。

【1699】

ここで、コマンド「SET」への置き換えを行うと、図203(a)のコマンド群を図203(b)のように変更することができる。図203(b)の1行目の指標「DYNMOUT:」は、当該DYNMOUTモジュールの先頭アドレスを示す。2行目のコマンド「LDQ HL, LOW __WIN__DAT+1」によって、Qレジスタの値をHレジスタに読み出し、アドレス「__WIN__DAT」の下位1バイトの値に1を加算した値をLレジスタに読み出す。3行目のコマンド「RES @CHN__DSP__BIT, (HL)」によって、HLレジスタで示されるアドレスに格納された値における@CHN__DSP__BITが示すビット（ここではビット7）に0を設定する（リセットする）。4行目のコマンド「LDQ A, (LOW __OUT__PT2)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__OUT__PT2」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（出力ポート2イメージ）をAレジスタに読み出す。5行目のコマンド「AND A, 00000011B」によって、Aレジスタの値と00000011Bとの論理積を演算し、Aレジスタをマスクする。6行目のコマンド「SET NZ, @CHN__DSP__BIT, (HL)」によって、5行目のコマンド「AND A, 00000011B」の演算結果がゼロでなければ（ゼロフラグが立っていなければ）、HLレジスタが示すアドレスの値（メイン表示器データバッファ2）における@CHN__DSP__BITが示すビット（ここでは、ビット7）に1を設定する。こうして、図203(a)同様、出力ポート2イメージの下位2ビットの値に応じて、メイン表示器データバッファ2のビット7を設定することが可能となる。

【1700】

ここで、図203(a)に示したDYNMOUTモジュールのコマンドの総コマンドサイズは、2行目のコマンド「RESQ @CHN__DSP__BIT, (LOW __WIN__DAT+1)」3バイト+3行目のコマンド「LDQ A, (LOW __OUT__PT2)」2バイト+4行目のコマンド「JAND Z, A, 00000011B, DYNMOUT02」3バイト+5行目のコマンド「SETQ @CHN__DSP__BIT, (LOW __WIN__DAT+1)」3バイト=11バイトとなり、総実行サイクルは、出力ポート2イメージの下位2ビットが0でない場合、2行目6サイクル+3行目3サイクル+4行目3サイクル+5行目6サイクル=18サイクルとなり、出力ポート2イメージの下位2ビットがいずれも0であった場合、2行目6サイクル+3行目3サイクル+4行目4サイクル=13サイクルとなる。

【1701】

一方、図203(b)に示したDYNMOUTモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LDQ HL, LOW __WIN__DAT+1」2バイト+3行目のコマンド「RES @CHN__DSP__BIT, (HL)」2バイト+4行目のコマンド「LDQ A, (LOW __OUT__PT2)」2バイト+5行目のコマンド「AND A, 00000011B」2バイト+6行目のコマンド「SET NZ, @CHN__DSP__BIT, (HL)」2バイト=10バイトとなり、総実行サイクルは、出力ポート2イメージの下位2ビットが0でない場合、2行目2サイクル+3行目5サイクル+4行目3サイクル+5行目2サイクル+6行目5サイクル=17サイクルとなり、出力ポート2イメージの下位2ビットがいずれも0であった場合、2行目2サイクル+3行目5サイクル+4行目3サイクル+5行目2サイクル+6行目2サイクル=14サイクルとなる。したがって、図203(a)の場合と比べ、総コマンドサイズが1バイト削減されていることが理解できる。かかるDYNMOUTモジュールによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1702】

図204は、EXT__PRCモジュールを説明するための説明図である。図26のステップS400-29で示したEXT__PRCモジュールでは、メインCPU500aが、外部情報管理処理、すなわち、遊技情報出力端子板312から外部へ出力する外部情報用

の出力データをセットする。

【1703】

図204(a)の1行目の指標「EXT_PRC:」は、当該EXT_PRCモジュールの先頭アドレスを示す。2行目のコマンド「LD BC, @D_SEC_LOOP*256+0」によって、Bレジスタにチェック対象ラム数(ループ回数)である@D_SEC_LOOP(ここでは8)が設定され、Cレジスタに合成ビットデータの初期値として0が設定される。3行目のコマンド「JR Z, EXT_PRC_15」によって、そのコマンドを実行する以前に演算された対象ラムアドレスの論理積がゼロであるか否かが判定され、ゼロであれば(ゼロフラグが立っていれば)、5行目の指標「EXT_PRC_15:」に移動し、ゼロでなければ(ゼロフラグが立っていなければ)、4行目の処理を行う。4行目のコマンド「SET @EXT_SEC_POS, C」によって、Cレジスタの値(合成ビットデータ)の@EXT_SEC_POSで定義される値(ここでは、ビット6)のビットを立たせる。5行目の指標「EXT_PRC_15:」は3行目のコマンド「JR Z, EXT_PRC_15」の移動先を示す。こうして、対象ラムアドレスの論理積の演算結果に応じて合成ビットデータのビット6を設定することが可能となる。

【1704】

ここで、コマンド「SET」への置き換えを行うと、図204(a)のコマンド群を図204(b)のように変更することができる。図204(b)の1行目の指標「EXT_PRC:」は、当該EXT_PRCモジュールの先頭アドレスを示す。2行目のコマンド「LD BC, @D_SEC_LOOP*256+0」によって、Bレジスタにチェック対象ラム数(ループ回数)である@D_SEC_LOOP(ここでは8)が設定され、Cレジスタに合成ビットデータの初期値として0が設定される。3行目のコマンド「SET NZ, @EXT_SEC_POS, C」によって、そのコマンドが実行される以前に演算された対象ラムアドレスの論理積がゼロであるか否かが判定され、ゼロでなければ(ゼロフラグが立っていなければ)、Cレジスタの値(合成ビットデータ)の@EXT_SEC_POSで定義される値(ここでは、ビット6)のビットを立たせる。こうして、図204(a)同様、対象ラムアドレスの論理積の演算結果に応じて合成ビットデータのビット6を設定することが可能となる。

【1705】

ここで、図204(a)に示したEXT_PRCモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD BC, @D_SEC_LOOP*256+0」2バイト+3行目のコマンド「JR Z, EXT_PRC_15」2バイト+4行目のコマンド「SET @EXT_SEC_POS, C」2バイト=6バイトとなり、総実行サイクルは、論理積が0でない場合、2行目2サイクル+3行目2サイクル+4行目2サイクル=6サイクルとなり、論理積が0であった場合、2行目2サイクル+3行目3サイクル=5サイクルとなる。

【1706】

一方、図204(b)に示したEXT_PRCモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LD BC, @D_SEC_LOOP*256+0」2バイト+3行目のコマンド「SET NZ, @EXT_SEC_POS, C」2バイト=4バイトとなり、総実行サイクルは、論理積が0でない場合、2行目2サイクル+3行目5サイクル=7サイクルとなり、論理積が0であった場合、2行目2サイクル+3行目2サイクル=4サイクルとなる。したがって、図204(a)の場合と比べ、総コマンドサイズが2バイト削減されていることが理解できる。かかるEXT_PRCモジュールによって、コマンドの更なる短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1707】

〈コマンド「JANDQ」〉

図205は、STOPDCTモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図92のステップS2500で示した回胴回転中処理を

10

20

30

40

50

実行するSTOPDCTモジュールの一部である停止表示器を確認する処理S2500-23を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該STOPDCTモジュールの説明中、所定のレジスタはHLレジスタである。

【1708】

メインCPU500aは、図205のように、任意の実行する。メインCPU500aは、まず、操作対象ビットの抽出処理を行う(S1)。かかる処理により、Aレジスタには、操作対象ビットが設定される。そして、メインCPU500aは、Aレジスタの操作対象ビットと、予めDレジスタに設定されている回動回転中フラグとの合成を行う(S2)。続いて、メインCPU500aは、停止表示器を示すアドレスの値と論理積をとることで停止表示器を確認し(S3)、停止表示器が赤色点灯中であれば(S4におけるYES)、当該STOPDCTモジュールの先頭から処理を繰り返し、停止表示器が赤色点灯中でなければ(S4におけるNO)、次の処理を実行する。そして、回転停止処理を実行するSTP_RELモジュールに移動することで、当該STOPDCTモジュールを終了する(S5)。こうして、停止表示器の状態に応じた押下処理が可能となる。

【1709】

図206は、STOPDCTモジュールを実現するためのコマンドの一例を説明するための説明図である。図205で示したフローチャートは、例えば、図206に示したプログラムによって実現される。

【1710】

図206の1行目の指標「STOPDCT:」は、当該STOPDCTモジュールの先頭アドレスを示す。2行目のコマンド「CALLF SMPLBIT」は、操作対象ビット抽出処理を実行するSMPLBITモジュールを呼び出すコマンドであり、このコマンドによって操作対象となった(実際に操作された)ストップスイッチに対応するビットがAレジスタに設定される。かかる2行目のコマンドが、図205のステップS1に対応する。3行目のコマンド「AND A, D」によって、操作対象ビットが設定されたAレジスタと、予め回動回転中フラグが設定されているDレジスタとの論理積を演算し、その結果をAレジスタに保持する。かかる3行目のコマンドが、図205のステップS2に対応する。

【1711】

図206の4行目のコマンド「ANDQ A, (LOW __OUT__PT4)」によって、Qレジスタの値を上位1バイトとし、アドレス「__OUT__PT4」の値(第1所定値)を下位1バイトとした値(停止表示器の値)とAレジスタの値の論理積を演算し、その結果をAレジスタに保持する。かかる4行目のコマンドが、図205のステップS3に対応する。5行目のコマンド「JR Z, STOPDCT」によって、4行目のコマンド「ANDQ A, (LOW __OUT__PT4)」の演算結果がゼロであるか否か判定され、ゼロであれば(ゼロフラグが立っていれば)、1行目の指標「STOPDCT:」(第2所定値)に移動し、ゼロでなければ(ゼロフラグが立っていなければ)、次の処理を行う。かかる5行目のコマンドが、図205のステップS4に対応する。

【1712】

このように、図206に示したSTOPDCTモジュールのコマンドの総コマンドサイズは、2行目のコマンド「CALLF SMPLBIT」2バイト+3行目のコマンド「AND A, D」1バイト+4行目のコマンド「ANDQ A, (LOW __OUT__PT4)」3バイト+5行目のコマンド「JR Z, STOPDCT」2バイト=8バイトとなり、総実行サイクルは、2行目4サイクル+3行目1サイクル+4行目4サイクル+5行目3サイクル(2サイクル)=12サイクル(11サイクル)となる。なお、括弧内のサイクル数は、コマンド「JR Z, STOPDCT」によって移動しなかった場合の実行サイクルを示している。かかるSTOPDCTモジュールにおいて4、5行目の処理を設けることによって、停止表示器の状態に応じた押下処理が可能となる。

【1713】

図207は、STOPDCTモジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図206のSTOPDCTモジュールのコマンド群を、図207のSTOPDCTモジュールのコマンド群に置き換えている。

【1714】

図207の1行目の指標「STOPDCT:」は、当該STOPDCTモジュールの先頭アドレスを示す。2行目のコマンド「CALLF SMPLBIT」は、操作対象ビット抽出処理を実行するSMPLBITモジュールを呼び出すコマンドであり、このコマンドによって操作対象となったストップスイッチに対応するビットがAレジスタに設定される。かかる2行目のコマンドが、図205のステップS1に対応する。3行目のコマンド「AND A, D」によって、操作対象ビットが設定されたAレジスタと、予め回動回転中フラグが設定されているDレジスタとの論理積を演算し、その結果をAレジスタに保持する。かかる3行目のコマンドが、図205のステップS2に対応する。

10

【1715】

図207の4行目のコマンド「JANDQ Z, A, (LOW __OUT__PT4), STOPDCT」によって、Qレジスタの値を上位1バイトとし、アドレス「__OUT__PT4」の値（第1所定値）を下位1バイトとした値（停止表示器の値）とAレジスタの値の論理積を演算し、その演算結果がゼロであるか否か判定され、ゼロであれば（ゼロフラグが立っていれば）、1行目の指標「STOPDCT:」（第2所定値）に移動し、ゼロでなければ（ゼロフラグが立っていなければ）、次の処理を行う。かかる4行目のコマンドが、図205のステップS3、S4に対応する。

20

【1716】

ここで、コマンド「JANDQ cc, A, (k), e」は、Qレジスタの値を上位1バイトとし、kで示されるアドレスの値を下位1バイトとした値（第1所定値）と、Aレジスタの値の論理積を演算し、その演算結果について、ccに対応するフラグ（ゼロフラグやキャリーフラグ）が真であれば、e（第2所定値）で示されるアドレスに移動し、ccに対応するフラグが真でなければ、次の処理に移るコマンドである。かかるコマンドのコマンドサイズは「4」であり、実行サイクルは「6」（「5」）である。なお、括弧内のサイクル数は、コマンド「JANDQ cc, A, (k), e」においてフラグが真ではなかった場合の実行サイクルを示している。

【1717】

図207に示したSTOPDCTモジュールのコマンドの総コマンドサイズは、2行目のコマンド「CALLF SMPLBIT」2バイト+3行目のコマンド「AND A, D」1バイト+4行目のコマンド「JANDQ Z, A, (LOW __OUT__PT4), STOPDCT」4バイト=7バイトとなり、総実行サイクルは、2行目4サイクル+3行目1サイクル+4行目6サイクル（5サイクル）=11サイクル（10サイクル）となる。なお、括弧内のサイクル数は、コマンド「JANDQ Z, A, (LOW __OUT__PT4), STOPDCT」によって移動しなかった場合の実行サイクルを示している。したがって、図206の場合と比べ、総コマンドサイズが1バイト削減され、総実行サイクルが少なくとも1サイクル削減される。かかるSTOPDCTモジュールによって、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

30

40

【1718】

また、図206と図207とを比較して理解できるように、図206において2行（4行目、5行目）を占有していたコマンド群を、図207においては1行（4行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1719】

〈コマンド「CALLEX」〉

E__SETTMモジュールは、図89のステップS2200-11や図95のステップS2800-39で示したエラーウェイト処理を実行するERRWAITモジュールにおいて、投入異常非監視タイマ設定処理、すなわち、投入異常非監視タイマを設定するため

50

のモジュールである。なお、E__SETTMモジュールは、メインROM500bの別領域（2000H～3FBFH）に配される。

【1720】

メインCPU500aは、メインROM500bから使用領域のプログラムを読み出し、読み出したプログラムを遂行し、任意の処理において、サブルーチンとして別領域のE__SETTMモジュールを呼び出し、E__SETTMモジュールを遂行する。E__SETTMモジュールでは、投入異常非監視タイマを設定する。

【1721】

図208は、E__SETTMモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図89のステップS2200-11や図95のステップS2800-39で示したエラーウェイト処理を実行するERRWAITモジュールにおいて、ブロッカー閉塞処理を実行するBLKSHUTモジュールの一部を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1722】

メインCPU500aは、図208（a）のように、任意の処理を実行する。メインCPU500aは、ブロッカー状態を確認して、出力ポート4イメージのビット3（ブロッカーソレノイド出力ビット番号）が0ではない場合に、別領域のE__SETTMモジュールを呼び出す。このため、メインCPU500aは、まず、割込を禁止し（S1）、サブルーチンとして別領域のE__SETTMモジュールを呼び出す（S2）。

【1723】

メインCPU500aは、図208（b）のように、E__SETTMモジュールにおいて、汎用レジスタ（Aレジスタ、Bレジスタ、Cレジスタ、Dレジスタ、Eレジスタ、Hレジスタ、Lレジスタ、Fレジスタ）を退避する（S3）。そして、メインCPU500aは、初期値としての投入異常非監視タイマ値をAレジスタに設定し（S4）、そのAレジスタの値を所定のアドレス（投入異常非監視タイマ）が示す値に格納して、投入異常非監視タイマを設定する（S5）。かかる処理が完了すると、メインCPU500aは、汎用レジスタを復帰させ（S6）、割込を許可し（S7）、当該E__SETTMモジュールを終了して1段上のルーチンに戻る（S8）。

【1724】

ここで、E__SETTMモジュールは、遊技機の公正さを担保しつつ、使用領域と別領域とで適切に役割分担するための、上述した（1）～（6）の条件を全て満たすように設計されている。

【1725】

図209は、E__SETTMモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図209（a）は、E__SETTMモジュールを呼び出す任意の処理のコマンド群を示し、図209（b）は、E__SETTMモジュールのコマンド群を示す。図208で示したフローチャートは、例えば、図209に示したプログラムによって実現される。

【1726】

図209（a）の1行目のコマンド「DI」によって、割込が禁止される。かかる1行目のコマンドが、図208（a）のステップS1に対応する。2行目のコマンド「CALL E__SETTM」によって、サブルーチンとしてE__SETTMモジュールが呼び出される。かかる2行目のコマンドが、図208（a）のステップS2に対応する。

【1727】

図209（b）の1行目の指標「E__SETTM:」は、当該E__SETTMモジュールの先頭アドレスを示す。2行目のコマンド「EX AF, AF'」によって、ペアレジスタAFの値と裏レジスタであるペアレジスタAF'の値とが入れ替わり、ペアレジスタAFの値が退避される。3行目のコマンド「EXX」によって、ペアレジスタBC、DE、HLの値と裏レジスタであるペアレジスタBC'、DE'、HL'の値とが入れ替わり、ペアレジスタBC、DE、HLの値が退避される。かかる2、3行目のコマンドが、図208（

10

20

30

40

50

b) のステップ S 3 に対応する。

【1728】

図209 (b) の4行目のコマンド「LD A, @MDL_WAT_TMR」によって、固定値「@MDL_WAT_TMR」、すなわち、投入異常非監視タイマ値（初期値）としての250.32msecに相当する(252/6+1)がAレジスタに読み出される。かかる4行目のコマンドが、図208 (b) のステップ S 4 に対応する。5行目のコマンド「LD (__EX_SLT M), A」によって、投入異常非監視タイマとしての変数「__EX_SLT M」にAレジスタの値を格納する。かかる5行目のコマンドが、図208 (b) のステップ S 5 に対応する。こうして、投入異常非監視タイマに投入異常非監視タイマ値を設定することが可能となる。

10

【1729】

図209 (b) の6行目のコマンド「EXX」によって、ペアレジスタBC、DE、HLの値と裏レジスタであるペアレジスタBC'、DE'、HL'の値とが入れ替わり、ペアレジスタBC、DE、HLの値が復帰する。7行目のコマンド「EX AF, AF'」によって、ペアレジスタAFの値と裏レジスタであるペアレジスタAF'の値とが入れ替わり、ペアレジスタAFの値が復帰する。かかる6、7行目のコマンドが、図208 (b) のステップ S 6 に対応する。図209 (b) の8行目のコマンド「EI」によって、割込が許可される。なお、ここでは、サブルーチンから戻る前にコマンド「EI」を行っても適切に割込が許可される。したがって、サブルーチン内にコマンド「EI」を記述することで、使用領域の容量を確保することが可能となる。かかる8行目のコマンドが、図208 (b) のステップ S 7 に対応する。そして、9行目のコマンド「RET」によって、1段上のルーチンに戻る。かかる9行目のコマンドが、図208 (b) のステップ S 8 に対応する。こうして、別領域において投入異常非監視タイマを設定することが可能となる。

20

【1730】

このように、図209 (a) に示したBLKSHUTモジュールと図209 (b) に示したE_SETTMモジュールの総コマンドサイズは、図209 (a) の1行目のコマンド「DI」1バイト+2行目のコマンド「CALL E_SETTM」3バイト+図209 (b) の2行目のコマンド「EX AF, AF'」1バイト+3行目のコマンド「EXX」1バイト+4行目のコマンド「LD A, @MDL_WAT_TMR」2バイト+5行目のコマンド「LD (__EX_SLT M), A」4バイト+6行目のコマンド「EXX」1バイト+7行目のコマンド「EX AF, AF'」1バイト+8行目のコマンド「EI」1バイト+9行目のコマンド「RET」1バイト=16バイトであり、総実行サイクルは、図209 (a) の1行目1サイクル+2行目5サイクル+図209 (b) の2行目1サイクル+3行目1サイクル+4行目2サイクル+5行目4サイクル+6行目1サイクル+7行目1サイクル+8行目1サイクル+9行目3サイクル=20サイクルとなる。かかるE_SETTMモジュールを設けることによって、上述した各モジュール内で投入異常非監視タイマを設定することなく、コマンドサイズ3バイトのコマンド「CALL E_SETTM」で賄うことができるので、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

30

【1731】

図210は、E_SETTMモジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図209 (a) のE_SETTMモジュールを呼び出す任意の処理のコマンド群を、図210 (a) のE_SETTMモジュールを呼び出す任意の処理のコマンド群に置き換え、図209 (b) のE_SETTMモジュールのコマンド群を、図210 (b) のE_SETTMモジュールのコマンド群に置き換えている。

40

【1732】

図210 (a) の1行目のコマンド「CALL EX E_SETTM」によって、サブルーチンとしてE_SETTMモジュールが呼び出される。かかる1行目のコマンドが、図208 (a) のステップ S 2 に対応する。

【1733】

50

ここで、コマンド「`CALLEX mn`」は、コマンド「`CALL mn`」同様、固定値 `mn` で示される先頭アドレスを有するサブルーチンを呼び出すコマンドである。ただし、コマンド「`CALLEX mn`」は、コマンド「`CALL mn`」のように、単にサブルーチンを呼び出すだけでなく、複数の機能を有している。例えば、当該コマンドを実行することで、自動的に、ノンマスカブル割込（`NMI`）およびマスカブル割込（`INT`）を禁止し、レジスタのバンクを第1レジスタバンク726から第2レジスタバンク728に切り換えてから固定値 `mn` で示される先頭アドレスを有するサブルーチンを呼び出す。したがって、割込に関するコマンド「`DI`」や、退避に関するコマンド「`EX AF, AF'`」、「`EXX`」（または、コマンド「`PUSH`」）を要さない。したがって、図210（a）の1行目のコマンド「`CALLEX E__SETTM`」は、図208（a）のステップS2のみならず、図208（a）のステップS1および図208（b）のステップS3にも対応する。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「4」である。

【1734】

ただし、コマンド「`CALLEX mn`」には、所定の制限がある。例えば、呼び出し先アドレスがアドレスマップ上の2000H～20FFHの範囲であれば、コマンドのコマンドサイズは「2」であり、実行サイクルは「4」であるが、それ以外の範囲では、コマンドのコマンドサイズは「4」であり、実行サイクルは「6」となる。したがって、コマンドの短縮化を図るため、本実施形態では、呼び出し先アドレス、例えば、「`E__SETTM`」が2000H～20FFH内に含まれるように配置する。

【1735】

図210（b）の1行目の指標「`E__SETTM:`」は、当該`E__SETTM`モジュールの先頭アドレスを示す。2行目のコマンド「`LD A, @MDL__WAT__TMR`」によって、固定値「`@MDL__WAT__TMR`」、すなわち、投入異常非監視タイマ値（初期値）としての250.32msecに相当する（ $252/6+1$ ）がAレジスタに読み出される。かかる2行目のコマンドが、図208（b）のステップS4に対応する。3行目のコマンド「`LD (__EX__SLTM), A`」によって、投入異常非監視タイマとしての変数「`__EX__SLTM`」にAレジスタの値を格納する。かかる3行目のコマンドが、図208（b）のステップS5に対応する。こうして、投入異常非監視タイマに投入異常非監視タイマ値を設定することが可能となる。

【1736】

図210（b）の4行目のコマンド「`RETEX`」によって、1段上のルーチンに戻る。かかる4行目のコマンドが、図208（b）のステップS8に対応する。こうして、別領域において投入異常非監視タイマを設定することが可能となる。

【1737】

ここで、コマンド「`RETEX`」は、コマンド「`RET`」同様、1段上のルーチンに戻るコマンドである。ただし、コマンド「`RETEX`」は、コマンド「`CALLEX mn`」とペアで用いられることが多く、コマンド「`RET`」のように、単にサブルーチンから戻るだけでなく、複数の機能を有している。例えば、当該コマンドを実行することで、自動的に、レジスタのバンクを第2レジスタバンク728から第1レジスタバンク726に切り換え、割込を許可してから1段上のルーチンに戻る。したがって、復帰に関するコマンド「`EXX`」、「`EX AF, AF'`」（または、コマンド「`POP`」）や、割込に関するコマンド「`EI`」を要さない。したがって、図210（b）の4行目のコマンド「`RETEX`」は、図208（b）のステップS8のみならず、図208（b）のステップS6およびステップS7にも対応する。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「5」である。

【1738】

図210（a）に示した`BLKSHUT`モジュールと図210（b）に示した`E__SETTM`モジュールの総コマンドサイズは、図210（a）の1行目のコマンド「`CALL EX E__SETTM`」2バイト+図210（b）の2行目のコマンド「`LD A, @MD`

10

20

30

40

50

L_WAT_TMR」2バイト+3行目のコマンド「LD (__EX_SLTM), A」4バイト+4行目のコマンド「RETEX」2バイト=10バイトであり、総実行サイクルは、図210(a)の1行目4サイクル+図210(b)の2行目2サイクル+3行目4サイクル+4行目5サイクル=15サイクルとなる。したがって、図209の場合と比べ、総コマンドサイズが6バイト削減され、総実行サイクルも少なくとも5サイクル削減される。かかるE_SETTMモジュールによって、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1739】

また、図209と図210とを比較して理解できるように、図209において8行(図209(a)の1、2行目および図209(b)の2、3、6~9行目)を占有していたコマンド群を、図210においては、2行(図210(a)の1行目および図210(b)の4行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1740】

かかるコマンド「CALLEX」は、上述したBLKSHUTモジュールのみならず、様々なモジュールに採用することができる。また、スロットマシンのみならず、パチンコ機における、例えば、DYM_OUTモジュール等にも採用することができる。

【1741】

図211、図212は、DYM_OUTモジュールを説明するための説明図である。図26のステップS400-5で示したDYM_OUTモジュールは、第1特別図柄表示器160、第2特別図柄表示器162、第1特別図柄保留表示器164、第2特別図柄保留表示器166、普通図柄表示器168、普通図柄保留表示器170、右打ち報知表示器172、性能表示モニタ184を点灯制御するダイナミックポート出力処理を実行する。

【1742】

図211(a)の1行目のコマンド「DI」によって、割込が禁止される。2行目のコマンド「CALL E_DMOUT」によって、サブルーチンとしてE_DMOUTモジュールが呼び出される。

【1743】

図211(b)の1行目の指標「E_DMOUT:」は、当該E_DMOUTモジュールの先頭アドレスを示す。2行目のコマンド「EX AF, AF'」によって、ペアレジスタAFの値と裏レジスタであるペアレジスタAF'の値とが入れ替わり、ペアレジスタAFの値が退避される。3行目のコマンド「EXX」によって、ペアレジスタBC、DE、HLの値と裏レジスタであるペアレジスタBC'、DE'、HL'の値とが入れ替わり、ペアレジスタBC、DE、HLの値が退避される。4行目のコマンド「LD HL, R_ERW_IOB」によって識別セグ出力要求バッファのアドレス「R_ERW_IOB」をHLレジスタに設定する。5行目のコマンド「LD A, (R_COM_CNT)」によって、「R_COM_CNT」で示されるアドレスに格納された1バイト値(コモンカウンタ)がAレジスタに読み出される。ここで、コモンカウンタはLEDダイナミック点灯制御のコモン番号を判定するために使用され、コモン番号を示す値が格納されている。6行目のコマンド「LD A, (HL+A)」によって、HLレジスタで示されるアドレスにAレジスタの値を加算した(オフセットさせた)アドレスに格納された1バイト値(表示データ)がAレジスタに読み出される。7行目のコマンド「OUT (@OTC_PRT), A」によって、Uレジスタの値をアドレスの上位1バイトとし、出力ポート12(内部レジスタI/Oポート)のアドレスの下位1バイトを示す固定値「@OTC_PRT」を下位1バイトとし、そのアドレスに、Aレジスタの値を出力する。8行目のコマンド「EXX」によって、ペアレジスタBC、DE、HLの値と裏レジスタであるペアレジスタBC'、DE'、HL'の値とが入れ替わり、ペアレジスタBC、DE、HLの値が復帰する。9行目のコマンド「EX AF, AF'」によって、ペアレジスタAFの値と裏レジスタであるペアレジスタAF'の値とが入れ替わり、ペアレジスタAFの値が復帰する。10行目の

10

20

30

40

50

コマンド「EI」によって、割込が許可される。なお、ここでは、サブルーチンから戻る前にコマンド「EI」を行っても適切に割込が許可される。したがって、サブルーチン内にコマンド「EI」を記述することで、使用領域の容量を確保することが可能となる。そして、11行目のコマンド「RET」によって、1段上のルーチンに戻る。こうして、別領域においてダイナミックポート出力処理を実行することができる。

【1744】

ここで、コマンド「CALLEX」への置き換えを行うと、図211のコマンド群を図212のように変更することができる。図212(a)の1行目のコマンド「CALLEX E_DMOUT」によって、サブルーチンとしてE_DMOUTモジュールが呼び出される。ここでは、呼び出し先アドレス、例えば、「E_DMOUT」が2000H~20FFH内に含まれるように配置されている。

10

【1745】

図212(b)の1行目の指標「E_DMOUT:」は、当該E_DMOUTモジュールの先頭アドレスを示す。2行目のコマンド「LD HL, R_ERW_IOB」によって識別セグ出力要求バッファのアドレス「R_ERW_IOB」をHLレジスタに設定する。3行目のコマンド「LD A, (R_COM_CNT)」によって、「R_COM_CNT」で示されるアドレスに格納された1バイト値(コモンカウンタ)がAレジスタに読み出される。4行目のコマンド「LD A, (HL+A)」によって、HLレジスタで示されるアドレスにAレジスタの値を加算した(オフセットさせた)アドレスに格納された1バイト値(表示データ)がAレジスタに読み出される。5行目のコマンド「OUT (@OTC_PRT), A」によって、Uレジスタの値をアドレスの上位1バイトとし、出力ポート12(内部レジスタI/Oポート)のアドレスの下位1バイトを示す固定値「@OTC_PRT」を下位1バイトとし、そのアドレスに、Aレジスタの値を出力する。6行目のコマンド「RETEX」によって、1段上のルーチンに戻る。こうして、図211同様、別領域においてダイナミックポート出力処理を実行することができる。

20

【1746】

このように、図211(a)に示したDYM_OUTモジュールと図211(b)に示したE_DMOUTモジュールの総コマンドサイズは、図211(a)の1行目のコマンド「DI」1バイト+2行目のコマンド「CALL E_DMOUT」3バイト+図211(b)の2行目のコマンド「EX AF, AF'」1バイト+3行目のコマンド「EXX」1バイト+4行目のコマンド「LD HL, R_ERW_IOB」3バイト+5行目のコマンド「LD A, (R_COM_CNT)」3バイト+6行目のコマンド「LD A, (HL+A)」3バイト+7行目のコマンド「OUT (@OTC_PRT), A」2バイト+8行目のコマンド「EXX」1バイト+9行目のコマンド「EX AF, AF'」1バイト+10行目のコマンド「EI」1バイト+11行目のコマンド「RET」1バイト=21バイトであり、総実行サイクルは、図211(a)の1行目1サイクル+2行目5サイクル+図211(b)の2行目1サイクル+3行目1サイクル+4行目3サイクル+5行目4サイクル+6行目4サイクル+7行目3サイクル+8行目1サイクル+9行目1サイクル+10行目1サイクル+11行目3サイクル=28サイクルとなる。

30

【1747】

一方、図212(a)に示したDYM_OUTモジュールと図212(b)に示したE_DMOUTモジュールの総コマンドサイズは、図212(a)の1行目のコマンド「CALLEX E_DMOUT」2バイト+図212(b)の2行目のコマンド「LD HL, R_ERW_IOB」3バイト+3行目のコマンド「LD A, (R_COM_CNT)」3バイト+4行目のコマンド「LD A, (HL+A)」3バイト+5行目のコマンド「OUT (@OTC_PRT), A」2バイト+6行目のコマンド「RETEX」2バイト=15バイトであり、総実行サイクルは、図212(a)の1行目4サイクル+図212(b)の2行目3サイクル+3行目4サイクル+4行目4サイクル+5行目3サイクル+6行目5サイクル=23サイクルとなる。したがって、図211の場合と比べ、総コマンドサイズが6バイト削減され、総実行サイクルも5サイクル削減される。かかるE_

40

50

D M O U Tモジュールによって、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1748】

ところで、図208～図210を用いて説明したように、コマンド「C A L L E X m n」は、呼び出し先アドレスがアドレスマップ上の2000H～20FFHの範囲であれば、コマンドのコマンドサイズは「2」であり、実行サイクルは「4」であるが、それ以外の範囲では、コマンドのコマンドサイズは「4」であり、実行サイクルは「6」となる。したがって、呼び出し先アドレスは全て2000H～20FFHの範囲に配置するのが望ましい。しかし、2000H～20FFHでは、記述できるバイト数が256バイトしかないので、かかる範囲にサブルーチンをそのまま記述すると、サブルーチンの数が制限されてしまう。例えば、比較的コマンドサイズが小さいE__S E T T Mモジュールであっても、図210(b)の2行目のコマンド「L D A, @M D L _W A T _T M R」2バイト+3行目のコマンド「L D (_E X _S L T M), A」4バイト+4行目のコマンド「R E T E X」2バイト=8バイトを占有するので、サブルーチンの数を多く配置することができなくなる。

10

【1749】

そこで、本実施形態では、アドレスマップ上の2000H～20FFHの範囲には、実質的に、移動に関するコマンド「J P m n」のみを配置し、サブルーチンの本体は移動先に配置する。かかるコマンド「J P m n」のコマンドサイズは「3」なので、アドレスマップ上の2000H～20FFHの範囲にコマンド「J P m n」を並置することで、 $256 / 3 = 86$ 個のモジュールをサブルーチンとして呼び出すことが可能となる。

20

【1750】

なお、コマンド「J P m n」の代わりにコマンドサイズ「3」のコマンド「C A L L m n」を用いることも可能であるが、その場合、コマンド「C A L L m n」の後に、コマンドサイズ「2」のコマンド「R E T E X」を配置する必要が生じるので、コマンド「J P m n」よりサブルーチンの数が制限されることとなる。したがって、コマンド「J P m n」を採用するのが望ましい。以下、かかる内容を反映したE__S E T T Mモジュールの具体的な処理を説明する。

【1751】

図213は、E__S E T T Mモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図208同様、ブロッカー閉塞処理を実行するB L K S H U Tモジュールの一部を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

30

【1752】

メインCPU500aは、図213(a)のように、任意の処理を実行する。メインCPU500aは、別領域のサブルーチンとしてR R E 0 1モジュールを呼び出す(S1)。

【1753】

続いて、メインCPU500aは、図213(b)のように、R R E 0 1モジュールにおいて、有意な処理を行うことなく、E__S E T T Mモジュールに移動する(S2)。このように、R R E 0 1モジュールは、E__S E T T Mモジュールに移動するためのモジュールであり、アドレスマップ上の2000H～20FFHの範囲に配置される。こうして、サブルーチンの数を多く配置することができる。

40

【1754】

メインCPU500aは、図213(c)のように、E__S E T T Mモジュールにおいて、投入異常非監視タイマ値をAレジスタに設定し(S3)、そのAレジスタの値を所定のアドレス(投入異常非監視タイマ)が示す値に格納して、投入異常非監視タイマを設定する(S4)。かかる処理が完了すると、メインCPU500aは、当該E__S E T T Mモジュールを終了して1段上のルーチンに戻る(S5)。かかるE__S E T T Mモジュールは、R R E 0 1モジュールの移動先なので、アドレスマップの制限を受けない。したがって、別領域のうちの2100H～3FBFHの範囲に任意に配置することができるので

50

、総コマンドサイズがある程度大きくとも許容される。

【1755】

図214は、E__SETTMモジュールを実現するためのコマンドのさらに他の例を説明するための説明図である。ここでは、図214(b)のように、新たに、E__SETTMモジュールに移動するためのRRE01モジュールを設けている。

【1756】

図214(a)の1行目のコマンド「CALLEX PRE01」によって、サブルーチンとしてE__SETTMモジュールではなく、RRE01モジュールが呼び出される。かかる1行目のコマンドが、図213(a)のステップS1に対応する。なお、コマンド「CALLEX PRE01」によって、割込の禁止、および、汎用レジスタの退避も実行される。

10

【1757】

図214(b)の1行目の指標「PRE01:」は、当該PRE01モジュールの先頭アドレスを示す。2行目のコマンド「JP E__SETTM」によって、E__SETTMモジュール本体へ移動する。かかるコマンドが、図213(b)のステップS2に対応する。

【1758】

なお、コマンド「CALLEX mn」を通じて複数のサブルーチンを呼び出す場合、その複数のサブルーチンに関する呼び出しアドレスを示す指標（例えば、「PRE01:」と、コマンド「JP mn」とを、2000H~20FFHの範囲に連続して並置する。こうすることで、2000H~20FFHの範囲を有効利用し、サブルーチンの数を多く配置することができる。

20

【1759】

図214(c)の1行目の指標「E__SETTM:」は、当該E__SETTMモジュールの先頭アドレスを示す。2行目のコマンド「LD A,@MDL__WAT__TMR」によって、固定値「@MDL__WAT__TMR」、すなわち、投入異常非監視タイマ値（初期値）としての250.32msecに相当する(252/6+1)がAレジスタに読み出される。かかる2行目のコマンドが、図213(c)のステップS3に対応する。3行目のコマンド「LD (__EX__SLTM),A」によって、投入異常非監視タイマとしての変数「__EX__SLTM」にAレジスタの値を格納する。かかる3行目のコマンドが、図213(c)のステップS4に対応する。こうして、投入異常非監視タイマに投入異常非監視タイマ値を設定することが可能となる。

30

【1760】

図214(c)の4行目のコマンド「RETEX」によって、1段上のルーチンに戻る。かかる4行目のコマンドが、図213(c)のステップS5に対応する。こうして、別領域において投入異常非監視タイマを設定することが可能となる。なお、コマンド「RETEX」によって、汎用レジスタの復帰、および、割込の許可も実行される。

【1761】

かかる図214の例では、図210の場合と比べ、コマンド「JP E__SETTM」の分だけ、総コマンドサイズおよび総実行サイクルが大きくなる。しかし、コマンド「JP E__SETTM」を経由してサブルーチン本体に移動させる構成とすることで、2000H~20FFHの範囲にサブルーチンを多く配することができ、サブルーチン数の制限を広げ、さらに、サブルーチン自体のコマンドサイズの制限が緩和されるので、結果、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

40

【1762】

なお、このように、2000H~20FFHの範囲にコマンド「JP mn」のみを配置しても、配置できるサブルーチンの数が足りない場合、すなわち、サブルーチンが86個以上ある場合、総コマンドサイズが小さいモジュールを2100H~217FHの範囲(128バイトの範囲)に配置し、コマンドサイズ「3」のコマンド「JP mn」では

50

なく、近距離の移動に適しているコマンドサイズ「2」のコマンド「JR mn」を用いるとしてもよい。こうすることで、2000H～20FFHの範囲に、さらに多くのコマンド「JR mn」を配置することが可能となる。

【1763】

〈コマンド「ICPLMQA」、「ICPLMQ」〉

RAM__INCモジュールは、RAM加算処理、すなわち、メインRAM500cの1バイト変数を、所定数を上限に、1だけインクリメントするためのモジュールである。なお、RAM__INCモジュールは、インクリメントのみならず、インクリメントした結果、キャリーフラグを設定している。また、RAM__INCモジュールは、汎用モジュールとして利用することもできる。

【1764】

かかるRAM__INCモジュールは、複数のモジュールからサブルーチンとして呼び出される。例えば、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、本前兆中処理(AT状態=「3」)を実行するHID__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、終了画面中処理(AT状態=「4」)を実行するFIN__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、REG中処理(AT状態=「6」)を実行するREG__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、BIG中処理(AT状態=「7」)を実行するBIT__LOTモジュール等から呼び出される。

【1765】

メインCPU500aは、メインROM500bからプログラムを読み出し、読み出したプログラムを遂行し、任意の処理において、メインRAM500cに保持された1バイト値をインクリメントする。

【1766】

図215は、RAM__INCモジュールの具体的な処理を示したフローチャートである。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該RAM__INCモジュールの説明中、所定レジスタはAレジスタであり、所定数は上限値であり、特定アドレスはインクリメントの対象となるアドレスである。

【1767】

メインCPU500aは、図215(a)のように、任意の処理において、インクリメント対象となるアドレスをHLレジスタに設定する(S1)。そして、メインCPU500aは、HLレジスタで示されるアドレスに格納された1バイト値をインクリメントし(S2)、インクリメントした値をHLアドレスで示されるアドレスに格納して、値を更新する(S3)。ただし、ここでは、上限値(所定数)が設定され、インクリメントした結果が、その上限値を超えないようになっている。

【1768】

図216、図217は、RAM__INCモジュールを実現するためのコマンドの一例を説明するための説明図である。図215で示したフローチャートは、例えば、図216に示したプログラムによって実現される。

【1769】

図216の1行目の指標「RAM__INC:」は、当該RAM__INCモジュールの先頭アドレスを示す。2行目のコマンド「LDQ HL, LOW __AT__SET」によって、Qレジスタの値をHレジスタに読み出し、アドレス「__AT__SET」の下位1バイトの値をLレジスタに読み出す。かかる2行目のコマンドが、図215のステップS1に対応する。3行目のコマンド「LD A, (HL)」によって、図217に示すように、H

10

20

30

40

50

Lレジスタで示されるアドレスに格納された1バイト値(例えば、02H)がAレジスタに読み出される。図216の4行目のコマンド「INC A」によって、図217に示すようにAレジスタの値が、例えば、02Hから03Hに1だけインクリメントされる。なお、汎用性を持たせるため、3行目のコマンド「LD A, (HL)」および4行目のコマンド「INC A」の代わりに、コマンド「LD A, 1」およびコマンド「ADD A, (HL)」を用いることもできる。この場合、Aレジスタに格納する「1」の値を変えることで、インクリメントする値を1以外に変更することができる。

【1770】

図216の5行目のコマンド「JCP C, A, 5, RAM_INC01」によって、Aレジスタの値と、上限値「5」とを比較し、その結果、5未満であれば(キャリーフラグが立てば)、アドレス「RAM_INC01」に移動する。こうして、Aレジスタの値が上限値の5未満であれば、次の処理を省略して7行目の指標「RAM_INC01」に移動することができる。6行目のコマンド「LD A, 5」によって、Aレジスタに上限値「5」を格納する。これは、4行目のコマンド「INC A」によって、Aレジスタの値が5を超えた場合、強制的にAレジスタの値を5に上書きすることで、インクリメントした結果を5以下とするためのである。なお、ここでは、インクリメント前のAレジスタの値が4の場合、すなわち、Aレジスタの値が4から5にインクリメントされた場合、既にAレジスタの値は5となっているので5に更新する必要はないが、そのことを判定する処理を追加すると処理負荷が高まるので、汎用性を優先し、5を超えた場合同様、強制的にAレジスタの値を5に更新することとしている。かかる3～6行目のコマンドが、図215のステップS2に対応する。

【1771】

図216の7行目の指標「RAM_INC01」は、5行目のコマンド「JCP C, A, 5, RAM_INC01」の移動先アドレスを示す。8行目のコマンド「LD (HL), A」によって、図217に示すように、Aレジスタの値(例えば、03H)がHLレジスタで示されるアドレスに格納される。かかる8行目のコマンドが、図215のステップS3に対応する。こうして、メインRAM500cの1バイト変数を、所定数を上限に、1だけインクリメントしつつ、Aレジスタにインクリメント後の値を保持すること(Aレジスタの値をその後の処理で利用すること)が可能となる。

【1772】

このように、図216に示したRAM_INCモジュールのコマンドの総コマンドサイズは、2行目のコマンド「LDQ HL, LOW __AT__SET」2バイト+3行目のコマンド「LD A, (HL)」1バイト+4行目のコマンド「INC A」1バイト+5行目のコマンド「JCP C, A, 5, RAM_INC01」3バイト+6行目のコマンド「LD A, 5」2バイト+8行目のコマンド「LD (HL), A」1バイト=10バイトとなり、総実行サイクルは、2行目2サイクル+3行目2サイクル+4行目1サイクル+5行目4サイクル+6行目2サイクル+8行目2サイクル=13サイクルとなる。かかるRAM_INCモジュールを設けることによって、1バイト変数を、画一的かつ容易にインクリメントできるので、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1773】

図218は、RAM_INCモジュールを実現するためのコマンドの他の例を説明するための説明図である。ここでは、図216のRAM_INCモジュールのコマンド群を、図218のRAM_INCモジュールのコマンド群に置き換えている。

【1774】

図218の1行目の指標「RAM_INC:」は、当該RAM_INCモジュールの先頭アドレスを示す。2行目のコマンド「ICPLMQA (LOW __AT__SET)、5」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__AT__SET」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレス(インクリメント対象となるアドレス)に格納された1バイト値と、上限値「5」とを比較し、その結果、5

10

20

30

40

50

未満であれば（キャリーフラグが立てば）、インクリメント対象となるアドレスに格納された1バイト値を1だけインクリメントして、Aレジスタに格納するとともに、インクリメント対象となるアドレスに格納された1バイト値を更新する。また、インクリメント後の値が5以上であれば（キャリーフラグが立っていなければ）、Aレジスタに上限値「5」を格納するとともに、インクリメント対象となるアドレスに格納された1バイト値を上限値「5」に更新する。かかる2行目のコマンドが、図215のステップS1～S3に対応する。こうして、メインRAM500cの1バイト変数を、所定数を上限に、1だけインクリメントしつつ、Aレジスタにインクリメント後の値を保持すること（Aレジスタの値をその後の処理で利用すること）が可能となる。

【1775】

ここで、コマンド「ICPLMQA (k)、n」は、Qレジスタの値をアドレスの上位1バイトとし、値kをアドレスの下位1バイトとし、その対象アドレスに格納された1バイト値と、上限値nとを比較し、その結果、n未満であれば、対象アドレスに格納された1バイト値を1だけインクリメントして、Aレジスタに格納するとともに、対象アドレスに格納された1バイト値を更新し、n以上であれば、Aレジスタに上限値nを格納するとともに、対象アドレスに格納された1バイト値を上限値nに更新するコマンドである。かかるコマンドサイズは「4」であり、実行サイクルは「7」である。

【1776】

図218のRAM__INCモジュールのコマンドの総コマンドサイズは、2行目のコマンド「ICPLMQA (LOW __AT__SET)、5」4バイト＝4バイトとなり、総実行サイクルは、2行目7サイクル＝7サイクルとなる。したがって、図216の場合と比べ、総コマンドサイズが6バイト削減され、総実行サイクルが6サイクル削減されている。かかるRAM__INCモジュールによって、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1777】

また、図216と図218とを比較して理解できるように、図216において7行（2～8行目）を占有していたコマンド群を、図218においては1行（2行目）で表すことができ、コマンド数自体の削減、および、設計負荷の軽減を図ることが可能となる。

【1778】

図219は、RAM__INCモジュールを実現するためのコマンドのさらに他の例を説明するための説明図である。図218を用いて説明したRAM__INCモジュールでは、メインRAM500cの1バイト変数を、所定数を上限に、1だけインクリメントしつつ、Aレジスタにインクリメント後の値を保持する例を挙げた。しかし、処理によっては、Aレジスタにその結果を残すことなく、メインRAM500cの1バイト変数を、所定数を上限に1だけインクリメントしさえすればよい場合もある。ここでは、Aレジスタを更新することなく、1バイト変数を1だけインクリメントする例を挙げる。

【1779】

図219の1行目の指標「RAM__INC:」は、当該RAM__INCモジュールの先頭アドレスを示す。2行目のコマンド「ICPLMQ (LOW __AT__SET)、5」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__AT__SET」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレス（インクリメント対象となるアドレス）に格納された1バイト値と、上限値「5」とを比較し、その結果、5未満であれば（キャリーフラグが立てば）、インクリメント対象となるアドレスに格納された1バイト値を1だけインクリメントして更新する。また、インクリメントされた値が5以上であれば（キャリーフラグが立っていなければ）、インクリメント対象となるアドレスに格納された1バイト値を上限値「5」に更新する。かかる2行目のコマンドが、図215のステップS1～S3に対応する。こうして、メインRAM500cの1バイト変数を、所定数を上限に、1だけインクリメントすることが可能となる。

【1780】

ここで、コマンド「ICPLMQ (k)、n」は、Qレジスタの値をアドレスの上位

10

20

30

40

50

1バイトとし、値kをアドレスの下位1バイトとし、その対象アドレスに格納された1バイト値と、上限値nとを比較し、その結果、n未満であれば、対象アドレスに格納された1バイト値を1だけインクリメントして更新し、n以上であれば、対象アドレスに格納された1バイト値を上限値nに更新するコマンドである。かかるコマンドサイズは「4」であり、実行サイクルは「7」である。

【1781】

図219のRAM__INCモジュールのコマンドの総コマンドサイズは、2行目のコマンド「ICPLMQ (LOW __AT__SET)、5」4バイト=4バイトとなり、総実行サイクルは、2行目7サイクル=7サイクルとなる。したがって、図216の場合と比べ、総コマンドサイズが6バイト削減され、総実行サイクルが6サイクル削減されている。かかるRAM__INCモジュールによって、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

10

【1782】

また、ここでは、図216の例と比較して、Aレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図216の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1783】

また、図216と図219とを比較して理解できるように、図216において7行（2～8行目）を占有していたコマンド群を、図219においては1行（2行目）で表すことができ、コマンド数自体の削減、および、設計負荷の軽減を図ることが可能となる。

20

【1784】

〈コマンド「LDINTQR」〉

TABLSETモジュールは、テーブルセット処理、すなわち、メインRAM500cの変数に所定の値（初期値）を設定するための汎用モジュールである。ここでは、TABLSETモジュールをメモリマップ上の0030Hに配置する例を挙げて説明する。

【1785】

メインCPU500aは、メインROM500bからプログラムを読み出し、読み出したプログラムを遂行し、任意の処理において、サブルーチンとしてTABLSETモジュールを呼び出し、TABLSETモジュールを遂行する。TABLSETモジュールでは、メインROM500bのプログラムデータに記述された複数の1バイト値を、メインRAM500cのワークエリアにおいて変数として扱われる複数のデータを保持する領域に転送する。こうして、複数の変数の値が設定される。ここで、転送するデータの数を単にデータ数と言う。

30

【1786】

図220は、TABLSETモジュールの具体的な処理を示したフローチャートである。ここでは、任意の処理として、図89のステップS2200-11や図95のステップS2800-39で示したエラーウェイト処理、すなわち、エラー表示、警告音要求およびエラー復帰待ちを実行するERRWAITモジュールの一部を挙げて説明する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。また、当該TABLSETモジュールの説明中、第1レジスタはBレジスタであり、第2レジスタはHLレジスタであり、第1レジスタおよび第2レジスタと異なるレジスタはAレジスタであり、所定値は「1」である。なお、所定値を任意に設定できるのは言うまでもない。

40

【1787】

メインCPU500aは、図220（a）のように、任意の処理において、転送元となる1バイトデータ群の先頭アドレスをHLレジスタに設定する（S1）。そして、サブルーチンとしてTABLSETモジュールを呼び出す（S2）。

【1788】

メインCPU500aは、図220（b）のように、TABLSETモジュールにおい

50

て、BCレジスタを退避し（S3）、割込を禁止する（S4）。そして、メインCPU500aは、HLレジスタで示されるアドレスに格納された1バイト値をBレジスタに読み出す（S5）。かかる1バイト値はデータ数を示す。続いて、メインCPU500aは、HLレジスタに「1」を加えた値で示されるアドレスに格納された値で特定されるアドレスに、HLレジスタに「2」を加えた値で示されるアドレスに格納された値を転送し、HLレジスタの値に「2」を加えて次に転送するアドレスを設定する（S6）。続いて、メインCPU500aは、Bレジスタの値をデクリメント（「1」減算）し、デクリメントした結果が0であるか否か判定する（S7）。ここで、デクリメントした結果が0でなければ（S7におけるNO）、ステップS6からの処理を繰り返し、デクリメントした結果が0であれば（S7におけるYES）、割込を許可し（S8）、BCレジスタを復帰させて（S9）、当該TABLESETモジュールを終了して1段上のルーチンに戻る（S10）。

10

【1789】

図221、図222は、TABLESETモジュールを実現するためのコマンドの一例を説明するための説明図である。このうち、図221（a）は、TABLESETモジュールを呼び出す任意の処理のコマンド群を示し、図221（b）は、TABLESETモジュールのコマンド群を示し、図221（c）は、メインROM500bのプログラムデータにおける1バイトデータ群の配置を示し、図221（d）は、メインRAM500cのワークエリアにおける1バイトデータ群の配置を示す。図220で示したフローチャートは、例えば、図221に示したプログラムによって実現される。

20

【1790】

図221（a）の1行目のコマンド「LD HL, T_ERR_RCV」によって、転送元となる1バイトデータ群の先頭アドレス「T_ERR_RCV」をHLレジスタに設定する。かかる1行目のコマンドが、図220（a）のステップS1に対応する。そして、2行目のコマンド「RST TABLESET」によって、サブルーチンとしてTABLESETモジュールが呼び出される。かかる2行目のコマンドが、図220（a）のステップS2に対応する。

【1791】

図221（b）の1行目の指標「TABLESET:」は、当該TABLESETモジュールの先頭アドレスを示す。2行目のコマンド「PUSH BC」によって、BCレジスタの値がスタック領域に退避される。かかる2行目のコマンドが、図220（b）のステップS3に対応する。3行目のコマンド「DI」によって割込が禁止される。かかる3行目のコマンドが、図220（b）のステップS4に対応する。

30

【1792】

図221（b）の4行目のコマンド「LD B, (HL)」によって、HLレジスタで示されるアドレスに格納された1バイト値がBレジスタに読み出される。このとき、HLレジスタには、図221（a）で示したように、アドレス「T_ERR_RCV」が設定されている。したがって、図221（c）の2行目の1バイト値「(T_ERR_RCV - T_ERR_RCV) / 2」がデータ数（転送の繰り返し数）としてBレジスタに読み出されることとなる。なお、「T_ERR_RCV」は、転送元となる1バイトデータ群の最終アドレスの次のアドレスであり、T_ERR_RCVは転送元となる1バイトデータ群の先頭アドレスなので、T_ERR_RCV - T_ERR_RCVの値（データ数を示す1バイト値）の値が、1バイト値の総バイト数となり、その値を、アドレスと値との組み合わせ数である2で除算すると、データ数が導出される。図221（c）の例では、データ数は、 $9 / 2 = 4$ となる。かかる図221（b）の4行目のコマンドが、図220（b）のステップS5に対応する。

40

【1793】

図221（b）の5行目の指標「TABLESET10:」は繰り返し処理の先頭アドレスを示す。6行目のコマンド「INLD AC, (HL)」および7行目のコマンド「LDQ (C), A」によって、HLレジスタに「1」を加えた値で示されるアドレスに格

50

納された値で特定されるアドレスに、HLレジスタに「2」を加えた値で示されるアドレスに格納された値が格納され、HLレジスタの値に「2」が加えられて次に転送するアドレスが設定される。かかる6, 7行目のコマンドが、図220(b)のステップS6に対応する。

【1794】

例えば、HLレジスタがアドレス「T__ERR__RCV」の値であった場合、図221(c)の3行目における2バイト変数「__ERR__NUM」の下位1バイト値がCレジスタに格納され、図221(c)の3行目における「0」の値がAレジスタに格納される。

【1795】

また、図221(b)の7行目のコマンド「LDQ (C), A」は、Qレジスタの値をアドレスの上位1バイトとし、Cレジスタの値をアドレスの下位1バイトとし、そのアドレスに、Aレジスタの値を格納するコマンドである。

【1796】

例えば、Cレジスタの値、すなわち「__ERR__NUM」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに、Aレジスタの値、すなわち、「0」の値を格納する。

【1797】

ここで、図221(c)、図221(d)において、変数「__ERR__NUM」は、エラー番号を示し、変数「__CRE__TMR」は、クレジットボタン検出タイマを示し、変数「__CRE__FLG」は、クレジットボタン検出フラグを示し、変数「__SNS__OLD」はメダル通過センサービット前回状態を示す。なお、図221(d)に示す変数の配置は図のように連続している必要はなく、離隔していてもよい。

【1798】

続いて、図221(b)の8行目のコマンド「DJNZ TABLSET10」によって、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0でなければ、アドレス「TABLSET10」に移動し、デクリメントした結果が0であれば、当該コマンドの次のコマンドに処理を移す。ここでは、データ数が「4」なので、「TABLSET10」からの処理を4回繰り返すとBレジスタの値が0となる。かかる8行目のコマンドが、図220(b)のステップS7に対応する。

【1799】

図221(b)の9行目のコマンド「EI」によって割込が許可される。かかる9行目のコマンドが、図220(b)のステップS8に対応する。10行目のコマンド「POP BC」によってスタック領域に退避されていたデータがBCレジスタに復帰される。かかる10行目のコマンドが、図220(b)のステップS9に対応する。そして、11行目のコマンド「RET」によって、1段上のルーチンに戻る。かかる11行目のコマンドが、図220(b)のステップS10に対応する。

【1800】

こうして、図222に示すように、メインROM500bのプログラムデータに記述された複数の1バイト値（55H、77H、66H、22H）を、メインRAM500cのワークエリアにおける変数（「__ERR__NUM」、「__CRE__TMR」、「__CRE__FLG」、「__SNS__OLD」）に転送することが可能となる。

【1801】

TABLSETモジュールは、複数のモジュールからサブルーチンとして呼び出される。例えば、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、本前兆中処理（AT状態＝「3」）を実行するHID__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、終了画面中処理（AT状態＝「4」）を実行するFIN__LOTモジュール、図91におけるステップS2400-11で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊

10

20

30

40

50

技状態および演出状態に応じて選択的に移行し、準備中処理（ＡＴ状態＝「５」）を実行するPRE__LOTモジュール、図９１におけるステップＳ２４００－１１で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、REG中処理（ＡＴ状態＝「６」）を実行するREG__LOTモジュール、図９１におけるステップＳ２４００－１１で示した実行フラグ設定処理を実行するEXE__SETモジュールにおいて遊技状態および演出状態に応じて選択的に移行し、BIG中処理（ＡＴ状態＝「７」）を実行するBIG__LOTモジュール、図８５のステップＳ２０２０で示した設定値切り替え処理を実行するRANKSETモジュール、図９１のステップＳ２４００で示した図柄コード設定処理を実行するFGSETUPモジュール、図９６のステップＳ２９００で示した遊技移行処理を実行するGAMESETモジュール、図９６のステップＳ２９００－３で示した役物作動中処理を実行するJCGMSETモジュール等から呼び出される。

【１８０２】

このように、図２２１（ｂ）に示したTABLSETモジュールの命令の総命令サイズは、２行目の命令「PUSH BC」１バイト＋３行目の命令「DI」１バイト＋４行目の命令「LD B, (HL)」１バイト＋６行目の命令「IN LD AC, (HL)」２バイト＋７行目の命令「LDQ (C), A」２バイト＋８行目の命令「DJNZ TABLSET10」２バイト＋９行目の命令「EI」１バイト＋１０行目の命令「POP BC」１バイト＋１１行目の命令「RET」１バイト＝１２バイトとなり、総実行サイクルは、２行目３サイクル＋３行目１サイクル＋４行目２サイクル＋６行目４サイクル＋７行目３サイクル＋８行目３サイクル（または２サイクル）＋９行目１サイクル＋１０行目３サイクル＋１１行目３サイクル＝２３サイクル（または２２サイクル）となる。なお、括弧内のサイクル数は、命令「DJNZ TABLSET10」によって移動しなかった場合の実行サイクルを示している。かかるTABLSETモジュールを設けることによって、上述した各モジュール内でテーブルセット処理を行うことなく、命令サイズ１バイトの命令「RST TABLSET」で賄うことができるので、命令の短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【１８０３】

図２２３は、TABLSETモジュールを実現するための命令の他の例を説明するための説明図である。ここでは、図２２１（ｂ）のTABLSETモジュールの命令群を、図２２３（ｂ）のTABLSETモジュールの命令群に置き換えており、他の、図２２１（ａ）のTABLSETモジュールを呼び出す任意の処理の命令群、図２２１（ｃ）のメインROM５００ｂのプログラムデータにおける１バイトデータ群、および、図２２１（ｄ）のメインRAM５００ｃのワークエリアにおける１バイトデータ群は、図２２３（ａ）、図２２３（ｃ）、図２２３（ｄ）として、そのまま用いている。ここでは、図２２３（ａ）、図２２３（ｃ）、図２２３（ｄ）のように、図２２１（ａ）、図２２１（ｃ）、図２２１（ｄ）と実質的に等しい処理についてはその説明を省略し、図２２３（ｂ）の異なる処理のみを説明する。

【１８０４】

図２２３（ｂ）の１行目の指標「TABLSET:」は、当該TABLSETモジュールの先頭アドレスを示す。２行目の命令「PUSH BC」によって、BCレジスタの値がスタック領域に退避される。かかる２行目の命令が、図２２０（ｂ）のステップＳ３に対応する。３行目の命令「DI」によって割込が禁止される。かかる３行目の命令が、図２２０（ｂ）のステップＳ４に対応する。

【１８０５】

図２２３（ｂ）の４行目の命令「LD B, (HL)」によって、HLレジスタで示されるアドレスに格納された１バイト値がBレジスタに読み出される。このとき、HLレジスタには、図２２３（ａ）で示したように、アドレス「T__ERR__RCV」が設定されている。したがって、図２２３（ｃ）の２行目の１バイト値「(T__ERR__RCV

「`__ERR__RCV`」／2」がデータ数（転送の繰り返し数）としてBレジスタに読み出されることとなる。図223（c）の例では、データ数は4となる。かかる図223（b）の4行目のコマンドが、図220（b）のステップS5に対応する。

【1806】

図223（b）の5行目のコマンド「`INC HL`」によって、HLレジスタの値が1だけインクリメントされる。6行目のコマンド「`LDINTQR (HL)`」によって、HLレジスタで示されるアドレスに格納された値で特定されるアドレスに、HLレジスタに「1」を加えた値で示されるアドレスに格納された値が転送され、HLレジスタの値に「2」を加えて次に転送するアドレスが設定され、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0になるまで、当該処理が繰り返される。かかる5、6行目のコマンドが、図220（b）のステップS6、S7に対応する。

【1807】

ここで、コマンド「`LDINTQR (HL)`」は、Qレジスタの値をアドレスの上位1バイトとし、HLレジスタで示されるアドレスに格納された値をアドレスの下位1バイトとし、そのアドレスに、HLレジスタに「1」を加えた値で示されるアドレスに格納された値を転送し、HLレジスタの値に「2」を加えてHLレジスタの値を更新し、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0になるまで、値の転送を繰り返すコマンドである。かかるコマンドのコマンドサイズは「2」であり、実行サイクルは「5」である。

【1808】

例えば、HLレジスタがアドレス「`__ERR__RCV`」の値であった場合、Qレジスタの値をアドレスの上位1バイトとし、図223（c）の3行目における「`__ERR__NUM`」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに、図223（c）の3行目における「0」の値が転送され、HLレジスタの値に「2」が加えられ、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0になるまで、すなわち、4回転送が繰り返される。

【1809】

続いて、図223（b）の7行目のコマンド「`EI`」によって割込が許可される。かかる7行目のコマンドが、図220（b）のステップS8に対応する。8行目のコマンド「`POP BC`」によってスタック領域に退避されていたデータがBCレジスタに復帰される。かかる8行目のコマンドが、図220（b）のステップS9に対応する。そして、9行目のコマンド「`RET`」によって、1段上のルーチンに戻る。かかる9行目のコマンドが、図220（b）のステップS10に対応する。

【1810】

図223（b）のTABLESETモジュールのコマンドの総コマンドサイズは、2行目のコマンド「`PUSH BC`」1バイト+3行目のコマンド「`DI`」1バイト+4行目のコマンド「`LD B, (HL)`」1バイト+5行目のコマンド「`INC HL`」1バイト+6行目のコマンド「`LDINTQR (HL)`」2バイト+7行目のコマンド「`EI`」1バイト+8行目のコマンド「`POP BC`」1バイト+9行目のコマンド「`RET`」1バイト＝9バイトとなり、総実行サイクルは、2行目3サイクル+3行目1サイクル+4行目2サイクル+5行目1サイクル+6行目5サイクル+7行目1サイクル+8行目3サイクル+9行目3サイクル＝19サイクルとなる。したがって、図221（b）の場合と比べ、総コマンドサイズが3バイト削減され、総実行サイクルも少なくとも4サイクル削減される。特に、図221（b）の例では、コマンド「`DJNZ TABLESET10`」におけるBレジスタの値が2以上であれば、その値に10サイクル（6行目4サイクル+7行目3サイクル+8行目3サイクル）を乗じた分だけ総実行サイクルが増えるので、図221（b）の総実行サイクルの削減量も多くなる。かかるTABLESETモジュールによって、コマンドの更なる短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1811】

10

20

30

40

50

また、ここでは、図 2 2 1 の例と比較して、A レジスタおよび C レジスタを利用していない。したがって、A レジスタの値や C レジスタの値を意図せず更新してしまうことはない。また、図 2 2 1 の例では、A レジスタや C レジスタが既に利用されている場合、スタックして A レジスタや C レジスタの値を退避する必要があったが、ここでは、A レジスタや C レジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1 8 1 2】

また、図 2 2 1 (b) と図 2 2 3 (b) とを比較して理解できるように、図 2 2 1 (b) において 4 行 (5 行目～8 行目) を占有していたコマンド群を、図 2 2 3 (b) においては 2 行 (5、6 行目) で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

10

【1 8 1 3】

〈D E C W M Q〉

図 2 2 4 は、I P T _ P C モジュールの具体的な処理を示したフローチャートである。I P T _ P C モジュールは、上記タイマ割込み処理 (図 9 8 参照) のステップ S 3 1 0 0 - 2 1 において 4 回に 1 回読み出される時間監視処理を実行する。かかる図におけるステップ S の数値は、本図の説明においてのみ用いることとする。

【1 8 1 4】

時間監視処理は、図 8 2 ～図 9 8 の様々なステップにおいてセットされたタイマ (例えば、1 遊技間タイマ等) を 1 ずつ減算する処理である。

20

【1 8 1 5】

メイン C P U 5 0 0 a は、図 2 2 4 (a) に示すように、タイマが格納されたアドレスを H L レジスタに読み出した後 (S 1)、図 2 2 4 (b) に示す W O R D D E C モジュールを実行し (S 2)、H L レジスタに示されたアドレスの値を 1 減算する (S 3)。そして、コマンド「R E T」によって、当該 W O R D D E C モジュールを終了して I P T _ P C モジュールに戻る (S 4)。

【1 8 1 6】

図 2 2 5 は、I P T _ P C モジュールを実現するためのコマンドの一例を説明するための説明図である。図 2 2 4 で示したフローチャートは、例えば、図 2 2 5 に示したプログラムによって実現される。なお、図 2 2 5 では、一例として、1 遊技間タイマを減算する処理について説明する。

30

【1 8 1 7】

図 2 2 5 (a) の 1 行目の指標「I P T _ P C :」は、当該 I P T _ P C モジュールの先頭アドレスを示す。そして、図 2 2 5 (a) の 2 行目のコマンド「L D Q H L , (L O W _ G A M _ T M R)」によって、Q レジスタの値を H レジスタに読み出し、アドレス「_ G A M _ T M R」の下位 1 バイトの値を L レジスタに読み出す。かかる 2 行目のコマンドが、図 2 2 4 (a) のステップ S 1 に対応する。

【1 8 1 8】

そして、図 2 2 5 (a) の 3 行目のコマンド「C A L L F W O R D D E C」によって、図 2 2 5 (b) に示す W O R D D E C モジュールが呼び出される。かかる 2 行目のコマンドが、図 2 2 4 (a) のステップ S 2 に対応する。

40

【1 8 1 9】

図 2 2 5 (b) の 1 行目の指標「W O R D D E C :」は、当該 W O R D D E C モジュールの先頭アドレスを示す。そして、図 2 2 5 (b) の 2 行目のコマンド「D C P W L D (H L) , 0」によって、H L レジスタに示されるアドレス (より厳密には、H L レジスタに示されるアドレスと、その次のアドレス) に格納された 2 バイト長の値 (1 遊技間タイマ) が読み出され、読み出された値から 1 が減算 (更新) される。そして、減算することによりキャリーフラグに 1 が立っていた場合、すなわち、読み出された値が 0 (所定の値) であった場合、H L レジスタに示されるアドレス (より厳密には、H L レジスタに示されるアドレスと、その次のアドレス) に固定値「0 (特定値)」が格納される。一方、

50

キャリーフラグに1が立っていなかった場合、すなわち、読み出された値が1以上であった場合、減算された値が、HLレジスタに示されるアドレス（より厳密には、HLレジスタに示されるアドレスと、その次のアドレス）に格納される。かかる2行目のコマンドが、図224（b）のステップS3に対応する。

【1820】

そして、図225（b）の3行目のコマンド「RET」によって、当該WORDDECモジュールを終了してIPT_PCモジュールに戻る。かかる3行目のコマンドが、図224（b）のステップS4に対応する。こうして、HLレジスタで示されるアドレスに格納された2バイト長の値（1遊技間タイマ）が0になるまで、1回の処理毎に1ずつデクリメントすることが可能となる。

10

【1821】

ここで、図225（a）および図225（b）のコマンド群を図225（c）のように変更することができる。ここでは、図225（a）および図225（b）と実質的に等しい処理についてはその説明を省略し、図225（c）の異なる処理のみを説明する。

【1822】

図225（c）の2行目のコマンド「DECWMQ（LOW __GAM__TMR）」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__GAM__TMR」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値が読み出される。そして、読み出された値から1が減算（デクリメント）され、減算することによりキャリーフラグに1が立っていた場合、すなわち、読み出された値が0であった場合、そのアドレスに固定値「0」が格納される。一方、キャリーフラグに1が立っていなかった場合、すなわち、読み出された値が1以上であった場合、減算された値が、そのアドレスに格納される。

20

【1823】

ここで、図225（a）および図225（b）に示したコマンド群の総コマンドサイズは、図225（a）の2行目のコマンド「LDQ A,（LOW __GAM__TMR）」2バイト+図225（a）の3行目のコマンド「CALLF WORDDEC」2バイト+図225（b）の2行目のコマンド「DCPWL D（HL），0」4バイト+図225（b）の3行目のコマンド「RET」1バイト=9バイトとなり、総実行サイクルは、図225（a）の2行目2サイクル+図225（a）の3行目4サイクル+図225（b）の2行目9サイクル+図225（b）の3行目3サイクル=18サイクルとなる。

30

【1824】

一方、図225（c）に示したコマンド群の総コマンドサイズは、2行目のコマンド「DECWMQ（LOW __GAM__TMR）」3バイト=3バイトとなり、総実行サイクルは、2行目8サイクル=8サイクルとなる。

【1825】

したがって、図225（a）および図225（b）のコマンド群を図225（c）のコマンド群に置き換えることで、総コマンドサイズが6バイト削減され、総実行サイクルも10サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

40

【1826】

また、図225（a）と図225（c）とを比較して理解できるように、図225（a）において2行（2行目～3行目）を占有していたコマンド群を、図225（c）においては1行（2行目）で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1827】

また、図225（c）の例では、WORDDECモジュールも削除することが可能であり、貴重な汎用モジュールの領域を空け、その領域に他のモジュールを配置することができる。こうしてリソースの有効活用が可能となる。

【1828】

50

〈RIBIT〉

図226は、CMDPROCモジュールの具体的な処理を示したフローチャートである。CMDPROCモジュールは、上記サブコマンド送信処理（図98のステップS3100-11参照）を実行する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1829】

メインCPU500aは、図226に示すように、割込み待ちモニタレジスタの値を取得する（S1）。ここで、割込み待ちモニタレジスタは、1バイトで構成されており、6ビット目に、電源断予告信号が入力される。そして、スロットマシン400の電源電圧が所定値以下になると、電源断予告信号が入力され、割込み待ちモニタレジスタの6ビット目が1になり、それ以外の場合は0になっている。

【1830】

メインCPU500aは、取得した割込み待ちモニタレジスタの6ビット目を参照することで、電源断予告信号が入力されているか、すなわち、外部割込み要求があるかを判定する（S2）。そして、外部割込み要求があれば（S2におけるYES）、当該CMDPROCモジュールを終了して1段上のルーチンに戻り、外部割込み要求がなければ（S2におけるNO）、次の処理に移り、サブコマンドを副制御基板502に送信する。

【1831】

図227は、CMDPROCモジュールを実現するためのコマンドの一例を説明するための説明図である。図226で示したフローチャートは、例えば、図227に示したプログラムによって実現される。

【1832】

図227（a）の1行目の指標「CMDPROC:」は、当該CMDPROCモジュールの先頭アドレスを示す。そして、図227（a）の2行目のコマンド「IN A, (@IRR_____）」によって、Uレジスタの値をアドレスの上位1バイトとし、割込み待ちモニタレジスタのアドレスの下位1バイトを示す固定値「@IRR_____」をアドレスの下位1バイトとし、そのアドレスに格納された値をAレジスタに読み出す。かかる2行目のコマンドが、図226のステップS1に対応する。

【1833】

そして、図227（a）の3行目のコマンド「RBIT NZ, 6, A」によって、Aレジスタに格納された値の6ビット目が0でなければ、すなわち、1であれば、コマンド「RET」を実行するように、当該CMDPROCモジュールを終了して1段上のルーチンに戻る。一方、Aレジスタに格納された値の6ビット目が0であれば、次のコマンドに処理を移す。かかる3行目のコマンドが、図226のステップS2に対応する。

【1834】

ここで、図227（a）のコマンド群を図227（b）のように変更することができる。ここでは、図227（a）と実質的に等しい処理についてはその説明を省略し、図227（b）の異なる処理のみを説明する。

【1835】

図227（b）の2行目のコマンド「RIBIT NZ, 6, (@IRR_____）」によって、Uレジスタの値をアドレスの上位1バイトとし、割込み待ちモニタレジスタのアドレスの下位1バイトを示す固定値「@IRR_____」をアドレスの下位1バイトとし、そのアドレスに格納された値の6ビット目が0でなければ、コマンド「RET」を実行するように、当該CMDPROCモジュールを終了して1段上のルーチンに戻る。一方、そのアドレスに格納された値の6ビット目が0であれば、次のコマンドに処理を移す。

【1836】

ここで、図227（a）に示したコマンド群の総コマンドサイズは、2行目のコマンド「IN A, (@IRR_____）」2バイト+3行目のコマンド「RBIT NZ, 6, A」2バイト=4バイトとなり、総実行サイクルは、2行目3サイクル+3行目5（3）サイクル=8（6）サイクルとなる。なお、括弧内のサイクル数は、コマンド「RBIT

10

20

30

40

50

N Z, 6, A」によって移動しなかった場合の実行サイクルを示している。

【1837】

一方、図227(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「RIBIT NZ, 6, (@IRR_____)」3バイト=3バイトとなり、総実行サイクルは、2行目7(5)サイクル=7(5)サイクルとなる。なお、括弧内のサイクル数は、コマンド「RIBIT NZ, 6, (@IRR_____)」によって移動しなかった場合の実行サイクルを示している。

【1838】

したがって、図227(a)のコマンド群を図227(b)のコマンド群に置き換えることで、総コマンドサイズが1バイト削減され、総実行サイクルも少なくとも1サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1839】

また、図227(b)の例では、図227(a)で利用していたAレジスタを利用していない。したがって、Aレジスタの値を意図せず更新してしまうことはない。また、図227(a)の例では、Aレジスタが既に利用されている場合、スタックしてAレジスタの値を退避する必要があったが、ここでは、Aレジスタを利用しないので、スタック処理も必要ない。こうして、リソースを有効利用することが可能となる。

【1840】

また、図227(a)と図227(b)とを比較して理解できるように、図227(a)において2行(2行目~3行目)を占有していたコマンド群を、図227(b)においては1行(2行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1841】

〈AND+OR〉

図228は、SET_PLSモジュールの具体的な処理を示したフローチャートである。SET_PLSモジュールは、上記ステッピングモータ制御処理(図98のステップS3100-13参照)において読み出される励磁パターン更新処理を実行する。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1842】

メインCPU500aは、図228に示すように、ステッピングモータ452の励磁パターンが格納された励磁パターンテーブルの先頭アドレスを取得する(S1)。メインCPU500aは、取得した励磁パターンテーブルの先頭アドレスに対して、予めAレジスタに格納されたオフセット値を加算、すなわち、オフセットする(S2)。

【1843】

そして、メインCPU500aは、所定の出力ポートに格納された出力イメージを取得し(S3)、取得した出力イメージの上位4ビットをマスクする(S4)。なお、ここで取得した出力イメージは、1バイト長で構成されており、上位4ビットに、ステッピングモータ452の励磁パターンが示されている。そこで、ここでは、出力イメージの上位4ビットをマスクすることで、励磁パターンをクリアしている。

【1844】

その後、メインCPU500aは、ステップS2においてオフセットされたアドレスに示される値と、取得した出力イメージとの論理和を取ることで、励磁パターンを合成し(S5)、合成した励磁パターンを出力イメージとして更新し(S6)、当該SET_PLSモジュールを終了して1段上のルーチンに戻る(S7)。

【1845】

図229は、SET_PLSモジュールを実現するためのコマンドの一例を説明するための説明図である。図228で示したフローチャートは、例えば、図229に示したプログラムによって実現される。

【1846】

図 2 2 9 (a) の 1 行目の指標「SET_PLS:」は、当該 SET_PLS モジュールの先頭アドレスを示す。そして、図 2 2 9 (a) の 2 行目のコマンド「LD HL, T_PLS_PTN」によって、ステッピングモータ 4 5 2 の励磁パターンが格納された励磁パターンテーブルのアドレス「T_PLS_PTN」を HL レジスタに読み出す。かかる 2 行目のコマンドが、図 2 2 8 のステップ S 1 に対応する。

【1 8 4 7】

そして、図 2 2 9 (a) の 3 行目のコマンド「ADDWB HL, A」によって、HL レジスタの値に A レジスタの値（オフセット値）が加算され、HL レジスタの値が更新される。かかる 3 行目のコマンドが、図 2 2 8 のステップ S 2 に対応する。

【1 8 4 8】

その後、図 2 2 9 (a) の 4 行目のコマンド「LD A, (IY+@OFS_OUT_PRT)」によって、出力ポートのアドレス「IY+@OFS_OUT_PRT」に格納された値（出力イメージ）を A レジスタに読み出す。かかる 4 行目のコマンドが、図 2 2 8 のステップ S 3 に対応する。

【1 8 4 9】

続いて、図 2 2 9 (a) の 5 行目のコマンド「AND A, 0 0 0 0 1 1 1 1 B」によって、A レジスタの値（出力イメージ）と、固定値「0 0 0 0 1 1 1 1 B」（第 1 の値）との論理積が演算され（上位 4 ビットがマスクされ）、演算結果が A レジスタに格納される。かかる 5 行目のコマンドが、図 2 2 8 のステップ S 4 に対応する。

【1 8 5 0】

そして、図 2 2 9 (a) の 6 行目のコマンド「OR A, (HL)」によって、A レジスタの値（マスクされた出力イメージ）と、HL レジスタに示されるアドレスに格納された値（励磁パターン、第 2 の値）との論理和が演算され、演算結果が A レジスタに格納される。かかる 6 行目のコマンドが、図 2 2 8 のステップ S 5 に対応する。

【1 8 5 1】

その後、図 2 2 9 (a) の 7 行目のコマンド「LD (IY+@OFS_OUT_PRT), A」によって、A レジスタの値が出力ポートのアドレス「IY+@OFS_OUT_PRT」に格納される。かかる 7 行目のコマンドが、図 2 2 8 のステップ S 6 に対応する。そして、図 2 2 9 (a) の 8 行目のコマンド「RET」によって、当該 SET_PLS モジュールが終了して 1 段上のルーチンに戻る。かかる 8 行目のコマンドが、図 2 2 8 のステップ S 7 に対応する。

【1 8 5 2】

ここで、図 2 2 9 (a) のコマンド群を図 2 2 9 (b) のように変更することができる。ここでは、図 2 2 9 (a) と実質的に等しい処理についてはその説明を省略し、図 2 2 9 (b) の異なる処理のみを説明する。

【1 8 5 3】

図 2 2 9 (b) の 3 行目のコマンド「RST CALADRS」によって、図 2 2 9 (c) に示す CALADRS モジュールが呼び出される。図 2 2 9 (c) の 1 行目の指標「CALADRS:」は、当該 CALADRS モジュールの先頭アドレスを示す。そして、図 2 2 9 (c) の 2 行目のコマンド「ADDWB HL, A」によって、HL レジスタの値に A レジスタの値（オフセット値）が加算され、HL レジスタの値が更新される。そして、3 行目のコマンド「LD A, (HL)」によって、HL レジスタで示されるアドレスに格納された 1 バイト値が A レジスタに読み出される。そして、図 2 2 9 (c) の 4 行目のコマンド「RET」によって、当該 CALADRS モジュールを終了して SET_PLS モジュールに戻る。

【1 8 5 4】

その後、図 2 2 9 (b) の 4 行目のコマンド「AND (IY+@OFS_OUT_PRT), 0 0 0 0 1 1 1 1 B」によって、アドレス「IY+@OFS_OUT_PRT」に格納された値（出力イメージ）と、固定値「0 0 0 0 1 1 1 1 B」との論理積が演算され（上位 4 ビットがマスクされ）、演算結果がアドレス「IY+@OFS_OUT_PR

10

20

30

40

50

T」に格納される。

【1855】

そして、図229(b)の5行目のコマンド「OR (IY+@OFS_OUT_PRT), A」によって、アドレス「IY+@OFS_OUT_PRT」に格納された値(マスクされた出力イメージ)と、Aレジスタの値(励磁パターン)との論理和が演算され、演算結果がアドレス「IY+@OFS_OUT_PRT」に格納される。

【1856】

ここで、図229(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD HL, T_PLS_PTN」3バイト+3行目のコマンド「ADDWB HL, A」1バイト+4行目のコマンド「LD A, (IY+@OFS_OUT_PRT)」3バイト+5行目のコマンド「AND A, 00001111B」2バイト+6行目のコマンド「OR A, (HL)」1バイト+7行目のコマンド「LD (IY+@OFS_OUT_PRT), A」3バイト+8行目のコマンド「RET」1バイト=14バイトとなり、総実行サイクルは、2行目3サイクル+3行目1サイクル+4行目4サイクル+5行目2サイクル+6行目2サイクル+7行目4サイクル+8行目3サイクル=19サイクルとなる。

【1857】

一方、図229(b)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LD HL, T_PLS_PTN」3バイト+3行目のコマンド「RST CALADRS」1バイト+4行目のコマンド「AND (IY+@OFS_OUT_PRT), 00001111B」4バイト+5行目のコマンド「OR (IY+@OFS_OUT_PRT), A」3バイト+6行目のコマンド「RET」1バイト=12バイトとなり、総実行サイクルは、2行目3サイクル+3行目4サイクル+4行目7サイクル+5行目6サイクル+6行目3サイクル=23サイクルとなる。

【1858】

したがって、図229(a)のコマンド群を図229(b)のコマンド群に置き換えることで、総コマンドサイズが2バイト削減される。かかる置き換えによって、コマンドの短縮化を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。なお、汎用モジュールであるCALADRSモジュールは、他のモジュールによって読み出されて実行されるものであり、新たに追加するものではない。

【1859】

また、図229(a)と図229(b)とを比較して理解できるように、図229(b)のコマンド群は、図229(a)のコマンド群よりも2行減らすことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1860】

〈CPLBQ〉

図230は、OTM_ATKモジュールの具体的な処理を示したフローチャートである。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【1861】

ここで、スロットマシン400の別例としては、所謂チャンスゾーン演出状態が設けられている。チャンスゾーン演出状態は、通常演出状態よりもAT抽選に当選しやすい状態であり、所定遊技に亘って移行される。例えば、チャンスゾーン演出状態が8遊技に亘って移行され、1遊技毎にAT抽選を行う。そして、スロットマシン400の別例では、チャンスゾーン演出状態における各遊技でのAT抽選の結果を1ビットで記憶する(当選=1、非当選=0)ことで、全てのAT抽選の結果を合計8ビット(1バイト長)の当否情報として管理する。

【1862】

このようなAT抽選の結果を管理する処理をOTM_ATKモジュールによって実行する。なお、OTM_ATKモジュールは、各遊技においてAT抽選に当選した場合にのみ、図91におけるステップS2400-11に示した実行フラグ設定処理において呼び出

10

20

30

40

50

されて実行される。

【1863】

メインCPU500aは、図230に示すように、チャンスゾーン演出状態の遊技数を取得するとともに（S1）、遊技数ビットテーブルのアドレスを取得する（S2）。なお、遊技数ビットテーブルには、詳しくは後述するように、チャンスゾーン演出状態における1遊技目から8遊技目にそれぞれ対応したビット情報が連続して配置されており、各ビット情報は、その遊技に対応したビット（1ビット情報）が1であり、その他のビットが0となっている1バイト長のデータとなっている。

【1864】

メインCPU500aは、遊技数ビットテーブルから、取得した遊技数に対応するビット情報を取得するとともに（S3）、AT抽選の結果が示される1バイト長の当否情報（1バイト情報）を取得する（S4）。そして、メインCPU500aは、取得したビット情報と当否情報との論理和を演算することにより、当否情報を更新（合成）し（S5）、更新した当否情報をセーブし（S6）。当該OTM_ATKモジュールを終了して1段上のルーチンに戻る（S7）。

【1865】

図231は、OTM_ATKモジュールを実現するためのコマンドの一例を説明するための説明図である。図230で示したフローチャートは、例えば、図231に示したプログラムによって実現される。

【1866】

図231（a）の1行目の指標「OTM_ATK：」は、当該OTM_ATKモジュールの先頭アドレスを示す。そして、図231（a）の2行目のコマンド「LDQ A,（LOW __CZ__CNT）」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__CZ__CNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（遊技数）をAレジスタに読み出す。かかる2行目のコマンドが、図230のステップS1に対応する。

【1867】

そして、図231（a）の3行目のコマンド「LD HL, T__XXX__XXX」によって、図231（b）に示す、遊技数ビットテーブルの先頭アドレス「T__XXX__XXX」をHLレジスタに読み出す。かかる3行目のコマンドが、図230のステップS2に対応する。

【1868】

図231（b）の1行目の指標「T__XXX__XXX：」は、テーブル「T__XXX__XXX」の先頭アドレスを示す。図231（b）の2行目の値「00000001B」は、チャンスゾーン演出状態における1遊技目に対応し、0ビットのみが1であり、他のビットが0である。同様に、図231（b）の3行目～9行目の値は、チャンスゾーン演出状態における2遊技目～8遊技目にそれぞれ対応し、その遊技に対応するビットが1であり、他のビットが0である。

【1869】

図231（a）の4行目のコマンド「RST CALADRS」によって、CALADRSモジュールが呼び出される。なお、CALADRSモジュールでは、HLレジスタの値にAレジスタの値（遊技数）が加算され、HLレジスタの値が更新されるとともに、更新されたHLレジスタに示されるアドレスに格納された値（遊技数に対応するビット情報）がAレジスタに読み出される。かかる4行目のコマンドが、図230のステップS3に対応する。

【1870】

図231（a）の5行目のコマンド「LDQ B,（LOW __ATW__BIT）」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__ATW__BIT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（当否情報）をBレジスタに読み出す。かかる5行目のコマンドが、図230のステップS4に対

10

20

30

40

50

応する。

【1871】

図231(a)の6行目のコマンド「OR A, B」によって、Aレジスタの値（遊技数に対応するビット情報）と、Bレジスタの値（当否情報）との論理和を演算する。ここでは、当該遊技においてAT抽選に当選したときに、当否情報のうち、その遊技数に対応するビットが1となる。かかる6行目のコマンドが、図230のステップS5に対応する。

【1872】

図231(a)の7行目のコマンド「LDQ (LOW __ATW__BIT), A」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__ATW__BIT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスにAレジスタの値（更新された当否情報）を格納する。かかる7行目のコマンドが、図230のステップS6に対応する。

【1873】

そして、図231(a)の8行目のコマンド「RET」によって、当該OTM__ATKモジュールが終了して1段上のルーチンに戻る。かかる8行目のコマンドが、図230のステップS7に対応する。

【1874】

ここで、図231(a)のコマンド群を図231(c)のように変更することができる。ここでは、図231(a)と実質的に等しい処理についてはその説明を省略し、図231(c)の異なる処理のみを説明する。

【1875】

図231(c)の3行目のコマンド「CPLBQ A, (LOW __ATW__BIT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「__ATW__BIT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値（当否情報）のうち、Aレジスタの値（遊技数）に対応するビットを反転する。したがって、ここでは、当否情報のうち、AT抽選に当選した遊技（遊技数）に対応するビットが反転されて1となる。

【1876】

ここで、図231(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW __CZ__CNT)」2バイト+3行目のコマンド「LD HL, T__XXX__XXX」3バイト+4行目のコマンド「RST CALADRS」1バイト+5行目のコマンド「LDQ B, (LOW __ATW__BIT)」3バイト+6行目のコマンド「OR A, B」1バイト+7行目のコマンド「LDQ (LOW __ATW__BIT), A」2バイト+8行目のコマンド「RET」1バイト=13バイトとなり、総実行サイクルは、2行目3サイクル+3行目3サイクル+4行目4サイクル+5行目4サイクル+6行目2サイクル+7行目3サイクル+8行目3サイクル=22サイクルとなる。

【1877】

一方、図231(c)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW __CZ__CNT)」2バイト+3行目のコマンド「CPLBQ A, (LOW __ATW__BIT)」3バイト+4行目のコマンド「RET」1バイト=6バイトとなり、総実行サイクルは、2行目3サイクル+3行目6サイクル+4行目3サイクル=12サイクルとなる。

【1878】

したがって、図231(a)のコマンド群を図231(c)のコマンド群に置き換えることで、総コマンドサイズが7バイト削減され、総実行サイクルも少なくとも10サイクル削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1879】

また、図231(a)と図231(c)とを比較して理解できるように、図231(a)において5行（3行目～7行目）を占有していたコマンド群を、図231(c)におい

10

20

30

40

50

ては1行(3行目)で表すことができ、コマンド数自体の削減と、設計負荷の軽減を図ることが可能となる。

【1880】

また、図231(c)のコマンド群に置き換えることで、遊技数ビットテーブルが不要になるため、遊技数ビットテーブル分のデータ容量も削減することができる。

【1881】

〈LDSB〉

図232は、KRS_JDGモジュールを実現するためのコマンドの一例を説明するための説明図である。

【1882】

ここで、遊技機100の別例としては、大役遊技における所定のラウンド遊技において、大入賞口内に設けられた確変領域(特定領域)に遊技球が進入した場合に、大役遊技後の遊技状態を高確率遊技状態に設定する。そして、所定の特別図柄が決定された大役遊技において、所定のラウンド遊技で、大入賞口に所定の遊技数が通過したときに、確変領域を開放し、確変領域に遊技球を進入可能としている。以下では、1回目のラウンド遊技において大入賞口に遊技球が3球入球した場合、および、5回目のラウンド遊技において大入賞口に遊技球が5球入球した場合に、確変領域が開放される例を挙げて説明する。

【1883】

このような確変領域の開放を管理する処理をKRS_JDGモジュールによって実行する。なお、KRS_JDGモジュールは、図47に示した大入賞口開放制御処理において呼び出されて実行される。

【1884】

図232(a)の1行目の指標「KRS_JDG:」は、当該KRS_JDGモジュールの先頭アドレスを示す。そして、図232(a)の2行目のコマンド「LDQ A, (LOW R_ROU_CNT)」によって、Qレジスタの値をアドレスの上位1バイトとし、アドレス「R_ROU_CNT」の下位1バイトの値をアドレスの下位1バイトとし、そのアドレスに格納された値(特別電動役物連続作動回数、すなわち、ラウンド遊技数)をAレジスタに読み出す。

【1885】

そして、図232(a)の3行目のコマンド「LDB, (HL)」によって、HLレジスタに格納された値が示すアドレスの値をBレジスタに読み出す。なお、HLレジスタには、予め、図232(b)に示す、確変領域判定テーブルのアドレス「D_KRS_JDG_2」が読み出されている。

【1886】

図232(b)の1行目の指標「D_KRS_JDG_2:」は、テーブル「D_KRS_JDG_2」の先頭アドレスを示す。図232(b)の2行目の値「(@D_KRS_JDG_2-D_KRS_JDG_2-1)/2」は、7行目の指標「@D_KRS_JDG_2:」のアドレスから1行目の指標「D_KRS_JDG_2:」のアドレスおよび1を減算した値を2で除算した値であり、判定回数(確変領域が開放されるラウンド遊技の回数)を示すものである。

【1887】

図232(b)の3行目の値「@KRS_TGT_ROU_01」は、確変領域を開放する1回目のラウンド遊技を示す値(対象ラウンド値、ここでは、1)であり、図232(b)の4行目の値「@KRS_JDG_01」は、確変領域を開放する1回目のラウンド遊技において大入賞口に何個の遊技球が入球したときに確変領域を開放するかを示す値(開放識別値、ここでは、3)である。同様に、図232(b)の5行目の値「@KRS_TGT_ROU_03」は、確変領域を開放する2回目のラウンド遊技を示す値(対象ラウンド値、ここでは、5)であり、図232(b)の6行目の値「@KRS_JDG_03」は、確変領域を開放する2回目のラウンド遊技において大入賞口に何個の遊技球が入球したときに確変領域を開放するかを示す値(開放識別値、ここでは、5)である。

10

20

30

40

50

【1888】

したがって、図232(a)の3行目のコマンド「LD B, (HL)」によって、判定回数がBレジスタに読み出されることになる。

【1889】

図232(a)の4行目の指標「KRS_JDG_10:」は、当該指標「KRS_JDG_10」のアドレスを示す。図232(a)の5行目のコマンド「INC HL」によって、HLレジスタの値が1加算（インクリメント）される。その後、図232(a)の6行目のコマンド「CP A, (HL)」によって、Aレジスタの値と、HLレジスタの値によって示されるアドレスに格納された値（対象ラウンド値）とが比較され、Aレジスタの値がHLレジスタの値と同一であれば、ゼロフラグが立って1となり、HLレジスタの値と異なれば、ゼロフラグが立たずに0となる。

10

【1890】

続いて、図232(a)の7行目のコマンド「INC HL」によって、HLレジスタの値が再び1加算（インクリメント）される。その後、図232(a)の8行目のコマンド「LD C, (HL)」によって、HLレジスタで示されるアドレスの値（開放識別値）がCレジスタに読み出される。

【1891】

そして、図232(a)の9行目のコマンド「RET Z」によって、ゼロフラグが立っている場合、当該KRS_JDGモジュールを終了して1つ上のモジュールに戻る。

【1892】

また、ゼロフラグが立っていない場合、図232(a)の10行目のコマンド「DJNZ KRS_JDG_10」によって、Bレジスタの値がデクリメント（「1」減算）され、デクリメントした結果が0でなければ、アドレス「KRS_JDG_10」に移動し、デクリメントした結果が0であれば、図232(a)の11行目のコマンド「RET」によって、当該KRS_JDGモジュールを終了して1つ上のモジュールに戻る。

20

【1893】

ここで、遊技機100においては、ラウンド遊技数は最大で10である。また、確変領域を開放するための大入賞口への入球数は、各ラウンド遊技の規定数が10であることから、2～5の間に設定されることになる。すなわち、対象ラウンド値は10以下の値であり、4ビットで表現可能である。したがって、対象ラウンド値を示す値「@KRS_TGT_TROU_01」および値「@KRS_TGT_TROU_03」は、1バイト長で構成されているが、実際には4ビットでよい。

30

【1894】

また、開放識別値は5以下の値であり、3ビットで表現可能である。したがって、開放識別値を示す値「@KRS_JDG_01」および値「@KRS_JDG_03」は、1バイト長で構成されているが、実際には3ビットでよい。

【1895】

そこで、以下では、4ビットで表現可能な対象ラウンド値、および、3ビットで表現可能な開放識別値の2つの値を、1バイト長の1つの値（データ）として扱う例について説明する。

40

【1896】

図233は、KRS_JDGモジュールを実現するためのコマンドの別例を説明するための説明図である。なお、図232と実質的に等しい処理についてはその説明を省略し、図233の異なる処理のみを説明する。

【1897】

図233(a)の3行目のコマンド「LD B, (HL)」によって、HLレジスタに示されるアドレスの値をBレジスタに読み出す。なお、HLレジスタには、予め、図233(b)に示す、確変領域判定テーブルのアドレス「D_KRS_JDG_2」が読み出されている。

【1898】

50

図233(b)の3行目の値「@KRS__TGT__ROU__01*8+@KRS__JDG__01」は、上位5ビットが、確変領域を開放する1回目のラウンド遊技を示す値（対象ラウンド値）であり、下位3ビットが、確変領域を開放する1回目のラウンド遊技において大入賞口に何個の遊技球が入球したときに確変領域を開放するかを示す値（開放識別値）である。同様に、図233(b)の4行目の値「@KRS__TGT__ROU__03*8+@KRS__JDG__03」は、上位5ビットが、確変領域を開放する2回目のラウンド遊技を示す値（対象ラウンド値）であり、下位3ビットが、確変領域を開放する2回目のラウンド遊技において大入賞口に何個の遊技球が入球したときに確変領域を開放するかを示す値（開放識別値）である。

【1899】

10

図233(a)の6行目のコマンド「LDSB 2, DE, (HL)」によって、HLレジスタの値の下位3ビットがEレジスタに読み出され（他のビットは0にされる）、HLレジスタの値の上位5ビットが3ビット右にシフトされてDレジスタに読み出される（他のビットは0にされる）。すなわち、対象ラウンド値および開放識別値を分けて、対象ラウンド値がDレジスタに読み出され、開放識別値がEレジスタに読み出される。なお、コマンド「LDSB 2, DE, (HL)」における「2」は、0ビットから2ビット目までの3ビットがEレジスタに読み出されることを示す値であるとともに、それよりも上位のビット（5ビット）がシフトされてDレジスタに読み出されることを示す値である。

【1900】

20

図233(a)の7行目のコマンド「CP A, D」によって、Aレジスタの値と、Dレジスタの値（対象ラウンド値）とが比較され、Aレジスタの値がDレジスタの値と同一であれば、ゼロフラグが立って1となり、対象ラウンド値と異なれば、ゼロフラグが立たずに0となる。

【1901】

ここで、図232(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__ROU__CNT)」2バイト+3行目のコマンド「LD B, (HL)」1バイト+5行目のコマンド「INC HL」1バイト+6行目のコマンド「CP A, (HL)」1バイト+7行目のコマンド「INC HL」1バイト+8行目のコマンド「LD C, (HL)」1バイト+9行目のコマンド「RET Z」1バイト+10行目のコマンド「DJNZ KRS__JDG__10」2バイト+11行目のコマンド「RET」1バイト=11バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+5行目1サイクル+6行目2サイクル+7行目1サイクル+8行目1サイクル+9行目3(1)サイクル+10行目3(2)サイクル+11行目3(1)サイクル=19(14)サイクルとなる。また、図232(b)に示す確変領域判定テーブルは、5バイトとなる。なお、括弧内のサイクル数は、コマンドによって移動しなかった場合の実行サイクルを示している。

30

【1902】

一方、図233(a)に示したコマンド群の総コマンドサイズは、2行目のコマンド「LDQ A, (LOW R__ROU__CNT)」2バイト+3行目のコマンド「LD B, (HL)」1バイト+5行目のコマンド「INC HL」1バイト+6行目のコマンド「LDSB 2, DE, (HL)」2バイト+7行目のコマンド「CP A, D」1バイト+8行目のコマンド「RET Z」1バイト+9行目のコマンド「DJNZ KRS__JDG__10」2バイト+10行目のコマンド「RET」1バイト=11バイトとなり、総実行サイクルは、2行目3サイクル+3行目2サイクル+5行目1サイクル+6行目3サイクル+7行目1サイクル+8行目3(1)サイクル+9行目3(2)サイクル+10行目3(1)サイクル=19(14)サイクルとなる。また、図233(b)に示す確変領域判定テーブルは、3バイトとなる。なお、括弧内のサイクル数は、コマンドによって移動しなかった場合の実行サイクルを示している。

40

【1903】

したがって、図232(a)のコマンド群および図232(b)の確変領域判定テーブ

50

ルを、図 233 (a) のコマンド群および図 233 (b) の確変領域判定テーブルに置き換えることで、総サイズが 2 バイト削減される。かかる置き換えによって、コマンドの短縮化および処理負荷の軽減を図り、遊技制御処理を行うための制御領域の容量を確保することが可能となる。

【1904】

なお、コマンド「LDSB」は、合計で 8 バイト以下の 2 つの値（情報）を、1 バイト長の 1 つのデータとして扱う際に使用可能であり、上記の例以外であっても適用することができる。

【1905】

〈レジスタバンクおよびレジスタグループの管理〉

図 101 を用いて説明したように、CPU コア 700 のレジスタユニット 716 には、排他的に切り換えて利用可能な 2 つのレジスタバンク（第 1 レジスタバンク 726 および第 2 レジスタバンク 728）が設けられる。また、第 1 レジスタバンク 726 および第 2 レジスタバンク 728 それぞれには、排他的にアクセス可能な 2 つのレジスタグループ（メインレジスタグループ 726a、728a およびサブレジスタグループ 726b、728b）が設けられ、それぞれ、8 ビットのレジスタ（Q、A、F、B、C、D、E、H、L）と、16 ビットのレジスタ（IX、IY）とを含むレジスタ群が含まれる。したがって、このような同一のレジスタ群は、2 つのレジスタバンク（第 1 レジスタバンク 726、第 2 レジスタバンク 728）と、2 つのレジスタグループ（メインレジスタグループ 726a、728a、サブレジスタグループ 726b、728b）との掛け合わせで 4 つの領域に存在することとなる。

【1906】

このようなレジスタ群は、それぞれレジスタバンク間、レジスタグループ間で排他的に利用されるとともに、独立して情報を記憶できるので、他の領域のレジスタの更新に影響を受けることがない。換言すれば、メイン CPU 500a は、現在対象としているレジスタバンクおよびレジスタグループのレジスタの内容は更新できるが、他のレジスタバンクおよびレジスタグループのレジスタ群にはアクセスすることができない。

【1907】

このように、レジスタ群が、レジスタバンク間およびレジスタグループ間で独立していることを利用し、それぞれのレジスタ群を、その種別（目的、機能、契機等）により区分けすることが可能な複数の複数のプログラムそれぞれに割り当てて検討する。例えば、本実施形態のメモリ空間では、メイン ROM 500b およびメイン RAM 500c において使用領域と別領域とが割り当てられている。上述したように、メイン ROM 500b の使用領域には、遊技の進行を制御する遊技制御処理を実行するためのプログラムやデータが格納され、メイン ROM 500b の別領域には、遊技の進行に影響を及ぼさない遊技制御外処理を遂行するプログラムの命令コードおよびプログラムデータが格納されている。このように使用領域と別領域とは、目的とするプログラムが異なるため、それぞれ異なるレジスタ群が割り当てられるべきである。

【1908】

そこで、本実施形態では、使用領域における遊技制御処理の実行に用いるレジスタ群として第 1 レジスタバンク 726 を割り当て、別領域における遊技制御外処理の実行に用いるレジスタ群として第 2 レジスタバンク 728 を割り当てる。そして、メイン CPU 500a は、メイン ROM 500b の 0000h ~ 1FFFh に格納されているプログラムを参照し、第 1 レジスタバンク 726 のレジスタ群を更新して遊技制御処理を実行するとともに、メイン ROM 500b の 2000h ~ 3FFFh に格納されているプログラムを参照し、第 2 レジスタバンク 728 のレジスタ群を更新して遊技制御外処理を実行する。かかる構成により、メモリおよびレジスタの両方において、使用領域と別領域との独立性および排他性を担保することができる。

【1909】

また、本実施形態では、遊技制御処理において実行されるプログラムを 2 つに区分けす

10

20

30

40

50

る。ここでは、例えば、メイン処理とタイマ割込み処理とに区分する。メイン処理（第1処理）は、例えば、図22～図25および図27～図59（遊技機100）、図82～図97（スロットマシン400）に示したように、メインループを含む、遊技者の操作とその結果に従って順次実行される処理であり、タイマ割込み処理（第2処理）は、例えば、図26（遊技機100）、図98（スロットマシン400）に示したように、メイン処理の実行中に生じる、所定時間（例えば、4ms、1.49ms）毎のタイマ割込に応じ、メイン処理を一旦中断して実行される処理である。

【1910】

そして、本実施形態においては、メイン処理およびタイマ割込み処理のいずれか一方に、第1レジスタバンク726の2つのレジスタグループ726a、726bの一方を割り当て、メイン処理およびタイマ割込み処理の他方に、第1レジスタバンク726の2つのレジスタグループ726a、726bの他方を割り当てる。以下、第1の実施形態で、メイン処理にメインレジスタグループ726aを割り当て、タイマ割込み処理にサブレジスタグループ726bを割り当てた例を説明し、第2の実施形態で、メイン処理にサブレジスタグループ726bを割り当て、タイマ割込み処理にメインレジスタグループ726aを割り当てた例を説明する。

【1911】

（第1の実施形態）

ここでは、メイン処理にメインレジスタグループ726aを割り当て、タイマ割込み処理にサブレジスタグループ726bを割り当てた例を説明する。したがって、メインCPU500aは、メインレジスタグループ726aを参照してメイン処理を順次実行し、タイマ割込みが生じた場合に、メイン処理を中断し、一旦、レジスタグループをメインレジスタグループ726aからサブレジスタグループ726bに切り換えてタイマ割込み処理を実行し、タイマ割込み処理が終了すると、レジスタグループをメインレジスタグループ726aに戻して、メイン処理を再開する。

【1912】

ところで、上述したように、本実施形態ではQレジスタを用いている。Qレジスタに記述された値は、任意のコマンドにおいてアドレスの一部として参照される。かかるQレジスタを用いることで、Qレジスタを参照するコマンドを記述でき、コマンドサイズやサイクル数を削減することが可能となる。

【1913】

例えば、メモリ空間からデータを読み出すロード命令「LD」に関し、コマンド「LD A, (mn)」をプログラムに記述すると、16ビットのアドレスmnに格納された値をAレジスタに読み出すことができる。かかるコマンドは、コマンドサイズが3バイトでサイクル数が4サイクルとなる。これに対し、Qレジスタを参照するコマンド「LDQ A, (n)」をプログラムに記述すると、Qレジスタの値をアドレスの上位1バイトとし、アドレスnをアドレスの下位1バイトとし、そのアドレスに格納された値をAレジスタに読み出すことができる。かかるコマンドは、コマンドサイズが2バイトでサイクル数が3サイクルとなる。したがって、コマンドサイズおよびサイクル数のいずれも削減することが可能となる。

【1914】

かかるQレジスタは、上述したように、レジスタグループ毎に管理されている。すなわち、メインレジスタグループ（第1のレジスタグループ）726aのQレジスタ（第1レジスタ）と、サブレジスタグループ（第2のレジスタグループ）726bのQ'レジスタ（第2レジスタ）が存在し、互いに独立して値を保持する。

【1915】

また、当該レジスタユニット716を含むメインCPU500aの電源投入時においても、メインレジスタグループ726aのQレジスタと、サブレジスタグループ726bのQ'レジスタは、互いに独立して初期値が設定される。例えば、電源投入時において、メインレジスタグループ726aのQレジスタは、メインRAM500cの記憶領域のアドレ

スにおける複数の上位バイトの値（例えば、F 0 h～F 3）のうちの、1の上位バイトの値（例えばF 0 h）と同じ「F 0 h」（第1の値）に設定され、サブレジスタグループ7 2 6 bのQ'レジスタは、メインRAM 5 0 0 cの記憶領域のアドレスにおける複数の上位バイトの値（例えば、F 0 h～F 3）のうちの、1の上位バイトの値（例えばF 1 h）と同じ「F 1 h」に設定される。これは、メインレジスタグループ7 2 6 aを利用する処理と、サブレジスタグループ7 2 6 bを利用する処理とで、互いに参照するメモリ領域を区分けすることを想定したものである。

【1 9 1 6】

ただし、本実施形態では、メインレジスタグループ7 2 6 aを利用するメイン処理と、サブレジスタグループ7 2 6 bを利用するタイマ割り込み処理とは、同一のメモリ領域に対応付けられた変数にアクセスする。したがって、アドレスの上位バイトとして参照されるメインレジスタグループ7 2 6 aのQレジスタおよびサブレジスタグループ7 2 6 bのQ'レジスタのいずれも同じ値（例えば、F 0 h）を保持させた方が、アドレスを調整するためのコマンドが不要な分、コマンドサイズを小さくできる。

【1 9 1 7】

そこで、本実施形態では、メインレジスタグループ7 2 6 aのQレジスタおよびサブレジスタグループ7 2 6 bのQ'レジスタのいずれも同じ値（ここではF 0 h）に設定することとする。なお、メインレジスタグループ7 2 6 aのQレジスタは、初期値がF 0 h（第1の値）なので、さらに別の値を設定する必要がない。したがって、ここでは、サブレジスタグループ7 2 6 bのQ'レジスタのみF 1 hをF 0 h（第2の値）に書き替える。なお、メインCPU 5 0 0 aは、Q'レジスタを参照するコマンドが用いられる前に、Q'レジスタに値を設定する必要がある。そこで、例えば、メイン処理の初期化処理（例えば、図8 2のCPU初期化処理）において、Q'レジスタを初期化することが考えられる。

【1 9 1 8】

図2 3 4は、Q'レジスタに値を設定するコマンドの一例を示した図である。ここで、メインCPU 5 0 0 aは、メイン処理の初期化処理において、Q'レジスタにF 0 hを設定する。ただし、上述したように、メインCPU 5 0 0 aは、現在対象としているレジスタグループであるメインレジスタグループ7 2 6 aのQレジスタの内容は更新できるが、他のレジスタグループであるサブレジスタグループ7 2 6 bのQ'レジスタにはアクセスすることができない。そこで、図2 3 4のように、1行目のコマンド「EX ALL」によって、メインレジスタグループ7 2 6 aにおける8ビットのレジスタ（Q、A、F、B、C、D、E、H、L）と、16ビットのレジスタ（IX、IY）とを含むレジスタ群と、サブレジスタグループ7 2 6 bにおける8ビットのレジスタ（Q'、A'、F'、B'、C'、D'、E'、H'、L'）と、16ビットのレジスタ（IX'、IY'）とを含むレジスタ群とを切り換える。そして、2行目のコマンド「LD Q, 0 F 0 H」によって、Q'レジスタにF 0 hを設定し、3行目のコマンド「EX ALL」によって、メインレジスタグループ7 2 6 aのレジスタ群に戻す。

【1 9 1 9】

図2 3 4に示したコマンド例の総コマンドサイズは、1行目のコマンド「EX ALL」2バイト+2行目のコマンド「LD Q, 0 F 0 H」1バイト+3行目のコマンド「EX ALL」2バイト=5バイトとなり、総実行サイクルは、1行目2サイクル+2行目1サイクル+3行目2サイクル=5サイクルとなる。

【1 9 2 0】

なお、ここでは、メインレジスタグループ7 2 6 aのQレジスタとサブレジスタグループ7 2 6 bのQ'レジスタとの切り換えに、コマンド「EX ALL」を用いる例を挙げて説明したが、かかる場合に限らず、Qレジスタ同士を切り換えれば足り、例えば、コマンド「EX Q, Q'」を用いてもよい。なお、コマンド「EX Q, Q'」のコマンドサイズは2バイトで、サイクルは2サイクルなので、コマンド「EX ALL」と置換したとしても総コマンドサイズおよび総実行サイクルは変わらない。

【1 9 2 1】

10

20

30

40

50

このようなQ'レジスタを参照するコマンド種別は、上記で例示したコマンド「LDQ」のみならず、多数あり、また、プログラムの中で繰り返し記述される。したがって、Q'レジスタを参照するコマンドを利用する度にコマンドサイズおよびサイクル数の削減効果が得られることとなる。このようなQ'レジスタを参照するコマンドの利用頻度の高さに伴い、Qレジスタの参照回数も多くなる。したがって、Q'レジスタの値は確実に設定されるべきである。そこで、Q'レジスタの設定を、メイン処理の初期化処理で1度のみ行うのに代えて、または、加えて、メイン処理のループ処理を通じて繰り返し行うことも考えられる。例えば、図234のコマンド群を図88～図97（スロットマシン400）のいずれかで行う。こうすることで、メイン処理のループ処理が実行される度にQ'レジスタにF0hを設定することができるので、仮に、Q'レジスタが初期値F1hに戻ることがあったとしても、Q'レジスタを確実に設定することができ、Q'レジスタの本来の値F0hを安定的に維持することが可能となる。

10

【1922】

ただし、Q'レジスタはタイマ割込み処理で利用するので、タイマ割込みを許可するまでにはQ'レジスタにF0hを設定する。

【1923】

また、Q'レジスタの設定を、メイン処理の初期化処理で1度のみ行う、または、メイン処理のループ処理で繰り返し行うのに代えて、または、加えて、タイマ割込み処理を通じて繰り返し行うことも考えられる。こうすることで、タイマ割込みが生じる度に、Q'レジスタにF0hを設定することができるので、仮に、Q'レジスタが初期値F1hに戻ることがあったとしても、Q'レジスタを確実に設定することができ、Q'レジスタの本来の値F0hを安定的に維持することが可能となる。以下、タイマ割込み処理を通じてQ'レジスタの設定を繰り返し行う例を詳述する。

20

【1924】

図235は、Q'レジスタに値を設定するコマンドの一例を示した図である。ここでは、まず、Q'レジスタの設定に言及せず、図235(a)を用いて、通常のタイマ割込み処理を説明する。図235(a)の1行目の指標「TMR_IPT:」は、当該タイマ割込み処理の先頭アドレスを示す。2行目のコマンド「EX ALL」によって、メインレジスタグループ726aにおける8ビットのレジスタ(Q、A、F、B、C、D、E、H、L)と、16ビットのレジスタ(IX、IY)とを含むレジスタ群と、サブレジスタグループ726bにおける8ビットのレジスタ(Q'、A'、F'、B'、C'、D'、E'、H'、L')と、16ビットのレジスタ(IX'、IY')とを含むレジスタ群とを切り換える。そして、タイマ割込み処理で必要な処理を実行した後、5行目のコマンド「EX ALL」によって、サブレジスタグループ726bのレジスタ群を戻す。6行目のコマンド「EI」によって、割込が許可される。なお、ここでは、サブルーチンから戻る前にコマンド「EI」を行っても適切に割込が許可される。そして、7行目のコマンド「RETI」によって、メイン処理に戻る。

30

【1925】

図235(a)に示したコマンドの総コマンドサイズは、2行目のコマンド「EX ALL」2バイト+5行目のコマンド「EX ALL」2バイト+6行目のコマンド「EI」1バイト+7行目のコマンド「RETI」2バイト=7バイトとなり、総実行サイクルは、2行目2サイクル+5行目2サイクル+6行目1サイクル+7行目4サイクル=9サイクルとなる。

40

【1926】

ここで、図235(a)のタイマ割込み処理のいずれかのタイミングに、Q'レジスタを設定するコマンドを記述する。上述した図234の例では、コマンド「LDQ, 0F0H」の前後に、レジスタグループを切り換えるコマンド(コマンド「EX ALL」)を必要としたが、タイマ割込み処理には、既に、コマンド「EX ALL」が存在している。すなわち、図235(a)の2行目のコマンド「EX ALL」が実行されると、対象となるレジスタグループは、既に、目的とするサブレジスタグループ726bに切り換わ

50

っていることとなる。そこで、タイマ割込み処理でQ'レジスタの設定を行う場合、図235(b)のように、コマンド「EX ALL」の後に、コマンド「LD Q, 0F0H」のみを記述すればよい。なお、タイマ割込み処理の場合、処理の最後(5行目)で、再度、コマンド「EX ALL」が実行されるので、新たにコマンド「EX ALL」を記述することなく、レジスタグループもメインレジスタグループ726aに戻される。

【1927】

なお、コマンド「LD Q, 0F0H」の位置は、タイマ割込み処理内においてQ'レジスタを参照するコマンドより前でなければならない。したがって、図235(b)のように、2行目のコマンド「EX ALL」の直後とするとよい。なお、コマンド「EX ALL」の後に、Q'レジスタを参照するコマンド以外のコマンドを記述し、コマンド「LD Q, 0F0H」を記述した後に、Q'レジスタを参照するコマンドを記述することでも実現できる。

【1928】

図235(b)に示したコマンドの総コマンドサイズは、2行目のコマンド「EX ALL」2バイト+3行目のコマンド「LD Q, 0F0H」1バイト+5行目のコマンド「EX ALL」2バイト+6行目のコマンド「EI」1バイト+7行目のコマンド「RETI」2バイト=8バイトとなり、総実行サイクルは、2行目2サイクル+3行目1サイクル+5行目2サイクル+6行目1サイクル+7行目4サイクル=10サイクルとなる。図235(a)と図235(b)とを比較して理解できるように、Q'レジスタの設定を追加しても、総コマンドサイズ=1バイト、総実行サイクル=1サイクルの増加に収まる。

【1929】

図234を用いて説明したように、メイン処理の初期化処理でQ'レジスタの設定を行うと、総コマンドサイズ=5バイト、総実行サイクル=5サイクル必要となる。また、メイン処理のループ処理を通じて繰り返し行う場合も同様に、総コマンドサイズ=5バイト、総実行サイクル=5サイクル必要となる。これに対し、タイマ割込み処理でQ'レジスタの設定を行うと、コマンド「EX ALL」2個分の記述を省略できるので、総コマンドサイズ=1バイト、総実行サイクル=1サイクルの増加に留まる。したがって、Q'レジスタの設定をタイマ割込み処理のみで実行すると、少なくとも、メモリの増加を抑えることが可能となる。

【1930】

(第2の実施形態)

続いて、メイン処理にサブレジスタグループ726bを割り当て、タイマ割込み処理にメインレジスタグループ726aを割り当てた例を説明する。したがって、メインCPU500aは、まず、レジスタグループをメインレジスタグループ726aからサブレジスタグループ726bに切り換え、サブレジスタグループ726bを参照してメイン処理を実行し、タイマ割込みが生じた場合に、メイン処理を中断し、一旦、レジスタグループをサブレジスタグループ726bからメインレジスタグループ726aに切り換えてタイマ割込み処理を実行し、タイマ割込み処理が終了すると、レジスタグループをサブレジスタグループ726bに戻して、メイン処理を再開する。

【1931】

また、ここでも、メインレジスタグループ726aのQレジスタは、初期値がF0hとになっているので、さらに別の値を設定する必要はなく、サブレジスタグループ726bのQ'レジスタのみF0hに書き替える。なお、上述したように、Q'レジスタを参照するコマンドが用いられる前に設定する必要がある。そこで、例えば、メイン処理の初期化処理(例えば、図82のCPU初期化処理)において、Q'レジスタを初期化することが考えられる。

【1932】

図236は、Q'レジスタに値を設定するコマンドの一例を示した図である。ここで、メインCPU500aは、メイン処理の初期化処理において、Q'レジスタにF0hを設定する。ただし、上述したように、メインCPU500aは、現在対象としているレジスタグ

10

20

30

40

50

ループであるメインレジスタグループ726aのQレジスタの内容は更新できるが、他のレジスタグループであるサブレジスタグループ726bのQ'レジスタにはアクセスすることができない。そこで、図234同様、図236のように、1行目のコマンド「EX ALL」によって、メインレジスタグループ726aにおける8ビットのレジスタ(Q、A、F、B、C、D、E、H、L)と、16ビットのレジスタ(IX、IY)とを含むレジスタ群と、サブレジスタグループ726bにおける8ビットのレジスタ(Q'、A'、F'、B'、C'、D'、E'、H'、L')と、16ビットのレジスタ(IX'、IY')とを含むレジスタ群とを切り換える。そして、2行目のコマンド「LD Q, 0F0H」によって、Q'レジスタにF0hを設定する。

【1933】

なお、図236の例では、図234における3行目のコマンド「EX ALL」を要さない。これは、メイン処理がサブレジスタグループ726bを参照する場合、コマンド「EX ALL」によって、メインレジスタグループ726aに戻す必要がないからである。

【1934】

図236に示したコマンドの総コマンドサイズは、1行目のコマンド「EX ALL」2バイト+2行目のコマンド「LD Q, 0F0H」1バイト=3バイトとなり、総実行サイクルは、1行目2サイクル+2行目1サイクル=3サイクルとなる。

【1935】

図234を用いて説明したように、メイン処理にメインレジスタグループ726aを割り当て、タイマ割込み処理にサブレジスタグループ726bを割り当てた上で、メイン処理の初期化処理でQ'レジスタの設定を行うと、総コマンドサイズ=5バイト、総実行サイクル=5サイクル必要となる。また、メイン処理のループ処理を通じて繰り返しQ'レジスタの設定を行う場合も同様に、総コマンドサイズ=5バイト、総実行サイクル=5サイクル必要となる。これに対し、メイン処理にサブレジスタグループ726bを割り当て、タイマ割込み処理にメインレジスタグループ726aを割り当て、メイン処理の初期化処理でQ'レジスタの設定を行うと、コマンド「EX ALL」1個分の記述を省略できるので、総コマンドサイズ=3バイト、総実行サイクル=3サイクルの増加に抑えることができる。

【1936】

また、上述したように、Q'レジスタは利用頻度が高いので、Q'レジスタの値は確実に設定されるべきである。そこで、Q'レジスタの設定を、メイン処理の初期化処理で1度のみ行うのに代えて、または、加えて、メイン処理のループ処理を通じて繰り返し行うことも考えられる。例えば、図236のコマンド群のうち、1行目のコマンド「EX ALL」のみをメイン処理の初期化処理で行い、図236のコマンド群のうち、2行目のコマンド「LD Q, 0F0H」を、図88～図97(スロットマシン400)のいずれかで行う。こうすることで、メイン処理のループ処理に応じて、毎回、Q'レジスタにF0hを設定することができるので、仮に、Q'レジスタが初期値F1hに戻るがあったとしても、Q'レジスタを確実に設定することができ、Q'レジスタの本来の値F0hを安定的に維持することが可能となる。

【1937】

ただし、メイン処理においてQ'レジスタを参照するコマンドを利用するので、Q'レジスタを参照するコマンドを利用するまでにQ'レジスタにF0hを設定する。

【1938】

また、Q'レジスタの設定を、メイン処理の初期化処理で1度のみ行う、または、メイン処理のループ処理で繰り返し行うのに代えて、または、加えて、タイマ割込み処理を通じて繰り返し行うことも考えられる。こうすることで、タイマ割込みが生じる度に、Q'レジスタにF0hを設定することができるので、仮に、Q'レジスタが初期値F1hに戻るがあったとしても、Q'レジスタを確実に設定することができ、Q'レジスタの本来の値F0hを安定的に維持することが可能となる。

【1939】

10

20

30

40

50

図237は、Q'レジスタに値を設定するコマンドの一例を示した図である。ここでは、タイマ割込み処理でQ'レジスタの設定を行う。具体的に、図237の1行目の指標「TM R__IPT:」は、当該タイマ割込み処理の先頭アドレスを示す。2行目のコマンド「LD Q, 0F0H」によって、Q'レジスタにF0hを設定する。そして、メイン処理で用いていたレジスタの値を退避するため、3行目のコマンド「EX ALL」によって、メインレジスタグループ726aにおける8ビットのレジスタ(Q、A、F、B、C、D、E、H、L)と、16ビットのレジスタ(IX、IY)とを含むレジスタ群と、サブレジスタグループ726bにおける8ビットのレジスタ(Q'、A'、F'、B'、C'、D'、E'、H'、L')と、16ビットのレジスタ(IX'、IY')とを含むレジスタ群とを切り換える。続いて、タイマ割込み処理で必要な処理を実行した後、5行目のコマンド「EX ALL」によって、サブレジスタグループ726bのレジスタ群を戻す。6行目のコマンド「EI」によって、割込が許可される。なお、ここでは、サブルーチンから戻る前にコマンド「EI」を行っても適切に割込が許可される。そして、7行目のコマンド「RETI」によって、メイン処理に戻る。

【1940】

図237に示したコマンドの総コマンドサイズは、2行目のコマンド「LD Q, 0F0H」1バイト+3行目のコマンド「EX ALL」2バイト+5行目のコマンド「EX ALL」2バイト+6行目のコマンド「EI」1バイト+7行目のコマンド「RETI」2バイト=8バイトとなり、総実行サイクルは、2行目1サイクル+3行目2サイクル+5行目2サイクル+6行目1サイクル+7行目4サイクル=10サイクルとなる。

【1941】

なお、コマンド「LD Q, 0F0H」の位置は、タイマ割込み処理内においてコマンド「EX ALL」が実行されるより前でなければならない。したがって、3行目のコマンド「EX ALL」の直前とするといよい。

【1942】

ここでは、メイン処理にサブレジスタグループ726bを割り当て、タイマ割込み処理にメインレジスタグループ726aを割り当てた上で、Q'レジスタの設定を、メイン処理の初期化処理で1度のみ行う、メイン処理のループ処理を通じて繰り返し行う、または、タイマ割込み処理を通じて繰り返し行っている。いずれの場合においても、メイン処理にメインレジスタグループ726aを割り当て、タイマ割込み処理にサブレジスタグループ726bを割り当てて、メイン処理の初期化処理でQ'レジスタの設定を行う場合に比べ、コマンド「EX ALL」1個分の記述を省略できるので、総コマンドサイズ=3バイト、総実行サイクル=3サイクルの増加に留めつつ、仮に、Q'レジスタが初期値F1hに戻ることがあったとしても、Q'レジスタを確実に設定することができ、Q'レジスタの本来の値F0hを安定的に維持することが可能となる。

【1943】

〈レジスタ群の切り換え〉

ところで、メイン処理の実行中にタイマ割込みが発生すると、その割込みタイミングにおいてレジスタがどのような値となっているかに拘わらず、タイマ割込み処理においても継続してレジスタが利用される。したがって、タイマ割込みが生じた場合に、その割込みタイミングにおけるレジスタの内容を別途保持する必要がある。例えば、タイマ割込み処理からメイン処理に戻ったときにメイン処理をスムーズに再開するため、割込みタイミングにおけるレジスタの内容を、退避コマンドを用いてメインRAM500cのスタック領域に退避しなければならない。しかし、退避コマンドは、退避すべきレジスタの数に応じて多くなり、コマンドサイズのみならず、サイクル数も増える。

【1944】

本実施形態では、上述したように、コマンド「EX ALL」によって、メインレジスタグループ726aにおける8ビットのレジスタ(Q、A、F、B、C、D、E、H、L)と、16ビットのレジスタ(IX、IY)とを含むレジスタ群と、サブレジスタグループ726bにおける8ビットのレジスタ(Q'、A'、F'、B'、C'、D'、E'、H'、L')

と、16ビットのレジスタ（IX'、IY'）とを含むレジスタ群とを切り換えている。かかるコマンドでは、メイン処理とタイマ割込み処理との切り換えにおいて、対象となるレジスタグループを一度に退避するとともに、退避していた他のレジスタグループを一度に復帰させることができる。かかるコマンド「EX ALL」は、コマンドサイズが2バイトに抑えられ、サイクル数も2サイクルしか要さない。

【1945】

また、コマンド「CALLEX mn」、コマンド「RETEX」によって、メインレジスタグループ726aおよびサブレジスタグループ726bを含むレジスタバンク単位で、第1レジスタバンク726と第2レジスタバンク728とを切り換えている。かかるコマンドでは、対象となるレジスタバンクを一度に退避するとともに、退避していた他のレジスタバンクを一度に復帰させることができる。かかるコマンド「CALLEX mn」は、アドレスmnを2000h~20FFhに限定すればコマンドサイズが2バイトに抑えられ、サイクル数も4サイクルしか要さない。コマンド「RETEX」は、コマンドサイズが2バイトに抑えられ、サイクル数も5サイクルしか要さない。以下、このような切換コマンドを第1の実施形態および第2の実施形態に当て嵌めて説明する。

【1946】

図238は、レジスタバンクおよびレジスタグループの遷移を説明するための図である。上述したように、メイン処理にメインレジスタグループ726aを割り当て、タイマ割込み処理にサブレジスタグループ726bを割り当てた例（第1の実施形態）において、メインCPU500aは、メインレジスタグループ726aを参照してメイン処理を順次実行し、タイマ割込みが生じた場合に、メイン処理を中断し、コマンド「EX ALL」により、一旦、レジスタグループをメインレジスタグループ726aからサブレジスタグループ726bに切り換えて（1）、タイマ割込み処理を実行し、タイマ割込み処理が終了すると、コマンド「EX ALL」により、レジスタグループをメインレジスタグループ726aに戻して（2）、メイン処理を再開する。また、メインCPU500aは、遊技制御処理における所定のタイミングで、コマンド「CALLEX mn」により、レジスタバンクを第1レジスタバンク726から第2レジスタバンク728に切り換えて（3）、遊技制御外処理を実行し、遊技制御外処理が完了すると、コマンド「RETEX」により、レジスタバンクを第2レジスタバンク728から第1レジスタバンク726に戻して（4）、遊技制御処理を再開する。

【1947】

また、メイン処理にサブレジスタグループ726bを割り当て、タイマ割込み処理にメインレジスタグループ726aを割り当てた例（第2の実施形態）において、メインCPU500aは、まず、コマンド「EX ALL」により、レジスタグループをメインレジスタグループ726aからサブレジスタグループ726bに切り換え（1）、サブレジスタグループ726bを参照してメイン処理を実行し、タイマ割込みが生じた場合に、メイン処理を中断し、コマンド「EX ALL」により、一旦、レジスタグループをサブレジスタグループ726bからメインレジスタグループ726aに切り換えて（2）、タイマ割込み処理を実行し、タイマ割込み処理が終了すると、コマンド「EX ALL」により、レジスタグループをサブレジスタグループ726bに戻して（1）、メイン処理を再開する。また、メインCPU500aは、遊技制御処理における所定のタイミングで、コマンド「CALLEX mn」により、レジスタバンクを第1レジスタバンク726から第2レジスタバンク728に切り換えて（3）、遊技制御外処理を実行し、遊技制御外処理が完了すると、コマンド「RETEX」により、レジスタバンクを第2レジスタバンク728から第1レジスタバンク726に戻して（4）、遊技制御処理を再開する。

【1948】

このように、2つのレジスタバンクについて、第1レジスタバンク726は使用領域に割り当て、第2レジスタバンク728は別領域に割り当て、第1レジスタバンク726のメインレジスタグループ726aおよびサブレジスタグループ726bの一方をメイン処理に割り当て、他方をタイマ割込み処理に割り当てることで、各領域の独立性、排他性を

10

20

30

40

50

担保することができる。また、これらの切替とともに、切替コマンドを用いて、少なくともレジスタ群全体を一体的に切り換えることで、各領域の独立性、排他性を確保しつつ、コマンドサイズやサイクル数を削減することが可能となる。

【1949】

続いて、メインCPU500aの電源が断たれた場合について説明する。メインCPU500aは、メイン処理を実行している間に電源が断たれると、メイン処理に割り当てられたレジスタグループ（メインレジスタグループ726aまたはサブレジスタグループ726b）を、コマンド「PUSH」（例えば、コマンド「PUSH ALL」）等を通じて、メインRAM500cに退避させ、その後、例えば、図97に示した電源断時退避処理を行う。

【1950】

ここでは、メイン処理に割り当てられたレジスタグループ（メインレジスタグループ726aまたはサブレジスタグループ726b）を退避させているが、タイマ割込み処理や遊技制御外処理に割り当てられたレジスタグループの退避を割愛している。かかるタイマ割込み処理に割り当てられたレジスタグループには、タイマ割込み処理が開始された後、必要な情報が設定され、すなわち、メイン処理からレジスタの内容が引き継がれることなく、タイマ割込み処理内でのみ参照され、タイマ割込み処理が終了すると、そのレジスタの内容は不要となる。同様に、遊技制御外処理に割り当てられたレジスタグループには、遊技制御外処理が開始された後、必要な情報が設定され、すなわち、遊技制御外処理からレジスタの内容が引き継がれることなく、遊技制御外処理内でのみ参照され、遊技制御外処理が終了すると、そのレジスタの内容は不要となる。したがって、タイマ割込み処理や遊技制御外処理に割り当てられたレジスタグループを退避させなくても問題ない。

【1951】

また、メイン処理中ではなく、タイマ割込み処理中や遊技制御外処理中に電源が断たれると、以下のように処理が遂行される。タイマ割込み処理は、電源断による割込みより優先されている。すなわち、タイマ割込み処理中に電源断の割込みが生じたとしても、電源断時退避処理は即座に実行されず、タイマ割込み処理が終了してはじめて実行される。換言すれば、電源断時退避処理が開始するタイミングでは、タイマ割込み処理が完了している。なお、タイマ割込み処理は、上記のように、タイマ割込み処理内でのみ参照され、タイマ割込み処理が終了すると、そのレジスタの内容は不要となる。したがって、タイマ割込み処理中に電源断になったとしても、電源断時退避処理が開始するタイミングでは、タイマ割込み処理にかかるレジスタの内容を退避する必要がなくなるので問題は生じない。同様に、遊技制御外処理は、電源断による割込みより優先されている。すなわち、遊技制御外処理中に電源断の割込みが生じたとしても、電源断時退避処理は即座に実行されず、遊技制御外処理が終了してはじめて実行される。換言すれば、電源断時退避処理が開始するタイミングでは、タイマ割込み処理が完了している。なお、遊技制御外処理は、上記のように、遊技制御外処理内でのみ参照され、遊技制御外処理が終了すると、そのレジスタの内容は不要となる。したがって、遊技制御外処理中に電源断になったとしても、電源断時退避処理が開始するタイミングでは、遊技制御外処理にかかるレジスタの内容を退避する必要がなくなるので問題が生じない。

【1952】

なお、電源断によって、電源断時のレジスタバンクおよびレジスタグループの情報は消去される。そして、電源断の後、電源復帰すると、レジスタバンクおよびレジスタグループは、初期値である第1レジスタバンク726およびそのメインレジスタグループ726aが設定される。したがって、電源復帰すると最初に実行される処理をメイン処理とし、そのメイン処理に、第1レジスタバンク726およびそのメインレジスタグループ726aを割り当てている。こうして、電源復帰後は、メイン処理から開始されることとなる。ここでは、電源断時に、第1レジスタバンク726およびそのメインレジスタグループ726aを退避させているので、メイン処理を安定かつ確実に遂行することが可能となる。

【1953】

なお、上述した実施形態では、第1のレジスタグループをメインレジスタグループとし、第1レジスタをQレジスタとし、第1の値をF0hとし、第2のレジスタグループをサブレジスタグループとし、第2レジスタをQ'レジスタとし、第2の値をF0hとする例を挙げて説明したが、かかる場合に限らず、第1のレジスタグループをサブレジスタグループとし、第1レジスタをQ'レジスタとし、第1の値をF1hとし、第2のレジスタグループをメインレジスタグループとし、第2レジスタをQレジスタとし、第2の値をF1hとしてもよい。かかる構成では、メインレジスタグループのQレジスタにF1hを書き替えることとなる。

【1954】

(メダルレス遊技機、管理遊技機)

ところで、上述した遊技機100（パチンコ機）の代わりに、パチンコ機を封入循環式にして遊技者が遊技球に触れることなく遊技を進行することができる管理遊技機が検討されている。また、スロットマシン400の代わりに、実物のメダルの介在なしに遊技を進行することができるメダルレス遊技機が検討されている。以下、管理遊技機およびメダルレス遊技機について説明する。ここでは、主としてメダルレス遊技機を挙げ、管理遊技機およびメダルレス遊技機に共通する仕様を説明し、管理遊技機に特有の内容については、随時、補足的に説明する。

【1955】

図239は、遊技システム900を示した機能ブロック図である。遊技システム900は、メダルレス遊技機902と、専用ユニット904（特定ユニット）と、ホールコンピュータ906と、管理コンピュータ908と、カード会社データセンタ910と、遊技機情報センタ912とを含む。なお、遊技システム900では、実物のメダルの代わりに、電子的な遊技媒体（遊技価値）として電子化メダルを用いている。したがって、上述したスロットマシン400におけるメダルを用いた遊技は、メダルレス遊技機902における電子化メダルを用いた遊技に読み替えて適用することができる。

【1956】

メダルレス遊技機902では、メダルの介在なしに、スロットマシン400と同様の遊技を実行することができる。また、メダルレス遊技機902は、メダルレス遊技機902内の各種情報を専用ユニット904に出力することができる。専用ユニット904は、メダルレス遊技機902と接続され、遊技者に電子化メダルを貸し出したり、遊技者が獲得した電子化メダルを計数することができる。ホールコンピュータ906は、ホール（遊技場）に設置され、不正監視情報を収集するため、メダルレス遊技機902を含むホール内の全ての遊技機の状態を管理する。管理コンピュータ908は、ホール内の全ての専用ユニット904と接続され、専用ユニット904を介してメダルレス遊技機902の各種情報を収集する。カード会社データセンタ910は、管理コンピュータ908に接続され、メダルレス遊技機902の各種情報に基づいてカードユニットやカード媒体を管理する。遊技機情報センタ912は、カード会社データセンタ910に接続され、メダルレス遊技機902の各種情報を収集および管理する。

【1957】

図240は、メダルレス遊技機902および専用ユニット904の概略的な機械的構成を説明するための外観図であり、図241は、メダルレス遊技機902および専用ユニット904の概略的な電氣的構成を示したブロック図である。

【1958】

図73および図76と、図240および図241とを比較して理解できるように、スロットマシン400とメダルレス遊技機902とは遊技の進行に関する構造および機能はほぼ等しい。ここでは、構造および機能がスロットマシン400と等しい構成については、同一の符号を付して説明を省略し、スロットマシン400と異なる構成について詳述する。

【1959】

メダルレス遊技機902では、実物のメダルの投入、および、実物のメダルの払い出しが不要となるため、スロットマシン400におけるメダル投入部414、メダル払出装置

10

20

30

40

50

442等のメダルを流通させる機能部を設ける必要がない。一方、メダルレス遊技機902は、スロットマシン400と異なり、後述する遊技球等貸出装置接続端子板を通じて専用ユニット904と接続するメダル数制御基板920を含む。メダル数制御基板920は、計数スイッチ922、遊技メダル数表示装置924と接続される。計数スイッチ922は、遊技メダル数表示装置924近傍において押圧スイッチで構成され、遊技者が獲得し、メダルレス遊技機902で電子的に保持された（遊技の用に供することができる）電子化メダル（かかる電子化メダルの数を「遊技メダル数」と称し、電子化メダルを保持する機能部を「メダル保持部」と称する場合がある）の一部または全部を専用ユニット904に転送する操作を検出する。計数スイッチ922の操作には、500ms未満の短押しと500ms以上の長押しが準備されている。計数スイッチ922の短押しでは、電子化メダルが1枚計数され、長押しでは、後述する図252、図253で示されるように、300ms毎の計数通知を行うタイミングで電子化メダルが50枚ずつ計数される（メダル保持部に保持されたメダルが50枚未満の場合は全部の電子化メダルが計数される）。なお、図示しない、遊技メダル数クリアスイッチは、遊技者が操作できない位置に押圧スイッチで構成され、遊技メダル数表示装置924に表示された遊技メダルの数をクリアする。具体的に、メダルレス遊技機902の電源をOFFした後、当該遊技メダル数クリアスイッチを操作しながらメダルレス遊技機902の電源をONすることでクリアされる。このとき遊技メダル数クリアステータスがONされ、遊技の終了を契機に遊技メダル数クリアステータスがOFFされる。

【1960】

遊技メダル数表示装置924は、メダル保持部に保持された電子化メダルの総数（遊技メダル数）を表示する。ただし、遊技メダル数にはベットした電子化メダルは含まれない。したがって、遊技メダル数表示装置924には、遊技者が獲得した電子化メダルから、ベットした電子化メダルの数を減算した数が表示される。なお、遊技メダル数の表示は、遊技者が視認できる位置に配された5桁または6桁の7セグメントの表示器等で以下のように表される。すなわち、その数値範囲は、1営業日における最大差枚数を勘案して0～16368で表され、有意な数値の上位の桁が0となる場合、その数値を空白とする。ここで、その数値が15000以上となると、遊技者に計数を促す報知を実行し、電子化メダルの貸出処理を制限するとともに、試験用計数信号を約3500msの間出力する。ただし、遊技者は遊技を引き続き継続可能である。そして、その数値が16369以上となった場合、遊技の進行を制限する。具体的には、ベットスイッチ416、スタートスイッチ418、精算スイッチ437の受付を禁止する。なお、その数値が16369以上となった場合の代わりに、その数値が16369以上となりそうな場合に、遊技者により計数が実行されるまで電子化メダルの払出を禁止するとしてもよい。例えば、その数値が、16369から、1遊技で取得可能な最大差枚数である12枚（最大払出枚数（例えば15枚）－規定数（例えば3枚））を減算した16357以上となった場合に遊技を停止させるとしてもよい。また、遊技メダル数表示装置924は、メダルレス遊技機902で保持される遊技メダル数が更新されてから概ね300ms以内に、更新された遊技メダル数を反映して表示する。また、遊技メダル数表示装置924は、エラーコード表示機能部と共用装置とし、エラーコードを表示することができる。ただし、エラーコードは遊技不可（ベットスイッチ416、スタートスイッチ418、精算スイッチ437の受付禁止、および、電子化メダルの貸出処理、計数処理を制限）の場合にのみ表示可能となる。

【1961】

なお、スロットマシン400と機能が共通するベットスイッチ416は、メダルレス遊技機902のメダル保持部で保持される電子化メダルのうち、1遊技で必要とされる規定数の電子化メダルをベットする操作を検出する。また、図示しない1ベットスイッチによって、メダルレス遊技機902のメダル保持部で保持される電子化メダルのうち、1枚の電子化メダルをベットすることもできる。かかるベットスイッチ416や1ベットスイッチは、スロットマシン400と機能が共通する精算スイッチ437等と共用装置とし、精算機能を備えることができる。また、精算スイッチ437は、ベットスイッチ416また

10

20

30

40

50

は1ベットスイッチの1回の操作によりベットされた電子化メダルの全てをメダル保持部に返却する。かかる精算スイッチ437の操作は、電子化メダルがベットされてから、スタートスイッチ418が操作されるまでの間のみ有効である。

【1962】

専用ユニット904は、メダルレス遊技機902のメダル数制御基板920と接続される専用ユニット制御基板930を含む。専用ユニット制御基板930は、現金投入部932、カード挿入部934、貸出スイッチ936、返却スイッチ938、遊技スイッチ940、度数表示装置942、獲得メダル数表示装置944と接続される。現金投入部932は、現金を投入する投入口として機能する。カード挿入部934は、電子化メダルを蓄積可能なカード媒体の挿入および引出を可能とする。貸出スイッチ936は、押圧スイッチで構成され、専用ユニット904で保持する現金の度数に対応する電子化メダルをメダルレス遊技機902に転送する操作を検出する。返却スイッチ938は、押圧スイッチで構成され、専用ユニット904で保持する電子化メダルをカード媒体に転送し、そのカード媒体を、カード挿入部934を通じて専用ユニット904から引き出す操作を検出する。遊技スイッチ940は、押圧スイッチで構成され、専用ユニット904で保持する電子化メダルをメダルレス遊技機902に転送する操作を検出する。度数表示装置942は、専用ユニット904で保持する度数、すなわち、現金投入部932から投入された現金に相当する度数を表示する。獲得メダル数表示装置944は、専用ユニット904で保持する電子化メダルの総数を表示する。

【1963】

遊技者は、メダルレス遊技機902において遊技を試みる場合、まず、図240に示す専用ユニット904の現金投入部932に現金を投入する。そうすると、専用ユニット904の度数表示装置942に、投入した現金に相当する度数（例えば、千円の投入に対し「10」）が表示される。遊技者が、貸出スイッチ936を操作すると、専用ユニット904で保持する度数に対応する電子化メダルがメダルレス遊技機902に転送される。そうすると、メダルレス遊技機902の遊技メダル数表示装置924に転送された電子化メダルの枚数（例えば「50」）が表示される。具体的に、メダルレス遊技機902は、専用ユニット904から後述する貸出通知を受信した場合に、貸出通知が正常であれば、電子化メダルの転送を受け、貸出受領結果応答において「正常」と通知する。一方、貸出通知のうち電文長（05h）とコマンド（13h）の値は正常だが、その他の情報が異常である場合、メダルレス遊技機902は、後述する貸出受領結果応答において「異常」と通知する。また、貸出通知を正常に受信していない、または、貸出通知の電文長とコマンドの値が異常な場合、メダルレス遊技機902は、エラー表示を行うことなく、当該受信電文を破棄して、少なくとも電文長とコマンドとが正常な貸出通知を受信できるまで待機する。なお、貸出処理ができない場合、すなわち、後述する遊技機情報通知が異常である、後述する計数通知の計数メダル数が「1」以上である、遊技メダル数表示装置924に表示された遊技メダル数が15000枚以上である、受信した貸出通知のチェックサムが異常である、受信した貸出通知の貸出通番が連続していない、受信した貸出通知の貸出メダル数が「51」以上である、または、後述するメダル数制御CPU946から専用ユニット904へ送信する遊技機情報通知において、ホールコン・不正監視情報以外を通知している場合、メダルレス遊技機902は、貸出受領結果応答において「異常」を通知する。

【1964】

続いて、遊技者が、ベットスイッチ416を操作すると、電子化メダルがベットされる。このとき、遊技メダル数表示装置924には、ベットされた電子化メダルの枚数（例えば「3」）が減算された値（例えば「47」）が表示される。こうして遊技者は、スロットマシン400同様、遊技を開始することが可能となる。仮に、遊技の結果、払出枚数が11枚となる小役が入賞した場合、払い出された電子化メダルの枚数（例えば「11」）が加算される（例えば「58」となる）。

【1965】

なお、遊技者が、ベットスイッチ416を操作して電子化メダルがベットされた後であ

り、スタートスイッチ４１８が操作される（遊技開始）前に、精算スイッチ４３７が操作されると、ベットされた電子化メダルの枚数（例えば「３」）が遊技メダル数表示装置９２４に加算され（例えば「５０」となり）、ベット状態が解除される。

【１９６６】

遊技者は、遊技を終了すると、計数スイッチ９２２を操作し、メダル保持部に保持された電子化メダルを専用ユニット９０４に転送する。そうすると、専用ユニット９０４の獲得メダル数表示装置９４４に、メダル保持部に保持されていた遊技メダル数が表示されるとともに、遊技メダル数表示装置９２４には「０」が表示される。遊技者が、返却スイッチ９３８を操作すると、専用ユニット９０４で保持する電子化メダルがカードに転送され、そのカードがカード挿入部９３４を通じて専用ユニット９０４から引き出される。こうして、遊技者は、獲得した電子化メダルをカード媒体に蓄積できる。

10

【１９６７】

遊技者は、再度、遊技を試みる場合、現金の代わりに、電子化メダルが蓄積されたカード媒体をカード挿入部９３４に挿入し、カード媒体に蓄積された電子化メダルで遊技することができる。そうすると、カード媒体に蓄積された電子化メダルが専用ユニット９０４に転送され、獲得メダル数表示装置９４４には、カード媒体に蓄積されていた電子化メダルの総数が表示される。遊技者は、貸出スイッチ９３６の代わりに、遊技スイッチ９４０を操作することで、専用ユニット９０４で保持する電子化メダルをメダルレス遊技機９０２に転送することができる。

20

【１９６８】

このようなメダルレス遊技機９０２では、実物のメダルが不要となるので、擬似的にメダルを投入したり、不正に持ち込んだメダルを使用するゴト行為を防止することが可能となる。また、遊技機内に遊技媒体の投入や払い出しを行う機構を設ける必要がなくなるので、設計コストや製造コストを削減することができる。さらに、遊技者への遊技媒体の貸し出しや、獲得した遊技媒体の計数等を一元管理することで、不正防止が可能となる。また、データを一元管理することで、射幸性を抑制し、ひいては、依存症対策を強化することが可能となる。

【１９６９】

なお、管理遊技機では、メダルレス遊技機９０２における、メダル数制御基板９２０、計数スイッチ９２２、遊技メダル数表示装置９２４それぞれと同機能を有する枠制御基板、計数スイッチ、遊技球数表示装置を適用することで、メダルレス遊技機９０２同様、管理遊技機と専用ユニット９０４とで遊技球を流通させることなく、遊技を進行することが可能となる。

30

【１９７０】

また、ここでは、メダルレス遊技機９０２において、主制御基板５００とは別にメダル数制御基板９２０を設け、メダル数制御基板９２０が独立して動作する例を挙げて説明したが、かかる場合に限らず、基板やＣＰＵの構成を異ならせることができる。

【１９７１】

図２４２は、他の基板構成を説明するための説明図である。上述した実施形態では、図２４２（ａ）のように、主制御基板５００に、遊技の進行を制御するメインＣＰＵ５００ａ（第１ＣＰＵ）が配され、メダル数制御基板９２０に、遊技に供する電子化メダルを管理するメダル数制御ＣＰＵ９４６（第２ＣＰＵ）が配され、主制御基板５００とメダル数制御基板９２０とがハーネスを介して接続され、メダル数制御基板９２０と遊技球等貸出装置接続端子板９４８とがハーネスを介して接続されている。ここで、遊技球等貸出装置接続端子板９４８は、メダルレス遊技機９０２と専用ユニット９０４を接続するための接続端子板であり、電子化メダルの貸し出しに係る信号の受信、電子化メダルの貸出受領結果の送信、電子化メダルの計数に係る信号の送信、メダルレス遊技機９０２の各情報の送信を行う。また、遊技球等貸出装置接続端子板９４８は、専用ユニット９０４から電力の供給を受け、その電力をフォトカプラ等の絶縁素子の入力とし、専用ユニット９０４との接続状態を示すＶＬ接続信号を生成してメダル数制御基板９２０に出力する。かかる基板

40

50

構成では、既存の主制御基板 500 の構成の改修や占有面積の増大を必要最低限に抑えることができるので、設計コストを削減することができる。

【1972】

なお、メダル数制御基板 920 は、必ずしも主制御基板 500 と別体である必要はなく、その機能さえ満たせば、主制御基板 500 と一体的に形成されてもよい。具体的に、図 242 (b) のように、主制御基板 500 に、遊技の進行を制御するメイン CPU 500 a、および、遊技に供する電子化メダルを管理するメダル数制御 CPU 946 のいずれもが配され、主制御基板 500 と遊技球等貸出装置接続端子板 948 とがハーネスを介して接続されていてもよい。ここでは、1 枚の主制御基板 500 に、メイン CPU 500 a とメダル数制御 CPU 946 とを共に配することで、両者の情報交換に用いられるコネクタやハーネスを排除し、占有面積を縮小するとともに、情報伝達の信頼性の向上を図ることができる。

10

【1973】

また、図 242 (c) の例では、主制御基板 500 に配されたメイン CPU 500 a が、メダル数制御 CPU 946 の代わりに、遊技に供する電子化メダルを管理し、主制御基板 500 と遊技球等貸出装置接続端子板 948 とがハーネスを介して接続されている。ここでは、主制御基板 500 において、メイン CPU 500 a が、遊技の進行を制御するとともに、電子化メダルを管理するので、メイン CPU 500 a とメダル数制御 CPU 946 との情報交換を担う接続線が必要なくなり、占有面積をさらに縮小するとともに、情報伝達の信頼性の向上を図ることができる。

20

【1974】

ここで、少なくとも主制御基板 500 は、不正防止のため主基板ケースに封入しなければならない。これに加え、遊技に供する電子化メダルを管理する機能を備えるメダル数制御基板 920 が別体である場合、主制御基板 500 と共に主基板ケースに封入しなければならない。

【1975】

図 243 は、ケースの封入態様を説明するための説明図である。例えば、図 242 (a) に示すように、主制御基板 500 とメダル数制御基板 920 と遊技球等貸出装置接続端子板 948 とがそれぞれ別体に形成され、主制御基板 500 とメダル数制御基板 920 とが接続され、メダル数制御基板 920 と遊技球等貸出装置接続端子板 948 とが接続されている場合、図 243 (a) に示すように、主制御基板 500、メダル数制御基板 920、遊技球等貸出装置接続端子板 948 を全て 1 の主基板ケース 500 e に封入することが考えられる。ここでは、図 243 (a) のように、主制御基板 500 とメダル数制御基板 920 とを、ハーネスを介することなくコネクタ同士を直接固定的に接続することで一体的に形成する例を挙げて説明しているが、両者を、ハーネスを介して接続してもよい。

30

【1976】

また、図 242 (b) および図 242 (c) に示すように、主制御基板 500 にメイン CPU 500 a とメダル数制御 CPU 946 とが配され、または、メイン CPU 500 a が単独で配され、主制御基板 500 と遊技球等貸出装置接続端子板 948 とがハーネスを介して接続されている場合、主制御基板 500 と遊技球等貸出装置接続端子板 948 とが接続された状態で、図 243 (b) のように、主制御基板 500 を主基板ケース 500 e に、遊技球等貸出装置接続端子板 948 を接続端子ケース 948 e に、それぞれ個別に封入してもよいし、図 243 (c) のように、主制御基板 500 と遊技球等貸出装置接続端子板 948 とを全て 1 の主基板ケース 500 e に封入することが考えられる。

40

【1977】

いずれにしても、ここでは、主制御基板 500、メダル数制御基板 920 および遊技球等貸出装置接続端子板 948 を全てケースに封入する。また、この場合に、基板表面のみならず、基板裏面の確認も容易となる構造としなければならない。

【1978】

図 242 (a) および図 242 (b) に示した例では、メダルレス遊技機 902 におい

50

て、メインCPU500aとメダル数制御CPU946とがそれぞれ独立して動作し、メインCPU500aが遊技の進行を制御しつつ、メダル数制御CPU946が遊技に供する電子化メダルを管理する。

【1979】

図244は、メダルレス遊技機902の各機能を実行するCPUおよび領域（上述した使用領域または使用外領域）を説明するための説明図である。図244において、「◎」は、図242（a）および図242（b）のように、メインCPU500aとメダル数制御CPU946との2つのCPUで機能を分担する場合の一般的な実行部を示し、「○」は、実行可能な実行部を示し、「×」は実行不能な実行部を示す。例えば、2つのCPUで機能を分担する場合、図244のNo.12に示すように、一般に、メダル数制御CPU946の使用領域において、遊技メダル数表示装置924を制御する機能や専用ユニット904との通信を制御する機能を担うが（図244中、「◎」で示す）、メインCPU500aの使用領域において、その一部または全部を負担することも可能である（図244中、「○」で示す）。

【1980】

（メダルレス遊技機と専用ユニットとの通信）

以上示したような機能部により、メダルレス遊技機902と専用ユニット904とは、通信を介して様々な情報（電文）を交換することで互いの正常な動作を確保する。例えば、メダルレス遊技機902（ここでは、例えば、メダル数制御基板920のメダル数制御CPU946）から専用ユニット904には、以下の遊技機情報通知、計数通知、貸出受領結果応答が行われる。

【1981】

図245～図248は、遊技機情報通知のフォーマットを説明するための説明図である。図245に示すように、遊技機情報通知は、遊技機性能情報、遊技機設置情報、ホールコン・不正監視情報の3つの遊技機情報のうち1の遊技機情報を専用ユニット904に伝達するもので、電文長は18～57バイトの可変長となっている。具体的に、遊技機情報通知の電文の1バイト目には電文長（12h～39h）が示され、2バイト目にはコマンドの種別（ここでは遊技機情報通知）を示す「01h」が示される。3バイト目には、通番として00h～FFhのシーケンス番号が示される。かかる通番は、電源投入時に00hを通知し、通知の度に1だけインクリメントする。ただし、FFhの次の通知では00hではなく01hとする。

【1982】

4バイト目には、遊技機種類が示される。遊技機種類のビット7は管理媒体を示し、遊技球が「0」、遊技メダルが「1」で表され、ビット6～4は団体区分を示し、日工組が「0」、日電協が「1」で表され、ビット3～0は遊技機種類を示し、パチンコ遊技機が「1」、回胴式遊技機が「2」、アレンジボール遊技機が「3」、じゃん球遊技機が「4」で表される。5バイト目には、遊技機情報種別が示される。例えば、遊技機情報が遊技機性能情報であれば「00h」、遊技機情報が遊技機設置情報であれば「01h」、遊技機情報がホールコン・不正監視情報であれば「02h」となる。6バイト目からは、遊技機情報（遊技機性能情報、遊技機設置情報、ホールコン・不正監視情報のいずれか）が可変長で示される。

【1983】

例えば、遊技機情報種別が「00h」であれば、遊技機情報として遊技機性能情報が51バイトで表され、図246に示すように、総投入枚数、総払出枚数、MY（最大差枚数）、役物総払出枚数、連続役物総払出枚数、役物比率、連続役物比率、有利区間比率、指示込役物比率、役物等状態比率、遊技回数、予備、予約1、予約2が含まれる。遊技機性能情報は、専用ユニット904に転送された後、さらに遊技機情報センタ912に送信される。なお、かかる情報のうち、総投入枚数、総払出枚数、MY、役物総払出枚数、連続役物総払出枚数、遊技回数のバイトオーダはリトルエンディアンである。

【1984】

10

20

30

40

50

また、遊技機情報種別が「01h」であれば、遊技機情報として遊技機設置情報が40バイトで表され、図247に示すように、主制御チップID番号、主制御チップメーカーコード、主制御チップ製品コード、メダル数制御チップID番号、メダル数制御チップメーカーコード、メダル数制御チップ製品コードが含まれる。ここで、主制御チップID番号（9バイト）およびメダル数制御チップID番号（9バイト）は、上位4バイトが0で表され、続く4バイトがチップ個別ナンバーまたはチップコードで表され、最下位バイトが識別コード（LEM50A＝「21h」、LES50A＝「22h」、LEM70A＝「23h」、IDNAC8701＝「41h」、IDNAC8702＝「42h」、IDNAC8703＝「43h」）で表される。ただし、メダル数制御CPU946を非搭載の場合、9バイトが全て0で表される。なお、主制御チップID番号、主制御チップメーカーコード、主制御チップ製品コード、メダル数制御チップID番号、メダル数制御チップメーカーコード、メダル数制御チップ製品コードのバイトオーダーはビッグエンディアンである。

【1985】

また、遊技機情報種別が「02h」であれば、遊技機情報としてホールコン・不正監視情報が12～16バイトで表され、図248に示すように、遊技メダル数、投入メダル数、払出メダル数、主制御状態1、主制御状態2、遊技機エラー状態、遊技機不正1、遊技機不正2、遊技機不正3、遊技情報数、種別情報1、カウント情報1、種別情報2、カウント情報2が含まれる。ここで、遊技メダル数、投入メダル数、払出メダル数の関係は、「遊技メダル数」＝前回送信した「遊技メダル数」－「投入メダル数」＋「払出メダル数」＋前回送信した遊技メダル数の後に受信した「貸出メダル数」－前回送信した遊技メダル数の後に送信した「計数メダル数」となるので、かかる関係を満たさなければ、ホールコン・不正監視情報が異常であると判断することができる。また、遊技機不正1のビット0における設定変更中信号は、設定変更中および設定変更が行われたことを示し、設定変更中～設定変更後の1遊技終了時まで継続して出力される。また、ビット1の設定確認中信号は、設定確認中を示し、ゴト対策として、少なくとも3秒間は出力しなければならない。なお、設定確認中信号の出力中に電断が生じた場合、3秒間を計時していたタイマをリセットして、再度、3秒間、設定確認中信号を出力してもよいし、電断時のタイマの値を保存し、電源投入時（電断復帰時）に、保存しているタイマの値から計時を再開し、設定確認中信号を出力するとしてもよい。この場合、電断前および電断復帰後における設定確認中信号を出力している総時間が3秒となる。種別情報1とカウント情報1との組み合わせ、または、種別情報2とカウント情報2との組み合わせからなる遊技情報は、種別情報1、2によって、規定数（スタートスイッチ418の操作時）か払出枚数（遊技終了時）かが示され、カウント情報1、2によってその枚数が示される。なお、リプレイ時は、再遊技作動時の規定数を通知する。遊技情報数は、遊技情報の数を示し、遊技情報数が0の場合、種別情報1、カウント情報1、種別情報2、カウント情報2の4項目は送信されない。

【1986】

図249は、計数通知のフォーマットを説明するための説明図である。計数通知は、後述する計数累積メダル数を専用ユニット904に伝達するもので、電文長は7バイトの固定長となっている。具体的に、電文の1バイト目には電文長（07h）が示され、2バイト目にはコマンドの種別（ここでは計数通知）を示す「02h」が示される。3バイト目には、計数通番として00h～FFhのシーケンス番号が示される。かかる計数通番は、電源投入時に00hを通知し、通知の度に1だけインクリメントする。ただし、FFhの次の通知では00hではなく01hとする。4バイト目には、計数メダル数が示される。計数メダル数は、計数通知のタイミングで計数した電子化メダルの枚数である。5バイト目には、計数累積メダル数が示される。計数累積メダル数は、メダルレス遊技機902の電源投入時に0000hにクリアされてから計数メダル数が累積された値であり、FFFhの次の値は0000hとなる。7バイト目には、チェックサムが示される。

【1987】

図250は、貸出受領結果応答のフォーマットを説明するための説明図である。貸出受

領結果応答は、専用ユニット904から貸出通知があった場合にその受領結果を応答するもので、電文長は5バイトの固定長となっている。具体的に、電文の1バイト目には電文長(05h)が示され、2バイト目にはコマンドの種別(ここでは貸出受領結果応答)を示す「03h」が示される。3バイト目には、貸出通番として00h~FFhのシーケンス番号が示される。かかる貸出通番は、電源投入時に00hを通知し、後述する貸出メダル数受領結果が正常であれば、専用ユニット904から受信した貸出通番をそのまま反映し、貸出メダル数受領結果が異常であれば、それ以前において、専用ユニット904から貸出メダル数受領結果を正常に受信したときの貸出通番を反映する。4バイト目には、貸出メダル数受領結果(正常=00h、異常=01h)が示される。5バイト目には、チェックサムが示される。

【1988】

また、専用ユニット904からメダルレス遊技機902(ここでは、例えば、メダル数制御CPU946)には、以下の貸出通知が行われる。

【1989】

図251は、貸出通知のフォーマットを説明するための説明図である。貸出通知は、メダルレス遊技機902から計数通知を受信した場合に、貸出メダル数をメダルレス遊技機902に伝達するもので、電文長は5バイトの固定長となっている。具体的に、電文の1バイト目には電文長(05h)が示され、2バイト目にはコマンドの種別(ここでは貸出通知)を示す「13h」が示される。3バイト目には、貸出通番として00h~FFhのシーケンス番号が示される。かかる貸出通番は、電源投入時に00hを通知し、通知の度に1だけインクリメントする。ただし、FFhの次の通知では00hではなく01hとする。4バイト目には、貸出メダル数が示される。貸出メダル数は、電子化メダルの貸出枚数を示す。なお、遊技機情報通知を受信していない場合、遊技機情報種別「02h:ホールコン・不正監視情報」以外で通知された場合、計数通知の計数メダル数が「1」以上で通知された場合は、貸出メダル数「0」が通知される。5バイト目には、チェックサムが示される。

【1990】

図252は、遊技機情報通知、計数通知、貸出通知、貸出受領結果応答の通知タイミングを示したタイミングチャートである。図252に示すように、メダルレス遊技機902(ここでは、例えば、メダル数制御CPU946)は、メダルレス遊技機902の起動完了から300msec(300msec以上、310msec以内)周期で、遊技機情報通知を専用ユニット904に送信する。また、メダルレス遊技機902は、遊技機情報通知開始から100msec(90msec以上、100msec以内)で計数通知を専用ユニット904に送信する。専用ユニット904は、計数通知の受信開始から170msec以内に貸出通知をメダルレス遊技機902に送信する。メダルレス遊技機902は、貸出通知の受信完了後、10msec以内に貸出受領結果応答を専用ユニット904に通知する。こうして、メダルレス遊技機902が貸出受領結果応答を通知してから、次の遊技機情報通知を行うまでに20msec以上の時間を確保できる。

【1991】

図253は、遊技機情報通知の送信タイミングを説明するための説明図である。遊技機情報通知は、上記のように、300msec周期で専用ユニット904に通知される。遊技機情報通知には、遊技機情報として遊技機性能情報、遊技機設置情報、ホールコン・不正監視情報の3つがあり、図253のように、それぞれ送信タイミングと優先順位が異なる。例えば、遊技機設置情報は、メダルレス遊技機902の起動が完了してから60sec経過後に通知され、それ以後は60sec周期で通知される。また、遊技機性能情報は、メダルレス遊技機902の起動が完了してから180sec経過後に通知され、それ以後は180sec周期で通知される。ホールコン・不正監視情報は、メダルレス遊技機902の起動が完了してから300msec周期で通知される。ただし、3つの通知は、その送信タイミングが重なる場合がある。送信タイミングが重なった場合、遊技機情報通知は、優先順位に従って順次通知される。例えば、180secにおいて、遊技機性能情報

10

20

30

40

50

と、遊技機設置情報と、ホールコン・不正監視情報とが重なった場合、ホールコン・不正監視情報において主制御状態の更新がなく、かつ、遊技情報がなければ、まず、優先順位が高い遊技機設置情報が通知され、次の300ms後に遊技機性能情報が通知され、次の300ms後にホールコン・不正監視情報が通知される。また、60secにおいて、遊技機設置情報と、ホールコン・不正監視情報とが重なった場合、ホールコン・不正監視情報において主制御状態の更新がなく、かつ、遊技情報がなければ、まず、優先順位が高い遊技機設置情報が通知され、次の300ms後にホールコン・不正監視情報が通知される。ただし、ホールコン・不正監視情報において主制御状態の更新がある、または、遊技情報があれば、まず、優先順位が高いホールコン・不正監視情報が通知され、その後に、遊技機設置情報、遊技機性能情報が通知される。かかる構成により、メダルレス遊技機902と専用ユニット904とは、遊技機情報通知（遊技機性能情報、遊技機設置情報、ホールコン・不正監視情報）、計数通知、貸出通知、貸出受領結果応答を、適切なタイミングかつ適切な優先度で確認し合うことができる。

【1992】

（RAMでの保持内容）

なお、ここでは、図242（a）および図242（b）を用いて説明したように、メインCPU500aとメダル数制御CPU946とをそれぞれ別体で設け、それぞれのCPUが独立してROMおよびRAMを有し、独立したプログラムに基づいて処理を行う例を挙げて説明した。ただし、例えば、総投入枚数、総払出枚数、MY（最大MY）、役物総払出枚数、連続役物総払出枚数、遊技回数といった情報は、共通して、電源のONで累積が開始され、電断後の復電でリセットされるまで蓄積が維持される情報である。ここで、メインCPU500aが情報の一部を管理し、メダル数制御CPU946が他の情報を独立して管理するとした場合、以下の問題が生じる。すなわち、仮に、メインCPU500aとメダル数制御CPU946のうちいずれか一方の電源がOFFされ、他方の電源がONのままとなると、例えば、総投入枚数と総払出枚数とはリセットされた状態からカウントされているのに、役物総払出枚数、連続役物総払出枚数がリセットされずに累積された状態でカウントされるといった事象が生じるおそれがある。したがって、総投入枚数、総払出枚数、MY（最大MY）、役物総払出枚数、連続役物総払出枚数、遊技回数といった累積を伴う情報は、一方のCPUで管理することとする。例えば、ここでは、メインCPU500aが、かかる情報を全て管理する。かかる構成により、情報の不整合を回避することができ、遊技を適切に進行することが可能となる。

【1993】

（エラーの優先順位）

ここで、メインCPU500aにおけるエラーに加えて、メダル数制御CPU946においても独自のエラー判定処理を行うように設定したとする。かかる設定では、メインCPU500aとメダル数制御CPU946において同タイミングでエラーが生じた場合、メインCPU500aで生じたエラーを優先して、メインクレジット表示部430やメイン払出表示部432、スピーカ428等のデバイスで報知するとしてもよい。なお、メインCPU500aとメダル数制御CPU946において同タイミングで生じ得る特定エラー（例えば、バックアップのエラー、RAMの読み書きが異常であることを示すエラーなどの重大な問題を有すると判定されるエラー）を設定しておき、メインCPU500aとメダル数制御CPU946において同タイミングでエラーが生じた場合、その特定エラーをデバイスで報知するとしてもよい。また、メダル数制御CPU946において同タイミングで複数のエラーが生じた場合、最も新たに生じたエラーを優先して、遊技メダル数表示装置924等のデバイスで報知するとしてもよい。また、かかる場合に限らず、複数のエラーに対し予め優先順位を設定しておき、メダル数制御CPU946において同タイミングで複数のエラーが生じた場合、設定された優先順位に基づいてエラーを報知するとしてもよい。このようにエラーを報知する優先順位を予め定めることで、エラーの緊急度や優先順位に応じ、迅速かつ効果的な対応を行うことが可能となる。例えば、複数のエラーが生じた場合、優先順位が高い1のエラーが報知される。ここで、エラーの原因を取り除

き、エラーの解除操作を行うと、その時点の1のエラー報知が消去され、次に優先順位が高いエラーが順次報知される。すなわち、1の解除操作に応じて1のエラー報知が消去されることとなる。また、このような1の解除操作に応じて1のエラー報知が消去されるように制御する場合のみならず、1の解除操作に応じて、エラーの原因が取り除かれた複数のエラー報知が同時に消去されるように制御することもできる。その場合、エラーの解除操作を行うと、エラーの原因が取り除かれた複数のエラー報知が消去され、エラーの原因が取り除かれていないエラーのうち最も優先順位の高いエラーが報知される。なお、メダル数制御基板920において、別途、7セグメント等の個別デバイスを設置しておき、メダル数制御CPU946においてエラーが生じた場合、遊技メダル数表示装置924の代わりに、または、加えて当該個別デバイスで報知してもよい。

10

【1994】

(RAM異常)

また、エラーのうちRAM異常に対する処理は、メインCPU500aにおいては、使用外領域において行うとし、メダル数制御CPU946においては、使用領域で行うとする。これは、メインCPU500aにおいては、使用領域の容量に余裕がなく、メダル数制御CPU946においては、使用領域の容量に余裕があるからである。かかる構成により、メインROM500bにおけるメモリ容量の増加を抑制しつつ、遊技を適切に進行することが可能となる。

【1995】

(メインCPUとメダル数制御CPUとの通信)

上記では、メダルレス遊技機902と専用ユニット904との通信について述べた。具体的には、上述したメダル数制御CPU946と専用ユニット904とのシリアル通信と並行して、メインCPU500aとメダル数制御CPU946とがシリアル通信しており、メダル数制御CPU946は、メインCPU500aとのコマンドの送受信により情報を取得し、その情報に基づいて、上述した専用ユニット904に情報を通知している。なお、シリアル通信は、例えば、125000bpsの通信速度で表され、データの各バイトが、1ビットのスタートビット、8ビットのデータビット、1ビットのストップビットで表される。ここでは、メダル数制御CPU946と専用ユニット904との通知の前段となる、メインCPU500aとメダル数制御CPU946との通信を詳述する。

20

【1996】

メインCPU500aは、メダル数制御CPU946に対し、所定のコマンドを送信する。例えば、メインCPU500aは、遊技終了後、次遊技のスタートスイッチ418の操作が行われる前の待機中に、コマンドを識別するための識別情報、投入要求枚数を示す伝達情報、チェックサムで構成される待機中コマンドをメダル数制御CPU946に送信するとしてもよい。また、メインCPU500aは、スタートスイッチの操作時に、識別情報、投入要求枚数を示す伝達情報、チェックサムで構成されるスタート時コマンドをメダル数制御CPU946に送信するとしてもよい。また、メインCPU500aは、払出終了時に、識別情報、払出枚数を示す伝達情報、チェックサムで構成される払出終了時コマンドをメダル数制御CPU946に送信するとしてもよい。また、メインCPU500aは、1遊技の終了時に、識別情報、役比関連の伝達情報、チェックサムで構成される1遊技終了時コマンドをメダル数制御CPU946に送信するとしてもよい。また、メインCPU500aは、起動時に、識別情報、起動時に必要な伝達情報、役比関連の伝達情報、チェックサムで構成される起動時コマンドをメダル数制御CPU946に送信するとしてもよい。また、メインCPU500aは、内部状態の変更時、エラーの発生時、または、エラーの解除時に、識別情報、チェックサムで構成される変更時コマンドをメダル数制御CPU946に送信するとしてもよい。上述したコマンドの長さは任意に設定できる。

30

40

【1997】

ここで、メインCPU500aがメダル数制御CPU946へ送信するコマンドを、仮に、可変長にしたとする。そうすると、メダル数制御CPU946は、コマンドの受信開始時には、コマンドの長さが不明なため、コマンドの受信状態を何時まで維持すべきか、

50

また、どのタイミングでコマンドの解析に入るべきかが明確ではなくなる。このようにコマンドを可変長にした場合、識別情報において、そのコマンドが、どのようなコマンドであり、何バイトのコマンドの長さであることを示すことで、メダル数制御CPU946が、コマンドを容易に把握できるようにすることができる。

【1998】

また、例えば、識別情報において情報が割り当てられていないビットが存在する場合、そのビットに、メダル数制御CPU946に必要な情報である「再遊技状態」に割り当てることもできる。こうして、送信データの空き領域の有効活用を図り、再遊技状態という1バイトのコマンドを生成、送信する別途のプログラムを生成しなくて済む。

【1999】

(メダル数制御CPU946の返信)

また、メダル数制御CPU946は、メインCPU500aに対し、所定のコマンドを送信する。例えば、メダル数制御CPU946は、メインCPU500aから上記のコマンドを受信すると、識別情報、投入許可枚数および遊技メダル数を示す伝達情報、チェックサムで構成される返信コマンドをメインCPU500aに送信するとしてもよい。

【2000】

ここで、仮に、返信コマンドの識別情報の所定のビットに、メインCPU500aから送信されたコマンドをメダル数制御CPU946が正常に受信したことを示す「ACK」を対応付けたとする。かかる「ACK」は他のビットと独立して処理される。例えば、メインCPU500aは、識別情報の所定のビットが1であれば、その他のビットの内容に拘わらず、遊技の次の処理に移行することができる。また、メインCPU500aは、所定のビット以外のビットが成立していれば、遊技を進行しつつ、その成立したビットに対応する処理を行う。かかる構成により、識別情報の領域を有効利用し、情報の転送効率を高めることができる。

【2001】

(VL接続信号)

図242を用いて説明したように、遊技球等貸出装置接続端子板948は、専用ユニット904から電力の供給を受け、かかる電力をフォトカプラ等の絶縁素子の入力とし、専用ユニット904との接続状態を示すVL接続信号(特定信号)を生成してメダル数制御基板920のメダル数制御CPU946に出力する。なお、メダル数制御CPU946は、VL接続信号を受け取った旨のコマンドをメインCPU500aに送信してもよいし、送信しないとしてもよい。かかるコマンドをメインCPU500aに送信する場合、メインCPU500aは、VL接続信号に基づくコマンドに応じて、主制御基板500のみ、メダル数制御基板920のみ、または、主制御基板500およびメダル数制御基板920の起動を制限(禁止)することができる。また、VL接続信号を受け取った旨のコマンドをメインCPU500aに送信しない場合、メダル数制御CPU946は、個別に、メダル数制御基板920の起動を制限(禁止)することができる。

【2002】

ここで、VL接続信号がONであれば、メダルレス遊技機902は、専用ユニット904と適切に接続され、かつ、専用ユニット904の電源がONしていると判断できるので、各処理を遂行することができる。一方、VL接続信号がOFFであれば、メダルレス遊技機902は、専用ユニット904と適切に接続されていない(未接続)、または、専用ユニット904の電源がOFFしていると判断し、遊技の進行を制限する。具体的に、メダルレス遊技機902は、VL接続信号がOFFであれば、遊技停止処理を実行する。かかる遊技停止処理では、主制御基板500のエラー状態として遊技を制限(電子化メダルのベット、電子化メダルの精算、スタートスイッチ418の操作、および、計数処理を全て制限(禁止))している。

【2003】

ここで、計数処理は、遊技者の計数スイッチ922の操作に応じて、メダル保持部に保持された電子化メダルの一部または全部を専用ユニット904に転送する処理を言う。具

10

20

30

40

50

体的に、計数スイッチ 922 が受け付けられていない場合、メダル保持部の遊技メダル数が 0 枚の場合、V L 接続信号が O F F の場合、および、計数不可状態の場合には、計数メダル数として「0」が設定され、計数スイッチ 922 の短押しを受け付けた場合には、計数メダル数として「1」が設定される。計数スイッチ 922 の長押しを受け付けると、メダル保持部の遊技メダル数が 50 枚未満の場合、計数メダル数として全部の電子化メダルが設定され、50 枚以上の場合、計数メダル数として「50」が設定される。そして、計数累積メダル数に当該計数メダル数が加算されるとともに、計数通番が更新される。メダルレス遊技機 902 は、計数通知の電文のデータ列を整数値の列とみなして和を求めるチェックサムを実行し、専用ユニット 904 に計数通知を行う。これと並行して、メダル保持部で保持された遊技メダル数から計数メダル数が減算される。

10

【2004】

ここで、遊技が可能な間に計数スイッチ 922 が操作された場合、いかなるタイミングであっても計数処理が実行されなければならない。「遊技が可能な間」とは、メダルレス遊技機 902 と専用ユニット 904 とが接続され、かつ、共に電源が O N されている状態（遊技者が電子化メダルを借り、メダルレス遊技機 902 で遊技を行い、当該遊技の結果を計数する一連の動作が可能な状態）を示す。なお、メダルレス遊技機 902 の電源が O N されているが専用ユニット 904 の電源が O N されていない場合、または、メダルレス遊技機 902 と専用ユニット 904 とが未接続であった場合は、メダルレス遊技機 902 が計数スイッチ 922 の操作を受け付けても当該計数メダル数が消失してしまうおそれがあるので、計数スイッチ 922 の操作を無効とし、その期間は「遊技が可能な間」に含めない。また、メダルレス遊技機 902 が単体として遊技が進行しない（できない）状態となる、電源投入後の初期化処理中、設定変更中および設定確認中、ならびに、リセット等による復帰処理が必要なエラー状態も、「遊技が可能な間」には含めない。

20

【2005】

このように、V L 接続信号が O N であれば、メダルレス遊技機 902 では、遊技を進行するとともに、遊技が可能な間、計数処理を受け付ける。一方、V L 接続信号が O F F であれば、メダルレス遊技機 902 では、遊技停止処理を実行し、電子化メダルのベット、電子化メダルの精算、スタートスイッチ 418 の操作、および、上述した計数処理を全て制限（禁止）する。これは、V L 接続信号が O F F ということは、メダルレス遊技機 902 が専用ユニット 904 と適切に接続されていない、または、専用ユニット 904 の電源が O F F していると判断できるからである。

30

【2006】

しかし、V L 接続信号が O N であっても、すなわち、メダルレス遊技機 902 と専用ユニット 904 とが適切に接続されており、かつ、専用ユニット 904 の電源が O N と認識できる場合であっても、直ちに計数処理を実行しない方がよい場合がある。例えば、V L 接続信号が O N の状態で、主制御基板 500 とメダル数制御基板 920 との通信が確立されていなかったり、一旦、確立されたが、その後、通信接続が切断されている場合である。この場合も、計数処理を正常に実行できないので、メダルレス遊技機 902 が計数スイッチ 922 の操作を受け付けて、専用ユニット 904 に情報を送信したとしても、当該計数メダル数が消失してしまうおそれがある。そこで、本実施形態の変形例では、V L 接続信号のみならず、主制御基板 500 とメダル数制御基板 920 との通信の確立状態も判断し、遊技停止処理を実行するか否か判断し、通信が確立されていなければ、遊技停止処理を行い、遊技の進行を制限する。なお、ここでは、V L 接続信号が O N の状態で、主制御基板 500 とメダル数制御基板 920 との通信が確立されていない場合、および、一旦、確立されたが、その後、通信接続が切断されている場合のいずれにおいても、遊技停止処理を行う例を挙げて説明したが、かかる場合に限らず、V L 接続信号が O N の状態で、主制御基板 500 とメダル数制御基板 920 との通信が確立されていなかった場合に遊技停止処理を行うが、一旦、確立されたが、その後、通信接続が切断されている場合には遊技停止処理のうち、電子化メダルのベット、電子化メダルの精算、スタートスイッチ 418 の操作の制限のみを行い、計数処理は可能な状態を維持してもよい。

40

50

【2007】

図254は、電源投入時の通信仕様を示したフローチャートである。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。例えば、メダルレス遊技機902の電源投入時において、主制御基板500のメインCPU500aは、メダル数制御基板920のメダル数制御CPU946と、所定のコマンド（起動時コマンド、返信コマンド）を送受信し、通信の確立状態を判断する。

【2008】

まず、メインCPU500aは、自己のレジスタを設定し（S1）、前回の電断時のバックアップを確認して（S2）、信号の待ち状態に移行する。かかる待ち状態において、メインCPU500aは、所定時間後に電断が生じることを示す電断予告信号がOFFであり、かつ、メダル数制御基板920から受信したコマンド許可信号がONであることを確認し（S3）、電断予告信号がONである、または、コマンド許可信号がOFFであれば（S3におけるNO）、待ち状態を維持する。

10

【2009】

これと並行して、メダル数制御CPU946は、自己のレジスタを設定し（S4）、前回の電断時のバックアップを確認し（S5）、起動時の初期設定処理を実行して（S6）、起動時コマンドの待ち状態に移行する。かかる起動時コマンド待ち状態において、メダル数制御CPU946は、起動時コマンドを受信したか確認し（S7）、受信した場合（S7におけるYES）、その起動時コマンドは正常であるか確認し（S8）、起動時コマンドを受信していない場合（S7におけるNO）、または、起動時コマンドが異常であれば（S8におけるNO）、起動時コマンドの待ち状態を維持する。なお、起動時コマンドおよび返信コマンドは、スタートビット、コマンドの種別を示す1バイト（8ビット）の情報、ストップビット、パリティビットのシリアル信号で構成されている。したがって、メダル数制御CPU946は、コマンドの種別が起動時コマンドを示しており、かつ、パリティビットが正常な場合に、起動時コマンドが正常であると判断する。

20

【2010】

メインCPU500aは、電断予告信号がOFFであり、かつ、コマンド許可信号がONであれば（S3におけるYES）、起動時コマンドをメダル数制御CPU946に送信する（S9）。そして、メインCPU500aは、通信タイマを設定して（S10）、返信コマンドの待ち状態に移行する。かかる返信コマンド待ち状態において、メインCPU500aは、返信コマンドを受信したか確認し（S11）、受信した場合（S11におけるYES）、その返信コマンドが正常であるか確認し（S12）、返信コマンドを受信していない場合（S11におけるNO）、または、返信コマンドが異常であれば（S12におけるNO）、返信コマンドの待ち状態を維持する。なお、メインCPU500aは、コマンドの種別が返信コマンドを示しており、かつ、パリティビットが正常な場合に、返信コマンドが正常であると判断する。

30

【2011】

メインCPU500aが起動時コマンドを送信すると（S9）、メダル数制御CPU946は、起動時コマンドを受信する（S7におけるYES）。メダル数制御CPU946は、その起動時コマンドが正常であると判断すると（S8におけるYES）、信号の待ち状態に移行する。かかる待ち状態において、メダル数制御CPU946は、主制御基板500から受信したコマンド許可信号がONであることを確認し（S13）、コマンド許可信号がOFFであれば（S13におけるNO）、待ち状態を維持する。一方、コマンド許可信号がONであれば（S13におけるYES）、メダル数制御CPU946は、返信コマンドをメインCPU500aに送信し（S14）、専用ユニット904との通信を開始して、計数可能状態に移行する。

40

【2012】

メダル数制御CPU946が返信コマンドを送信すると（S14）、メインCPU500aは、返信コマンドを受信する（S11におけるYES）。メインCPU500aは、その返信コマンドが正常であると判断すると（S12におけるYES）、設定変更処理ま

50

たは電断復帰処理に移行する。なお、ここでは、メインCPU500aが、返信コマンドを受信し、その返信コマンドが正常であると判断した場合に設定変更処理または電断復帰処理に移行する例を挙げて説明した。しかし、かかる場合に限らず、メインCPU500aは、起動時コマンドを送信した（S9）後、返信コマンドを待つことなく、設定変更処理または電断復帰処理に移行し、設定変更処理または電断復帰処理が終了した後、返信コマンドの受信状態を確認し、返信コマンドが受信されていない、もしくは、返信コマンドは受信されたが正常でなければ、エラー状態に移行するとしてもよい。

【2013】

なお、返信コマンドの待ち状態では、ステップS10で設定された通信タイマが計時され、所定の時間が経過するとタイムアウトして、エラー状態に移行し、遊技メダル数表示装置924に、その内容を表示するとともに、通信処理を終了する。かかる通信処理は、電源を再投入しない限り復帰しないように設定されている。メインCPU500aは、このようなタイムアウトを通じ、メインCPU500aとメダル数制御CPU946との通信が確立されていない、もしくは、確立されたが、その後、通信接続が切断されていると判断して、遊技停止処理を行い、遊技の進行を制限する。

10

【2014】

図255は、操作時の通信仕様を示したフローチャートである。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。例えば、ベットスイッチ416、精算スイッチ437、スタートスイッチ418等が操作されると、主制御基板500のメインCPU500aは、メダル数制御基板920のメダル数制御CPU946と、コマンド（スタート時コマンド、返信コマンド）を送受信し、通信の確立状態を判断する。

20

【2015】

例えば、スタートスイッチ418が操作されると、メインCPU500aは、信号の待ち状態に移行する。かかる待ち状態において、メインCPU500aは、電断予告信号がOFFであり、かつ、コマンド許可信号がONであることを確認し（S1）、電断予告信号がONである、または、コマンド許可信号がOFFであれば（S1におけるNO）、待ち状態を維持する。

【2016】

一方、電断予告信号がOFFであり、かつ、コマンド許可信号がONであれば（S1におけるYES）、メインCPU500aは、スタート時コマンドをメダル数制御CPU946に送信し（S2）、通信タイマを設定する（S3）。

30

【2017】

メインCPU500aがスタート時コマンドを送信すると（S2）、メダル数制御CPU946は、受信したスタート時コマンドが正常か判断し（S4）、正常でなければ（S4におけるNO）、当該操作時の処理は実行しない。一方、受信したスタート時コマンドが正常であれば（S4におけるYES）、メダル数制御CPU946は、主制御基板500から受信したコマンド許可信号がONとなるまで待ち状態に移行する。かかる待ち状態において、メダル数制御CPU946は、コマンド許可信号がONであることを確認し（S5）、コマンド許可信号がOFFであれば（S5におけるNO）、待ち状態を維持する。一方、コマンド許可信号がONであれば（S5におけるYES）、メダル数制御CPU946は、返信コマンドをメインCPU500aに送信し（S6）、当該操作時の処理を終了する。

40

【2018】

メインCPU500aは、返信コマンドを受信すると、その返信コマンドが正常であるか確認し（S7）、正常であれば（S7におけるYES）、遊技の進行が許可されているか確認し（S8）、返信コマンドが異常の場合（S7におけるNO）、および、遊技の進行が許可されていない場合（S8におけるNO）、返信コマンドの待ち状態を維持する。ここで、返信コマンドが正常であり（S7におけるYES）、遊技の進行が許可されていれば（S8におけるYES）、メインCPU500aは、遊技を進行する。

【2019】

50

なお、電源投入時の通信仕様同様、返信コマンドの待ち状態では、ステップ S 3 で設定された通信タイマが計時され、所定の時間が経過するとタイムアウトして、ベットスイッチ 4 1 6、精算スイッチ 4 3 7、スタートスイッチ 4 1 8 等の操作情報が破棄される（無効となる）。このように操作を無効とすることで、遊技を進行させない設定としている。

【2020】

図 2 5 6 は、遊技終了時の通信仕様を示したフローチャートである。かかる図におけるステップ S の数値は、本図の説明においてのみ用いることとする。例えば、リール 4 1 0 が全て停止し、表示判定処理 S 2 7 0 0 が終了すると、主制御基板 5 0 0 のメイン CPU 5 0 0 a は、メダル数制御基板 9 2 0 のメダル数制御 CPU 9 4 6 と、コマンド（払出終了時コマンド、返信コマンド）を送受信し、通信の確立状態を判断する。

10

【2021】

リール 4 1 0 が全て停止し、表示判定処理 S 2 7 0 0 が終了すると、メイン CPU 5 0 0 a は、信号の待ち状態に移行する。かかる待ち状態において、メイン CPU 5 0 0 a は、電断予告信号が OFF であり、かつ、コマンド許可信号が ON であることを確認し（S 1）、電断予告信号が ON である、または、コマンド許可信号が OFF であれば（S 1 における NO）、待ち状態を維持する。

【2022】

一方、電断予告信号が OFF であり、かつ、コマンド許可信号が ON であれば（S 1 における YES）、メイン CPU 5 0 0 a は、払出終了時コマンドをメダル数制御 CPU 9 4 6 に送信し（S 2）、通信タイマを設定する（S 3）。

20

【2023】

メイン CPU 5 0 0 a が払出終了時コマンドを送信すると（S 2）、メダル数制御 CPU 9 4 6 は、受信した払出終了時のコマンドが正常か判断し（S 4）、正常でなければ（S 4 における NO）、当該遊技終了時の処理は実行しない。一方、受信した払出終了時コマンドが正常であれば（S 4 における YES）、メダル数制御 CPU 9 4 6 は、主制御基板 5 0 0 から受信したコマンド許可信号が ON となるまで待ち状態に移行する。かかる待ち状態において、メダル数制御 CPU 9 4 6 は、コマンド許可信号が ON であることを確認し（S 5）、コマンド許可信号が OFF であれば（S 5 における NO）、待ち状態を維持する。一方、コマンド許可信号が ON であれば（S 5 における YES）、メダル数制御 CPU 9 4 6 は、返信コマンドをメイン CPU 5 0 0 a に送信し（S 6）、当該遊技終了時の処理を終了する。

30

【2024】

メイン CPU 5 0 0 a は、返信コマンドを受信すると、その返信コマンドが正常であるか確認し（S 7）、返信コマンドが異常の場合（S 7 における NO）、返信コマンドの待ち状態を維持する。ここで、返信コマンドが正常であれば（S 7 における YES）、メイン RAM 5 0 0 c の更新処理が実行され、遊技が進行する。

【2025】

なお、電源投入時の通信仕様同様、返信コマンドの待ち状態では、ステップ S 3 で設定された通信タイマが計時され、所定の時間が経過するとタイムアウトして、メイン RAM 5 0 0 c の更新は行われない。なお、ここで、タイムアウトが生じなくても、図 2 5 5 を用いて説明したように、ベットスイッチ 4 1 6、精算スイッチ 4 3 7、スタートスイッチ 4 1 8 等の操作情報が破棄される（無効となる）ので、いずれにしても遊技は進行しない。

40

【2026】

ここでは、メイン CPU 5 0 0 a とメダル数制御 CPU 9 4 6 との通信の確立状態も判断し、通信が確立されていなければ遊技停止処理を実行する構成により、遊技メダル数や計数メダル数の不整合を防ぐことができ、適切に計数処理を遂行することが可能となる。

【2027】

（設定変更・設定確認）

上述したように、本実施形態では、設定値設定手段によって設定変更が行われる。設定変更を試みる場合、まず、メダルレス遊技機 9 0 2 の電源が OFF される。そして、設定

50

ドアキー４３８に所定の操作キーが挿入されてＯＦＦの位置からＯＮの位置へ回転された状態で電源スイッチ４４４を介して電源が投入されると設定変更モードに移行して、設定変更が可能な状態となり、設定変更中ステータスがＯＮとなる。ここで、設定変更中ステータスは設定変更が実行されている間ＯＮであり、それ以外はＯＦＦとなる。なお、設定変更モードの開始時にメインＲＡＭ５００ｃが異常であれば、メインＲＡＭ５００ｃの変数を全てクリアし、メインＲＡＭ５００ｃが正常であれば、メインＲＡＭ５００ｃの設定変更領域のみクリアする。なお、メインＣＰＵ５００ａが設定変更中ステータスを専用ユニット９０４に伝達する際、メダル数制御ＣＰＵ９４６は、設定変更中ステータスを取得できるが、かかる設定変更中ステータスに応じ、メダル数制御ＣＰＵ９４６が管理する遊技メダル数をクリアすることはない。設定変更が可能な状態において設定変更スイッチが押下されると、その度に設定値が１ずつ加算され、１→２→３→４→５→６→１→…といったように１から６の数値を昇順に繰り返す。ここでは、６段階の設定値のうちのいずれかの設定値に更新され、スタートスイッチ４１８が操作されると、設定値が確定する。そして、設定ドアキー４３８を元の位置（ＯＦＦの位置）に戻すことで設定変更モードが終了して遊技が可能となる。なお、設定変更中ステータスは、設定変更モード終了後、１遊技終了時にＯＦＦされる。

【２０２８】

また、設定変更モードにおいて、メインＣＰＵ５００ａは、設定変更中ステータスを専用ユニット９０４に伝達すべく、設定変更中信号を生成している。設定変更中信号は、設定変更中ステータスのＯＮ／ＯＦＦを表し、図２４８に示した遊技機情報通知におけるホールコン・不正監視情報（遊技機情報）の遊技機不正１におけるビット０に設定されており、図２５３に示したように、３００ｍｓｅｃ周期で専用ユニット９０４に通知される。

【２０２９】

ここで、図２４２（ａ）および図２４２（ｂ）で示したようにＣＰＵを独立して２つ（メインＣＰＵ５００ａおよびメダル数制御ＣＰＵ９４６）設置した場合において、メインＣＰＵ５００ａで設定変更中ステータスを管理すると仮定する。そうすると、メインＣＰＵ５００ａは、設定ドアキー４３８に所定の操作キーが挿入されてＯＦＦの位置からＯＮの位置へ回転された状態で電源スイッチ４４４を介して電源が投入されると設定変更中ステータスがＯＮであることを示すため、設定変更中信号をＯＮにして、メダル数制御ＣＰＵ９４６を通じ、遊技機情報通知を専用ユニット９０４に送信する。一方、メインＣＰＵ５００ａは、設定ドアキー４３８を元の位置（ＯＦＦの位置）に戻した後、さらに１遊技が終了した後に、設定変更中ステータスがＯＦＦになったことを示すため、設定変更中信号をＯＦＦにして、メダル数制御ＣＰＵ９４６を通じ、遊技機情報通知を専用ユニット９０４に送信する。換言すれば、設定変更中信号は１遊技の終了時までＯＮ状態を維持しなければならない。かかる処理をメインＣＰＵ５００ａで実現しようとする、設定変更中信号を設定する処理のみならず、１遊技の終了を待って設定変更中信号をＯＦＦにする処理を実行するプログラムが必要となり、使用領域の特に制御領域が圧迫されるおそれがある。そこで、ここでは、設定変更中信号を設定する処理および１遊技の終了を待って設定変更中信号をＯＦＦにする処理を、メインＣＰＵ５００ａではなく、メダル数制御ＣＰＵ９４６で実現する。

【２０３０】

図２５７は、設定変更中信号の更新処理を示したフローチャートである。かかる図におけるステップＳの数値は、本図の説明においてのみ用いることとする。所定の操作キーがＯＮの状態では電源スイッチ４４４がＯＮされることで、メインＣＰＵ５００ａは、設定値を変更する設定変更処理を実行する（Ｓ１）。そして、設定変更を含む初期化処理が完了すると、遊技者によるベットスイッチ４１６の操作に応じて、メインＣＰＵ５００ａは、電子化メダルをベットするベット処理を実行する（Ｓ２）。続いて、遊技者によるスタートスイッチ４１８の操作に応じて、メインＣＰＵ５００ａは、抽選処理を実行する（Ｓ３）。そして、メインＣＰＵ５００ａは、リール４１０ａ、４１０ｂ、４１０ｃの回転を開始してリール回転処理を実行する（Ｓ４）。続いて、遊技者によるストップスイッチ４２

0 a、420 b、420 cの操作に応じて、操作されたストップスイッチ420 a、420 b、420 cに対応するリール410 a、410 b、410 cを停止制御するリール停止処理を実行する(S5)。リール410 a、410 b、410 cが停止すると、メインCPU500 aは、有効ラインA上に表示された図柄組み合わせが、いずれかの当選役に対応するか否か判定する判定処理を実行し(S6)、その判定結果に基づき、有効ラインA上に小役に対応する図柄組み合わせが表示されると、当該小役に対応するメダルの払出処理を実行する(S7)。このように、ステップS2からステップS7までの一連の処理を通じて1遊技が実行される。以後は、ステップS2からステップS7までを繰り返すこととなる。

【2031】

また、メインCPU500 aは、設定変更処理(S1)の開始時に、設定変更が開始したことを示す変更時コマンド(第1コマンド)をメダル数制御CPU946に送信する。メダル数制御CPU946は、かかる変更時コマンドを受信すると、図248に示したホールコン・不正監視情報の遊技機不正1におけるビット0を1に設定することで、設定変更中信号をONする(S11)。また、メインCPU500 aは、払出処理(S7)において、メダルの払い出しが終了すると、払出終了時コマンド(第2コマンド)をメダル数制御CPU946に送信する。メダル数制御CPU946は、かかる払出終了時コマンドを受信すると、ホールコン・不正監視情報の遊技機不正1におけるビット0を0に設定することで、設定変更中信号をOFFする(S12)。このように、設定値の変更の開始時に設定変更中信号をONして専用ユニット904に送信し、設定値の変更を完了した後、1遊技の終了に応じ設定変更中信号をOFFして専用ユニット904に送信する機能部を設定変更中信号更新部と称する場合がある。なお、第1コマンドは、変更時コマンドに限らず、設定値の変更の開始時に、その旨が特定されさえすれば、任意のコマンドとすることができ、第2コマンドは、払出終了時コマンドに限らず、遊技が完了する度に、その旨が特定されさえすれば、任意のコマンドとすることができる。

【2032】

このように、メインCPU500 aの代わりに、メダル数制御CPU946において、設定変更中信号を設定する処理を実現することで、メインROM500 bのプログラムの増加を抑制できる。したがって、使用領域の特に制御領域が圧迫されることがない。なお、メダル数制御CPU946のROMは、メモリ容量に余裕(2.5Kバイト)があるので、プログラムをメダル数制御CPU946で実現しても問題ない。

【2033】

なお、ここでは、メインCPU500 aが、毎遊技、払出処理(S7)において払出終了時コマンドをメダル数制御CPU946に送信し、メダル数制御CPU946が、かかる払出終了時コマンドに応じて設定変更中信号をOFFしている(S12)。かかる構成では、メダル数制御CPU946が繰り返し設定変更中信号をOFFすることとなる。しかし、設定変更中信号はOFFのまま変わらず、一旦、OFFとなった設定変更中信号は、300msec周期で、適切に専用ユニット904に通知される。したがって、メダル数制御CPU946が繰り返し設定変更中信号をOFFすることは問題ない。ここでは、払出終了時コマンドをメダル数制御CPU946に送信する場合に、メインCPU500 aにおいて、設定変更処理が終了した最初の遊技であるという条件を判断する必要がなくなるので、設定変更処理が終了した最初の遊技という条件を判断するプログラムが不要となり、より、メインROM500 bのプログラムの増加を抑制することができる。

【2034】

なお、ここでは、メインCPU500 aが、払出処理(S7)において、メダル数制御CPU946にコマンドを送信する例を挙げて説明したが、1遊技が終了してさえすれば、リール停止処理(S5)や判定処理(S6)においてコマンドを送信してもよい。

【2035】

また、メダルレス遊技機902では、設定変更の他に、設定されている設定値を設定確認によって確認することができる。設定確認を試みる場合、まず、設定ドアキー438に

10

20

30

40

50

所定の操作キーが挿入されてOFFの位置からONの位置へ回転される。そうすると設定確認モードに移行して、設定値の確認が可能な状態となり、設定確認中ステータスがONされる。ここで、設定確認中ステータスは、設定確認が実行されている間ONであり、それ以外はOFFとなる。なお、設定確認中ステータス中に、ベットスイッチ416、精算スイッチ437、スタートスイッチ418の操作を無効としてもよい。そして、設定値が表示されるとともにステータスタイマの計時を開始する。ここで、ステータスタイマは、設定確認中ステータスを維持する時間を計時するタイマである。そして、操作キーを元の位置（OFFの位置）に戻すことで設定確認モードが終了して遊技が可能となる。なお、設定確認中ステータスは、設定ドアキー438がOFFであり、かつ、ステータスタイマが5秒以上経過するとOFFされる。

10

【2036】

また、設定確認モードにおいて、メインCPU500aは、設定確認中ステータスを専用ユニット904に伝達すべく、設定確認中信号を生成している。設定確認中信号は、設定確認中ステータスのON/OFFを表し、図248に示すように、遊技機情報通知におけるホールコン・不正監視情報（遊技機情報）の遊技機不正1におけるビット1に設定されており、図253に示したように、300msec周期で専用ユニット904に通知される。

【2037】

ここで、図242（a）および図242（b）で示したようにCPUを独立して2つ（メインCPU500aおよびメダル数制御CPU946）設置した場合において、メインCPU500aで設定確認中ステータスを管理すると、設定確認中信号を設定する処理のみならず、ステータスタイマで5秒を計時する処理を実行するプログラムが必要となり、使用領域の特に制御領域が圧迫されるおそれがある。そこで、設定変更モード同様、設定確認中信号を設定する処理およびステータスタイマで5秒を計時する処理を、メインCPU500aではなく、メダル数制御CPU946で実現する。

20

【2038】

図258は、設定確認中信号の更新処理を示したフローチャートである。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。図258に示す一連の処理を通じて1遊技が実行されている途中、前回の遊技の払出処理（S1）が終了し、電子化メダルのベット待ち状態において、所定の操作キーがONされることで、メインCPU500aは、設定値を確認する設定確認処理を開始する（S2）。そして、設定確認が完了すると、所定の操作キーがOFFされて、メインCPU500aは、設定確認処理を終了する（S3）。こうして、メインCPU500aは、ベット処理（S4）が可能となる。

30

【2039】

また、メインCPU500aは、設定確認処理の開始時（S2）に、設定確認の開始に応じて変更時コマンド（第3コマンド）をメダル数制御CPU946に送信する。メダル数制御CPU946は、かかる変更時コマンドを受信すると、図248に示したホールコン・不正監視情報の遊技機不正1におけるビット1を1に設定することで、設定確認中信号をONする（S11）。また、メインCPU500aは、設定確認処理の終了時（S3）に、変更時コマンド（第4コマンド）をメダル数制御CPU946に送信する。メダル数制御CPU946は、かかる変更時コマンドを受信すると、ステータスタイマに5秒を設定し、計時を開始する（S12）。そして、メダル数制御CPU946は、ステータスタイマが5秒経過したか否か判定し（S13）、5秒経過するまで（S13におけるNO）待機し、5秒経過すると（S13におけるYES）、ホールコン・不正監視情報の遊技機不正1におけるビット1を0に設定することで、設定確認中信号をOFFする（S14）。このように、設定値の確認の開始時に設定確認中信号をONして専用ユニット904に送信し、設定値の確認を完了した後、所定時間経過するのを待ち設定確認中信号をOFFして専用ユニット904に送信する機能部を設定確認中信号更新部と称する場合がある。なお、第3コマンドは、上記に限らず、設定値の確認の開始時に、その旨が特定されさ

40

50

えすれば、任意のコマンドとすることができ、第4コマンドは、上記に限らず、設定値の確認の完了時に、その旨が特定されさえすれば、任意のコマンドとすることができる。

【2040】

このように、メインCPU500aの代わりに、メダル数制御CPU946において、設定確認中信号を設定する処理およびステータスタイマで5秒を計時する処理を実現することで、メインROM500bのプログラムの増加を抑制できるので、使用領域の特に制御領域が圧迫されることがない。

【2041】

なお、上述したように、設定確認中信号は、設定確認モードに滞在していることを示しており、ゴト対策として少なくとも3秒間は出力しなければならない。しかし、設定確認中信号を含む遊技機情報通知は、300msec周期でしか専用ユニット904に送信されない。そうすると、例えば、メダル数制御CPU946が、遊技機情報通知が送信された直後に、変更時コマンドを受信した場合、設定確認中信号のONからOFFの間の時間が300msec程度削減されるおそれがある。ここでは、ステータスタイマでの待機時間を5秒とすることで、遊技機情報通知の送信タイミングに拘わらず、設定確認中信号のONからOFFの間の時間を3秒以上とすることができる。また、5秒の計時開始を、設定確認の開始を示すコマンドの受信時とせず、設定確認の終了を示すコマンドの受信時としているので、5秒に、さらに、設定確認の開始を示すコマンドから設定確認の終了を示すコマンドまでの間の時間が加算されるので、設定確認中信号のONからOFFの間の時間を確実に3秒以上とすることができる。

【2042】

(リプレイ役の入賞)

また、ここでは、設定変更中信号や設定確認中信号をメインCPU500aではなく、メダル数制御CPU946において生成することで、メインCPU500aの処理負荷およびメインROM500bのメモリ容量を削減した。しかし、かかる場合に限らず、メインCPU500aからメダル数制御CPU946への様々な情報を最小限に留め、専用ユニット904への通知を要する情報はメダル数制御CPU946で生成してもよい。かかる構成によっても、メインCPU500aの負担を軽減することができる。例えば、メダルレス遊技機902は、300msec毎に、図248で示した、遊技機情報通知におけるホールコン・不正監視情報と、遊技情報とを専用ユニット904に送信しなければならない。ここで、例えば、遊技結果がリプレイ役の入賞の場合、メインCPU500aは、再遊技フラグと払出枚数「0」という情報のみをメダル数制御CPU946に送信し、メダル数制御CPU946は、専用ユニット904に必要な各情報、例えば、スタートスイッチ418操作時には、投入メダル数として再遊技作動時の規定数（その遊技でベットした規定数）を生成し、遊技終了時には、払出メダル数として規定数を生成し、専用ユニット904に通知する。かかる構成により、メインCPU500aの処理負荷およびメインROM500bのメモリ容量を削減することができる。

【2043】

また、このとき、メインCPU500aがメダル数制御CPU946へ送信する再遊技フラグを、1バイトのコマンドではなく、識別情報の所定のビット「再遊技状態」（1ビット）に割り当てることで、再遊技状態という1バイトのコマンドを準備する必要がなくなり、コマンドの集約化を図るとともに、処理負荷を軽減することが可能となる。

【2044】

(メダル数制御CPUとの通信制御)

図242(a)および図242(b)で示したようにCPUを独立して2つ（メインCPU500aおよびメダル数制御CPU946）設置した場合、メインCPU500aとメダル数制御CPU946との間でシリアル通信が実行される。例えば、メインCPU500aは、メダル数制御CPU946にシリアル通信を通じて所定のコマンドを送信する

。そして、メダル数制御CPU946は、専用ユニット904に、図245～図253に示したような遊技機情報通知を送信する。

【2045】

図259は、メインCPU500aとメダル数制御CPU946との通信処理の比較例を示した説明図である。ここでは、メインCPU500aからメダル数制御CPU946にチェックサムを含む5バイト（固定長）のコマンドを送信する例を挙げて説明する。

【2046】

図259に示すように、主制御基板500には、シリアル送信用の送信FIFO950（特定FIFO）が設けられている。送信FIFO950は、64バイトのメモリ容量を有し、最先で入力された1バイトのデータを保持し、送信可能タイミング（例えば、80μsec間隔）においてその1バイトのデータを最先に出力するバッファ機能を有する。また、メインCPU500aに対応するメインRAM500cには、4バイトのコマンド（データ）を保持する送信候補バッファ952が設けられている。

【2047】

メインCPU500aでは、メダル数制御CPU946に送信するコマンド（チェックサムを除く4バイト）が生じると、まず、送信候補バッファ952に既にコマンドが保持されているか否か判定する。ここで、コマンドが保持されていれば、メインCPU500aは、送信候補バッファ952が空くのを待つ。そして、コマンドが保持されていなければ、図259（a）のように、メインCPU500aは、送信候補バッファ952に、クロスハッチングで示した4バイトのコマンドを保持させる。続いて、メインCPU500aは、送信候補バッファ952にコマンドが保持されたことに応じて、送信FIFO950に10バイトの空きがあるか否か判定する。ここで、10バイトの空きとは、送信FIFO950の64バイトのバッファに保持されているデータが54（64－10）以下であり、少なくとも10バイト分のデータを保持できる状態をいう。なお、ここで、10バイトの空きを判定しているのは、チェックサムを含む5バイトのコマンドを2つ連続して送信する場合があるからである。ここで、送信FIFO950に10バイトの空きがなければ、メインCPU500aは、送信FIFO950が10バイト分空くのを待つ。そして、送信FIFO950に10バイトの空きがあれば、図259（b）のように、メインCPU500aは、送信FIFO950にコマンドを保持させる。

【2048】

また、メインCPU500aは、図259（c）に示すように、送信FIFO950にコマンドを保持させる際に、送信候補バッファ952に保持された4バイトのコマンドのチェックサムを計算し、4バイトのコマンドに加えて、送信FIFO950に、黒の塗りつぶしで示したチェックサムも保持させる。こうして、送信FIFO950に5バイトのコマンドが設定される。そして、送信FIFO950は、図259（d）に示すように、80μsec間隔で5バイトのコマンドをメダル数制御CPU946に送信する。

【2049】

図259では、コマンドが固定長の比較例を説明したが、コマンドが可変長の場合もある。例えば、メインCPU500aは、例えば、3バイト～80バイトの可変長のコマンドをメダル数制御CPU946に送信する。このように、コマンドが可変長であると、以下の問題が生じる。例えば、3バイトのコマンドであれば、4バイトの送信候補バッファ952や、64バイトの送信FIFO950に一度に保持させることができるが、80バイトのコマンドはいずれにも一度に保持させることができない。そこで、送信するコマンドを複数に区分し、少ないバイト数（例えば1バイト）単位でコマンドを送信する。また、ここでは、送信候補バッファ952の構成を削除し、チェックサムの生成処理を変更する。

【2050】

図260は、コマンドの1バイト送信の概念を示したフローチャートとコマンドを示した図である。かかる図におけるステップSの数値は、本図の説明においてのみ用いることとする。

【2051】

図260(a)に示すように、メインCPU500aは、まず、送信FIFO950の情報を取得する(S1)。具体的に、図260(b)の1行目のコマンド「PUSH AF」によって、AFレジスタの値がスタック領域に退避され、3行目のコマンド「IN A, (@SOST___)」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@SOST___」を下位1バイトとし、そのアドレスで示される乗算結果レジスタの値をAレジスタに読み出す。こうして、「@SOST___」から送信FIFO950の情報が取得される。続いて、図260(a)に示すように、メインCPU500aは、送信FIFO950に1バイト以上の空きがあるか否かを判定し(S2)、送信FIFO950に1バイト以上の空きが生じるまで(S2におけるNO)、ステップS1の処理を繰り返す。具体的に、図260(b)の4行目のコマンド「JBIT Z, 6, A, MCD__SND01」によって、Aレジスタのビット6が0であれば、すなわち、送信FIFO950に1バイト以上の空きがなければ、2行目の指標「MCD__SND01」からの処理を繰り返し、ビット6が0でなければ、次の5行目のコマンドを実行する。

10

【2052】

続いて、図260(a)に示すように、送信FIFO950に1バイト以上の空きがあれば(S2におけるYES)、メインCPU500aは、送信FIFO950に1バイト分のデータを送信する(S3)。具体的に、図260(b)の5行目のコマンド「POP AF」によってスタック領域に退避されていたデータがAFレジスタに復帰される。6行目のコマンド「OUT (@SODT___), A」によって、Uレジスタの値をアドレスの上位1バイトとし、アドレス「@SODT___」を下位1バイトとし、そのアドレスで示される送信FIFO950に復帰されたAレジスタの値を出力する。続いて、図260(a)に示すように、1バイト分のチェックサムを更新する(S4)。具体的に図260(b)の7行目のコマンド「ADD A, E」によって、Aレジスタの値に、チェックサムを保持するEレジスタの値を加算し、8行目のコマンド「LD E, A」によって、加算結果(Aレジスタの値)をEレジスタに戻し、9行目のコマンド「RET」によって、1段上のルーチンに戻る。

20

【2053】

図261は、メインCPU500aとメダル数制御CPU946との通信処理を示した説明図である。ここでは、メインCPU500aからメダル数制御CPU946に可変長のコマンドのうち80バイトのコマンドを送信する例を挙げて説明する。

30

【2054】

図261に示すように、メインCPU500aには、シリアル送信用の送信FIFO950が設けられている。送信FIFO950は、図259同様、64バイトのメモリ容量を有し、最先で入力された1バイトのデータを保持し、送信可能タイミング(例えば、80μsec間隔)においてその1バイトのデータを最先に出力するバッファ機能を有する。また、メインCPU500aに属する(対応する)メインRAM500cには、1バイトのチェックサムを保持するチェックサムバッファ954が設けられている。

【2055】

メインCPU500aでは、メダル数制御CPU946に送信するコマンド(チェックサムを除く76バイト)が生じると、チェックサムバッファ954をクリアし(0に設定し)、送信FIFO950に1バイトの空きがあるか否かを判定する。ここで、送信FIFO950に1バイトの空きがなければ、メインCPU500aは、送信FIFO950が1バイト分空くのを待つ。そして、送信FIFO950に1バイトの空きがあれば、図261(a)のように、メインCPU500aは、送信FIFO950にコマンドの先頭に位置する1バイトのデータを保持させる。

40

【2056】

また、メインCPU500aは、送信FIFO950に1バイトのデータを保持させる際に、チェックサムバッファ954のデータと、当該1バイトのデータとのチェックサムを計算し、黒の塗りつぶしで示したチェックサム(計算結果)をチェックサムバッファ9

50

54に上書きする。こうして、1バイトのデータを送信する毎にチェックサムが更新される。

【2057】

そして、図261(b)のように、メインCPU500aは、送信FIFO950に1バイトの空きがあれば、送信FIFO950に、コマンドを1バイトずつ保持させ、送信FIFO950は、80 μ sec間隔で1バイトずつコマンドをメダル数制御CPU946に送信する。このように、メインCPU500aは、送信FIFO950に1バイト以上の空きがあれば、1バイトずつ区分されたコマンドを、送信FIFO950に順次保持させる。ただし、上述したように送信するコマンドは、送信FIFO950の容量よりバイト数が大きいので、図261(c)のように、送信FIFO950において空きがない状態が生じ得る。この場合、メインCPU500aは、送信FIFO950が1バイト分空くのを待つこととなる。

10

【2058】

メインCPU500aは、コマンドを全て送信FIFO950に保持させると、図261(d)のように、チェックサムバッファ954に保持されたチェックサムを送信FIFO950に保持させる。こうして、送信FIFO950に80バイトのコマンドが設定される。そして、送信FIFO950は、80 μ sec間隔で80バイトのコマンドをメダル数制御CPU946に送信する。

【2059】

このように、コマンドを区分して送信FIFO950に保持させる構成により、コマンドの長さに拘わらず、仮に、コマンドの長さが送信FIFO950の容量より大きかったとしても、メインCPU500aは、メダル数制御CPU946に、適切にコマンドを送信することが可能となる。

20

【2060】

また、コマンドのバイト数に拘わらず、例えば、コマンドが3バイトであっても、メインCPU500aが画一的にコマンドを区分して送信FIFO950に保持させる構成により、プログラムを単純化することができ、メインROM500bのプログラムの増加を抑制できる。したがって、使用領域の特に制御領域が圧迫されることがない。また、図259の構成と異なり、送信候補バッファ952を削減しているので、メインRAM500cの容量も確保することができる。

30

【2061】

さらに、ここでは、メインCPU500aが、送信FIFO950に1バイトのデータを保持させる度に、画一的にチェックサムを更新しているので、コマンドの送信完了時点でチェックサムの計算が完了している。したがって、複数のデータのチェックサムを一度に計算する別途のプログラムが不要となり、使用領域の特に制御領域が圧迫されることがないのみならず、その処理時間を削減することが可能となる。

【2062】

なお、ここでは、メインCPU500aが、コマンドを1バイトずつ区分して送信FIFO950に保持させる例を挙げて説明したが、かかる場合に限らず、例えば、2バイトずつ区分するといったように、コマンドの最大長さ未満の所定のバイト単位で区分することができる。ここで、区分する単位がコマンドの長さの公約数とならない場合、メインCPU500aは、区分した長さでコマンドを繰り返し送信し、最後の送信で余りのデータを送信する。例えば、コマンドの長さが80バイトであり、バイト単位が5バイトであれば、メインCPU500aは、コマンドを5バイトずつ区分して繰り返し(16回)送信FIFO950に保持させる。

40

【2063】

また、ここでは、メダルレス遊技機902において、メインCPU500a(第1CPU)からメダル数制御CPU946(第2CPU)にコマンドを送信する例を挙げて説明したが、かかる場合に限らず、例えば、メダルレス遊技機902内の他の基板同士や、管理遊技機におけるメインCPU300aから枠制御基板のCPUにコマンドを送信する場

50

合等、第1CPUや第2CPUとして、独立した様々なCPU同士のシリアル通信に適用することができる。

【2064】

また、ここでは、送信FIFO950の容量が64バイトの例を挙げて説明したが、かかる場合に限らず、任意の容量を適用できる。本実施形態は、特に、送信FIFO950の容量よりコマンドが長い場合に効果が高い。また、ここでは、Z80系CPUをベースとするエルイーテック（LEITECH）社が販売するマイクロプロセッサのLEM50Aを採用しており、その送信FIFO950の容量が64バイトであるが、その他の例えばLC701Aであっても、送信FIFO950の容量が64バイトで動作するモードがあり、例えば、LEM50AのプログラムをLC701Aに流用する場合にそのモードが選

10

【2065】

（メインCPU500aのコマンド1）

また、上記のように、ここでは、メインCPU500aがメダル数制御CPU946へ送信するコマンドを可変長としたとする。そうすると、コマンドを送信開始してから、送信完了するまでの時間が異なることとなる。ここで、仮に、電断が生じた場合、その電断処理でコマンドの送信が完了しない場合が生じ得る。コマンドには、送信が完了しなくとも、他の処理により補完できるものもあれば、送信完了を必須とするものもある。例えば、1遊技終了時コマンドは、コマンドの送信が完了せず、そのコマンドが伝達されなくとも、復電時（起動時）に再送されるため問題は生じない。したがって、コマンドの送信完了を待つことなく、電断処理を例えば即座に開始することができる。かかる構成により、不要なコマンドの送信を回避でき、割込禁止時間の増加を抑制することが可能となる。

20

【2066】

一方、払出終了時コマンドでは、払出枚数を示す伝達情報が送信される。払出枚数は、その送信を複数回行うべきではない。これは、遊技終了時に払出枚数を送信した後、さらに復電時（起動時）に再送されると、複数回払い出されたこととなり、払出枚数が多く（2倍）になってしまうからである。また、払出枚数は、遊技者の利益に直接関係する重要度の高い情報なので、確実にメダル数制御CPU946へ伝達する必要がある。そこで、メインCPU500aがメダル数制御CPU946へ送信する複数のコマンドのうち、払出終了時コマンドのように、重要度の高いコマンドについては、電断予告信号の検出から電断処理開始までの間に確実にメダル数制御CPU946へコマンドを送信する。すなわち、コマンドの送信が完了してから電断処理を開始する。なお、電断処理が開始されるまでの短時間にコマンドの送信を完了しなければならないので、重要度の高いコマンドのバイト数を短く（例えば10バイトと）する。かかる構成により、メインCPU500aは、重要度の高いコマンドを確実にメダル数制御CPU946へ送信し、遊技を適切に進行することが可能となる。

30

【2067】

（メインCPU500aのコマンド2）

また、上記のように、ここでは、メインCPU500aがメダル数制御CPU946へ送信するコマンドを可変長としたとする。そうすると、コマンドの送信順やその内容によっては、送信開始してから、送信完了するまでの時間が異なることとなる。例えば、比較的長い（例えば、60バイトの）1遊技終了時コマンドが送信された直後に電断予告信号が生じると、コマンドの送信が完了することなく、電断が生じるおそれがある。また、電源を投入すると、メインCPU500aは、1遊技終了時コマンド以上に長い（例えば、80バイトの）起動時コマンドをメダル数制御CPU946へ送信し、電断が生じる前の状態から処理を再開する。ここで、1遊技が終了すると、役比関連のデータが更新され、起動時コマンドの後に、1遊技終了時コマンドが送信される。そうすると、未だ、送信FIFO950に起動時コマンドが残っている可能性が高いので、さらに1遊技終了時コマンドの送信が待たされる。このような1遊技終了時コマンドが送信された直後に電断予告

40

50

信号が生じると、さらに、コマンドの送信完了が遅延し、コマンドの送信完了を待たずに電断が生じるおそれがある。

【2068】

ただし、上述したように、1遊技終了時コマンドがメダル数制御CPU946に伝達されなくとも、復電時（起動時）に再送されるため問題は生じない。なお、復電時（起動時）は、送信FIFO950にデータが保持されていないので、電断が生じない限り、起動時コマンドは適切にメダル数制御CPU946に伝達される。

【2069】

そこで、1遊技終了時コマンドのように、コマンドに「役比関連の伝達情報」を含む場合、電断予告信号を検出したら、コマンドの送信完了を待つことなく、即座に電断処理を開始する。例えば、メインCPU500aは、図260（a）のステップS2において、送信FIFO950に1バイト以上の空きがある場合、ステップS3において、送信FIFO950に1バイトのデータを送信する前に、コマンドに「役比関連の伝達情報」が含まれ、かつ、電断予告信号が検出されたか否かを判定する。そして、電断予告信号が検出されると、メインCPU500aは、以降のステップS3、S4を行うことなく、すなわち、1バイトのデータを送信することなく（チェックサムを含む1遊技終了時コマンドの送信をキャンセルして）、当該処理を終了し、電断処理に移行する。かかる構成により、不要なコマンドの送信を回避でき、割込禁止時間の増加を抑制することが可能となる。

【2070】

一方、コマンドに「役比関連の伝達情報」が含まれていない、または、電断予告信号が検知されていなければ、メインCPU500aは、ステップS3、S4を実行する。かかる構成により、重要度の低い1遊技終了時コマンドの送信中に電断予告信号が検出された場合、早急にコマンドの送信をキャンセルし、適切に電断処理を実行することが可能となる。

【2071】

また、仮に、電源を投入し、起動時コマンドが送信され、電断が生じる前の状態から処理が再開し、直後に、1遊技終了時コマンドが送信され、その直後に電断予告信号が検出された場合を仮定する。この場合、電源投入時には送信FIFO950にデータが保持されていないので、起動時コマンドのうち、前半64バイト分は直ぐに送信FIFO950に保持される。その後、1遊技終了時コマンドを送信する際には、起動時コマンドの後半部分が送信FIFO950にまだ残っている。したがって、コマンドが送信完了するまで長時間を要する。しかし、上記の電断予告信号が検出されたか否かを判定する構成により、送信FIFO950に空きが生じたときには、1遊技終了時コマンドの送信がキャンセルされるので、適切に電断処理を実行することが可能となる。

【2072】

なお、「役比関連の伝達情報」は、1遊技終了時コマンドのみならず、起動時コマンドにも含まれている。したがって、起動時コマンドを送信する際に電断予告信号が検出されると、それ以降のデータを送信することなく（チェックサムを含む起動時コマンドの送信をキャンセルして）、当該処理を終了し、電断処理に移行することとなる。

【2073】

ただし、起動時コマンドのうち「起動時に必要な伝達情報」の送信は停止せず、送信が完了するまで電断処理を行わない。これは以下の理由による。すなわち、「起動時に必要な伝達情報」は、遊技機情報、チップID、メカコード等を含み、電断予告信号の確認も複数箇所に分けて記載する必要があり、処理負荷およびメモリ容量の増大を招くおそれがある。また、上述したように、起動時には送信FIFO950にはデータが保持されていないので、「起動時に必要な伝達情報」までは、短時間で送信FIFO950に保持させることができ、かつ、送信完了までに長時間を要さないからである。

【2074】

ここでは、1遊技終了時コマンド、および、起動時コマンドについて、コマンドに「役比関連の伝達情報」が含まれており、かつ、電断予告信号が検知されていれば、それ以降

10

20

30

40

50

のデータを送信することなく、即座に電断処理に移行する例を挙げて説明した。その他のコマンド（待機中コマンド、スタート時コマンド、払出終了時コマンド、変更時コマンド）については、図260（a）の処理をそのまま実行してもよいし、図260（a）のステップS1の前、すなわち送信開始時に、電断予告信号が検知されているか否か判定し、検出されていれば、ステップS1～S4を全て行うことなく、すなわち、それ以降のデータを送信することなく、即座に電断処理に移行するとしてもよい。かかる構成により、全てのコマンドについて適切に電断処理に移行することとなる。

【2075】

以上、添付図面を参照しながら本発明の好適な実施形態について説明したが、本発明はかかる実施形態に限定されないことは言うまでもない。当業者であれば、特許請求の範囲に記載された範疇において、各種の変更例または修正例に想到し得ることは明らかであり、それらについても当然に本発明の技術的範囲に属するものと了解される。

【2076】

また、上述した実施形態では、主制御基板500と副制御基板502とメダル数制御基板920が、遊技を進行するための機能部を分担するように配したが、主制御基板500の一部の機能部を副制御基板502やメダル数制御基板920に配しても、副制御基板502の一部の機能部を主制御基板500やメダル数制御基板920に配してもよく、メダル数制御基板920の一部の機能部を主制御基板500や副制御基板502に配してもよく、また、全ての機能部を1の制御基板に纏めて配することもできる。

【2077】

また、上述した主制御基板300、500、副制御基板330、502、メダル数制御基板920が行う各処理は、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいはサブルーチンによる処理を含んでもよい。

【符号の説明】

【2078】

100 遊技機

300、500 主制御基板

300a、500a メインCPU

300b、500b メインROM

300c、500c メインRAM

330、502 副制御基板

330a、502a サブCPU

330b、502b サブROM

330c、502c、サブRAM

400 スロットマシン

900 遊技システム

902 メダルレス遊技機

904 専用ユニット

920 メダル数制御基板

946 メダル数制御CPU

950 送信FIFO

954 チェックサムバッファ

10

20

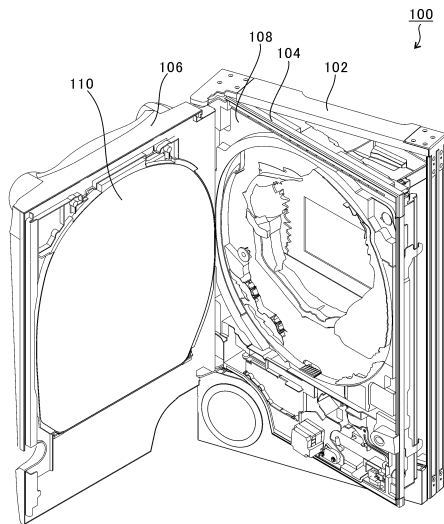
30

40

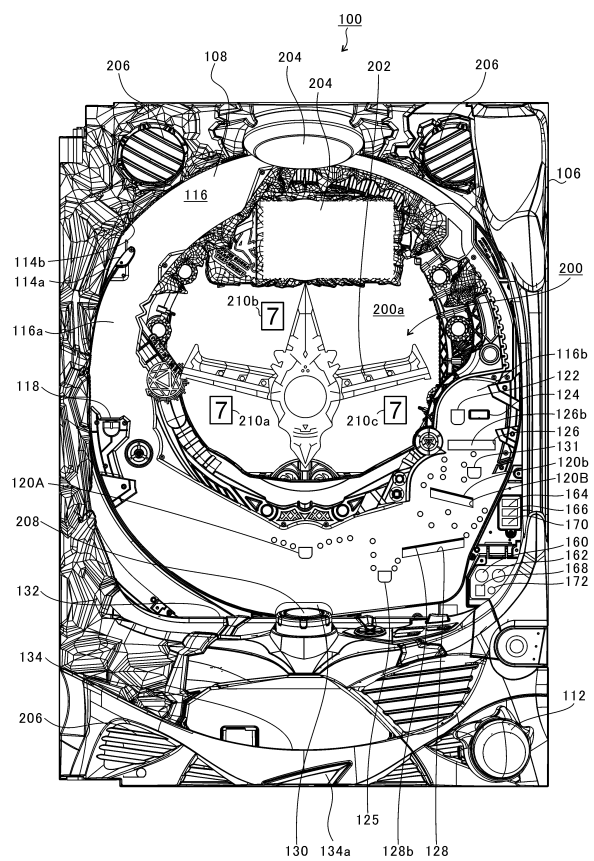
50

【図面】

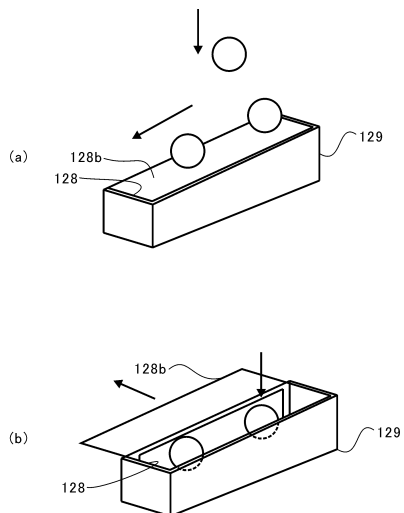
【図 1】



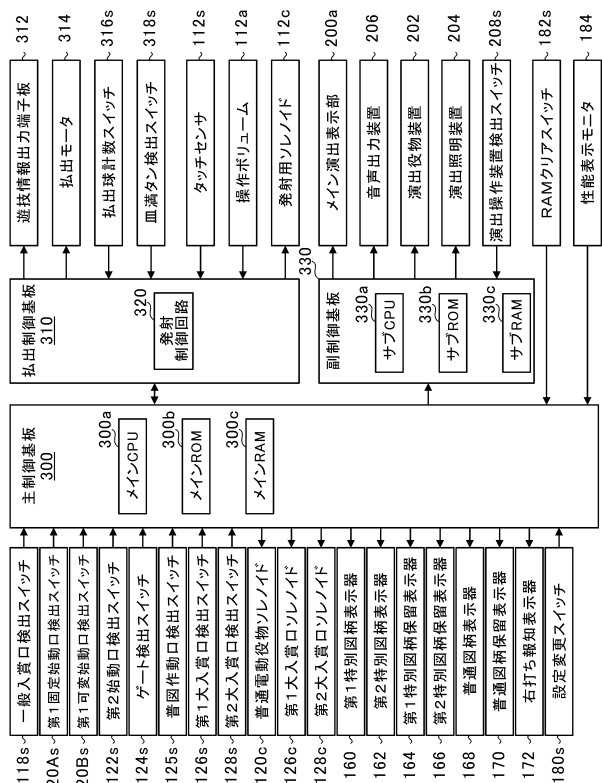
【図 2】



【圖 3】



【図 4】



【図 5】

アドレス	内容	ハードウェア	
0000H～0A89H	プログラム領域	使用領域	メインROM
0A8AH～11FFH	未使用領域		
1200H～1BF3H	データ領域		
1BF4H～1DFFH	未使用領域		
1E00H～1EFFH	ROMコメント領域		
1F00H～1FFFH	未使用領域	使用外領域	
2000H～27FFH	プログラム領域		
2800H～2BFFH	データ領域		
2C00H～2BFFH	未使用領域		
2FC0H～2FFFFH	プログラム管理領域		
3000H～EFFFFH	なし		
F000H～F12AH	ワーク領域	使用領域	メインRAM
F12BH～F1D7H	未使用領域		
F1D8H～F1FFH	スタック領域		
F200H～F20FH	未使用領域		
F210H～F21FH	ワーク領域	使用外領域	
F220H～F228H	スタック領域		
F229H～F3FFH	未使用領域		
F400H～FDFFH	なし		
FE00H～FEFFH	入出力部		
FF00H～FFFFH	なし		

【図 6】

(a) 低確時大当たり決定乱数判定テーブルa(設定値＝1用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10218	大当たり	$\approx 1/300.6$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(b) 低確時大当たり決定乱数判定テーブルb(設定値＝2用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10225	大当たり	$\approx 1/291.2$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(c) 低確時大当たり決定乱数判定テーブルc(設定値＝3用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10232	大当たり	$\approx 1/282.4$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(d) 低確時大当たり決定乱数判定テーブルd(設定値＝4用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10239	大当たり	$\approx 1/274.2$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(e) 低確時大当たり決定乱数判定テーブルe(設定値＝5用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10246	大当たり	$\approx 1/266.4$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(f) 低確時大当たり決定乱数判定テーブルf(設定値＝6用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10253	大当たり	$\approx 1/259.0$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

【図 7】

(a) 高確時大当たり決定乱数判定テーブルa(設定値＝1用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10620	大当たり	$\approx 1/105.7$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(b) 高確時大当たり決定乱数判定テーブルb(設定値＝2用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10640	大当たり	$\approx 1/102.4$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(c) 高確時大当たり決定乱数判定テーブルc(設定値＝3用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10660	大当たり	$\approx 1/99.3$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(d) 高確時大当たり決定乱数判定テーブルd(設定値＝4用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10680	大当たり	$\approx 1/96.4$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(e) 高確時大当たり決定乱数判定テーブルe(設定値＝5用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10700	大当たり	$\approx 1/93.6$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

(f) 高確時大当たり決定乱数判定テーブルf(設定値＝6用)

大当たり決定乱数 (0～65535)	判定結果	確率
10001～10720	大当たり	$\approx 1/91.0$
20001～38996	小当たり	$\approx 1/3.45$
上記以外	ハズレ	—

【図 8】

(a) 特1用当たり図柄乱数判定テーブルa

当たり図柄乱数 (0～99)	特別図柄種別	選択確率
0～29	特別図柄A	30%
30～49	特別図柄B	20%
50～69	特別図柄C	20%
70～84	特別図柄D	15%
85～99	特別図柄E	15%

(b) 特2用当たり図柄乱数判定テーブルb

当たり図柄乱数 (0～99)	特別図柄種別	選択確率
0～29	特別図柄F	30%
30～49	特別図柄G	20%
50～69	特別図柄H	20%
70～84	特別図柄I	15%
85～99	特別図柄J	15%

(c) 特1用小当たり図柄乱数判定テーブルa

当たり図柄乱数 (0～99)	特別図柄種別	選択確率
0	特別図柄Z1	1%
1～69	特別図柄Z2	69%
70～99	特別図柄Z3	30%

(d) 特2用小当たり図柄乱数判定テーブルb

当たり図柄乱数 (0～99)	特別図柄種別	選択確率
0～19	特別図柄Z4	20%
20～39	特別図柄Z5	20%
40～99	特別図柄Z6	60%

10

20

30

40

50

【図 9】

(a) リーチグループ決定乱数判定テーブル1 (通常遊技状態用)

保留種別	大当たり判定結果	特1保留数	リーチグループ決定乱数 (0~10006)	グループ種別
特1保留	ハズレ	0	0~8999	グループx
			9000~9099	グループx
			9100~9299	グループx
			9300~9499	グループx
			9500~9699	グループx
			9700~9799	グループx
			9800~9849	グループx
			9850~9949	グループx
			9950~10006	グループx

(b) リーチグループ決定乱数判定テーブル2 (通常遊技状態用)

保留種別	大当たり判定結果	特1保留数	リーチグループ決定乱数 (0~10006)	グループ種別
特1保留	ハズレ	1、2	0~8999	グループx
			9000~9099	グループx
			9100~9299	グループx
			9300~9499	グループx
			9500~9699	グループx
			9700~9799	グループx
			9800~9849	グループx
			9850~9949	グループx
			9950~10006	グループx

(c) リーチグループ決定乱数判定テーブル3 (通常遊技状態用)

保留種別	大当たり判定結果	特1保留数	リーチグループ決定乱数 (0~10006)	グループ種別
特1保留	ハズレ	3	0~8999	グループx
			9000~9099	グループx
			9100~9299	グループx
			9300~9499	グループx
			9500~9699	グループx
			9700~9799	グループx
			9800~9849	グループx
			9850~9949	グループx
			9950~10006	グループx

【図 11】

(a) 変動パターン乱数判定テーブルx

変動パターン乱数 (0~238)	変動パターン番号
0~19	〇〇H
20~39	〇〇H
40~69	〇〇H
70~99	〇〇H
100~129	〇〇H
130~159	〇〇H
160~179	〇〇H
180~199	〇〇H
200~209	〇〇H
210~229	〇〇H
230~238	〇〇H

【図 10】

(a) グループx用ハズレ時リーチモード決定乱数判定テーブル

リーチモード決定乱数 (0~250)	変動モード番号	変動パターン乱数判定テーブル
0~99	〇〇H	テーブルx
100~199	〇〇H	テーブルx
200~250	〇〇H	テーブルx

(b) 特1用大当たり時リーチモード決定乱数判定テーブル

リーチモード決定乱数 (0~250)	変動モード番号	変動パターン乱数判定テーブル
0~49	〇〇H	テーブルx
50~199	〇〇H	テーブルx
200~250	〇〇H	テーブルx

(c) 特2用大当たり時リーチモード決定乱数判定テーブル

リーチモード決定乱数 (0~250)	変動モード番号	変動パターン乱数判定テーブル
0~29	〇〇H	テーブルx
30~99	〇〇H	テーブルx
100~149	〇〇H	テーブルx
150~199	〇〇H	テーブルx
200~250	〇〇H	テーブルx

(d) 特1用小当たり時リーチモード決定乱数判定テーブル

リーチモード決定乱数 (0~250)	変動モード番号	変動パターン乱数判定テーブル
0~99	〇〇H	テーブルx
100~199	〇〇H	テーブルx
200~250	〇〇H	テーブルx

(e) 特2用小当たり時リーチモード決定乱数判定テーブル

リーチモード決定乱数 (0~250)	変動モード番号	変動パターン乱数判定テーブル
0~250	〇〇H	テーブルx

【図 12】

(a) 変動時間1決定テーブル

変動モード番号	変動時間1
〇〇H	0秒
〇〇H	8秒
〇〇H	8秒
〇〇H	12秒
〇〇H	16秒
〇〇H	20秒
〇〇H	20秒
⋮	⋮
⋮	⋮

(b) 変動時間2決定テーブル

変動パターン番号	変動時間2
〇〇H	2秒
〇〇H	2秒
〇〇H	4秒
〇〇H	4秒
〇〇H	8秒
〇〇H	20秒
〇〇H	12秒
⋮	⋮
⋮	⋮

10

20

30

40

50

【図 1 3】

(a) 遊技状態と遊技状態名

特別遊技	普通遊技				遊技状態
	普通遊技状態	確率	変動時間	開放時間	
低確率遊技状態	非時短遊技状態	低	長	短	通常状態
高確率遊技状態	非時短遊技状態	低	長	短	最優位状態
低確率遊技状態	時短遊技状態	高	短	長	低確時短状態
高確率遊技状態	時短遊技状態	高	短	長	高確前兆状態
高確率遊技状態	中時短遊技状態	中	短	長	ペナルティ状態

(b) 遊技状態と図柄と変動時間

遊技状態	実質 変動対象	保留種別	図柄種別	変動時間
通常状態	特 1	特 1 保留	ハズレ	3～100秒
			小当たり	3～100秒
			大当たり	40～100秒
			ハズレ	10分
最優位状態	特 2	特 2 保留	ハズレ	10分
			小当たり	10分
			大当たり	10分
			ハズレ	10秒
低確時短状態 高確時短状態	特 1	特 1 保留	ハズレ	1秒～3秒
			小当たり	1秒～3秒
			大当たり	3秒～10秒
			ハズレ	1秒
高確前兆状態	特 1	特 1 保留	ハズレ	1秒
			小当たり	1秒
			大当たり	30秒
			ハズレ	10分
ペナルティ状態	特 1	特 1 保留	ハズレ	10分
			小当たり	10分
			大当たり	10分
			ハズレ	10分

【図 1 4】

特別図柄種別		特別図柄 A、B、D		特別図柄 C、E	特別図柄Z (Z1) (Z2) (Z3)		
オープニング時間		5. 0秒	5. 0秒	5. 0秒	10秒	0. 1秒	0. 1秒
開放大入賞口		第1 大入賞口		第1 大入賞口	第2 大入賞口		
特別電動役物最大作動回数 (ラウンド数)		4回		10回	1回		
特別電動役物 開閉切替回数 (ラウンド中開放回数)	1R	1回	1回	1回	1回		
	2R	1回	1回	1回	—		
	3R	1回	1回	1回	—		
	4R	1回	1回	1回	—		
	5R	—	1回	1回	—		
	6R	—	1回	1回	—		
	7R	—	1回	1回	—		
	8R	—	1回	1回	—		
	9R	—	1回	1回	—		
	10R	—	1回	1回	—		
ソレノイド通電時間 (1回の入賞口開放時間)	1R	29. 0秒	29. 0秒	29. 0秒	0. 1秒		
	2R	29. 0秒	29. 0秒	29. 0秒	—		
	3R	29. 0秒	29. 0秒	29. 0秒	—		
	4R	29. 0秒	29. 0秒	29. 0秒	—		
	5R	—	29. 0秒	29. 0秒	—		
	6R	—	29. 0秒	29. 0秒	—		
	7R	—	29. 0秒	29. 0秒	—		
	8R	—	29. 0秒	29. 0秒	—		
	9R	—	29. 0秒	29. 0秒	—		
	10R	—	29. 0秒	29. 0秒	—		
大入賞口の規定数 (1ラウンド中最大入賞可能数)		10個	10個	10個	10個		
大入賞口閉鎖有効時間 (ラウンド間のインターバル時間)		2. 0秒	2. 0秒	2. 0秒	0. 9秒		
エンディング時間		5. 0秒	5. 0秒	5. 0秒	0. 1秒	0. 1秒	0. 1秒

10

20

【図 1 5】

特別図柄種別	特別図柄 F、G、I		特別図柄Z (Z4) (Z5) (Z6)		
	5. 0秒	5. 0秒	0. 02秒	0. 02秒	0. 02秒
オープニング時間					
開放大入賞口	第1 大入賞口	第1 大入賞口	第2 大入賞口		
特別電動役物最大作動回数 (ラウンド数)	4回	10回	1回	1回	1回
特別電動役物 開閉切替回数 (ラウンド中開放回数)	1R	1回	1回	2回	3回
	2R	1回	1回	—	—
	3R	1回	1回	—	—
	4R	1回	1回	—	—
	5R	—	1回	—	—
	6R	—	1回	—	—
	7R	—	1回	—	—
	8R	—	1回	—	—
	9R	—	1回	—	—
	10R	—	1回	—	—
ソレノイド通電時間 (1回の入賞口開放時間)	1R	29. 0秒	29. 0秒	0. 1秒	0. 1秒
	2R	29. 0秒	29. 0秒	—	—
	3R	29. 0秒	29. 0秒	—	—
	4R	29. 0秒	29. 0秒	—	—
	5R	—	29. 0秒	—	—
	6R	—	29. 0秒	—	—
	7R	—	29. 0秒	—	—
	8R	—	29. 0秒	—	—
	9R	—	29. 0秒	—	—
	10R	—	29. 0秒	—	—
大入賞口の規定数 (1ラウンド中最大入賞可能数)	10個	10個	10個	10個	10個
大入賞口閉鎖有効時間 (ラウンド間のインターバル時間)	2. 0秒	2. 0秒	0. 9秒	0. 9秒	0. 9秒
ラウンド中インターバル時間	—	—	1. 76秒	0. 84秒	0. 84秒
エンディング時間	5. 0秒	5. 0秒	0. 1秒	0. 1秒	0. 1秒

【図 1 6】

大当たり当選時			保留 種別	特別 図柄 種別	大役遊技後				
遊技状態	特別 遊技状態	普通 遊技状態			遊技状態	特別 遊技状態	高確 回数	普通 遊技状態	時短 回数
通常	低確	非時短	特 1 保留	A	低確時短	低確	0	時短	100
				B	高確時短	高確	次回	時短	次回
				C	高確前兆	高確	次回	時短	100
				D	高確時短	高確	次回	時短	次回
				E	高確前兆	高確	次回	時短	100
ペナルティ	高確	中時短	特 2 保留	F	通常	低確	0	非時短	0
				G	ペナルティ	高確	次回	中時短	次回
				H	ペナルティ	高確	次回	中時短	次回
				I	ペナルティ	高確	次回	中時短	次回
				J	ペナルティ	高確	次回	中時短	次回
最優位	高確	非時短	特 1 保留	A	低確時短	低確	0	時短	100
				B	高確時短	高確	次回	時短	次回
				C	高確前兆	高確	次回	時短	100
				D	高確時短	高確	次回	時短	次回
				E	高確前兆	高確	次回	時短	100
低確時短	低確	時短	特 2 保留	F	低確時短	低確	0	時短	100
				G	高確時短	高確	次回	時短	次回
				H	高確前兆	高確	次回	時短	100
				I	高確時短	高確	次回	時短	次回
				J	高確前兆	高確	次回	時短	100
高確時短 高確前兆	高確	時短	特 1 保留	A	低確時短	低確	0	時短	100
				B	高確時短	高確	次回	時短	次回
				C	高確前兆	高確	次回	時短	100
				D	高確時短	高確	次回	時短	次回
				E	高確前兆	高確	次回	時短	100
高確時短 高確前兆	高確	時短	特 2 保留	F	通常	低確	0	非時短	0
				G	ペナルティ	高確	次回	中時短	次回
				H	ペナルティ	高確	次回	中時短	次回
				I	ペナルティ	高確	次回	中時短	次回
				J	ペナルティ	高確	次回	中時短	次回

30

40

50

【図 17】

(a) 非時短遊技状態用当たり決定乱数判定テーブル

当たり決定乱数 (0~99)	普通図柄種別	当選確率
0	当たり図柄	1/100
1~99	ハズレ図柄	

(b) 中時短遊技状態用当たり決定乱数判定テーブル

当たり決定乱数 (0~99)	普通図柄種別	当選確率
0~49	当たり図柄	50/100
50~99	ハズレ図柄	

(c) 時短遊技状態用当たり決定乱数判定テーブル

当たり決定乱数 (0~99)	普通図柄種別	当選確率
0~98	当たり図柄	99/100
99	ハズレ図柄	

【図 18】

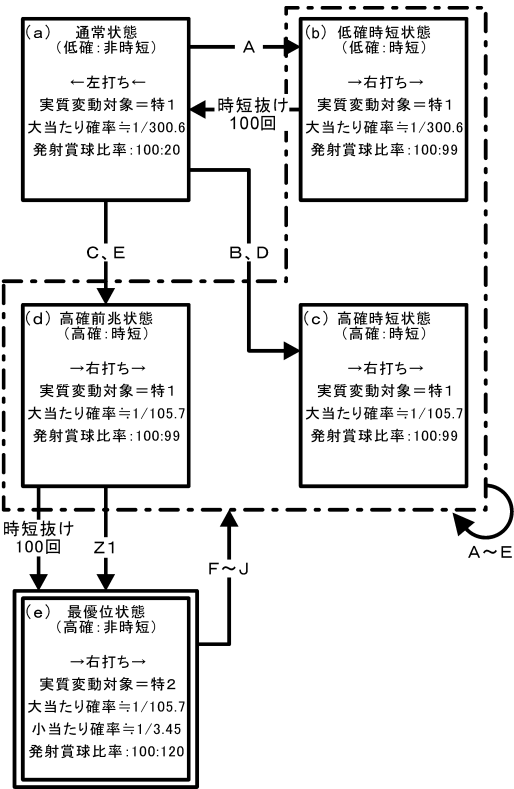
(a) 普通図柄変動時間データテーブル

遊技状態	変動時間
非時短遊技状態	10秒
中時短遊技状態	10秒
時短遊技状態	1秒

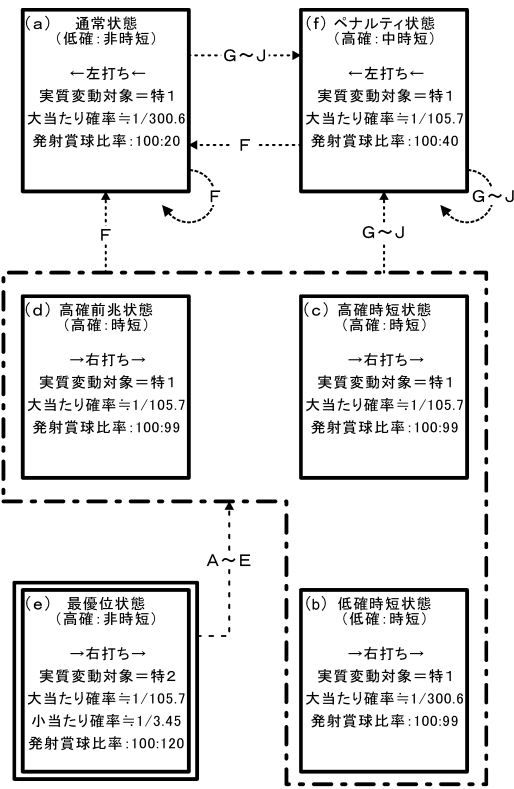
(b) 開閉制御パターンテーブル

遊技状態	非時短	中時短	時短
普通開放前時間	1.0秒	1.0秒	1.0秒
普通電動役物最大開閉切替回数	2回	2回	2回
ソレノイド通電時間 (1回の開放時間)	1回目	0.1秒	2.9秒
	2回目	0.1秒	2.9秒
規定数 (全開放中の最大入賞可能数)	8個	8個	8個
普通閉鎖有効時間 (停止時間)	1.0秒	1.0秒	1.0秒
普通有効状態時間	0.1秒	0.1秒	0.1秒
普通終了ウェイト時間	0.5秒	0.5秒	0.5秒
発射賞球比率	100:20	100:40	100:99

【図 19】



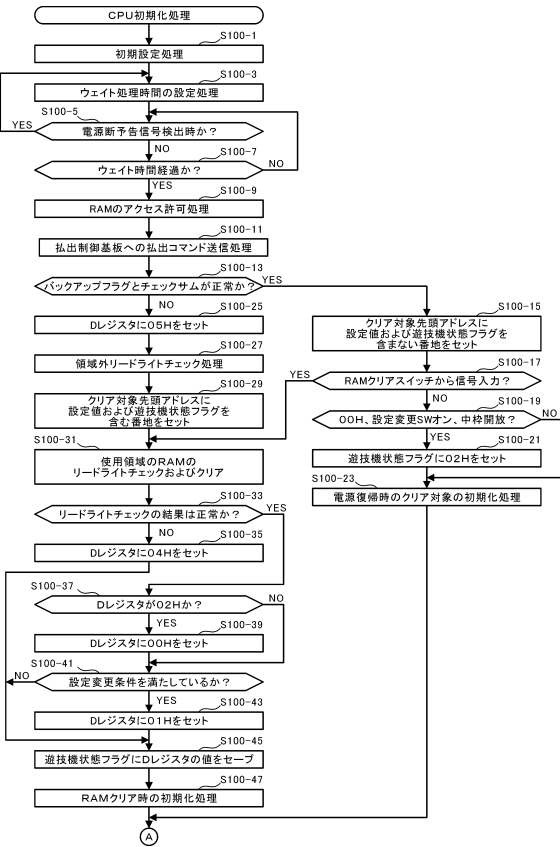
【図 20】



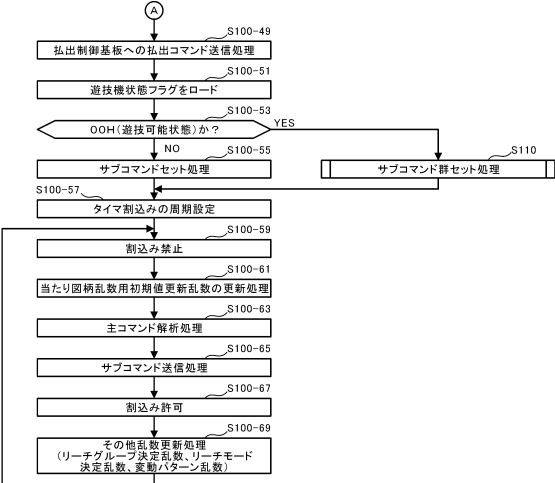
【図 2 1】

遊技機状態フラグ	
00H	遊技可能状態
01H	設定変更状態
02H	設定確認状態
03H	設定異常状態
04H	RAM異常状態
05H	チェックサム異常状態

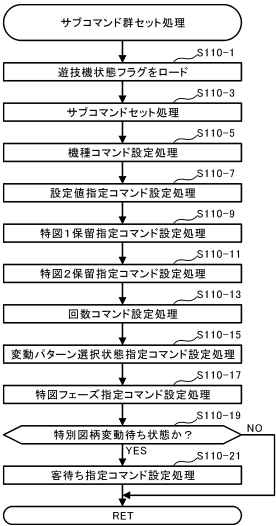
【図 2 2】



【図 2 3】



【図 2 4】



10

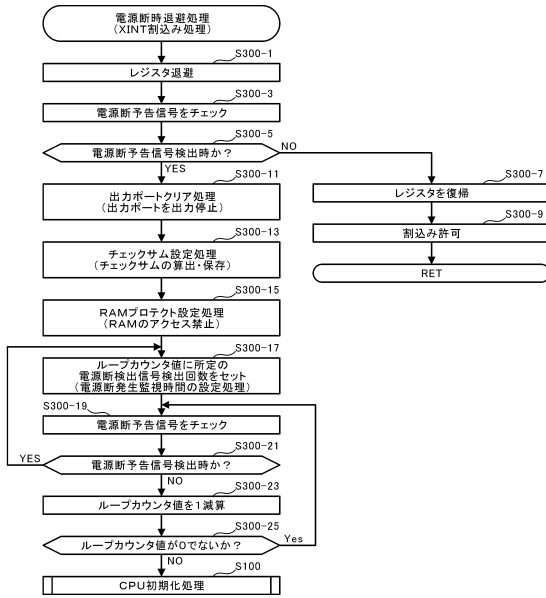
20

30

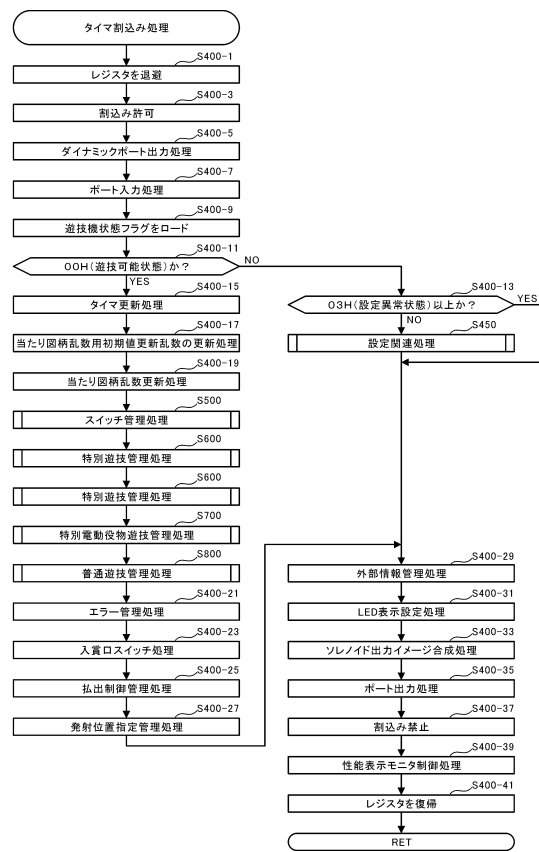
40

50

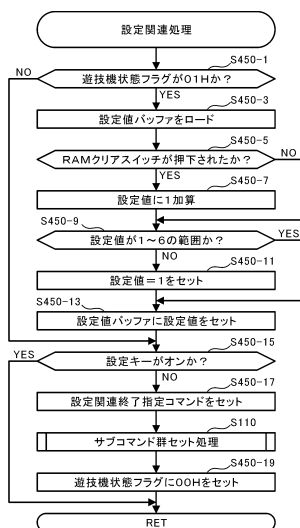
【図 25】



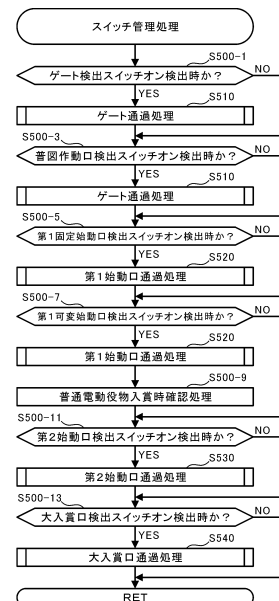
【図 26】



【図 27】



【図 28】



10

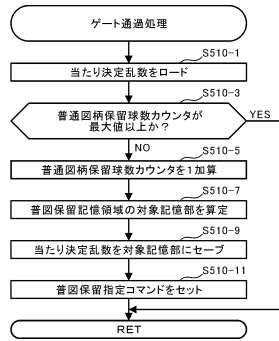
20

30

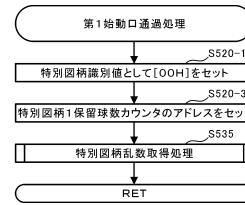
40

50

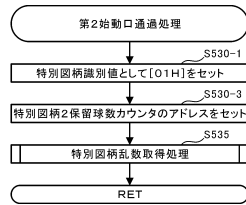
【図 29】



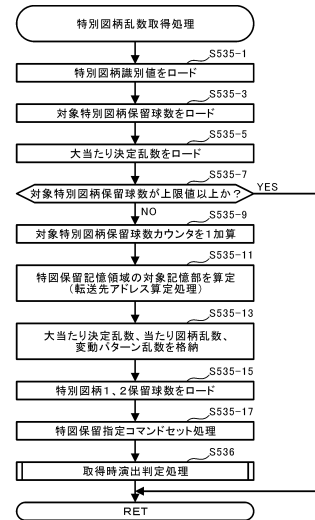
【図 30】



【図 31】



【図 32】



10

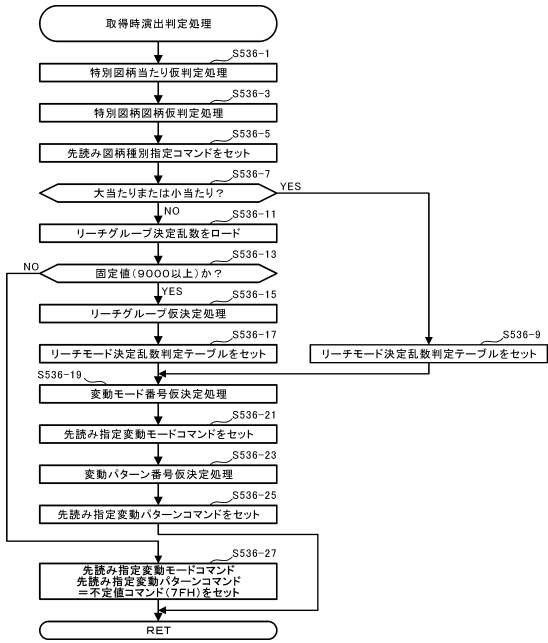
20

30

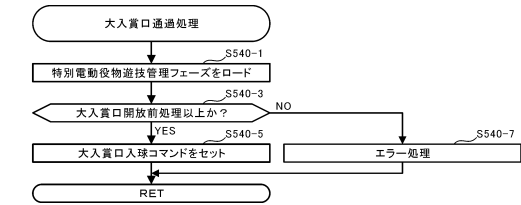
40

50

【図 3 3】



【図 3 4】

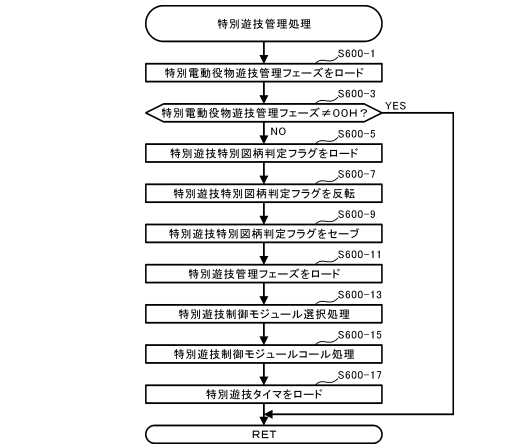


【図 3 5】

特別遊技管理フェーズ	特別遊技制御モジュール
00H	特別図柄変動待ち処理
01H	特別図柄変動中処理
02H	特別図柄停止図柄表示処理

特別電動役物遊技管理フェーズ	特別電動役物遊技制御モジュール
00H	—
01H、05H	大入賞口開放前処理
02H、06H	大入賞口開放制御処理
03H、07H	大入賞口閉鎖有効処理
04H、08H	大入賞口終了ウェイト処理

【図 3 6】



10

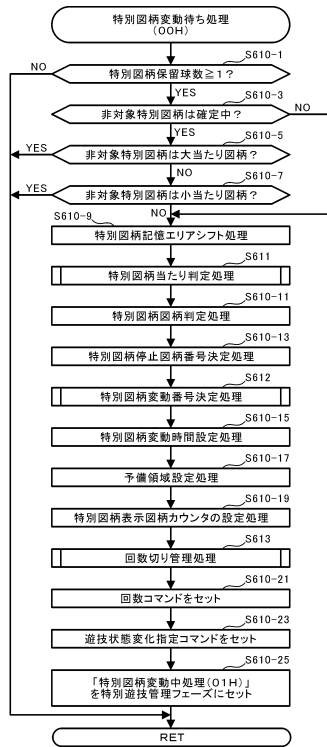
20

30

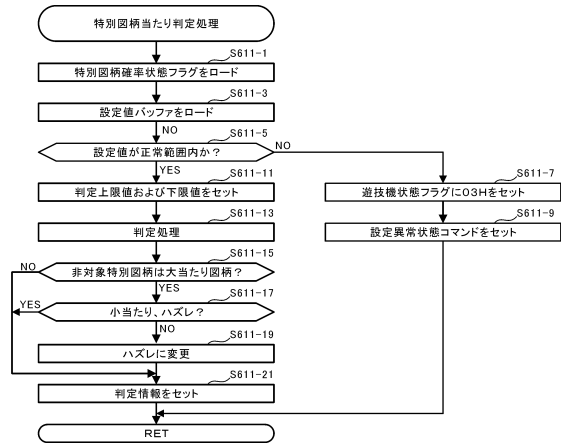
40

50

【図 37】



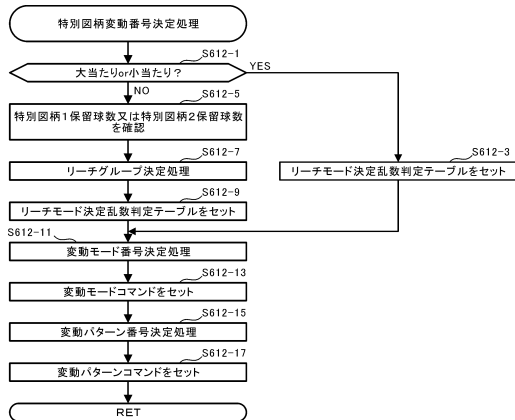
【図 38】



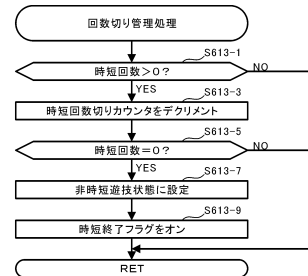
10

20

【図 39】



【図 40】

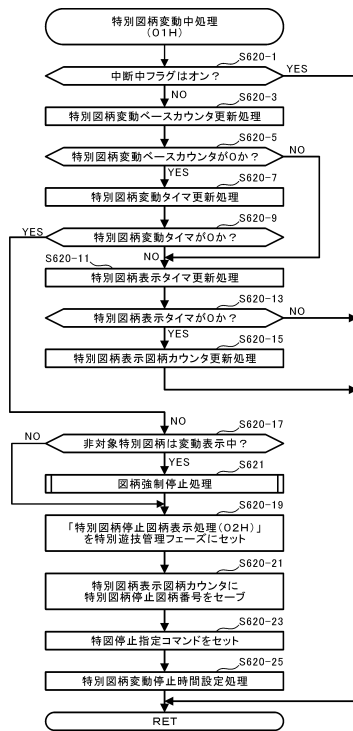


30

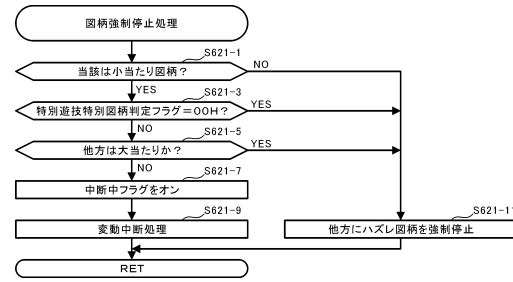
40

50

【図 4 1】



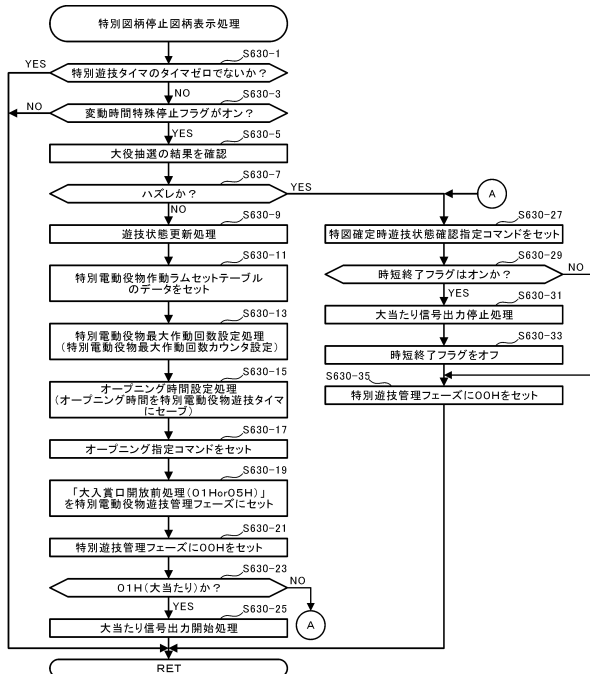
【図 4 2】



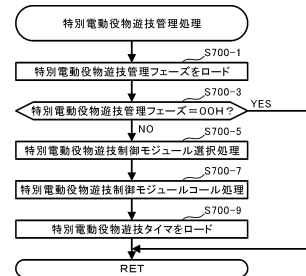
10

20

【図 4 3】



【図 4 4】

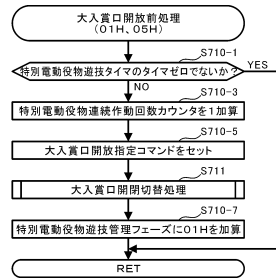


30

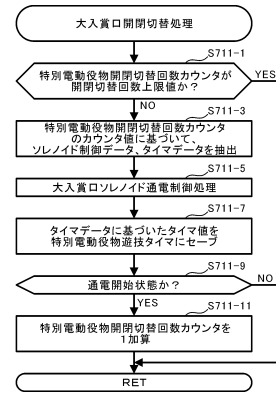
40

50

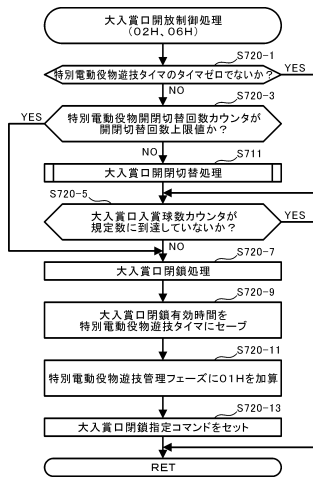
【図 45】



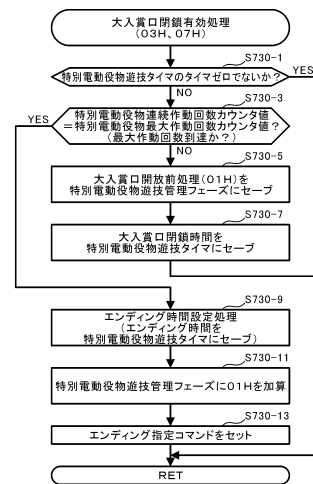
【図 46】



【図 47】



【図 48】



10

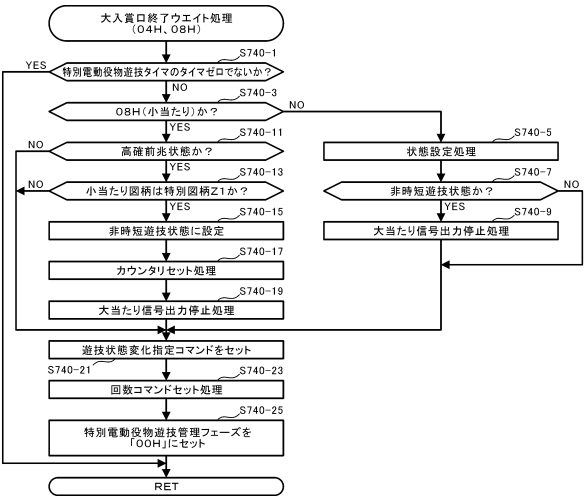
20

30

40

50

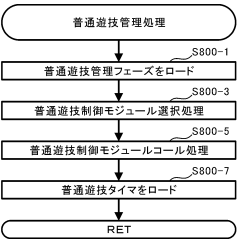
【図 49】



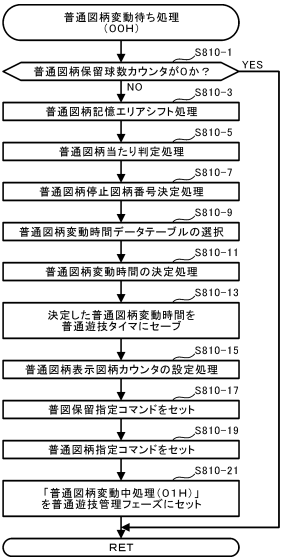
【図 50】

普通遊技管理フェーズ	普通遊技制御モジュール
00H	普通図柄変動待ち処理
01H	普通図柄変動中処理
02H	普通図柄停止図柄表示処理
03H	普通電動役物入賞口開放前処理
04H	普通電動役物入賞口開放制御処理
05H	普通電動役物入賞口閉鎖有効処理
06H	普通電動役物入賞口終了ウェイト処理

【図 51】



【図 52】



10

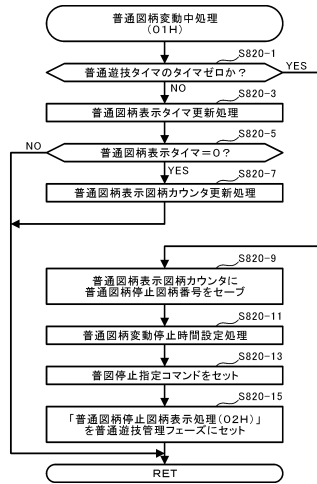
20

30

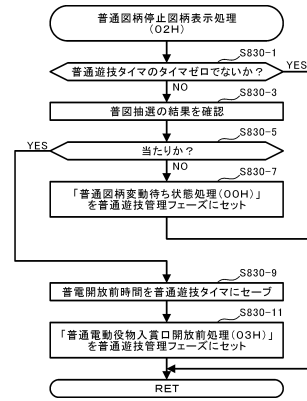
40

50

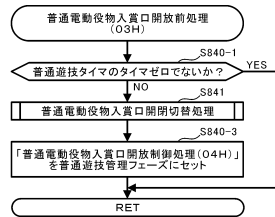
【図 5 3】



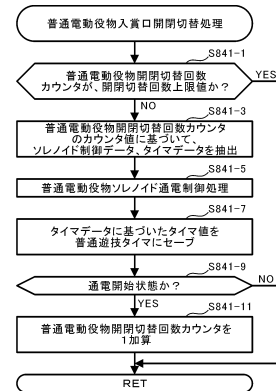
【図 5 4】



【図 5 5】



【図 5 6】



10

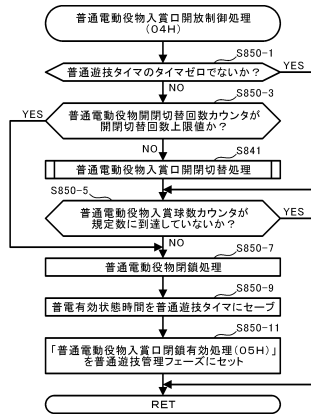
20

30

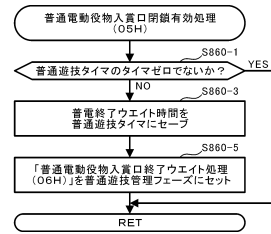
40

50

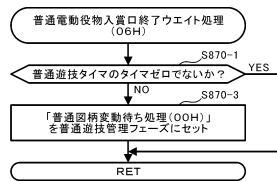
【図 57】



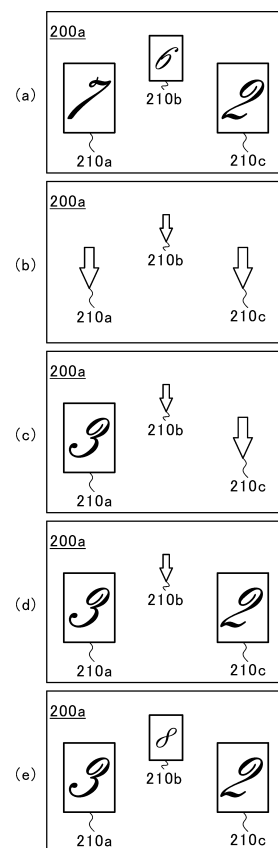
【図 58】



【図 59】



【図 60】



10

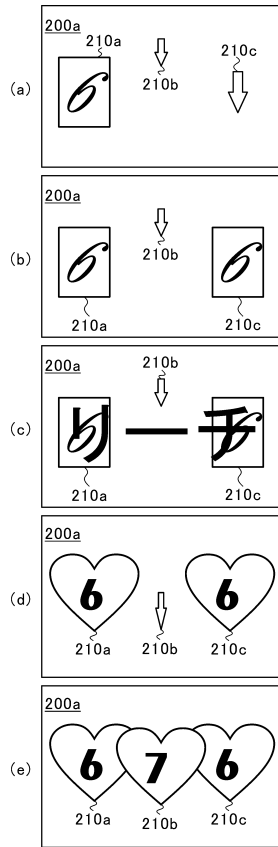
20

30

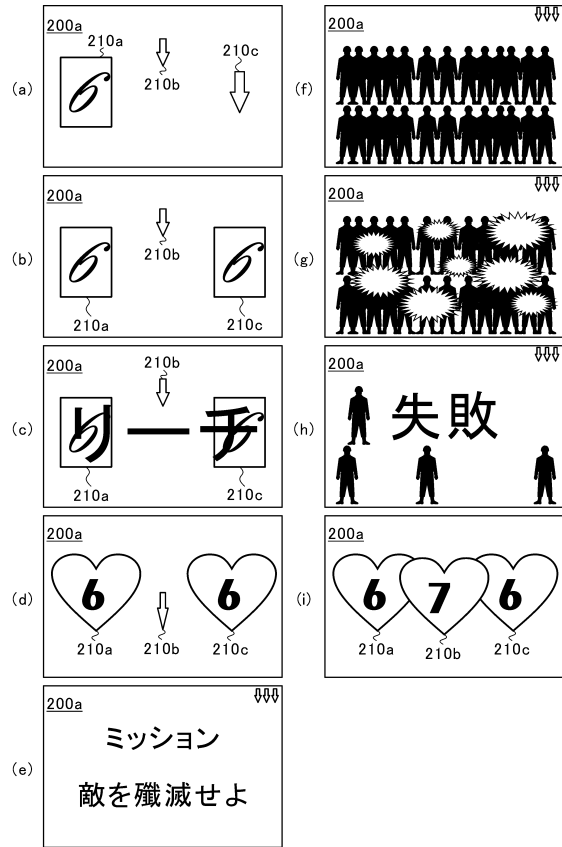
40

50

【図 6 1】



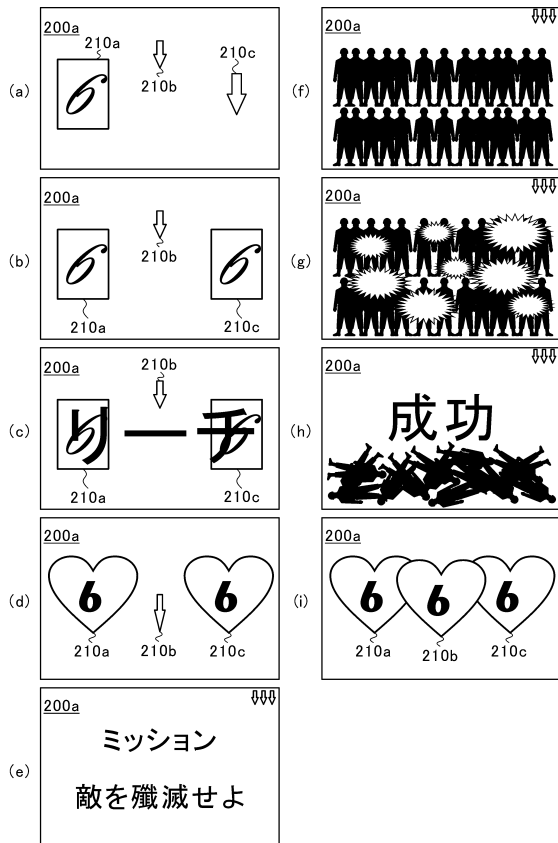
【図 6 2】



10

20

【図 6 3】



【図 6 4】

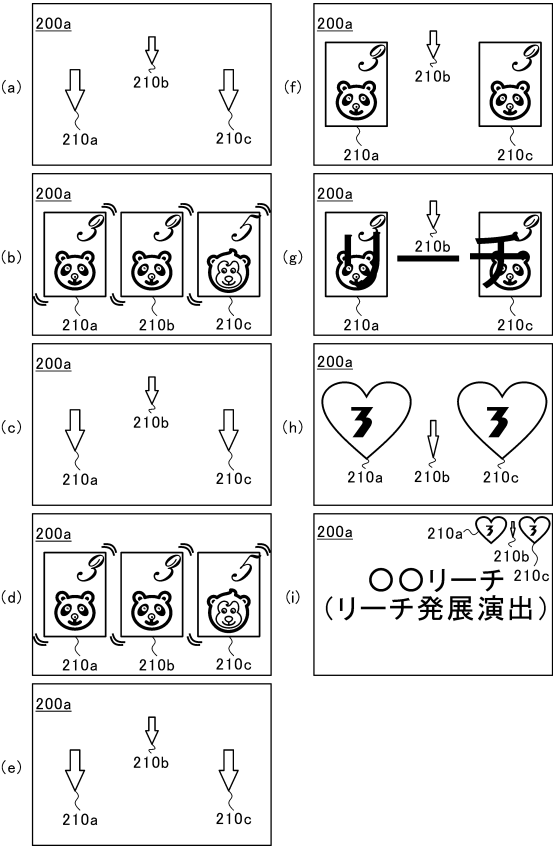


30

40

50

【図 6 5】



【図 6 6】

(a) 前半変動演出決定テーブル

変動モード番号 (変動モードコマンド)	なし	ノーマル リーチ1	ノーマル リーチ2	擬似2a	擬似2b	擬似2c	擬似3a	擬似3b	擬似3c
01H	250	0	0	0	0	0	0	0	0
00H	0	250	0	0	0	0	0	0	0
00H	0	0	250	0	0	0	0	0	0
00H	0	0	0	250	0	0	0	0	0
00H	0	0	0	0	250	0	0	0	0
00H	0	0	0	0	0	250	0	0	0
00H	0	0	0	0	0	0	250	0	0
00H	0	0	0	0	0	0	0	250	0

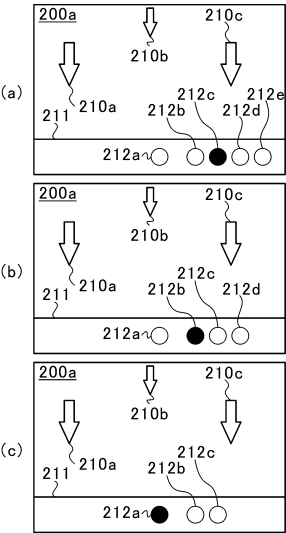
(b) 後半変動演出決定テーブル

変動パターン番号 (変動パターンコマンド)	ノーマル ハズレ1	ノーマル ハズレ2	特殊 ハズレ1	特殊 ハズレ2	発展1	発展2	発展3	発展4	発展5
00H	250	0	0	0	0	0	0	0	0
00H	0	250	0	0	0	0	0	0	0
00H	0	0	250	0	0	0	0	0	0
00H	0	0	0	250	0	0	0	0	0
00H	0	0	0	0	250	0	0	0	0
00H	0	0	0	0	0	250	0	0	0
00H	0	0	0	0	0	0	250	0	0
00H	0	0	0	0	0	0	0	250	0

10

20

【図 6 7】



【図 6 8】

(a) 最終保留表示パターン決定テーブル

変動パターン番号 (先読み指定コマンド)	デフォルト (白)	点滅	青	黄	緑	黒	赤	プレミア (紅)
00H	200	50	0	0	0	0	0	0
00H	200	0	40	5	5	0	0	0
00H	200	0	30	0	10	5	5	0
00H	200	0	0	0	0	0	50	0
00H	200	0	10	10	10	10	10	0
00H	200	0	0	0	0	0	40	10
00H	0	0	0	0	250	0	0	0

(b) 1つ前保留表示パターン決定テーブル

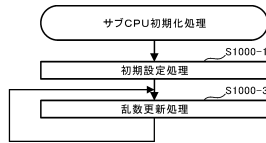
保留表示パターン	デフォルト (白)	点滅	青	黄	緑	黒	赤	プレミア (紅)
デフォルト(白)	250	0	0	0	0	0	0	0
点滅	0	250	0	0	0	0	0	0
青	0	200	50	0	0	0	0	0
黄	0	200	30	20	0	0	0	0
緑	0	190	30	20	10	0	0	0
黒	0	200	20	10	10	10	0	0
赤	0	100	50	40	30	20	10	0
プレミア(紅)	0	50	50	50	50	50	0	0

30

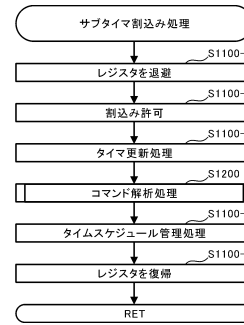
40

50

【図 69】

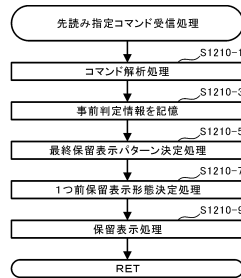


【図 70】

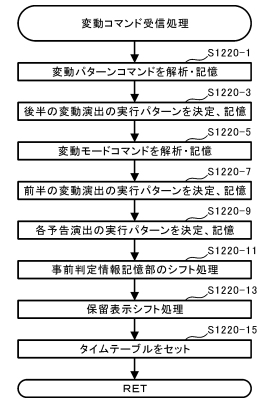


10

【図 71】



【図 72】



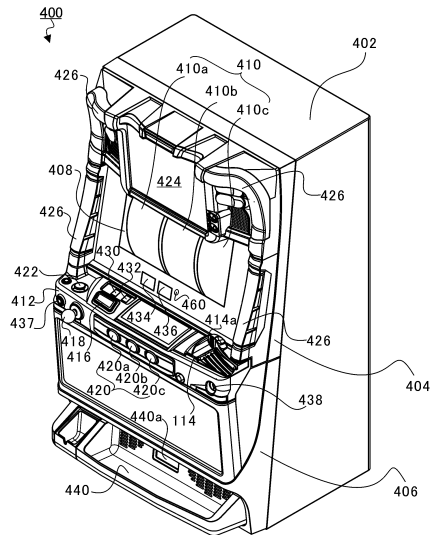
20

30

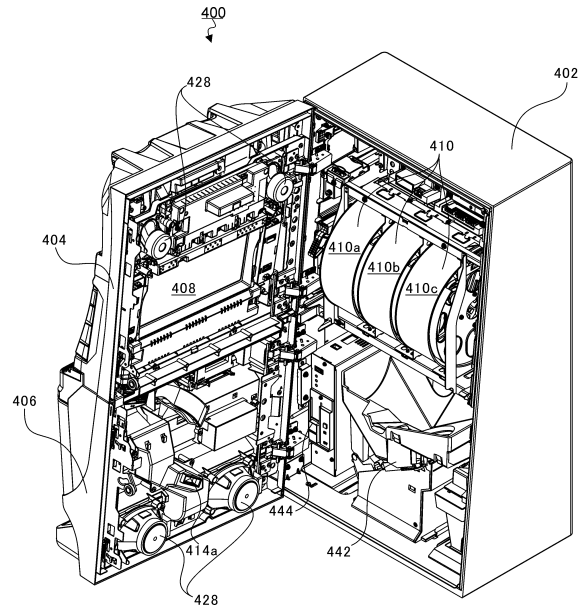
40

50

【圖 7 3】



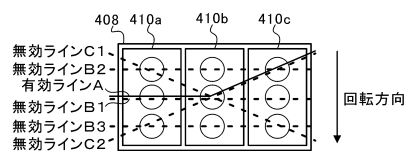
【圖 7 4】



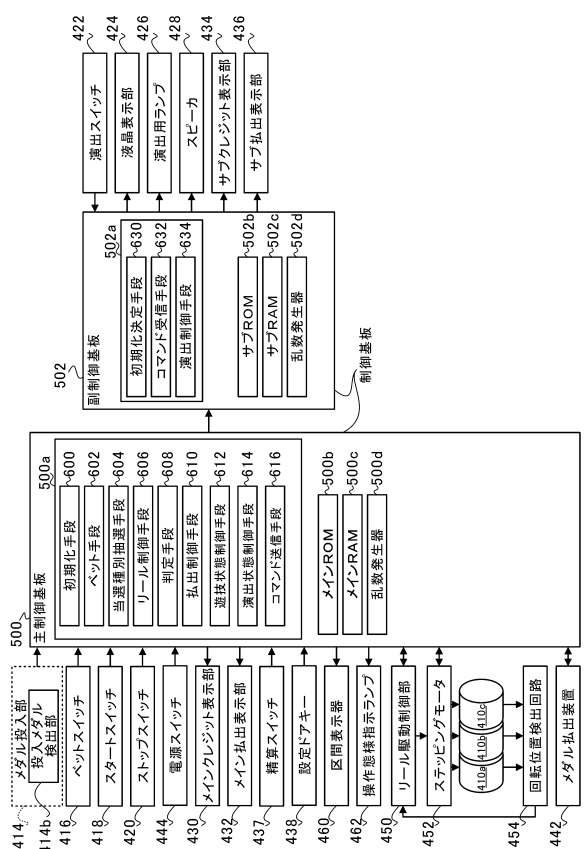
【図 7 5】

図柄番号	左リール410a	中リール410b	右リール410c
0	リプレイA	リプレイA	ベル
1	チェリー	チェリー	スイカ
2	BAR1	BAR1	BAR1
3	スイカ	黒丸	チェリー
4	ベル	ベル	リプレイA
5	リプレイA	リプレイA	ベル
6	ベル	チェリー	スイカ
7	赤7	BAR2	BAR2
8	スイカ	黒丸	チェリー
9	ベル	ベル	リプレイA
10	リプレイA	リプレイA	ベル
11	チェリー	チェリー	スイカ
12	白丸	白丸	白丸
13	スイカ	スイカ	チェリー
14	ベル	ベル	リプレイA
15	リプレイA	リプレイA	ベル
16	BAR2	チェリー	スイカ
17	リプレイB	赤7	赤7
18	スイカ	スイカ	チェリー
19	ベル	ベル	リプレイA

(b)



【図 7 6】



【図 7 7】

当選役	左リール410a	中リール410b	右リール410c	払出枚数
リプレイ1	△	△	BAR・BAR	—
リプレイ2	△	△	△	—
リプレイ3	BAR・△	BAR・△	△	—
リプレイ4	BAR・△	BAR・△	△	—
リプレイ5	BAR	△	BAR	—
リプレイ6	BAR	△	BAR	—
リプレイ7	△	△	△	—
小役1	△	△	△	11
小役2	△	△	△	11
小役3	△	△	△	11
小役4	△	△	△	11
小役5	△	△	△	11
小役6	△	△	△	11
小役7	△	△	△	3
小役8	△	△	△	3
小役9	△	△	△	3
小役10	△	△	△	3
小役11	△	△	△	1
小役12	BAR	△	BAR	1
小役13	BAR	△	BAR	1
小役14	△	△	△	1
小役15	△	△	△	1
小役16	△	△	△	1
小役17	△	△	△	1
小役18	△	△	△	1
小役19	△	△	△	1
小役20	△	△	△	1
小役21	△	△	△	1
小役22	△	△	△	1
小役23	△	△	△	1
小役24	BAR	△	BAR	1
小役25	BAR	△	BAR	1
小役26	BAR	△	BAR	1
小役27	△	△	△	1
小役28	△	△	△	1
小役29	△	△	△	1
小役30	△	△	△	1
小役31	△	△	△	1
小役32	△	△	△	1
小役33	△	△	△	1
小役34	△	△	△	1
小役35	BAR・△	△	△	1
小役36	△	△	△	1
小役37	△	△	△	1
小役38	BAR・△	△	△	1
小役39	BAR・△	△	△	1
RBB	BAR	△	△	—

【図 7 8】

当選種別抽選テーブル					
当選領域	当選種別	当選役	非内部	RBB内部中	RBB動作中
打順に応じて優先的に表示可能な図柄組み合わせ					
0	ハズレ	—			—
1	小役ALL	小役1～39			小役1～39
2	1枚ALLA	小役11～39			小役11～39
3	1枚ALLB	小役12～27			小役12～27
4	戻1枚1	小役12、20			小役12、20
5	戻1枚2	小役13、21			小役12、20
6	戻1枚3	小役14、22			小役12、20
7	戻1枚4	小役15、23			小役12、20
8	弱チェリー	小役35、36			小役35、36
9	強チェリー	小役37～39			小役37～39
10	スイカA	小役7、8、11			小役7
11	スイカB	小役7～10			小役8
12	強ベル	小役1、3、5			小役1、3、5
13	共通ベルA	小役1～6			小役1、小役2、小役3、小役4、小役5、小役6
14	共通ベルB	小役1～6、11			小役1、小役2、小役3、小役4、小役5、小役6
15	打順ベル1A	小役1、12、16、28			小役1、小役28、小役12、小役16
16	打順ベル2A	小役2、13、17、29			小役29、小役2、小役13、小役17
17	打順ベル3A	小役3、20、30、31			小役20、小役3、小役30、小役20
18	打順ベル4A	小役4、21、30、32			小役21、小役30、小役4、小役21
19	打順ベル5A	小役5、24、33、34			小役24、小役5、小役33、小役34
20	打順ベル6A	小役6、25、33			小役25、小役33、小役6
21	打順ベル1B	小役1、14、19、28			小役1、小役28、小役14、小役19
22	打順ベル2B	小役2、15、18、29			小役29、小役2、小役15、小役18
23	打順ベル3B	小役3、22、30、31			小役22、小役3、小役30、小役22
24	打順ベル4B	小役4、23、30、32			小役23、小役30、小役4、小役23
25	打順ベル5B	小役5、26、33、34			小役26、小役5、小役33、小役34
26	打順ベル6B	小役6、27、33			小役27、小役33、小役6

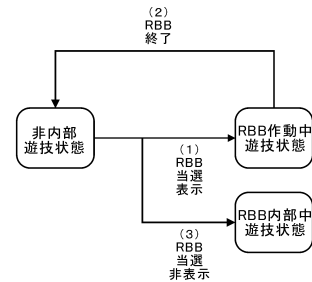
10

20

【図 7 9】

当選種別抽選テーブル					
当選領域	当選種別	当選役	非内部	RBB内部中	RBB動作中
打順に応じて優先的に表示可能な図柄組み合わせ					
27	シンボルリプレイA1	リプレイ1～3			リプレイ1、リプレイ3: B■、リプレイ3: ■BB
28	シンボルリプレイA2	リプレイ1～4			リプレイ1、リプレイ3: ■B■、リプレイ3: BBB
29	シンボルリプレイA3	リプレイ1～5			リプレイ1、リプレイ3C: BB■、リプレイ3: BBB
30	シンボルリプレイA4	リプレイ1～3、5			リプレイ1、リプレイ3: ■■B、リプレイ3: ■BB
31	シンボルリプレイA5	リプレイ1～3、6			リプレイ1、リプレイ3: B■B、リプレイ3: ■BB
32	シンボルリプレイA6	リプレイ1～4、6			リプレイ1、リプレイ3: ■BB、リプレイ3: BBB
33	シンボルリプレイA7	リプレイ1～6			リプレイ1、リプレイ3: BBB、リプレイ3: BBB
34	シンボルリプレイB1	リプレイ1～3、7			リプレイ1、リプレイ3: B■、リプレイ1
35	シンボルリプレイB2	リプレイ1～4、7			リプレイ1、リプレイ3: ■B■、リプレイ1
36	シンボルリプレイB3	リプレイ1～3、5、7			リプレイ1、リプレイ3: BB■、リプレイ1
37	シンボルリプレイB4	リプレイ1～3、6、7			リプレイ1、リプレイ3: ■■B、リプレイ1
38	シンボルリプレイB5	リプレイ1～5、7			リプレイ1、リプレイ3: B■B、リプレイ1
39	シンボルリプレイB6	リプレイ1～3、5～7			リプレイ1、リプレイ3: ■BB、リプレイ1
40	通常リプレイ	リプレイ1、2			リプレイ1
41	RBB1	RBB、小役12～27			小役12～27
42	RBB2	RBB、小役12、20			小役12、20
43	RBB3	RBB、小役13、21			小役12、20
44	RBB4	RBB、小役14、22			小役12、20
45	RBB5	RBB、小役15、23			小役12、20
46	RBB6	RBB			RBB

【図 8 0】

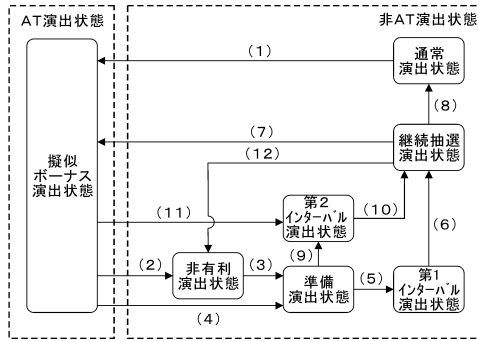


30

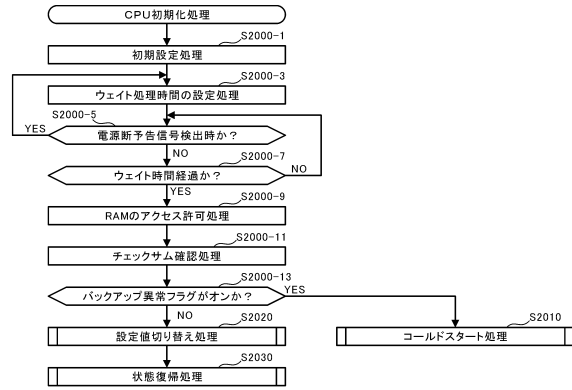
40

50

【図 8 1】

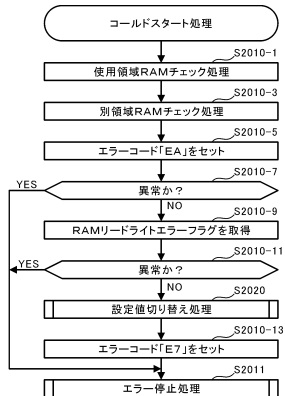


【図 8 2】

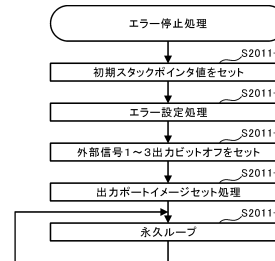


10

【図 8 3】



【図 8 4】



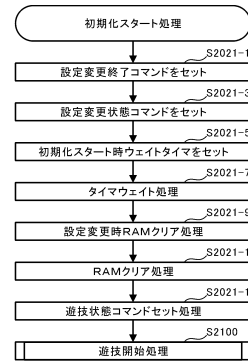
20

30

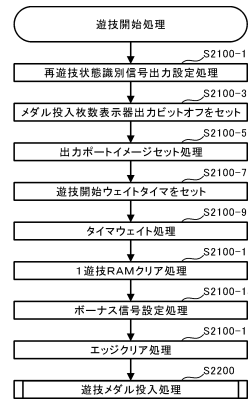
40

50

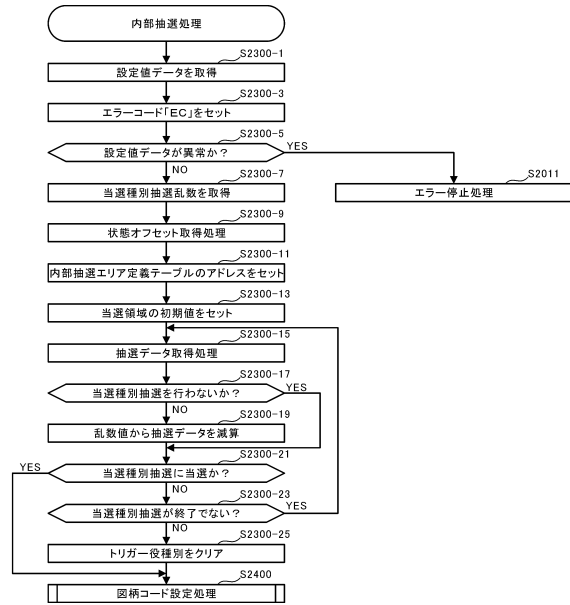
20



40

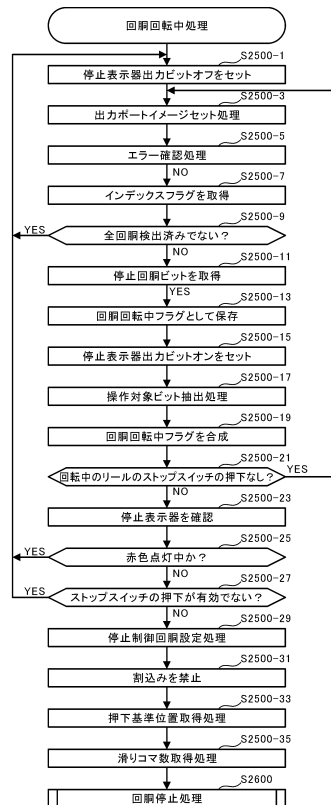


【圖 90】



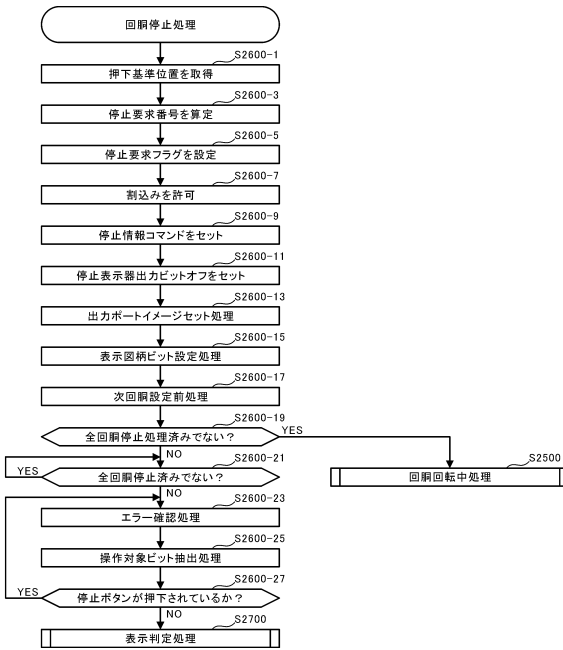
20

【圖 9 2】

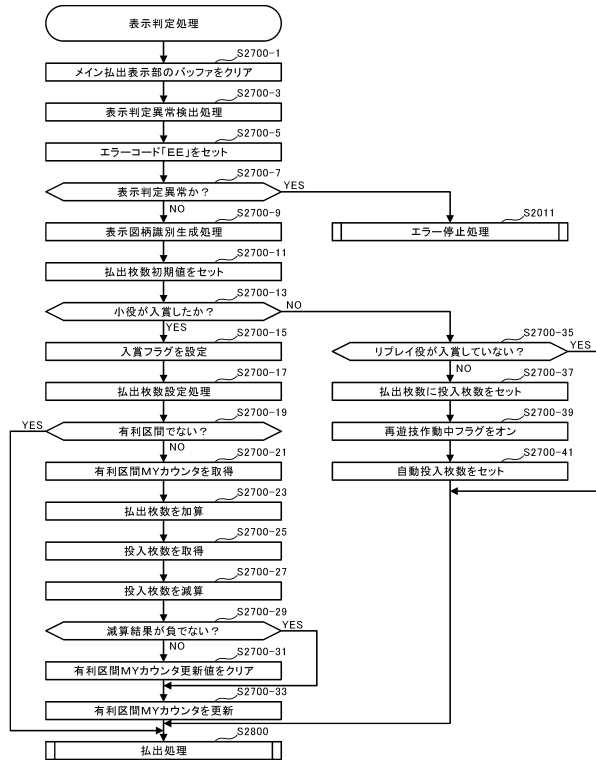


40

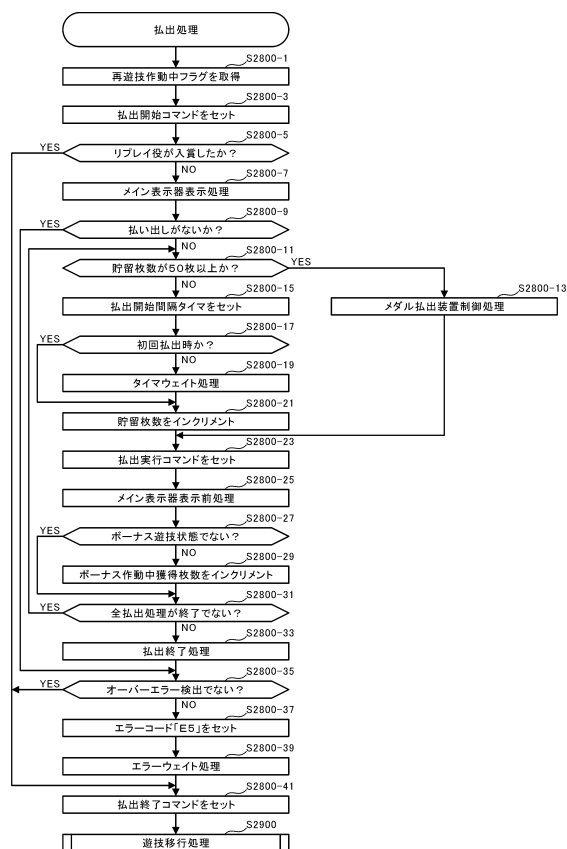
【図 9 3】



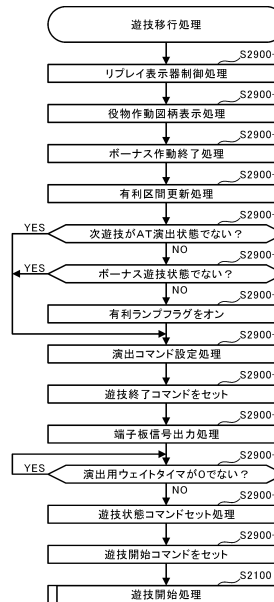
【図 9 4】



【図 9 5】



【図 9 6】



10

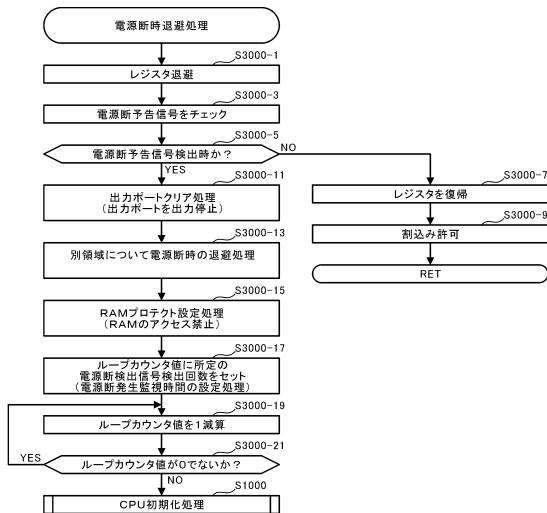
20

30

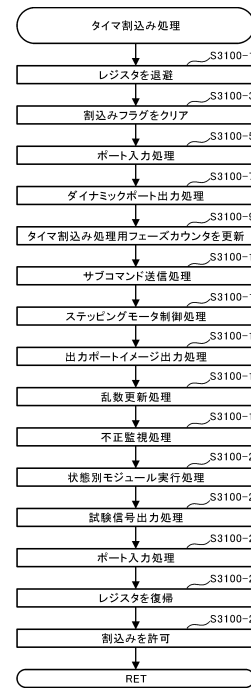
40

50

【図 97】



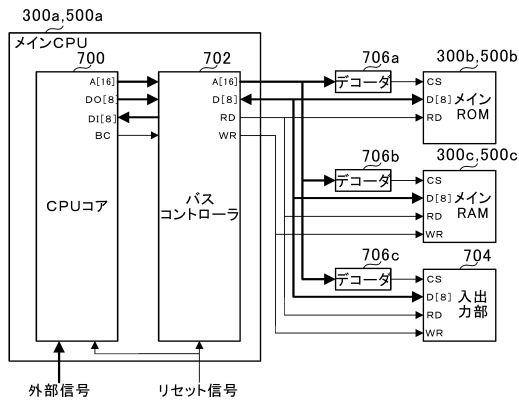
【図 98】



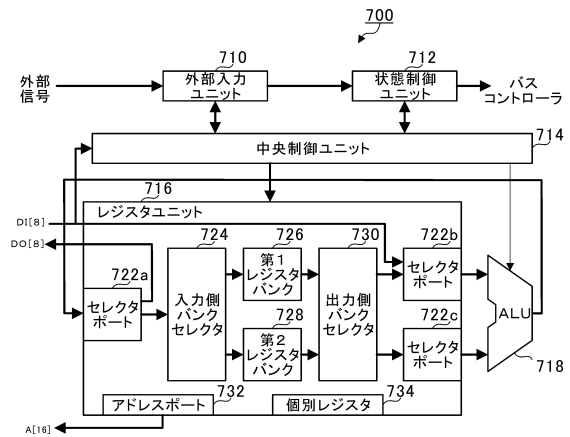
10

20

【図 99】



【図 100】

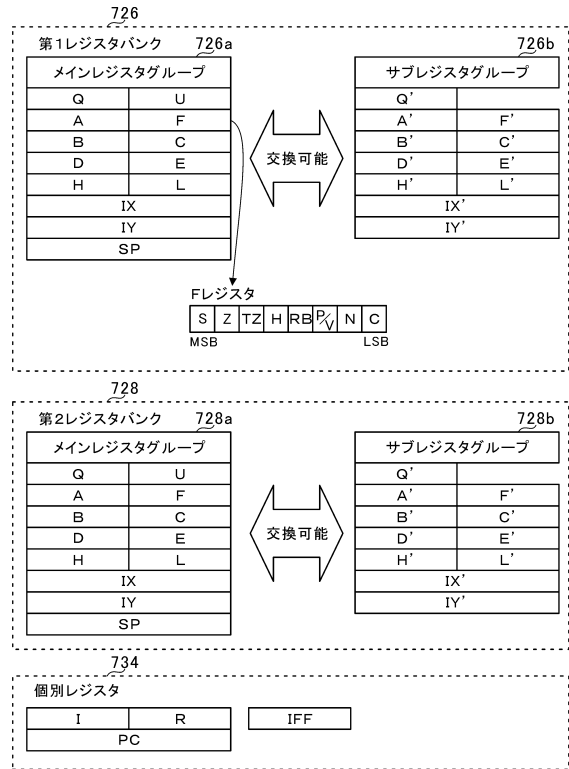


30

40

50

【図101】



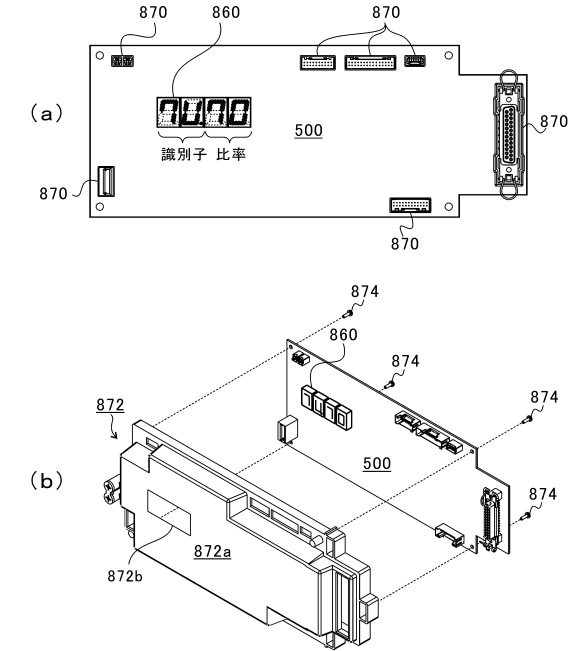
【図102】

メモリマップ				
アドレス	メモリ種別	用途		
0000h～11FFh	メインROM	使用領域	プログラムの命令コード	} 4. 5kbyte } 3. 0kbyte
1200h～1DF3h			プログラムデータ	
1DF4h～1DFFh		別領域	未使用	
1E00h～1FFFh			コメント領域	
2000h～3FBFh			プログラムの命令コード	
3FC0h～3FFFh			プログラムデータ	
3FC0h～3FFFh		プログラム管理領域		
4000h～EFFFh	未割当			
F000h～F13Fh	メインRAM	使用領域	ワークエリア	
F140h～F1BFh			未使用	
F1C0h～F1FFh			スタック領域	
F200h～F20Fh			未使用	
F210h～F22Fh		別領域	ワークエリア	
F230h～F246h			スタック領域	
F247h～F3FFh			未使用	
F400h～FDFh			未割当	
FE00h～FEFFh	入出力部			
FF00h～FFFFh	未割当			

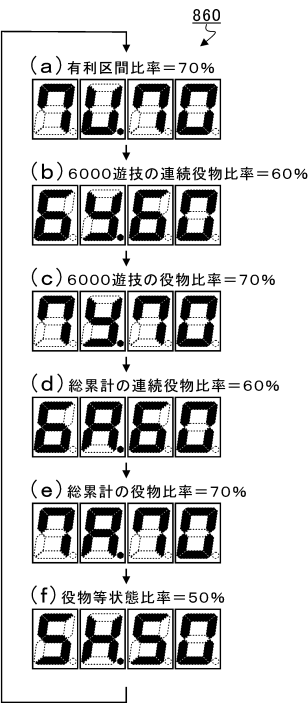
10

20

【図103】



【図104】

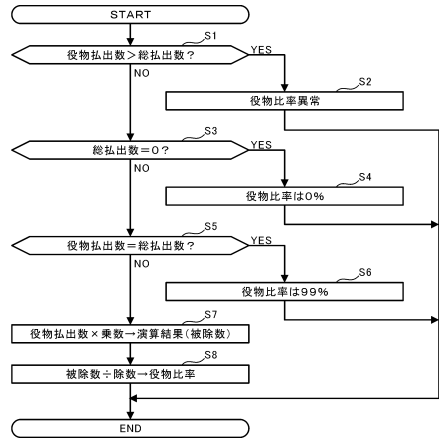


30

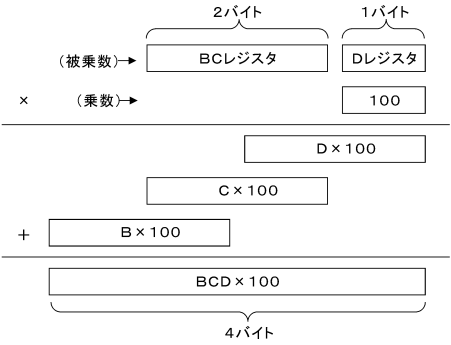
40

50

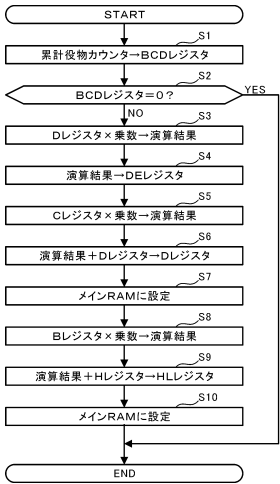
【図 1 0 5】



【図 1 0 6】



【図 1 0 7】



【図 1 0 8】

行数	コマンド	サイズ	サイクル数
1	E_PYSAV:	0	0
2	LD A, (_EX_YAKR)	3	4
3	LD D, A	1	1
4	LD BC, (_EX_YAKR+1)	4	6
5	OR A, B	1	1
6	A, C	1	1
7	JR Z, E_PYSAV28	2	3/2
8	LD A, D	1	1
9	MUL HL, A, 100	3	4
10	EX DE, HL	1	1
11	LD A, C	1	1
12	MUL HL, A, 100	3	4
13	ADDWB HL, D	2	2
14	LD D, L	2	2
15	EX DE, HL	1	1
16	LD (_EX_WRK2), HL	3	5
17	LD A, B	1	1
18	MUL HL, A, 100	3	4
19	ADDWB HL, D	2	2
20	LD (_EX_WRK2+2), HL	3	5
単純合計		38	49/48

10

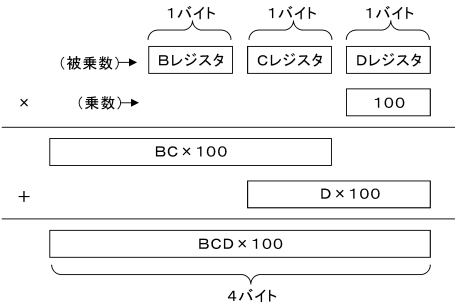
20

30

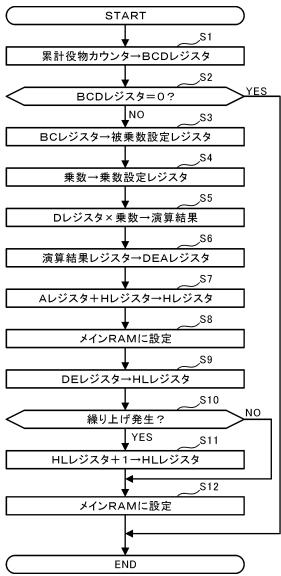
40

50

【図 1 0 9】



【図 1 1 0】



10

【図 1 1 1】

行数	コマンド	サイズ	サイクル数
1	E_PYSAV:	0	0
2	LD A, (_EX_YAKR)	3	4
3	LD D, A	1	1
4	LD BC, (_EX_YAKR+1)	4	6
5	OR A, B	1	1
6	OR A, C	1	1
7	JR Z, E_PYSAV28	2	3/2
8	LD @MUL16_, BC	4	6
9	LD A, 100	2	2
10	OUT (LOW @MULB16_), A	2	3
11	LD A, D	1	1
12	MUL HL, A, 100	3	4
13	IN A, (LOW @MUL32_+0)	2	3
14	LD DE, (@MUL32_+1)	4	6
15	ADD A, H	2	2
16	LD H, A	1	1
17	OUT (LOW @DIVA32_+0), HL	3	5
18	EX DE, HL	1	1
19	JR NC, E_PYSAVX1	2	3/2
20	INC HL	1	1
21	E_PYSAVX1:	0	0
22	OUT (LOW @DIVA32_+2), HL	3	5
単純合計		43	59/57

【図 1 1 2】

行数	コマンド	サイズ	サイクル数
1	E_PYSAV:	0	0
2	LD A, (_EX_YAKR)	3	4
3	LD D, A	1	1
4	LD BC, (_EX_YAKR+1)	4	6
5	OR A, B	1	1
6	OR A, C	1	1
7	JR Z, E_PYSAV28	2	3/2
8	LD A, D	1	1
9	MUL HL, A, 100	3	4
10	EX DE, HL	1	1
11	LD A, C	1	1
12	MUL HL, A, 100	3	4
13	ADDWB HL, D	2	2
14	LD D, L	2	2
15	EX DE, HL	1	1
16	OUT (LOW @DIVA32_+0), HL	3	5
17	LD A, B	1	1
18	MUL HL, A, 100	3	4
19	ADDWB HL, D	2	2
20	OUT (LOW @DIVA32_+2), HL	3	5
単純合計		38	49/48

20

30

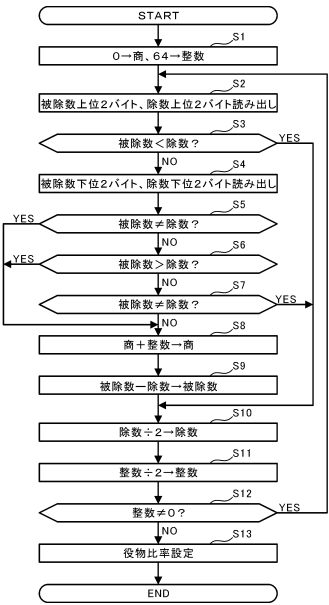
40

50

【図 1 1 3】

(減算ステップ)				(加算ステップ)			
被除数	整数×除数	更新された被除数	商	整数	更新された商		
(1回目): 712300	− 64×10000 =	72300	0 + 64 =	64			
(2回目):	減算しない		維持				
(3回目):	減算しない		維持				
(4回目):	減算しない		維持				
(5回目): 72300	− 4×10000 =	32300	64 + 4 =	68			
(6回目): 32300	− 2×10000 =	12300	68 + 2 =	70			
(7回目): 12300	− 1×10000 =	2300	70 + 1 =	71			役物比率

【図 1 1 4】

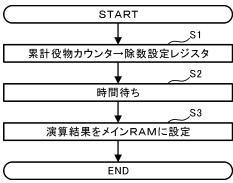


10

【図 1 1 5】

行数	コマンド	サイズ	サイクル数
1	XOR A, A	1	1
2	LD C, 64	2	2
3	E_PYS25:	0	0
4	LD HL, _EX_WRK2+2	3	3
5	LD DE, (HL+4)	3	5
6	JCP C, (HL), DE, E_PYS27	3	6/5
7	LD HL, _EX_WRK1	3	3
8	LD DE, (HL+4)	2	4
9	JR NZ, E_PYS26	2	3/2
10	JCP C, (HL), DE, E_PYS26	3	6/5
11	JR NZ, E_PYS27	2	3/2
12	E_PYS26:	0	0
13	ADD A, C	1	1
14	SUB DE, (HL)	2	4
15	LD (HL+4), DE	2	4
16	LD DE, (HL+2)	2	4
17	LD HL, (HL+6)	2	4
18	SBC HL, DE	2	2
19	LD (_EX_WRK2+2), HL	3	5
20	E_PYS27:	0	0
21	LD HL, _EX_WRK1+3	3	3
22	SRL (HL)	2	5
23	DEC HL	1	1
24	RR (HL)	2	5
25	DEC HL	1	1
26	RR (HL)	2	5
27	DEC HL	1	1
28	RR (HL)	2	5
29	SRL C	2	2
30	JR NZ, E_PYS25	2	3/2
31	E_PYS28:	0	0
32	LD (_EX_DPA7), A	3	4
単純合計		59	95/90

【図 1 1 6】



20

30

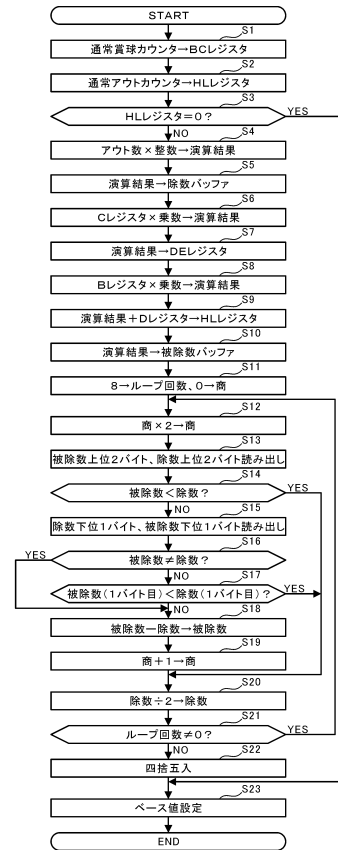
40

50

【図 1 1 7】

行数		コマンド	サイズ	サイクル数
1		LD HL, (_EX_PRZR)	3	5
2		LD HL, (_EX_PRZR+2)	3	4
3		OUT (LOW @DIVB32 +0), HL	3	5
4		OUT (LOW @DIVB32 +2), A	2	3
5		NOP	1	1
6		NOP	1	1
7		NOP	1	1
8		NOP	1	1
9		IN A, (LOW @DIV32 ++0)	2	3
10	E_PYSAV28:		0	0
11		LD (_EX_DPA7), A	3	4
単純合計			20	28

【図 1 1 8】



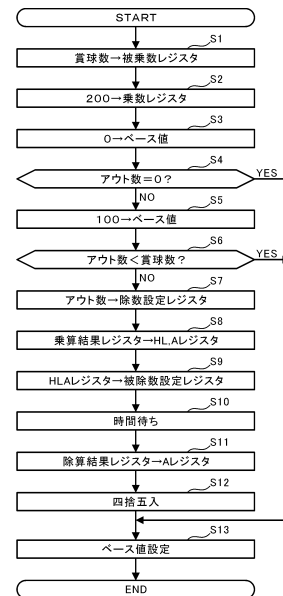
10

20

【図 1 1 9】

行数	コマンド	サイズ	サイクル数
1	E_SHYMT_40:	0	0
2	XOR A, A	1	1
3	LD BC, (R_ERW_TJS)	4	6
4	HL, (R_ERW_TJO)	3	5
5	JR TZ, E_SHYMT_60	2	3/2
6	SRL H	2	2
7	RR L	2	2
8	RRA L	1	4
9	LD (R_ERW_BF2), A	3	4
10	LD (R_ERW_BF2+1), HL	3	5
11	LD A, C	1	1
12	MUL HL, A, 200	3	4
13	EX DE, HL	1	1
14	LD A, B	1	1
15	MUL HL, A, 200	3	4
16	ADDWB HL, D	2	2
17	LD A, E	1	1
18	LD (R_ERW_BF1), A	3	4
19	LD (R_ERW_BF1+1), HL	3	5
20	LD BC, 8*256+0	3	3
21	E_SHYMT_45:	0	0
22	SLA C	2	2
23	LD HL, R_ERW_BF1+1	3	3
24	LD DE, (HL+3)	2	4
25	JCP C, (HL), DE, E_SHYMT_55	3	6/5
26	LD L, LOW R_ERW_BF2	2	2
27	LD A, (HL-3)	3	4
28	JR NZ, E_SHYMT_50	2	3/2
29	CP A, (HL)	2	2
30	JR C, E_SHYMT_55	2	3/2
31	E_SHYMT_50:	0	0
32	SUB A, (HL)	1	2
33	LD (HL-3), A	3	4
34	LD HL, (HL-2)	3	5
35	SBC HL, DE	2	2
36	LD (R_ERW_BF1+1), HL	3	5
37	INC C	1	1
38	E_SHYMT_55:	0	0
39	LD HL, R_ERW_BF2+2	3	3
40	(HL)	2	5
41	SRL HL	1	1
42	DEC (HL)	2	5
43	DEC HL	1	1
44	RR HL	2	5
45	DJNZ E_SHYMT_45	2	3/2
46	LD A, C	1	1
47	SRL A	2	2
48	ADC A, 0	2	2
49	E_SHYMT_60:	0	0
50	LD HL, R_ERW_BAS	3	3
51	LD (HL), A	1	2
単純合計		97	136/131

【図 120】



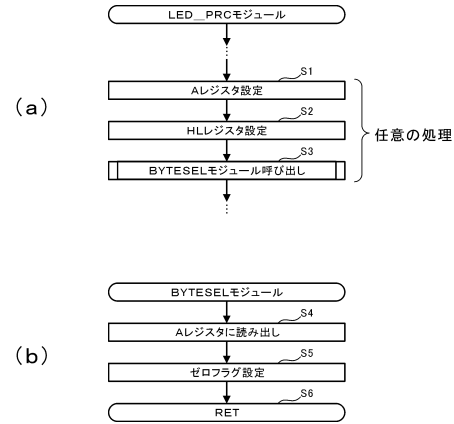
30

40

【図 1 2 1】

行数	コマンド	サイズ	サイクル数
1	LD HL, (R_ERW_TJS)	3	5
2	OUT (LOW @MULA16_), HL	3	5
3	EX DE, HL	1	1
4	LD A, 200	2	2
5	OUT (LOW @MULB16_), A	2	3
6	XOR A, A	1	1
7	LD HL, (R_ERW_TJ0)	3	5
8	JR TZ, E_SHYMT_60	2	3/2
9	LD A, 100	2	2
10	JCP C, HL, DE, E_SHYMT_60	3	4/3
11	OUT (LOW @DIVB32_), HL	3	5
12	IN A, (LOW @MUL32_)	2	3
13	IN HL, (LOW @MUL32_+1)	3	5
14	OUT (LOW @DIVA32_+0), A	2	3
15	OUT (LOW @DIVA32_+1), HL	3	5
16	NOP	1	1
17	NOP	1	1
18	NOP	1	1
19	NOP	1	1
20	IN A, (LOW @DIV32_+0)	2	3
21	SRL A	2	2
22	ADC A, 0	2	2
23	E_SHYMT_60: LD HL, R_ERW_BAS	3	3
24	LD (HL), A	1	2
25			
単純合計		49	68/66

【図 1 2 2】



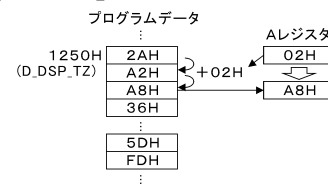
【図 1 2 3】

(a) (1行目) LDQ A,(LOW R_TZ1_DSP)
 (2行目) LD HL,D_DSP_TZ
 (3行目) RST BYTESEL

(b) (1行目) BYTESEL:
 (2行目) ADDWB HL,A
 (3行目) LD A,(HL)
 (4行目) OR A,A
 (5行目) RET

(c) (1行目) D_DSP_TZ:
 (2行目) DB 2AH
 (3行目) DB A2H
 (4行目) DB 5DH
 (5行目) DB FDH

【図 1 2 4】



【図 1 2 5】

(a) (1行目) LDQ A,(LOW R_TZ1_DSP)
 (2行目) LD HL,D_DSP_TZ
 (3行目) RST BYTESEL

(b) (1行目) BYTESEL:
 (2行目) ADDALD A,(HL)
 (3行目) OR A,A
 (4行目) RET

(c) (1行目) D_DSP_TZ:
 (2行目) DB 2AH
 (3行目) DB A2H
 (4行目) DB 5DH
 (5行目) DB FDH

【図 1 2 6】

(a) (1行目) LDQ A,(LOW R_TZ1_DSP)
 (2行目) LD HL,D_DSP_TZ
 (3行目) ADDALD A,(HL)

(b) (1行目) D_DSP_TZ:
 (2行目) DB 2AH
 (3行目) DB A2H
 (4行目) DB 5DH
 (5行目) DB FDH

10

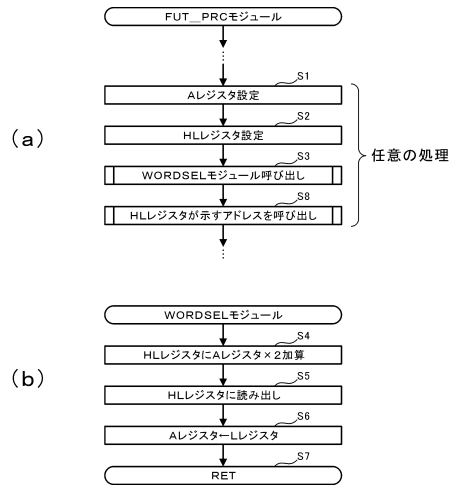
20

30

40

50

【図 1 2 7】



【図 1 2 8】

(a)

```

(1行目) LDQ A,(LOW R,FUT_PHS)
(2行目) LD HL,J JMP_FUT
(3行目) RST WORDSEL
(4行目) CALL (HL)
  
```

(b)

```

(1行目) WORDSEL:
(2行目) ADDWB HL,A
(3行目) ADDWB HL,A
(4行目) LD HL,(HL)
(5行目) LD A,L
(6行目) RET
  
```

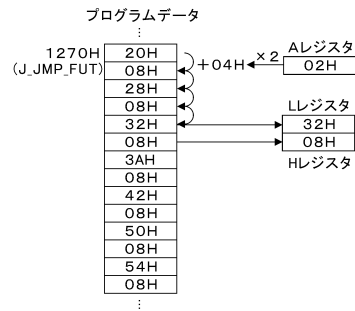
(c)

```

(1行目) J JMP_FUT:
(2行目) DW FZ_STA
(3行目) DW FZ_SPN
(4行目) DW FZ_STP
(5行目) DW FD_PRE
(6行目) DW FD_OPN
(7行目) DW FD_CLS
(8行目) DW FD_END
  
```

10

【図 1 2 9】



【図 1 3 0】

(a)

```

(1行目) LDQ A,(LOW R,FUT_PHS)
(2行目) LD HL,J JMP_FUT
(3行目) RST WORDSEL
(4行目) CALL (HL)
  
```

(b)

```

(1行目) WORDSEL:
(2行目) ADDALDW HL,(HL)
(3行目) LD A,L
(4行目) RET
  
```

(c)

```

(1行目) J JMP_FUT:
(2行目) DW FZ_STA
(3行目) DW FZ_SPN
(4行目) DW FZ_STP
(5行目) DW FD_PRE
(6行目) DW FD_OPN
(7行目) DW FD_CLS
(8行目) DW FD_END
  
```

20

【図 1 3 1】

(a)

```

(1行目) LDQ A,(LOW R,FUT_PHS)
(2行目) LD HL,J JMP_FUT
(3行目) ADDALDW HL,(HL)
(4行目) CALL (HL)
  
```

(b)

```

(1行目) J JMP_FUT:
(2行目) DW FZ_STA
(3行目) DW FZ_SPN
(4行目) DW FZ_STP
(5行目) DW FD_PRE
(6行目) DW FD_OPN
(7行目) DW FD_CLS
(8行目) DW FD_END
  
```

【図 1 3 2】

(a)

```

(1行目) CAL_MOD:
(2行目) LD HL,T_INT_PHASE
(3行目) ADD A,A
(4行目) ADDWB HL,A
(5行目) LD HL,(HL)
(6行目) JP (HL)
  
```

(b)

```

(1行目) CAL_MOD:
(2行目) LD HL,T_INT_PHASE
(3行目) ADDALDW HL,(HL)
(4行目) JP (HL)
  
```

30

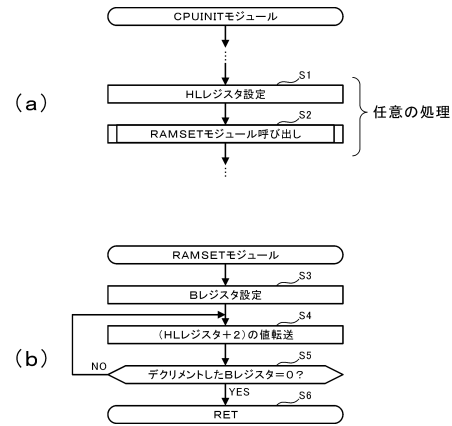
40

50

【図 1 3 3】

(a) (1行目) HID_JUG;
(2行目) LDQ A,(LOW_NMD_KND)
(3行目) LD HL,T_DAT_PGM-2
(4行目) ADD A,A
(5行目) ADDWB HL,A
(6行目) LD HL,(HL)
↓
(b) (1行目) HID_JUG;
(2行目) LDQ A,(LOW_NMD_KND)
(3行目) LD HL,T_DAT_PGM-2
(4行目) ADDALDW HL,(HL)
↓

【図 1 3 4】



【図 1 3 5】

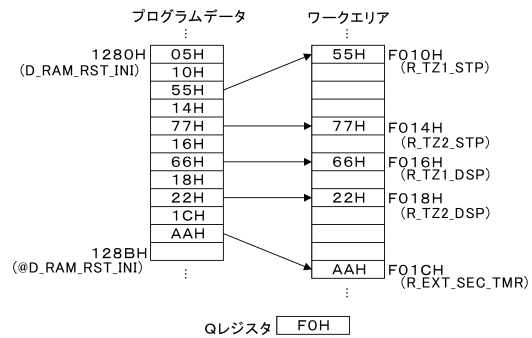
(a) (1行目) LD HL,D_RAM_RST_INI
(2行目) RST RAMSET

(b) (1行目) RAMSET;
(2行目) LD B,(HL)
(3行目) RAMSET_10:
(4行目) INLD AC,(HL)
(5行目) LDQ (C),A
(6行目) DJNZ RAMSET_10
(7行目) RET

(c) (1行目) D_RAM_RST_INI:
(2行目) DB (@D_RAM_RST_INI-D_RAM_RST_INI-1)/2
(3行目) DB LOW R_TZ1_STP ,@DSP_HAZ
(4行目) DB LOW R_TZ2_STP ,@DSP_HAZ
(5行目) DB LOW R_TZ1_DSP ,@DSP_HAZ
(6行目) DB LOW R_TZ2_DSP ,@DSP_HAZ
(7行目) DB LOW R_EXT_SEC_TMR ,@TMR_EXT_RMC
(8行目) @D_RAM_RST_INI:

(d) (1行目) R_TZ1_STP DS 1
(2行目) R_TZ2_STP DS 1
(3行目) R_TZ1_DSP DS 1
(4行目) R_TZ2_DSP DS 1
(5行目) R_EXT_SEC_TMR DS 1

【図 1 3 6】



10

20

30

40

50

【図 1 3 7】

(a) (1行目) LD HL,D_RAM_RST_INI
(2行目) RST RAMSET

(b) (1行目) RAMSET:
(2行目) LD B,(HL)
(3行目) INLDTQR AC,(HL)
(4行目) RET

(c) (1行目) D_RAM_RST_INI:
(2行目) DB (@D_RAM_RST_INI-D_RAM_RST_INI-1)/2
(3行目) DB LOW_R_TZ1_STP ,@DSP_HAZ
(4行目) DB LOW_R_TZ2_STP ,@DSP_HAZ
(5行目) DB LOW_R_TZ1_DSP ,@DSP_HAZ
(6行目) DB LOW_R_TZ2_DSP ,@DSP_HAZ
(7行目) DB LOW_R_EXT_SEC_TMR ,@TMR_EXT_RMC
(8行目) @D_RAM_RST_INI:

(d) (1行目) R_TZ1_STP DS 1
(2行目) R_TZ2_STP DS 1
(3行目) R_TZ1_DSP DS 1
(4行目) R_TZ2_DSP DS 1
(5行目) R_EXT_SEC_TMR DS 1

【図 1 3 8】

(a) (1行目) LD HL,T_ERR_RCV
(2行目) RST TABLSET

(b) (1行目) TABLSET:
(2行目) PUSH BC
(3行目) DI
(4行目) LD B,(HL)
(5行目) TABLSET10:
(6行目) INLD AC,(HL)
(7行目) LDQ (C),A
(8行目) DJNZ TABLSET10
(9行目) EI
(10行目) POP BC
(11行目) RET

(c) (1行目) T_ERR_RCV:
(2行目) DEFB (T_ERR_RCV-T_ERR_RCV)/2
(3行目) DEFB LOW_ERR_NUM ,0
(4行目) DEFB LOW_CRE_TMR ,0
(5行目) DEFB LOW_CRE_FLG ,0
(6行目) DEFB LOW_SNS_OLD ,0
(7行目) T_ERR_RCV_:

(d) (1行目) _ERR_NUM DEFS 1
(2行目) _CRE_TMR DEFS 1
(3行目) _CRE_FLG DEFS 1
(4行目) _SNS_OLD DEFS 1

10

【図 1 3 9】

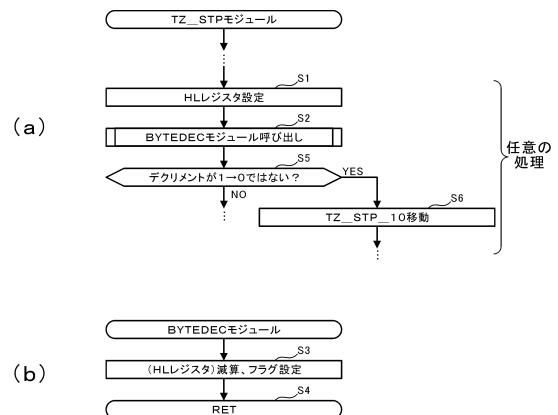
(a) (1行目) LD HL,T_ERR_RCV
(2行目) RST TABLSET

(b) (1行目) TABLSET:
(2行目) PUSH BC
(3行目) DI
(4行目) LD B,(HL)
(5行目) INLDTQR (HL)
(6行目) EI
(7行目) POP BC
(8行目) RET

(c) (1行目) T_ERR_RCV:
(2行目) DEFB (T_ERR_RCV-T_ERR_RCV)/2
(3行目) DEFB LOW_ERR_NUM ,0
(4行目) DEFB LOW_CRE_TMR ,0
(5行目) DEFB LOW_CRE_FLG ,0
(6行目) DEFB LOW_SNS_OLD ,0
(7行目) T_ERR_RCV_:

(d) (1行目) _ERR_NUM DEFS 1
(2行目) _CRE_TMR DEFS 1
(3行目) _CRE_FLG DEFS 1
(4行目) _SNS_OLD DEFS 1

【図 1 4 0】



20

30

40

50

【図 1 4 1】

デクリメント前	デクリメント後	ゼロフラグ	キャリーフラグ
2以上	1以上	0	0
1	0	1	1
0	0	1	0

【図 1 4 2】

(a) (1行目) LDQ HL,LOW R,EXT_HCT
(2行目) CALLF BYTEDEC
(3行目) JR NC,TZ_STP_10

(b) (1行目) BYTEDEC:
(2行目) LD A,(HL)
(3行目) RT Z,A
(4行目) CP A,2
(5行目) DEC (HL)
(6行目) RET

【図 1 4 3】

(a) (1行目) LDQ HL,LOW R,EXT_HCT
(2行目) CALLF BYTEDEC
(3行目) JR NC,TZ_STP_10

(b) (1行目) BYTEDEC:
(2行目) DECM (HL)
(3行目) RET NZ
(4行目) CCF
(5行目) RET

【図 1 4 4】

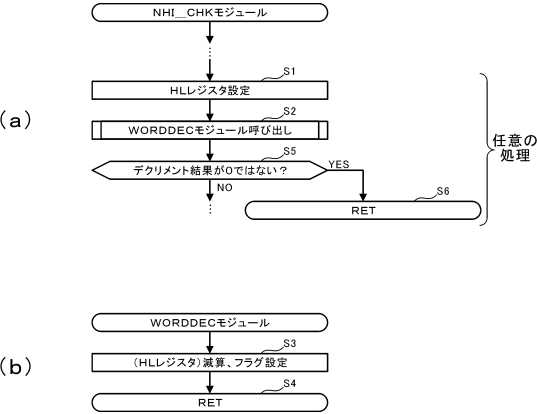
(a) (1行目) RAM_DEC:
(2行目) LD A,(HL)
(3行目) RT Z,A
(4行目) CP A,2
(5行目) DEC A
(6行目) LD (HL),A
(7行目) RET

↓

(b) (1行目) RAM_DEC:
(2行目) DECM (HL)
(3行目) RET NZ
(4行目) CCF
(5行目) RET

10

【図 1 4 5】



【図 1 4 6】

(a) (1行目) POP HL
(2行目) CALLF WORDDEC
(3行目) RET NTZ

(b) (1行目) WORDDEC:
(2行目) JTW Z,(HL),WORDDEC_99
(3行目) DECW (HL)
(4行目) RET NTZ
(5行目) SCF
(6行目) WORDDEC_99:
(7行目) RET

20

【図 1 4 7】

(a) (1行目) POP HL
(2行目) CALLF WORDDEC
(3行目) RET NTZ

(b) (1行目) WORDDEC:
(2行目) DECWM (HL)
(3行目) RET NZ
(4行目) CCF
(5行目) RET

【図 1 4 8】

(a) (1行目) CP A,n
(2行目) RET cc

↓

(b) (1行目) RCP cc,A,n

30

40

50

【図 149】

(1行目) PY_CMDA:
 (2行目) LDQ A,(LOW R_BCR_BUF)
 ⋮
 (a) (3行目) SUB A,@BCR_PAY_CLR
 (4行目) CP A,00CH
 (5行目) RET NC
 ⋮



(1行目) PY_CMDA:
 (2行目) LDQ A,(LOW R_BCR_BUF)
 ⋮
 (b) (3行目) SUB A,@BCR_PAY_CLR
 (4行目) RCP NC,A,00CH
 ⋮

【図 150】

(1行目) GAT_PAS:
 (2行目) LDQ DE,LOW R_FZ_MEM
 (3行目) LD A,(DE)
 (4行目) CP A,@FZ_MEM_MAX
 (5行目) RET NC
 ⋮



(1行目) GAT_PAS:
 (2行目) LDQ DE,LOW R_FZ_MEM
 (3行目) LD A,(DE)
 (4行目) RCP NC,A,@FZ_MEM_MAX
 ⋮

10

【図 151】

(1行目) TDN_PAS:
 ⋮
 (a) (2行目) LDQ A,(LOW R_TDN_PHS)
 (3行目) JCP Z,A,@TD_OPN_BHT,TDN_PAS_10
 (4行目) CP A,@TD_OPN_SHT
 (5行目) RET NZ
 (6行目) TDN_PAS_10:
 ⋮



(1行目) TDN_PAS:
 ⋮
 (b) (2行目) LDQ A,(LOW R_TDN_PHS)
 (3行目) JCP Z,A,@TD_OPN_BHT,TDN_PAS_10
 (4行目) RCP NZ,A,@TD_OPN_SHT
 (5行目) TDN_PAS_10:
 ⋮

【図 152】

(1行目) FD_OPN:
 ⋮
 (a) (2行目) LDQ A,(LOW R_FDN_CNT)
 (3行目) CP A,@FDN_CNT
 (4行目) RET C
 ⋮



(1行目) FD_OPN:
 ⋮
 (b) (2行目) LDQ A,(LOW R_FDN_CNT)
 (3行目) RCP C,A,@FDN_CNT
 ⋮

20

【図 153】

(1行目) TZ_STA:
 ⋮
 (a) (2行目) RST DATSEL
 (3行目) CP A,@TZ_BHT
 (4行目) RET Z
 (5行目) CP A,@TZ_SHT
 (6行目) RET Z
 ⋮



(1行目) TZ_STA:
 ⋮
 (b) (2行目) RST DATSEL
 (3行目) RCP Z,A,@TZ_BHT
 (4行目) RCP Z,A,@TZ_SHT
 ⋮

【図 154】

(1行目) TZ_RGET:
 ⋮
 (a) (2行目) LD A,(DE)
 (3行目) CP A,@TZ_MEM_MAX
 (4行目) RET NC
 ⋮



(1行目) TZ_RGET:
 ⋮
 (b) (2行目) LD A,(DE)
 (3行目) RCP NC,A,@TZ_MEM_MAX
 ⋮

30

40

50

【図 155】

(1行目) TRSVSEL:
 (a) (2行目) RST DATSEL
 (3行目) CP A,@TZ_BHT
 (4行目) RET NZ



(1行目) TRSVSEL:
 (b) (2行目) RST DATSEL
 (3行目) RCP NZ,A,@TZ_BHT

【図 156】

(1行目) TDOVCHK:
 (a) (2行目) LDQ A,(LOW R.TDN.PHS)
 (3行目) JCP NC,A,@TD_PRE_SHT,TDOVCHK_10
 (4行目) CP A,@TD_OPN_BHT
 (5行目) RET Z



(1行目) TDOVCHK:
 (b) (2行目) LDQ A,(LOW R.TDN.PHS)
 (3行目) JCP NC,A,@TD_PRE_SHT,TDOVCHK_10
 (4行目) RCP Z,A,@TD_OPN_BHT
 (5行目) TDOVCHK_10:

10

【図 157】

(1行目) BER_CHK:
 (a) (2行目) LDQ HL,LOW R_BAS_CNT
 (3行目) LD A,(HL)
 (4行目) CP A,@BAS_CHK_CNT
 (5行目) RET C



(1行目) BER_CHK:
 (b) (2行目) LDQ HL,LOW R_BAS_CNT
 (3行目) LD A,(HL)
 (4行目) RCP C,A,@BAS_CHK_CNT

【図 158】

(1行目) TEF_SEL:
 (a) (2行目) LDQ A,(LOW R.ZUG_CHK_FIX)
 (3行目) CP A,@ZUG_SML1
 (4行目) RET NC



(1行目) TEF_SEL:
 (b) (2行目) LDQ A,(LOW R.ZUG_CHK_FIX)
 (3行目) RCP NC,A,@ZUG_SML1

20

【図 159】

(1行目) SET_RIG:
 (a) (2行目) LDQ A,(LOW_HIT_NUM)
 (3行目) SUB A,10
 (4行目) CP A,57-10+1
 (5行目) RET NC
 (6行目) LDQ A,(LOW_HIT_NUM)
 (7行目) CP A,34
 (8行目) RET NC



(1行目) SET_RIG:
 (b) (2行目) LDQ A,(LOW_HIT_NUM)
 (3行目) SUB A,10
 (4行目) RCP NC,A,57-10+1
 (5行目) LDQ A,(LOW_HIT_NUM)
 (6行目) RCP NC,A,34

【図 160】

(1行目) PRE_LOT:
 (a) (2行目) LDQ A,(LOW_TRG_KND)
 (3行目) CP A,@LOT_OFS_1FK
 (4行目) RET C



(1行目) PRE_LOT:
 (b) (2行目) LDQ A,(LOW_TRG_KND)
 (3行目) RCP C,A,@LOT_OFS_1FK

30

40

50

【図 1 6 1】

(a) (1行目) REG_LOT:
(2行目) LDQ A,(LOW_TRG_KND)
(3行目) CP A,@LOT_OFS_PDJ
(4行目) RET NZ
⋮



(b) (1行目) REG_LOT:
(2行目) LDQ A,(LOW_TRG_KND)
(3行目) RCP NZ,A,@LOT_OFS_PDJ
⋮

【図 1 6 2】

(a) (1行目) BIG_SLT:
(2行目) LDQ A,(LOW_TRG_KND)
(3行目) CP A,@LOT_OFS_LCY
(4行目) RET C
⋮



(b) (1行目) BIG_SLT:
(2行目) LDQ A,(LOW_TRG_KND)
(3行目) RCP C,A,@LOT_OFS_LCY
⋮

10

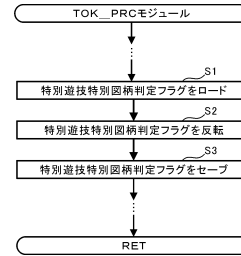
【図 1 6 3】

(a) (1行目) NAV_SET:
(2行目) LDQ A,(LOW_YRI_LMP)
(3行目) CP A,@AT_MOD_PRE
(4行目) RET C
⋮



(b) (1行目) NAV_SET:
(2行目) LDQ A,(LOW_YRI_LMP)
(3行目) RCP C,A,@AT_MOD_PRE
⋮

【図 1 6 4】



20

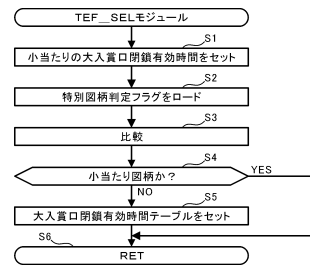
【図 1 6 5】

(a) (1行目) TOK_PRC:
(2行目) LDQ A,(LOW_R_STA_TZ)
(3行目) XOR 001H
(4行目) LDQ (LOW_R_STA_TZ),A
⋮



(b) (1行目) TOK_PRC:
(2行目) XORQ (LOW_R_STA_TZ),001H
⋮

【図 1 6 6】



30

40

50

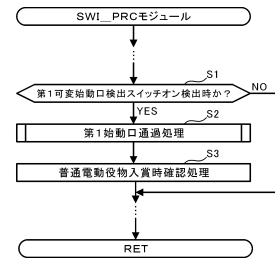
【図 1 6 7】

(1行目) TEF_SEL:
 (2行目) LD HL,@TMR_TDN_EF3
 (3行目) LDQ A,(LOW_R_ZUG_CHK_FIX)
 (4行目) CP A,@ZUG_SML1
 (5行目) RET NC
 (6行目) LDQ A,(LOW_R_TDN_FLG)
 (7行目) LD HL,D_EFF_TMR-2
 (8行目) RST WORDSEL
 (9行目) RET

↓

(1行目) TEF_SEL:
 (2行目) LD HL,@TMR_TDN_EF3
 (3行目) CPQ (LOW_R_ZUG_CHK_FIX),@ZUG_SML1
 (4行目) RET NC
 (5行目) LDQ A,(LOW_R_TDN_FLG)
 (6行目) LD HL,D_EFF_TMR-2
 (7行目) RST WORDSEL
 (8行目) RET

【図 1 6 8】



10

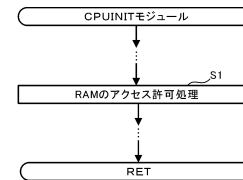
【図 1 6 9】

(1行目) SWI_PRC:
 (2行目) LDQ A,(LOW_R_IN3_PON)
 (3行目) AND A,@IN3_ST2_BIT
 (4行目) JR Z,SWI_PRC_20
 (5行目) CALLF STA_PAS
 (6行目) CALLF FDN_CHK
 (7行目) SWI_PRC_20:

↓

(1行目) SWI_PRC:
 (2行目) JTANDQ Z,(LOW_R_IN3_PON),@IN3_ST2_BIT,SWI_PRC_20
 (3行目) CALLF STA_PAS
 (4行目) CALLF FDN_CHK
 (5行目) SWI_PRC_20:

【図 1 7 0】



20

【図 1 7 1】

(1行目) CPUINIT:
 (2行目) LD A,@RAMENBL
 (3行目) OUT (@RAP____),A

↓

(1行目) CPUINIT:
 (2行目) OUT (@RAP____),@RAMENBL

【図 1 7 2】

(1行目) TMR_IPT:
 (2行目) LD A,@TOCRVAL
 (3行目) OUT (@PTOCR_),A

↓

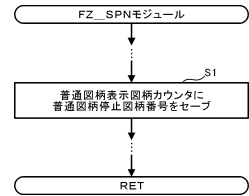
(1行目) TMR_IPT:
 (2行目) OUT (@PTOCR_),@TOCRVAL

30

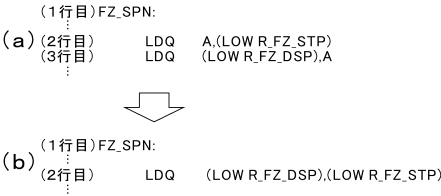
40

50

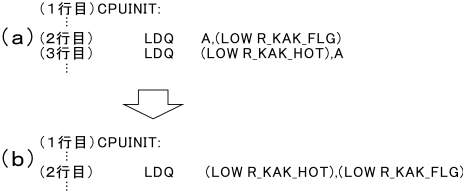
【図 1 7 3】



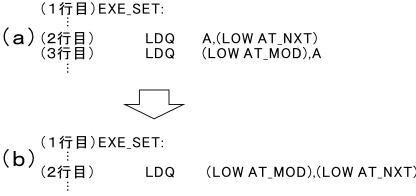
【図 1 7 4】



【図 1 7 5】



【図 1 7 6】

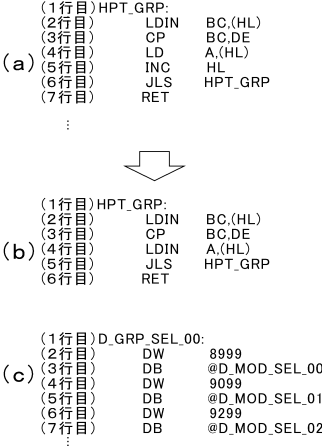


10

【図 1 7 7】



【図 1 7 8】



20

30

40

50

【図 179】

```

(1行目) E_ILGER:
(2行目) LD HL, EX_ILTM
(3行目) LD A, (HL)
(4行目) INC HL
(5行目) JT Z, A, E_ILGER01
(6行目) E_ILGER01:
↓
(1行目) E_ILGER:
(2行目) LD LDIN HL, EX_ILTM
(3行目) LDIN A, (HL)
(4行目) JT Z, A, E_ILGER01
(5行目) E_ILGER01:

```

【図 180】

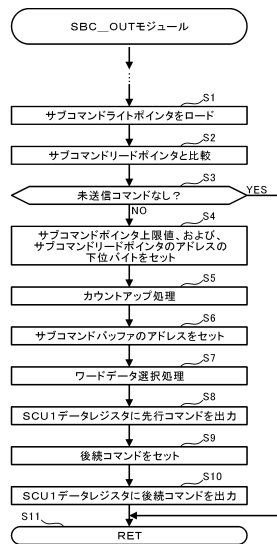
```

(1行目) E_LEVOT:
(2行目) LD HL, T_EXM_STP
(3行目) ADDWB HL, A
(4行目) LD B, 6
(5行目) E_LEVOT01:
(6行目) LD A, (HL)
(7行目) OUT (@S2DT_), A
(8行目) INC HL
(9行目) DJNZ E_LEVOT01
↓
(1行目) E_LEVOT:
(2行目) LD HL, T_EXM_STP
(3行目) ADDWB HL, A
(4行目) LD B, 6
(5行目) E_LEVOT01:
(6行目) LDIN A, (HL)
(7行目) OUT (@S2DT_), A
(8行目) DJNZ E_LEVOT01

```

10

【図 181】



【図 182】

```

(1行目) SBC_OUT:
(2行目) LDQ A, (LOW R_SBC_WPT)
(3行目) CPQ A, (LOW R_SBC_RPT)
(4行目) RET Z
(5行目) LD DE, @LMT_CMD_BFP*256+LOW R_SBC_RPT
(6行目) RST COUNTUP
(7行目) LD HL, R_SBC_SBF
(8行目) RST WORDSEL
(9行目) OUT (@S1DT_), A
(10行目) LD A, H
(11行目) OUT (@S1DT_), A
(12行目) RET
↓
(1行目) SBC_OUT:
(2行目) LDQ A, (LOW R_SBC_WPT)
(3行目) CPQ A, (LOW R_SBC_RPT)
(4行目) RET Z
(5行目) LD DE, @LMT_CMD_BFP*256+LOW R_SBC_RPT
(6行目) RST COUNTUP
(7行目) LDQP HL, LOW R_SBC_SBF
(8行目) RST WORDSEL
(9行目) OUT (@S1DT_), A
(10行目) LD A, H
(11行目) OUT (@S1DT_), A
(12行目) RET

```

20

30

40

50

【図 183】

```

(1行目) SBC_OUT:
(2行目) LDQ A,(LOW R,SBC,WPT)
(3行目) CPQ A,(LOW R,SBC,RPT)
(a) (4行目) RET Z
(5行目) LD DE,@LMT_CMD_BFP*256+LOW R,SBC_RPT
(6行目) RST COUNTUP
(7行目) LD HL,R,SBC,SBF
(8行目) RST WORDSEL
(9行目) OUT (@S1DT_),A
(10行目) LD A,H
(11行目) OUT (@S1DT_),A
(12行目) RET

```



```

(1行目) SBC_OUT:
(2行目) LDQ A,(LOW R,SBC,WPT)
(3行目) RCPQ Z,A,(LOW R,SBC,RPT)
(b) (4行目) LD DE,@LMT_CMD_BFP*256+LOW R,SBC_RPT
(5行目) RST COUNTUP
(6行目) LD HL,R,SBC,SBF
(7行目) RST WORDSEL
(8行目) OUT (@S1DT_),A
(9行目) LD A,H
(10行目) OUT (@S1DT_),A
(11行目) RET

```

【図 184】

```

(1行目) FZ_OPN:
(a) (2行目) LDQ A,(LOW R,FZN_CNT)
(3行目) CP A,@FDN_CNT
(4行目) RET C

```



```

(1行目) FZ_OPN:
(b) (2行目) RCPQ C,(LOW R,FZN_CNT),@FDN_CNT

```

10

【図 185】

```

(1行目) TD_OPN:
(a) (2行目) LDQ A,(LOW R,TDN_FLG)
(3行目) LD HL,D,TDN_MAX-1
(4行目) RST BYTESEL
(5行目) CPQ A,(LOW R,TDN_CNT)
(6行目) RET NC

```



```

(1行目) TD_OPN:
(b) (2行目) LDQ A,(LOW R,TDN_FLG)
(3行目) LD HL,D,TDN_MAX-1
(4行目) RST BYTESEL
(5行目) RCPQ NC,A,(LOW R,TDN_CNT)

```

【図 186】

```

(1行目) HSY_PRC:
(2行目) XOR A,A
(3行目) LDQ HL,LOW R,TDN_FLG
(4行目) JCP NZ,(HL),@TDK_AT2,HSY_PRC_10
(a) (5行目) LDQ HL,LOW R,ZUG_CHK_FIX
(6行目) JCP NZ,(HL),@ZUG_SML1,HSY_PRC_20

```

(7行目) HSY_PRC_10:

(8行目) HSY_PRC_20:

20



```

(1行目) HSY_PRC:
(b) (2行目) XOR A,A
(3行目) JCPQ NZ,(LOW R,TDN_FLG),@TDK_AT2,HSY_PRC_10
(4行目) JCPQ NZ,(LOW R,ZUG_CHK_FIX),@ZUG_SML1,HSY_PRC_20

```

(5行目) HSY_PRC_10:

(6行目) HSY_PRC_20:

【図 187】

```

(1行目) FDN_CHK:
(2行目) LDQ HL,(LOW R,FZN_EFF)
(3行目) RET NTZ
(a) (4行目) LDQ A,(LOW R,FUT_PHS)
(5行目) JCP NZ,A,@FD_OPN,FDN_CHK_10
(6行目) INCQ (LOW R,FZN_CNT)
(7行目) RET
(8行目) FDN_CHK_10:

```



```

(1行目) FDN_CHK:
(b) (2行目) LDQ HL,(LOW R,FZN_EFF)
(3行目) RET NTZ
(4行目) JCPQ NZ,(LOW R,FUT_PHS),@FD_OPN,FDN_CHK_10
(5行目) INCQ (LOW R,FZN_CNT)
(6行目) RET
(7行目) FDN_CHK_10:

```

【図 188】

```

(1行目) FZ_STP:
(2行目) RET NTZ
(3行目) LDQ A,(LOW R,FZ_ATA)
(a) (4行目) JCP Z,A,@FZ_ATA,FZ_STP_10
(5行目) CLRQ (LOW R,FUT_PHS)
(6行目) CLRQ (LOW R,FZ_STP)
(7行目) RET
(8行目) FZ_STP_10:

```



```

(1行目) FZ_STP:
(b) (2行目) RET NTZ
(3行目) JCPQ Z,(LOW R,FZ_ATA),@FZ_ATA,FZ_STP_10
(4行目) CLRQ (LOW R,FUT_PHS)
(5行目) CLRQ (LOW R,FZ_STP)
(6行目) RET
(7行目) FZ_STP_10:

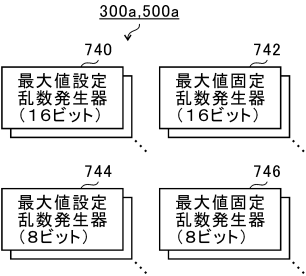
```

30

40

50

【図 1 8 9】



【図 1 9 0】

(a)

乱数	範囲	乱数生成部
大当たり決定乱数	0～65535	最大値設定乱数発生器(16ビット)
当たり図柄乱数	0～99	ソフトウェア乱数生成部(8ビット)
リーチグループ決定乱数	0～10006	最大値設定乱数発生器(16ビット)
リーチモード決定乱数	0～250	ソフトウェア乱数生成部(8ビット)
変動パターン乱数	0～238	ソフトウェア乱数生成部(8ビット)
当たり決定乱数	0～99	最大値設定乱数発生器(8ビット)
普図図柄乱数	0～99	ソフトウェア乱数生成部(8ビット)
普図変動パターン乱数	0～99	ソフトウェア乱数生成部(8ビット)

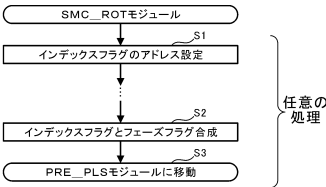
(b)

乱数	範囲	乱数生成部
大当たり決定乱数	0～65535	最大値設定乱数発生器(16ビット)
当たり図柄乱数	0～99	最大値設定乱数発生器(8ビット)
リーチグループ決定乱数	0～10006	最大値設定乱数発生器(16ビット)
リーチモード決定乱数	0～1000	最大値設定乱数発生器(16ビット)
変動パターン乱数	0～255	最大値設定乱数発生器(8ビット)
当たり決定乱数	0～65535	最大値設定乱数発生器(16ビット)
普図図柄乱数	0～99	最大値設定乱数発生器(8ビット)
普図変動パターン乱数	0～255	最大値固定乱数発生器(8ビット)

(c)

乱数	範囲	乱数生成部
大当たり決定乱数	0～65535	最大値固定乱数発生器(16ビット)
当たり図柄乱数	0～99	最大値設定乱数発生器(8ビット)
リーチグループ決定乱数	0～10006	最大値設定乱数発生器(16ビット)
リーチモード決定乱数	0～1000	最大値設定乱数発生器(16ビット)
変動パターン乱数	0～255	最大値固定乱数発生器(8ビット)
当たり決定乱数	0～65535	最大値固定乱数発生器(16ビット)
普図図柄乱数	0～99	最大値設定乱数発生器(8ビット)
普図変動パターン乱数	0～255	最大値固定乱数発生器(8ビット)

【図 1 9 1】



【図 1 9 2】

(1行目) SMC_ROT:

(2行目) LDQ HL, LOW_IDX_FND

⋮

(3行目) LD A, (HL)

(4行目) OR A, C

(5行目) LD (HL), A

(6行目) JP PRE_PLS

10

20

30

40

50

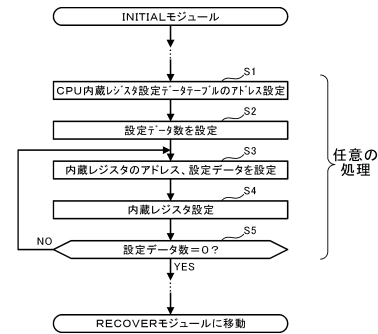
【図 193】

```

(1行目) SMC_ROT:
(2行目) LDQ      HL,LOW_IDX_FND
(3行目) OR      (HL),C
(4行目) JP      PRE_PL5

```

【図 194】



10

【図 195】

```

(1行目) INITIAL:
(2行目) LD      HL,T_CPU_REG
(3行目) LD      B,@NUM_CPU_REG
(a) (4行目) INITIAL01:
(5行目) LDIN   AC,(HL)
(6行目) OUT    (C),A
(7行目) DJNZ   INITIAL01

```

```

(1行目) T_CPU_REG:
(2行目) DEFB   @PT0PR_
(3行目) DEFB   120
(4行目) DEFB   @PT0CR
(5行目) DEFB   @T0CRVAL
(6行目) DEFB   @PT0SV_
(7行目) DEFB   149
(8行目) DEFB   @IMR_
(9行目) DEFB   @IMRVAL_
(10行目) DEFB   @S1BR_+0
(b) (11行目) DEFB   8
(12行目) DEFB   @S1FM_
(13行目) DEFB   10001000B
(14行目) DEFB   @S1CMS_
(15行目) DEFB   00000010B
(16行目) DEFB   @S2BR_+0
(17行目) DEFB   13
(18行目) DEFB   @S2FM_
(19行目) DEFB   10001000B
(20行目) DEFB   @S2CMS_
(21行目) DEFB   00000010B
(22行目) @NUM_CPU_REG EQU ($-T_CPU_REG)/2

```

【図 196】

```

(1行目) INITIAL:
(2行目) LDF    HL,T_CPU_REG
(3行目) LD      B,@NUM_CPU_REG
(a) (4行目) INITIAL01:
(5行目) LDIN   AC,(HL)
(6行目) OUT    (C),A
(7行目) DJNZ   INITIAL01

```

```

(1行目) T_CPU_REG:
(2行目) DEFB   @PT0PR_
(3行目) DEFB   120
(4行目) DEFB   @PT0CR
(5行目) DEFB   @T0CRVAL
(6行目) DEFB   @PT0SV_
(7行目) DEFB   149
(8行目) DEFB   @IMR_
(9行目) DEFB   @IMRVAL_
(10行目) DEFB   @S1BR_+0
(b) (11行目) DEFB   8
(12行目) DEFB   @S1FM_
(13行目) DEFB   10001000B
(14行目) DEFB   @S1CMS_
(15行目) DEFB   00000010B
(16行目) DEFB   @S2BR_+0
(17行目) DEFB   13
(18行目) DEFB   @S2FM_
(19行目) DEFB   10001000B
(20行目) DEFB   @S2CMS_
(21行目) DEFB   00000010B
(22行目) @NUM_CPU_REG EQU ($-T_CPU_REG)/2

```

20

【図 197】

```

(1行目) RANKSET:
(a) (2行目) LD      HL,T_RNK_SET
(3行目) RST     TABLSET

```



```

(1行目) RANKSET:
(b) (2行目) LDF    HL,T_RNK_SET
(3行目) RST     TABLSET

```

【図 198】

```

(1行目) PWRFAIL:
(2行目) LD      HL,D_OUT_PRT
(3行目) LD      B,@PRT_CLR_LOP
(a) (4行目) PWRFAIL_20:
(5行目) LD      C,(HL)
(6行目) OUT    (C),A
(7行目) INC    HL
(8行目) DJNZ   PWRFAIL_20

```



```

(1行目) PWRFAIL:
(2行目) LDF    HL,D_OUT_PRT
(3行目) LD      B,@PRT_CLR_LOP
(b) (4行目) PWRFAIL_20:
(5行目) LD      C,(HL)
(6行目) OUT    (C),A
(7行目) INC    HL
(8行目) DJNZ   PWRFAIL_20

```

30

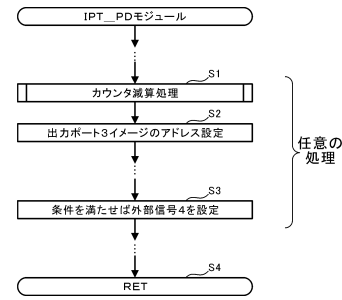
40

50

【図 199】

(1行目) DYM_OUT:
⋮
(a) (2行目) LD HL,D_DYM_SEL
(3行目) RST WORDSEL
⋮
↓
(1行目) DYM_OUT:
⋮
(b) (2行目) LDF HL,D_DYM_SEL
(3行目) RST WORDSEL
⋮

【図 200】



10

【図 201】

(1行目) IPT_PD:
⋮
(2行目) RST RAM_DEC
(3行目) LDQ HL,LOW_OUT_PT3
⋮
(4行目) JR Z,IPT_PD03
(5行目) SET @EXT_SG4_POS,(HL)
(6行目) IPT_PD03:
⋮
(7行目) RET

【図 202】

(1行目) IPT_PD:
⋮
(2行目) RST RAM_DEC
(3行目) LDQ HL,LOW_OUT_PT3
⋮
(4行目) SET NZ,@EXT_SG4_POS,(HL)
(5行目) RET

【図 203】

(1行目) DYNMOUT:
⋮
(a) (2行目) RESQ @CHN_DSP_BIT,(LOW_WIN_DAT+1)
(3行目) LDQ A,(LOW_OUT_PT2)
(4行目) JAND Z,A,00000011B,DYNMOUT02
(5行目) SETQ @CHN_DSP_BIT,(LOW_WIN_DAT+1)
(6行目) DYNMOUT02:
⋮
↓
(1行目) DYNMOUT:
⋮
(b) (2行目) LDQ HL,LOW_WIN_DAT+1
(3行目) RES @CHN_DSP_BIT,(HL)
(4行目) LDQ A,(LOW_OUT_PT2)
(5行目) AND A,00000011B
(6行目) SET NZ,@CHN_DSP_BIT,(HL)
⋮

【図 204】

(1行目) EXT_PRC:
⋮
(a) (2行目) LD BC,@D_SEC_LOP*256+0
(3行目) JR Z,EXT_PRC_15
(4行目) SET @EXT_SEC_POS,C
(5行目) EXT_PRC_15:
⋮
↓
(1行目) EXT_PRC:
⋮
(b) (2行目) LD BC,@D_SEC_LOP*256+0
(3行目) SET NZ,@EXT_SEC_POS,C
⋮

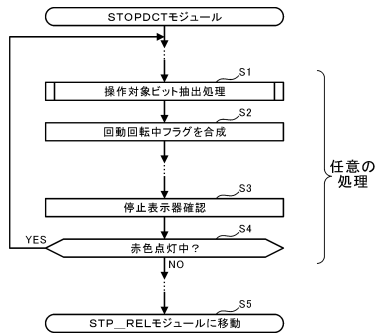
20

30

40

50

【図 205】



【図 206】

```

(1行目) STOPDCT:
(2行目)      CALLF  SMPLBIT
(3行目)      AND    A,D
(4行目)      ANDQ   A,(LOW_OUT_PT4)
(5行目)      JR     Z,STOPDCT
  
```

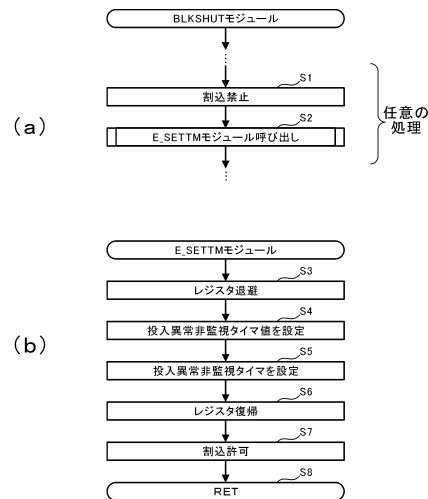
10

【図 207】

```

(1行目) STOPDCT:
(2行目)      CALLF  SMPLBIT
(3行目)      AND    A,D
(4行目)      JANDQ  Z,A,(LOW_OUT_PT4),STOPDCT
  
```

【図 208】



20

【図 209】

```

(a) (1行目)      DI
      (2行目)      CALL  E_SETTM
  
```

```

(b) (1行目) E_SETTM:
      (2行目)      EX    AF,AF'
      (3行目)      EXX
      (4行目)      LD    A,@MDL_WAT_TMR
      (5行目)      LD    (.EX_SLTM),A
      (6行目)      EXX
      (7行目)      EX    AF,AF'
      (8行目)      EI
      (9行目)      RET
  
```

【図 210】

```

(a) (1行目)      CALLEX  E_SETTM
  
```

```

(b) (1行目) E_SETTM:
      (2行目)      LD    A,@MDL_WAT_TMR
      (3行目)      LD    (.EX_SLTM),A
      (4行目)      RETEX
  
```

30

40

50

【図 2 1 1】

(a) (1行目) DI
(2行目) CALL E_DMOUT

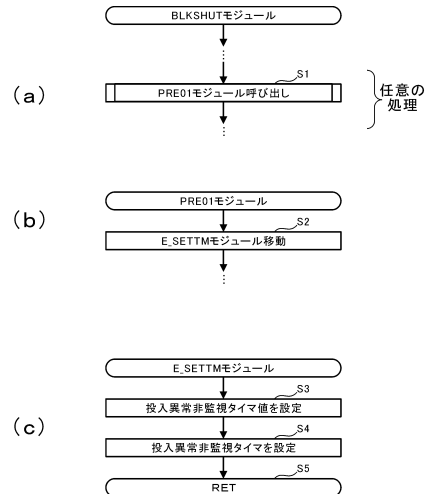
(b) (1行目) E_DMOUT:
(2行目) EX AF,AF'
(3行目) EXX
(4行目) LD HL,R.ERW_IOB
(5行目) LD A,(R.COM_CNT)
(6行目) LD A,(HL+A)
(7行目) OUT (@OTC_PRT),A
(8行目) EXX
(9行目) EX AF,AF'
(10行目) EI
(11行目) RET

【図 2 1 2】

(a) (1行目) CALLEX E_DMOUT

(b) (1行目) E_DMOUT:
(2行目) LD HL,R.ERW_IOB
(3行目) LD A,(R.COM_CNT)
(4行目) LD A,(HL+A)
(5行目) OUT (@OTC_PRT),A
(6行目) RETEX

【図 2 1 3】



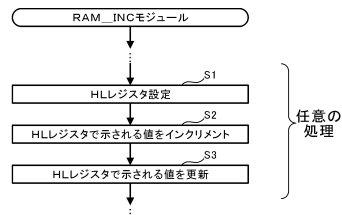
【図 2 1 4】

(a) (1行目) CALLEX PRE01

(b) (1行目) PRE01:
(2行目) JP E_SETTM

(c) (1行目) E_SETTM:
(2行目) LD A,@MDL_WAT_TMR
(3行目) LD (EX_SLTM),A
(4行目) RETEX

【図 2 1 5】



【図 2 1 6】

(1行目) RAM_INC:
(2行目) LDQ HL,LOW_AT_SET
(3行目) LD A,(HL)
(4行目) INC A
(5行目) JCP C,A,5,RAM_INC01
(6行目) LD A,5
(7行目) RAM_INC01:
(8行目) LD (HL),A

10

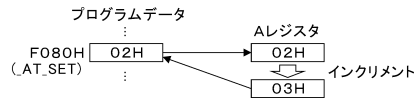
20

30

40

50

【図 2 1 7】



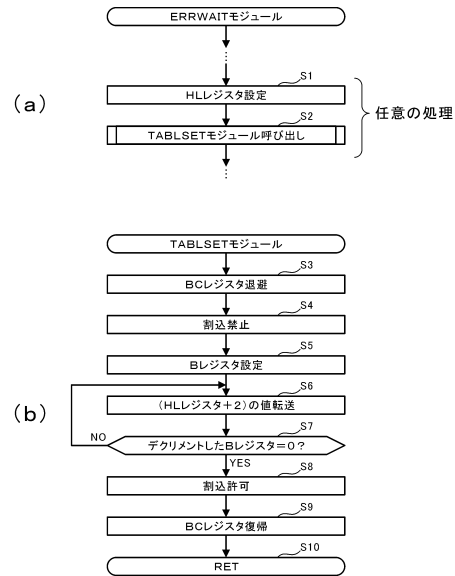
【図 2 1 8】

(1行目)RAM_INC:
(2行目) ICPLMQ (LOW_AT_SET),5

【図 2 1 9】

(1行目)RAM_INC:
(2行目) ICPLMQ (LOW_AT_SET),5

【図 2 2 0】



【図 2 2 1】

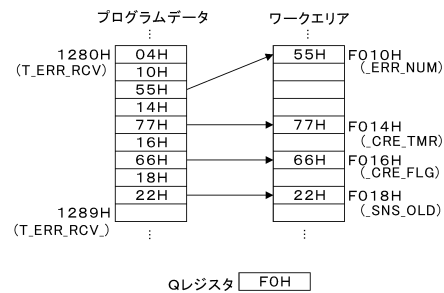
(a) (1行目) LD HL,T_ERR_RCV
(2行目) RST TABLSET

(b) (1行目) TABLSET:
(2行目) PUSH BC
(3行目) DI
(4行目) LD B,(HL)
(5行目) TABLSET10:
(6行目) INLD AC,(HL)
(7行目) LDQ (C),A
(8行目) DJNZ TABLSET10
(9行目) EI
(10行目) POP BC
(11行目) RET

(c) (1行目) T_ERR_RCV:
(2行目) DEFB (T_ERR_RCV-T_ERR_RCV)/2
(3行目) DEFB LOW_ERR_NUM ,0
(4行目) DEFB LOW_CRE_TMR ,0
(5行目) DEFB LOW_CRE_FLG ,0
(6行目) DEFB LOW_SNS_OLD ,0
(7行目) T_ERR_RCV,:

(d) (1行目) ERR_NUM DEFS 1
(2行目) CRE_TMR DEFS 1
(3行目) CRE_FLG DEFS 1
(4行目) SNS_OLD DEFS 1

【図 2 2 2】



10

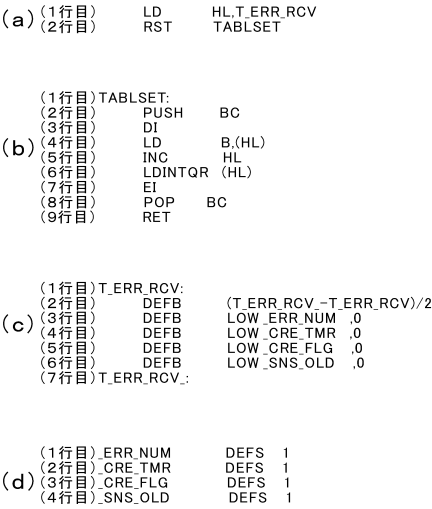
20

30

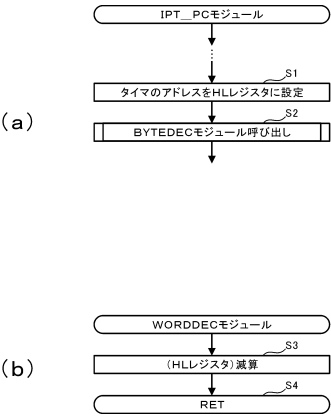
40

50

【図 2 2 3】

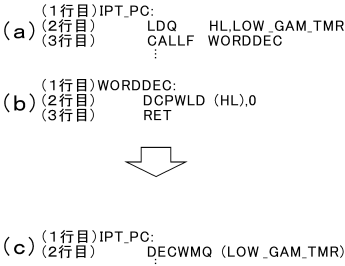


【図 2 2 4】

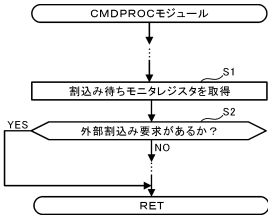


10

【図 2 2 5】

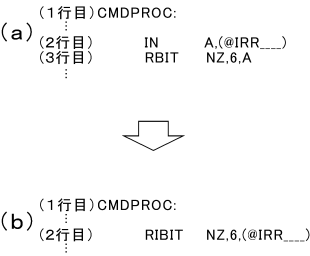


【図 2 2 6】

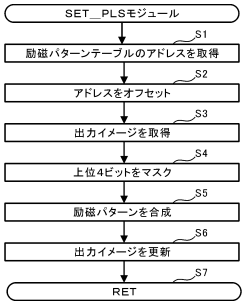


20

【図 2 2 7】



【図 2 2 8】



30

40

50

【図 2 2 9】

```

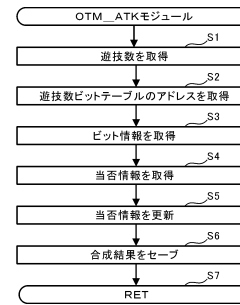
(1行目) SET_PLS:
(2行目) LD HL,T_PLS_PTN
(3行目) ADDWB HL,A
(4行目) LD A,(IY+@OFS_OUT_PRT)
(5行目) AND A,00001111B
(6行目) OR A,(HL)
(7行目) LD (IY+@OFS_OUT_PRT),A
(8行目) RET

(1行目) SET_PLS:
(2行目) LD HL,T_PLS_PTN
(3行目) RST CALADRS
(4行目) AND (IY+@OFS_OUT_PRT),00001111B
(5行目) OR (IY+@OFS_OUT_PRT),A
(6行目) RET

(1行目) CALADRS:
(2行目) ADDWB HL,A
(3行目) LD A,(HL)
(4行目) RET

```

【図 2 3 0】



10

【図 2 3 1】

```

(1行目) OTM_ATK:
(2行目) LDQ A,(LOW_CZ_CNT)
(3行目) LD HL,T_XXX_XXX
(4行目) RST CALADRS
(5行目) LDQ B,(LOW_ATW_BIT)
(6行目) OR A,B
(7行目) LDQ A,(LOW_ATW_BIT),A
(8行目) RET

(1行目) T_XXX_XXX:
(2行目) DEFB 00000001B
(3行目) DEFB 00000010B
(4行目) DEFB 00000100B
(5行目) DEFB 00001000B
(6行目) DEFB 00010000B
(7行目) DEFB 00100000B
(8行目) DEFB 01000000B
(9行目) DEFB 10000000B

(1行目) OTM_ATK:
(2行目) LDQ A,(LOW_CZ_CNT)
(3行目) CPLBQ A,(LOW_ATW_BIT)
(4行目) RET

```

【図 2 3 2】

```

(1行目) KRS_JDG:
(2行目) LDQ A,(LOW_R_ROU_CNT)
(3行目) LD B,(HL)
(4行目) KRS_JDG_10:
(5行目) INC HL
(6行目) CP A,(HL)
(7行目) INC HL
(8行目) LD C,(HL)
(9行目) RET Z
(10行目) DJNZ KRS_JDG_10
(11行目) RET

(1行目) D_KRS_JDG_2:
(2行目) DB (@D_KRS_JDG_2-D_KRS_JDG2-1)/2
(3行目) DB @KRS_TGT_ROU_01
(4行目) DB @KRS_JDG_01
(5行目) DB @KRS_TGT_ROU_03
(6行目) DB @KRS_JDG_03
(7行目) @D_KRS_JDG_2:

```

20

【図 2 3 3】

```

(1行目) KRS_JDG:
(2行目) LDQ A,(LOW_R_ROU_CNT)
(3行目) LD B,(HL)
(4行目) KRS_JDG_10
(5行目) INC HL
(6行目) LDSB 2,DE,(HL)
(7行目) CP A,D
(8行目) RET Z
(9行目) DJNZ KRS_JDG_10
(10行目) RET

```

【図 2 3 4】

行数	コマンド
1	EX ALL
2	LD Q, 0F0H
3	EX ALL

30

```

(1行目) D_KRS_JDG_2:
(2行目) DB (@D_KRS_JDG_2-D_KRS_JDG2-1)/2
(3行目) DB @KRS_TGT_ROU_01*8+@KRS_JDG_01
(4行目) DB @KRS_TGT_ROU_03*8+@KRS_JDG_03
(5行目) @D_KRS_JDG_2:

```

40

50

【図 2 3 5】

(a)

行数	コマンド
1	TMR_IPT:
2	EX ALL
3	LD Q.0F0H
4	EX ALL
5	EI
6	RETI
7	

(b)

行数	コマンド
1	TMR_IPT:
2	EX ALL
3	LD Q.0F0H
4	EX ALL
5	EI
6	RETI
7	

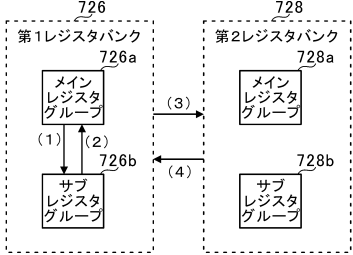
【図 2 3 6】

行数	コマンド
1	EX ALL
2	LD Q.0F0H

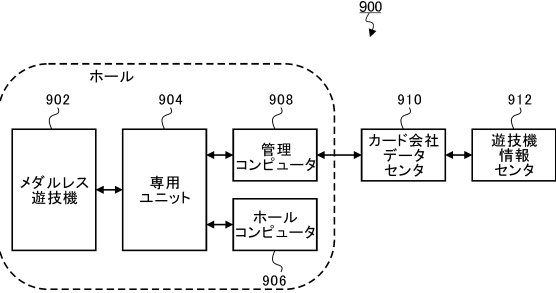
【図 2 3 7】

行数	コマンド
1	TMR_IPT:
2	LD Q.0F0H
3	EX ALL
4	EX ALL
5	EI
6	RETI
7	

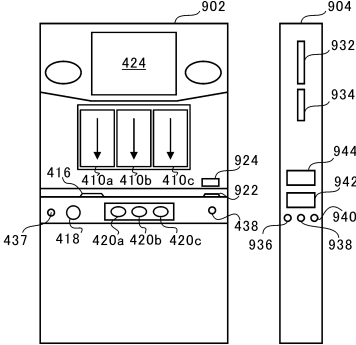
【図 2 3 8】



【図 2 3 9】



【図 2 4 0】



10

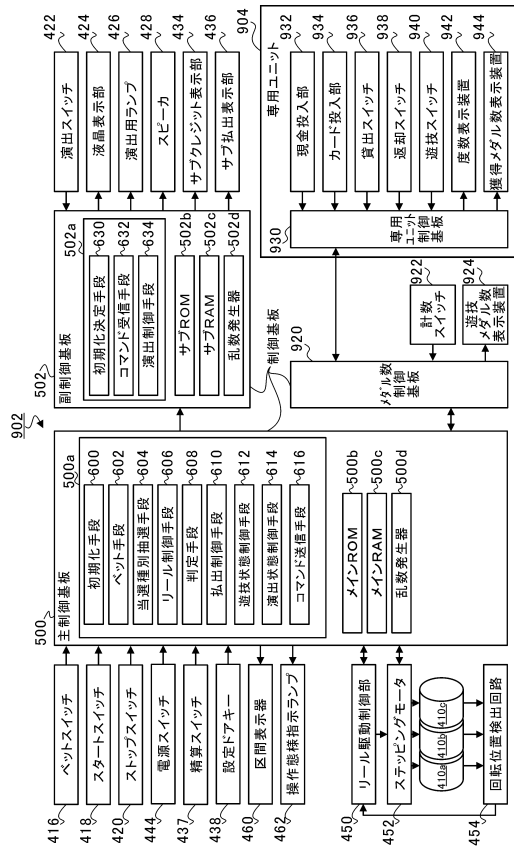
20

30

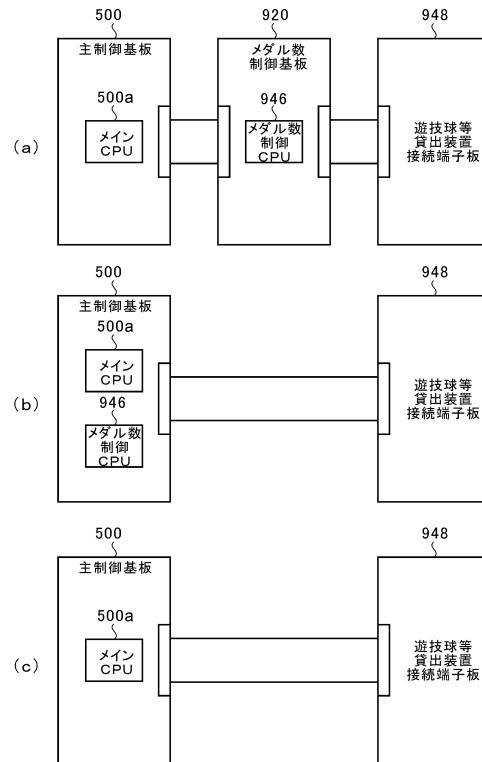
40

50

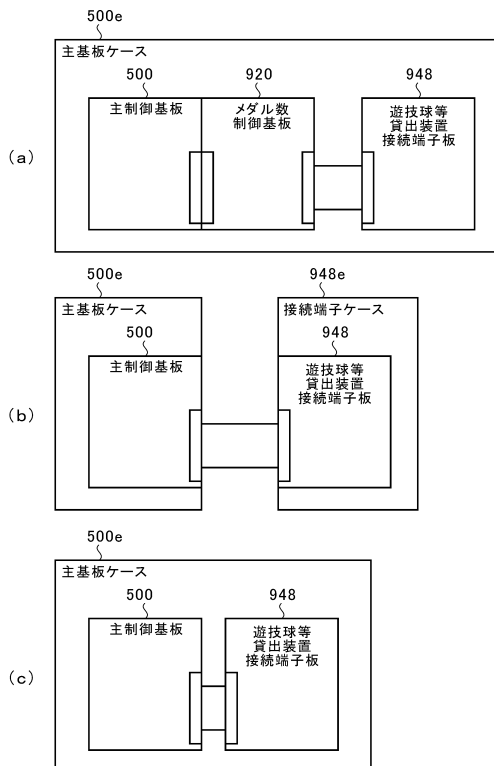
【図 2 4 1】



【図 2 4 2】



【図 2 4 3】



【図 2 4 4】

No	機能	メインCPU		メダル数制御CPU	
		使用領域	使用外領域	使用領域	使用外領域
1	遊技の実行を制御する機能	○	×	×	×
2	回転制御	○	×	×	×
3	遊技メダルの投入・積算	○	×	×	×
4	回転回数の検出	○	×	×	×
5	内部抽籤用乱数生成	○	×	×	×
6	内部抽籤による抽籤動作	○	×	×	×
7	回転停止装置の動作	○	×	×	×
8	表示された図柄の組合せを判定	○	×	×	×
9	入賞となる遊技メダルの払い出し	○	×	×	×
10	前遊技・後遊技・後遊技抽籤動作を制御	○	×	×	×
11	その他、遊技の実行に係る制御	○	×	×	×
12	遊技メダル数表示装置を制御する機能	○	×	○	×
13	遊技メダル数を電磁的に記憶	○	×	○	×
14	遊技メダル数を表示	○	×	○	×
15	払出された遊技メダルの数を加算	○	×	○	×
16	払出された遊技メダルの数を減算	○	×	○	×
17	投入された遊技メダルの数を減算	○	×	○	×
18	積算された遊技メダルの数を加算	○	×	○	×
19	払出された遊技メダルの数を加算	○	×	○	×
20	遊技メダル数リニアポタンにより遊技メダル数を初期化	○	×	○	×
21	遊技球等貸出装置との連係を制御する機能	○	×	○	×
22	VFL監視装置	○	×	○	×
23	遊技制御・遊タイム管理	○	×	○	×
24	遊技機性能通知(遊技機性能情報)電文生成・送信	○	×	○	×
25	遊技機性能通知(遊技機設置情報)電文生成・送信	○	×	○	×
26	遊技機性能通知(メダル不正監視情報)電文生成・送信	○	×	○	×
27	計数通知電文生成・送信	○	×	○	×
28	貸出通知受信	○	×	○	×
29	貸出受領結果電文生成・送信	○	×	○	×
30	主制御CPU・メダル数制御CPU間通信制御	○	×	○	×
31	遊技機性能情報の集計	○	×	○	×
32	総投入枚数	○	×	○	×
33	総払出枚数	○	×	○	×
34	MY	○	×	○	×
35	役物総払出枚数	○	×	○	×
36	連勝役物総払出枚数	○	×	○	×
37	役物比率	○	×	○	×
38	連勝役物比率	○	×	○	×
39	有利区間比率	○	×	○	×
40	前示込役物比率	○	×	○	×
41	役物等状態比率	○	×	○	×
42	前示回数	○	×	○	×
43	遊技機設置情報の生成	○	×	○	×
44	主制御チップの警告	○	×	○	×
45	主制御チップメモリーコード	○	×	○	×
46	主制御チップ製造コード	○	×	○	×
47	メダル数制御チップの警告	○	×	○	×
48	メダル数制御チップ製造コード	○	×	○	×
49	メダル数制御チップ製造コード	○	×	○	×
50	メダル不正監視情報の生成	○	×	○	×
51	遊技メダル数	○	×	○	×
52	投入メダル数	○	×	○	×
53	払出メダル数	○	×	○	×
54	前示回数	○	×	○	×
55	主制御状態2	○	×	○	×
56	遊技機エラー状態	○	×	○	×
57	遊技機不正1	○	×	○	×
58	遊技機不正2	○	×	○	×
59	規定数(IN)	○	×	○	×
60	払出枚数(OUT)	○	×	○	×
61	表示装置を制御する機能	○	×	○	×
62	周辺基板に送出に係る情報を送信する機能	○	×	○	×
63	周辺基板に遊技機の情報を送信する機能	○	×	○	×
64	不正監視(Nos65, Nos66を除く)	○	×	○	×
65	遊技メダルの払い出しに係る不正監視	○	×	○	×
66	遊技メダルの払い出しに係る不正監視	○	×	○	×
67	エラー処理(遊技機停止)	○	×	○	×
68	エラー処理(遊技機停止)	○	×	○	×
69	役物モニタの表示を制御する機能	○	×	○	×
70	遊技機の試験に必要な信号を生成する機能	○	×	○	×

10

20

30

40

50

【図 2 4 5】

遊技機情報通知			
バイト	データ名	バイト数	内容
1	電文長	1	12h～39h
2	コマンド	1	01h
3	通番	1	シーケンス番号 00h～FFh
4	遊技機種類	1	遊技機の種類
5	遊技機情報種別	1	00h～02h
6～	遊技機情報		
	遊技機性能情報	51	総投入枚数、総払出枚数、MYなど
	遊技機設置情報	40	主制御/メダル数制御のメーカーコード、製品コード、チップID番号
	ホールコン・不正監視情報	12～16	IN、OUT、BB、RB、AT、ドアオープン信号、セキュリティ信号など
最終チェックサム		1	

【図 2 4 6】

遊技機性能情報（遊技機情報）			
バイト	データ名	バイト数	内容
6～	総投入枚数	3	電源ONから累積した投入枚数（再遊技は含まない）
	総払出枚数	3	電源ONから累積した払出枚数（再遊技は含まない）
	MY	3	電源ONから累積した遊技回数
	役物総払出枚数	3	電源ONから累積した役物の作動による払出枚数
	連続役物総払出枚数	3	電源ONから累積した第一種特別役物の作動による払出枚数
	役物比率	1	役比モニタの情報（非該当／規定遊技数未達時はFFh）
	連続役物比率	1	役比モニタの情報（非該当／規定遊技数未達時はFFh）
	有利区間比率	1	役比モニタの情報（非該当／規定遊技数未達時はFFh）
	指示込役物比率	1	役比モニタの情報（非該当／規定遊技数未達時はFFh）
	役物等状態比率	1	役比モニタの情報（規定遊技数未達時はFFh）
	遊技回数	2	電源ON以降算出される最大MY
	予備	24	予備 1バイト×24（使用しない場合00h固定）
	予約1	3	専用ユニットで使用（00h固定）（専用ユニットで算出される最大MY）
	予約2	2	専用ユニットで使用（00h固定）（専用ユニットで算出される遊技回数）

【図 2 4 7】

遊技機設置情報（遊技機情報）			
バイト	データ名	バイト数	内容
6～	主制御チップID番号	9	遊技機の主制御チップID番号
	主制御チップメーカーコード	3	主制御の管理エリアに記載したメーカーコード
	主制御チップ製品コード	8	主制御の管理エリアに記載した製品コード
	メダル数制御チップID番号	9	遊技機のメダル数制御チップID番号（非搭載時は0）
	メダル数制御チップメーカーコード	3	メダル数制御の管理エリアに記載したメーカーコード（非搭載時は0）
	メダル数制御チップ製品コード	8	メダル数制御の管理エリアに記載した製品コード（非搭載時は0）

【図 2 4 8】

ホールコン・不正監視情報（遊技機情報）			
バイト	データ名	バイト数	内容
6～	遊技メダル数	3	遊技メダル数表示装置に記録されている現在の遊技メダル数 000000h～FF3F00h(0～16383)
	投入メダル数	1	投入した電子化メダルの枚数 F0h～03h（-3枚～3枚）
	払出メダル数	1	払い出された電子化メダルの枚数 00h～0Fh（0枚～15枚）
	主制御状態1	1	Bit0:RB（従来外部信号CN2-3） Bit1:RB（従来外部信号CN2-4） Bit2:AT（従来外部信号CN2-5） Bit3:遊技機状態信号1 Bit4:遊技機状態信号2 Bit5:遊技機状態信号3 Bit6:遊技機状態信号4 Bit7:未使用
	主制御状態2	1	Bit0:遊技機状態信号5 Bit1:遊技機状態信号6 Bit2:遊技機状態信号7 Bit3:未使用 Bit4:未使用 Bit5:未使用 Bit6:未使用 Bit7:未使用
	遊技機エラー状態	1	遊技機で発生中のエラーコード Bit0～5:エラーコード 000000＝エラー無 Bit6:0＝メダル数制御、1＝主制御 Bit7:0＝発報のみ、1＝発報＋HIC出力
	遊技機不正1(主制御)	1	Bit0:設定変更中信号 Bit1:設定確認中信号 Bit2:不正検知信号1 Bit3:不正検知信号2 Bit4:不正検知信号3 Bit5:セキュリティ信号（従来外部信号CN2-7） Bit6:未使用 Bit7:未使用
	遊技機不正2(主制御又はメダル数制御)	1	Bit0:設定ドアオープン信号 Bit1:ドアオープン信号（従来外部信号CN2-8） Bit2:未使用（使用禁止） Bit3:遊技メダル数クリア検知 Bit4:未使用（使用禁止） Bit5:未使用（使用禁止） Bit6:未使用 Bit7:未使用
	遊技機不正3(主制御又はメダル数制御)	1	Bit0:未使用（使用禁止） Bit1:未使用（使用禁止） Bit2:未使用 Bit3:未使用 Bit4:未使用 Bit5:未使用 Bit6:未使用 Bit7:未使用
	遊技情報数	1	種別情報・カウント情報の個数(n) n=00h～02h(0個～2個)
	種別情報1	1	種別情報1
	カウント情報1	1	カウント情報1
	種別情報2	1	種別情報2
	カウント情報2	1	カウント情報2

10

20

30

40

50

【図 2 4 9】

計数通知

バイト	データ名	バイト数	内容
1	電文長	1	07h
2	コマンド	1	02h
3	計数通番	1	シーケンス番号 00h~FFh(0~255)
4	計数メダル数	1	計数メダル数 00h~32h(0~50)
5	計数累積メダル数	2	計数累積メダル数 0000h~FFFFh(0~65535)
7	チェックサム	1	

【図 2 5 1】

貸出通知

バイト	データ名	バイト数	内容
1	電文長	1	05h
2	コマンド	1	13h
3	貸出通番	1	シーケンス番号 0x00~0xFF(0~255)
4	貸出メダル数	1	電子化メダルの貸出メダル数 00h~32h(0~50)
5	チェックサム	1	

【図 2 5 3】

遊技機情報通知

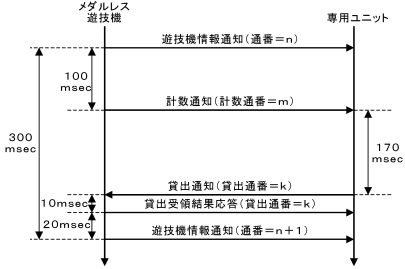
優先順位	遊技機情報	通知タイミング
1	ホールコン・不正監視情報 (主制御状態の更新あり) (遊技情報あり)	メダルレス遊技機起動完了から 300msec周期で通知
2	遊技機設置情報	メダルレス遊技機起動完了から 60sec経過後、60sec 周期で通知
3	遊技機性能情報	メダルレス遊技機起動完了から 180sec経過後、180sec 周期で通知
4	ホールコン・不正監視情報 (主制御状態の更新なし) (遊技情報なし)	メダルレス遊技機起動完了から 300msec周期で通知

【図 2 5 0】

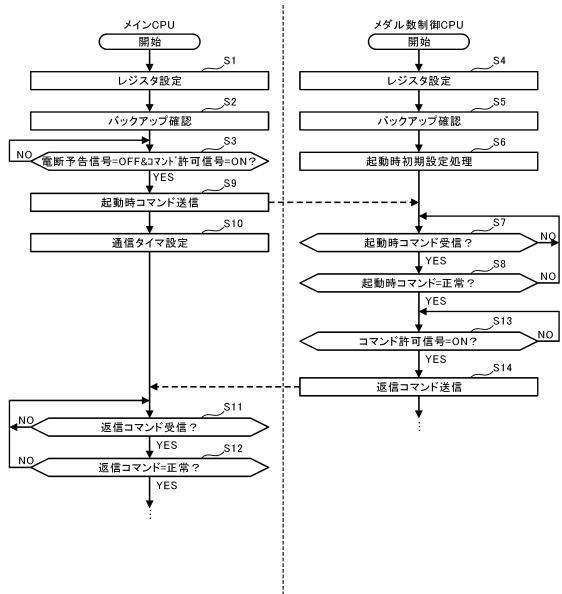
貸出受領結果応答

バイト	データ名	バイト数	内容
1	電文長	1	05h
2	コマンド	1	03h
3	貸出通番	1	シーケンス番号 0x00~0xFF(0~255)
4	貸出メダル数受領結果	1	00h=正常 01h=異常
5	チェックサム	1	

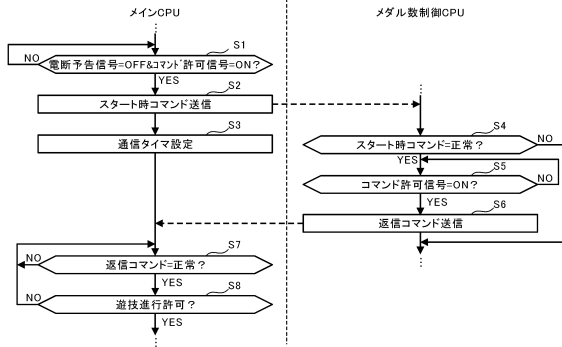
【図 2 5 2】



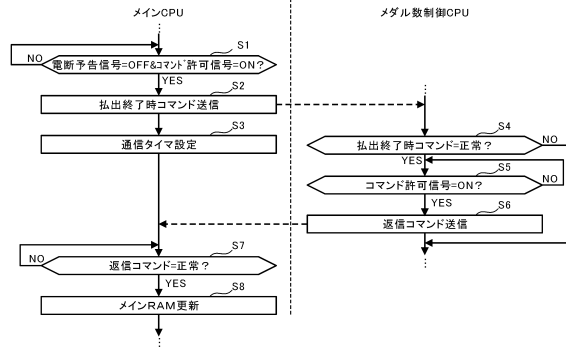
【図 2 5 4】



【図 255】

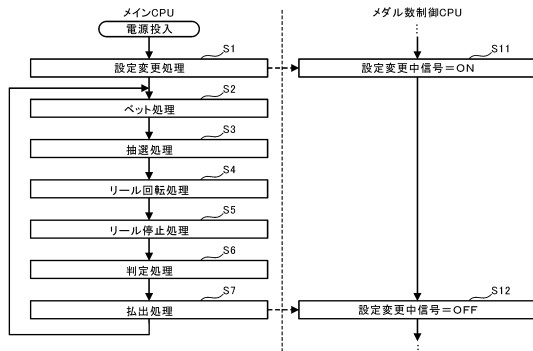


【図 256】

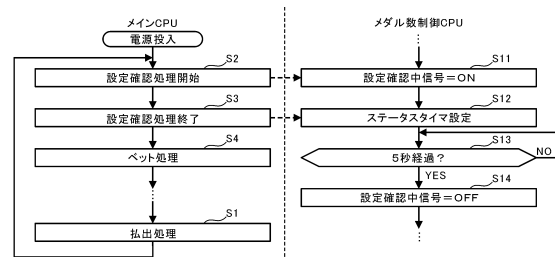


10

【図 257】



【図 258】



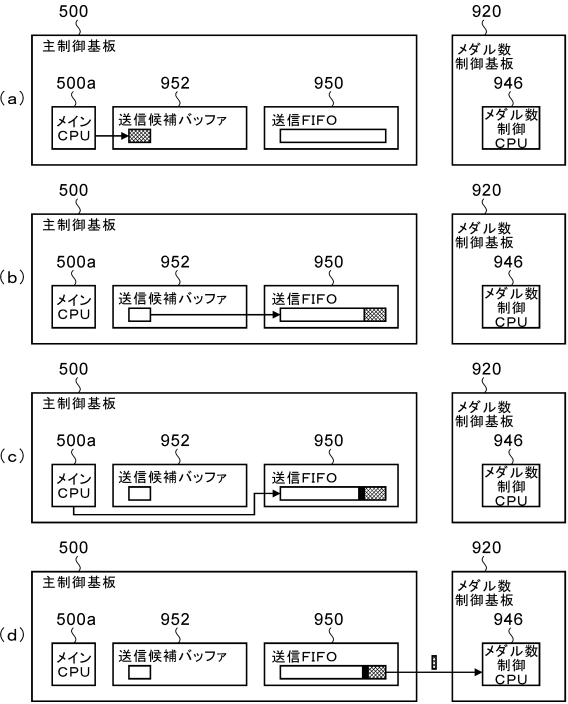
20

30

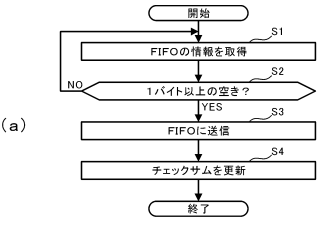
40

50

【図 2 5 9】



【図 2 6 0】



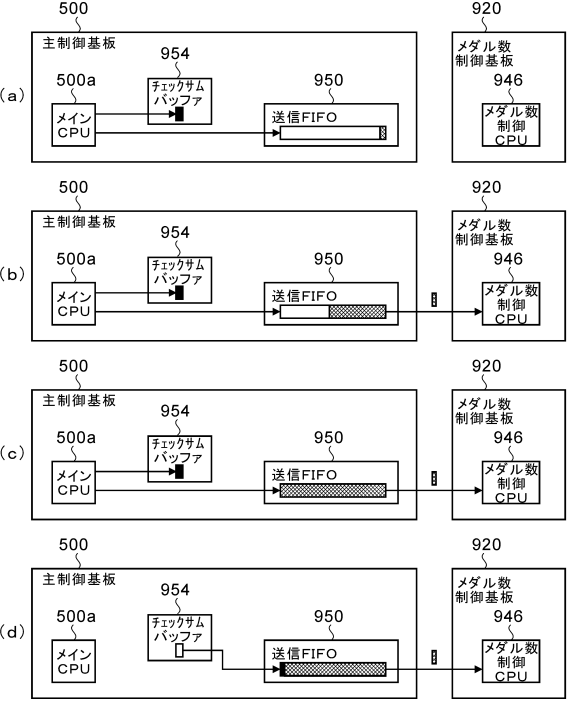
(b)

行数	コマンド
1	PUSH AF
2	MCD_SND01:
3	IN A, (@SOST_)
4	JBIT Z, 6, A, MCD_SND01
5	POP AF
6	OUT (@SOD1_), A
7	ADD A, E
8	LD E, A
9	RET

10

20

【図 2 6 1】



30

40

50

フロントページの続き

東京都台東区東上野一丁目16番1号 株式会社平和内
(72)発明者 西澤 暢
東京都台東区東上野一丁目16番1号 株式会社平和内
審査官 金子 和孝
(56)参考文献 特開2019-187772 (JP, A)
特開2021-058535 (JP, A)
(58)調査した分野 (Int.Cl., DB名)
A63F 5/04
A63F 7/02