

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7108595号
(P7108595)

(45)発行日 令和4年7月28日(2022.7.28)

(24)登録日 令和4年7月20日(2022.7.20)

(51)国際特許分類

A 6 3 F 5/04 (2006.01)

F I

A 6 3 F

5/04

6 1 1 B

A 6 3 F

5/04

6 1 2

請求項の数 1 (全510頁)

(21)出願番号 特願2019-231456(P2019-231456)
(22)出願日 令和1年12月23日(2019.12.23)
(65)公開番号 特開2021-97903(P2021-97903A)
(43)公開日 令和3年7月1日(2021.7.1)
審査請求日 令和3年10月18日(2021.10.18)

(73)特許権者 598098526
株式会社ユニバーサルエンターテインメント
東京都江東区有明三丁目7番26号 有明フロンティアビルA棟
(74)代理人 110000925
特許業務法人信友国際特許事務所
(72)発明者 ▲高▼橋 一也
東京都江東区有明3丁目7番26号
審査官 岡崎 彦哉

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】

演算制御部、第1記憶部及び第2記憶部を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、1バイトで構成されたレジスタを有し、遊技者にとって有利な特別遊技を制御可能であり、

前記第1記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定のシフト命令が含まれ、

前記特定のシフト命令では、オペランドとして、命令、前記レジスタを示す情報、及び、前記レジスタ内の所定のビット位置がこの順で配置され、

前記演算制御部により前記特定のシフト命令が実行されると、前記レジスタ内の前記所定のビット位置から前記命令が示す方向の最端ビット位置までの領域に格納された値を、前記命令が示す方向にシフトさせることが可能であり、

前記特別遊技の作動中に行われる所定の抽籤で使用される所定のテーブルデータであって、複数種の情報が格納された1バイトの複合データが記憶されている前記所定のテーブルデータから、前記複合データ内のビットデータを取得する場合に、前記特定のシフト命令を使用することが可能であり、

遊技状態には、少なくとも遊技者にとって有利となる報知を行う報知遊技状態と、報知を行うことが無い通常遊技状態とがあり、

前記特別遊技の作動中以外に行われる前記所定の抽籤で使用される前記所定のテーブルデータ以外のテーブルデータにおいても前記所定のテーブルデータと同じ構成で前記複合データが記憶され、
前記所定のテーブルデータは、前記報知遊技状態及び前記通常遊技状態で作動する前記特別遊技で使用される

ことを特徴とする遊技機。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、遊技機に関する。

【背景技術】

【0002】

従来、複数の図柄がそれぞれの表面に設けられた複数のリールと、スタートスイッチと、ストップスイッチと、各リールに対応して設けられたステッピングモータと、制御部とを備えた、パチスロと呼ばれる遊技機が知られている。スタートスイッチは、メダルやコインなどの遊技媒体が遊技機に投入された後、スタートレバーが遊技者により操作されたこと（以下、「開始操作」ともいう）を検出し、全てのリールの回転の開始を要求する信号を出力する。ストップスイッチは、各リールに対応して設けられたストップボタンが遊技者により押されたこと（以下、「停止操作」ともいう）を検出し、該当するリールの回転の停止を要求する信号を出力する。ステッピングモータは、その駆動力を対応するリールに伝達する。また、制御部は、スタートスイッチ及びストップスイッチにより出力された信号に基づいて、ステッピングモータの動作を制御し、各リールの回転動作及び停止動作を行う。

【0003】

このような遊技機では、開始操作が検出されると、プログラム上で乱数を用いた抽籤処理（以下、「内部抽籤処理」という）が行われ、その抽籤の結果（以下、「内部当籤役」という）と停止操作のタイミングとに基づいてリールの回転の停止を行う。そして、全てのリールの回転が停止され、入賞の成立に係る図柄の組合せ（表示役）が表示されると、その図柄の組合せに対応する特典が遊技者に付与される。なお、遊技者に付与される特典の例としては、遊技媒体（メダル等）の払い出し、遊技媒体を消費することなく再度、内部抽籤処理を行う再遊技（以下、「リプレイ」ともいう）の作動、遊技媒体の払い出し機会が増加するボーナスゲームの作動等を挙げることができる。

【0004】

また、従来、上記構成の遊技機において、特定の小役（遊技媒体の払い出しに係る役）の成立をランプ等でナビゲートする機能、すなわち、アシストタイム（以下、「ＡＴ」という）の機能を備える遊技機が開発されている。また、従来、特定の図柄組合せが表示された場合にリプレイの当籤確率が通常時より高い遊技状態が作動する機能、すなわち、リプレイタイム（以下、「ＲＴ」という）の機能を備える遊技機も開発されている。さらに、従来、ＡＴとＲＴとが同時に作動するアシストリプレイタイム（以下、「ＡＲＴ」という）の機能を備えたパチスロが開発されている。

【0005】

上述した遊技機は、通常、内部当籤役の決定、各リールの回転及び停止、入賞の有無の判定等の遊技機の主な遊技動作を制御する回路（主制御回路）が実装された主制御基板と、映像の表示等による演出動作を制御する回路（副制御回路）が実装された副制御基板とを備える。そして、遊技動作は、主制御回路に搭載されたＣＰＵ（Central Processing Unit）により制御される。この際、ＣＰＵの制御により、主制御回路のＲＯＭ（Read Only Memory）に記憶されたプログラム及び各種テーブルデータ等により、各種遊技動作に関する処理が実行される。

【0006】

ところで、従来、リセット入力からプログラムの起動までの時間をランダム化させる遊技

10

20

30

40

50

機が開示されている（例えば、特許文献1参照）。

【先行技術文献】

【特許文献】

【0007】

【文献】特開2009-034162号公報

【発明の概要】

【発明が解決しようとする課題】

【0008】

上述した特許文献1で開示されている遊技機では、電源投入時の不正行為を有効に阻止することができるが、マイクロプロセッサに関しては、他の機能についても更なる改良が求められている。

【0009】

本発明は上記課題を鑑みなされたものであり、本発明の目的は、各種機能に対して更なる改良が施されたマイクロプロセッサを搭載した遊技機を提供することである。

【課題を解決するための手段】

【0010】

上記課題を解決するために、本発明では、以下のような構成の遊技機を提供する。

【0011】

演算制御部（例えば、後述のメインCPU101）、第1記憶部（例えば、後述のメインROM102）及び第2記憶部（例えば、後述のメインRAM103）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、1バイトで構成されたレジスタ（例えば、後述の汎用レジスタ）を有し、遊技者に取って有利な特別遊技（例えば、後述の「通常BB」、「ART中BB」）を制御可能であり、

前記第1記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定のシフト命令（例えば、後述の「SLBIT A, n」、「SRBIT A, n」等）が含まれ、

前記特定のシフト命令では、オペランドとして、命令（例えば、後述の「SLBIT」、「SRBIT」）、前記レジスタを示す情報、及び、前記レジスタ内の所定のビット位置（例えば、後述の指定ビット位置n）がこの順で配置され、

前記演算制御部により前記特定のシフト命令が実行されると、前記レジスタ内の前記所定のビット位置から前記命令が示す方向の最端ビット位置までの領域に格納された値を、前記命令が示す方向にシフトさせることが可能であり、

前記特別遊技の作動中に行われる所定の抽籤（例えば、後述のBB中ART抽籤）で使用する所定のテーブルデータ（例えば、後述のBB中ART抽籤テーブル）であって、複数種の情報が格納された1バイトの複合データが記憶されている前記所定のテーブルデータから、前記複合データ内のビットデータを取得する場合に、前記特定のシフト命令を使用することが可能であり、

遊技状態には、少なくとも遊技者にとって有利となる報知を行う報知遊技状態（例えば、後述のART状態）と、報知を行うことが無い通常遊技状態（例えば、後述の通常状態）とがあり、

前記特別遊技の作動中以外に行われる前記所定の抽籤で使用する前記所定のテーブルデータ以外のテーブルデータにおいても前記所定のテーブルデータと同じ構成で前記複合データが記憶され、

前記所定のテーブルデータは、前記報知遊技状態及び前記通常遊技状態で作動する前記特別遊技で使用される

ことを特徴とする遊技機。

【発明の効果】

【0015】

10

20

30

40

50

上記構成の本発明の遊技機によれば、各種機能に対して更なる改良が施されたマイクロプロセッサを搭載した遊技機を提供することができる。

【図面の簡単な説明】

【0016】

【図1】本発明の第1実施形態における遊技機の機能フローを説明するための図である。

【図2】本発明の第1実施形態における遊技機の外観構造を示す斜視図である。

【図3】本発明の第1実施形態における遊技機の内部構造を示す図である。

【図4】本発明の第1実施形態における遊技機の内部構造を示す図である。

【図5】本発明の第1実施形態のサブ表示装置に表示される各種表示画面の概略構成を示す図である。

10

【図6】本発明の第1実施形態におけるサブ表示装置の表示画面の遷移例を示す図である。

【図7】本発明の第1実施形態の遊技機が備える回路の全体構成を示すブロック図である。

【図8】本発明の第1実施形態における主制御回路の内部構成を示すブロック図である。

【図9】本発明の第1実施形態におけるマイクロプロセッサの内部構成を示すブロック図である。

【図10】本発明の第1実施形態における副制御回路の内部構成を示すブロック図である。

【図11】本発明の第1実施形態におけるメインCPUが有する各種レジスタの構成図である。

【図12】本発明の第1実施形態における主制御回路のメモリマップを示す図である。

【図13】本発明の第1実施形態におけるパチスロのボーナス状態及び非ボーナス状態間における遊技状態の遷移フローを示す図である。

20

【図14】本発明の第1実施形態におけるパチスロのART遊技状態、非ART遊技状態及びボーナス状態間における遊技状態の遷移フローを示す図である。

【図15】本発明の第1実施形態における図柄配置テーブルの一例を示す図である。

【図16】本発明の第1実施形態における内部抽籤テーブルの一例を示す図である。

【図17】本発明の第1実施形態における内部抽籤テーブルの一例を示す図である。

【図18】本発明の第1実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図19】本発明の第1実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図20】本発明の第1実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図21】本発明の第1実施形態における図柄組合せ決定テーブルの一例を示す図である。

30

【図22】本発明の第1実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図23】本発明の第1実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図24】本発明の第1実施形態における内部当籤役と停止図柄組合せとの対応関係を示す図である。

【図25】本発明の第1実施形態における内部当籤役と停止図柄組合せとの対応関係を示す図である。

【図26】本発明の第1実施形態におけるリール停止初期設定テーブルの一例を示す図である。

【図27】本発明の第1実施形態における引込優先順位テーブルの一例を示す図である。

【図28】本発明の第1実施形態における当り要求フラグ格納領域、入賞作動フラグ格納領域の構成（その1）を示す図である。

40

【図29】本発明の第1実施形態における当り要求フラグ格納領域、入賞作動フラグ格納領域の構成（その2）を示す図である。

【図30】本発明の第1実施形態における当り要求フラグ格納領域、入賞作動フラグ格納領域の（その3）を示す図である。

【図31】本発明の第1実施形態における持越役格納領域の構成を示す図である。

【図32】本発明の第1実施形態における遊技状態フラグ格納領域の構成を示す図である。

【図33】本発明の第1実施形態における作動ストップボタン格納領域の構成を示す図である。

【図34】本発明の第1実施形態における押下順序格納領域の構成を示す図である。

50

【図 3 5】本発明の第 1 実施形態における図柄コード格納領域の構成を示す図である。

【図 3 6】本発明の第 1 実施形態における内部当籤役とサブフラグとの対応表（その 1）を示す図である。

【図 3 7】本発明の第 1 実施形態における内部当籤役とサブフラグとの対応表（その 2）を示す図である。

【図 3 8】本発明の第 1 実施形態の遊技機において、サブフラグ EX「3 連チリリプ」又は「リーチ目リプ」が当籤した際の報知動作を説明するための図である。

【図 3 9】本発明の第 1 実施形態における一般遊技状態中の遊技の流れを説明するための図である。

【図 4 0】本発明の第 1 実施形態における通常中高確率抽籤テーブルの一例を示す図である。

10

【図 4 1】本発明の第 1 実施形態における C Z 抽籤テーブルの一例を示す図である。

【図 4 2】本発明の第 1 実施形態における C Z 1 中モードアップ抽籤テーブルの一例を示す図である。

【図 4 3】本発明の第 1 実施形態における C Z 2 中ポイント抽籤テーブルの一例を示す図である。

【図 4 4】本発明の第 1 実施形態における C Z 中 A R T 抽籤テーブル（C Z 1，C Z 2 用）の一例を示す図である。

【図 4 5】本発明の第 1 実施形態における C Z 中 A R T 抽籤テーブル（C Z 3 用）の一例を示す図である。

20

【図 4 6】本発明の第 1 実施形態における通常 A R T 中の遊技の流れを説明するための図である。

【図 4 7】本発明の第 1 実施形態における A R T 中フラグ変換抽籤テーブルの一例を示す図である。

【図 4 8】本発明の第 1 実施形態における A R T レベル決定テーブルの一例を示す図である。

【図 4 9】本発明の第 1 実施形態における通常 A R T 中高確率抽籤テーブルの一例を示す図である。

【図 5 0】本発明の第 1 実施形態における A R T 中 C T 抽籤テーブルの一例を示す図である。

30

【図 5 1】本発明の第 1 実施形態における通常 A R T 中上乗せ抽籤テーブルの一例を示す図である。

【図 5 2】本発明の第 1 実施形態における C T 状態中の遊技の流れを説明するための図である。

【図 5 3】本発明の第 1 実施形態における C T 中テーブル抽籤テーブルの一例を示す図である。

【図 5 4】本発明の第 1 実施形態における C T 中フラグ変換抽籤テーブルの一例を示す図である。

【図 5 5】本発明の第 1 実施形態における C T 中上乗せ抽籤テーブルの一例を示す図である。

40

【図 5 6】本発明の第 1 実施形態における C T 中セット数上乗せ抽籤テーブルの一例を示す図である。

【図 5 7】本発明の第 1 実施形態におけるボーナス状態中の遊技の流れを説明するための図である。

【図 5 8】本発明の第 1 実施形態におけるボーナス種別抽籤テーブルの一例を示す図である。

【図 5 9】本発明の第 1 実施形態におけるボーナス中 A R T ゲーム数上乗せ抽籤テーブルの一例を示す図である。

【図 6 0】本発明の第 1 実施形態におけるボーナス終了時 C T 抽籤テーブルの一例を示す図である。

50

【図 6 1】本発明の第 1 実施形態における一般遊技状態中の遊技（その他）の流れを説明するための図である。

【図 6 2】本発明の第 1 実施形態における非 A R T 中フラグ変換抽籤テーブルの一例を示す図である。

【図 6 3】本発明の第 1 実施形態におけるメイン側ナビデータとサブ側ナビデータとの対応関係を示す図である。

【図 6 4】本発明の第 1 実施形態における遊技機の主制御回路により実行される電源投入（リセット割込み）時処理の例を示すフローチャートである。

【図 6 5】本発明の第 1 実施形態における電源投入時処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

10

【図 6 6】本発明の第 1 実施形態における遊技復帰処理の例を示すフローチャートである。

【図 6 7】本発明の第 1 実施形態における遊技復帰処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 6 8】本発明の第 1 実施形態における設定変更確認処理の例を示すフローチャートである。

【図 6 9】本発明の第 1 実施形態における設定変更確認処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 7 0】本発明の第 1 実施形態における設定変更コマンド生成格納処理の例を示すフローチャートである。

【図 7 1】本発明の第 1 実施形態における設定変更コマンド生成格納処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

20

【図 7 2】本発明の第 1 実施形態における通信データ格納処理の例を示すフローチャートである。

【図 7 3】本発明の第 1 実施形態における通信データ格納処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 7 4】本発明の第 1 実施形態における通信データポインタ更新処理の例を示すフローチャートである。

【図 7 5】本発明の第 1 実施形態における通信データポインタ更新処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 7 6】本発明の第 1 実施形態における電断時（外部）処理の例を示すフローチャートである。

30

【図 7 7】本発明の第 1 実施形態におけるチェックサム生成処理（規定外）の例を示すフローチャートである。

【図 7 8】本発明の第 1 実施形態におけるチェックサム生成処理のフローチャート中の各種処理を実行するためのソースプログラムの一例、並びに、チェックサム生成処理で実行されるスタックポインタの更新動作及びレジスタへのデータの読み出し動作の様子を示す図である。

【図 7 9】本発明の第 1 実施形態におけるサムチェック処理（規定外）の例を示すフローチャートである。

【図 8 0】本発明の第 1 実施形態におけるサムチェック処理（規定外）の例を示すフローチャートである。

40

【図 8 1】本発明の第 1 実施形態におけるサムチェック処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 8 2】本発明の第 1 実施形態における遊技機の主制御回路により実行されるメイン処理（主要動作処理）の例を示すフローチャートである。

【図 8 3】本発明の第 1 実施形態におけるメダル受付・スタートチェック処理の例を示すフローチャートである。

【図 8 4】本発明の第 1 実施形態におけるメダル受付・スタートチェック処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 8 5】本発明の第 1 実施形態におけるメダル投入処理の例を示すフローチャートであ

50

る。

【図 8 6】本発明の第 1 実施形態におけるメダル投入処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 8 7】本発明の第 1 実施形態におけるメダル投入チェック処理の例を示すフローチャートである。

【図 8 8】本発明の第 1 実施形態におけるメダル投入チェック処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 8 9】本発明の第 1 実施形態におけるエラー処理の例を示すフローチャートである。

【図 9 0】本発明の第 1 実施形態におけるエラー処理のソースプログラム上で、実際に参照されるエラーテーブルの構成の一例を示す図である。

10

【図 9 1】本発明の第 1 実施形態における乱数取得処理の例を示すフローチャートである。

【図 9 2】本発明の第 1 実施形態における内部抽籤処理の例を示すフローチャートである。

【図 9 3】本発明の第 1 実施形態における内部抽籤処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 9 4】本発明の第 1 実施形態における内部抽籤処理のソースプログラム上で、実際に参照される内部抽籤テーブル（一般遊技用）の構成の一例を示す図である。

【図 9 5】本発明の第 1 実施形態における内部抽籤処理のソースプログラム上で、実際に参照される R T 状態別抽籤値選択テーブルの構成の一例を示す図である。

【図 9 6】本発明の第 1 実施形態における内部抽籤処理のソースプログラム上で、実際に参照される、内部抽籤値テーブル選択テーブル、1 バイト内部抽籤値テーブル、2 バイト内部抽籤値テーブル、1 バイト設定別内部抽籤値テーブル及び 2 バイト設定別内部抽籤値テーブルの構成の一例を示す図である。

20

【図 9 7】本発明の第 1 実施形態における図柄設定処理の例を示すフローチャートである。

【図 9 8】本発明の第 1 実施形態における特賞（ボーナス）当籤番号及び小役当籤番号と、内部当籤役との対応を示す図である。

【図 9 9】本発明の第 1 実施形態における図柄設定処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 0 0】本発明の第 1 実施形態における図柄設定処理のソースプログラム上で、実際に参照される当り要求フラグテーブルの構成の一例を示す図である。

【図 1 0 1】本発明の第 1 実施形態における圧縮データ格納処理の例を示すフローチャートである。

30

【図 1 0 2】本発明の第 1 実施形態における第 2 インターフェースボード制御処理（規定外）の例を示すフローチャートである。

【図 1 0 3】本発明の第 1 実施形態における第 2 インターフェースボード出力処理の例を示すフローチャートである。

【図 1 0 4】本発明の第 1 実施形態における状態別制御処理の例を示すフローチャートである。

【図 1 0 5】本発明の第 1 実施形態におけるサブフラグ変換処理の例を示すフローチャートである。

【図 1 0 6】本発明の第 1 実施形態におけるサブフラグ変換処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

40

【図 1 0 7】本発明の第 1 実施形態におけるサブフラグ変換処理のソースプログラム上で、実際に参照されるサブフラグ変換テーブルの構成の一例を示す図である。

【図 1 0 8】本発明の第 1 実施形態におけるナビセット処理の例を示すフローチャートである。

【図 1 0 9】本発明の第 1 実施形態におけるナビセット処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 1 0】本発明の第 1 実施形態におけるナビセット処理のソースプログラム上で、実際に参照されるナビデータテーブルの構成の一例を示す図である。

【図 1 1 1】本発明の第 1 実施形態におけるフラグ変換処理の例を示すフローチャートで

50

ある。

【図 1 1 2】本発明の第 1 実施形態における通常中スタート時処理の例を示すフローチャートである。

【図 1 1 3】本発明の第 1 実施形態における C Z 中スタート時処理の例を示すフローチャートである。

【図 1 1 4】本発明の第 1 実施形態における C Z 1 (C Z 2) 中処理の例を示すフローチャートである。

【図 1 1 5】本発明の第 1 実施形態における C Z 1 (C Z 2) 中処理の例を示すフローチャートである。

【図 1 1 6】本発明の第 1 実施形態における C Z 3 中処理の例を示すフローチャートである。

10

【図 1 1 7】本発明の第 1 実施形態における通常 A R T 中スタート時処理の例を示すフローチャートである。

【図 1 1 8】本発明の第 1 実施形態における C T 中スタート時処理の例を示すフローチャートである。

【図 1 1 9】本発明の第 1 実施形態における C T 中 C T 抽籤処理の例を示すフローチャートである。

【図 1 2 0】本発明の第 1 実施形態におけるテーブルデータ取得処理の例を示すフローチャートである。

【図 1 2 1】本発明の第 1 実施形態におけるテーブルデータ取得処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

20

【図 1 2 2】本発明の第 1 実施形態におけるテーブルデータ取得処理のソースプログラム上で、実際に参照される C T 中 C T 抽籤テーブルの構成の一例を示す図である。

【図 1 2 3】本発明の第 1 実施形態における 1 バイト抽籤処理の例を示すフローチャートである。

【図 1 2 4】本発明の第 1 実施形態における 1 バイト抽籤処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 2 5】本発明の第 1 実施形態における B B 中スタート時処理の例を示すフローチャートである。

【図 1 2 6】本発明の第 1 実施形態における引込優先順位格納処理の例を示すフローチャートである。

30

【図 1 2 7】本発明の第 1 実施形態における引込優先順位格納処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 2 8】本発明の第 1 実施形態における図柄コード取得処理の例を示すフローチャートである。

【図 1 2 9】本発明の第 1 実施形態における図柄コード取得処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 3 0】本発明の第 1 実施形態における図柄コード取得処理のソースプログラム上で、実際に参照される第 1 回胴 (左リール) 図柄配置テーブル、及び、第 1 回胴図柄配置テーブルセット時に参照される図柄対応入賞作動テーブルの構成の一例を示す図である。

40

【図 1 3 1】本発明の第 1 実施形態における図柄コード取得処理のソースプログラム上で、実際に参照される第 2 回胴 (中リール) 図柄配置テーブル、及び、第 2 回胴図柄配置テーブルセット時に参照される図柄対応入賞作動テーブルの構成の一例を示す図である。

【図 1 3 2】本発明の第 1 実施形態における図柄コード取得処理のソースプログラム上で、実際に参照される第 3 回胴 (右リール) 図柄配置テーブル、及び、第 3 回胴図柄配置テーブルセット時に参照される図柄対応入賞作動テーブルの構成の一例を示す図である。

【図 1 3 3】本発明の第 1 実施形態における論理積演算処理の例を示すフローチャートである。

【図 1 3 4】本発明の第 1 実施形態における引込優先順位取得処理の例を示すフローチャートである。

50

【図 1 3 5】本発明の第 1 実施形態における引込優先順位取得処理の例を示すフローチャートである。

【図 1 3 6】本発明の第 1 実施形態における引込優先順位取得処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 3 7】本発明の第 1 実施形態における引込優先順位取得処理のソースプログラム上で、実際に参照される引込優先順位テーブルの構成の一例を示す図である。

【図 1 3 8】本発明の第 1 実施形態におけるリール停止制御処理の例を示すフローチャートである。

【図 1 3 9】本発明の第 1 実施形態におけるリール停止制御処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

10

【図 1 4 0】本発明の第 1 実施形態におけるリール停止制御処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 4 1】本発明の第 1 実施形態におけるリール停止可能信号 OFF 処理（規定外）の例を示すフローチャートである。

【図 1 4 2】本発明の第 1 実施形態におけるリール停止可能信号 ON 処理（規定外）の例を示すフローチャートである。

【図 1 4 3】本発明の第 1 実施形態における規定外ポート出力処理の例を示すフローチャートである。

【図 1 4 4】本発明の第 1 実施形態における規定外ポート出力処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

20

【図 1 4 5】本発明の第 1 実施形態における入賞検索処理の例を示すフローチャートである。

【図 1 4 6】本発明の第 1 実施形態における入賞検索処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 4 7】本発明の第 1 実施形態における入賞検索処理のソースプログラム上で、実際に参照される払出枚数データテーブルの構成の一例を示す図である。

【図 1 4 8】本発明の第 1 実施形態におけるイリーガルヒットチェック処理の例を示すフローチャートである。

【図 1 4 9】本発明の第 1 実施形態におけるイリーガルヒットチェック処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

30

【図 1 5 0】本発明の第 1 実施形態における入賞チェック・メダル払出処理の例を示すフローチャートである。

【図 1 5 1】本発明の第 1 実施形態における入賞チェック・メダル払出処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 5 2】本発明の第 1 実施形態におけるメダル払出枚数チェック処理の例を示すフローチャートである。

【図 1 5 3】本発明の第 1 実施形態におけるメダル払出枚数チェック処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図 1 5 4】本発明の第 1 実施形態における BB チェック処理の例を示すフローチャートである。

40

【図 1 5 5】本発明の第 1 実施形態における RT チェック処理の例を示すフローチャートである。

【図 1 5 6】本発明の第 1 実施形態における RT チェック処理の例を示すフローチャートである。

【図 1 5 7】本発明の第 1 実施形態における CZ・ART 終了時処理の例を示すフローチャートである。

【図 1 5 8】本発明の第 1 実施形態における遊技機の主制御回路により実行される割込処理の例を示すフローチャートである。

【図 1 5 9】本発明の第 1 実施形態における 7 セグ LED 駆動処理の例を示すフローチャートである。

50

【図１６０】本発明の第１実施形態における７セグＬＥＤ駆動処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図１６１】本発明の第１実施形態における７セグ表示データ生成処理の例を示すフローチャートである。

【図１６２】本発明の第１実施形態における７セグ表示データ生成処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図１６３】本発明の第１実施形態における７セグ表示データ生成処理のソースプログラム上で、実際に参照される７セグカソードテーブルの構成の一例を示す図である。

【図１６４】本発明の第１実施形態におけるタイマー更新処理の例を示すフローチャートである。

10

【図１６５】本発明の第１実施形態におけるタイマー更新処理のフローチャート中の各種処理を実行するためのソースプログラムの一例を示す図である。

【図１６６】本発明の第１実施形態における試射試験信号制御処理（規定外）の例を示すフローチャートである。

【図１６７】本発明の第１実施形態における回胴制動信号生成処理の例を示すフローチャートである。

【図１６８】本発明の第１実施形態における特賞信号制御処理の例を示すフローチャートである。

【図１６９】本発明の第１実施形態における条件装置信号制御処理の例を示すフローチャートである。

20

【図１７０】本発明の第１実施形態における条件装置信号制御処理の例を示すフローチャートである。

【図１７１】本発明の第１実施形態における遊技機の副制御回路により実行されるサブ側ナビ制御処理の例を示すフローチャートである。

【図１７２】本発明の第１実施形態における遊技者登録処理の例を示すフローチャートである。

【図１７３】本発明の第１実施形態における履歴管理処理の例を示すフローチャートである。

【図１７４】本発明の変形例１におけるＣＴ前兆中の遊技の流れを示す図である。

【図１７５】本発明の変形例２における内部当籤役と停止図柄組合せとの対応関係を示す図である。

30

【図１７６】本発明の変形例３におけるメイン側ナビデータとサブ側ナビデータとの対応関係を示す図である。

【図１７７】本発明の変形例５における押し順とロック状態との対応関係を示す図である。

【図１７８】本発明の第２実施形態におけるパチスロのボーナス状態及び非ボーナス状態間における遊技状態の遷移フローを示す図である。

【図１７９】本発明の第２実施形態における図柄配置テーブルの一例を示す図である。

【図１８０】本発明の第２実施形態における内部抽籤テーブルの一例を示す図である。

【図１８１】本発明の第２実施形態における内部抽籤テーブルの一例を示す図である。

【図１８２】本発明の第２実施形態における内部抽籤テーブルの一例を示す図である。

40

【図１８３】本発明の第２実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図１８４】本発明の第２実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図１８５】本発明の第２実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図１８６】本発明の第２実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図１８７】本発明の第２実施形態における図柄組合せ決定テーブルの一例を示す図である。

50

【図 1 8 8】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図 1 8 9】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図 1 9 0】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図 1 9 1】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図 1 9 2】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

10

【図 1 9 3】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図 1 9 4】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図 1 9 5】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図 1 9 6】本発明の第 2 実施形態における図柄組合せ決定テーブルの一例を示す図である。

【図 1 9 7】本発明の第 2 実施形態における図柄組合せの内容の一例を示す図である。

【図 1 9 8】本発明の第 2 実施形態における図柄組合せの内容の一例を示す図である。

20

【図 1 9 9】本発明の第 2 実施形態における図柄組合せの内容の一例を示す図である。

【図 2 0 0】本発明の第 2 実施形態における図柄組合せの内容の一例を示す図である。

【図 2 0 1】本発明の第 2 実施形態における図柄組合せの内容の一例を示す図である。

【図 2 0 2】本発明の第 2 実施形態における図柄組合せの内容の一例を示す図である。

【図 2 0 3】本発明の第 2 実施形態における図柄組合せの内容の一例を示す図である。

【図 2 0 4】本発明の第 2 実施形態における図柄組合せの内容の一例を示す図である。

【図 2 0 5】本発明の第 2 実施形態における内部当籤役と停止図柄組合せとの対応関係を示す図である。

【図 2 0 6】本発明の第 2 実施形態におけるパチスロの報知を考慮した遊技状態の遷移フローを示す図である。

30

【図 2 0 7】本発明の第 2 実施形態におけるパチスロの報知を考慮した遊技状態の遷移条件を示す図である。

【図 2 0 8】本発明の第 2 実施形態における内部当籤役と抽籤用フラグとの対応関係を示す図である。

【図 2 0 9】本発明の第 2 実施形態における通常状態中の遊技の流れを示す図である。

【図 2 1 0】本発明の第 2 実施形態における C Z 中の遊技の流れを示す図である。

【図 2 1 1】本発明の第 2 実施形態における C Z 中の遊技の流れを示す図である。

【図 2 1 2】本発明の第 2 実施形態における A R T 状態中の遊技の流れを示す図である。

【図 2 1 3】本発明の第 2 実施形態における A R T 状態中の遊技の流れを示す図である。

【図 2 1 4】本発明の第 2 実施形態におけるボーナス状態中の遊技の流れを示す図である。

40

【図 2 1 5】本発明の第 2 実施形態におけるモード移行抽籤テーブルの一例を示す図である。

【図 2 1 6】本発明の第 2 実施形態において、ソースプログラム上で実際に参照されるモード移行抽籤テーブルの構成の一例を示す図である。

【図 2 1 7】本発明の第 2 実施形態における状態移行抽籤テーブルの一例を示す図である。

【図 2 1 8】本発明の第 2 実施形態における状態移行抽籤テーブルの一例を示す図である。

【図 2 1 9】本発明の第 2 実施形態における高確保証ゲーム数抽籤テーブルの一例を示す図である。

【図 2 2 0】本発明の第 2 実施形態におけるモード別 C Z 抽籤テーブルの一例を示す図で

50

ある。

【図 2 2 1】本発明の第 2 実施形態において、ソースプログラム上で実際に参照されるモード別 C Z 抽籤テーブルの構成の一例を示す図である。

【図 2 2 2】本発明の第 2 実施形態における B B 当籤時状態別 C Z 抽籤テーブルの一例を示す図である。

【図 2 2 3】本発明の第 2 実施形態における状態別 C Z 抽籤テーブルの一例を示す図である。

【図 2 2 4】本発明の第 2 実施形態における C Z 前兆ゲーム数抽籤テーブルの一例を示す図である。

【図 2 2 5】本発明の第 2 実施形態における B B 当籤時 A R T 抽籤テーブルの一例を示す図である。

10

【図 2 2 6】本発明の第 2 実施形態における通常時 A R T 抽籤テーブルの一例を示す図である。

【図 2 2 7】本発明の第 2 実施形態における A R T 前兆ゲーム数抽籤テーブルの一例を示す図である。

【図 2 2 8】本発明の第 2 実施形態における C Z 中 A R T 抽籤テーブルの一例を示す図である。

【図 2 2 9】本発明の第 2 実施形態における C Z 中 A R T 抽籤テーブルの一例を示す図である。

【図 2 3 0】本発明の第 2 実施形態における A R T 中ストック抽籤テーブルの一例を示す図である。

20

【図 2 3 1】本発明の第 2 実施形態における A R T 中ストック抽籤テーブルの一例を示す図である。

【図 2 3 2】本発明の第 2 実施形態における A R T 中ストック抽籤テーブルの一例を示す図である。

【図 2 3 3】本発明の第 2 実施形態における A R T 中 B B 当籤時ストック抽籤テーブルの一例を示す図である。

【図 2 3 4】本発明の第 2 実施形態における A R T 当籤時ストック数抽籤テーブルの一例を示す図である。

【図 2 3 5】本発明の第 2 実施形態における特殊 C Z 移行抽籤テーブルの一例を示す図である。

30

【図 2 3 6】本発明の第 2 実施形態における A R T 当籤時ランク抽籤テーブルの一例を示す図である。

【図 2 3 7】本発明の第 2 実施形態における A R T 当籤時ランク抽籤テーブルの一例を示す図である。

【図 2 3 8】本発明の第 2 実施形態における A R T 当籤時ランク抽籤テーブルの一例を示す図である。

【図 2 3 9】本発明の第 2 実施形態における A R T 当籤時ランク抽籤テーブルの一例を示す図である。

【図 2 4 0】本発明の第 2 実施形態における A R T 当籤時ランク抽籤テーブルの一例を示す図である。

40

【図 2 4 1】本発明の第 2 実施形態におけるストック放出順序抽籤テーブルの一例を示す図である。

【図 2 4 2】本発明の第 2 実施形態におけるランク決め A R T 中ランク昇格抽籤テーブルの一例を示す図である。

【図 2 4 3】本発明の第 2 実施形態におけるナビ高確ゲーム数獲得抽籤テーブルの一例を示す図である。

【図 2 4 4】本発明の第 2 実施形態における C Z ゲーム数獲得抽籤テーブルの一例を示す図である。

【図 2 4 5】本発明の第 2 実施形態における A R T ゲーム数獲得抽籤テーブルの一例を示す図である。

50

す図である。

【図 2 4 6】本発明の第 2 実施形態における転落モード抽籤テーブルの一例を示す図である。

【図 2 4 7】本発明の第 2 実施形態における転落モード移行抽籤テーブルの一例を示す図である。

【図 2 4 8】本発明の第 2 実施形態における B B 当籤時転落モード移行抽籤テーブルの一例を示す図である。

【図 2 4 9】本発明の第 2 実施形態における B B 中 A R T 抽籤テーブル、B B 中 C Z 抽籤テーブル、B B 中 E P ストック抽籤テーブルの一例を示す図である。

【図 2 5 0】本発明の第 2 実施形態におけるパチスロの通常状態中の抽籤内容を示す図である。

10

【図 2 5 1】本発明の第 2 実施形態におけるパチスロの C Z 前兆中の抽籤内容を示す図である。

【図 2 5 2】本発明の第 2 実施形態におけるパチスロの A R T 前兆中の抽籤内容を示す図である。

【図 2 5 3】本発明の第 2 実施形態におけるパチスロの初当り C Z、及び、初当り C Z 終了の次遊技の抽籤内容を示す図である。

【図 2 5 4】本発明の第 2 実施形態におけるパチスロの A R T 準備中の抽籤内容を示す図である。

【図 2 5 5】本発明の第 2 実施形態におけるパチスロのランク決め A R T 中の抽籤内容を示す図である。

20

【図 2 5 6】本発明の第 2 実施形態におけるパチスロの通常 A R T 中の抽籤内容を示す図である。

【図 2 5 7】本発明の第 2 実施形態におけるパチスロの E P 準備中の抽籤内容を示す図である。

【図 2 5 8】本発明の第 2 実施形態におけるパチスロの E P 中の抽籤内容を示す図である。

【図 2 5 9】本発明の第 2 実施形態におけるパチスロの C Z (A R T 後)、C Z (A R T 後) 終了の次遊技、及び特殊 C Z 中の抽籤内容を示す図である。

【図 2 6 0】本発明の第 2 実施形態におけるパチスロの通常フラグ間中の抽籤内容を示す図である。

30

【図 2 6 1】本発明の第 2 実施形態におけるパチスロの通常 B B 中の抽籤内容を示す図である。

【図 2 6 2】本発明の第 2 実施形態におけるパチスロの A R T 中フラグ間中の抽籤内容を示す図である。

【図 2 6 3】本発明の第 2 実施形態におけるパチスロの A R T 中 B B 中の抽籤内容を示す図である。

【図 2 6 4】本発明の第 2 実施形態におけるパチスロの C Z の残りゲーム数と C Z 中の各抽籤との関係性を示す図である。

【図 2 6 5】本発明の第 2 実施形態におけるパチスロの C Z 中のストックの有無と各抽籤との関係性を示す図である。

40

【図 2 6 6】本発明の第 2 実施形態におけるパチスロの R T 4 状態又は R T 5 状態中の A R T 抽籤の概要を示す図である。

【図 2 6 7】本発明の第 2 実施形態におけるパチスロの押し順ベル当籤時の押し順と図柄組合せとの対応関係を示す図である。

【図 2 6 8】本発明の第 2 実施形態におけるパチスロの押し順ベル当籤時の押し順と R T 制御との対応関係を示す図である。

【図 2 6 9】本発明の第 2 実施形態におけるパチスロの押し順ベル当籤時の押し順と R T 制御との対応関係の別例を示す図である。

【図 2 7 0】本発明の第 2 実施形態におけるパチスロのナビ高確ゲーム数の管理方法を示す図である。

50

【図 2 7 1】本発明の第 2 実施形態におけるパチスロのナビ高確の有無と各抽籤との関係性を示す図である。

【図 2 7 2】本発明の第 2 実施形態におけるパチスロのメイン R A M (Random Access Memory) のメモリマップを示す図である。

【図 2 7 3】本発明の第 2 実施形態において、ソースプログラム上で実際に参照される乱数値格納領域の構成を示す図である。

【図 2 7 4】本発明の第 2 実施形態において、ソースプログラム上で実際に参照される状態移行抽籤テーブル (B B 終了時用) の構成を示す図である。

【図 2 7 5】本発明の第 2 実施形態における状態移行抽籤テーブル (B B 終了時用) 内で規定される、乱数格納領域のオフセット値と抽籤回数との複合データのビット構成を示す図である。

10

【図 2 7 6】本発明の第 2 実施形態において、ソースプログラム上で実際に参照される乱数指定用オフセット番号とそのラベルとの対応テーブルを示す図である。

【図 2 7 7】本発明の第 2 実施形態における状態移行抽籤テーブル (B B 終了時用) 内で規定される抽籤指定データのビット構成を示す図である。

【図 2 7 8】本発明の第 2 実施形態において、ソースプログラム上で実際に参照される B B 中 A R T 抽籤テーブル (通常 B B、又は、A R T 中 B B 中用) の構成を示す図である。

【図 2 7 9】本発明の第 2 実施形態において、ソースプログラム上で実際に参照される当否判定用係数テーブルの構成を示す図である。

【図 2 8 0】本発明の第 2 実施形態において、ソースプログラム上で実際に参照される B B 当籤時状態別 C Z 抽籤テーブル (A) ~ (C) の構成を示す図である。

20

【図 2 8 1】本発明の第 2 実施形態において、ソースプログラム上で実際に参照される C Z 中 A R T 抽籤テーブル (A) ~ (H) の構成を示す図である。

【図 2 8 2】本発明の第 2 実施形態における設定変更確認処理の例を示すフローチャートである。

【図 2 8 3】本発明の第 2 実施形態における設定変更確認処理のフローチャート中の一部の処理を実行するためのソースプログラムの一例を示す図である。

【図 2 8 4】本発明の第 2 実施形態における設定値 7 セグ表示処理の例を示すフローチャートである。

【図 2 8 5】本発明の第 2 実施形態における設定値 7 セグ表示処理のフローチャート中の一部の処理を実行するためのソースプログラムの一例を示す図である。

30

【図 2 8 6】本発明の第 2 実施形態におけるパチスロの主制御回路により実行されるメイン処理 (主要動作処理) の例を示すフローチャートである。

【図 2 8 7】本発明の第 2 実施形態における図柄設定処理の例を示すフローチャートである。

【図 2 8 8】本発明の第 2 実施形態における図柄設定処理の例を示すフローチャートである。

【図 2 8 9】本発明の第 2 実施形態における図柄設定処理のソースプログラム上で、実際に参照されるボーナス作動中小役当籤番号変換テーブルの構成の一例を示す図である。

【図 2 9 0】本発明の第 2 実施形態における変換後の小役当籤番号を示す図である。

40

【図 2 9 1】本発明の第 2 実施形態における図柄設定処理のフローチャート中の一部の処理を実行するためのソースプログラムの一例を示す図である。

【図 2 9 2】本発明の第 2 実施形態における遊技ロック設定処理の例を示すフローチャートである。

【図 2 9 3】本発明の第 2 実施形態における遊技ロック設定処理のフローチャート中の一部の処理を実行するためのソースプログラムの一例を示す図である。

【図 2 9 4】本発明の第 2 実施形態における試験試験信号制御処理 (規定外) のソースプログラム上で、実際に参照される規定外出力ポート格納領域の構成の一例を示す図である。

【図 2 9 5】本発明の第 2 実施形態における規定外出力ポート格納領域に格納される各試験信号のビット構成と、各ビットに割り付けられる情報内容との関係を示す図である。

50

【図 2 9 6】本発明の第 2 実施形態における試射試験信号制御処理（規定外）の例を示すフローチャートである。

【図 2 9 7】本発明の第 2 実施形態における回胴制動信号生成処理の例を示すフローチャートである。

【図 2 9 8】本発明の第 2 実施形態における特賞信号制御処理の例を示すフローチャートである。

【図 2 9 9】本発明の第 2 実施形態における特賞信号制御処理の例を示すフローチャートである。

【図 3 0 0】本発明の第 2 実施形態における条件装置信号制御処理の例を示すフローチャートである。

【図 3 0 1】本発明の第 2 実施形態における条件装置信号制御処理の例を示すフローチャートである。

【図 3 0 2】本発明の第 2 実施形態における試験信号出力処理の例を示すフローチャートである。

【図 3 0 3】本発明の第 2 実施形態における試験信号出力処理を実行するためのソースプログラムの一例を示す図である。

【図 3 0 4】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なエクステンジ命令コードの一覧表を示す図である。

【図 3 0 5】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なエクステンジ命令コード「EX ss, qq」の一覧表を示す図である。

【図 3 0 6】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なエクステンジ命令コード「EX r, (ss)」の一覧表を示す図である。

【図 3 0 7】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なエクステンジ命令コード「EX r, (qq)」の一覧表を示す図である。

【図 3 0 8】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なエクステンジ命令コード「EX Q r, (k)」の一覧表を示す図である。

【図 3 0 9】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コードの一覧表（その 1）を示す図である。

【図 3 1 0】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コードの一覧表（その 2）を示す図である。

【図 3 1 1】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コードの一覧表（その 3）を示す図である。

【図 3 1 2】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コードの一覧表（その 4）を示す図である。

【図 3 1 3】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD r, s」の一覧表（その 1）を示す図である。

【図 3 1 4】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD r, s」の一覧表（その 2）を示す図である。

【図 3 1 5】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD r, s」の一覧表（その 3）を示す図である。

【図 3 1 6】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD r, n」の一覧表を示す図である。

【図 3 1 7】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD r, (qq)」の一覧表を示す図である。

【図 3 1 8】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD r, (DE)」の一覧表を示す図である。

【図 3 1 9】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD r, (HL)」の一覧表を示す図である。

【図 3 2 0】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD Q r, (k)」の一覧表を示す図である。

10

20

30

40

50

【図 3 2 1】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD (HL), r」の一覧表を示す図である。

【図 3 2 2】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LD (qq), r」の一覧表を示す図である。

【図 3 2 3】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビットロード命令コード「LDQ (k), r」の一覧表を示す図である。

【図 3 2 4】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 16 ビットロード命令コードの一覧表（その 1）を示す図である。

【図 3 2 5】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 16 ビットロード命令コードの一覧表（その 2）を示す図である。

10

【図 3 2 6】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なジャンプ命令コードの一覧表（その 1）を示す図である。

【図 3 2 7】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なジャンプ命令コードの一覧表（その 2）を示す図である。

【図 3 2 8】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なジャンプ命令コード「JR cc, e」の一覧表を示す図である。

【図 3 2 9】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（加算）コードの一覧表（その 1）を示す図である。

【図 3 3 0】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（加算）コードの一覧表（その 2）を示す図である。

20

【図 3 3 1】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（加算）コードの一覧表（その 3）を示す図である。

【図 3 3 2】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（加算）コード「ICPLD A, r」の一覧表を示す図である。

【図 3 3 3】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（加算）コード「ICPLD (HL), r」の一覧表を示す図である。

【図 3 3 4】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（減算）コードの一覧表（その 1）を示す図である。

【図 3 3 5】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（減算）コードの一覧表（その 2）を示す図である。

30

【図 3 3 6】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（減算）コードの一覧表（その 3）を示す図である。

【図 3 3 7】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（減算）コード「DCPLD A, r」の一覧表を示す図である。

【図 3 3 8】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコンペアロード命令（減算）コード「DCPLD (HL), r」の一覧表を示す図である。

【図 3 3 9】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なシフト命令（左）コードの一覧表を示す図である。

40

【図 3 4 0】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なシフト命令（左）コード「SLA r」の一覧表を示す図である。

【図 3 4 1】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なシフト命令（右）コードの一覧表を示す図である。

【図 3 4 2】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なシフト命令（右）コード「SRA r」の一覧表を示す図である。

【図 3 4 3】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコール・リターン命令コードの一覧表（その 1）を示す図である。

【図 3 4 4】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能なコール・リターン命令コードの一覧表（その 2）を示す図である。

50

【図 3 4 5】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な CPU コントロール 命令コードの一覧表を示す図である。

【図 3 4 6】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 論理演算命令コードの一覧表を示す図である。

【図 3 4 7】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 論理演算命令コード「AND r」の一覧表を示す図である。

【図 3 4 8】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 論理演算命令コード「OR r」の一覧表を示す図である。

【図 3 4 9】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビット算術命令（比較）コードの一覧表を示す図である。

10

【図 3 5 0】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビット算術命令（比較）コード「CP r」の一覧表を示す図である。

【図 3 5 1】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビット算術命令（比較）コード「CP r, (HL)」の一覧表を示す図である。

【図 3 5 2】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビット算術命令（比較）コード「CPQ r, (k)」の一覧表を示す図である。

【図 3 5 3】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 16 ビット算術命令（比較）コードの一覧表を示す図である。

【図 3 5 4】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビット算術命令（加算・減算）コードの一覧表を示す図である。

20

【図 3 5 5】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 8 ビット算術命令（加算・減算）コード「INC r」及び「DEC r」の一覧表を示す 図である。

【図 3 5 6】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 16 ビット算術命令（加算・減算）コードの一覧表を示す図である。

【図 3 5 7】本発明の第 3 実施形態における遊技機の各種ソースプログラムで使用可能な 16 ビット算術命令（加算）コード「ADDWB HL, r」の一覧表を示す図である。

【図 3 5 8】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 1 を示す図である。

【図 3 5 9】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 2 を示す図である。

30

【図 3 6 0】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 3 を示す図である。

【図 3 6 1】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 4 を示す図である。

【図 3 6 2】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 5 を示す図である。

【図 3 6 3】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 6 を示す図である。

【図 3 6 4】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 7 を示す図である。

40

【図 3 6 5】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 8 を示す図である。

【図 3 6 6】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 9 を示す図である。

【図 3 6 7】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 10 を示す図である。

【図 3 6 8】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム 例 11 を示す図である。

【図 3 6 9】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム

50

例 1 2 を示す図である。

【図 3 7 0】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 1 3 を示す図である。

【図 3 7 1】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 1 4 を示す図である。

【図 3 7 2】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 1 5 を示す図である。

【図 3 7 3】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 1 6 を示す図である。

【図 3 7 4】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 1 7 を示す図である。

10

【図 3 7 5】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 1 8 を示す図である。

【図 3 7 6】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 1 9 を示す図である。

【図 3 7 7】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 0 を示す図である。

【図 3 7 8】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 1 を示す図である。

【図 3 7 9】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 2 を示す図である。

20

【図 3 8 0】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 3 を示す図である。

【図 3 8 1】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 4 を示す図である。

【図 3 8 2】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 5 を示す図である。

【図 3 8 3】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 6 を示す図である。

【図 3 8 4】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 7 を示す図である。

30

【図 3 8 5】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 8 を示す図である。

【図 3 8 6】本発明の第 3 実施形態で新たに設けられた命令コードを使用したプログラム例 2 9 を示す図である。

【発明を実施するための形態】

【0 0 1 7】

以下、本発明の各種実施形態に係る遊技機としてパチスロを例に挙げ、図面を参照しながら、その構成及び動作について説明する。なお、本実施形態では、ボーナス作動機能及び ART 機能を備えたパチスロについて説明する。

40

【0 0 1 8】

1. 第 1 実施形態

<機能フロー>

まず、図 1 を参照して、パチスロの機能フローについて説明する。本実施形態のパチスロでは、遊技を行うための遊技媒体としてメダルを用いる。なお、遊技媒体としては、メダル以外にも、例えば、コイン、遊技球、遊技用のポイントデータ又はトークン等を適用することもできる。

【0 0 1 9】

遊技者によりパチスロにメダルが投入され、スタートレバーが操作されると、予め定められた数値範囲（例えば、0～6 5 5 3 5）の乱数から 1 つの値（以下、乱数値という）が

50

抽出される。

【0020】

内部抽籤手段は、抽出された乱数値に基づいて抽籤を行い、内部当籤役を決定する。この内部抽籤手段は、後述の主制御回路が備える各種処理手段（処理機能）の一つである。内部当籤役の決定により、後述の有効ライン（入賞判定ライン）に沿って表示を行うことを許可する図柄の組合せが決定される。なお、図柄の組合せの種別としては、メダルの払い出し、再遊技（リプレイ）の作動、ボーナスの作動等といった特典が遊技者に与えられる「入賞」に係るものと、それ以外のいわゆる「はずれ」に係るものとが設けられる。なお、以下では、メダルの払い出しに係る役を「小役」と称し、再遊技（リプレイ）の作動に係る役を「リプレイ役」と称する。また、ボーナスの作動（ボーナスゲーム）に係る役を「ボーナス役」ともいう。

10

【0021】

また、スタートレバーが操作されると、複数のリールの回転が行われる。その後、遊技者により所定のリールに対応するストップボタンが押されると、リール停止制御手段は、内部当籤役とストップボタンが押されたタイミングとに基づいて、該当するリールの回転を停止する制御を行う。このリール停止制御手段は、後述の主制御回路が備える各種処理手段（処理機能）の一つである。

【0022】

パチスロでは、基本的に、ストップボタンが押されたときから規定時間（190msec）内に、該当するリールの回転を停止する制御が行われる。本実施形態では、この規定時間内にリールの回転に伴って移動する図柄の数を「滑り駒数」という。そして、本実施形態では、規定期間が190msecである場合には、滑り駒数の最大数（最大滑り駒数）を図柄4個分に定める。

20

【0023】

リール停止制御手段は、入賞に係る図柄の組合せ表示を許可する内部当籤役が決定されているときは、通常、190msec（図柄4駒分）の規定時間内に、その図柄の組合せが有効ラインに沿って極力表示されるようにリールの回転を停止させる。また、リール停止制御手段は、規定時間を利用して、内部当籤役によってその表示が許可されていない図柄の組合せが有効ラインに沿って表示されないようにリールの回転を停止させる。

【0024】

このようにして、複数のリールの回転がすべて停止されると、入賞判定手段は、有効ラインに沿って表示された図柄の組合せが、入賞に係るものであるか否かの判定を行う。この入賞判定手段もまた、後述の主制御回路が備える各種処理手段（処理機能）の一つである。そして、表示された図柄の組合せが、入賞判定手段により入賞に係るものであると判定されると、メダルの払い出し等の特典が遊技者に与えられる。パチスロでは、以上のような一連の流れが1回の遊技（単位遊技）として行われる。

30

【0025】

また、パチスロでは、前述した一連の遊技動作の流れの中で、表示装置などによる映像の表示、各種ランプによる光の出力、スピーカによる音の出力、或いは、これらの組合せを利用して様々な演出が行われる。

40

【0026】

具体的には、スタートレバーが操作されると、上述した内部当籤役の決定に用いられた乱数値とは別に、演出用の乱数値が抽出される。演出用の乱数値が抽出されると、演出内容決定手段は、内部当籤役に対応づけられた複数種類の演出内容の中から今回実行する演出を抽籤により決定する。この演出内容決定手段は、後述の副制御回路が備える各種処理手段（処理機能）の一つである。

【0027】

次いで、演出内容決定手段により演出内容が決定されると、演出実行手段は、リールの回転開始時、各リールの回転停止時、入賞の有無の判定時等の各契機に連動させて対応する演出を実行する。このように、パチスロでは、例えば、内部当籤役に対応づけられた演出

50

内容を実行することによって、決定された内部当籤役（言い換えると、狙うべき図柄の組合せ）を知る機会又は予想する機会が遊技者に提供され、遊技者の興味の向上を図ることができる。

【００２８】

＜パチスロの構造＞

次に、図２～図４を参照して、本発明の第１実施形態に係るパチスロの構造について説明する。

【００２９】

〔外観構造〕

図２は、パチスロ１の外部構造を示す斜視図である。

10

【００３０】

パチスロ１は、図２に示すように、外装体（遊技機本体）２を備える。外装体２は、リールや回路基板等を収容するキャビネット２ａと、キャビネット２ａの開口を開閉可能に取り付けられるフロントドア２ｂとを有する。

【００３１】

キャビネット２ａの内部には、３つのリール３Ｌ、３Ｃ、３Ｒ（変動表示手段、表示列）が横一列に並べて設けられている。以下、各リール３Ｌ、３Ｃ、３Ｒ（メインリール）を、それぞれ左リール３Ｌ、中リール３Ｃ、右リール３Ｒともいう。各リール３Ｌ、３Ｃ、３Ｒは、円筒状に形成されたリール本体と、リール本体の周面に装着された透光性のシート材を有する。そして、シート材の表面には、複数（例えば２０個）の図柄が周方向（リールの回転方向）に沿って所定の間隔をあけて描かれている。

20

【００３２】

フロントドア２ｂは、ドア本体９と、フロントパネル１０と、腰部パネル１２と、台座部１３とを備える。ドア本体９は、ヒンジ（不図示）を用いてキャビネット２ａに開閉可能に取り付けられる。ヒンジは、パチスロ１の前方側（遊技者側）から見て、ドア本体９の左側の側端部に設けられる。

【００３３】

フロントパネル１０は、ドア本体９の上部に設けられている。このフロントパネル１０は、開口１０ａを有する枠状部材で構成される。フロントパネル１０の開口１０ａは、表示装置カバー３０によって塞がれ、表示装置カバー３０は、キャビネット２ａの内部に配置された後述の表示装置１１と対向して配置される。

30

【００３４】

表示装置カバー３０は、黒色の半透明な合成樹脂により形成される。それゆえ、遊技者は、後述の表示装置１１により表示された映像（画像）を、表示装置カバー３０を介して視認することができる。また、本実施形態では、表示装置カバー３０を黒色の半透明な合成樹脂で形成することにより、キャビネット２ａ内への外光の入り込みを抑制して、表示装置１１により表示された映像（画像）を鮮明に視認できるようにしている。

【００３５】

フロントパネル１０には、ランプ群２１が設けられている。ランプ群２１は、例えば、遊技者側から見て、フロントパネル１０の上部に設けられたランプ２１ａ、２１ｂを含む。ランプ群２１を構成する各ランプは、ＬＥＤ（Light Emitting Diode）等で構成され（後述の図７中のＬＥＤ群８５参照）、演出内容に対応するパターンで、光を点灯及び消灯する。

40

【００３６】

腰部パネル１２は、ドア本体９の略中央部に設けられる。腰部パネル１２は、任意の画像が描かれた装飾パネルと、この装飾パネルを背面側から照明するための光を出射する光源（後述のＬＥＤ群８５に含まれるＬＥＤ）とを有する。

【００３７】

台座部１３は、フロントパネル１０と腰部パネル１２との間に設けられる。台座部１３には、図柄表示領域４と、遊技者による操作の対象となる各種装置（メダル投入口１４、Ｍ

50

MAXベットボタン15a、1ベットボタン15b、スタートレバー16、3つのストップボタン17L、17C、17R、精算ボタン（不図示）等）とが設けられる。

【0038】

図柄表示領域4は、正面から見て、3つのリール3L、3C、3Rに重畳する領域で、かつ、3つのリール3L、3C、3Rより遊技者側の位置に配置されており、3つのリール3L、3C、3Rを視認可能にするサイズを有する。この図柄表示領域4は、表示窓としての機能を果たすものであり、その背後に設けられた各リール3L、3C、3Rを視認することが可能な構成になっている。以下、図柄表示領域4を、リール表示窓4という。

【0039】

リール表示窓4は、その背後に設けられた3つのリール3L、3C、3Rの回転が停止されたとき、各リールの周面に設けられた複数の図柄のうち、連続して配置された3つの図柄がその枠内に表示されるように構成されている。すなわち、3つのリール3L、3C、3Rの回転が停止されたとき、リール表示窓4の枠内には、リール毎に上段、中段及び下段の各領域にそれぞれ1個の図柄（合計で3個）が表示される（リール表示窓4の枠内には、3行×3列の態様で図柄が表示される）。そして、本実施形態では、リール表示窓4の枠内において、左リール3Lの中段領域、中リール3Cの中段領域、及び、右リール3Rの中段領域を結ぶ擬似的なライン（センターライン）を、入賞か否かの判定を行う有効ラインとして定義する。

10

【0040】

リール表示窓4は、台座部13に設けられた枠部材31の開口により形成される。また、リール表示窓4を画成する枠部材31の下方には、略水平面の台座領域が設けられる。そして、遊技者側から見て、台座領域の右側にはメダル投入口14が設けられ、左側にはMAXベットボタン15a及び1ベットボタン15bが設けられる。

20

【0041】

メダル投入口14は、遊技者によって外部からパチスロ1に投下されるメダルを受け入れるために設けられる。メダル投入口14から受け入れられたメダルは、予め設定された所定枚数（例えば3枚）を上限として1回の遊技に使用され、所定枚数を超えたメダルの枚数分は、パチスロ1の内部に預けることができる（いわゆるクレジット機能（遊技媒体貯留手段））。

【0042】

MAXベットボタン15a及び1ベットボタン15bは、キャビネット2aの内部に預けられているメダルから1回の遊技に使用する枚数を決定するために設けられる。なお、MAXベットボタン15aの内部には、メダル投入が可能に点灯するベットボタンLED（不図示）が設けられている。また、精算ボタンは、パチスロ1の内部に預けられているメダルを外部に引き出す（排出する）ために設けられる。

30

【0043】

なお、遊技者がMAXベットボタン15aを押下操作すると、単位遊技のベット枚数（3枚）のメダルが投入され、有効ラインが有効化される。一方、1ベットボタン15bが1回、押下操作される度に1枚のメダルが投入される。1ベットボタン15bが3回操作されると、単位遊技のベット枚数（3枚）のメダルが投入され、有効ラインが有効化される。

40

【0044】

なお、以下では、MAXベットボタン15aの操作、1ベットボタン15bの操作及びメダル投入口14にメダルを投入する操作（遊技を行うためにメダルを投入する操作）をいずれも「投入操作」という。

【0045】

スタートレバー16は、全てのリール（3L、3C、3R）の回転を開始するために設けられる。ストップボタン17L、17C、17Rは、それぞれ、左リール3L、中リール3C、右リール3Rに対応づけて設けられ、各ストップボタンは対応するリールの回転を停止するために設けられる。以下、ストップボタン17L、17C、17Rを、それぞれ左ストップボタン17L、中ストップボタン17C、右ストップボタン17Rともいう。

50

【0046】

また、リール表示窓4の下方の略水平面の台座領域の略中央には、情報表示器6が設けられる。なお、情報表示器6は、透明の窓カバー（不図示）によって覆われている。

【0047】

情報表示器6には、特典として遊技者に対して払い出されるメダルの枚数（以下、「払出枚数」という）の情報を遊技者に対してデジタル表示（報知）するための2桁の7セグメントLED（以下、「7セグLED」という）や、パチスロ1の内部に預けられているメダルの枚数（以下、「クレジット枚数」という）などの情報を遊技者に対してデジタル表示（報知）するための2桁の7セグLEDが設けられる。なお、本実施形態では、メダルの払出枚数表示用の2桁の7セグLEDは、エラー発生及びエラー種別の情報を遊技者に対してデジタル表示（報知）するための2桁の7セグLEDとしても用いられる。それゆえ、エラー発生時には、メダルの払出枚数表示用の2桁の7セグLEDの表示態様は、払出枚数の表示態様からエラー種別の情報の表示態様に切り替わる。

【0048】

さらに、情報表示器6には、内部当籤役として決定された役に応じた図柄組合せを有効ラインに沿って表示するために必要な停止操作の情報を報知する指示モニタ（不図示）が設けられている。指示モニタ（指示表示器）は、例えば、2桁の7セグメントLEDにより構成される。そして、指示モニタでは、報知する停止操作の情報と一義的に対応する態様で、2桁の7セグLEDが点灯、点滅又は消灯することにより、遊技者に対して必要な停止操作の情報を報知する。

【0049】

なお、ここでいう、報知する停止操作の情報と一義的に対応する態様とは、例えば、押し順「1st（第1停止操作を左リール3Lに対して行うこと）」を報知する場合には指示モニタに数値「1」を表示し、押し順「2nd（第1停止操作を中リール3Cに対して行うこと）」を報知する場合には指示モニタに数値「2」を表示し、押し順「3rd（第1停止操作を右リール3Rに対して行うこと）」を報知する場合には指示モニタに数値「3」を表示するなどの態様のことである。なお、指示モニタにおける停止操作の情報の報知態様（後述のメイン側で決定されるナビデータ）については、後述の図63を参照しながら後で詳述する。

【0050】

情報表示器6は、後述の図7に示すように、ドア中継端子板68及び遊技動作表示基板81を介して主制御基板71に電氣的に接続され、情報表示器6の表示動作は、主制御基板71内の後述の主制御回路90により制御される。また、上述した各種7セグLEDの制御方式は、ダイナミック点灯制御である。

【0051】

なお、本実施形態のパチスロ1では、主制御基板71により制御される指示モニタに加えて、副制御基板72により制御される他の手段を用いて停止操作の情報を報知する構成を設ける。具体的には、後述のプロジェクト機構211及び表示ユニット212（図3及び後述の図7参照）により構成される後述の表示装置11により停止操作の情報を報知する。

【0052】

このような構成を適用した場合、指示モニタにおける報知の態様と、副制御基板72により制御されるその他の手段における報知の態様とは、互いに異なる態様であってもよい。すなわち、指示モニタでは、報知する停止操作の情報と一義的に対応する態様で報知すればよく、必ずしも、停止操作の情報を直接的に報知する必要はない（例えば、指示モニタにおいて数値「1」が表示されたとしても、遊技者によっては報知内容を特定できない可能性もあり、直接的な報知とは言えない）。一方、後述の表示装置11等のその他の手段によるサブ側（副制御基板側）での報知では、停止操作の情報を直接的に報知してもよい。例えば、押し順「1st」を報知する場合、指示モニタでは報知する押し順と一義的に対応する数値「1」を表示するが、その他の手段（例えば、表示装置11等）では、左リール3Lに対して第1停止操作を行わせるための指示情報を直接的に報知してもよい。

【0053】

このような構成のパチスロ1では、副制御基板72の制御だけでなく、主制御基板71の制御によっても、内部当籤役に応じた必要な停止操作の情報を報知することができる。また、このような停止操作の情報の報知の有無は、遊技状態に応じて制御されるようにしてもよい。例えば、後述の一般遊技状態（非ART遊技状態）では停止操作の情報を報知せずに、後述のART遊技状態（後述の図14参照）において停止操作の情報を報知するようにしてもよい。

【0054】

また、遊技者側から見て、リール表示窓4の左方には、サブ表示装置18が設けられる。サブ表示装置18は、図2に示すように、ドア本体9の前面部のうち、台座部13の略水平面の台座領域から略垂直に立設するように設けられる。サブ表示装置18は、液晶ディスプレイや有機EL（Electro-Luminescence）ディスプレイで構成され、各種情報を表示する。

10

【0055】

また、サブ表示装置18の表示面上には、タッチセンサ19が設けられている（後述の図7参照）。タッチセンサ19は、静電容量方式などの所定の動作原理に従い動作し、遊技者の操作を受け付けると、タッチ入力情報として当該操作に応じた信号を出力する。そして、本実施形態のパチスロ1は、タッチセンサ19を介して受け付けた遊技者の操作（タッチセンサ19から出力されるタッチ入力情報）に応じて、サブ表示装置18の表示を切り替え可能にする機能を有する。なお、サブ表示装置18は、タッチセンサ19から出力されるタッチ入力情報に基づいて後述の副制御基板72（後述の図7参照）により制御される。

20

【0056】

ドア本体9の下部には、メダル払出口24、メダル受皿25、2つのスピーカ用孔20L、20R等が設けられる。メダル払出口24は、後述のメダル払出装置51の駆動により排出されるメダルを外部に導く。メダル受皿25は、メダル払出口24から排出されたメダルを貯める。また、2つのスピーカ用孔20L、20Rからは、演出内容に対応する効果音や楽曲等の音声出力される。

【0057】

〔内部構造〕

次に、パチスロ1の内部構造を、図3及び図4を参照しながら説明する。図3は、キャビネット2aの内部構造を示す図であり、図4は、フロントドア2bの裏面側の内部構造を示す図である。

30

【0058】

キャビネット2aは、図3に示すように、上面板27aと、底面板27bと、左右の側面板27c、27dと、背面板27eとを有する。そして、キャビネット2a内の上部には、表示装置11が配設される。

【0059】

表示装置11は、プロジェクタ機構211と、プロジェクタ機構211から投射された映像光が投影される箱状の被投影部材212aとを有し、プロジェクションマッピングによる映像表示を行う。具体的には、表示装置11では、立体物となる被投影部材212aの位置（投影距離や角度など）や形状に基づいて映像光を生成し、その映像光が、プロジェクタ機構211により被投影部材212aの表面に投影される。このような演出機能を設けることにより、高度で且つ迫力のある演出を行うことができる。また、図3には示さないが、箱状の被投影部材212aの裏側には、表示面が湾曲した別の被投影部材が設けられ、遊技状態に応じて、どちらか一方の被投影部材が、映像光が投影されるスクリーンとして使用される。それゆえ、キャビネット2a内は、遊技状態に応じて、被投影部材を切り換える機能（不図示）も設けられる。

40

【0060】

キャビネット2a内の下部には、メダル払出装置（以下、ホッパー装置という）51と、

50

メダル補助収納庫 5 2 と、電源装置 5 3 とが配設される。

【0061】

ホッパー装置 5 1 は、キャビネット 2 a における底面板 2 7 b の中央部に取り付けられる。このホッパー装置 5 1 は、多量のメダルを収容可能で、それらを 1 枚ずつ排出可能な構造を有する。ホッパー装置 5 1 は、貯留されたメダルが例えば 5 0 枚を超えたとき、又は、精算ボタンが押下されてメダルの精算が実行されるときに、メダルを払い出す。そして、ホッパー装置 5 1 によって払い出されたメダルは、メダル払出口 2 4（図 2 参照）から排出される。

【0062】

メダル補助収納庫 5 2 は、ホッパー装置 5 1 から溢れ出たメダルを収納する。このメダル補助収納庫 5 2 は、キャビネット 2 a 内部を正面から見て、ホッパー装置 5 1 の右側に配置される。また、メダル補助収納庫 5 2 は、キャビネット 2 a の底面板 2 7 b に対して着脱可能に取り付けられている。

10

【0063】

電源装置 5 3 は、電源スイッチ 5 3 a と、電源基板 5 3 b（電源供給手段）とを有している（後述の図 7 参照）。この電源装置 5 3 は、キャビネット 2 a 内部を正面から見て、ホッパー装置 5 1 の左側に配置されており、左側面板 2 7 c に取り付けられている。電源装置 5 3 は、サブ電源装置（不図示）から供給された交流電圧 1 0 0 V の電力を各部で必要な直流電圧の電力に変換して、変換した電力を各部へ供給する。

【0064】

また、キャビネット 2 a 内の電源装置 5 3 の上方には、副制御基板 7 2（後述の図 7 参照）を収容する副制御基板ケース 5 7 が配設される。副制御基板ケース 5 7 に収納された副制御基板 7 2 には、後述の副制御回路 2 0 0（後述の図 1 0 参照）が搭載されている。この副制御回路 2 0 0 は、映像の表示等による演出の実行を制御する回路である。副制御回路 2 0 0 の具体的な構成については後述する。

20

【0065】

キャビネット 2 a 内の副制御基板ケース 5 7 の上方には、副中継基板 6 1 が配設される。この副中継基板 6 1 は、副制御基板 7 2 と後述の主制御基板 7 1 とを接続する配線が実装された中継基板である。また、副中継基板 6 1 は、副制御基板 7 2 と副制御基板 7 2 の周辺に配設された基板や各種装置部（ユニット）などとを接続する配線が実装された中継基板である。

30

【0066】

また、図 3 には示さないが、キャビネット 2 a 内には、キャビネット側中継基板 4 4（後述の図 7 参照）が配設される。このキャビネット側中継基板 4 4 は、主制御基板 7 1（後述の図 7 参照）と、ホッパー装置 5 1、遊技メダル補助収納庫スイッチ 7 7（後述の図 7 参照）及びメダル払出カウントスイッチ（不図示）のそれぞれとを接続する配線が実装された中継基板である。

【0067】

フロントドア 2 b の裏面側の中央部には、図 4 に示すように、ミドルドア 4 1 が、配設され、リール表示窓 4（図 2 参照）を裏側から開閉可能に取り付けられている。また、図 4 には示さないが、ミドルドア 4 1 のリール表示窓 4 側には、3 つのリール 3 L、3 C、3 R が取り付けられ、ミドルドア 4 1 のリール表示窓 4 側とは反対側には、主制御基板 7 1（後述の図 7 参照）が収納された主制御基板ケース 5 5 が取り付けられている。なお、3 つのリール 3 L、3 C、3 R には、所定の減速比をもったギアを介してステッピングモータ（不図示）が接続されている。

40

【0068】

主制御基板ケース 5 5 に収納された主制御基板 7 1 は、後述する主制御回路 9 0（後述の図 9 参照）を有する。主制御回路 9 0（主制御手段）は、内部当籤役の決定、各リール 3 L、3 C、3 R の回転及び停止、入賞の有無の判定といった、パチスロ 1 における遊技の主な流れを制御する回路である。また、本実施形態では、例えば、ART の決定の有無の

50

抽籤処理、ナビ情報の指示モニタへの表示処理、各種試験信号の送信処理などの制御も主制御回路 90 により行われる。なお、主制御回路 90 の具体的な構成は後述する。

【0069】

フロントドア 2b の裏面側において、ミドルドア 41 の下方には、スピーカ 65L, 65R が配設される。スピーカ 65L, 65R は、それぞれスピーカ用孔 20L, 20R (図 2 参照) と対向する位置に配置されている。

【0070】

また、スピーカ 65L の上方には、セクタ 66 と、ドア開閉監視スイッチ 67 とが配設される。セクタ 66 は、メダルの材質や形状等が適正であるか否かを選別する装置であり、メダル投入口 14 に投入された適正なメダルをホッパー装置 51 へ案内する。セクタ 66 内においてメダルが通過する経路上には、適正なメダルが通過したことを検出するメダルセンサ (遊技媒体検出手段: 不図示) が設けられている。

【0071】

ドア開閉監視スイッチ 67 は、フロントドア 2b を裏面側から見て、セクタ 66 の左斜め下に配置される。このドア開閉監視スイッチ 67 は、フロントドア 2b の開閉を報知するためのセキュリティ信号をパチスロ 1 の外部に出力する。

【0072】

また、図 4 には示さないが、フロントドア 2b を裏面において、ミドルドア 41 により開閉された領域であり且つリール表示窓 4 の下方には、ドア中継端子板 68 が配設される (後述の図 7 参照)。このドア中継端子板 68 は、主制御基板ケース 55 内の主制御基板 71 と、各種のボタンやスイッチ、副中継基板 61、セクタ 66、遊技動作表示基板 81、試験機用第 1 インターフェースボード 301 及び試験機用第 2 インターフェースボード 302 のそれぞれとを接続する配線が実装された中継基板である。なお、各種のボタン及びスイッチとしては、例えば、MAX ベットボタン 15a、1 ベットボタン 15b、ドア開閉監視スイッチ 67、後述の BET スイッチ 77、スタートスイッチ 79 等が挙げられる。

【0073】

＜サブ表示装置の表示例＞

ここで、図 5A ～図 5E を参照して、サブ表示装置 18 に表示される各種表示画面について説明する。なお、図 5A は、サブ表示装置 18 に表示されるトップ画面 221 を示す図であり、図 5B は、サブ表示装置 18 に表示されるメニュー画面 222 を示す図である。また、図 5C ～図 5E は、サブ表示装置 18 に表示される遊技情報画面 223, 224, 225 を示す図である。

【0074】

サブ表示装置 18 には、遊技者のタッチ操作により様々な表示画面が表示され、図 5A ～図 5E に示すように、トップ画面 221、メニュー画面 222 及び遊技情報画面 223, 224, 225 を含む各種表示画面が表示される。これらの表示画面は、タッチセンサ 19 を介して受け付けた遊技者の操作信号に基づいて切り替えられる。

【0075】

トップ画面 221 は、サブ表示装置 18 に表示される表示画面のうちの初期画面であり、トップ画面 221 では、「MENU」ボタン 221a と、概要遊技履歴 221b とが表示される。「MENU」ボタン 221a は、図 5B に示すメニュー画面 222 を呼び出すための操作ボタンであり、「MENU」ボタン 221a に対して遊技者による所定操作 (例えばタップ) が行われると、メニュー画面 222 が呼び出される。また、トップ画面 221 では、概要遊技履歴 221b として、パチスロ 1 の一部の遊技履歴 (概要遊技履歴) を表示する。本実施形態では、概要遊技履歴 221b として、例えば、ボーナス回数、ART 回数及びゲーム数 (遊技回数) が表示される。

【0076】

メニュー画面 222 は、サブ表示装置 18 で表示可能なメニューを表示する画面であり、メニュー画面 222 では、「戻る」ボタン 222a、「登録」ボタン 222b、「説明」

10

20

30

40

50

ボタン２２２ｃ、「配列配当」ボタン２２２ｄ、「リーチ目」ボタン２２２ｅ、「WEBサイト」ボタン２２２ｆ及び「音量」ボタン２２２ｇが表示される。「戻る」ボタン２２２ａは、トップ画面２２１を呼び出すための操作ボタンであり、「戻る」ボタン２２２ａに対して遊技者による操作が行われると、トップ画面２２１が呼び出される。また、「登録」ボタン２２２ｂ～「音量」ボタン２２２ｇは、対応するメニュー内容の表示画面を呼び出すための操作ボタンであり、各ボタンに対して遊技者による操作が行われると、対応するメニュー内容の表示画面が呼び出される。

【００７７】

例えば、メニュー画面２２２において「登録」ボタン２２２ｂが遊技者により操作された場合、遊技中の遊技者を登録するための登録画面（不図示）がサブ表示装置１８の表示画面に呼び出される。近年のパチスロでは、機種ごとに遊技者を登録しておき、当該遊技者のこれまでの遊技履歴から、定められたミッションの達成状況などの様々な情報を管理するサービスが広く行われている。「登録」ボタン２２２ｂにより呼び出される登録画面は、このサービスの提供を受ける際に遊技者を登録するための表示画面である。

【００７８】

また、例えば、メニュー画面２２２において「説明」ボタン２２２ｃが遊技者により操作された場合、パチスロ１の説明画面（不図示）がサブ表示装置１８の表示画面に呼び出される。説明画面で表示される情報には、例えば、設定値ごとのボーナス当籤確率やART当籤確率などのパチスロ１の仕様に関する説明や、パチスロ１の演出に登場するキャラクターの紹介説明などが含まれる。

【００７９】

また、例えば、メニュー画面２２２において「配列配当」ボタン２２２ｄが遊技者により操作された場合、パチスロ１の配列配当画面（不図示）がサブ表示装置１８の表示画面に呼び出される。配列配当画面には、例えば、パチスロ１において入賞と判定される図柄の組合せと、入賞と判定された際の特典との対応関係（配当表）や、各リール３Ｌ，３Ｃ，３Ｒに描かれた図柄列（リール配列）などが表示される。

【００８０】

また、例えば、メニュー画面２２２において「リーチ目」ボタン２２２ｅが遊技者により操作された場合、パチスロ１のリーチ目画面（不図示）がサブ表示装置１８の表示画面に呼び出される。リーチ目画面には、パチスロ１で設定されている「リーチ目」と称される図柄組合せの情報が表示される。なお、「リーチ目」と称する図柄組合せは、該図柄組合せが有効ラインに沿って表示されることにより、特別な特典が付与される図柄組合せであり、本実施形態のパチスロ１では、後述の図２８～図３０の入賞作動フラグ格納領域の内容欄に示す略称「リーチ目リブ」に対応する図柄組合せが該当する。そして、本実施形態では、「リーチ目リブ」に係る図柄組合せが有効ラインに沿って表示された場合、その後、遊技者にとって有利な状態（例えば、ボーナス状態、通常ART又はCT（後述の図１４参照））に移行することが確定する。

【００８１】

また、例えば、メニュー画面２２２において「WEBサイト」ボタン２２２ｆが遊技者により操作された場合、パチスロ１のWEB紹介画面（不図示）がサブ表示装置１８の表示画面に呼び出される。WEB紹介画面には、例えば、パチスロ１の機種ごとに設けられた特設WEBサイトやパチスロ１のメーカーのWEBサイトなど任意のWEBサイトのURLを示す二次元コード（例えば、QRコード（登録商標））が表示される。遊技者は、携帯電話などでWEB紹介画面に表示される二次元コードを読み込むことにより、対応するWEBサイトにアクセスすることができる。

【００８２】

また、例えば、メニュー画面２２２において「音量」ボタン２２２ｇが遊技者により操作された場合、スピーカ６５Ｌ，６５Ｒから出力する音の音量を調整することが可能な音量調整画面（不図示）がサブ表示装置１８の表示画面に呼び出される。遊技者は、音量調整画面を介してパチスロ１の演出音の音量を調整することができる。

【0083】

なお、サブ表示装置18は、上述した表示装置11（プロジェクタ機構211及び表示ユニット212）とは別体に設けられるため、表示装置11とは別個に制御することができる。それゆえ、本実施形態のパチスロ1では、遊技中（表示装置11による演出の実行中）であっても、サブ表示装置18の表示画面を遊技者の操作により切り替えることができる。その結果、例えば、遊技者が、表示装置11の演出において登場するキャラクタのことを知りたいと思った場合、遊技者は、「説明」ボタン222cを操作して説明画面を呼び出すことにより、キャラクタ間の関係性などの情報を遊技中に把握することができる。また、例えば、遊技者が、遊技中に、いわゆる「レア役」が当籤した場合のリール回転中にレア役を入賞させるために目安とすべき図柄を把握したいと思った場合、遊技者は、「配列配当」ボタン222dを操作して配列配当画面を呼び出すことにより、リール配列を把握することができる。

10

【0084】

遊技情報画面223、224、225は、パチスロ1の遊技履歴のうちのトップ画面221に表示する概要遊技履歴を含む詳細遊技履歴情報を表示する表示画面である。

【0085】

遊技情報画面223には、「戻る」ボタン223aと、「MENU」ボタン223bと、「前へ」ボタン223cと、「次へ」ボタン223dと、遊技履歴223eとが表示される。「戻る」ボタン223a及び「MENU」ボタン223bは、それぞれトップ画面221及びメニュー画面222をサブ表示装置18の表示画面に呼び出すための操作ボタンであり、各ボタンを遊技者が操作することにより、対応する表示画面が呼び出される。また、「前へ」ボタン223c及び「次へ」ボタン223dは、遊技情報画面を所定の順序で切り替えるための操作ボタンであり、「前へ」ボタン223cが遊技者により操作されると、表示画面が遊技情報画面223から遊技情報画面225に切り替わり、「次へ」ボタン223dが遊技者により操作されると、表示画面が遊技情報画面223から遊技情報画面224に切り替わる。また、遊技履歴223eとしては、図5Cに示すように、ゲーム数（遊技回数）、ボーナス回数、ART回数及びCZ（チャンスゾーン）回数が表示される。

20

【0086】

遊技情報画面224には、「戻る」ボタン224aと、「MENU」ボタン224bと、「前へ」ボタン224cと、「次へ」ボタン224dと、遊技履歴214eとが表示される。「戻る」ボタン224a及び「MENU」ボタン224bは、それぞれトップ画面221及びメニュー画面222をサブ表示装置18の表示画面に呼び出すための操作ボタンであり、各ボタンを遊技者が操作することにより、対応する表示画面が呼び出される。また、「前へ」ボタン224c及び「次へ」ボタン224dは、遊技情報画面を所定の順序で切り替える操作ボタンであり、「前へ」ボタン224cが遊技者に操作されると、表示画面が遊技情報画面224から遊技情報画面223に切り替わり、「次へ」ボタン224dが遊技者により操作されると、表示画面が遊技情報画面224から遊技情報画面225に切り替わる。また、遊技履歴224eとしては、後述のCZ（チャンスゾーン）の突入回数及び成功回数が表示される。なお、後述するように、本実施形態では、CZとして、CZ1、CZ2、CZ3の3種類のCZが設けられる。それゆえ、遊技履歴224eとしては、図5Dに示すように、CZ1～CZ3のそれぞれの突入回数及び成功回数が表示される。

30

40

【0087】

遊技情報画面225には、「戻る」ボタン225aと、「MENU」ボタン225bと、「前へ」ボタン225cと、「次へ」ボタン225dと、遊技履歴225eとが表示される。「戻る」ボタン225a及び「MENU」ボタン225bは、それぞれトップ画面221及びメニュー画面222をサブ表示装置18の表示画面に呼び出すための操作ボタンであり、各ボタンを遊技者が操作することにより、対応する表示画面が呼び出される。また、「前へ」ボタン225c及び「次へ」ボタン225dは、所定の順序で遊技情報画面

50

を切り替えるための操作ボタンであり、「前へ」ボタン 225c が遊技者に操作されると、表示画面が遊技情報画面 225 から遊技情報画面 224 に切り替わり、「次へ」ボタン 225d が遊技者により操作されると、表示画面が遊技情報画面 225 から遊技情報画面 223 に切り替わる。また、遊技履歴 225e としては、図 5E に示すように、小役の当籤回数及び当籤確率（分子が 1 の分数）が表示される。

【0088】

なお、サブ表示装置 18 に表示される表示画面の切り替え手法としては、例えば、それぞれの表示画面に表示される操作ボタンに対するタップ操作に基づいて切り替える手法を採用してもよいし、また、例えば、表示画面に対するスワイプ操作に基づいて切り替える手法を採用してもよい。

【0089】

<サブ表示装置の表示画面の各種切り替え機能>

次に、本実施形態のパチスロ 1 におけるサブ表示装置 18 の表示画面の各種切り替え機能について説明する。

【0090】

[サブ表示装置の表示画面の遷移例]

まず、図 6A 及び 6B を参照して、本実施形態のパチスロ 1 におけるサブ表示装置 18 の表示画面の遷移例（切り替え態様）について説明する。なお、図 6A は、遊技者登録状態がセットされていない状況におけるサブ表示装置 18 の表示画面の遷移例を示す図であり、図 6B は、遊技者登録状態がセットされている状況におけるサブ表示装置 18 の表示画面の遷移例を示す図である。

【0091】

遊技者登録状態がセットされていない状況では、図 6A に示すように、サブ表示装置 18 の表示画面は、トップ画面 221 とメニュー画面 222 との間、並びに、メニュー画面 222 とメニュー画面 222 から遷移可能な各種表示画面との間でのみ遷移可能であり、これらの表示画面間の遷移は副制御基板 72（後述のサブ CPU 201）により制御される。例えば、副制御基板 72 は、タッチセンサ 19 を介して取得したタッチ操作（例えば、所定のボタンに対するタップ操作や、表示画面上におけるスワイプ操作）に基づいて、トップ画面 221 及びメニュー画面 222 間で、表示画面を切り替える。しかしながら、遊技者登録状態がセットされていない状況では、副制御基板 72 は、遊技情報画面 223，224，225 の表示が不可能となるように制御する。すなわち、遊技者登録状態がセットされていない状況では、遊技者は、サブ表示装置 18 に遊技情報画面 223，224，225 を表示することができない。

【0092】

一方、遊技者登録状態がセットされている状況では、図 6B に示すように、サブ表示装置 18 の表示画面は、トップ画面 221 とメニュー画面 222 との間、並びに、メニュー画面 222 とメニュー画面 222 から遷移可能な各種表示画面との間に加え、メニュー画面 222 と遊技情報画面 223，224，225 との間においても遷移可能となり、これらの表示画面間の遷移は副制御基板 72（後述のサブ CPU 201）により制御される。すなわち、副制御基板 72 は、タッチセンサ 19 を介して取得したタッチ操作に基づいて、トップ画面 221 とメニュー画面 222 との間だけでなく、トップ画面 221 及びメニュー画面 222 のそれぞれと、遊技情報画面 223，224，225 との間においても表示画面を切り替えることができる。それゆえ、本実施形態において、遊技者登録状態がセットされている場合、遊技者は、サブ表示装置 18 に遊技情報画面 223，224，225 を表示することができる。

【0093】

なお、図 6B に示すように、トップ画面 221、メニュー画面 222 及び遊技情報画面 223，224，225 間における表示画面の遷移順序は任意である。それゆえ、例えば、トップ画面 221 から遊技情報画面 223，224，225 に直接遷移可能となる構成にしてもよいし、トップ画面 221 からメニュー画面 222 を介してのみ遊技情報画面 22

10

20

30

40

50

3, 224, 225に遷移可能となる構成にしてもよい。

【0094】

本実施形態のパチスロ1では、トップ画面221からメニュー画面222を介してのみ遊技情報画面223, 224, 225に遷移可能な構成(トップ画面221から遊技情報画面223, 224, 225に直接遷移できない構成)を採用している。なお、本実施形態のパチスロ1では、メニュー画面222において、遊技者が表示画面に対してスワイプ操作(メニュー選択操作ではない)を行うことにより、表示画面をメニュー画面222から遊技情報画面223, 224, 225に遷移させることができる。

【0095】

なお、本実施形態では、遊技情報画面223, 224, 225は、メニュー画面222とは完全に独立して設けられた表示画面である。すなわち、本実施形態のパチスロ1では、遊技履歴という、遊技者が遊技中に強い関心を抱く遊技の結果を示す情報を、配当配列や音量調節などの遊技の結果とは関係のない情報として独立して表示する。そして、本実施形態では、遊技者登録状態がセットされている状況において、メニュー画面222に対して遊技者がメニュー選択操作を行うことなく、遊技情報画面223, 224, 225を表示可能にしている。それゆえ、本実施形態では、遊技者登録状態がセットされている場合、遊技者が所望する情遊技履歴情報へのアクセスを容易に行うことができる。

【0096】

また、本実施形態では、メニュー画面222に対するメニュー選択操作では、表示画面を遊技情報画面223, 224, 225に遷移させることができず、メニュー画面222に対してスワイプ操作(メニュー表示では指定されていない操作)を行わなければ、表示画面を遊技情報画面223, 224, 225に遷移させることができない。それゆえ、本実施形態のパチスロ1では、表示画面を遊技情報画面223, 224, 225に遷移させるためのスワイプ操作を、遊技者登録状態がセットされている状況における隠しコマンドとして扱うことができる。この場合、遊技者にとってみれば、パチスロ1に対する自身の知識により、知識の少ない他の遊技者では見ることのできない、より詳細な遊技履歴情報を見ることができるため、当該他の遊技者よりも有利に遊技を行うことができ、結果、遊技者が積極的に遊技を行うことを期待することができる。

【0097】

[遊技情報画面からトップ画面への表示切り替え機能]

本実施形態のパチスロ1は、サブ表示装置18の表示画面を、遊技情報画面223, 224, 225から、遊技者の手動により、又は、自動的に、トップ画面221に遷移させる機能を有する。具体的には、本実施形態では、遊技情報画面223, 224, 225において「戻る」ボタンが操作されると、表示画面が遊技情報画面223, 224, 225からトップ画面221に遷移する(手動遷移機能)。また、本実施形態では、遊技情報画面223, 224, 225が表示されている状態において所定の条件を満たした場合には、遊技者の操作とは関係なく自動的に表示画面がトップ画面221に遷移する(自動遷移機能)。

【0098】

より具体的には、パチスロ1では、遊技情報画面223, 224, 225が表示されている状態において、投入操作(MAXベットボタン15aへの操作、1ベットボタン15bへの操作及びメダル投入口14にメダルを投入する操作)が行われると、サブ表示装置18の表示画面が自動的にトップ画面221に遷移する。なお、ART遊技状態のように、リプレイ役が内部当籤役として決定される確率が高い遊技状態(高リプ状態)では、リプレイ役入賞に伴う再遊技の作動によりメダルが自動的に投入されてしまう結果、高リプ状態では、遊技情報画面223, 224, 225を表示する機会が制限されてしまう可能性がある。そこで、本実施形態のパチスロ1では、再遊技の作動によりメダルが自動的に投入された場合には、メダルの投入操作ではなく、開始操作を契機として、自動的に表示画面が遊技情報画面223, 224, 225からトップ画面221に遷移する。

【0099】

すなわち、本実施形態では、再遊技が作動し、かつ、遊技情報画面 223, 224, 225 が表示されている場合には、開始操作を契機として、自動的に表示画面が遊技情報画面 223, 224, 225 からトップ画面 221 に遷移する。一方、再遊技の作動が行われていない場合には、投入操作を契機として、自動的に表示画面が遊技情報画面 223, 224, 225 からトップ画面 221 に遷移する。

【0100】

〔メニュー内容表示画面からトップ画面（又はメニュー画面）への表示切り替え機能〕

本実施形態のパチスロ 1 は、サブ表示装置 18 の表示画面を、メニュー画面 222 に対するメニュー選択操作により遷移可能な各種メニュー内容表示画面（登録画面、説明画面、配列配当画面、リーチ目画面、WEB 紹介画面及び音量調整画面）から、遊技者の手動により、又は、自動的に、トップ画面 221（又はメニュー画面 222）に遷移させる機能を有する。具体的には、本実施形態では、メニュー内容表示画面において所定のボタン（例えば、「TOP へ戻る」ボタン）が操作されると、表示画面が当該メニュー内容表示画面からトップ画面 221 に遷移する（手動遷移機能）。また、本実施形態では、メニュー内容表示画面において特定のボタン（例えば、「戻る」ボタン）が操作されると、表示画面が当該メニュー内容表示画面からメニュー画面 222 に表示画面を遷移する（手動遷移機能）。

【0101】

さらに、本実施形態では、メニュー内容表示画面が表示されている状態において所定の時間が経過すると、遊技者の操作とは関係なく、自動的に表示画面がトップ画面 221（又はメニュー画面 222）に遷移する（自動遷移機能）。なお、この際、トップ画面 221（又はメニュー画面 222）に自動遷移する契機となる所定の時間は、現在表示しているメニュー内容表示画面の種類に応じて異なる。例えば、パチスロ 1 から出力する音量の調整を行う音量調整画面を長時間表示していると、音量が誤操作により意図しない音量に調整されてしまうおそれがあるだけでなく、誤操作により他の遊技者を不快にしまうおそれもある。それゆえ、音量調整画面では、他のメニュー内容表示画面よりも短い時間で、自動的にトップ画面 221（又はメニュー画面 222）に遷移するように設定されている。一方、登録画面は、遊技者の登録を行い易くするために、他のメニュー内容表示画面よりも長い時間で、自動的にトップ画面 221（又はメニュー画面 222）に遷移するように設定されている。

【0102】

すなわち、各メニュー内容表示画面には、トップ画面 221（又はメニュー画面 222）に自動遷移する契機となる経過時間（自動遷移時間）が、当該メニュー内容表示画面の種別に応じて適宜設定されており、音量調整画面には、他のメニュー内容表示画面よりも短い自動遷移時間が設定され、登録画面には、他のメニュー内容表示画面よりも長い自動遷移時間が設定されている。

【0103】

〔メニューの操作可否の選択機能〕

本実施形態のパチスロ 1 では、サブ表示装置 18 の表示画面を、メニュー画面 222 から、登録画面、説明画面、配列配当画面、リーチ目画面、WEB 紹介画面及び音量調整画面に遷移させることにより、遊技者が、これらのメニュー内容表示画面に応じた各種操作を行うことができ、また、各種情報を確認することができる。なお、このような遊技者がメニュー選択できる機能を遊技店側の設定に応じて制限できるような機能（メニューの操作可否の選択機能）を設けてもよい。

【0104】

例えば、遊技店側の設定により、表示画面をメニュー画面 222 から音量調節画面に遷移不可能にする（例えば、メニュー画面 222 に「音量」ボタン 222g を表示しない）ようにしてもよい。この場合、遊技者による音量調節を不可能にすることができる。

【0105】

<パチスロが備える制御系>

次に、パチスロ 1 が備える制御系について、図 7 を参照して説明する。図 7 は、パチスロ 1 の制御系の構成を示す回路ブロック図である。

【0106】

パチスロ 1 は、ミドルドア 4 1 に設けられた主制御基板 7 1 と、フロントドア 2 b に設けられた副制御基板 7 2 とを有する。また、パチスロ 1 は、主制御基板 7 1 に接続された、リール中継端子板 7 4、設定用鍵型スイッチ 5 4（設定スイッチ）及びキャビネット側中継基板 4 4 を有する。さらに、パチスロ 1 は、キャビネット側中継基板 4 4 を介して主制御基板 7 1 に接続された外部集中端子板 4 7、ホッパー装置 5 1、メダル補助収納庫スイッチ 7 5、リセットスイッチ 7 6 及び電源装置 5 3 を有する。なお、ホッパー装置 5 1 の構成については上述したので、ここでは、その説明を省略する。

10

【0107】

リール中継端子板 7 4 は、各リール 3 L、3 C、3 R のリール本体の内側に配設されている。リール中継端子板 7 4 は、各リール 3 L、3 C、3 R のステッピングモータ（不図示）に電氣的に接続されており、主制御基板 7 1 からステッピングモータに出力される信号を中継する。

【0108】

設定用鍵型スイッチ 5 4 は、主制御基板 ケース 5 5 に設けられる。設定用鍵型スイッチ 5 4 は、パチスロ 1 の設定（設定 1 ～設定 6）を変更するとき、若しくは、パチスロ 1 の設定を確認するときを使用される。

【0109】

キャビネット側中継基板 4 4 は、主制御基板 7 1 と、外部集中端子板 4 7、ホッパー装置 5 1、メダル補助収納庫スイッチ 7 5、リセットスイッチ 7 6 及び電源装置 5 3 のそれぞれとを接続する配線が実装された中継基板である。外部集中端子板 4 7 は、メダル投入信号、メダル払出信号及びセキュリティ信号などの信号をパチスロ 1 の外部へ出力するために設けられる。メダル補助収納庫スイッチ 7 5 は、メダル補助収納庫 5 2 に設けられ、メダル補助収納庫 5 2 がメダルで満杯になっているか否かを検出する。リセットスイッチ 7 6 は、例えば、パチスロ 1 の設定を変更する際に用いられる。

20

【0110】

電源装置 5 3 は、電源基板 5 3 b と、電源基板 5 3 b に接続された電源スイッチ 5 3 a とを有する。電源スイッチ 5 3 a は、パチスロ 1 に必要な電源を供給するときに押下される。電源基板 5 3 b は、キャビネット側中継基板 4 4 を介して主制御基板 7 1 に接続されるとともに、副中継基板 6 1 を介して副制御基板 7 2 にも接続される。

30

【0111】

また、パチスロ 1 は、ドア中継端子板 6 8、並びに、該ドア中継端子板 6 8 を介して、主制御基板 7 1 に接続された、セレクト 6 6、ドア開閉監視スイッチ 6 7、BET スwitch 7 7、精算スイッチ 7 8、スタートスイッチ 7 9、ストップスイッチ基板 8 0、遊技動作表示基板 8 1、副中継基板 6 1、試験機用第 1 インターフェースボード 3 0 1 及び試験機用第 2 インターフェースボード 3 0 2 を有する。なお、セレクト 6 6、ドア開閉監視スイッチ 6 7 及び副中継基板 6 1 については、上述したので、ここでは、それらの説明を省略する。

40

【0112】

BET スwitch 7 7（投入操作検出手段）は、MAX ベットボタン 1 5 a 又は 1 ベットボタン 1 5 b が遊技者により押下されたことを検出する。精算スイッチ 7 8 は、精算ボタン（不図示）が遊技者により押下されたことを検出する。スタートスイッチ 7 9（開始操作検出手段）は、スタートレバー 1 6 が遊技者により操作されたこと（開始操作）を検出する。

【0113】

ストップスイッチ基板 8 0（停止操作検出手段）は、回転しているメインリールを停止させるための回路と、停止可能なメインリールを LED などにより表示するための回路とを備える。また、ストップスイッチ基板 8 0 には、ストップスイッチ（不図示）が設けられ

50

る。ストップスイッチは、各ストップボタン 17 L, 17 C, 17 R が遊技者により押下されたこと（停止操作）を検出する。

【0114】

遊技動作表示基板 81 は、情報表示器（7セグ表示器）6 及び LED 82 に接続される。LED 82 には、例えば、今回の遊技に投入されたメダルの枚数（以下、「投入枚数」という）に対応して点灯する、メダル投入枚数表示用の 3 つの LED（以下、「第 1 LED」～「第 3 LED」という）や、遊技動作表示基板 81 から入力される信号に基づいて、メダル投入が可能であることを表示するマーク、遊技開始を表示するマーク、再遊技を行うマークなどを点灯させる LED などが含まれる。第 1 LED～第 3 LED（表示手段）では、メダルが 1 枚投入されると、第 1 LED が点灯し、メダルが 2 枚投入されると、第 1 及び第 2 LED が点灯し、メダルが 3 枚（遊技開始可能枚数）投入されると、第 1 LED～第 3 LED が点灯する。なお、情報表示器 6 については、上述したので、ここでは、それらの説明を省略する。

10

【0115】

試験機用第 1 インターフェースボード 301 及び試験機用第 2 インターフェースボード 302 はともに、パチスロ 1 の検定試験（試射試験）において、遊技に関する各種信号を試験機に出力する際に用いられる中継基板である（なお、販売用のリリース製品としてのパチスロ 1 にはこれらの中継基板は搭載されていないので、販売用の主制御基板 71 の主制御回路 90 には、試験機用第 1 インターフェースボード 301 及び試験機用第 2 インターフェースボード 302 に接続するために必要な各種電子部品もまた実装されていない）。例えば、遊技に係る主要な動作（例えば、内部抽籤、リール停止制御等）を制御するための試験信号は、試験機用第 1 インターフェースボード 301 を介して出力され、例えば、主制御基板 71 で決定された押し順ナビに係る試験信号などは、試験機用第 2 インターフェースボード 302 を介して出力される。

20

【0116】

副制御基板 72 は、ドア中継端子板 68 及び副中継基板 61 を介して主制御基板 71 に接続される。また、パチスロ 1 は、副中継基板 61 を介して副制御基板 72 に接続された、スピーカ群 84、LED 群 85、24h ドア開閉監視ユニット 63、タッチセンサ 19 及び表示ユニット 212 を有する。なお、タッチセンサ 19 については、上述したので、ここでは、その説明を省略する。

30

【0117】

スピーカ群 84 は、スピーカ 65 L, 65 R や図示しない各種スピーカを含んで構成される。LED 群 85 は、フロントパネル 10 に設けられたランプ群 21 や、腰部パネル 12 の装飾パネルを背面側から照明するための光を出射する光源などを含んで構成される。24h ドア開閉監視ユニット 63 は、ミドルドア 41 の開閉の履歴情報を保存する。また、24h ドア開閉監視ユニット 63 は、ミドルドア 41 が開放されたときに、表示装置 11 によりエラー表示を行うための信号を副制御基板 72（副制御回路 200）に出力する。表示ユニット 212 は、例えば、表示装置 11 を構成する被投影部材 212 a、及び、被投影部材 212 a の裏側に設けられた表示面が湾曲した別の被投影部材を含んで構成される。

40

【0118】

また、パチスロ 1 は、副制御基板 72 に接続された、ロムカートリッジ基板 86 及び液晶中継基板 87 を有する。なお、ロムカートリッジ基板 86 及び液晶中継基板 87 は、副制御基板 72 とともに副制御基板 ケース 57 に収納されている。

【0119】

ロムカートリッジ基板 86 は、サブ CPU 201 により実行される各種制御プログラムと、演出用の画像（映像）、音声（スピーカ群 84）、光（LED 群 85）及び通信のデータを管理するための基板である。液晶中継基板 87 は、副制御基板 72 と、表示装置 11 を構成するプロジェクタ機構 211、及び、サブ表示装置 18 との間の接続配線の中継する基板である。なお、プロジェクタ機構 211 及びサブ表示装置 18 については、上述し

50

たので、ここでは、それらの説明を省略する。

【0120】

<主制御回路>

次に、図8を参照して、主制御基板71に実装される主制御回路90の構成について説明する。図8は、パチスロ1の主制御回路90の構成例を示すブロック図である。

【0121】

主制御回路90は、マイクロプロセッサ91と、クロックパルス発生回路92と、電源管理回路93と、スイッチングレギュレータ94（電源供給手段）とを備える。

【0122】

マイクロプロセッサ91は、遊技機用のセキュリティ機能付きマイクロプロセッサである。なお、本実施形態のマイクロプロセッサ91では、後述するように、ソースプログラム上で規定可能な該マイクロプロセッサ91に特有の様々な命令コード（例えば、後述の「LDQ」命令等：以下、「メインCPU101専用命令コード」という）が設けられている。本実施形態では、このメインCPU101専用命令コードを用いることにより、処理の効率化やプログラム容量の削減などを実現している。マイクロプロセッサ91の内部構成については、後述の図9を参照して詳述し、マイクロプロセッサ91に設けられているメインCPU101専用命令コードについては、後述の主制御回路が実行する各種処理において詳述する。

【0123】

クロックパルス発生回路92は、メインCPU作動用のクロックパルス信号を生成し、該生成したクロックパルス信号をマイクロプロセッサ91に出力する。マイクロプロセッサ91は、入力されたクロックパルス信号に基づいて、制御プログラムを実行する。

【0124】

電源管理回路93は、電源基板53b（図7参照）から供給される直流12Vの電源電圧の変動を管理する。そして、電源管理回路93は、例えば、電源が投入された際（電源電圧が0Vから起動電圧値（10V）を上回った際）には、リセット信号をマイクロプロセッサ91の「XSRST」端子に出力し、電断が発生した際（電源電圧が12Vから停電電圧値（10.5V）を下回った際）には、電断検知信号をマイクロプロセッサ91の「XINT」端子に出力する。すなわち、電源管理回路93は、電源投入時に、マイクロプロセッサ91にリセット信号（起動信号）を出力する手段（起動手段）、及び、電断発生時に、マイクロプロセッサ91に電断検知信号（停電信号）を出力する手段（停電手段）も兼ねる。

【0125】

スイッチングレギュレータ94は、DC／DC変換回路であり、マイクロプロセッサ91の直流駆動電圧（直流5Vの電源電圧）を生成し、該生成した直流駆動電圧をマイクロプロセッサ91の「VCC」端子に出力する。

【0126】

<マイクロプロセッサ>

次に、図9を参照して、マイクロプロセッサ91の内部構成について説明する。図9は、マイクロプロセッサ91の内部構成を示すブロック図である。

【0127】

マイクロプロセッサ91は、メインCPU101と、メインROM102（第1記憶手段）と、メインRAM103（第2記憶手段）と、外部バスインターフェース104と、クロック回路105と、リセットコントローラ106と、演算回路107と、乱数回路110（乱数発生回路）と、パラレルポート111と、割込みコントローラ112と、タイマー回路113と、第1シリアル通信回路114と、第2シリアル通信回路115と、を有する。そして、マイクロプロセッサ91を構成するこれらの各部は信号バス116を介して互いに接続されている。

【0128】

メインCPU101は、クロック回路105で生成されたクロックパルスに基づいて、各

10

20

30

40

50

種制御プログラムを実行して、遊技動作全般に係る制御を行う。ここで、メインCPU 101の制御動作の一例としてリール停止制御について説明する。

【0129】

メインCPU 101は、リールインデックスを検出してから各リール3L, 3C, 3L (メインリール)のステッピングモータに対してパルスを出した回数をカウントする。これにより、メインCPU 101は、各リールの回転角度(主に、リールが図柄何個分だけ回転したか)を管理する。なお、リールインデックスとは、リールが一回転したことを示す情報である。このリールインデックスは、例えば、発光部及び受光部を有する光センサと、各リールの所定の位置に設けられ、各メインリールの回転により発光部と受光部との間に介在される検知片とを備えたリール位置検出部(不図示)により検出される。

10

【0130】

ここで、各リール3L, 3C, 3L (メインリール)の回転角度の管理について、具体的に説明する。ステッピングモータに対して出力されたパルス数は、メインRAM 103に設けられたパルスカウンタによって計数される。そして、図柄1つ分の回転に必要な所定回数のパルスの出力がパルスカウンタで計数される毎に、メインRAM 103に設けられた図柄カウンタが1ずつ加算される。図柄カウンタは、各リールに応じて設けられている。図柄カウンタの値は、リール位置検出部(不図示)によってリールインデックスが検出されるとクリアされる。

【0131】

すなわち、本実施形態では、図柄カウンタを管理することにより、リールインデックスが検出されてから図柄何個分の回転が行われたのかを管理する。したがって、各リールの各図柄の位置は、リールインデックスが検出される位置を基準として検出される。

20

【0132】

メインROM 102には、メインCPU 101により実行される各種制御プログラム、各種データテーブル、副制御回路200に対して各種制御指令(コマンド)を送信するためのデータ等が記憶される。メインRAM 103には、制御プログラムの実行により決定された内部当籤役等の各種データを格納する格納領域が設けられる。なお、メインROM 102及びメインRAM 103の内部構成(メモリーマップ)については、後述の図12を参照して詳述する。

【0133】

外部バスインターフェース104は、マイクロプロセッサ91の外部に設けられた各種構成部(例えば、各リール等)が接続された外部信号バス(不図示)と、マイクロプロセッサ91とを電気的に接続するためのインターフェース回路である。クロック回路105は、例えば分周器(不図示)等を含んで構成され、クロックパルス発生回路92から入力されたCPU作動用のクロックパルス信号を、その他の構成部(例えば、タイマー回路113)で使用される周波数のクロックパルス信号に変換する。なお、クロック回路105で生成されたクロックパルス信号は、リセットコントローラ106にも出力される。

30

【0134】

リセットコントローラ106は、電源管理回路93から入力されたリセット信号に基づいて、IAT (Illegal Address Trap)やWDT (watchdog timer)のリセットを行う。演算回路107は、乗算回路及び除算回路を含んで構成される。例えば、ソースプログラム上において、後述する「MUL (乗算)」命令(後述の図93B参照)を実行するときには、演算回路107がこの「MUL」命令に基づく乗算処理を実行する。

40

【0135】

乱数回路110は、予め定められた範囲の乱数(例えば、0~65535又は0~255)を発生させる。また、図示しないが、乱数回路110は、2バイトのハードラッチ乱数を得るための乱数レジスタ0と、2バイトのソフトラッチ乱数を得るための乱数レジスタ1~3と、1バイトのソフトラッチ乱数を得るための乱数レジスタ4~7とで構成されている。なお、メインCPU 101は、乱数回路110で発生させた所定範囲の乱数の中から1つの値を、例えば内部抽籤用の乱数値として抽出する。パラレルポート111は、マ

50

マイクロプロセッサ 9 1 と、マイクロプロセッサ 9 1 の外部に設けられた各種回路（例えば、電源管理回路 9 3 等）との間で入出力される信号のポート（メモリーマップ I / O）である。また、パラレルポート 1 1 1 は、乱数回路 1 1 0 及び割込みコントローラ 1 1 2 にも接続される。スタートスイッチ 7 9 はパラレルポート 1 1 1 の P I 0 ~ P I 4 のいずれかの入力ポートに接続され、スタートスイッチ 7 9 がオン状態になったタイミング（オンエッジ）で、パラレルポート 1 1 1 から乱数回路 1 1 0 の乱数レジスタ 0 ヘラッチ信号が出力される。そして、乱数回路 1 1 0 では、ラッチ信号が入力されることにより乱数レジスタ 0 がラッチされ、2 バイトのハードラッチ乱数が取得される。

【0136】

割込みコントローラ 1 1 2 は、パラレルポート 1 1 1 を介して電源管理回路 9 3 から入力される電断検知信号、又は、タイマー回路 1 1 3 から 1 . 1 1 7 2 m s 周期で入力されるタイムアウト信号に基づいて、メイン C P U 1 0 1 による割込処理の実行タイミングを制御する。電源管理回路 9 3 から電断検知信号が入力された場合、又は、タイマー回路 1 1 3 からタイムアウト信号が入力された場合には、割込みコントローラ 1 1 2 は、割込処理開始指令を示す割込要求信号をメイン C P U 1 0 1 に出力する。メイン C P U 1 0 1 は、タイマー回路 1 1 3 からのタイムアウト信号に応じて割込みコントローラ 1 1 2 から入力される割込要求信号に基づいて、入力ポートチェック処理、リール制御処理、通信データ送信処理、7 セグ L E D 駆動処理、タイマー更新処理等の各種割込処理（後述の図 1 5 8 参照）を行う。

【0137】

タイマー回路 1 1 3（P T C）は、クロック回路 1 0 5 で生成されたクロックパルス信号（メイン C P U 作動用のクロックパルス信号を分周器（不図示）で分周された周波数のクロックパルス信号）で動作する（経過時間をカウントする）。そして、タイマー回路 1 1 3 は、1 . 1 1 7 2 m s e c の周期で割込みコントローラ 1 1 2 にタイムアウト信号（トリガー信号）を出力する。

【0138】

第 1 シリアル通信回路 1 1 4 は、主制御基板 7 1 から副制御基板 7 2 にデータ（各種制御指令（コマンド））を送信する際のシリアル送信動作を制御する回路である。第 2 シリアル通信回路 1 1 5 は、主制御基板 7 1 から試験機用第 2 インターフェースボード 3 0 2 にデータを送信する際のシリアル送信動作を制御する回路である。

【0139】

<副制御回路>

次に、図 1 0 を参照して、副制御基板 7 2 に実装される副制御回路 2 0 0（副制御手段）の構成について説明する。図 1 0 は、パチスロ 1 の副制御回路 2 0 0 の構成例を示すブロック図である。

【0140】

副制御回路 2 0 0 は、主制御回路 9 0 と電氣的に接続されており、主制御回路 9 0 から送信されるコマンドに基づいて演出内容の決定や実行等の処理を行う。副制御回路 2 0 0 は、基本的に、サブ C P U 2 0 1、サブ R A M 2 0 2、レンダリングプロセッサ 2 0 3、描画用 R A M 2 0 4、ドライバ 2 0 5 を含んで構成される。

【0141】

なお、サブ C P U 2 0 1 は、ロムカートリッジ基板 8 6 に接続される。ドライバ 2 0 5 は、液晶中継基板 8 7 に接続される。すなわち、ドライバ 2 0 5 は、液晶中継基板 8 7 を介してプロジェクタ機構 2 1 1 及びサブ表示装置 1 8 に接続される。

【0142】

サブ C P U 2 0 1 は、主制御回路 9 0 から送信されたコマンドに応じて、ロムカートリッジ基板 8 6 に記憶されている制御プログラムに従い、映像、音、光の出力の制御を行う。ロムカートリッジ基板 8 6 は、基本的に、プログラム記憶領域とデータ記憶領域とによって構成される。

【0143】

プログラム記憶領域には、サブCPU201が実行する制御プログラムが記憶される。例えば、制御プログラムには、主制御回路90との通信を制御するための主基板通信タスクや、演出用の乱数値を抽出し、演出内容（演出データ）の決定及び登録を行うための演出登録タスクを実行するための各種プログラムが含まれる。また、制御プログラムには、決定した演出内容に基づいて表示装置11による映像の表示を制御する描画制御タスク、LED群85等の光源による光の出力を制御するランプ制御タスク、スピーカ群84による音の出力を制御する音声制御タスク等を実行するための各種プログラムも含まれる。

【0144】

データ記憶領域には、各種データテーブルを記憶する記憶領域、各演出内容を構成する演出データを記憶する記憶領域、映像の作成に関するアニメーションデータを記憶する記憶領域が含まれる。また、データ記憶領域には、BGMや効果音に関するサウンドデータを記憶する記憶領域、光の点消灯のパターンに関するランプデータを記憶する記憶領域等も含まれる。

10

【0145】

サブRAM202には、決定された演出内容や演出データを登録する格納領域や、主制御回路90から送信されるサブフラグ（内部当籤役）等の各種データを格納する格納領域が設けられる。

【0146】

サブCPU201、レンダリングプロセッサ203、描画用RAM（フレームバッファを含む）204及びドライバ205は、演出内容により指定されたアニメーションデータに従って映像を作成し、作成した映像を表示装置11（プロジェクタ機構211）及び／又はサブ表示装置18に表示させる。なお、表示装置11（プロジェクタ機構211）及びサブ表示装置18は、副制御基板72により、それぞれ個別に制御される。

20

【0147】

また、サブCPU201は、演出内容により指定されたサウンドデータに従ってBGMなどの音をスピーカ群84により出力させる。また、サブCPU201は、演出内容により指定されたランプデータに従ってLED群85の点灯及び消灯を制御する。

【0148】

<メインCPUが有する各種レジスタ>

次に、図11を参照しながら、メインCPU101が有する各種レジスタについて説明する。なお、図11は、メインCPU101に含まれる各種レジスタの概略構成図である。

30

【0149】

メインCPU101は、メイン・レジスタとして、アキュムレータA（以下、「Aレジスタ」という）、フラグ・レジスタF（フラグレジスタ）、汎用レジスタB（以下、「Bレジスタ」という）、汎用レジスタC（以下、「Cレジスタ」という）、汎用レジスタD（以下、「Dレジスタ」という）、汎用レジスタE（以下、「Eレジスタ」という）、汎用レジスタH（以下、「Hレジスタ」という）及び汎用レジスタL（以下、「Lレジスタ」という）を有する。また、メインCPU101は、サブ・レジスタとして、アキュムレータA'、フラグ・レジスタF'、汎用レジスタB'、汎用レジスタC'、汎用レジスタD'、汎用レジスタE'、汎用レジスタH'及び汎用レジスタL'を汎用レジスタとして有する。なお、各レジスタは、1バイトのレジスタで構成される。

40

【0150】

また、本実施形態では、BレジスタとCレジスタとをペアレジスタ（以下、「BCレジスタ」という）として用い、DレジスタとEレジスタとをペアレジスタ（以下、「DEレジスタ」という）として用いる。さらに、本実施形態では、HレジスタとLレジスタとをペアレジスタ（以下、「HLレジスタ」という）として用いる。

【0151】

フラグ・レジスタF、F'の各ビットには、図11に示すように、演算処理の結果等を示す所定のフラグ情報がセットされる。例えばビット6（D6）には、演算結果の判定処理において演算結果が「0」であるか否かを示すデータ（ゼロフラグ）がセットされる。具

50

体的には、演算結果が「0」である場合、ビット6にデータ「1」がセットされ、演算結果が「0」でない場合には、ビット6にデータ「0」がセットされる。そして、演算結果の判定処理では、メインCPU101は、ビット6のデータ「0」／「1」を参照して判定（YES／NO）を行う。

【0152】

また、メインCPU101は、拡張レジスタQ（以下、「Qレジスタ」という）を有する。Qレジスタは、1バイトのレジスタで構成される。なお、本実施形態では、後述の各種処理フローの中で説明するように、ソースプログラム上において、このQレジスタを用いてアドレス指定を行う各種メインCPU101専用命令コードが設けられており、この命令コードの使用により、処理の効率化やメインROM102の容量削減などを実現している。なお、Qレジスタを用いてアドレス指定を行う各種メインCPU101専用命令コードでは、Qレジスタには、アドレスの上位側のアドレスデータ（アドレス値）が格納される。なお、Qレジスタには、メインCPU101のリセット直後に、初期値として「F0H」がセットされる。また、Qレジスタを用いた「LD Q, n（8ビットデータ）」命令において、「n」に任意の1バイトのデータをセットして該命令を実行することにより、Qレジスタの値を変更することができる。

【0153】

さらに、メインCPU101は、1バイトのレジスタで構成された、インタラプト・ページアドレス・レジスタI及びメモリ・リフレッシュ・レジスタR、並びに、2バイトのレジスタで構成された、インデックス・レジスタIX、インデックス・レジスタIY、スタックポインタSP及びプログラムカウンタPCを専用レジスタとして有する。

【0154】

<メインROM及びメインRAMの内部構成（メモリマップ）>

次に、図12A～図12Cを参照しながら、主制御回路90（マイクロプロセッサ91）に含まれるメインROM102及びメインRAM103の内部構成（以下「メモリマップ」という）について説明する。なお、図12Aは、メモリ全体のメモリマップを示す図であり、図12Bは、メインROM102のメモリマップを示す図であり、図12Cは、メインRAM103のメモリマップを示す図である。

【0155】

主制御回路90（マイクロプロセッサ91）が備えるメモリ全体のメモリマップでは、図12Aに示すように、アドレスの先頭（0000H）側から、メインROM102のメモリ領域、メインRAM103のメモリ領域、内蔵レジスタエリア及びXCSデコードエリアが、不使用領域を間に挟んでこの順で、それぞれ所定のアドレスに配置される。

【0156】

メインROM102のメモリマップでは、図12Bに示すように、メインROM102のアドレスの先頭（0000H）側から、プログラムエリア、データエリア、規定外エリア、商標記録エリア、プログラム管理エリア及びセキュリティ設定エリアが、この順で、それぞれ所定のアドレスに配置される。

【0157】

なお、プログラムエリアには、遊技者により実施される遊技の遊技性に関連する各種制御処理において、メインCPU101により実行される各種処理の制御プログラムが記憶される。データエリアには、遊技者により実施される遊技の遊技性に関連する各種制御処理において、メインCPU101により使用される各種データ（例えば、内部抽籤テーブル等のデータテーブル、副制御回路42に対して各種制御指令（コマンド）を送信するためのデータ等）が記憶される。すなわち、プログラムエリアとデータエリアとからなる遊技用ROM領域（遊技用記憶領域）には、遊技店で遊技者が実際に行う遊技の遊技性に関連する制御処理（遊技性に関する処理）に必要な各種プログラム及び各種データが格納される。

【0158】

また、規定外エリアには、遊技者により実施される遊技の遊技性に直接関与しない各種処

10

20

30

40

50

理（遊技性に影響を与えない処理）の制御プログラム及びデータが記憶される。例えば、パチスロ1の検定試験（試射試験）で使用されるプログラム及びデータ、電断時のチェックサム生成処理や電源復帰時のサムチェック処理などで使用される制御プログラム及びデータ、並びに、不正対策プログラム及びそれに必要なデータ等が、規定外エリアに格納される。

【0159】

メインRAM103のメモリマップでは、図12Cに示すように、メインRAM103のアドレスの先頭（F000H）側から、遊技用RAM領域（所定格納領域、遊技用一時記憶領域）及び規定外RAM領域（規定外一時記憶領域）が、この順で、それぞれ所定のアドレスに配置される。

【0160】

遊技用RAM領域には、遊技者により実施される遊技の遊技性に関連する制御プログラムの実行により決定された例えば内部当籤役等の各種データを一時的に格納する作業領域及びスタックエリアが設けられる。そして、各種データのそれぞれは、遊技用RAM領域内の所定アドレスの作業領域に格納される。

【0161】

また、規定外RAM領域には、遊技者により実施される遊技の遊技性に直接関与しない各種処理の作業領域となる規定外作業領域と、規定外スタックとが設けられる。本実施形態では、この規定外RAM領域を使用して、例えばサムチェック処理等の遊技者により実施される遊技の遊技性に直接関与しない各種処理が実行される。

【0162】

上述のように、本実施形態のパチスロ1では、メインROM102内において、遊技者により実施される遊技の遊技性に直接関与しない各種処理に使用される各種プログラム及び各種データ（テーブル）を、遊技用ROM領域とは異なるアドレスに配置された規定外ROM領域（規定外記憶領域）に格納する。また、そのような遊技者により実施される遊技の遊技性に直接関与しない各種処理は、メインRAM103内において、遊技用RAM領域とは異なるアドレスに配置された規定外RAM領域を使用して行われる。

【0163】

このようなメインROM102の構成では、従来の規則上においてプログラム等の配置不可とされていたROM領域（規定外ROM領域）に、遊技者が実際に行う遊技そのものには不要なプログラム及びデータを配置することができる。それゆえ、本実施形態では、遊技用ROM領域の容量の圧迫を回避することができる。

【0164】

<遊技状態の遷移フロー>

次に、図13及び図14を参照しながら、本実施形態のパチスロ1の主制御回路90（メインCPU101）により管理される各種遊技状態及びその遷移フローについて説明する。なお、図13Aは、パチスロ1の基本的な遊技状態の遷移フロー図であり、図13Bは、その遊技状態の移行条件をまとめた表である。また、図14Aは、報知（ART）機能の作動の有無を考慮した遊技状態の遷移フロー図であり、図14Bは、その遊技状態の移行条件をまとめた表である。

【0165】

〔基本的な遊技状態の遷移フロー〕

本実施形態のパチスロ1では、ボーナスゲームの種類として、ビッグボーナス（以下、「BB」と記す）が設けられる。BBは、第1種特別役物と呼ばれるレギュラーボーナス（以下、「RB」と記す）に係る役物連続作動装置であり、RBを連続して作動させる。

【0166】

それゆえ、本実施形態では、主制御回路90は、ボーナス役の当籤／作動（入賞）の有無に基づいて遊技状態を管理する。具体的には、図13Aに示すように、主制御回路90は、ボーナス役（後述の名称「F__BB1」、「F__BB2」の内部当籤役）の当籤／作動（入賞）の有無に基づいて、「ボーナス非当籤状態」、「フラグ間状態」及び「ボーナス

10

20

30

40

50

状態」と称する3種類の遊技状態を管理する。

【0167】

なお、ボーナス非当籤状態は、ボーナスに非当籤であり、かつ、ボーナスが作動（入賞）していない状態であり、ボーナス状態は、ボーナスが作動している状態である。また、本実施形態では、ボーナス役が内部当籤役として決定されると、ボーナスが入賞するまで複数回の遊技に渡りボーナス役が内部当籤役として持ち越された状態が発生する。フラグ間状態は、ボーナス役が内部当籤役として持ち越されている状態、すなわち、ボーナス役が当籤し、かつ、ボーナスが作動していない状態である。

【0168】

なお、ボーナス役の当籤の有無は、メインRAM103に設けられる後述の当り要求フラグ格納領域（後述の図28～図30参照）及び持越役格納領域（後述の図31参照）に格納されるデータに基づいて管理される。また、ボーナスの作動（入賞）の有無は、メインRAM103に設けられる後述の遊技状態フラグ格納領域（後述の図32参照）に格納されるデータに基づいて管理される。

【0169】

また、本実施形態では、図13Aに示すように、ボーナスが作動していない遊技状態（ボーナス非当籤状態及びフラグ間状態）において、リプレイに係る内部当籤役の種別及びその当籤確率が互いに異なる、RT0遊技状態～RT5遊技状態の6種類の状態（以下、それぞれ「RT0状態」～「RT5状態」という）が設けられる。なお、RT0状態、RT2状態及びRT5状態は、リプレイ役が内部当籤役として決定される確率が低確率となる遊技状態であり、RT1状態はリプレイ役が内部当籤役として決定される確率が中程度の中確率となる遊技状態である。また、RT3状態及びRT4状態は、リプレイ役が内部当籤役として決定される確率が高確率となる遊技状態である。なお、本実施形態では、ボーナス非当籤状態のRT状態は、RT0状態～RT4状態のいずれかとなり、フラグ間状態のRT状態はRT5状態となる。

【0170】

それゆえ、本実施形態では、主制御回路90は、ボーナスが作動していない遊技状態（ボーナス非当籤状態及びフラグ間状態）において、さらに、リプレイに係る内部当籤役の種別及びその当籤確率に基づいて、RT1状態～RT5状態の6種類の状態も管理する。

【0171】

なお、RT0状態～RT5状態は、メインRAM103に設けられる後述の遊技状態フラグ格納領域（後述の図32参照）に格納されるデータに基づいて管理される。具体的には、本実施形態のパチスロ1では、RT1状態フラグ～RT5状態フラグの5つのRT状態を示すフラグが設けられ、これらのフラグのオン／オフ状態をメインRAM103により管理することによりRT状態が管理される。そして、主制御回路90は、オン状態であるRT状態フラグに対応するRT状態を現在のRT状態として特定する。なお、全てのRT状態フラグがオフ状態である場合には、主制御回路90は、現在のRT状態がRT0状態であると特定する。

【0172】

図13A及び13Bに示すように、ボーナス非当籤状態においてボーナス役（後述の名称「F_BB1」，「F_BB2」の内部当籤役）が内部当籤役として決定されると（移行条件（1）が成立すると）、主制御回路90は、遊技状態をボーナス非当籤状態からフラグ間状態に移行させる。また、フラグ間状態においてボーナス役が入賞すると（移行条件（2）が成立すると）、主制御回路90は、遊技状態をフラグ間状態からボーナス状態に移行させる。

【0173】

また、ボーナス状態において規定枚数（216枚）を超えるメダルが払い出され、ボーナス状態が終了すると（移行条件（3）が成立すると）、主制御回路90は、遊技状態をボーナス状態からRT1状態（ボーナス非当籤状態）に移行させる。

【0174】

R T 1 状態において、20ゲームが経過すると（移行条件（4）が成立すると）、主制御回路90は、遊技状態をR T 1 状態からR T 0 状態に移行させる。また、R T 1 状態において、20ゲームが経過する前に、略称「ベルこぼし目」に係る図柄組合せ（後述の図28参照）が有効ライン上に表示されると（移行条件（5）が成立すると）、主制御回路90は、遊技状態をR T 1 状態からR T 2 状態に移行させる。

【0175】

R T 0 状態において、略称「ベルこぼし目」に係る図柄組合せが有効ライン上に表示されると（移行条件（5）が成立すると）、主制御回路90は、遊技状態をR T 0 状態からR T 2 状態に移行させる。R T 2 状態において、略称「R T 3 移行リップ」に係る図柄組合せ（後述の図28参照）が有効ライン上に表示されると（移行条件（6）が成立すると）、主制御回路90は、遊技状態をR T 2 状態からR T 3 状態に移行させる。

10

【0176】

R T 3 状態において、略称「R T 4 移行リップ」に係る図柄組合せ（後述の図28参照）が有効ライン上に表示されると（移行条件（7）が成立すると）、主制御回路90は、遊技状態をR T 3 状態からR T 4 状態に移行させる。また、R T 3 状態において、略称「ベルこぼし目」又は「R T 2 移行リップ」に係る図柄組合せ（後述の図28参照）が有効ライン上に表示されると（移行条件（8）が成立すると）、主制御回路90は、遊技状態をR T 3 状態からR T 2 状態に遊技状態を移行させる。さらに、R T 4 状態において、略称「ベルこぼし目」又は「R T 2 移行リップ」に係る図柄組合せが有効ライン上に表示されると（移行条件（8）が成立すると）、主制御回路90は、遊技状態をR T 4 状態からR T 2 状態に遊技状態を移行させる。

20

【0177】

なお、略称「ベルこぼし目」に係る図柄組合せは、後述の名称「F__3 択ベル__1 s t」、「F__3 択ベル__2 n d」又は「F__3 択ベル__3 r d」に係る内部当籤役（小役）が決定され、かつ、停止操作の順序が該小役の種別ごとに定められた押し順に対して不正解であるときに表示される図柄の組合せである（後述の図24参照）。略称「R T 2 移行リップ」に係る図柄組合せは、後述の名称「F__維持リップ__1 s t」、「F__維持リップ__2 n d」又は「F__維持リップ__3 r d」に係る内部当籤役（リプレイ役）が決定され、かつ、停止操作の順序が該リプレイ役の種別ごとに定められた押し順に対して不正解であるときに表示される図柄の組合せである。

30

【0178】

略称「R T 3 移行リップ」に係る図柄組合せは、後述の名称「F__R T 3 リップ__1 s t」、「F__R T 3 リップ__2 1 3」、「F__R T 3 リップ__2 3 1」又は「F__R T 3 リップ__3 r d」に係る内部当籤役（リプレイ役）が決定され、かつ、停止操作の順序が該リプレイ役の種別ごとに定められた押し順に対して正解であるときに表示される図柄の組合せである。また、略称「R T 4 移行リップ」に係る図柄組合せは、後述の名称「F__R T 4 リップ__1 2 3」、「F__R T 4 リップ__1 3 2」、「F__R T 4 リップ__2 n d」又は「F__R T 4 リップ__3 r d」に係る内部当籤役（リプレイ役）が決定され、かつ、停止操作の順序が該リプレイ役の種別ごとに定められた押し順に対して正解であるときに表示される図柄の組合せである。

40

【0179】

〔報知（A R T）機能の作動の有無を考慮した遊技状態の遷移フロー〕

本実施形態では、主制御回路90（メインCPU101）により、遊技者にとって有利な停止操作を報知する機能（A R T 機能）の作動の有無が決定される。それゆえ、本実施形態では、ボーナス非作動状態においてA R T 機能の作動／非作動状態も遊技状態として管理する。

【0180】

本実施形態のパチスロ1では、図14Aに示すように、主制御回路90は、非ボーナス作動状態において、報知（A R T）の有無に基づいて「一般遊技状態」と「A R T 遊技状態」とを別個の遊技状態として管理する。すなわち、報知（A R T）の有無を考慮した遊技

50

状態の管理では、図14Aに示すように、主制御回路90は、大きな分類として、「ボーナス状態」、「一般遊技状態」及び「ART遊技状態」の3種類の遊技状態を管理する。

【0181】

なお、一般遊技状態は、基本的には、遊技者にとって有利な停止操作の情報を報知しない遊技状態（非ART）であり、遊技者にとって不利な遊技状態である。また、一般遊技状態は、RT0～RT4状態のいずれかの状態であり、かつ、ART非当籤の遊技状態である。

【0182】

一方、ART遊技状態は、遊技者にとって有利な停止操作の情報を報知する遊技状態であり、遊技者にとって有利な遊技状態である。また、ART遊技状態は、基本的には、RT4状態であり、かつ、ART当籤中の遊技状態である。なお、本実施形態では、ART当籤後、RT状態がRT4状態まで移行すると、ART遊技が開始される。

【0183】

また、本実施形態では、図14Aに示すように、一般遊技状態として、「通常遊技状態」及び「CZ（チャンスゾーン）」と称する2種類の状態が設けられる。

【0184】

通常遊技状態は、遊技者にとって最も不利な遊技状態であるが、通常遊技状態の遊技ではCZへの移行抽籤を行っている。そして、図14A及び14Bに示すように、通常遊技状態の遊技において、CZへの移行抽籤に当籤すると（移行条件（A）が成立すると）、主制御回路90は、遊技状態を、通常遊技状態からCZに移行させる。

【0185】

CZは、ART遊技状態への移行に対する期待度が高い遊技状態（チャンスゾーン）であり、CZ中の遊技ではARTへの移行抽籤が行われている。そして、図14A及び14Bに示すように、CZ中の遊技において、ARTへの移行抽籤に非当籤である場合には（移行条件（B）が成立すると）、主制御回路90は、遊技状態を、CZから通常遊技状態に移行させる。一方、CZ中の遊技において、ARTへの移行抽籤に当籤すると（移行条件（C）が成立すると）、主制御回路90は、遊技状態を、CZからART遊技状態に移行させる。この際、図14Aには示さないが、主制御回路90は、遊技状態を、CZから後述のART準備状態を経由してART遊技状態（後述の通常ART又はCT）に移行させる。

【0186】

ART遊技状態は、上述のように、ART当籤後にRT状態がRT4状態まで移行すると開始される。なお、図13Aで示したように、RT4状態は、RT0～RT2状態からRT3状態を経由して移行するので、ART当籤後であってもすぐにART遊技状態が開始されない。そこで、本実施形態のパチスロ1では、ART当籤後からRT状態がRT4状態に移行するまでの期間の遊技状態をART準備状態とする。そして、このART準備状態の遊技では、RT状態をRT4状態に移行させるために必要な停止操作の情報が報知される。

【0187】

また、本実施形態では、図14Aに示すように、ART遊技状態として、遊技性が互いに異なる、「通常ART」及び「CT（上乗せチャンス）」と称する2種類の状態が設けられる。

【0188】

通常ARTは、所定ゲーム数の期間、遊技者にとって有利な停止操作（例えば、払い出されるメダルの枚数が多い図柄組合せを表示させるための停止操作や、RT4状態を維持するために必要な停止操作）が報知される遊技状態である。また、通常ART中の遊技では、CTへの移行抽籤が行われる。

【0189】

CTは、遊技者にとって有利な停止操作が報知されるとともに、特定期間（1セット8ゲームの期間）、通常ARTの継続期間を上乗せすることが可能となる遊技状態であり、上

10

20

30

40

50

乗せチャンスゾーンとして機能する遊技状態である。また、C T中では、通常A R Tの継続期間を消化せずに遊技が行われる。なお、C T中の遊技性については、後述の図5 2 A～5 2 Cを参照して後で詳述する。

【0190】

図1 4 A及び1 4 Bに示すように、通常A R T中の遊技において、C Tへの移行抽籤に当籤すると（移行条件（D）が成立すると）、主制御回路9 0は、遊技状態を、通常A R TからC Tに遊技状態を移行させる。また、通常A R Tにおいて、当該通常A R Tの継続期間が終了すると（移行条件（E）が成立すると）、主制御回路9 0は、遊技状態を、通常A R Tから一般遊技状態（通常遊技状態又はC Z）に移行させる。なお、本実施形態では、ゲーム数により通常A R Tの継続期間を管理するが、本発明はこれに限定されず、通常A R Tの継続期間の管理方法は任意である。例えば、通常A R Tの継続期間を、通常A R T中に払い出されるメダルの枚数や差枚数により管理してもよいし、通常A R T中にメダルの払い出しに影響を与える報知を行った回数（ナビ回数）により管理してもよい。

10

【0191】

図1 4 A及び1 4 Bに示すように、C T中の遊技において、C Tの継続期間（1セット8ゲーム）が終了すると（移行条件（F）が成立すると）、主制御回路9 0は、遊技状態を、C Tから通常A R Tに移行させる。

【0192】

また、図1 4 Aに示すように、一般遊技状態（通常遊技状態又はC Z）又はA R T遊技状態（通常A R T又はC T）において、ボーナス役が入賞すると（図1 3 A及び1 3 B中で説明した移行条件（2）が成立すると）、主制御回路9 0は、遊技状態を、一般遊技状態又はA R T遊技状態からボーナス状態に移行させる。

20

【0193】

ボーナス状態の遊技では、上述のように、A R Tへの移行抽籤を行っており、ボーナス状態の遊技において、A R Tへの移行抽籤が非当籤である場合には（移行条件（G）が成立すると）、主制御回路9 0は、遊技状態を、ボーナス状態から一般遊技状態（通常遊技状態又はC Z）に移行させる。ただし、A R T遊技状態（通常A R T又はC T）からボーナス状態に移行していた場合には、ボーナス状態の遊技においてA R Tの移行抽籤に非当籤であっても、主制御回路9 0は、遊技状態を、ボーナス状態からA R T遊技状態（通常A R T又はC T）に移行させる。一方、ボーナス状態の遊技において、A R Tへの移行抽籤に当籤すると（移行条件（H）が成立すると）、主制御回路9 0は、遊技状態を、ボーナス状態からA R T遊技状態（通常A R T又はC T）に移行させる。なお、上述のように、ボーナス状態の終了時には、R T状態がR T 1状態に移行するので、ボーナス状態からA R T遊技状態に遊技状態を移行する場合には、主制御回路9 0は、遊技状態を、A R T準備状態を経由してA R T遊技状態に移行させる。

30

【0194】

<メインROMに記憶されているデータテーブルの構成>

次に、図1 5～図2 7を参照して、メインROM 1 0 2に記憶されている各種データテーブルの構成について説明する。なお、一般遊技状態中及びA R T遊技状態中の遊技性（C Z、通常A R T、C Tの遊技性）に係る各種抽籤で用いられる各種データテーブルについては、別途、各遊技性の説明と一緒に後述する。

40

【0195】

〔図柄配置テーブル〕

まず、図1 5を参照して、図柄配置テーブルについて説明する。図柄配置テーブルは、左リール3 L、中リール3 C及び右リール3 Rのそれぞれの回転方向における各図柄の位置と、各位置に配置された図柄の種類を特定するデータ（以下、図柄コード（図1 5中の図柄コード表を参照）という）との対応関係を規定する。

【0196】

図柄配置テーブルでは、リールインデックスが検出されたときに、リール表示窓4の枠内における各リールの中段領域に位置する図柄の位置を「0」と規定する。そして、各リー

50

ルにおいて、図柄位置「0」を基準としてリールの回転方向（図15中の図柄位置「19」から図柄位置「0」に向かう方向）に進む順に、図柄カウンタの値に対応する「0」～「19」が、図柄位置として、各図柄に割り当てられる。

【0197】

すなわち、図柄カウンタの値（「0」～「19」）と、図柄配置テーブルとを参照することにより、リール表示窓4の枠内における各リールの上段領域、中段領域及び下段領域に表示されている図柄の種類を特定することができる。なお、本実施形態では、図柄として、「白7」、「青7」、「チリ上1」、「チリ上2」、「チリ下」、「リプレイ」、「帽子」、「サボテン1」、「サボテン2」及び「サボテン3」の10種類の図柄を用いる。

【0198】

また、本実施形態では、図柄コード表に示すように、図柄「白7」（図柄コード1）には、データとして「00000001」が割り当てられ、図柄「青7」（図柄コード2）には、データとして「00000010」が割り当てられている。図柄「チリ上1」（図柄コード3）には、データとして「00000011」が割り当てられ、図柄「チリ上2」（図柄コード4）には、データとして「00000100」が割り当てられている。

【0199】

図柄「チリ下」（図柄コード5）には、データとして「00000101」が割り当てられ、図柄「リプレイ」（図柄コード6）には、データとして「00000110」が割り当てられている。図柄「帽子」（図柄コード7）には、データとして「00000111」が割り当てられ、図柄「サボテン1」（図柄コード8）には、データとして「00001000」が割り当てられている。また、図柄「サボテン2」（図柄コード9）には、データとして「00001001」が割り当てられ、図柄「サボテン3」（図柄コード10）には、データとして「00001010」が割り当てられている。

【0200】

〔内部抽籤テーブル〕

次に、図16及び図17を参照して、内部当籤役を決定する際に参照される内部抽籤テーブルについて説明する。なお、図16は、RT0状態～RT4状態のそれぞれにおいて参照される内部抽籤テーブルである。また、図17Aは、RT5状態において参照される内部抽籤テーブルであり、図17Bは、ボーナス状態において参照される内部抽籤テーブルである。

【0201】

内部抽籤テーブルは、遊技状態毎に設けられ、各種内部当籤役と、各内部当籤役が決定されときの抽籤値との対応関係を規定する。なお、抽籤値は、予め設定されたボーナス役や小役等の内部当籤の期待値を調整するための設定（設定1～6）毎に規定される。この設定は、例えば、リセットスイッチ76及び設定用鍵型スイッチ54（図7参照）を用いて変更される。

【0202】

本実施形態の内部抽籤処理では、まず、乱数回路110の乱数レジスタ0により、予め定められた数値の範囲（例えば、0～65535）から抽出される乱数値を、各内部当籤役に対応して規定された抽籤値で順次加算する。次いで、抽籤結果（抽籤値+乱数値）が65535を超えたか否か（抽籤結果がオーバーフローしたか否か）の判定を行う。そして、所定の内部当籤役において、抽籤結果が65535を超えた場合、該内部当籤役が当籤したと判定される。なお、本実施形態の内部抽籤処理では、抽出した乱数値に抽籤値を加算して抽籤を行う例を説明したが、本発明はこれに限定されず、乱数値から抽籤値を減算して、減算結果（抽籤結果）が「0」を下回ったか否か（抽籤結果がアンダーフローしたか否か）を判定して、内部抽籤の当籤／非当籤を決定してもよい。

【0203】

それゆえ、本実施形態の内部抽籤処理では、抽籤値として規定されている数値が大きい内部当籤役ほど、決定される確率が高い。なお、各内部当籤役の当籤確率は、「各当籤番号に規定された抽籤値／抽出される可能性のある全ての乱数値の個数（乱数分母：6553

10

20

30

40

50

6) 」によって表すことができる。

【0204】

RT0状態～RT4状態のそれぞれにおいて参照される内部抽籤テーブルでは、図16に示すように、基本的には、RT状態の種別に応じて、内部当籤役として決定されるリプレイ役の種別及び当籤確率に変化する。例えば、名称「F__チリリブ(No.25)」～「F__リーチ目リブD(No.31)」に係るリプレイ役は、RT0状態～RT3状態では内部当籤役として決定されることなく、RT4状態でのみ内部当籤役として決定される。なお、本実施形態のパチスロ1では、RT4状態中に、名称「F__チリリブ(No.25)」～「F__リーチ目リブD(No.31)」に係るリプレイ役が内部当籤役として決定された場合、特有の制御(後述のフラグ変換)を行う。このフラグ変換については、後で詳述する。

10

【0205】

また、図16に示すように、RT0状態～RT3状態では、名称「F__リーチ目リブA」～「F__リーチ目リブD」のそれぞれの内部当籤役は、名称「F__BB1」又は「F__BB2」に係るボーナス役と重複して決定されることはあるが(No.3～6、15～18参照)、名称「F__リーチ目リブA」～「F__リーチ目リブD」のそれぞれの内部当籤役(リプレイ役)が単独で内部当籤役として決定されることはない。それゆえ、本実施形態において、RT0状態～RT3状態中に名称「F__リーチ目リブA」～「F__リーチ目リブD」に係るリプレイ役が内部当籤役として決定された場合(遊技者からすると名称「F__リーチ目リブA」～「F__リーチ目リブD」に係るリプレイ役に応じた図柄組合せが表示された場合)、ボーナス役(名称「F__BB1」又は「F__BB2」)が同時に内部当籤役として決定されていることになる。

20

【0206】

また、フラグ間状態であるRT5状態は、上述のようにボーナス役を内部当籤役として持ち越す遊技状態である。それゆえ、図17Aに示すように、RT5状態において参照される内部抽籤テーブルでは、持ち越しているボーナス役が必ず内部当籤役として決定されるようになっている。また、図17Bに示すように、ボーナス状態において参照される内部抽籤テーブルでは、名称「F__RB役1」～「F__RB役4」のいずれかに係る内部当籤役が必ず当籤する構成になっている(「はずれ」が当籤することはない)。

【0207】

【内部当籤役と図柄組合せ(入賞役)との対応表(図柄組合せ決定テーブル)】

次に、図18～図23を参照して、内部当籤役と図柄組合せとの対応表(図柄組合せ決定テーブル)について説明する。図柄組合せ決定テーブルは、各種内部当籤役と、各内部当籤役に対応付けられた、有効ライン(センターライン)上に表示可能な図柄組合せ(コンビネーション)との対応関係を規定する。すなわち、内部当籤役が決定されると、有効ライン上に表示可能な図柄組合せの種別(入賞可能な表示役の種別)が一義的に決定される。

30

【0208】

各図柄組合せ決定テーブル中の図柄組合せ欄に記載の各種データは、左リール3L、中リール3C及び右リール3Rに渡って設定された有効ラインに沿って表示を許可する図柄組合せを識別するためのデータである。なお、図柄組合せ(表示役)欄に記載の各名称と、具体的な図柄組合せとの関係は、後述の図28～図30の入賞作動フラグ格納領域に示す。

40

【0209】

また、図柄組合せ決定テーブル中に記載の「○」印は、決定された内部当籤役において、有効ライン上に表示可能な図柄組合せ(コンビネーション)、すなわち、入賞可能となる表示役を示す。例えば、内部当籤役「F__チリリブ」が決定された場合、図18及び図19に示すように、コンビネーション名称「C__維持リブA__01」～「C__維持リブG__01」、「C__チリリブA__01」～「C__チリリブD__01」に係る図柄組合せが停止表示可能となる。なお、図柄組合せ決定テーブルには、「内部当籤役」が「はずれ」となる場合が規定されていないが、これは、図18～図23に示した図柄組合せテーブルにより規定されている全ての図柄組合せの表示が許可されないことを示す。

50

【0210】

本実施形態のパチスロ1では、主制御回路90（メインCPU101）は、内部当籤役及び遊技状態に応じて停止制御を異ならせ、所定の役が内部当籤役として決定された場合に、図18～図23に示す対応関係の図柄組合せ（コンビネーション）を表示可能とするように左リール3L、中リール3C及び右リール3Rの回転停止制御を行う。なお、図18～図23に示す対応表では、決定された内部当籤役に対して表示可能な全ての図柄組合せを「○」印で列挙しているが、「○」印が付された図柄組合せであっても、表示されないことがある。

【0211】

本実施形態では、停止表示可能な図柄組合せや現在の遊技状態に応じて停止制御（例えば、優先して引き込む図柄）を異ならせる機能を有し、優先して引き込む図柄の関係上、「○」印が付された図柄組合せであっても表示されないことがある。内部当籤役の種別と実際に表示される図柄組合せとの対応関係については、後述の図24及び図25を参照して説明する。

10

【0212】

〔非フラグ間状態中の当籤役と停止表示される図柄組合せとの対応関係〕

ここで、図24を参照して、フラグ間状態を除く遊技状態（非フラグ間状態）における内部当籤役と停止表示される図柄組合せとの対応関係について説明する。なお、図24は、非フラグ間状態において決定され得る各種内部当籤役と、各内部当籤役決定時に停止表示される図柄組合せ（略称）との対応関係（一部の役については省略）を示す図である。なお、図24中に記載の図柄組合せの名称は、後述の図28～図30の入賞作動フラグ格納領域に示す内容欄に記載の「略称」である。

20

【0213】

本実施形態のパチスロ1では、遊技者の停止操作の順序（押し順）に応じて表示される図柄組合せが異なる役、いわゆる「押し順役」を設ける。なお、図24に記載の「押し順正解」に対応付けられた図柄組合せは、押し順に応じて表示される図柄組合せのうち、遊技者にとって有利な図柄組合せであり、「押し順不正解」に対応付けられた図柄組合せは、押し順に応じて表示される図柄組合せのうち、遊技者にとって不利な図柄組合せである。遊技者にとって有利な停止操作を報知する場合、正解となる押し順が報知され、その報知に従って停止操作が行われれば、「押し順正解」に対応付けられた図柄組合せが表示される。また、ART遊技状態であっても、不正解となる押し順が報知されることもあるが、その内容については、後で詳述する。

30

【0214】

なお、本実施形態では、押し順役の一部に対しては、その名称の末尾に、正解となる押し順を示す。具体的には、内部当籤役の名称の末尾「1st」は、正解となる押し順が、第1停止操作（1番目に行われる停止操作）が左リール3Lに対するものであることを意味し、内部当籤役の名称の末尾「2nd」は、正解となる押し順が、第1停止操作が中リール3Cに対するものであることを意味し、内部当籤役の名称の末尾「3rd」は、正解となる押し順が、第1停止操作が右リール3Rに対するものであることを意味する。また、内部当籤役の名称の末尾「123」は、正解となる押し順が「左、中、右」の順であることを意味し、内部当籤役の名称の末尾「132」は、正解となる押し順が「左、右、中」の順であることを意味し、内部当籤役の名称の末尾「213」は、正解となる押し順が「中、左、右」の順であることを意味し、内部当籤役の名称の末尾「231」は、正解となる押し順が「左、右、中」の順であることを意味する。

40

【0215】

また、以下では、第1停止操作が左リール3Lに対して行われた場合の停止操作順序、具体的には、「左、中、右」及び「左、右、中」の押し順を「順押し」ともいう。さらに、以下では、第1停止操作が中リール3C又は右リール3Rに対して行われた場合の停止操作順序、具体的には、「中、左、右」、「中、右、左」、「右、中、左」、及び、「右、左、中」の押し順を、「変則押し」ともいう。

50

【0216】

本実施形態では、図24に示すように、内部当籤役「F__チリリブ」は、押し順に応じて表示される図柄組合せが異なる押し順役であり、押し順が正解である場合には、略称「チリリブ」に係る図柄組合せ（後述の図28参照）のうちの図18～図23に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。一方、押し順が正解でない場合には、略称「リプレイ」に係る図柄組合せ（後述の図28参照）のうちの図18～図23に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。なお、内部当籤役「F__チリリブ」が決定された場合、図18～図23に示すように、コンビネーション名称「C__チリリブA__01」、「C__チリリブB__01」又は「C__チリリブC__01」（略称「単チリリブ」又は「2連チリリブ」：後述の図28中の略称「チリリブ（否3連）」に対応）に係る図柄組合せは表示できるが、コンビネーション名称「C__チリリブD__01」～「C__1確チリリブD__01」（略称「3連チリリブ」：後述の図28中の略称「チリリブ（3連）」に対応）に係る図柄組合せを表示できない。すなわち、内部当籤役「F__チリリブ」は、略称「3連チリリブ」に係る図柄組合せを表示できない役である。

10

【0217】

また、内部当籤役「F__確チリリブ」及び「F__1確チリリブ」はともに、押し順に応じて表示される図柄組合せが異なる押し順役であり、押し順が正解である場合には、略称「チリリブ」に係る図柄組合せ（後述の図28参照）のうちの図18～図23に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。一方、押し順が正解でない場合には、略称「リプレイ」に係る図柄組合せ（後述の図28参照）のうちの図18～図23に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。なお、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定された場合、図18～図23に示すように、略称「3連チリリブ」に係る図柄組合せを表示できる。すなわち、内部当籤役「F__確チリリブ」及び「F__1確チリリブ」は、略称「3連チリリブ」に係る図柄組合せを表示できる役である。

20

【0218】

また、内部当籤役「F__リーチ目リブA」～「F__リーチ目リブD」は、押し順に応じて表示される図柄組合せが異なる押し順役であり、押し順が正解である場合には、略称「リーチ目リブ」に係る図柄組合せ（後述の図28及び図29参照）のうちの図18～図23に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。一方、押し順が正解でない場合には、略称「リプレイ」に係る図柄組合せ（後述の図28参照）のうちの図18～図23に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

30

【0219】

なお、本実施形態では、内部当籤役「F__チリリブ」、「F__確チリリブ」、「F__1確チリリブ」及び「F__リーチ目リブA」～「F__リーチ目リブD」の当籤時における正解の押し順は、左リール3Lに対して第1停止操作を行うものである。それゆえ、例えば、内部当籤役「F__リーチ目リブA」が決定されている遊技において、遊技者が左リール3Lに対して第1停止操作を行った場合には、略称「リーチ目リブ」に係る図柄組合せが停止表示される。なお、本発明はこれに限定されず、内部当籤役「F__チリリブ」、「F__確チリリブ」、「F__1確チリリブ」及び「F__リーチ目リブA」～「F__リーチ目リブD」の当籤時における正解の押し順は、任意に設定することができる。

40

【0220】

また、内部当籤役「F__維持リブA」及び「F__維持リブB」はともに、押し順役ではなく、押し順に関わらず略称「リプレイ」に係る図柄組合せ（後述の図28参照）のうちの図18～図23に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

【0221】

また、内部当籤役「F__維持リブ__1st」～「F__維持リブ__3rd」はいずれも、押

50

し順に応じて表示される図柄組合せが異なる押し順役であり、押し順が正解である場合には、略称「リプレイ」に係る図柄組合せ（後述の図 2 8 参照）のうちの図 1 8～図 2 3 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。一方、押し順が正解でない場合には、略称「R T 2 移行リブ」に係る図柄組合せ（後述の図 2 8 参照）のうちの図 1 8～図 2 3 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

【0 2 2 2】

また、内部当籤役「F__R T 3 リブ__1 s t」～「F__R T 3 リブ__3 r d」はいずれも、押し順に応じて表示される図柄組合せが異なる押し順役であり、押し順が正解である場合には、略称「R T 3 移行リブ」に係る図柄組合せ（後述の図 2 8 参照）が有効ラインに沿って表示される。一方、押し順が正解でない場合には、略称「リプレイ」に係る図柄組合せ（後述の図 2 8 参照）のうちの図 1 8～図 2 3 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

10

【0 2 2 3】

また、内部当籤役「F__R T 4 リブ__1 2 3」～「F__R T 4 リブ__3 r d」はいずれも、押し順に応じて表示される図柄組合せが異なる押し順役であり、押し順が正解である場合には、略称「R T 4 移行リブ」に係る図柄組合せ（後述の図 2 8 参照）が有効ラインに沿って表示される。一方、押し順が正解でない場合には、略称「リプレイ」に係る図柄組合せ（後述の図 2 8 参照）のうちの図 1 8～図 2 3 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

20

【0 2 2 4】

また、内部当籤役「F__3 択ベル__1 s t」～「F__3 択ベル__3 r d」はいずれも、押し順に応じて表示される図柄組合せが異なる押し順役であり、押し順が正解である場合には、略称「ベル」に係る図柄組合せ（後述の図 1 9 参照）が有効ラインに沿って表示される。一方、押し順が正解でない場合には、略称「ベルこぼし目」に係る図柄組合せ（後述の図 2 8 参照）、又は、略称「1 枚出目」に係る図柄組合せ（後述の図 3 0 参照）が表示される。

【0 2 2 5】

また、内部当籤役「F__共通ベル」は、押し順役ではなく、押し順に関わらず略称「ベル」に係る図柄組合せ（後述の図 2 9 参照）のうちの図 1 8～図 2 3 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。また、内部当籤役「F__サボ 1」及び「F__サボ 2」はいずれも、押し順役ではなく、押し順に関わらず略称「サボテン」に係る図柄組合せ（後述の図 3 0 参照）のうちの図 1 8～図 2 3 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

30

【0 2 2 6】

また、内部当籤役「弱チェリー」は、押し順役ではなく、押し順に関わらず略称「弱チェリー」に係る図柄組合せ（後述の図 3 0 参照）のうちの図 1 8～図 2 3 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。また、内部当籤役「F__強チリ 1」及び「F__強チリ 2」はいずれも、押し順役ではなく、押し順に関わらず略称「強チェリー」に係る図柄組合せ（後述の図 3 0 参照）のうちの図 1 8～図 2 3 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

40

【0 2 2 7】

〔フラグ間状態中の当籤役と停止表示される図柄組合せとの対応関係〕

次に、図 2 5 を参照して、フラグ間状態における、内部当籤役と停止表示される図柄組合せとの対応関係について説明する。なお、図 2 5 は、フラグ間状態における、内部当籤役と停止表示される図柄組合せとの対応関係（一部の役については省略）を示す図であり、特に、フラグ間状態中にボーナス役（B B 役）に係る図柄組合せ（コンビネーション名称「C__B B 1」又は「C__B B 2」）を表示可能であるか否を示す図である。

【0 2 2 8】

図 2 5 の対応表中の「B B の成立可否」欄に記載の「○」印は、B B 役に係る図柄組み合

50

わせが表示可能であることを示し、「×」印は、B B 役に係る図柄組合せが表示不可能であることを示す。なお、B B 役に係る図柄組合せが表示不可能である場合には、内部当籤役としてボーナス役と重複して決定されている役に係る図柄組合せが表示される。例えば、内部当籤役「F__B B 1 + F__チリリプ」が当籤した場合（内部当籤役「F__B B 1」と、内部当籤役「F__チリリプ」とが重複当籤した場合）、図 2 5 に示すように、内部当籤役「F__B B 1」に係る図柄組合せを停止表示することはできず、内部当籤役「F__チリリプ」に係る図柄組合せが停止表示される。

【0 2 2 9】

また、フラグ間状態中において、B B 役に係る図柄組合せが表示不可能であり、かつ、ボーナス役と重複して決定されている役に係る図柄組合せが表示される場合、図 2 4 で説明した押し順正解時の図柄組合せのみを表示可能にしてもよいし、押し順不正解時の図柄組合せのみを表示可能にしてもよい。

10

【0 2 3 0】

例えば、内部当籤役「F__B B 1 + F__3 択ベル__1 s t」が当籤した場合、図 2 5 に示すように、内部当籤役「F__B B 1」に係る図柄組合せを停止表示することはできないので、内部当籤役「F__3 択ベル__1 s t」に係る図柄組合せが停止表示されるが、この際、押し順正解時に表示される略称「ベル」に係る図柄組合せのみを表示可能にし、押し順不正解時に表示される略称「ベルこぼし目」又は「1 枚出目」に係る図柄組合せを表示不可能にしてもよい（図 2 4 参照）。また、例えば、内部当籤役「F__B B 1 + F__R T 3 リプ__1 s t」が当籤した場合に、押し順不正解時に表示される略称「リプレイ」に係る図柄組合せのみを表示可能にし、押し順正解時に表示される略称「R T 3 移行リプ」に係る図柄組合せを表示不可能にしてもよい（図 2 4 参照）。

20

【0 2 3 1】

なお、フラグ間状態では、図 2 5 に示すように、ボーナス役（B B 役）と、内部当籤役「はずれ」、「F__特殊 1」、「F__特殊 2」及び「F__特殊 3」のいずれかとが重複して決定された場合、B B 役に係る図柄組合せを停止表示することができる。

【0 2 3 2】

〔リール停止初期設定テーブル〕

次に、図 2 6 を参照して、リール停止初期設定テーブルについて説明する。リール停止初期設定テーブルは、内部当籤役と、後述のリール停止制御処理で用いられる各種データとの対応関係を規定する。

30

【0 2 3 3】

図 2 6 に示すリール停止初期設定テーブルは、内部当籤役（小役当籤番号）と、引込優先順位テーブル選択テーブル番号、引込優先順位テーブル番号及び停止テーブル番号との対応関係を規定する。なお、図 2 6 には、参照される遊技状態、及び、内部当籤役の名称も併せて記載する。

【0 2 3 4】

引込優先順位テーブル選択テーブル番号、及び、引込優先順位テーブル番号は、引込優先順位テーブルの選択処理で用いられるデータである。例えば、リール停止初期設定テーブルにおいて、停止テーブル番号に対応する引込優先順位テーブル番号が規定されていれば、引込優先順位テーブル（後述の図 2 7 参照）に規定された引込優先順位テーブル番号に対応する表示役の優先順位に関するデータを取得することができる。一方、リール停止初期設定テーブルにおいて、停止テーブル番号に対応する引込優先順位テーブル番号が規定されていなければ、引込優先順位テーブル選択テーブル（不図示）を参照して、引込優先順位テーブル選択テーブル番号に対応する引込優先順位テーブル番号が決定される。

40

【0 2 3 5】

ここで、本実施形態のパチスロ 1 におけるリールの停止制御（停止図柄位置の決定手法）について簡単に説明する。本実施形態では、ストップスイッチにより停止操作が検出された後、該当するリールの回転が 1 9 0 m s e c 以内に停止するようにリールの停止制御が行われる。具体的には、停止操作が検出されたときの該当リルに応じた図柄カウンタの値

50

に、滑り駒数「0」～「4」のうちの何れかを加算し、得られた値に対応する図柄位置を、リールの回転が停止する図柄位置（以下、「停止予定位置」という）として決定する。なお、停止操作が検出されたときの該当リールに応じた図柄カウンタの値に対応する図柄位置は、リールの回転の停止が開始される図柄位置（以下、「停止開始位置」という）である。

【0236】

すなわち、滑り駒数は、ストップスイッチにより停止操作が検出されてから該当するリールの回転が停止するまでのリールの回転量である。言い換えれば、ストップスイッチにより停止操作が検出されてから該当するリールの回転が停止するまでの期間において、リール表示窓4の該当するリールの中段領域を通過する図柄の数である。これは、ストップス

10

【0237】

図示しない停止テーブルを参照すると、各リールの停止開始位置に応じて滑り駒数が取得される。なお、本実施形態では、停止テーブルに基づいて滑り駒数が取得されるが、これは仮のものであり、取得した滑り駒数が直ちにリールの停止予定位置が決定されるものではない。本実施形態では、停止テーブルに基づいて取得された滑り駒数（以下、「滑り駒数決定データ」という）より適切な滑り駒数が存在する場合には、後述する引込優先順位テーブル（後述の図27参照）を参照して滑り駒数を変更する。そして、滑り駒数決定データは、停止開始位置から最大滑り駒数である4個先の図柄位置までの各図柄について、優先順位の比較を行う際の検索順序を決定するために参照される。

20

【0238】

〔引込優先順位テーブル〕

次に、図27を参照して、引込優先順位テーブルについて説明する。引込優先順位テーブルは、引込優先順位テーブル番号「00」～「05」のそれぞれにおける、後述の入賞作動フラグ格納領域（後述の図28～図30参照）の種別毎の引込データ（入賞作動フラグデータ）と、予め定められたその優先順位との対応関係を規定する。

【0239】

引込優先順位テーブルは、停止テーブル（不図示）に基づいて得られた滑り駒数の他に、より適切な滑り駒数が存在するか否かを検索するために使用される。優先順位は、入賞に係る図柄組合せ（入賞作動フラグ）の種別間で優先的に停止表示される（引き込まれる）順位を規定するデータである。また、図27では、説明の便宜上、引込データ（入賞作動フラグデータ）の欄には、入賞作動フラグのコンビネーション名称を記載するが、実際の引込優先順位テーブルでは、各引込データは、後述の入賞作動フラグ格納領域（後述の図28～図30参照）に示すように、1バイトのデータで表され、該1バイトデータ中の各ビットに対して固有の図柄組合せ（入賞作動フラグ）が割り当てられる。

30

【0240】

本実施形態のリール停止制御では、まず、停止テーブル（不図示）に基づいて滑り駒数が取得される。しかしながら、優先順位に基づいて、この滑り駒数の他に、より適切な滑り駒数が存在する場合には、その適切な滑り駒数に変更する。すなわち、本実施形態では、停止テーブルにより取得された滑り駒数に関係なく、内部当籤役によって停止表示を許可する図柄組合せの優先順位に基づいて、より適切な滑り駒数を決定する。

40

【0241】

本実施形態では、優先順位が上位である図柄組合せの停止表示（引き込み）が、優先順位が下位である図柄組合せの停止表示よりも優先的に行われる。

【0242】

また、本実施形態では、図27に示すように、引込優先順位テーブル番号に応じて図柄組合せ（入賞作動フラグ）の優先順位が異なるだけでなく、優先順位の区分数も異なる。具体的には、引込優先順位テーブル番号が「00」である場合には、優先順位の区分数を5とし、引込優先順位テーブル番号が「01」又は「04」である場合には、優先順位の区分数を4とする。また、引込優先順位テーブル番号が「02」又は「03」である場合に

50

は、優先順位の区分数を2とし、引込優先順位テーブル番号が「05」である場合には、優先順位の区分数を3とする。

【0243】

ここでは、引込優先順位テーブル番号が「00」である場合の優先順位について説明し、それ以外の引込優先順位テーブル番号における優先順位の説明は省略する。引込優先順位テーブル番号が「00」である場合の優先順位「1」（最上位の優先順位）には、コンビネーション名称「C__9枚A__01」、「C__1確チリリプC__01」、「C__1確チリリプD__01」及び「C__RT3リプ__01」に対応する引込データが規定される。

【0244】

引込優先順位テーブル番号が「00」である場合の優先順位「2」には、コンビネーション名称「C__強2枚C__01」～「C__強2枚C__09」、「C__弱2枚B__01」～「C__弱2枚B__03」、「C__3枚E__01」、「C__3枚E__02」、「C__9枚F__01」～「C__9枚F__03」、「C__1確チリリプB__01」、「C__チリリプD__01」及び「C__チリリプC__01」に対応する引込データが規定される。

【0245】

引込優先順位テーブル番号が「00」である場合の優先順位「3」には、コンビネーション名称「C__1確チリリプA__01」、「C__チリリプA__01」、「C__チリリプB__01」及び「C__維持リプE__01」～「C__維持リプE__04」に対応する引込データが規定される。

【0246】

引込優先順位テーブル番号が「00」である場合の優先順位「4」には、コンビネーション名称「C__SP1__01」、「C__SP2__01」、「C__リーチ目リプP__01」、「C__リーチ目リプP__02」、「C__リーチ目リプO__01」、「C__リーチ目リプO__02」、「C__リーチ目リプN__01」、「C__リーチ目リプN__02」、「C__リーチ目リプM__01」、「C__リーチ目リプM__02」、「C__リーチ目リプL__01」～「C__リーチ目リプL__03」、「C__リーチ目リプK__01」～「C__リーチ目リプK__03」、「C__リーチ目リプJ__01」、「C__リーチ目リプI__01」～「C__リーチ目リプI__09」、「C__リーチ目リプH__01」～「C__リーチ目リプH__03」、「C__リーチ目リプG__01」、「C__リーチ目リプF__01」、「C__リーチ目リプF__02」、「C__リーチ目リプE__01」、「C__リーチ目リプD__01」、「C__リーチ目リプD__02」、「C__リーチ目リプC__01」～「C__リーチ目リプC__03」、「C__リーチ目リプB__01」、「C__リーチ目リプB__02」、「C__リーチ目リプA__01」、「C__維持リプF__01」、「C__維持リプF__02」、「C__維持リプD__01」～「C__維持リプD__04」、「C__維持リプC__01」～「C__維持リプC__03」、「C__維持リプB__01」、「C__維持リプB__02」及び「C__維持リプA__01」に対応する引込データが規定される。

【0247】

また、引込優先順位テーブル番号が「00」である場合の優先順位「5」（最下位の優先順位）には、コンビネーション名称「C__BB1」及び「C__BB2」に対応する引込データが規定される。

【0248】

<メインRAMに設けられている格納領域の構成>

次に、図28～図35を参照して、メインRAM103に設けられる各種格納領域の構成について説明する。

【0249】

[当り要求フラグ格納領域及び入賞作動フラグ格納領域]

まず、図28～図30を参照して、当り要求フラグ格納領域（内部当籤役格納領域）及び入賞作動フラグ格納領域（表示役格納領域）の構成について説明する。なお、本実施形態では、当り要求フラグ格納領域（フラグデータ格納領域、当籤フラグデータ格納領域）と、入賞作動フラグ格納領域（入賞フラグデータ格納領域）とは、互いに同じ構成を有する。

【0250】

本実施形態では、当り要求フラグ格納領域は、それぞれ1バイトのデータにより表される当り要求格納領域0～11で構成され、入賞作動フラグ格納領域は、それぞれ1バイトのデータにより表される入賞作動格納領域0～11で構成される。なお、当り要求フラグ格納領域及び入賞作動フラグ格納領域の各格納領域に格納されるデータは、図28～図30中の「データ」欄の1バイトデータのみであるが、図28～図30では、説明の便宜上、各格納領域のビットに対応付けられた、各リールの図柄組合せ（図中では、左リール3Lの図柄、中リール3Cの図柄及び右リール3Rの図柄の順で記載）、その名称（コンビネーション名称）及び略称、並びに、メダルの払出枚数も併せて記載する。

【0251】

当り要求フラグ格納領域0～11のそれぞれにおいて、所定のビットに「1」が格納されているとき、その所定のビットに対応する内部当籤役が内部当籤したことを示す。また、入賞作動格納領域0～11のそれぞれにおいて、所定のビットに「1」が格納されているとき、その所定のビットに対応する表示役（入賞作動フラグ）が入賞したことを示す。すなわち、所定のビットに「1」が格納されているとき、その所定のビットに対応する内部当籤役の各種図柄組合せが有効ライン上に表示されたことを示す。

【0252】

また、当り要求フラグ格納領域及び入賞作動フラグ格納領域では、図28～図30に示すように、各格納領域内の一つのビット（フラグ）に対して、複数の図柄組合せ（コンビネーション）が割り当てられているものもある。すなわち、そのようなフラグに対しては、停止表示可能な図柄組合せ（入賞可能なコンビネーション）が複数存在することを意味する。

【0253】

例えば、当り要求格納領域5及び入賞作動格納領域5のビット5には、図柄組合せ「サボテン2」－「白7」－「帽子」（コンビネーション名称「C__維持リブC__01」）、図柄組合せ「サボテン2」－「チリ上1」－「帽子」（コンビネーション名称「C__維持リブC__02」）、及び、図柄組合せ「サボテン2」－「サボテン2」－「帽子」（コンビネーション名称「C__維持リブC__03」）の3つの図柄組合せが割り当てられている。それゆえ、当り要求格納領域5のビット5に「1」が格納されている場合には、この3つの図柄組合せが有効ライン上に停止表示可能であることを示す。また、入賞作動格納領域5のビット5に「1」が格納されている場合には、この3つの図柄組合せのいずれかが有効ライン上に表示されたことを示す。

【0254】

〔持越役格納領域〕

次に、図31を参照して、持越役格納領域の構成について説明する。本実施形態では、持越役格納領域は、1バイトのデータ格納領域で構成される。

【0255】

内部抽籤の結果、内部当籤役「F__BB1」又は「F__BB2」が決定されたときには、その内部当籤役（BB役）は、持越役として持越役格納領域に格納される。持越役格納領域に格納された持越役は、対応する図柄組合せが有効ライン上に表示されるまでクリアされずに保持される。また、持越役格納領域に持越役が格納されている間、内部抽籤によって決定された内部当籤役に加えて、持越役が当り要求格納領域に格納される。

【0256】

〔遊技状態フラグ格納領域〕

次に、図32を参照して、遊技状態フラグ格納領域の構成について説明する。遊技状態フラグ格納領域は、1バイトのデータ格納領域で構成される。本実施形態では、図32に示すように、遊技状態フラグ格納領域の各ビットに対して固有のボーナスの種別又はRTの種別が割り当てられる。

【0257】

遊技状態フラグ格納領域において、所定のビットに「1」が格納されているとき、その所

10

20

30

40

50

定のビットに該当するボーナスゲーム又はR Tの作動が行われていることを示す。例えば、遊技状態フラグ格納領域のビット0に「1」が格納されているときには、ビッグボーナス「B B」の作動が行われており、遊技状態がB B遊技状態であることを示す。また、例えば、遊技状態フラグ格納領域のビット3に「1」が格納されているときは、遊技状態がR T 3状態であることを示す。

【0258】

〔作動ストップボタン格納領域〕

次に、図33を参照して、作動ストップボタン格納領域の構成について説明する。作動ストップボタン格納領域は、1バイトのデータ格納領域で構成され、1バイトからなる作動ストップボタンフラグを格納する。作動ストップボタンフラグにおいて、各ビットには、ストップボタンの操作状態が割り当てられる。

10

【0259】

例えば、左ストップボタン17Lが今回押されたストップボタン、つまり、作動ストップボタンである場合には、作動ストップボタン格納領域のビット0に「1」が格納される。また、例えば、左ストップボタン17Lが未だに押されていないストップボタン、つまり、有効ストップボタンである場合には、ビット4に「1」が格納される。メインCPU101は、作動ストップボタン格納領域に格納されているデータに基づいて、今回押されたストップボタンと未だに押されていないストップボタンとを識別する。

【0260】

〔押下順序格納領域〕

20

次に、図34を参照して、押下順序格納領域の構成について説明する。押下順序格納領域は、1バイトのデータ格納領域で構成され、1バイトからなる押下順序フラグを格納する。

【0261】

押下順序フラグにおいて、各ビットには、ストップボタンの押下順序の種別が割り当てられる。例えば、ストップボタンの押下順序が「左、中、右」である場合には、押下順序格納領域のビット0に「1」が格納される。

【0262】

〔図柄コード格納領域〕

次に、図35を参照して、図柄コード格納領域の構成について説明する。本実施形態では、図柄コード格納領域は、それぞれ1バイトのデータにより表される図柄コード格納領域0～11で構成される。なお、図柄コード格納領域は、当り要求フラグ格納領域及び入賞作動フラグ格納領域（図28～図30参照）と同様の構成となる。

30

【0263】

図柄コード格納領域では、有効ライン上に停止可能な図柄組合せ（コンビネーション）に対応するビットに「1」が格納される。なお、全てのリールが停止後、図柄コード格納領域0～11には、表示役（入賞作動フラグ）に対応する図柄コードが格納される。

【0264】

〔内部当籤役と各種サブフラグとの関係〕

一般遊技状態やART遊技状態において、主制御回路90による各種抽籤では各種データテーブルを参照するが、この際に用いるパラメータとして、本実施形態では、内部当籤役だけでなく、内部当籤役に対応する別の名称の各種パラメータ（以下、「サブフラグ（第1のサブフラグ）」、「サブフラグEX（第2のサブフラグ）」及び「サブフラグD」という）も用いる。それゆえ、本実施形態では、主制御回路90により、内部当籤役を各種サブフラグに変換する処理を行う（後述の図104中のサブフラグ変換処理、フラグ変換処理、サブフラグ圧縮処理参照）。なお、本実施形態では、内部当籤役に関する情報（通信パラメータ）として、サブフラグがスタートコマンドにセットされ、主制御回路90から副制御回路200に送信される。

40

【0265】

ここで、図36及び図37を参照して、内部当籤役と各種サブフラグとの対応関係について説明する。図36は、内部当籤役（小役当籤番号）と各種サブフラグとの対応関係を示

50

す図であり、図 3 7 は、内部当籤役（特賞当籤番号）とサブフラグとの対応関係を示す図である。

【0266】

本実施形態のフラグ変換処理では、まず、同じ種別に属する複数の内部当籤役を一つのサブフラグにまとめる。本実施形態では、このフラグ変換処理により、図 3 6 に示すように、小役及びリプレイ役に関する 3 2 種類の内部当籤役（小役当籤番号）が、1 8 種類のサブフラグ（「0 1」～「1 8」：フラグデータ）に変換される。例えば、内部当籤役「F__維持リブ__1 s t（1 0：小役当籤番号）」～「F__維持リブ__3 r d（1 2）」は、サブフラグ「押し順リブ 1（0 9：フラグデータ）」にまとめられる。なお、内部当籤役「はずれ」に対しては、サブフラグ「ハズレ（0 0）」が割り当てられる。

10

【0267】

また、本実施形態のフラグ変換処理では、図 3 6 に示すように、サブフラグ「ハズレ（0 0）」を含む 1 9 種類のサブフラグ（「0 0」～「1 8」）が、9 種類のサブフラグ E X（「0 0」～「0 8」：フラグデータ）に変換される。それゆえ、この変換処理では、サブフラグデータをさらに圧縮することができる。なお、この際、本実施形態では、抽籤（フラグ変換抽籤）によりサブフラグをサブフラグ E X に変換する。具体的には、次のように変換される。

【0268】

サブフラグ「ハズレ（0 0）」は、フラグ変換抽籤の結果に関係なく、サブフラグ E X「ハズレ（0 0）」に変換され、サブフラグ「2 連チリリブ（0 1）」は、フラグ変換抽籤の結果に関係なく、サブフラグ E X「リプレイ（0 1）」に変換される。

20

【0269】

サブフラグ「3 連チリリブ A（0 2）」及びサブフラグ「3 連チリリブ B（0 3）」は、フラグ変換抽籤に当籤した場合（後述の「変換有り」の場合）、サブフラグ E X「確定役（0 6）」又は「3 連チリリブ（0 7）」に変換され、フラグ変換抽籤に非当籤であった場合（後述の「変換無し」の場合）には、サブフラグ E X「リプレイ（0 1）」に変換される。

【0270】

サブフラグ「リーチ目リブ 1（0 4）」～「リーチ目リブ 4（0 7）」は、フラグ変換抽籤に当籤した場合、サブフラグ E X「確定役（0 6）」又は「リーチ目リブ（0 8）」に変換され、フラグ変換抽籤に非当籤であった場合には、サブフラグ E X「リプレイ（0 1）」に変換される。

30

【0271】

サブフラグ「リプレイ（0 8）」及び「押し順リブ 1（0 9）」～「押し順リブ 3（1 1）」は、フラグ変換抽籤の結果に関係なく、サブフラグ E X「リプレイ（0 1）」に変換され、サブフラグ「押し順ベル（1 2）」及び「共通ベル（1 3）」は、フラグ変換抽籤の結果に関係なく、サブフラグ E X「ベル（0 2）」に変換される。

【0272】

サブフラグ「サボテン（1 4）」、「弱チェリー（1 5）」及び「強チェリー（1 6）」は、フラグ変換抽籤の結果に関係なく、それぞれサブフラグ E X「サボテン（0 3）」、「弱チェリー（0 4）」及び「強チェリー（0 5）」に変換される。また、サブフラグ「リーチ目 1（1 7）」及び「リーチ目 2（1 8）」は、フラグ変換抽籤の結果に関係なく、サブフラグ E X「ハズレ（0 0）」に変換される。

40

【0273】

上述のように、本実施形態では、実質、サブフラグ「3 連チリリブ A（0 2）」、「3 連チリリブ B（0 3）」及び「リーチ目リブ 1（0 4）」～「リーチ目リブ 4（0 7）」のみがフラグ変換抽籤の対象となる。なお、上述したフラグ変換抽籤に用いられる抽籤テーブルについては、後で詳述する。

【0274】

さらに、本実施形態のフラグ変換処理では、図 3 6 に示すように、9 種類のサブフラグ E

50

X（「00」～「08」）が7種類のサブフラグD（「00」～「06」）に変換される。それゆえ、この変換処理では、より一層、サブフラグデータを圧縮することができる。なお、この変換処理では抽籤を行わず、次のようにして、サブフラグEXとサブフラグDとを対応付けて変換を行う。

【0275】

サブフラグEX「ハズレ（00）」、「リプレイ（01）」及び「ベル（02）」は、サブフラグD「ハズレ（00）」に変換される。サブフラグEX「サボテン（03）」は、サブフラグD「サボテン（01）」に変換され、サブフラグEX「弱チェリー（04）」は、サブフラグD「弱チェリー（02）」に変換され、サブフラグEX「強チェリー（05）」は、サブフラグD「強チェリー（03）」に変換される。

10

【0276】

また、サブフラグEX「確定役（06）」は、サブフラグD「確定役（04）」に変換され、サブフラグEX「3連チリリブ（07）」は、サブフラグD「3連チリリブ（05）」に変換され、サブフラグEX「リーチ目リブ（08）」は、サブフラグD「リーチ目リブ（06）」に変換される。

【0277】

また、本実施形態のフラグ変換処理では、図37に示すように、内部当籤役「F__BB1（01：特賞当籤番号）」及び「F__BB2（02）」はいずれも、サブフラグ「BB」に変換される。

20

【0278】

〔サブフラグEX変換時の遊技性〕

ここで、上述した内部当籤役をサブフラグ及びサブフラグEXに変換する処理の過程、及び、サブフラグEX変換時の遊技性の一例を、図38A及び38Bを参照して説明する。図38Aは、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定された場合のフラグ変換過程を示す図であり、図38Bは、内部当籤役「F__リーチ目リブA」～「F__リーチ目リブD」のいずれかが決定された場合のフラグ変換過程を示す図である。

【0279】

なお、本実施形態のパチスロ1では、RT4遊技状態中に内部当籤役「F__確チリリブ」、「F__1確チリリブ」及び「F__リーチ目リブA」～「F__リーチ目リブD」のいずれかが単独で内部当籤役として決定されると、フラグ変換抽籤を行う。そして、本実施形態では、このフラグ変換抽籤に当籤した場合、特別な特典（例えば、ARTゲーム数の上乘せやCT当籤）が付与される。

30

【0280】

例えば、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定された場合、図38Aに示すように、内部当籤役「F__確チリリブ」及び「F__1確チリリブ」は、それぞれサブフラグ「3連チリリブA（02）」及び「3連チリリブB（03）」に変換される。

【0281】

次いで、サブフラグ「3連チリリブA（02）」及び「3連チリリブB（03）」は、フラグ変換抽籤に当籤すると、サブフラグEX「3連チリリブ（07）」又は「確定役（06）」に変換される。一方、フラグ変換抽籤に非当籤であった場合には、サブフラグ「3連チリリブA（02）」及び「3連チリリブB（03）」はともに、サブフラグEX「リプレイ（01）」に変換される。

40

【0282】

なお、図24で説明したように、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が当籤した場合、押し順正解時には略称「3連チリリブ」に係る図柄組合せが表示され、押し順不正解時には略称「リプレイ」に係る図柄組合せが表示される。それゆえ、本実施形態では、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定され、かつ、フラグ変換抽籤に当籤した場合、内部当籤役「F__確チリリブ」及び「F__1確チリリブ」はいずれも、サブフラグEX「3連チリリブ（07）」又は「確定役（06）」の役

50

として扱われる。

【0283】

そして、このフラグ変換過程によって内部当籤役「F__確チリリプ」又は「F__1確チリリプ」がサブフラグEX「3連チリリプ(07)」又は「確定役(06)」に変換されると、略称「3連チリリプ」に係る図柄組合せを表示するための情報が報知される(例えば、遊技者に対して順押しでチリ図柄を狙わせる旨の情報が報知される)。一方、このフラグ変換過程において、フラグ変換抽籤が非当籤となり、内部当籤役「F__確チリリプ」又は「F__1確チリリプ」がサブフラグEX「リプレイ(01)」に変換されると、略称「リプレイ」に係る図柄組合せを表示するための情報が報知される(例えば、順押し以外の押し順(変則押し)が報知される)。

10

【0284】

また、例えば、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」のいずれかが決定された場合、図38Bに示すように、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」は、それぞれサブフラグ「リーチ目リプ1(04)」～「リーチ目リプ4(07)」に変換される。

【0285】

次いで、サブフラグ「リーチ目リプ1(04)」～「リーチ目リプ4(07)」は、フラグ変換抽籤に当籤すると、サブフラグEX「リーチ目リプ(08)」又は「確定役(06)」に変換される。一方、フラグ変換抽籤に非当籤であった場合には、サブフラグ「リーチ目リプ1(04)」～「リーチ目リプ4(07)」は、サブフラグEX「リプレイ(01)」に変換される。

20

【0286】

なお、図24で説明したように、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」のいずれかが当籤した場合、押し順正解時には略称「リーチ目リプ」に係る図柄組合せが表示され、押し順不正解時には略称「リプレイ」に係る図柄組合せが表示される。それゆえ、本実施形態では、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」のいずれかが決定され、かつ、フラグ変換抽籤に当籤した場合、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」はいずれも、サブフラグEX「リーチ目リプ(08)」又は「確定役(06)」の役として扱われる。

【0287】

そして、このフラグ変換過程によって内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」が例えばサブフラグEX「リーチ目リプ(08)」又は「確定役(06)」に変換されると、略称「リーチ目リプ」に係る図柄組合せを表示するための情報が報知される(例えば、遊技者に対して順押しで図柄「白7」を狙わせる旨の情報が報知される)。一方、このフラグ変換過程において、フラグ変換抽籤が非当籤となり、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」がサブフラグEX「リプレイ(01)」に変換されると、略称「リプレイ」に係る図柄組合せを表示するための情報が報知される(例えば、順押し以外の押し順(変則押し)が報知される)。

30

【0288】

また、本実施形態では、図38A又は38Bに示すフラグ変換過程において、フラグ変換抽籤に当籤して報知に従い遊技者が停止操作を行うと、略称「3連チリリプ」又は「リーチ目リプ」に係る図柄組合せが有効ライン上に停止表示され、特別な特典が付与される。この付与処理は、実質的には処理上において、パチスロ1がフラグ変換抽籤に当籤したことに応じて特別な特典を遊技者に付与することになるが、遊技者に対しては、略称「3連チリリプ」に係る図柄組合せが表示されたことにより、特別な特典が付与されたと感じさせることができる。

40

【0289】

パチスロの遊技性を高めるためには、特典が付与される図柄組合せの出現頻度が一定であるよりも、状態に応じて異なる方が好ましい場合がある。停止制御(表示される図柄組合せ)は、内部当籤役の種類によって異なるため、特典が付与される図柄組合せの出現頻度

50

を状態に応じて異ならせる手法としては、内部当籤役の当籤確率を異ならせる手法も考えられる（パチスロ1では、内部当籤役の当籤確率は、ボーナスの作動の有無やRT状態に応じて異ならせることができるため、例えば、ART遊技状態に対応するRT状態として、RT4状態だけでなく、RT6状態やRT7状態などの他のRT状態を設けるという手法も考えられる）。しかしながら、内部当籤役の当籤確率を異ならせる契機（RT状態の移行契機）は限定されているため、遊技性（興趣）の向上という観点では、この手法は柔軟性に欠けている。

【0290】

それに対して、本実施形態のパチスロ1では、内部当籤役の当籤確率を変えなく、内部当籤役を決定するための内部抽籤に加え、フラグ変換抽籤及びその抽籤結果に基づく報知を行うことにより、特典が付与される図柄組合せの出現頻度を状態に応じて柔軟に異ならせることができる。すなわち、フラグ変換抽籤に当籤し易い状態では、特典が付与される図柄組合せの出現頻度を上げることができ、逆に、フラグ変換抽籤に当籤し難い状態では、特典が付与される図柄組合せの出現頻度を下げることができる。

【0291】

＜一般遊技状態中の遊技性＞

次に、図39A～39Cを参照して、一般遊技状態中の遊技の流れについて説明する。本実施形態のパチスロ1では、一般遊技状態中において、遊技状態が通常遊技状態からCZに移行し、その後、遊技状態がCZからART遊技状態に移行することにより、一般遊技状態（非ART遊技状態）からART遊技状態への移行が行われる（図14A及び14B参照）。

【0292】

図39Aは、一般遊技状態中において、遊技状態が通常遊技状態からCZに移行する際の遊技の流れを示す図である。通常遊技状態は、図39Aに示すように、CZの抽籤状態として低確率状態と高確率状態とを有する。この低確率状態及び高確率状態は、通常遊技状態中に行われるCZ抽籤に当籤する期待度が互いに異なる状態であり、低確率状態はCZ抽籤に当籤し難い状態であり、高確率状態はCZ抽籤に当籤し易い状態である。そして、通常遊技状態中の遊技において行われるCZ抽籤に当籤した場合には、遊技状態が通常遊技状態からCZに移行する。

【0293】

なお、本実施形態のパチスロ1では、CZ（チャンスゾーン）として、「CZ1」、「CZ2」及び「CZ3」の複数のチャンスゾーンを設ける。CZ1～CZ3は、CZ中の遊技で行われるART抽籤に当籤する期待度が互いに異なるチャンスゾーンであり、CZ3は、ART抽籤に必ず当籤するチャンスゾーンであり、CZ1及びCZ2は、所定の確率でART抽籤に当籤するチャンスゾーンである。通常遊技状態中の遊技で行われるCZ抽籤では、CZの当籤／非当籤だけでなく、当籤時に移行するCZの種別（CZ1～CZ3のいずれか）も決定される（後述の図41参照）。

【0294】

図39Bは、遊技状態が一般遊技状態のCZ1及びCZ2からART遊技状態に移行する際の遊技の流れを示す図である。CZ1及びCZ2はともに、前半部と後半部とから構成される。前半部は、CZ中の遊技で行われるART抽籤に当籤する期待度のランクを昇格させる期間であり、後半部は、ランクに基づくART抽籤の抽籤結果を所定の演出（本実施形態では、キャラクタによるバトル演出）により報知する期間である。

【0295】

CZ1中では、ランクとして6段階のモード（モード1～6）が用意され、モードが上がるほど、ART抽籤に当籤する期待度が高くなる。CZ1の前半部では、第1の所定ゲーム数（例えば、最大で12ゲーム）の期間、継続して遊技が行われ、内部当籤役に基づいてモードの昇格抽籤が行われる。そして、CZ1の後半部の1ゲーム目では、前半部で昇格させたモード（前半部終了時点のモード）に基づいてART抽籤が行われる。

【0296】

また、C Z 2 中では、ランクとして10段階のポイントが用意され、ポイントが上がるほど、A R T 抽籤に当籤する期待度が高くなる。C Z 2 の前半部では、第2の所定ゲーム数（例えば、最大で15ゲーム）の期間、継続して遊技が行われ、内部当籤役に基づいてポイントの昇格抽籤が行われる。そして、C Z 2 の後半部の1ゲーム目では、前半部で昇格させたポイント（前半部終了時点のポイント）に基づいてA R T 抽籤が行われる。

【0297】

C Z 1 の後半部では、味方キャラクタと敵キャラクタAとが対戦するバトル演出が行われ、C Z 2 の後半部では、味方キャラクタと敵キャラクタBとが対戦するバトル演出が行われる。このバトル演出は、第3の所定ゲーム数（例えば、最大で4ゲーム）の期間の遊技に渡って行われる。また、バトル演出の勝敗は、A R T 抽籤の結果に基づいて管理（決定）され、A R T 抽籤に当籤している場合には、バトル演出で味方キャラクタが勝利し、非当籤である場合には、バトル演出で敵キャラクタが勝利する。

【0298】

また、C Z 1 及びC Z 2 の各後半部（バトル演出中）では、毎ゲーム、内部当籤役に基づいてA R T 抽籤が行われる。そして、このA R T 抽籤に当籤すると、バトル演出の結果が書き換えられる。例えば、バトル演出中にいわゆる「レア」役が内部当籤役として決定されると、A R T 抽籤が行われ、その結果に基づいてバトル演出の結果が書き換えられる。

【0299】

C Z 1 及びC Z 2 において、A R T に非当籤の場合には、後半部のバトル演出で敗北し、基本的には、その後、遊技状態が通常遊技状態に移行する。一方、C Z 1 及びC Z 2 において、A R T に当籤している場合には、後半部のバトル演出で勝利し、その後、遊技状態がC Z からA R T 準備状態を経由して通常A R T に移行する。なお、本実施形態では、C Z 1 及びC Z 2 の前半部の遊技において、フリーズが発生する場合があります、その場合には、遊技状態がC Z からA R T 準備状態を経由して、通常A R T ではなくC T （上乗せチャンスゾーン）に移行する。

【0300】

図39Cは、遊技状態が一般遊技状態のC Z 3 からA R T 遊技状態に移行する際の遊技の流れを示す図である。C Z 3 は、第4の所定ゲーム数（例えば、最大で17ゲーム）の期間、継続して遊技が行われる。そして、C Z 3 では、毎ゲーム、内部当籤役に基づいてA R T 抽籤が行われる。

【0301】

C Z 3 は、A R T 抽籤に当籤した時点で終了し、その次のゲーム以降、遊技状態がC Z 3 からA R T 準備状態を経由してC T （上乗せチャンスゾーン）に移行する。また、C Z 3 では、フリーズが発生する場合があります、その場合にも、次ゲーム以降、遊技状態がC Z 3 からA R T 準備状態を経由してC T （上乗せチャンスゾーン）に移行する。一方、C Z 3 において、A R T 抽籤に当籤せずにC Z 3 の遊技期間（第4の所定ゲーム数）が経過した場合、遊技状態がC Z 3 からA R T 準備状態を経由して通常A R T に移行する。すなわち、本実施形態では、C Z 3 は、A R T 遊技状態への移行が確定しているチャンスゾーンである。

【0302】

<一般遊技状態中に用いる各種データテーブル>

続いて、図40～図45を参照して、一般遊技状態中に行われる遊技性に関する抽籤処理で用いられる各種データテーブルについて説明する。なお、以下に説明する各種データテーブルは、メインROM102に格納される。

【0303】

また、以下に示す各種データテーブルでは、抽籤値の情報を概念的に示す。データテーブル中の「0」は、当籤確率「0%」に相当する抽籤値が規定されていることを意味し、「極々低」は、当籤確率「0%～1%未満」に相当する抽籤値が規定されていることを意味し、「極低」は、当籤確率「1%～10%未満」に相当する抽籤値が規定されていることを意味する。また、データテーブル中の「低」は、当籤確率「10%～30%未満」に相

10

20

30

40

50

当する抽籤値が規定されていることを意味し、「中」は、当籤確率「30%～60%未満」に相当する抽籤値が規定されていることを意味し、「高」は、当籤確率「60%～80%未満」に相当する抽籤値が規定されていることを意味する。さらに、データテーブル中の「極高」は、当籤確率「80%～99%未満」に相当する抽籤値が規定されていることを意味し、「極々高」は、当籤確率「99%～100%未満」に相当する抽籤値が規定されていることを意味し、「確定」は、当籤確率「100%」に相当する抽籤値が規定されていることを意味する。

【0304】

そして、以下に示す各種データテーブルでは、乱数回路110の乱数レジスタ1により、予め定められた数値の範囲（0～65535）から抽出される抽籤用乱数値を、規定された抽籤値で順次減算し、減算の結果が負となったか否か（いわゆる「桁かり」が生じたか否か）の判定を行うことによって内部的な抽籤が行われる。なお、本実施形態では、一般遊技状態中に行われる遊技性に関する抽籤処理において抽籤用乱数値から抽籤値を減算して当籤／非当籤を判定する例を説明したが、本発明はこれに限定されず、抽出した抽籤用乱数値に抽籤値を加算し、加算結果が65536を超えたか否か（いわゆる「桁あふれ」が生じたか否か）を判定して、当籤／非当籤を決定してもよい。

【0305】

〔通常中高確率抽籤テーブル〕

まず、図40A及び40Bを参照して、CZの抽籤状態（低確率及び高確率）の移行抽籤で用いられる通常中高確率抽籤テーブルについて説明する。なお、本実施形態のパチスロ1では、毎ゲーム、内部当籤役に基づいてCZの抽籤状態の移行抽籤が行われるだけでなく、例えばボーナス終了時やCZ，ART終了時などの場合にもCZの抽籤状態の移行抽籤が行われる。図40Aは、通常遊技状態中に毎ゲーム参照される通常中高確率抽籤テーブルの構成図であり、図40Bは、例えば設定変更時、ボーナス終了時又はCZ，ART終了時等に参照される通常中高確率抽籤テーブルの構成図である。なお、図40Aに示す内部当籤役の名称は、上述したサブフラグの名称に対応する。

【0306】

図40Aに示す通常中高確率抽籤テーブルは、現在のCZの抽籤状態と内部当籤役との各組合せと、移行後のCZの抽籤状態の抽籤結果（低確率／高確率）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。

【0307】

図40Aに示す通常中高確率抽籤テーブルから明らかなように、現在のCZの抽籤状態が低確率である場合には、内部当籤役がサブフラグ「弱チェリー」に対応する役であるときに、CZの抽籤状態が高確率に移行し易くなる。一方、現在のCZの抽籤状態が高確率である場合には、内部当籤役がサブフラグ「共通ベル」、「サボテン」、「弱チェリー」及び「強チェリー」のいずれかに対応する役であるときに、CZの抽籤状態が高確率に維持される。

【0308】

図40Bに示す通常中高確率抽籤テーブルは、該テーブルを参照する際の各状況と、移行後のCZの抽籤状態の抽籤結果（低確率／高確率）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。図40Bに示す通常中高確率抽籤テーブルから明らかなように、ボーナス終了時にはCZの抽籤状態が必ず高確率に移行する。

【0309】

〔CZ抽籤テーブル〕

次に、図41A及び41Bを参照して、CZ抽籤で用いられるCZ抽籤テーブルについて説明する。図41Aは、通常遊技状態中に内部当籤役に基づいてCZ抽籤を行う際に用いられるCZ抽籤テーブルの構成図であり、図41Bは、例えばCZ失敗時やART終了時などにおいて、CZの引き戻しを行うか否かのCZ抽籤を行う際に用いられるCZ抽籤テーブルの構成図である。なお、図41Aに示す内部当籤役の名称は、上述したサブフラグの名称に対応する。

10

20

30

40

50

【0310】

図41Aに示すCZ抽籤テーブルは、現在のCZの抽籤状態と内部当籤役との各組合せと、CZ1、CZ2、CZ3の当籤／非当籤（抽籤結果）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。図41Aに示すCZ抽籤テーブルから明らかなように、現在のCZの抽籤状態が高確率中である場合には、現在のCZの抽籤状態が低確率中である場合よりも、CZ抽籤に当籤する確率が高くなる。

【0311】

図41Bに示すCZ抽籤テーブルは、CZ失敗時やART終了時における、CZ1、CZ2、CZ3の当籤／非当籤（抽籤結果）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。CZ失敗時（CZ1、CZ2中のART抽籤に非当籤時）やART遊技状態の終了時には、このCZ抽籤テーブルを用いてCZの引き戻し抽籤が行われる。

【0312】

[CZ1中モードアップ抽籤テーブル]

次に、図42を参照して、CZ1の前半部において行われるCZ1のモードアップ抽籤で用いられるCZ1中モードアップ抽籤テーブルについて説明する。図42は、CZ1中モードアップ抽籤テーブルの構成図である。なお、図42に示す内部当籤役の名称は、上述したサブフラグの名称に対応する。

【0313】

CZ1中モードアップ抽籤テーブルは、現在のモードと内部当籤役との各組合せと、モードアップ抽籤の結果（当籤／非当籤）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。後述の図44Aに示すように、CZ1では、モードが上がる（モードの値が高くなる）ほどART抽籤に当籤する確率が上がり、モードがモード6まで上がると、ART抽籤に必ず当籤する。

【0314】

なお、図42中の抽籤結果「モード1UP」とは、CZ1のモードが1段階上がることを意味し、抽籤結果「モード2UP」とは、CZ1のモードが2段階上がることを意味する。それゆえ、例えば、現在のモードがモード2である状況において、抽籤結果「モード2UP」に当籤すると、CZ1のモードはモード2からモード4に上がる。また、例えば、抽籤結果「モード6UP__フリーズ発生」に当籤すると、フリーズが発生し、ART抽籤の当籤及びCTの付与が決定される。

【0315】

[CZ2中ポイント抽籤テーブル]

次に、図43を参照して、CZ2の前半部において行われるCZ2のポイントアップ抽籤で用いられるCZ2中ポイント抽籤テーブルについて説明する。図43は、CZ2中ポイント抽籤テーブルの構成図である。なお、図43に示す内部当籤役の名称は、上述したサブフラグの名称に対応する。

【0316】

CZ2中ポイント抽籤テーブルは、現在のポイントと内部当籤役との各組合せと、ポイントアップ抽籤の結果（当籤／非当籤）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。後述の図44Bに示すように、CZ2では、ポイントが上がるほどART抽籤に当籤する確率が上がり、ポイントが「ポイント10」まで上がると、ART抽籤に必ず当籤する。なお、図43中の抽籤結果「ポイント2UP」とは、現在のCZ2のポイントに「2」が加算されることを意味し、例えば、現在のポイントが「2」である状況において、抽籤結果「ポイント2UP」に当籤すると、CZ2のポイントは「2」から「4」に上がる。また、例えば、抽籤結果の「ポイント10UP__フリーズ発生」に当籤すると、フリーズが発生し、ART抽籤の当籤及びCTの付与が決定される。

【0317】

[CZ中ART抽籤テーブル]

次に、図44A～44C及び図45を参照して、CZ中に実行されるART抽籤で用いられるCZ中ART抽籤テーブルについて説明する。なお、図44Aは、CZ1の後半部の

10

20

30

40

50

1 ゲーム目で用いられる C Z 中 A R T 抽籤テーブル (C Z 1 用) の構成図であり、図 4 4 B は、C Z 2 の後半部の 1 ゲーム目で用いられる C Z 中 A R T 抽籤テーブル (C Z 2 用) の構成図であり、図 4 4 C は、C Z 1, C Z 2 の後半部で用いられる C Z 中 A R T 抽籤テーブル (C Z 1, C Z 2 共通 後半バトル中用) の構成図である。また、図 4 5 は、C Z 3 中に実行される A R T 抽籤で用いられる C Z 中 A R T 抽籤テーブル (C Z 3 用) の構成図である。なお、図 4 4 C 及び図 4 5 に示す内部当籤役の名称は、上述したサブフラグの名称に対応する。

【0318】

図 4 4 A に示す C Z 中 A R T 抽籤テーブル (C Z 1 用) は、現在のモードと、A R T 抽籤の結果 (当籤の有無) と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。また、図 4 4 B に示す C Z 中 A R T 抽籤テーブル (C Z 2 用) は、現在のポイントと、A R T 抽籤の結果 (当籤の有無) と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。

10

【0319】

C Z 中 A R T 抽籤テーブル (C Z 1 用) 及び C Z 中 A R T 抽籤テーブル (C Z 2 用) から明らかなように、C Z 1 及び C Z 2 では前半部のランク (モード又はポイント) が上がるほど、A R T 抽籤に当籤し易くなる。

【0320】

図 4 4 C に示す C Z 中 A R T 抽籤テーブル (C Z 1, C Z 2 共通 後半バトル中用) は、内部当籤役と、A R T 抽籤の結果 (当籤の有無) と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。C Z 中 A R T 抽籤テーブル (C Z 1, C Z 2 共通 後半バトル中用) から明らかなように、C Z 1 及び C Z 2 の後半部において、レア役 (サブフラグ「弱チェリー」、「サボテン」又は「強チェリー」に対応する役) が内部当籤役として決定されると、所定の確率で A R T 抽籤に当籤する。

20

【0321】

図 4 5 に示す C Z 中 A R T 抽籤テーブル (C Z 3 用) は、C Z 3 の消化ゲーム数と内部当籤役との各組合せと、A R T 抽籤の結果 (当籤の有無) と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。C Z 中 A R T 抽籤テーブル (C Z 3 用) から明らかなように、本実施形態では、C Z 3 中において A R T 抽籤に当籤すると必ず C T にも当籤する。

30

【0322】

< 通常 A R T 中の遊技性 >

次に、図 4 6 A 及び 4 6 B を参照して、遊技 A R T 中の遊技の流れについて説明する。本実施形態のパチスロ 1 では、上述のように、A R T 遊技状態として、通常 A R T と C T とが設けられ (図 1 4 A 及び 1 4 B 参照)、C T 中を上乗せチャンスゾーンとしている。それゆえ、本実施形態では、遊技者は、通常 A R T 中の遊技において、C T への移行を目指して遊技を行うことになる。

【0323】

[通常 A R T から C T への移行態様]

図 4 6 A は、通常 A R T から C T への遊技状態の移行態様を示す図である。本実施形態のパチスロ 1 では、図 4 6 A に示すように、通常 A R T 中に行われる C T 抽籤に当籤した場合、遊技状態が通常 A R T から C T に移行する。なお、本実施形態のパチスロ 1 は、図 4 6 A に示すように、通常 A R T 中に行われる様々な抽籤に影響を与えるパラメータとして、A R T レベル及び C T 抽籤状態が設けられる。

40

【0324】

A R T レベルとしては、レベル 1 ~ レベル 4 の 4 段階のレベルが設けられ、この A R T レベルは、主に通常 A R T 中の継続 (消化) ゲーム数に基づいて制御 (決定) される。そして、A R T レベルは、C T 抽籤状態の決定や後述する通常 A R T 中のフラグ変換抽籤などに対して影響を与える。

【0325】

50

C T抽籤状態としては、低確率、通常、高確率及び超高確率の4段階の状態が設けられ、C T抽籤状態は、主に、A R Tレベルや通常A R T中の内部当籤役などに基づいて制御（決定）される。そして、C T抽籤状態は、通常A R T中に行うC T抽籤や後述する通常A R T中のフラグ変換抽籤などに対して影響を与える。

【0326】

〔通常A R T中のフラグ変換〕

上述のように、本実施形態のパチスロ1では、R T 4状態中、すなわち、A R T遊技状態中に、内部当籤役「F__確チリリプ」、「F__1確チリリプ」及び「F__リーチ目リプA」～「F__リーチ目リプD」のいずれかが単独で内部当籤役として決定されると、フラグ変換抽籤を行い、その抽籤結果に応じて特別な特典（例えば、A R Tゲーム数の上乘せやC T当籤）を付与する。図46Bは、通常A R T中に行われるフラグ変換抽籤の手法の概要を示す図である。

【0327】

本実施形態では、図46Bに示すように、通常A R T中において、A R Tレベル及びC T抽籤状態を参照して、フラグ変換抽籤が行われる。その結果、フラグ変換抽籤に当籤した場合には、特別な特典を付与するとともに、略称「3連チリリプ」に係る図柄組合せや略称「リーチ目リプ」に係る図柄組合せなどを有効ライン上に停止表示させるためのナビ（例えば、順押しで所定の図柄を狙わせる旨の情報の報知）が行われる。一方、フラグ変換抽籤に非当籤であった場合には、略称「リプレイ」に係る図柄組合せを有効ライン上に停止表示させるためのナビ（例えば、順押し以外の押し順の報知）が行われる。

【0328】

そして、遊技者がこの報知（ナビ）に従い停止操作を行うと、報知内容に応じた図柄組合せが有効ライン上に停止表示される。具体的には、フラグ変換抽籤に当籤した場合には略称「3連チリリプ」に係る図柄組合せや略称「リーチ目リプ」に係る図柄組合せなどが有効ライン上に停止表示され、フラグ変換抽籤に非当籤であった場合には略称「リプレイ」に係る図柄組合せが有効ライン上に停止表示される。

【0329】

＜通常A R T中に用いる各種データテーブル＞

次に、図47～図51を参照して、通常A R T中の抽籤処理で用いられる各種データテーブルについて説明する。なお、以下に説明する各種データテーブルは、メインROM102に格納される。

【0330】

〔A R T中フラグ変換抽籤テーブル〕

図47A及び47Bは、通常A R T中に行われるフラグ変換抽籤で用いられるA R T中フラグ変換抽籤テーブルの構成図である。

【0331】

本実施形態に係るパチスロ1では、通常A R T中のフラグ変換抽籤を2段階で行う。具体的には、内部当籤役「F__確チリリプ」又は「F__1確チリリプ」が当籤した場合、まず、1段階目のフラグ変換抽籤が行われ、この1段階目のフラグ変換抽籤に当籤すると、その後、2段階目のフラグ変換抽籤が行われる。そして、この2段階目のフラグ変換抽籤に当籤すると、内部当籤役「F__確チリリプ」又は「F__1確チリリプ」がサブフラグEX「3連チリリプ」に変換される。一方、1段階目のフラグ変換抽籤又は2段階目のフラグ変換抽籤が非当籤であった場合には、内部当籤役「F__確チリリプ」又は「F__1確チリリプ」はサブフラグEX「リプレイ」に変換される（通常のリプレイ役として扱う）。なお、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」のいずれかが当籤した場合には、2段階目のフラグ変換抽籤のみが行われる。

【0332】

図47Aは、1段階目のフラグ変換抽籤で用いられるA R T中フラグ変換抽籤テーブルの構成図であり、図47Bは、2段階目のフラグ変換抽籤で用いられるA R T中フラグ変換抽籤テーブルの構成図である。

【0333】

図47Aに示すART中フラグ変換抽籤テーブルは、内部当籤役（「F__確チリリプ」又は「F__1確チリリプ」）と、1段階目のフラグ変換抽籤の抽籤結果（変換無し／変換有り（仮））と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。

【0334】

図47Bに示すART中フラグ変換抽籤テーブルは、内部当籤役とARTレベルとCT抽籤状態との各組合せと、2段階目のフラグ変換抽籤の抽籤結果（変換無し／変換有り）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。なお、通常ARTにおいて1回、CTに当籤するまでの遊技では、図47B中の項目「ARTレベル」の「初回（一度CTに当籤するまで）」欄のテーブルが参照される。

10

【0335】

本実施形態において、図47A及び47Bに示すように、ART中フラグ変換抽籤テーブルのそれぞれを用いた段階目及び2段階目のフラグ変換抽籤では、確率分母が「256」となる乱数値（0～255）を用いて抽籤が行われる。それゆえ、本実施形態では、上述した2段階のフラグ変換抽籤は、確率分母が「65536」となる乱数値を用いて一回抽籤を行う場合と実質同一の抽籤であるとみなすことができる。

【0336】

近年のパチスロでは、従来、副制御基板72側（以下、「サブ側」という）で行っていた出玉に関する抽籤（ART抽籤など）を主制御基板71側（以下、「メイン側」という）で行うことが求められている。しかしながら、メイン側の記憶手段（メインROM102）の容量が小容量に制限されているため、処理容量の増加を抑えつつ遊技性を損なうことのない抽籤を可能にする仕組みが求められている。

20

【0337】

この点に関して、本実施形態のパチスロ1では、確率分母が「256」となる抽籤を2段階で行うことにより、確率分母が「65536」となる抽籤を行うことができるので、抽籤処理に係るメイン側の容量の増加を抑えることができる。また、2段階目の抽籤では、ARTレベルやCT抽籤状態などを参照するので、内部当籤役だけでなく現在の状態に応じたフラグ変換抽籤を行うことができ、その結果、多様な遊技性を持ったフラグ変換抽籤を行うことができる。

【0338】

[ARTレベル決定テーブル]

図48A及び48Bは、ARTレベルを決定する際に用いられるARTレベル決定テーブルの構成図である。なお、ARTレベルの決定処理は、ART遊技状態への移行が決まったART当籤時、及び、通常ART中に行われる。図48Aは、ART当籤時に用いられるARTレベル決定テーブルの構成図であり、図48Bは、通常ART中に用いられるARTレベル決定テーブルの構成図である。

30

【0339】

図48Aに示すARTレベル決定テーブルは、ARTレベル1～4（抽籤結果）と、各ARTレベルに対応付けられた抽籤値の情報との対応関係を規定する。なお、本実施形態では、ART当籤時にフリーズが発生している場合には、ARTレベルとしてARTレベル2が決定される。

40

【0340】

図48Bに示すARTレベル決定テーブルは、現在のARTレベルと通常ARTの経過（消化）ゲーム数との各組合せと、移行先の各種ARTレベルと、移行先の各ARTレベルに対応付けられた抽籤値の情報との対応関係を規定する。また、図48Bに示すARTレベル決定テーブルは、現在のARTレベルとCT突入時の通常ARTの経過ゲーム数との各組合せと、移行先の各種ARTレベルと、移行先の各ARTレベルに対応付けられた抽籤値の情報との対応関係を規定する。すなわち、通常ART中では、通常ARTの経過（消化）ゲーム数が所定ゲーム数に到達したタイミングでARTレベルが移行可能となるだけでなく、通常ART中においてCTに突入したタイミングにおいてもARTレベルが移

50

行可能となる。

【0341】

〔通常ART中高確率抽籤テーブル〕

図49は、通常ART中においてCT抽籤状態を決定する際に用いられる通常ART中高確率抽籤テーブルの構成図である。

【0342】

通常ART中高確率抽籤テーブルは、現在のCT抽籤状態と内部当籤役との各組合せと、移行先の各種CT抽籤状態と、各CT抽籤状態に対応付けられた抽籤値の情報との対応関係を規定する。なお、図49に示す内部当籤役の名称は、上述したサブフラグの名称に対応する。

【0343】

通常ART中高確率抽籤テーブルから明らかなように、サブフラグ「3連チリリプ（3連チリリプA及び3連チリリプB）」やサブフラグ「リーチ目リプ（リーチ目リプ1～4）」に対応する内部当籤役が当籤している場合、CT抽籤状態が「低確率」に移行（転落）し易くなる。ただし、後述の図50に示すように、サブフラグ「3連チリリプ」や「リーチ目リプ」に対応する内部当籤役が当籤している場合には、CT抽籤状態が転落しても、CT抽籤に必ず当籤する構成になっている。

【0344】

〔ART中CT抽籤テーブル〕

図50は、通常ART中に行われるCT抽籤で用いられるART中CT抽籤テーブルの構成図である。

【0345】

ART中CT抽籤テーブルは、現在のCT抽籤状態と内部当籤役との各組合せと、CT抽籤の各種抽籤結果（非当籤／通常CT／高確率CT）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。なお、図50に示す内部当籤役の名称は、上述したサブフラグの名称に対応する。

【0346】

本実施形態において、内部当籤役として、サブフラグ「サボテン」、「弱チェリー」、「強チェリー」、「3連チリリプ（3連チリリプA及び3連チリリプB）」、「リーチ目リプ（リーチ目リプ1～4）」又は「BB」に対応する役が決定されている場合、ART中CT抽籤テーブルを用いたCT抽籤処理では、確率分母が「256」となる範囲の乱数値を用いたCT抽籤が行われる。また、内部当籤役としてこれらの役以外の内部当籤役（例えば、サブフラグ「リプレイ」、「共通ベル」、「押し順ベル」などに対応する役）が決定されている場合には、ART中CT抽籤テーブルを用いたCT抽籤処理において、確率分母が「65536」となる範囲の乱数値を用いたCT抽籤が行われる。

【0347】

なお、本実施形態のパチスロ1では、CTとして「通常CT」及び「高確率CT」と称する2種類のCTを設ける。通常CTと高確率CTとでは、CT（上乗せチャンスゾーン）中に上乗せされるARTゲーム数の期待度が互いに異なり、高確率CTは、通常CTに比べて多くのARTゲーム数が上乗せされ易いCTである（後述の図55参照）。

【0348】

〔通常ART中上乗せ抽籤テーブル〕

図51は、通常ART中に行われるARTゲーム数の上乗せ抽籤で用いられる通常ART中上乗せ抽籤テーブルの構成図である。なお、図51に示す内部当籤役の名称は、上述したサブフラグの名称に対応する。

【0349】

通常ART中上乗せ抽籤テーブルは、内部当籤役と、上乗せ抽籤の各種抽籤結果（非当籤／上乗せ10G～300G）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。

【0350】

10

20

30

40

50

< C T 中の遊技性 >

次に、図 5 2 A ~ 5 2 C を参照して、C T 中の遊技の流れについて説明する。なお、図 5 2 A 及び 5 2 B は、主に、サブフラグ E X 「3 連チリリプ」当籤時における C T 中の遊技フローの概要を示す図であり、図 5 2 C は、C T 中に行われるフラグ変換処理の概要を示す図である。

【 0 3 5 1 】

〔 C T 中の遊技内容 〕

本実施形態のパチスロ 1 において、C T では、1 セット 8 回（8 ゲーム）の遊技が行われる。C T 期間中には、毎ゲーム、内部当籤役に基づいて、A R T ゲーム数の上乗せ抽籤が行われる。そして、その上乗せ抽籤に当籤した場合には、C T 遊技の単位遊技数（ゲーム数）の減算は行われず、一方、上乗せ抽籤が非当籤であった場合には、C T 遊技の単位遊技数（ゲーム数）の減算が行われる。それゆえ、C T 期間中において、A R T ゲーム数が上乗せされた遊技では、C T が終了することではなく、同一のセット内で A R T ゲーム数が上乗せされない遊技が 8 回実施されると、C T が終了する。

【 0 3 5 2 】

また、本実施形態では、図 5 2 A 及び 5 2 B に示すように、C T 期間中にサブフラグ E X 「3 連チリリプ」が当籤した場合、すなわち、内部当籤役「F __ 確チリリプ」又は「F __ 1 確チリリプ」が当籤し、かつ、フラグ変換抽籤に当籤した場合、1 セット 8 回の C T 遊技が再セット（ストック）される。そして、この再セット（ストック）された C T 遊技のセットは、当該 C T 遊技のセットが終了した後に開始される。

【 0 3 5 3 】

例えば、同一セット内で A R T ゲーム数の上乗せ抽籤に非当籤である単位遊技が 7 回行われた後、A R T ゲーム数が上乗せされない C T 遊技が 1 回行われると C T が終了するが、このゲームにおいてサブフラグ E X 「3 連チリリプ」が当籤していると、C T 遊技の再セットが行われる。その結果、C T 遊技が再セットされた後、A R T ゲーム数の上乗せ抽籤に非当籤である単位遊技が 8 回行われるまで C T が終了しないことになる。それゆえ、C T の遊技期間は、サブフラグ E X 「3 連チリリプ」が当籤するほど長くなる。

【 0 3 5 4 】

〔 C T 中のフラグ変換 〕

次に、図 5 2 C を参照して、C T 中に行われるフラグ変換抽籤の手法について説明する。上述のように、本実施形態では、C T 期間中にサブフラグ E X 「3 連チリリプ」が当籤すると（内部当籤役「F __ 確チリリプ」又は「F __ 1 確チリリプ」が当籤し、かつ、フラグ変換抽籤に当籤すると）、C T が再セット（ストック）される。また、後述の図 5 4 の C T 中フラグ変換抽籤テーブルに示すように、C T 中に内部当籤役「F __ 確チリリプ」又は「F __ 1 確チリリプ」が当籤すると、フラグ変換抽籤に必ず当籤する（サブフラグ E X 「3 連チリリプ」に必ず変換される）。すなわち、本実施形態では、C T 中において、内部当籤役「F __ 確チリリプ」又は「F __ 1 確チリリプ」が当籤すると、C T が必ず再セットされる。

【 0 3 5 5 】

また、内部当籤役「F __ リーチ目リプ A」～「F __ リーチ目リプ D」のいずれかが当籤した場合におけるフラグ変換抽籤では、3 種類のフラグ変換テーブル（テーブル 0 ~ 2）に基づいて、フラグ変換抽籤の当籤確率が制御される。具体的には、図 5 2 C に示すように、テーブル 0 はサブフラグ E X 「リーチ目リプ」に変換される確率が最も低いフラグ変換テーブルであり、テーブル 1 はサブフラグ E X 「リーチ目リプ」に変換される確率が次に低いフラグ変換テーブルであり、テーブル 2 はサブフラグ E X 「リーチ目リプ」に変換される確率が最も高いフラグ変換テーブルである。なお、C T 中にサブフラグ E X 「リーチ目リプ」に当籤すると、後述の図 5 6 の C T 中セット数上乗せ抽籤テーブルに示すように、C T が新たに付与される。

【 0 3 5 6 】

また、本実施形態において、通常 C T では、図 5 2 C に示すように、A R T レベルに基づ

10

20

30

40

50

いてフラグ変換テーブルが決定される。一方、高確率ＣＴでは、ＡＲＴレベルに関係なく、フラグ変換テーブルとして、テーブル０が必ず決定される。

【０３５７】

<ＣＴ中に用いる各種データテーブル>

次に、図５３～図５６を参照して、ＣＴ中に行われる抽籤処理で用いられる各種データテーブルについて説明する。なお、以下に説明する各種データテーブルは、メインＲＯＭ１０２に格納される。

【０３５８】

〔ＣＴ中テーブル抽籤テーブル〕

図５３は、３段階のフラグ変換テーブル（テーブル０～２）の中からフラグ変換抽籤に用いるテーブルを決定する際に用いられるＣＴ中テーブル抽籤テーブルの構成図である。

10

【０３５９】

ＣＴ中テーブル抽籤テーブルは、ＡＲＴレベルやこれから実行するＣＴの種別などの各状態と、フラグ変換テーブル（テーブル０～２）の種別と、各種別に対応付けられた抽籤値の情報との対応関係を規定する。なお、ＣＴ中テーブル抽籤テーブルは、ＣＴ抽籤に当籤してＣＴに移行することが決定された時、又は、ＣＴの開始時に参照される。

【０３６０】

〔ＣＴ中フラグ変換抽籤テーブル〕

図５４は、ＣＴ中に行われるフラグ変換抽籤で用いられるＣＴ中フラグ変換抽籤テーブルの構成図である。

20

【０３６１】

ＣＴ中フラグ変換抽籤テーブルは、内部当籤役（「Ｆ__確チリリプ」、「Ｆ__１確チリリプ」及び「Ｆ__リーチ目リリプＡ」～「Ｆ__リーチ目リリプＤ」のいずれか）と、各フラグ変換テーブル（テーブル０～２）におけるフラグ変換抽籤の抽籤結果（変換無し／変換有り）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。ＣＴ中フラグ変換抽籤テーブルから明らかなように、ＣＴ中に内部当籤役「Ｆ__確チリリプ」又は「Ｆ__１確チリリプ」が当籤すると、フラグ変換抽籤に必ず当籤する（サブフラグＥＸ「３連チリリプ」に必ず変換される）。

【０３６２】

〔ＣＴ中上乗せ抽籤テーブル〕

図５５は、ＣＴ中に行われるＡＲＴゲーム数の上乗せ抽籤で用いられるＣＴ中上乗せ抽籤テーブルの構成図である。

30

【０３６３】

ＣＴ中上乗せ抽籤テーブルは、現在のＣＴ状態と内部当籤役との各組合せと、上乗せ抽籤の各種抽籤結果（非当籤／上乗せ１０ゲーム／…／上乗せ３００ゲーム）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。なお、図５５に示す内部当籤役の名称は、上述したサブフラグの名称に対応し、図５５に示す内部当籤役（サブフラグ）以外の役が内部当籤した場合には、ＣＴ中の上乗せ抽籤に当籤することはない。

【０３６４】

また、本実施形態の通常ＣＴ中における上乗せ抽籤では、サブフラグ「３連チリリプ」（内部当籤役「Ｆ__確チリリプ」又は「Ｆ__１確チリリプ」）の当籤回数に応じて上乗せゲーム数の付与形態が変化する。

40

【０３６５】

具体的には、同一のＣＴセット中におけるサブフラグ「３連チリリプ」の当籤回数が１～８回である場合には、図５５中に示す抽籤結果「上乗せ__１０Ｇ」及び「上乗せ__２０Ｇ」で付与される上乗せゲーム数はそれぞれ１０ゲーム及び２０ゲームとなる。それゆえ、この場合には、ＡＲＴの上乗せゲーム数として１０ゲームが決定され易くなる。また、同一のＣＴセット中におけるサブフラグ「３連チリリプ」の当籤回数が９～１６回である場合には、図５５中に示す抽籤結果「上乗せ__１０Ｇ」及び「上乗せ__２０Ｇ」で付与される上乗せゲーム数はともに２０ゲームとなる。すなわち、図５５中のサブフラグ「３連チ

50

リリプ」の抽籤値「極高」に対応する上乗せゲーム数（抽籤結果）が20ゲームに昇格する。それゆえ、この場合には、ARTの上乗せゲーム数として20ゲームが決定され易くなる。

【0366】

また、同一のCTセット中におけるサブフラグ「3連チリリプ」の当籤回数が17～24回である場合には、図55中に示す抽籤結果「上乗せ__10G」及び「上乗せ__20G」で付与される上乗せゲーム数はともに30ゲームとなる。すなわち、図55中のサブフラグ「3連チリリプ」の抽籤値「極高」及び「極低」に対応する上乗せゲーム数（抽籤結果）が30ゲームに昇格する。それゆえ、この場合には、ARTの上乗せゲーム数として「30ゲーム」が決定され易くなる。さらに、同一のCTセット中におけるサブフラグ「3連チリリプ」の当籤回数が25回以上である場合には、図55中に示す抽籤結果「上乗せ__10G」及び「上乗せ__20G」で付与される上乗せゲーム数はともに50ゲームとなる。すなわち、図55中のサブフラグ「3連チリリプ」の抽籤値「極高」及び「極低」に対応する上乗せゲーム数（抽籤結果）が50ゲームに昇格する。それゆえ、この場合には、ARTの上乗せゲーム数として「50ゲーム」が決定され易くなる。

【0367】

上述のように、本実施形態のパチスロ1では、CT中のサブフラグ「3連チリリプ」の当籤回数に応じて1回の上乗せ抽籤により上乗せできるARTゲーム数を増やすことが可能になる。また、上述のように、本実施形態では、ARTゲーム数の上乗せが行われている限り、CTは終了することなく、さらに、サブフラグEX「3連チリリプ」に当籤するとCTの再セット（ストック）が行われる。それゆえ、本実施形態では、遊技者に対して、CTが継続するほど、1ゲーム当りの上乗せ量の増加に対する期待を抱かせることができ、CT中の興趣を向上させることができる。また、1ゲーム当りの上乗せ量を増やす契機となるサブフラグ「3連チリリプ」の当籤回数は、CT1セット分の基本遊技回数（8回）よりも多い回数（9回以上）であるため、遊技者に対して過大な利益を与えてしまうことを防止でき、遊技者及び遊技店間において、利益のバランスをとることができる。

【0368】

なお、本実施形態では、上述したサブフラグ「3連チリリプ」（内部当籤役「F__確チリリプ」又は「F__1確チリリプ」）の当籤回数は、同一のCTセット中において計数された回数とするが、本発明はこれに限定されない。例えば、CT中に行われるセット数上乗せ抽籤に当籤した場合に付与される新たなCTも「同一のCTセット中」に含めるようにしてもよい。

【0369】

【CT中セット数上乗せ抽籤テーブル】

図56は、CT中に行われるCTセットの上乗せ抽籤で用いられるCT中セット数上乗せ抽籤テーブルの構成図である。

【0370】

CT中セット数上乗せ抽籤テーブルは、現在のCT状態と内部当籤役（サブフラグ「リーチ目リプ（リーチ目リプ1～4）」）との各組合せと、上乗せ抽籤の各種抽籤結果（非当籤／通常CT当籤／高確率CT当籤）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。

【0371】

CT中セット数上乗せ抽籤テーブルから明らかなように、CT中に内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」のいずれかに当籤すると、CTセットの上乗せ抽籤に必ず当籤する（CTのセットが必ずストックされる）。なお、ストックされたCTのセットは、現在作動中のCTのセットが終了した後に開始される。

【0372】

<ボーナス状態中の遊技性>

次に、図57A～57Cを参照して、ボーナス状態中の遊技の流れについて説明する。図57Aは、一般遊技状態（ART非当籤）中に遊技状態がボーナス状態に移行した場合に

10

20

30

40

50

における遊技の流れを示す図であり、図 5 7 B は、通常 A R T 中に遊技状態がボーナス状態に移行した場合における遊技の流れを示す図であり、図 5 7 C は、C T 中に遊技状態がボーナス状態に移行した場合における遊技の流れを示す図である。

【0373】

なお、本実施形態のパチスロ 1 では、図 5 7 A ～ 5 7 C に示すように、遊技性の面において、ボーナスの種別として通常 B B と特殊 B B とを設け、ボーナス状態への移行時にこのボーナスの種別が決定される。この際、特殊 B B が決定された場合には、ボーナス状態の終了後、遊技状態は A R T 準備状態を経由して C T に移行する。一方、通常 B B が決定された場合には、移行先の遊技状態は、ボーナス状態に移行する前の状態に応じて異なる。

【0374】

遊技状態が一般遊技状態から通常 B B に移行した場合、図 5 7 A に示すように、通常 B B 中の遊技では、内部当籤役に基づいて A R T 抽籤が行われる。そして、この A R T 抽籤に当籤すると、ボーナス状態の終了後、遊技状態が A R T 準備状態を経由して通常 A R T に移行する。なお、この場合、A R T 抽籤に当籤した後のボーナス状態中の遊技では、A R T ゲーム数の上乗せ抽籤が行われる。

【0375】

遊技状態が通常 A R T から通常 B B に移行した場合、図 5 7 B に示すように、通常 B B の終了時に C T 抽籤が行われる。この C T 抽籤の当籤確率は 5 0 % であり、当籤するとボーナス状態の終了後に、遊技状態は A R T 準備状態を経由して C T に移行する。一方、C T 抽籤に非当籤である場合には、ボーナス状態の終了後に、遊技状態は A R T 準備状態を経由して通常 A R T に移行する。なお、通常 A R T から移行したボーナス状態中の遊技では、A R T ゲーム数の上乗せ抽籤も行われる。

【0376】

遊技状態が C T から通常 B B 又は特殊 B B に移行した場合、図 5 7 C に示すように、ボーナス状態の終了後に、遊技状態は A R T 準備状態を経由して C T に移行する。なお、C T から移行したボーナス状態中の遊技では、A R T ゲーム数の上乗せ抽籤も行われる。

【0377】

<ボーナス状態中の遊技で用いる各種データテーブル>

続いて、図 5 8 ～ 図 6 0 を参照して、ボーナス状態中の遊技で行われる抽籤処理で用いられる各種データテーブルについて説明する。なお、以下に説明する各種データテーブルは、メイン R O M 1 0 2 に格納される。

【0378】

[ボーナス種別抽籤テーブル]

図 5 8 は、ボーナス種別（通常 B B、特殊 B B）を決定する際に用いられるボーナス種別抽籤テーブルの構成図である。

【0379】

ボーナス種別抽籤テーブルは、ボーナス状態に移行する前の各遊技状態（C T 及びそれ以外）と、各種抽籤結果（ボーナス種別：通常 B B ／ 特殊 B B）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。なお、ボーナス種別抽籤テーブルを参照したボーナス種別の決定処理は、ボーナス状態の開始時に行われる。

【0380】

[ボーナス中 A R T ゲーム数上乗せ抽籤テーブル]

図 5 9 は、ボーナス状態中の遊技で行われる A R T 抽籤及び A R T ゲーム数の上乗せ抽籤で用いられるボーナス中 A R T ゲーム数上乗せ抽籤テーブルの構成図である。

【0381】

ボーナス中 A R T ゲーム数上乗せ抽籤テーブルは、現在のボーナス種別と内部当籤役との各組合せと、上乗せ抽籤の各種抽籤結果（非当籤／ 5 ゲーム／ … ／ 3 0 0 ゲーム）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。

【0382】

本実施形態では、A R T 非当籤の状態（一般遊技状態から移行した通常 B B において、A

10

20

30

40

50

R T抽籤に当籤するまでの状態)では、ボーナス中A R Tゲーム数上乘せ抽籤テーブルは、A R T抽籤に用いられる。具体的には、A R T非当籤の状態において、ボーナス中A R Tゲーム数上乘せ抽籤テーブルを用いた抽籤により1ゲーム以上(図59に示す例では50ゲーム以上)の上乗せゲーム数が決定されると、A R T抽籤に当籤するとともに、対応するゲーム数がA R Tゲーム数として付与される。一方、A R T当籤後の状態では、ボーナス中A R Tゲーム数上乘せ抽籤テーブルは、A R Tゲーム数の上乗せ抽籤のみに用いられる。

【0383】

[ボーナス終了時C T抽籤テーブル]

図60は、ボーナス状態の終了時に行われるC T抽籤で用いられるボーナス終了時C T抽籤テーブルの構成図である。

10

【0384】

ボーナス終了時C T抽籤テーブルは、ボーナス種別(通常B B、特殊B B)とボーナス状態に移行する前の遊技状態(通常C T中、高確率C T中)との各組合せと、C T抽籤の各種抽籤結果(非当籤/通常C T当籤/高確率C T当籤)と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。ボーナス終了時C T抽籤テーブルから明らかなように、例えば、通常A R T中に通常B Bが行われると、ボーナス状態の終了時に50%の確率でC Tに当籤する。

【0385】

<一般遊技状態中の例外的な遊技性>

20

次に、図61を参照して、一般遊技状態中の例外的な遊技の流れについて説明する。

【0386】

本実施形態のパチスロ1における基本的な遊技状態の流れでは、一般遊技状態中に遊技状態が通常遊技状態からC Zに移行し、C ZにおいてA R T抽籤に当籤することにより遊技状態がA R T遊技状態に移行する。そして、本実施形態では、R T 4状態において報知を行うことによりA R T遊技状態を実現している。また、本実施形態では、図61に示すように、停止表示される図柄組合せに応じてR T状態の移行制御を行う。

【0387】

なお、R T状態を移行させるための図柄組合せは、遊技者の停止操作の順序(押し順)に応じて停止表示されるもの(図24参照)であるので、報知が行われない場合であっても偶然、R T状態がR T 4状態に移行することもある。また、R T 4状態では、内部当籤役「F__リーチ目リブA」～「F__リーチ目リブD」のいずれかが決定される可能性があるため、一般遊技状態(非A R T)中であっても、特別な特典が付与されるリーチ目(略称「リーチ目リブ」に係る図柄組合せ)を表示することができる。

30

【0388】

そこで、本実施形態のパチスロ1では、図61に示すように、一般遊技状態(非A R T)中に偶然、R T状態がR T 4状態に移行し、略称「リーチ目リブ」に係る図柄組合せが表示可能な状態になると、C Zを経由することなく、遊技状態をA R T遊技状態(通常A R T)に移行可能にする。

【0389】

より具体的には、一般遊技状態で、かつ、R T 4状態中に内部当籤役「F__リーチ目リブA」～「F__リーチ目リブD」のいずれかが決定されると、フラグ変換抽籤を行い、このフラグ変換抽籤に当籤すると、略称「リーチ目リブ」に係る図柄組合せを表示するための報知(ナビ)が行われるとともに、A R Tの権利が付与される。一方、一般遊技状態で、かつ、R T 4状態中に内部当籤役「F__リーチ目リブA」～「F__リーチ目リブD」のいずれかが決定された場合において、フラグ変換抽籤に非当籤となると、略称「リブレイ」に係る図柄組合せを表示するための報知が行われ、略称「リーチ目リブ」に係る図柄組合せが表示されないような制御が行われる。

40

【0390】

<一般遊技状態中の例外的な遊技制御で用いる各種データテーブル>

50

次に、図 6 2 を参照して、上述した一般遊技状態中の例外的な遊技制御で行われる抽籤処理で用いるデータテーブルについて説明する。なお、以下に説明するデータテーブルは、メイン ROM 1 0 2 に格納される。

【 0 3 9 1 】

〔非 ART 中フラグ変換抽籤テーブル〕

図 6 2 は、一般遊技状態で、かつ、RT 4 状態中の遊技で行われるフラグ変換抽籤で用いられる非 ART 中フラグ変換抽籤テーブルの構成図である。

【 0 3 9 2 】

非 ART 中フラグ変換抽籤テーブルは、内部当籤役（「F__リーチ目リプ A」～「F__リーチ目リプ D」）と、フラグ変換抽籤の抽籤結果（変換無し／変換あり）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。

【 0 3 9 3 】

<メイン側の制御による報知機能>

従来のパチスロでは、ART 中にサブ（副制御基板 7 2）側の制御により、リールの停止操作の情報（押し順など）の報知（ナビ）を行っていた。しかしながら、この報知の有無が遊技者の利益（いわゆる、出玉）に影響を与えるため、近年では、遊技者の利益を管理するメイン（主制御基板 7 1）側で報知を行うことが求められている。そこで、本実施形態のパチスロ 1 では、上述のように、メイン側で制御される情報表示器 6 に停止操作の情報を報知するための指示モニタ（不図示）を設け、メイン側の制御により、リールの停止操作の情報を報知する機能が設けられている。

【 0 3 9 4 】

ここで、図 6 3 A ～ 6 3 D に、本実施形態のパチスロ 1 において、メイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係を示す。なお、図 6 3 A は、ART 準備状態における、メイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係を示す図であり、図 6 3 B は、ART（通常 ART 又は CT）中における、メイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係を示す図である。また、図 6 3 C は、RT 5 状態中（BB 1 フラグ間）における、メイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係を示す図であり、図 6 3 D は、RT 5 状態中（BB 2 フラグ間）における、メイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係を示す図である。

【 0 3 9 5 】

本実施形態では、図 6 3 A ～ 6 3 D に示すように、メイン（主制御基板 7 1）側では、指示モニタに「1」～「11」の数値を表示することにより、リールの停止操作の情報を報知する。なお、指示モニタに表示されるこの「1」～「11」の数値は、それぞれが報知する停止操作の内容に一義的に対応している。

【 0 3 9 6 】

具体的には、数値「1」～「3」はそれぞれ、第 1 停止操作を行うリールの種別を示しており、数値「1」は第 1 停止操作を左リール 3 L に対して行うことを意味し、数値「2」は第 1 停止操作を中リール 3 C に対して行うことを意味し、数値「3」は第 1 停止操作を右リール 3 R に対して行うことを意味する。

【 0 3 9 7 】

また、数値「4」～「9」はそれぞれ、報知する押し順を示しており、数値「4」は押し順が「左、中、右」の順であることを意味し、数値「5」は押し順が「左、右、中」の順であることを意味し、数値「6」は押し順が「中、左、右」の順であることを意味し、数値「7」は押し順が「中、右、左」の順であることを意味し、数値「8」は押し順が「右、左、中」の順であることを意味し、数値「9」は押し順が「右、中、左」の順であることを意味する。

【 0 3 9 8 】

また、数値「10」及び「11」はそれぞれ、ボーナス役を報知するものであり、数値「10」は、コンビネーション名称「C__BB 1」に係る図柄組合せ（図柄「白 7」－図柄

10

20

30

40

50

「白 7」－図柄「白 7」)を意味し、数値「1 1」は、コンビネーション名称「C__B B 2」に係る図柄組合せ(図柄「青 7」－図柄「青 7」－図柄「青 7」)を意味する。

【0 3 9 9】

なお、メイン側(指示モニタ)で報知する数値「1」～「1 1」は、報知する停止操作の内容に一義的に対応しているものの、全ての遊技者が、その数値に基づいて、明確に報知内容を把握できるとは限らない。例えば、メイン側で指示モニタに数値「6」を表示しただけでは、遊技者によっては報知内容を把握できない可能性もある。

【0 4 0 0】

そこで、本実施形態のパチスロ 1 では、メイン側の報知と併せてサブ側でもストップボタンの停止操作に係る情報を報知する。具体的には、サブ側で制御される表示装置 1 1 (プロジェクタ機構 2 1 1 及び表示ユニット 2 1 2)を用いて、サブ側の制御により停止操作に係る情報の報知を行う。

【0 4 0 1】

例えば、第 1 停止操作を左リール 3 L に対して行う押し順を報知する場合、メイン側で指示モニタに数値「1」を表示するとともに、サブ側では、表示装置 1 1 の表示画面内の左リール 3 L の上方に数値「1」と表示し、左リール 3 L が第 1 停止操作の対象であることを報知する。また、押し順「中、左、右」を報知する場合、メイン側で指示モニタに数値「6」を表示するとともに、サブ側では、表示装置 1 1 の表示画面内の中リール 3 C の上方に数値「1」を表示し、左リール 3 L の上方に数値「2」を表示し、右リール 3 R の上方に数値「3」を表示し、この表示により押し順が「中、左、右」の順であることを報知する。また、内部当籤役「F__B B 1」が決定されている場合、メイン側で指示モニタに数値「1 0」を表示するとともに、サブ側では、表示装置 1 1 の表示画面に「白 7」－「白 7」－「白 7」の図柄組合せに関する情報を表示し、遊技者に対して狙うべき図柄を報知する。

【0 4 0 2】

なお、メイン側で報知を行うタイミングは、少なくとも報知を行う一遊技の期間であれば任意のタイミングに設定することができる。例えば、遊技者の開始操作を検知した(受け付けた)タイミングでメイン側の報知を行ってもよいし、リールの回転開始時にメイン側の報知を行ってもよいし、第 1 停止操作～第 3 停止操作のいずれかを検知したタイミングでメイン側の報知を行ってもよい。一方、サブ側で報知を行うタイミングは、少なくとも第 1 停止操作よりも前のタイミングであることが好ましい。それゆえ、本実施形態のパチスロ 1 では、開始操作を検知したタイミング、又は、リールの回転開始時のタイミングで、メイン側及びサブ側の双方において報知(ナビ)を行う。これにより、遊技者が停止操作を行う前に、メイン側の指示モニタ及びサブ側の表示装置 1 1 の双方において停止操作の情報が報知される。

【0 4 0 3】

A R T 準備状態では、図 6 3 A に示すように、メイン側の制御により、「ベルナビ」、「維持リブナビ」、「R T 3 移行リブナビ」及び「R T 4 移行リブナビ」と称する報知(ナビ)が行われる。「ベルナビ」では、内部当籤役「F__3 択ベル__1 s t」～「F__3 択ベル__3 r d」が決定された際に、略称「ベル」に係る図柄組合せ(図 2 9 参照)を有効ライン上に停止表示させるための押し順が報知される。「維持リブナビ」では、内部当籤役「F__維持リブ__1 s t」～「F__維持リブ__3 r d」が決定された際に、略称「リブレイ」に係る図柄組合せ(図 2 8 参照)を有効ライン上に停止表示させるための押し順が報知される。「R T 3 移行リブナビ」では、内部当籤役「F__R T 3 移行リブ__1 s t」～「F__R T 3 移行リブ__3 r d」が決定された際に、略称「R T 3 移行リブ」に係る図柄組合せ(図 2 8 参照)を有効ライン上に停止表示させるための押し順が報知される。また、「R T 4 移行リブナビ」では、内部当籤役「F__R T 4 移行リブ__1 2 3」～「F__R T 4 移行リブ__3 r d」が決定された際に、略称「R T 4 移行リブ」に係る図柄組合せ(図 2 8 参照)を有効ライン上に停止表示させるための押し順が報知される。

【0 4 0 4】

また、ART遊技状態（通常ART又はCT）中では、図63Bに示すように、メイン側の制御により、「ベルナビ」、「維持リブナビ」、「RT3移行リブナビ」及び「RT4移行リブナビ」と称する報知（ナビ）が行われる。なお、ART遊技状態（RT4状態）中の遊技は、フラグ変換抽籤が行われ、この抽籤結果に基づいて略称「3連チリリブ」、「リーチ目リブ」又は「リプレイ」に係る図柄組合せを表示させるための押し順が報知されるが、この報知は、サブ側のみで行われ、メイン側では行われない。

【0405】

上述のように、略称「3連チリリブ」又は「リーチ目リブ」に係る図柄組合せは、特別な特典の付与に関係しているため、報知の有無が遊技者の利益（出玉）に影響を与えるように見えるが、実際には、本実施形態のパチスロ1では、特別な特典は、フラグ変換抽籤の抽籤結果に基づいて付与されるものであるため、表示される図柄組合せは付与する特典に対して影響を与えない。それゆえ、例えば、フラグ変換抽籤に当籤している状態において、仮に、略称「リプレイ」に係る図柄組合せが停止表示されてしまっても、特別な特典が付与される。一方、フラグ変換抽籤に当籤していない状態において、仮に、略称「3連チリリブ」又は「リーチ目リブ」に係る図柄組合せを停止表示できたとしても、特別な特典は付与されない。本実施形態のパチスロ1では、このように表示される図柄組合せが遊技者の利益（出玉）に影響を与えない場合、メイン側の指示モニタでの報知を行わずに、サブ側で制御される表示装置11でのみ報知を行う。

【0406】

また、RT5状態（フラグ間状態）中では、図63C及び63Dに示すように、内部当籤役として持ち越されているボーナス役に係る図柄組合せを遊技者に狙わせる旨の情報が報知される。例えば、内部当籤役「F__BB1」が持ち越されている場合には、図63Cに示すように、メイン側の制御により、「白7ナビ」と称する報知（ナビ）が行われ、内部当籤役「F__BB2」が持ち越されている場合には、図63Dに示すように、メイン側の制御により、「青7ナビ」と称する報知（ナビ）が行われる。

【0407】

「白7ナビ」では、内部当籤役「F__BB1」に対応する図柄組合せ、すなわち、コンビネーション名称「C__BB1」に係る図柄組合せ（「白7」－「白7」－「白7」：図28参照）を有効ライン上に停止表示させるための停止操作の情報が報知される。また、「青7ナビ」では、内部当籤役「F__BB2」に対応する図柄組合せ、すなわち、コンビネーション名称「C__BB2」に係る図柄組合せ（「青7」－「青7」－「青7」：図28参照）を有効ライン上に停止表示させるための停止操作の情報が報知される。

【0408】

フラグ間状態において、ボーナス役と、内部当籤役「はずれ」、「F__特殊1」、「F__特殊2」及び「F__特殊3」のいずれかが決定されている場合、図24で説明したように、ボーナス役（BB役）に係る図柄組合せを有効ライン上に停止表示することができる。しかしながら、内部当籤役「はずれ」、「F__特殊1」、「F__特殊2」及び「F__特殊3」以外の内部当籤役とボーナス役とが当籤している場合には、ボーナス役に係る図柄組合せを有効ライン上に停止表示することができない。それゆえ、本実施形態では、図63C及び63Dに示すように、持ち越されているボーナス役と、内部当籤役「はずれ」、「F__特殊役1」、「F__特殊役2」及び「F__特殊役3」のいずれかが当籤としている場合に限り、メイン側の制御による「白7ナビ」又は「青7ナビ」と称する報知が行われる。それゆえ、本実施形態では、メイン側の制御による「白7ナビ」又は「青7ナビ」と称する報知（ナビ）を、ボーナス役を入賞させることのできる適切なタイミングで行うことができる。

【0409】

なお、本実施形態のパチスロ1には、例えば、ボーナス確定画面を表示することや、ボーナス確定ランプを点灯させることなどにより、ボーナス告知を行う機能も設けられている。そこで、メイン側では、ボーナス役が内部当籤役として決定されていることを告知（ボーナス告知）した後にのみ、「白7ナビ」又は「青7ナビ」と称するナビを行うようにし

10

20

30

40

50

てもよい。

【0410】

ボーナス告知としては、例えば、複数回の遊技期間に渡って行われる演出（いわゆる連続演出）を行い、この連続演出の結果に応じてボーナス確定画面を表示するような演出が一般的に行われている。このような連続演出の最中にメイン側で「白7ナビ」などを行うと、連続演出の結果が途中で分かってしまうので、興趣を損ねてしまう可能性がある。そこで、本実施形態では、主制御基板71は、ボーナス告知が行われた後に、メイン側の制御による「白7ナビ」又は「青7ナビ」と称する報知（ナビ）が行われる。

【0411】

なお、ボーナス告知が行われたタイミングをメイン側で把握可能にする手法は任意である。その一手法として、ボーナス役が内部当籤役として決定されると、主制御基板71がボーナス告知終了までに要するゲーム数を決定し、このゲーム数の遊技を消化した後に、「白7ナビ」又は「青7ナビ」と称する報知（ナビ）を行う手法が考えられる。より具体的には、主制御基板71は、ボーナス告知終了までに要するゲーム数を決定すると、このゲーム数を副制御基板72に通知する。副制御基板72は、このゲーム数に従い演出の制御を行い、該ゲーム数の遊技が消化されたタイミングでボーナス確定画面を表示することにより、メイン側においてボーナス告知が行われたタイミングを把握することができる。すなわち、主制御基板71は、ボーナス役が持ち越されていない状態でボーナス役を内部当籤役として決定してからの単位遊技の回数を計数し、その計数結果が所定回数に達した後、ボーナス役と、内部当籤役「はずれ」、「F__特殊役1」、「F__特殊役2」及び「F__特殊役3」のいずれかが内部当籤役として決定された場合に、「白7ナビ」又は「青7ナビ」を行う。

【0412】

また、他の手法としては、ボーナス告知をサブ側ではなくメイン側において制御する手法が考えられる。より具体的には、主制御基板71は、ボーナス役が持ち越されていない状態でボーナス役を内部当籤役として決定すると、表示装置11で実行する演出（少なくとも演出に要するゲーム数）を決定し、副制御基板72に通知する。副制御基板72が通知された演出を実行し、ボーナス確定画面を表示することにより、メイン側においてボーナス告知が行われたタイミングを把握することができる。

【0413】

なお上述した2つの手法以外の他の手法によりボーナス告知が行われたタイミングをメイン側で把握可能にする構成にしてもよい。この場合、主制御基板71は、副制御基板72などからの信号を受け付けることができないため、主制御基板71が受け付け可能な信号に基づいてボーナス告知が行われたタイミングを把握する必要がある。例えば、停止操作に伴う信号は、主制御基板71が受け付け可能であるため、ボーナス役が内部当籤役として決定されている状態で、所定の停止操作（例えば、順押し以外）が行われた場合に、ボーナス告知を行う手法も考えられる。具体的には、副制御基板72は、主制御基板71から内部当籤役に関する情報と停止操作に関する情報とを取得し、これらの情報の組合せが所定の組合せである場合にボーナス告知を行う。このようなボーナス告知の手法を採用することにより、ボーナス告知の契機を主制御基板71でも把握することができるため、メイン側においてボーナス告知が行われたタイミングを把握することができる。

【0414】

<主制御回路の動作説明>

次に、図64～図170を参照して、主制御回路90のメインCPU101が、プログラムを用いて実行する各種処理の内容について説明する。

【0415】

[電源投入（リセット割込）時処理]

まず、メインCPU101の制御により行われるパチスロ1の電源投入（リセット割込）時処理を、図64及び図65を参照して説明する。図64は、電源投入（リセット割込）時処理の手順を示すフローチャートであり、図65A～65Cは、それぞれ、該フローチ

ヤート中の S 2、S 7 及び S 8、並びに、S 1 3 の処理を実行するためのソースプログラムの一例を示す図である。なお、図 6 4 に示す電源投入（リセット割込）時処理は、電源管理回路 9 3 が、マイクロプロセッサ 9 1 に電源電圧の供給が開始されたことを検知した際に、リセット信号をマイクロプロセッサ 9 1 の「X S R S T」端子に出力し、それにより、マイクロプロセッサ 9 1 の割込みコントローラ 1 1 2 からメイン CPU 1 0 1 に出力される割込要求信号に基づいて、実行される。

【0 4 1 6】

まず、メイン CPU 1 0 1 は、電源監視ポート（電源監視手段）がオン状態であるか否かを判別する（S 1）。

【0 4 1 7】

S 1 において、メイン CPU 1 0 1 が、電源監視ポートがオン状態であると判別したとき（S 1 が Y E S 判定の場合）、メイン CPU 1 0 1 は、S 1 の処理を繰り返す。なお、ここでいう、電源監視ポートがオン状態とは、メイン CPU 1 0 1 に供給されている電源電圧（D C + 5 V）が安定していない状態のことである。

【0 4 1 8】

一方、S 1 において、メイン CPU 1 0 1 が、電源監視ポートがオン状態でないと判別したとき（S 1 が N O 判定の場合）、メイン CPU 1 0 1 は、タイマー回路 1 1 3（P T C）の初期化処理を行う（S 2）。この処理では、メイン CPU 1 0 1 は、タイマー回路 1 1 3 の初期設定を行う。具体的には、メイン CPU 1 0 1 は、タイマー用プリスケアラレジスタ（不図示）に分周比をセットし、タイマー用制御レジスタ（不図示）に割り込み可等の設定を行い、タイマー用カウンタ（不図示）の初期カウント値を設定する。

【0 4 1 9】

次いで、メイン CPU 1 0 1 は、主制御回路 9 0 及び副制御回路 2 0 0 間用の第 1 シリアル通信回路 1 1 4（S C U 1）の初期化処理、及び、第 2 インターフェースボード用の第 2 シリアル通信回路 1 1 5（S C U 2）の初期化処理を行う（S 3）。次いで、メイン CPU 1 0 1 は、乱数回路 1 1 0（R D G）の初期化処理を行う（S 4）。次いで、メイン CPU 1 0 1 は、メイン RAM 1 0 3 の書き込みテストを行う（S 5）。

【0 4 2 0】

次いで、メイン CPU 1 0 1 は、書き込みテストの結果、メイン RAM 1 0 3 への書き込みが正常に行われたか否かを判別する（S 6）。

【0 4 2 1】

S 6 において、メイン CPU 1 0 1 が、メイン RAM 1 0 3 への書き込みが正常に行われなかったと判別したとき（S 6 が N O 判定の場合）、メイン CPU 1 0 1 は、後述の S 1 3 の処理を行う。一方、S 6 において、メイン CPU 1 0 1 が、メイン RAM 1 0 3 への書き込みが正常に行われたと判別したとき（S 6 が Y E S 判定の場合）、メイン CPU 1 0 1 は、タイマー回路 1 1 3 のタイマー用制御レジスタ（不図示）の状態を取得する（S 7）。

【0 4 2 2】

次いで、メイン CPU 1 0 1 は、取得したタイマー用制御レジスタの状態に基づいて、現在の状態が割込処理の発生タイミングであるか否かを判別する（S 8）。具体的には、メイン CPU 1 0 1 は、取得したタイマー用制御レジスタの状態に基づいて、タイマーカウンタ開始後から 1 . 1 1 7 2 m s 経過したか否かを判別する。

【0 4 2 3】

なお、本実施形態では、S 2 のタイマー回路 1 1 3 の初期化処理によりタイマー時間 1 . 1 1 7 2 m s がセットされると、CPU 内蔵タイマーのカウント処理が開始される。その後、タイマー用制御レジスタ（不図示）の情報を読み込むことによりタイマー回路 1 1 3 のステータスを取得することができる。そして、本実施形態では、タイマー用制御レジスタに、現在の状態が割込処理の発生タイミングであるか否か（タイマー割込状態であるか否か）を判別（参照）可能なビット（判別ビット）が設けられる。

【0 4 2 4】

10

20

30

40

50

それゆえ、上記 S 7 の処理では、メイン CPU 101 は、タイマー用制御レジスタ（不図示）の情報を読み込み、上記 S 8 の処理では、メイン CPU 101 は、タイマー用制御レジスタ内の判別ビットのオン／オフ状態（「1」／「0」）を参照することにより、現在の状態が割込処理の発生タイミングであるか否かを判別する。なお、タイマー回路 113 によるカウント開始から 1.1172ms 経過したとき（タイマー回路 113 のカウント値が 0 であれば）、該判別ビットはオン状態となる。

【0425】

S 8 において、メイン CPU 101 が、現在の状態が割込処理の発生タイミングでないと判別したとき（S 8 が NO 判定の場合）、メイン CPU 101 は、処理を S 7 の処理に戻し、S 7 以降の処理を繰り返す。

10

【0426】

一方、S 8 において、メイン CPU 101 が、現在の状態が割込処理の発生タイミングであると判別したとき（S 8 が YES 判定の場合）、メイン CPU 101 は、サムチェック処理（規定外）を行う（S 9）。この処理では、メイン CPU 101 は、メイン RAM 103 のサムチェック処理を行うが、この処理の作業は、メイン RAM 103 内の規定外作業領域（図 12C 参照）で行われる。また、このサムチェック処理で用いられるプログラムはメイン ROM 102 内の規定外エリアに格納されている（図 12B 参照）。なお、サムチェック処理の詳細については、後述の図 79 及び図 80 を参照しながら後で説明する。

【0427】

また、S 8 において、メイン CPU 101 が、現在の状態が割込処理の発生タイミングであると判別したとき（S 8 が YES 判定の場合）には、メイン CPU 101 は、S 9 の処理の前に、後述の割込処理（後述の図 158 参照）を実行する。そして、この割込処理により、主制御回路 90（主制御基板 71）から副制御回路 200（副制御基板 72）には、無操作コマンドが送信される。

20

【0428】

S 9 の処理後、メイン CPU 101 は、設定用鍵型スイッチ 54 がオン状態であるか否かを判別する（S 10）。

【0429】

S 10 において、メイン CPU 101 が、設定用鍵型スイッチ 54 がオン状態であると判別したとき（S 10 が YES 判定の場合）、メイン CPU 101 は、後述の S 15 の処理を行う。一方、S 10 において、メイン CPU 101 が、設定用鍵型スイッチ 54 がオン状態でないと判別したとき（S 10 が NO 判定の場合）、メイン CPU 101 は、S 9 のサムチェック処理の結果に基づいて、サムチェック判定結果が正常であったか否かを判別する（S 11）。

30

【0430】

S 11 において、メイン CPU 101 が、サムチェック判定結果が正常でないと判別したとき（S 11 が NO 判定の場合）、メイン CPU 101 は、後述の S 13 の処理を行う。一方、S 11 において、メイン CPU 101 が、サムチェック判定結果が正常であると判別したとき（S 11 が YES 判定の場合）、メイン CPU 101 は、遊技復帰処理を行う（S 12）。この処理では、メイン CPU 101 は、遊技の状態を電断検知前の状態に戻す処理を行う。なお、遊技復帰処理の詳細については、後述の図 66 を参照しながら後で説明する。

40

【0431】

S 6 又は S 11 が NO 判定の場合、メイン CPU 101 は、情報表示器 6（7セグ LED 表示器）に、エラー発生を意味する文字列「rr」を表示する（S 13）。その後、メイン CPU 101 は、WDT のクリア処理を繰り返す（S 14）。

【0432】

ここで再度、S 10 の処理に戻って、S 10 が YES 判定の場合、メイン CPU 101 は、設定変更確認処理を行う（S 15）。この処理では、メイン CPU 101 は、主に、設定変更開始時における設定変更コマンドの生成格納処理を行う。なお、設定変更確認処理

50

の詳細については、後述の図 6 8 を参照しながら後で説明する。

【0433】

次いで、メインCPU101は、RAM初期化処理を行う(S16)。この処理では、メインCPU101は、図12Cに示すメインRAM103の遊技用RAM領域内の「RAM異常時又は設定変更開始時」のアドレスを、初期化開始の先頭アドレスとして設定し、該先頭アドレスから遊技用RAM領域の最終アドレスまでの情報を消去(クリア)する。そして、S16の処理後、メインCPU101は、後述のメイン処理(後述の図82参照)を開始する。

【0434】

本実施形態では、上述のようにして電源投入(リセット割込)時処理が行われる。そして、上述した電源投入(リセット割込)時処理中のS2の処理は、メインCPU101が、図65Aのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【0435】

図65Aのソースプログラムでは、CPUクロックとして10MHz(供給クロックは20MHz)が設定され、プリスケアラレジスタ設定値として228が設定され、初期カウント値として49が設定される。この結果、割り込み処理のタイマー時間(実行周期)として、 $1.1172\text{ms} (= 1 / (10\text{MHz} / 288) \times 49)$ が算出される。

【0436】

また、上述した電源投入(リセット割込)時処理中のS7及びS8の処理は、メインCPU101が、図65Bのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。具体的には、S7のタイマー用制御レジスタの状態の取得処理は、図65B中の「LD」命令により実行され、S8の判定処理は、図65B中の「JBIT」命令により実行される。そして、タイマー回路113によるカウント開始から1.1172ms経過するまで、図65B中の「LD」命令及び図65B中の「JBIT」命令が繰り返し行われ、タイマー回路113によるカウント開始から1.1172ms経過すると、タイマー回路113から割込みコントローラ112を介してメインCPU101に割込要求信号が出力される。メインCPU101は、この割込要求信号の入力を契機として、電源復帰後の最初の1.1172ms周期の割込処理を開始する。

【0437】

なお、この電源復帰直後(電源投入時の初期化後)の最初の1.1172ms周期の割込処理では、遊技動作に関するコマンドはセットされていないので、主制御回路90から副制御回路200には無操作コマンドが送信される。このように電源復帰直後に割込処理を許可することにより、電源復帰後、最短時間で無操作コマンドが送信され、主制御回路90及び副制御回路200間の通信接続を確立することができ、主制御回路90及び副制御回路200間の通信動作を安定化させることができる。

【0438】

また、この通信動作で送信される無操作コマンドを構成する通信パラメータ1~5には、電源復帰時に、それぞれLレジスタ、Hレジスタ、Eレジスタ、Dレジスタ及びCレジスタに格納されているデータがセットされる。それゆえ、本実施形態では、電源復帰後の最初の割込処理で送信される無操作コマンドの通信パラメータ1~5にそれぞれセットされるデータを、電源復帰毎に異ならせる(不定にする)ことができる。すなわち、電源復帰直後(電源投入時の初期化後)の割込処理で送信される無操作コマンドのサム値(BCC)を、電源復帰毎に異ならせることができる。この場合、ゴト等の不正行為を抑制することができる。

【0439】

また、上述した電源投入(リセット割込)時処理中のS13の処理(割込み禁止処理)は、メインCPU101が、図65Cのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。そして、エラーコード「rr」を、情報表示器6内の2桁の7セグLEDに表示する際の制御では、図65Cに示すように、まず、ソースコ

10

20

30

40

50

ード「LD HL, cPA__SEGCOM」が実行されることにより、7セグLEDへの出力のための2バイトのアドレスが指定され、次いで、ソースコード「LDW (HL), 100H*cZCHARAR+cBX__PAYSEG」が実行され、2桁の7セグLEDへの7セグコモン出力データの出力動作と7セグカソード出力データの出力動作とが同時に行われる。

【0440】

すなわち、本実施形態のパチスロ1では、2桁の7セグLEDをダイナミック点灯制御する際に、7セグコモン出力データと、7セグカソード出力データとが同時に出力される。この場合、ソースプログラム上において、7セグLEDのダイナミック点灯制御に必要な命令コード数を減らすことができ、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。それゆえ、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【0441】

なお、ここでいう、「7セグコモン出力データ」は、7セグLEDをダイナミック制御する際に、7セグLEDのコモン（共通）端子に出力されるLED駆動データであり、「7セグカソード出力データ」は、7セグLEDをダイナミック点灯制御する際に、7セグLEDの各カソード端子に出力されるLED駆動データである。

【0442】

また、7セグLEDの各カソード端子及びコモン端子は、ドア中継端子板68及び遊技動作表示基板81を介して主制御基板71に配置されたマイクロプロセッサ91の外部バスインターフェース104に接続され、これにより、7セグLEDがメインCPU101により制御されている。ラベル「cPA__SEGCOM」は、7セグコモン出力データ及び7セグカソード出力データを7セグLEDに出力するための外部インターフェース104のアドレスを定義（指定）している。なお、メインCPU101がパラレルポート111のPOP（図9参照）を使用して外部に信号を出力し、PIPを使用して外部から信号の入力を行う入出力方式は、ポート（I/O）マップドI/O（主に、「OUT」、「IN」等の命令を使用する方式）と呼ばれ、メインCPU101が外部バスインターフェース104を使用して外部と信号の入出力を行う入出力方式は、メモリマップドI/O（「LD」、「LDW」等の命令を使用して、あたかもRAMへのデータの読み書きを行うように、ポートの入出力を行う方式）と呼ばれている。

【0443】

〔遊技復帰処理〕

次に、図66及び図67を参照して、電源投入（リセット割込）時処理（図64参照）中のS12で行う遊技復帰処理について説明する。なお、図66は、遊技復帰処理の手順を示すフローチャートであり、図67は、該フローチャート中のS25～S32の処理を実行するためのソースプログラムの一例を示す図である。

【0444】

まず、メインCPU101は、スタックポインタ（SP）に、電断時のスタックポインタをセットする（S21）。次いで、メインCPU101は、入力ポートの1割込処理前のオンエッジデータ、及び、現在セットされているオンエッジデータをクリア（オフ）する（S22）。次いで、メインCPU101は、出力ポートのバックアップデータを出力ポートにセットする（S23）。次いで、メインCPU101は、入力ポートのデータを読み込み、該データを、入力ポートの現在及び1割込処理前のデータ格納領域に保存する（S24）。

【0445】

次いで、メインCPU101は、回胴制御データ格納領域のアドレスをセットする（S25）。次いで、メインCPU101は、チェックするリール数（本実施形態では「3」）をセットする（S26）。

【0446】

次いで、メインCPU101は、セットされた回胴制御データ格納領域のアドレスに基づいて、所定のリールのリール制御管理情報（電断発生時の表示列の変動制御に関するデータ）を取得する（S27）。なお、リール制御管理情報（表示列の変動制御管理情報）は、各リールの制御状態（回転状況）に関する情報であり、電断時には、バックアップされて保存される。

【0447】

次いで、メインCPU101は、リール制御管理情報がリールの加速中、定速待ち又は定速中の回転状況に対応する情報であるか否かを判別する（S28）。

【0448】

S28において、メインCPU101が、S28の条件を満たさないと判別したとき（S28がNO判定の場合）、メインCPU101は、後述のS31の処理を行う。一方、S28において、メインCPU101が、S28の条件を満たすと判別したとき（S28がYES判定の場合）、メインCPU101は、回胴制御データ（リール制御管理情報）をクリアする（S29）。この処理により、遊技復帰後、リールの回転制御が加速処理から開始される。次いで、メインCPU101は、リールの作動タイミング値（回胴制御データの実行開始タイミング「1」）をセットする（S30）。なお、リールの作動タイミングに「1」がセットされると、リール制御処理（後述の図158中のS903参照）内で、励磁変更タイミングとなるため、メインCPU101は、リールの回転制御を加速処理から開始する。

【0449】

S30の処理後又はS28がNO判定の場合、メインCPU101は、リール数の値を1減算する（S31）。次いで、メインCPU101は、減算後のリール数の値が「0」であるか否かを判別する（S32）。

【0450】

S32において、メインCPU101が、減算後のリール数の値が「0」でないと判別したとき（S32がNO判定の場合）、メインCPU101は、チェック対象のリールを変えて、処理をS27の処理に戻し、S27以降の処理を繰り返す。

【0451】

一方、S32において、メインCPU101が、減算後のリール数の値が「0」であると判別したとき（S32がYES判定の場合）、メインCPU101は、RAM初期化処理を行う（S33）。この処理では、メインCPU101は、図12Cに示すメインRAM103の遊技用RAM領域内の「電源復帰時」のアドレスを、初期化開始の先頭アドレスとして設定し、該先頭アドレスから遊技用RAM領域の最終アドレスまでの情報を消去（クリア）する。

【0452】

次いで、メインCPU101は、電断検知時に退避させた全てのレジスタのデータを全てのレジスタに復帰させる（S34）。そして、S34の処理後、メインCPU101は、遊技復帰処理を終了し、処理を電断検知時の処理に戻す。

【0453】

本実施形態では、上述のようにして遊技復帰処理が行われる。本実施形態の遊技復帰処理では、上述のように、電断発生時の各ポートの入出力状態を電源復帰時に担保するとともに、電断時にリール回転中の場合には、電源復帰時にリール制御管理情報を取得してリールの再回転開始に必要な処理も行う（S25～S32の処理参照）。それゆえ、本実施形態では、回胴回転中の電断から復帰したときであっても、安定して、リールの再回転制御を行うことができ、遊技者に不快感を与えることが無くなる。

【0454】

また、上述した遊技復帰処理中のS25～S32の処理は、メインCPU101が、図67のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。なお、上述のように、本実施形態のパチスロ1で用いられている遊技機用セキュリティ機能付きのマイクロプロセッサ91では、メインCPU101専用の各種命令コードが

10

20

30

40

50

設けられている。例えば、図 6 7 中の「LDQ」命令（所定の読み出し命令）は、メイン CPU 101 専用命令コードの一つである。

【0455】

ソースプログラム上において、例えば、ソースコード「LDQ HL, k」が実行されると、Qレジスタの内容（格納データ）と、1バイトの整数k（直値）とで指定されたアドレスが、HLレジスタにロードされる。この際、Qレジスタの内容が指定先アドレスの上位側のアドレス値となり、整数k（直値）が指定先アドレスの下位側のアドレス値となる。それゆえ、図 6 7 中のソースコード「LDQ HL, .LOW.wr1__CTRL」が実行されると、Qレジスタの内容（回胴制御データ格納領域のアドレスの上位側のアドレス値）と、整数値「.LOW.wr1__CTRL」（回胴制御データ格納領域のアドレスの下位側のアドレス値）とで指定されるアドレス（回胴制御データ格納領域のアドレス）が、HLレジスタにロードされる。なお、「.LOW.」は、実際の命令ではなく、擬似命令と呼ばれるものである。この疑似命令の機能では、「.LOW.」に続いて規定される格納領域のアドレスの下位側アドレスのみが有効にされる。また、疑似命令は、実際の ROM に格納される命令ではなく、ソースファイルを ROM に格納するための形式に変換する際に、変換プログラム（アセンブラ）が参照するための命令である。

10

【0456】

上述のように、本実施形態では、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行うメイン CPU 101 専用命令コードを使用することにより、直値により、メイン ROM 102、メイン RAM 103 やメモリーマップ E/O にアクセスすることができる。この場合、ソースプログラム上において、アドレス設定に係る命令コードを省略することができる。その分、ソースプログラムの容量（メイン ROM 102 の使用容量）を低減することができる。この結果、本実施形態では、メイン ROM 102 において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

20

【0457】

〔設定変更確認処理〕

次に、図 6 8 及び図 6 9 を参照して、電源投入（リセット割込）時処理（図 6 4 参照）中の S 15 で行う設定変更確認処理について説明する。なお、図 6 8 は、設定変更確認処理の手順を示すフローチャートであり、図 6 9 A は、該フローチャート中の S 44 ~ S 47 の処理を実行するためのソースプログラムの一例を示す図であり、図 6 9 B は、該フローチャート中の S 57 の処理を実行するためのソースプログラムの一例を示す図である。

30

【0458】

まず、メイン CPU 101 は、メイン RAM 103 内の規定外 RAM 領域の初期化処理を行う（S 41）。次いで、メイン CPU 101 は、1 割り込み待ち処理を行う（S 42）。この処理では、メイン CPU 101 は、割込処理による無操作コマンドの副制御回路 200 への送信処理が終了するまで待機する。

【0459】

次いで、メイン CPU 101 は、RAM 初期化処理を行う（S 43）。この処理では、メイン CPU 101 は、図 12 C に示すメイン RAM 103 の遊技用 RAM 領域内の「RAM 異常時又は設定変更開始時」のアドレスを、初期化開始の先頭アドレスとして設定し、該先頭アドレスから遊技用 RAM 領域の最終アドレスまでの情報を消去（クリア）する。

40

【0460】

次いで、メイン CPU 101 は、設定用鍵型スイッチ 54 がオン状態であるか否かを判別する（S 44）。なお、設定用鍵型スイッチ 54 に差し込まれる設定キー（不図示）は、パチスロ 1 の設定（設定 1 ~ 6）を操作するための操作キーであり、設定キーがオンされていると、設定用鍵型スイッチ 54 がオン状態となる。

【0461】

S 44 において、メイン CPU 101 が、設定用鍵型スイッチ 54 がオン状態でないと判別したとき（S 44 が NO 判定の場合）、メイン CPU 101 は、設定変更確認処理を終

50

了し、処理を電源投入（リセット割込）時処理（図64参照）のS16の処理に移す。一方、S44において、メインCPU101が、設定用鍵型スイッチ54がオン状態であると判別したとき（S44がYES判定の場合）、メインCPU101は、メダル受付禁止の処理を行う（S45）。この処理により、セレクト66（図7参照）のソレノイドの駆動が行われず、投入されたメダルがメダル払出口24（図2参照）から排出される。

【0462】

次いで、メインCPU101は、レジスタに設定変更開始又は設定確認開始の情報（005H：第1の値）をセットし、設定変更コマンド（設定変更／設定確認開始）の生成格納処理を行う（S46）。この処理では、メインCPU101は、設定変更処理又は設定確認処理の開始時に主制御回路90から副制御回路200に送信される設定変更コマンドデータ（第1のコマンドデータ）を生成し、該コマンドデータをメインRAM103に設けられた通信データ格納領域に保存する。なお、設定変更コマンド生成格納処理の詳細については、後述の図70を参照しながら後で説明する。また、通信データ格納領域に保存された設定変更コマンド（設定変更／設定確認開始）は、後述の図158で説明する割込処理内の通信データ送信処理により、主制御回路90から副制御回路200に送信される。

10

【0463】

次いで、メインCPU101は、エラーカウトリレーをオン状態にセットする（S47）。次いで、メインCPU101は、設定変更及び設定確認のいずれが行われたかを判別する（S48）。

【0464】

S48において、メインCPU101が、設定変更が行われていない（設定確認が行われた）と判別したとき（S48がNO判定の場合）、メインCPU101は、後述のS55の処理を行う。

20

【0465】

一方、S48において、メインCPU101が、設定変更が行われた（設定確認が行われていない）と判別したとき（S48がYES判定の場合）、メインCPU101は、設定値の更新処理を行う（S49）。次いで、メインCPU101は、設定値の7セグ表示設定処理を行う（S50）。この処理により、更新後の設定値が情報表示器6内の7セグLEDで表示可能になる。

【0466】

次いで、メインCPU101は、リセットスイッチ76がオン状態であるか否かを判別する（S51）。

30

【0467】

S51において、メインCPU101が、リセットスイッチ76がオン状態であると判別したとき（S51がYES判定の場合）、メインCPU101は、処理をS49の処理に戻し、S49以降の処理を繰り返す。一方、S51において、メインCPU101が、リセットスイッチ76がオン状態でないと判別したとき（S51がNO判定の場合）、メインCPU101は、スタートスイッチ79がオン状態であるか否かを判別する（S52）。

【0468】

S52において、メインCPU101が、スタートスイッチ79がオン状態でないと判別したとき（S52がNO判定の場合）、メインCPU101は、処理をS51の処理に戻し、S51以降の処理を繰り返す。一方、S52において、メインCPU101が、スタートスイッチ79がオン状態であると判別したとき（S52がYES判定の場合）、メインCPU101は、メインRAM103に設けられた設定値格納領域（不図示）に設定値を格納する（S53）。

40

【0469】

次いで、メインCPU101は、設定用鍵型スイッチ54がオフ状態であるか否かを判別する（S54）。

【0470】

S54において、メインCPU101が、設定用鍵型スイッチ54がオフ状態でないと判

50

別したとき（S 5 4 が N O 判定の場合）、メイン C P U 1 0 1 は、S 5 4 の処理を繰り返す。一方、S 5 4 において、メイン C P U 1 0 1 が、設定用鍵型スイッチ 5 4 がオフ状態であると判別したとき（S 5 4 が Y E S 判定の場合）、メイン C P U 1 0 1 は、後述の S 5 5 の処理を行う。

【0 4 7 1】

S 4 8 が N O 判定の場合又は S 5 4 が Y E S 判定の場合、メイン C P U 1 0 1 は、設定変更及び設定確認のいずれが行われたか否かを判別する（S 5 5）。

【0 4 7 2】

S 5 5 において、メイン C P U 1 0 1 が、設定変更が行われていない（設定確認が行われた）と判別したとき（S 5 5 が N O 判定の場合）、メイン C P U 1 0 1 は、後述の S 5 7 の処理を行う。一方、S 5 5 において、メイン C P U 1 0 1 が、設定変更が行われた（設定確認が行われていない）と判別したとき（S 5 5 が Y E S 判定の場合）、メイン C P U 1 0 1 は、R A M 初期化処理を行う（S 5 6）。この処理では、メイン C P U 1 0 1 は、図 1 2 C に示すメイン R A M 1 0 3 の遊技用 R A M 領域内の図示しない「設定変更終了時」のアドレス（設定値格納領域の次のアドレス）を、初期化開始の先頭アドレスとして設定し、該先頭アドレスから遊技用 R A M 領域の最終アドレスまでの情報を消去（クリア）する。

【0 4 7 3】

S 5 6 の処理後又は S 5 5 が N O 判定の場合、メイン C P U 1 0 1 は、L レジスタに設定変更終了又は設定確認終了の情報（0 0 4 H：第 2 の値）をセットし、設定変更コマンド（設定変更／設定確認終了）の生成格納処理を行う（S 5 7）。この処理では、メイン C P U 1 0 1 は、設定変更処理又は設定確認処理の終了時に主制御回路 9 0 から副制御回路 2 0 0 に送信される設定変更コマンドデータ（第 2 のコマンドデータ）を生成し、該コマンドデータをメイン R A M 1 0 3 に設けられた通信データ格納領域に保存する。なお、設定変更コマンド生成格納処理の詳細については、後述の図 7 0 を参照しながら後で説明する。また、通信データ格納領域に保存された設定変更コマンド（設定変更／設定確認終了）は、後述の図 1 5 8 で説明する割込処理内の通信データ送信処理により、主制御回路 9 0 から副制御回路 2 0 0 に送信される。そして、S 5 7 の処理後、メイン C P U 1 0 1 は、設定変更確認処理を終了し、処理を電源投入（リセット割込）時処理（図 6 4 参照）の S 1 6 の処理に移す。

【0 4 7 4】

本実施形態では、上述のようにして、設定変更確認処理が行われる。上述した設定変更確認処理中の S 4 4 ～ S 4 7 の処理は、メイン C P U 1 0 1 が、図 6 9 A のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中で、例えば、S 4 4 の設定キーの状態判定処理は、図 6 9 A 中のソースコード「B I T Q 7, (. L O W. (w I B U F + 4))」により実行され、S 4 7 のエラーカウントリレーをオン状態にセットする処理は、図 6 9 A 中のソースコード「S E T Q 1, (. L O W. w E C R R E Q)」により実行される。

【0 4 7 5】

「B I T Q」命令及び「S E T Q」命令はともに、Q レジスタ（拡張レジスタ）を用いてアドレス指定を行うメイン C P U 1 0 1 専用命令コードである。

【0 4 7 6】

ソースプログラム上において、例えば、ソースコード「B I T Q b, (k)」が実行されると、Q レジスタの格納データ（上位側アドレス値）と、1 バイトの整数値 k（直値：下位側アドレス値）とで指定されるアドレスのメモリのビット b がチェックされ、該ビット b に「1」が格納されていれば、フラグ・レジスタ F のゼロフラグ（ビット 6：図 1 1 参照）に「0」がセットされ、該ビット b に「0」が格納されていれば、フラグ・レジスタ F のゼロフラグ（所定のビット領域）に「1」がセットされる。それゆえ、図 6 9 A 中のソースコード「B I T Q 7, (. L O W. (w I B U F + 4))」が実行されると、Q レジスタの格納データと、整数値「. L O W. (w I B U F + 4)」とで指定されるア

10

20

30

40

50

ドレスのメモリのビット7がチェックされ、該ビット7に「1」が格納されていれば、フラグ・レジスタFのゼロフラグに「0」がセットされ、該ビット7に「0」が格納されていれば、フラグ・レジスタFのゼロフラグに「1」がセットされる。

【0477】

また、ソースプログラム上において、例えば、ソースコード「SETQ b, (k)」が実行されると、Qレジスタの格納データ（上位側アドレス値）と、1バイトの整数値k（直値：下位側アドレス値）とで指定されるアドレスのメモリのビットbに「1」がセットされる。それゆえ、図69A中のソースコード「SETQ 1, (.LOW.wECRREQ)」が実行されると、Qレジスタの格納データと、整数値「.LOW.wECRREQ」とで指定されるアドレスのメモリのビット1に「1」がセットされる。

10

【0478】

すなわち、本実施形態の設定変更確認処理では、上述のようなQレジスタ（拡張レジスタ）を用いた各種メインCPU101専用命令コードが使用されており、これらのメインCPU101専用命令コードの使用により、直値で、メインROM102、メインRAM103やメモリーマップI/Oにアクセスすることができる。この場合、アドレス設定に係る命令コードを省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102の空き容量を増やすことが可能となるとともに、処理の高速化も図ることができる。

【0479】

また、上述した設定変更確認処理中のS46の設定変更／設定確認開始時に行う設定変更コマンド（初期化コマンド）の生成格納処理は、メインCPU101が図69A中の「CALLF」命令を実行することにより行われ、上述したS57の設定変更／設定確認終了時に行う設定変更コマンド（初期化コマンド）の生成格納処理は、メインCPU101が図69B中の「CALLF」命令を実行することにより行われる。なお、「CALLF」命令もまた、メインCPU101専用命令コードである。

20

【0480】

ソースプログラム上において、例えば、ソースコード「CALLF mn」が実行されると、現在のPCレジスタ（プログラム・カウンタPC：図11参照）の値（格納データ）がスタックポインタ（SP）で指定されているメモリに保存され、スタックポインタが-2更新され、「mn」がPCレジスタに格納されて、「mn」で指定されているアドレスに処理がジャンプする。ただし、「CALLF」命令は、2バイト命令であり、ジャンプできるアドレス範囲は、0000H～11FFHの範囲となる。それゆえ、例えば、図69A中のソースコード「CALLF SB_PCINIT_00」が実行されると、現在のPCレジスタの値がスタックポインタ（SP）で指定されているメモリに保存され、スタックポインタが-2更新され、「SB_PCINIT_00」のアドレスがPCレジスタに格納されて、「SB_PCINIT_00」で指定されているアドレスに処理がジャンプする。

30

【0481】

なお、本実施形態では、「CALLF」命令と同種の命令コードとして、「CALL」命令と呼ばれる命令コードも用意されている。そして、ソースプログラム上において、例えば、ソースコード「CALL mn」が実行されると、「CALLF」命令と同様に、現在のPCレジスタ（プログラム・カウンタPC：図11参照）の値（格納データ）がスタックポインタ（SP）で指定されているメモリに保存され、スタックポインタが-2更新され、「mn」がPCレジスタに格納されて、「mn」で指定されているアドレスに処理がジャンプする。ただし、「CALL」命令は、3バイト命令であり、ジャンプできるアドレス範囲が、「CALLF」命令のそれと異なり、ジャンプできるアドレス範囲は、0000H～FFFFHの範囲である。なお、「CALLF」命令は、「CALL」命令に比べてバイト数の少ない命令コードであるので、ソースプログラムの容量（メインROM102の使用容量）を低減することができるとともに、処理の効率化も図ることができる。

40

【0482】

50

また、本実施形態の設定変更確認処理では、図 6 9 A 及び 6 9 B に示すように、S 4 6 の「CALLF」命令で指定するジャンプ先のアドレス「SB_PCINIT_00」は、S 5 7 の「CALLF」命令で指定するジャンプ先のアドレスと同じである。すなわち、本実施形態では、設定変更時（遊技機起動時）、設定確認開始時（通常動作中）及び設定確認終了時に副制御回路 2 0 0 に送信する設定変更コマンド（初期化コマンド）の生成格納処理を実行するためのソースプログラムが、互いに同じであり、S 4 6 及び S 5 7 の両処理において用いられる、設定変更コマンド生成格納処理のソースプログラムが共有化（モジュール化）されている。

【0483】

この場合、S 4 6 及び S 5 7 の両処理において、それぞれ別個に設定変更コマンド生成格納処理のソースプログラムを設ける必要が無くなるので、その分、ソースプログラムの容量（メインROM 1 0 2 の使用容量）を低減することができる。この結果、本実施形態では、メインROM 1 0 2 において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【0484】

〔設定変更コマンド生成格納処理〕

次に、図 7 0 及び図 7 1 を参照して、設定変更確認処理（図 6 8 参照）中の S 4 6 及び S 5 7 で行う設定変更コマンド生成格納処理について説明する。なお、図 7 0 は、設定変更コマンド生成格納処理の手順を示すフローチャートであり、図 7 1 は、設定変更コマンド生成格納処理を実行するためのソースプログラムの一例を示す図である。

【0485】

まず、メインCPU 1 0 1 は、設定値（1～6）の情報をEレジスタにセットする（S 6 1）。次いで、メインCPU 1 0 1 は、RT状態の情報をCレジスタにセットする（S 6 2）。次いで、メインCPU 1 0 1 は、設定変更コマンドのコマンド種別情報（0 2 H）をAレジスタにセットする（S 6 3）。

【0486】

次いで、メインCPU 1 0 1 は、通信データ格納処理を行う（S 6 4）。この処理では、メインCPU 1 0 1 は、S 6 1～S 6 3 で各レジスタにセットされた情報と、S 4 6 又は S 5 7（図 6 8 参照）でDレジスタにセットされた情報（設定ステータスである設定変更開始／設定変更終了／設定確認開始／設定確認終了）とを用いて、設定変更コマンドデータを生成し、該生成されたコマンドデータを通信データ格納領域に保存する。なお、通信データ格納処理の詳細については、後述の図 7 2 を参照しながら後で説明する。

【0487】

S 6 4 の処理後、メインCPU 1 0 1 は、設定変更コマンド生成格納処理を終了する。なお、設定変更確認処理（図 6 8 参照）中の S 4 6 で行う設定変更コマンド生成格納処理を終了する際には、メインCPU 1 0 1 は、S 6 4 の処理後、処理を設定変更確認処理（図 6 8 参照）の S 4 7 の処理に移す。また、設定変更確認処理（図 6 8 参照）中の S 5 7 で行う設定変更コマンド生成格納処理を終了する際には、メインCPU 1 0 1 は、S 6 4 の処理後、設定変更コマンド生成格納処理を終了するとともに、設定変更確認処理（図 6 8 参照）も終了する。

【0488】

本実施形態では、上述のようにして設定変更コマンド生成格納処理が行われる。なお、上述した設定変更コマンド生成格納処理は、メインCPU 1 0 1 が、図 7 1 のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【0489】

上述のように、設定変更コマンド生成格納処理では、設定変更コマンド生成格納処理が実行される直前に設定ステータスが通信パラメータ 4 としてDレジスタに格納され、設定変更コマンド生成格納処理の実行中に設定値が通信パラメータ 3 としてEレジスタに格納され、RT情報が通信パラメータ 5 としてCレジスタに格納される。すなわち、設定変更コマンド（初期化コマンド）を構成する通信パラメータ 1～5 のうち、通信パラメータ 3～

10

20

30

40

50

5は副制御回路200側で使用(解析)される通信パラメータ(使用パラメータ)であり、これらの通信パラメータには新たな情報がセットされる。一方、設定変更コマンド(初期化コマンド)を構成するその他の通信パラメータ1及び2は、副制御回路200側で使用(解析)されない通信パラメータ(未使用パラメータ)であり、通信パラメータ1及び2に対しては、現時点でLレジスタ及びHレジスタにそれぞれ格納されている値がセットされる。それゆえ、設定変更コマンド(初期化コマンド)送信時における通信パラメータ1及び2の値は不定値となる。この場合、設定変更コマンドのサム値(BCC)を送信毎に不定値にすることができ、ゴト等の不正行為を抑制することができる。

【0490】

[通信データ格納処理]

次に、図72及び図73を参照して、例えば、設定変更コマンド生成格納処理(図70参照)中のS64で行う通信データ格納処理について説明する。なお、通信データ格納処理は、設定変更コマンド生成時だけでなく、他のコマンド生成時にも実行される。図72は、通信データ格納処理の手順を示すフローチャートであり、図73は、通信データ格納処理中のS71～S76の処理を実行するためのソースプログラムの一例を示す図である。

【0491】

まず、メインCPU101は、Aレジスタにセットされているデータを通信コマンド種別のデータとして、メインRAM103内の通信データ一時格納領域(不図示)に格納する(S71)。次いで、メインCPU101は、Hレジスタ及びLレジスタにセットされているデータを、それぞれ通信コマンドのパラメータ1及び2として、メインRAM103内の通信データ一時格納領域(所定の格納領域)に格納する(S72)。

【0492】

次いで、メインCPU101は、Dレジスタ及びEレジスタにセットされているデータを、それぞれ通信コマンドのパラメータ3及び4として、メインRAM103内の通信データ一時格納領域に格納する(S73)。次いで、メインCPU101は、Bレジスタ及びCレジスタにセットされているデータを、それぞれ通信コマンドのパラメータ5及びRT状態のデータとして、メインRAM103内の通信データ一時格納領域に格納する(S74)。

【0493】

次いで、メインCPU101は、Aレジスタ～Lレジスタにセットされているデータ値から通信コマンドのBCCデータ(サム値)を生成する(S75)。次いで、メインCPU101は、生成したBCCデータをメインRAM103内の通信データ一時格納領域に格納する(S76)。

【0494】

S76の処理後、メインCPU101は、メインRAM103内の通信データ格納領域に空きがあるか否かを判別する(S77)。なお、本実施形態では、通信データ格納領域に最大9個のコマンドデータが格納可能である(後述の図75B参照)。

【0495】

S77において、メインCPU101が、通信データ格納領域に空きがないと判別したとき(S77がNO判定の場合)、メインCPU101は、通信データ格納処理を終了するとともに、例えば、設定変更コマンド生成格納処理(図70参照)も終了する。

【0496】

一方、S77において、メインCPU101が、通信データ格納領域に空きがあると判別したとき(S77がYES判定の場合)、メインCPU101は、上述したS71～S76の処理により通信データ一時格納領域に格納された通信データを通信コマンドデータとして、通信データ格納領域に格納する(S78)。

【0497】

次いで、メインCPU101は、通信データポインタ更新処理を行う(S79)。この処理では、メインCPU101は、主に、通信データ格納領域内における通信データの格納アドレスを示す通信データポインタの更新処理を行う。なお、通信データポインタ更新処

10

20

30

40

50

理の詳細については、後述の図 7 4 を参照しながら後で説明する。

【0498】

そして、S79の処理後、メインCPU101は、通信データ格納処理を終了するとともに、例えば、設定変更コマンド生成格納処理（図70参照）も終了する。

【0499】

本実施形態では、上述のようにして通信データ格納処理が行われる。なお、上述した通信データ格納処理中のS71～S76の処理は、メインCPU101が、図73のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。そして、この一連の処理において、コマンドデータに含まれる、通信コマンドの種別データ、各種通信パラメータ、遊技状態フラグデータ及びBCCデータの格納処理は、図73に示すように、ソースプログラム上では、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行うメインCPU101専用命令コードである、「LDQ」命令を用いて実行される。

【0500】

具体的には、ソースコード「LDQ（.LOW.（WPDT_TMP+0）），A」の実行により、Aレジスタに格納された通信コマンドの種別データが、Qレジスタの格納データ（上位側アドレス値）と1バイトの整数値「.LOW.（WPDT_TMP+0）」（下位側アドレス値）とで指定されたアドレスの通信データ一時格納領域に格納される。また、ソースコード「LDQ（.LOW.（WPDT_TMP+1）），HL」の実行により、Lレジスタに格納された通信パラメータ1が、Qレジスタの格納データと1バイトの整数値「.LOW.（WPDT_TMP+1）」とで指定されたアドレスの通信データ一時格納領域に格納され、Hレジスタに格納された通信パラメータ2が、その次のアドレスの通信データ一時格納領域に格納される。また、ソースコード「LDQ（.LOW.（WPDT_TMP+3）），DE」の実行により、Eレジスタに格納された通信パラメータ3が、Qレジスタの格納データと1バイトの整数値「.LOW.（WPDT_TMP+3）」とで指定されたアドレスの通信データ一時格納領域に格納され、Dレジスタに格納された通信パラメータ4が、その次のアドレスの通信データ一時格納領域に格納される。そして、ソースコード「LDQ（.LOW.（WPDT_TMP+5）），BC」の実行により、Cレジスタに格納された通信パラメータ5が、Qレジスタの格納データと1バイトの整数値「.LOW.（WPDT_TMP+5）」とで指定されたアドレスの通信データ一時格納領域に格納され、Bレジスタに格納された遊技状態フラグデータが、その次のアドレスの通信データ一時格納領域に格納される。

【0501】

さらに、通信データ格納処理でセットされたコマンドデータのサム値となるBCCデータは、一連のソースコード「ADD（加算命令コード）A，H」～「ADD A，B」の実行により算出され、Aレジスタに格納される。そして、ソースコード「LDQ（.LOW.（WPDT_TMP+7）），A」の実行により、Aレジスタに格納されたBCCデータが、Qレジスタの格納データと1バイトの整数値「.LOW.（WPDT_TMP+7）」とで指定されたアドレスの通信データ一時格納領域に格納される。

【0502】

上述のように、本実施形態では、1パケット（8バイト）の通信データ（コマンドデータ）を作成する際に、各種パラメータをレジスタから転送して通信データ一時格納領域（通信バッファ）に格納する。このようなコマンドデータの作成手法では、コマンド生成時に各レジスタに格納されているデータがそのままコマンドデータの各種パラメータとして通信データ一時格納領域に格納される。それゆえ、未使用パラメータを含むコマンドデータを作成した時には、作成時毎に、未使用パラメータの値が不定値となる。この場合、同じ種別のコマンドデータあり、かつ、使用パラメータの値が同一であっても、コマンド作成毎に、コマンドデータのサム値（BCCデータ）が可変可能となる。また、本実施形態では、誤り符号の一つであるサム値の計算をADD（加算命令コード）により算出したが、加算命令コードに換えて、SUB（減算命令コード）、XOR（排他的論理和命令コード）により誤り符号を算出しても同様の効果が得られる。さらに、メインCPU101専用

10

20

30

40

50

命令である、MUL（乗算命令コード）又はDIV（除算命令コード）を使用して誤り符号を算出しても同様の効果が得られる。

【0503】

それゆえ、本実施形態では、未使用パラメータを不定値とすることにより、通信データの解析を困難にしてゴト等の不正行為を抑止することができるとともに、不必要なゴト対策処理を加える必要がないため、ゴト対策処理の追加による、主制御回路90のプログラム容量の圧迫を抑制することができる。

【0504】

〔通信データポインタ更新処理〕

次に、図74及び図75を参照して、通信データ格納処理（図72参照）中のS79で行う通信データポインタ更新処理について説明する。なお、図74は、通信データポインタ更新処理の手順を示すフローチャートであり、図75Aは、通信データポインタ更新処理を実行するためのソースプログラムの一例を示す図であり、図75Bは、通信データポインタ更新処理のソースプログラム上で実際にセットされる通信データ格納領域の構成を示す図である。

【0505】

まず、メインCPU101は、現在、セットされている通信データポインタの値を取得する（S81）。

【0506】

次いで、メインCPU101は、通信データポインタの値を1パケット分（8バイト）加算更新する（S82）。なお、この処理において、更新後の通信データポインタの値が、通信データ格納領域（図75B参照）の上限サイズ以上となる場合には、メインCPU101は、更新後の通信データポインタの値を「0」にセットし、これにより、通信データ格納領域に格納されているコマンドデータを全て無効にする（破棄した状態と同様の状態にする）。

【0507】

本実施形態では、1回の送信動作で送信されるデータ量（1パケット）は8バイトである。すなわち、本実施形態では、一つの送信動作で一つのコマンドデータを送信することができる。また、本実施形態では、通信データ格納領域に最大9個のコマンドデータを格納可能であるので（図75B参照）、通信データ格納領域の上限サイズは、72バイト（＝8バイト×9）となる。それゆえ、本実施形態では、通信データポインタの範囲を「0」～「71」とし、S82の処理において、更新後（通信データポインタを+8更新した場合）の通信データポインタの値が「71（上限値）」を超えるような値となる場合には、更新後の通信データポインタの値を「0」にセットして（通信データの格納先のアドレスを先頭アドレスに戻して）、通信データ格納領域に格納されているコマンドデータを全て無効にする（破棄した状態と同様の状態にする）。なお、通信データポインタの値を「0」にセットすると、次にコマンドデータを通信データ格納領域に格納する場合には、通信データ格納領域の先頭アドレスから格納されるので、その前に格納されていたコマンドデータは新たなコマンドデータで上書きされることになる。それゆえ、本実施形態では、通信データポインタの値が「71（上限値）」を超えた場合に、通信データ格納領域を初期化（クリア）する必要はない。

【0508】

そして、S82の処理後、メインCPU101は、通信データポインタ更新処理を終了するとともに、通信データ格納処理（図72参照）も終了する。

【0509】

本実施形態では、上述のようにして通信データポインタ更新処理が行われる。そして、上述した通信データポインタ更新処理は、メインCPU101が、図75Aのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中で、S82の通信データポインタの更新処理は、図75A中の「ADD」命令及び「ICPLD」命令（所定の更新命令）により実行されるが、この「ICPLD」命令もまた、メイン

10

20

30

40

50

CPU101専用命令コードである。

【0510】

ソースプログラム上において、例えば、ソースコード「ICPLD A, n」が実行されると、Aレジスタの内容（格納データ）と整数nとが比較され、Aレジスタの内容が整数n未満である場合には、Aレジスタの内容に「1」が加算され、Aレジスタの内容が整数n以上である場合には、Aレジスタに「0」がセットされる。

【0511】

それゆえ、S82の通信データポインタの更新処理を実行する場合、図75Aのソースプログラム上では、まず、ソースコード「ADD A, 7」が実行され、Aレジスタの内容（更新前の通信データポインタの値）に「7」が加算され、該加算結果がAレジスタに格納される。次いで、ソースコード「ICPLD A, 71」が実行され、Aレジスタの内容（7加算後の通信データポインタの値）と整数「71」とを比較し、Aレジスタの内容が整数「71」未満である場合には、Aレジスタの内容に「1」を加算し、Aレジスタの内容が整数「71」以上である場合には、Aレジスタに「0」をセットする。すなわち、S82の処理において、通信データポインタの値を+7更新したときに、更新後の通信データポインタの値が上限値「71」を超えるような場合には、通信データポインタをゼロクリアする処理（通信データの格納アドレスを通信データ格納領域の先頭アドレスに戻す処理）が行われる。一方、更新後の通信データポインタの値が上限値「71」を超えない場合には、「ICPLD」命令でさらに通信データポインタに「1」を加算することにより、トータルで通信データポインタの値を+8更新する。

【0512】

上述のように、本実施形態では、通信データポインタ更新処理において、一つの「ICPLD」命令コード（送信バッファの上限判定命令と、判断分岐命令とが一体になっている命令コード）により、通信データポインタの更新（1加算）処理、更新後の通信データポインタの判定チェック処理及び通信データポインタのクリア処理をまとめて実行することができる。この場合、各処理を別個に実行するための命令コードを設ける必要がなくなる。例えば、更新後の通信データポインタの値がその上限値「71」を超えるか否かの判断分岐命令コードを省略することができる。

【0513】

それゆえ、通信データポインタ更新処理等において、メインCPU101専用の「ICPLD」命令コードを用いることにより、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【0514】

〔電断時（外部）処理〕

次に、メインCPU101の制御により行われるパチスロ1の電断時（外部）処理を、図76を参照して説明する。図76は、電断時（外部）処理の手順を示すフローチャートである。なお、図76に示す電断時（外部）処理は、電源管理回路93が、マイクロプロセッサ91に供給される電源電圧の低下（電断）を検知した際に、電断検知信号をマイクロプロセッサ91の「XINT」端子に出力し、これにより、マイクロプロセッサ91の割込みコントローラ112からメインCPU101に出力される割込要求信号に基づいて、実行される。

【0515】

まず、メインCPU101は、全てのレジスタにセットされているデータを退避させる（S91）。次いで、メインCPU101は、電断検知ポートにセットされているデータを読み込む（S92）。

【0516】

次いで、メインCPU101は、電断検知ポートがオン状態であるか否かを判別する（S93）。

【0517】

S93において、メインCPU101が、電断検知ポートがオン状態でないと判別したとき（S93がNO判定の場合）、メインCPU101は、割込処理許可をセットする（S94）。そして、S94の処理後、メインCPU101は、電断時（外部）処理を終了する。なお、S93がNO判定である場合に行われるこれらの処理は、電源管理回路93が瞬時的に電断を検知した場合等に発生する瞬停対策の処理に対応する。

【0518】

一方、S93において、メインCPU101が、電断検知ポートがオン状態であると判別したとき（S93がYES判定の場合）、メインCPU101は、メダル投入不可を設定し、ホッパー装置51の停止を設定する（S95）。

10

【0519】

次いで、メインCPU101は、現在セットされているスタックポインタ（SP）の値をメインRAM103内の遊技用RAM領域のスタックエリアに保存する（S96）。

【0520】

次いで、メインCPU101は、メインRAM103のチェックサム生成処理を行う（S97）。なお、この処理は、メインRAM103内の規定外作業領域（図12C参照）で行われる。また、このチェックサム生成処理で用いられるプログラムはメインROM102内の規定外エリアに格納されている（図12B参照）。なお、チェックサム生成処理の詳細については、後述の図77を参照しながら後で説明する。

【0521】

次いで、メインCPU101は、メインRAM103へのアクセス禁止を設定する（S98）。そして、S98の処理後、電源が停止するまで（電源電圧が、メインCPU101が動作できない電圧に達するまで）無限ループ処理が行われる。

20

【0522】

[チェックサム生成処理（規定外）]

次に、図77及び図78を参照して、電断時（外部）処理（図76参照）中のS97で行うチェックサム生成処理について説明する。なお、図77は、チェックサム生成処理の手順を示すフローチャートであり、図78Aは、チェックサム生成処理を実行するためのソースプログラムの一例を示す図であり、図78Bは、チェックサム生成処理で実行されるスタックポインタの更新動作及びメインRAM103からレジスタへのデータの読み出し動作の様子を示す図である。

30

【0523】

まず、メインCPU101は、現在のスタックポインタ（SP）の値（遊技用RAM領域のスタックエリアの使用アドレス）をメインRAM103の規定外RAM領域の規定外スタックエリアに保存する（S101）。次いで、メインCPU101は、スタックポインタに規定外スタックエリアのアドレスをセットする（S102）。次いで、メインCPU101は、RAMアドレス（規定外スタックエリアのアドレス）の上位側のアドレス値（F0H）をQレジスタにセットする（S103）。次いで、メインCPU101は、電断発生フラグを設定する（S104）。

【0524】

次いで、メインCPU101は、スタックポインタに、遊技用RAM領域内のサム値の計算開始アドレスをセットし、サム算出カウンタに、サム値の算出対象格納領域のバイト数を「2」で除算した値をセットする（S105）。なお、サム算出カウンタは、サム値算出の終了契機を判定するためのカウンタであり、メインRAM103に設けられる。そして、S105で設定されたサム算出カウンタが「0」になれば、メインRAM103の遊技用RAM領域のサム値算出処理を終了する。

40

【0525】

次いで、メインCPU101は、HLレジスタを0クリア（値「0」をセット）する（S106）。この処理により、サム値の初期値「0」がセットされる。

【0526】

50

次いで、メインCPU101は、「POP命令」（特定の命令）と呼ばれる命令コード（図78A中に記載のソースコード「POP DE」）を実行し、スタックポインタ（SP）にセットされたメインRAM103の格納領域のアドレスから2バイト分の領域のデータ（保存値）をDEレジスタに読み出す（S107）。

【0527】

なお、「POP」命令が実行されると、スタックポインタで指定されたアドレスの1バイト領域に保存されているデータ（メモリ内容）が、ペアレジスタの下位側のレジスタにロードされ、スタックポインタで指定されたアドレスを1更新したアドレスの1バイト領域に保存されているデータ（メモリ内容）が、ペアレジスタの上位側のレジスタにロードされる。また、「POP」命令が実行されると、スタックポインタ（SP）にセットされたアドレスに対して2バイト分のアドレス更新処理（アドレスを「2」加算する処理）が行われる。

【0528】

それゆえ、S107の処理では、スタックポインタで指定されたアドレスに保存されているデータ（メモリ内容）がEレジスタにロードされ、スタックポインタで指定されたアドレスに「1」を加算したアドレスに保存されているデータ（メモリ内容）がDレジスタにロードされる。

【0529】

図78Bには、「POP」命令実行時における、DEレジスタへのデータの読み込み動作、及び、スタックポインタにセットされるアドレスの更新動作の様子を示す。サム値の算出開始時に、スタックポインタ（SP）にセットされているアドレスが「F010h」である場合には、アドレス「F010h」に保存されているデータ（メモリ内容）がEレジスタにロードされ、アドレス「F011h」に保存されているデータ（メモリ内容）がDレジスタにロードされる。また、この際、スタックポインタ（SP）にセットされているアドレスに2加算する更新処理が行われ、スタックポインタ（SP）にセットされているアドレスが「F010h」から「F012h」に変更される。次いで、再度、「POP」命令が実行されると、アドレス「F012h」に保存されているデータ（メモリ内容）がEレジスタにロードされ、アドレス「F013h」に保存されているデータ（メモリ内容）がDレジスタにロードされる。また、この際、スタックポインタ（SP）にセットされているアドレスの更新処理が行われ、スタックポインタ（SP）にセットされているアドレスが「F012h」から「F014h」に変更される。その後、「POP」命令が実行される度に上述した、DEレジスタへのデータの読み込み動作及びスタックポインタにセットされるアドレスの更新動作が繰り返される。

【0530】

S107の処理後、メインCPU101は、サム値の算出処理を行う（S108）。具体的には、メインCPU101は、HLレジスタに格納されている値にDEレジスタに格納されている値を加算し、該加算された値をサム値としてHLレジスタに格納する。

【0531】

次いで、メインCPU101は、サム算出カウンタの値を1減算する（S109）。次いで、メインCPU101は、更新後のサム算出カウンタの値が「0」であるか否かを判別する（S110）。

【0532】

S110において、メインCPU101が、サム算出カウンタの値が「0」でないと判別したとき（S110がNO判定の場合）、メインCPU101は、処理をS107の処理に戻し、S107以降の処理を繰り返す。すなわち、メインRAM103の遊技用RAM領域のサム値算出処理が終了するまで、S107～S110の処理が繰り返される。

【0533】

一方、S110において、メインCPU101が、サム算出カウンタの値が「0」であると判別したとき（S110がYES判定の場合）、メインCPU101は、DEレジスタに、メインRAM103内の規定外RAM領域のサム値の計算開始アドレスをセットし、

10

20

30

40

50

サム算出カウンタに、規定外用サムカウント値をセットする（S111）。なお、規定外用サムカウント値は、規定外用格納領域のバイト数となる。それゆえ、S111で設定されたサム算出カウンタが「0」になれば、メインRAM103の規定外RAM領域のサム値算出処理、すなわち、メインRAM103全体のサム値算出処理が終了する。

【0534】

次いで、メインCPU101は、DEレジスタにセットされた規定外RAM領域のアドレスから1バイト分の領域のデータ（保存値）をAレジスタに読み出す（S112）。

【0535】

次いで、メインCPU101は、サム値の算出処理を行う（S113）。具体的には、メインCPU101は、HLレジスタに格納されている値にAレジスタに格納されている値を加算し、該加算された値をサム値としてHLレジスタに格納する。

【0536】

次いで、メインCPU101は、DEレジスタに格納されているアドレスを1加算し、サム算出カウンタの値を1減算する（S114）。次いで、メインCPU101は、更新後のサム算出カウンタの値が「0」であるか否かを判別する（S115）。

【0537】

S115において、メインCPU101が、サム算出カウンタの値が「0」でないと判別したとき（S115がNO判定の場合）、メインCPU101は、処理をS112の処理に戻し、S112以降の処理を繰り返す。すなわち、メインRAM103の規定外RAM領域のサム値を遊技用RAM領域のサム値に加算する処理が終了するまで、S112～S115の処理が繰り返される。

【0538】

一方、S115において、メインCPU101が、サム算出カウンタの値が「0」であると判別したとき（S115がYES判定の場合）、メインCPU101は、HLレジスタに格納されている値を電断発生時のサム値として、メインRAM103内のサム値格納領域（不図示）に保存する（S116）。次いで、メインCPU101は、S101で規定外スタックエリアに保存されたスタックポインタ（SP）の値をスタックポインタにセットする（S117）。そして、S117の処理後、メインCPU101は、チェックサム生成処理を終了し、処理を電断時（外部）処理（図76参照）のS98の処理に移す。

【0539】

本実施形態では、上述のようにしてチェックサム生成処理が行われる。そして、上述したチェックサム生成処理は、メインCPU101が、図78Aのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。上述のように、本実施形態では、電断発生時のメインRAM103のチェックサムは、加算式で算出される。この際、遊技用RAM領域のサム値算出では、2バイト単位で加算処理（図78A中のソースコード「ADD HL, DE」参照）が行われ、規定外RAM領域では、1バイト単位で加算処理（図78A中のソースコード「ADDWB HL, A」参照）が行われる。

【0540】

〔サムチェック処理（規定外）〕

次に、図79～図81を参照して、電源投入時処理（図64参照）中のS9で行うサムチェック処理について説明する。なお、図79及び図80は、サムチェック処理の手順を示すフローチャートであり、図81は、サムチェック処理中のS122～S132の処理を実行するためのソースプログラムの一例を示す図である。

【0541】

まず、メインCPU101は、現在のスタックポインタ（SP）の値を規定外スタックエリアに保存する（S121）。次いで、メインCPU101は、スタックポインタにサム値格納領域のアドレスをセットし、サム算出カウンタに、サム値の算出対象格納領域のバイト数を「2」で除算した値をセットする（S122）。なお、ここでセットされるサム算出カウンタは、サム値算出（サム値の減算処理）の終了契機を判定するためのカウンタであり、メインRAM103に設けられる。次いで、メインCPU101は、サム値格納

10

20

30

40

50

領域からサム値（チェックサム）を取得する（S123）。この処理により、電断発生時に生成されたチェックサム（減算前の初期値）がHLレジスタに格納される。

【0542】

次いで、メインCPU101は、「POP」命令を実行し、スタックポインタ（SP）にセットされたメインRAM103の格納領域のアドレスから2バイト分の領域のデータ（保存値）をDEレジスタに読み出す（S124）。なお、この際、「POP」命令の実行により、スタックポインタで指定されたアドレスの1バイト領域に保存されているデータ（メモリ内容）が、Eレジスタにロードされ、スタックポインタで指定されたアドレスを1更新したアドレスの1バイト領域に保存されているデータ（メモリ内容）が、Dレジスタにロードされる（図78B参照）。また、「POP」命令が実行されると、スタックポインタ（SP）にセットされたアドレスに対して2バイト分のアドレス更新処理（アドレスを2加算する処理）が行われる。

10

【0543】

次いで、メインCPU101は、サム値の算出（減算）処理を行う（S125）。具体的には、メインCPU101は、HLレジスタに格納されている値（サム値の初期値又は前回の減算処理後のサム値）からDEレジスタに格納されている値を減算し、該減算された値をサム値としてHLレジスタに格納する。

【0544】

次いで、メインCPU101は、サム算出カウンタの値を1減算する（S126）。次いで、メインCPU101は、更新後のサム算出カウンタの値が「0」であるか否かを判別する（S127）。

20

【0545】

S127において、メインCPU101が、サム算出カウンタの値が「0」でないと判別したとき（S127がNO判定の場合）、メインCPU101は、処理をS124の処理に戻し、S124以降の処理を繰り返す。すなわち、メインRAM103の遊技用RAM領域の全域に渡ってサム値の減算処理が終了するまで、S124～S127の処理が繰り返される。

【0546】

一方、S127において、メインCPU101が、サム算出カウンタの値が「0」であると判別したとき（S127がYES判定の場合）、メインCPU101は、DEレジスタに、メインRAM103内の規定外RAM領域のサム値の計算開始アドレスをセットし、サム算出カウンタに、規定外用サムカウント値をセットする（S128）。なお、規定外用サムカウント値は、規定外RAM領域のバイト数となる。

30

【0547】

次いで、メインCPU101は、DEレジスタにセットされた規定外RAM領域のアドレスから1バイト分の領域のデータ（保存値）をAレジスタに読み出す（S129）。

【0548】

次いで、メインCPU101は、サム値の算出（減算）処理を行う（S130）。具体的には、メインCPU101は、HLレジスタに格納されている値からAレジスタに格納されている値を減算し、該減算された値をサム値としてHLレジスタに格納する。

40

【0549】

次いで、メインCPU101は、DEレジスタに格納されているアドレスを1加算し、サム算出カウンタの値を1減算する（S131）。次いで、メインCPU101は、更新後のサム算出カウンタの値が「0」であるか否かを判別する（S132）。

【0550】

S132において、メインCPU101が、サム算出カウンタの値が「0」でないと判別したとき（S132がNO判定の場合）、メインCPU101は、処理をS129の処理に戻し、S129以降の処理を繰り返す。すなわち、メインRAM103の規定外RAM領域の全域に渡ってサム値の減算処理が終了するまで、S129～S132の処理が繰り返される。

50

【0551】

一方、S132において、メインCPU101が、サム算出カウンタの値が「0」であると判別したとき（S132がYES判定の場合）、メインCPU101は、サムチェック処理の判定結果に「サム異常」をセットする（S133）。次いで、メインCPU101は、算出されたサム値が「0」であるか否かを判別する（S134）。

【0552】

なお、この処理では、メインCPU101は、フラグ・レジスタFのゼロフラグ（ビット6）の状態（1／0）を参照して、サム値が「0」であるか否かを判別する。本実施形態では、S128でセットされたサム算出カウンタの値が「0」になった時点、すなわち、メインRAM103の全域に渡ってサム値の減算処理が終了した時点において、サム値が「0」である場合には、フラグ・レジスタFのゼロフラグには「1」がセットされ、サム値が「0」でない場合には、フラグ・レジスタFのゼロフラグには「0」がセットされている。それゆえ、S134の処理の時点において、フラグ・レジスタFのゼロフラグに「1（オン状態）」がセットされていれば、メインCPU101はサム値が「0」であると判定する。

【0553】

S134において、メインCPU101が、算出されたサム値が「0」でないと判別したとき（S134がNO判定の場合）、メインCPU101は、後述のS139の処理を行う。一方、S134において、メインCPU101が、算出されたサム値が「0」であると判別したとき（S134がYES判定の場合）、メインCPU101は、判定結果に「電断異常」をセットする（S135）。

【0554】

次いで、メインCPU101は、電断発生フラグを取得する（S136）。次いで、メインCPU101は、電断発生フラグが電断なしの状態（オフ状態）であるか否かを判別する（S137）。

【0555】

S137において、メインCPU101が、電断発生フラグが電断なしの状態であると判別したとき（S137がYES判定の場合）、メインCPU101は、後述のS139の処理を行う。一方、S137において、メインCPU101が、電断発生フラグが電断なしの状態でないと判別したとき（S137がNO判定の場合）、メインCPU101は、判定結果に「正常」をセットする（S138）。

【0556】

S138の処理後、S134がNO判定の場合、又は、S137がYES判定の場合、メインCPU101は、サムチェック判定結果に判定結果を保存し、電断発生フラグをクリア（オフ）する（S139）。次いで、メインCPU101は、S121で規定外スタックエリアに保存されたスタックポインタ（SP）の値をスタックポインタにセットする（S140）。そして、S140の処理後、メインCPU101は、サムチェック処理を終了し、処理を電源投入時処理（図64参照）のS10の処理に移す。

【0557】

本実施形態では、上述のようにしてサムチェック処理が行われる。そして、上述したサムチェック処理中のS122～S132の処理は、メインCPU101が、図81のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【0558】

上述のように、本実施形態における、メインRAM103のサムチェックの判定処理では、まず、電断発生時に生成されたチェックサムの値を、電源復帰時のメインRAM103に格納されたデータで順次減算する。この際、遊技用RAM領域では、2バイト単位で減算処理（図81中のソースコード「SUB HL, DE」参照）が行われ、規定外RAM領域では、1バイト単位で減算処理（図81中のソースコード「SUBWB HL, A」参照）が行われる。次いで、最終的な減算結果が「0」であるか否か（ゼロフラグがオン状態であるか否か）に基づいて、異常の発生の有無を判定する。そして、減算結果が「0

10

20

30

40

50

」である場合（ゼロフラグがオン状態である場合）には、正常と判定され、減算結果が「0」でない場合（ゼロフラグがオフ状態である場合）には、異常と判定される。

【0559】

すなわち、本実施形態では、電断発生時のチェックサムの生成処理は加算方式で行われ、電源復帰時のチェックサムの判定処理は減算方式で行われる。そして、チェックサムの最終的な減算結果に基づいて、正常／異常の判定が行われる。このようなチェックサムの生成処理及び判定処理を行った場合、電源復帰時に再度チェックサムを生成して、該チェックサムを電断発生時のチェックサムと照合する処理が不要となる。この場合、ソースプログラム上において、照合命令コードを省略することができ、ソースプログラムの容量を低減することができる。この結果、本実施形態では、メインROM102において、照合命令コードの省略分に対応する空き容量を確保することができ、増えた空き容量を活用して、遊技性を高めることが可能になる。なお、上述の電断発生時のチェックサムの生成処理、及び、電源復帰時のチェックサムの判定処理で実行される「POP」命令は、スタックポインタ（SP）操作専用命令であり、ソースコード「POP DE」以外にもソースコード「POP HL」、「POP AF」等が存在する。

【0560】

【メインCPUの制御によるパチスロのメイン処理】

次に、図82を参照して、メインCPU101の制御により実行されるパチスロ1のメイン処理（主要動作処理）について説明する。なお、図82は、メイン処理の手順を示すフローチャート（以下、メインフローという）である。

【0561】

まず、メインCPU101は、RAM初期化処理を行う（S201）。この処理では、メインCPU101は、図12Cに示すメインRAM103の遊技用RAM領域内の「一遊技終了時」のアドレスを、初期化開始の先頭アドレスとして設定し、該先頭アドレスから遊技用RAM領域の最終アドレスまでの情報を消去（クリア）する。なお、この範囲の格納領域は、例えば、内部当籤役格納領域や表示役格納領域などの1回の単位遊技（ゲーム）ごとにデータの消去が必要な格納領域である。

【0562】

次いで、メインCPU101は、メダル受付・スタートチェック処理を行う（S202）。この処理では、メインCPU101は、メダルセンサ（不図示）やスタートスイッチ79などの入力チェック処理等を行う。なお、メダル受付・スタートチェック処理の詳細については、後述の図83を参照しながら後で説明する。

【0563】

次いで、メインCPU101は、乱数取得処理を行う（S203）。この処理では、メインCPU101は、内部当籤役抽籤用の乱数値（0～65535：ハードラッチ乱数となる乱数回路110の乱数レジスタ0の値）やART関連の各種抽籤で用いられる演出用乱数値（0～65535：ソフトラッチ乱数となる乱数回路110の乱数レジスタ1～3の各値、0～255：ソフトラッチ乱数となる乱数回路110の乱数レジスタ4～7の各値）などを抽出し、該抽出した各種乱数値をメインRAM103に設けられた乱数値格納領域（不図示）に格納する。なお、乱数取得処理の詳細については、後述の図91を参照しながら後で説明する。

【0564】

次いで、メインCPU101は、内部抽籤処理を行う（S204）。この処理では、メインCPU101は、S203で抽出した乱数値（ハードラッチ乱数）に基づいた抽籤により内部当籤役の決定処理を行う。なお、内部抽籤処理の詳細については、後述の図92を参照しながら後で説明する。

【0565】

次いで、メインCPU101は、図柄設定処理を行う（S205）。この処理では、メインCPU101は、例えば、当り要求フラグステータス（フラグステータス情報）から内部当籤役を生成する処理、当り要求フラグデータの展開処理、当り要求フラグデータを当

10

20

30

40

50

り要求フラグ格納領域を格納する処理等を行う。なお、図柄設定処理の詳細については、後述の図 97 を参照しながら後で説明する。

【0566】

次いで、メインCPU101は、スタートコマンド生成格納処理を行う（S206）。この処理では、メインCPU101は、副制御回路200に送信するスタートコマンドのデータを生成し、該コマンドデータをメインRAM103に設けられた通信データ格納領域（図75B参照）に保存する。通信データ格納領域に保存されたスタートコマンドは、後述の図158で説明する割込処理内の通信データ送信処理により、主制御回路90から副制御回路200に送信される。なお、スタートコマンドは、内部当籤役等を特定するパラメータ（サブフラグ等）を含んで構成される。

10

【0567】

次いで、メインCPU101は、第2インターフェースボード制御処理を行う（S207）。なお、第2インターフェースボード制御処理は、メインRAM103の規定外用作業領域で実行される。第2インターフェースボード制御処理の詳細については、後述の図102を参照しながら後で説明する。

【0568】

次いで、メインCPU101は、状態別制御処理を行う（S208）。この処理では、メインCPU101は、主に、遊技状態に応じた遊技開始時処理（スタート処理）を行う。なお、状態別制御処理の詳細については、後述の図104を参照しながら後で説明する。

【0569】

次いで、メインCPU101は、リール停止初期設定処理を行う（S209）。この処理では、メインCPU101は、リール停止初期設定テーブル（図26参照）を参照し、内部当籤役及び遊技状態に基づいて、引込優先順位テーブル選択テーブル番号、引込優先順位テーブル番号、停止テーブル番号を取得する処理や、ストップボタン未作動カウンタに「3」を格納する処理などを行う。

20

【0570】

次いで、メインCPU101は、リール回転開始処理を行う（S210）。この処理では、メインCPU101は、全リールの回転開始を要求する。そして、全リールの回転開始が要求されると、一定の周期（1.1172msec）で実行される後述の割込処理（後述の図158参照）により、3つのステッピングモータ（不図示）の駆動が制御され、左リール3L、中リール3C及び右リール3Rの回転が開始される。次いで、各リールは、その回転速度が定速度に達するまで加速制御され、その後、該定速度が維持されるように制御される。

30

【0571】

次いで、メインCPU101は、リール回転開始コマンド生成格納処理を行う（S211）。この処理では、メインCPU101は、副制御回路200に送信するリール回転開始コマンドのデータを生成し、該コマンドデータをメインRAM103に設けられた通信データ格納領域（図75B参照）に保存する。通信データ格納領域に保存されたリール回転開始コマンドは、後述の図158で説明する割込処理内の通信データ送信処理により、主制御回路90から副制御回路200に送信される。なお、リール回転開始コマンドは、リールの回転開始動作開始されたことを示すパラメータを含んで構成される。

40

【0572】

次いで、メインCPU101は、引込優先順位格納処理を行う（S212）。この処理では、メインCPU101は、引込優先順位データを取得して、引込優先順位データ格納領域に格納する。なお、引込優先順位格納処理の詳細については、後述の図126を参照しながら後で説明する。

【0573】

次いで、メインCPU101は、リール停止制御処理を行う（S213）。この処理では、メインCPU101は、左ストップボタン17L、中ストップボタン17C及び右ストップボタン17Rがそれぞれ押されたタイミングと内部当籤役とに基づいて該当するリー

50

ルの回転の停止制御を行う。なお、リール停止制御処理の詳細については、後述の図 1 3 8 を参照しながら後で説明する。

【0574】

次いで、メインCPU101は、入賞検索処理を行う（S214）。この処理では、メインCPU101は、図柄コード格納領域（図35参照）のデータを入賞作動フラグ格納領域（図28～図30参照）に格納する。また、この処理では、メインCPU101は、有効ラインに表示役が表示されたか否かを判定し、その判定結果に基づいて、メダルの払出枚数をセットする。なお、入賞検索処理の詳細については、後述の図145を参照しながら後で説明する。

【0575】

次いで、メインCPU101は、イリーガルヒットチェック処理を行う（S215）。この処理では、メインCPU101は、当り要求フラグ（内部当籤役）と入賞作動フラグ（表示役）とを合成し、その合成結果に基づいてイリーガルヒットエラーの有無を判定する。なお、イリーガルヒットチェック処理の詳細については、後述の図148を参照しながら後で説明する。

【0576】

次いで、メインCPU101は、入賞チェック・メダル払出処理を行う（S216）。この処理では、メインCPU101は、入賞作動コマンドの生成処理を行う。また、この処理では、メインCPU101は、S214において決定された表示役の払出枚数に基づいて、ホッパー装置51の駆動やクレジット枚数の更新を行い、メダルの払い出し処理を行う。なお、入賞チェック・メダル払出処理の詳細については、後述の図150を参照しながら後で説明する。

【0577】

次いで、メインCPU101は、BBチェック処理を行う（S217）。この処理では、メインCPU101は、ボーナス状態の作動及び終了を制御する。なお、BBチェック処理の詳細については、後述の図154を参照しながら後で説明する。

【0578】

次いで、メインCPU101は、RTチェック処理を行う（S218）。この処理では、メインCPU101は、有効ライン上に停止表示された図柄組合せに基づいてRT状態の移行制御を行う。なお、RTチェック処理の詳細については、後述の図155及び図156を参照しながら後で説明する。

【0579】

次いで、メインCPU101は、CZ・ART終了時処理を行う（S219）。この処理では、メインCPU101は、主に、CZの引き戻し抽籤処理を行う。なお、CZ・ART終了時処理の詳細については、後述の図157を参照しながら後で説明する。そして、S219の処理後（一遊技終了後）、メインCPU101は、処理をS201の処理に戻す。

【0580】

〔メダル受付・スタートチェック処理〕

次に、図83及び図84を参照して、メインフロー（図82参照）中のS202で行うメダル受付・スタートチェック処理について説明する。なお、図83は、メダル受付・スタートチェック処理の手順を示すフローチャートであり、図84は、メダル受付・スタートチェック処理中のS231～S233の処理を実行するためのソースプログラムの一例を示す図である。

【0581】

まず、メインCPU101は、自動投入メダルカウンタの値が「0」であるか否か（自動投入要求はあるか否か）を判別する（S221）。なお、この処理において、自動投入メダルカウンタが「1」以上であるときは、メインCPU101は、自動投入要求があると判別する。また、自動投入メダルカウンタは、前回の単位遊技において再遊技（リプレイ）に係る表示役が成立したか否かを識別するためのデータである。再遊技に係る表示役が

10

20

30

40

50

成立したときには、前回の単位遊技において投入された枚数分のメダルが自動投入メダルカウンタに自動的に投入される。

【0582】

S221において、メインCPU101が、自動投入メダルカウンタの値が「0」であると判別したとき（S221がYES判定の場合）、メインCPU101は、後述のS225の処理を行う。

【0583】

一方、S221において、メインCPU101が、自動投入メダルカウンタの値が「0」でないと判別したとき（S221がNO判定の場合）、メインCPU101は、メダル投入処理を行う（S222）。この処理では、メインCPU101は、メダル投入コマンドの生成格納処理やメダル投入枚数のLED点灯制御処理などを行う。なお、メダル投入処理の詳細については、後述の図85を参照しながら後で説明する。

【0584】

次いで、メインCPU101は、自動投入メダルカウンタの値を1減算する（S223）。次いで、減算後の自動投入メダルカウンタの値が「0」であるか否かを判別する（S224）。

【0585】

S224において、メインCPU101が、自動投入メダルカウンタの値が「0」でないと判別したとき（S224がNO判定の場合）、メインCPU101は、処理をS222の処理に戻し、S222以降の処理を繰り返す。

【0586】

一方、S224において、メインCPU101が、自動投入メダルカウンタの値が「0」であると判別したとき（S224がYES判定の場合）、又は、S221がYES判定の場合、メインCPU101は、メダル補助収納庫スイッチチェック処理を行う（S225）。この処理では、メインCPU101は、メダル補助収納庫スイッチ75のオン／オフ状態に基づいて、メダル補助収納庫52がメダルで満杯になっているか否かを検出する。

【0587】

次いで、メインCPU101は、メダル投入状態チェック処理を行う（S226）。次いで、メインCPU101は、メダル投入状態チェック処理の結果に基づいて、メダル投入可能な状態であるか否かを判別する（S227）。

【0588】

S227において、メインCPU101が、メダル投入可能な状態でないと判別したとき（S227がNO判定の場合）、メインCPU101は、後述のS231の処理を行う。

【0589】

一方、S227において、メインCPU101が、メダル投入可能な状態であると判別したとき（S227がYES判定の場合）、メインCPU101は、メダル投入チェック処理を行う（S228）。この処理では、メインCPU101は、例えば、メダルセンサ入力状態に基づいて、メダルが正常に通過したか否かの判定処理や、規定数を超えてメダル投入が行われた場合に該メダルをクレジットする処理などを行う。なお、メダル投入チェック処理の詳細については、後述の図87を参照しながら後で説明する。

【0590】

次いで、メインCPU101は、メダル投入チェック処理の結果に基づいて、メダル投入又はクレジット可能な状態であるか否かを判別する（S229）。

【0591】

S229において、メインCPU101が、メダル投入又はクレジット可能な状態であると判別したとき（S229がYES判定の場合）、メインCPU101は、後述のS231の処理を行う。一方、S229において、メインCPU101が、メダル投入又はクレジット可能な状態でないと判別したとき（S229がNO判定の場合）、メインCPU101は、メダル受付禁止の処理を行う（S230）。この処理により、セレクト66（図4参照）のソレノイドの駆動が行われず、投入されたメダルがメダル払出口24から排出

10

20

30

40

50

される。

【0592】

S230の処理後、S227がNO判定の場合、又は、S229がYES判定の場合、メインCPU101は、現在のメダルの投入枚数が遊技可能開始枚数であるか否かを判別する（S231）。なお、本実施形態では、遊技開始可能枚数は3枚である（図28～図30参照）。

【0593】

S231において、メインCPU101が、現在のメダルの投入枚数が遊技可能開始枚数であると判別したとき（S231がYES判定の場合）、メインCPU101は、後述のS234の処理を行う。一方、S231において、メインCPU101が、現在のメダルの投入枚数が遊技可能開始枚数でないと判別したとき（S231がNO判定の場合）、メインCPU101は、メダル投入があるか否かを判別する（S232）。

10

【0594】

S232において、メインCPU101が、メダル投入があると判別したとき（S232がYES判定の場合）、メインCPU101は、処理をS226に戻し、S226以降の処理を繰り返す。一方、S232において、メインCPU101が、メダル投入がないと判別したとき（S232がNO判定の場合）、メインCPU101は、図68で説明した設定変更確認処理を行う（S233）。この処理では、メインCPU101は、設定確認開始時の設定変更コマンドの生成格納処理などを行う。

【0595】

S233の処理後又はS231がYES判定の場合、メインCPU101は、スタートスイッチ79がオン状態であるか否かを判別する（S234）。

20

【0596】

S234において、メインCPU101が、スタートスイッチ79がオン状態でないと判別したとき（S234がNO判定の場合）、メインCPU101は、処理をS226に戻し、S226以降の処理を繰り返す。

【0597】

一方、S234において、メインCPU101が、スタートスイッチ79がオン状態であると判別したとき（S234がYES判定の場合）、メインCPU101は、メダル受付禁止の処理を行う（S235）。この処理により、セクタ66（図4参照）のソレノイドの駆動が行われず、投入されたメダルがメダル払出口24から排出される。そして、S235の処理後、メインCPU101は、メダル受付・スタートチェック処理を終了し、処理をメインフロー（図82参照）のS203に移す。

30

【0598】

本実施形態では、上述のようにしてメダル受付・スタートチェック処理が行われる。そして、上述したメダル受付・スタートチェック処理中のS231～S233の処理は、メインCPU101が、図84のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中でも、S233の処理では、メインCPU101専用命令コードである「CALLF」命令により、処理を設定変更確認処理の実行プログラムのアドレス「SB_WVSC_00」にジャンプさせ、図68及び図69で説明した設定変更確認処理を行う。

40

【0599】

そして、上述したメダル受付・スタートチェック処理中のS233の処理、すなわち、設定変更確認処理は、遊技状態に関係なく実行される。それゆえ、本実施形態では、遊技状態に関係なく、すなわち、遊技状態がボーナス状態（特賞作動状態）であっても、設定値及びホールメニュー（各種履歴データ（エラー、電断履歴等））を確認することができ、ゴト等の不正行為を抑制することができる。

【0600】

〔メダル投入処理〕

次に、図85及び図86を参照して、メダル受付・スタートチェック処理（図83参照）

50

中の S 2 2 2 で行うメダル投入処理について説明する。なお、図 8 5 は、メダル投入処理の手順を示すフローチャートであり、図 8 6 は、メダル投入処理を実行するためのソースプログラムの一例を示す図である。

【0601】

まず、メインCPU101は、メダルカウンタの値に「1」を加算する(S241)。なお、メダルカウンタは、メダルの投入枚数をカウント(計数)するためのカウンタであり、メインRAM103に設けられる。

【0602】

次いで、メインCPU101は、メダル投入コマンド生成格納処理を行う(S242)。この処理では、メインCPU101は、副制御回路200に送信するメダル投入コマンドのデータを生成し、該コマンドデータをメインRAM103に設けられた通信データ格納領域(図75B参照)に保存する。通信データ格納領域に保存されたメダル投入コマンドは、後述の図158で説明する割込処理内の通信データ送信処理により、主制御回路90から副制御回路200に送信される。すなわち、メダル投入コマンドは、メダルが1枚投入される度に、主制御回路90から副制御回路200に送信される。なお、メダル投入コマンドは、投入枚数等を特定するためのパラメータを含んで構成される。

【0603】

次いで、メインCPU101は、LED82(図7参照)に含まれるメダル投入枚数表示用の第1~第3LEDを消灯させる(S243)。次いで、メインCPU101は、メダル投入枚数(メダルカウンタの値)に基づいて、該メダル投入枚数に対応するLED点灯データ(点灯制御データ)を算出する(S244)。この処理において、例えば、メダル投入枚数が1枚である場合には、メダル投入枚数表示用の第1LEDのみを点灯させるLED点灯データが算出され、また、例えば、メダル投入枚数が3枚である場合には、メダル投入枚数表示用の第1~第3LEDの全てを点灯させるLED点灯データが算出される。なお、このLED点灯データの算出手法については、後で詳述する。

【0604】

次いで、メインCPU101は、算出されたLED点灯データを用いて、対応するメダル投入枚数表示用のLEDを点灯させる(S245)。そして、S245の処理後、メインCPU101は、メダル投入処理を終了し、処理をメダル受付・スタートチェック処理(図83参照)のS223に移す。

【0605】

本実施形態では、上述のようにしてメダル投入処理が行われる。なお、上述したメダル投入処理は、メインCPU101が、図86のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中でS244の処理では、メダル投入枚数表示用のLED点灯データがテーブルを参照したループ処理でなく、演算処理により生成される。この演算処理は、メインCPU101が図86に示すソースプログラム中のソースコード「LD A, L」~「OR L」を順次実行することにより行われる。

【0606】

この演算処理では、まず、ソースコード「LD A, L」の実行により、Lレジスタに格納されたメダル投入枚数のデータがAレジスタに格納される。例えば、メダル投入枚数が3枚である場合には、「00000011B」(10進数で「3」)がAレジスタに格納される。なお、本実施形態において、1バイトデータを「*****B」と記すが、最後の文字「B」は、文字「B」の前に示された「0」又は「1」がビットデータであることを意味する。

【0607】

次いで、ソースコード「ADD A, A」の実行により、Aレジスタに格納されたデータにAレジスタに格納されたデータが加算され、該加算結果がAレジスタに格納される。例えば、この「ADD」命令の実行前の時点でAレジスタに格納されているデータが「00000011B」である場合(メダル投入枚数が3枚である場合)、この「ADD」命令により、加算結果となる「00000110B」がAレジスタに格納される。

【0608】

次いで、ソースコード「DEC A」の実行により、Aレジスタに格納されているデータが1減算され、該減算結果がAレジスタに格納される。例えば、この「DEC」命令の実行前の時点でAレジスタに格納されているデータが「00000110B」である場合、この「DEC」命令により、減算結果となる「00000101B」がAレジスタに格納される。

【0609】

次いで、ソースコード「OR L」の実行により、Lレジスタに格納されているデータ（メダル投入枚数）とAレジスタに格納されているデータとの論理和演算が行われ、該演算結果がAレジスタに格納される。例えば、この「OR」命令の実行前の時点でAレジスタに格納されているデータが「00000101B」であり、Lレジスタに格納されているデータが「00000011B」である場合（メダル投入枚数が3枚である場合）、この「OR」命令により、両データの論理和演算の結果となる「00000111B」がAレジスタに格納される。そして、「OR」命令の実行によりAレジスタに格納されたデータが、メダル投入枚数表示用のLED点灯データ（メダル投入LEDの点灯状態を示すデータ）となる。

【0610】

例えば、メダル投入枚数が3枚である場合には、上述のように、S244のメダル投入枚数表示用のLED点灯データの算出処理により、最終的な算出結果「00000111B」がメダル投入枚数表示用のLED点灯データとなる。そして、本実施形態では、最終的に算出されたLED点灯データのビット0の「1/0」が1枚目のメダル投入枚数表示用のLED（第1LED）への出力ポートの「オン/オフ」状態に対応し、ビット1の「1/0」が2枚目のメダル投入枚数表示用のLED（第2LED）への出力ポートの「オン/オフ」状態に対応し、ビット2の「1/0」が3枚目のメダル投入枚数表示用のLED（第3LED）への出力ポートの「オン/オフ」状態に対応する。それゆえ、メダル投入枚数が3枚である場合には、上述のように、LED点灯データとして「00000111B」が生成されるので、メダル投入枚数表示用の第1～第3LEDの全ての出力ポートがオン状態にセットされ、メダル投入枚数表示用の第1～第3LEDが全て点灯状態となる。

【0611】

上述のようにしてメダル投入枚数表示用のLED点灯データを演算処理により生成した場合、メダル投入枚数表示用のLED点灯データを生成する際に参照するテーブルデータが不要となるのでメインROM102のテーブル領域の空き容量を増やすことができるとともに、プログラムの容量増を最小限に抑えることができる。すなわち、本実施形態の上述したメダル投入処理では、メダル投入LED表示の処理を効率化することができるとともに、メインROM102の空き容量を確保し（増やし）、該増えた空き領域を活用して、遊技性を高めることが可能になる。

【0612】

[メダル投入チェック処理]

次に、図87及び図88を参照して、メダル受付・スタートチェック処理（図83参照）中のS228で行うメダル投入チェック処理について説明する。なお、図87は、メダル投入チェック処理の手順を示すフローチャートであり、図88は、メダル投入チェック処理中のS255～S258の処理を実行するためのソースプログラムの一例を示す図である。

【0613】

まず、メインCPU101は、再遊技中であるか否かを判別する（S251）。

【0614】

S251において、メインCPU101が、再遊技中であると判別したとき（S251がYES判定の場合）、メインCPU101は、メダル投入チェック処理を終了し、処理をメダル受付・スタートチェック処理（図83参照）のS229に移す。

【0615】

一方、S 2 5 1において、メインCPU 1 0 1が、再遊技中でないと判別したとき（S 2 5 1がNO判定の場合）、メインCPU 1 0 1は、メダル受付許可を行う（S 2 5 2）。この処理では、セクタ 6 6（図 4 参照）のソレノイドの駆動が行われ、メダル投入口 1 4 から投入されたメダルが受け入れられる。受け入れられたメダルは計数されてからホッパー装置 5 1 へ案内される。

【0 6 1 6】

次いで、メインCPU 1 0 1は、ベットボタンチェック処理を行う（S 2 5 3）。この処理では、メインCPU 1 0 1は、BETスイッチ 7 7 のオン／オフ状態に基づいて、ベットボタン（MAXベットボタン 1 5 a 又は 1 ベットボタン 1 5 b）の操作が行われたか否かを判別する。次いで、メインCPU 1 0 1は、S 2 5 3 のベットボタンチェック処理の結果に基づいて、ベット動作が完了したか否かを判別する（S 2 5 4）。10

【0 6 1 7】

S 2 5 4において、メインCPU 1 0 1が、ベット動作が完了したと判別したとき（S 2 5 4がYES判定の場合）、メインCPU 1 0 1は、メダル投入チェック処理を終了し、処理をメダル受付・スタートチェック処理（図 8 3 参照）のS 2 2 9に移す。

【0 6 1 8】

一方、S 2 5 4において、メインCPU 1 0 1が、ベット動作が完了していないと判別したとき（S 2 5 4がNO判定の場合）、メインCPU 1 0 1は、現処理時のメダルセンサ入力状態（遊技媒体の受付状態）と、前回処理時のメダルセンサ入力状態とを取得する（S 2 5 5）。なお、メダルセンサ入力状態は、メダル投入口 1 4 に受け入れられたメダルのセクタ 6 6 内の通過状況を示す情報であり、セクタ 6 6 に入口及び出口に設けられた各メダルセンサ（不図示）の検知結果により生成される。20

【0 6 1 9】

本実施形態では、メダルセンサ入力状態は、1 バイト（8 ビット）のデータで表され、セクタ 6 6 の出口にメダルの通過方向に並んで設けられた上流側の第 1 メダルセンサ（不図示）の検知結果がビット 0 の情報（「0」又は「1」）に対応し、下流側の第 2 メダルセンサ（不図示）の検知結果がビット 1 の情報（「0」又は「1」）に対応する。第 1 メダルセンサによりメダルの通過が検知された場合には、ビット 0 に「1」がセットされ、第 2 メダルセンサによりメダルの通過が検知された場合には、ビット 1 に「1」がセットされる。それゆえ、メダルセンサ入力状態「0 0 0 0 0 0 0 0 B」は、メダル通過前又は通過後（通過時）の状態を示し、メダルセンサ入力状態「0 0 0 0 0 0 0 1 B」は、メダル通過開始時の状態を示し、メダルセンサ入力状態「0 0 0 0 0 0 1 1 B」は、メダル通過中の状態を示し、メダルセンサ入力状態「0 0 0 0 0 0 1 0 B」は、メダル通過完了直前の状態を示す。30

【0 6 2 0】

次いで、メインCPU 1 0 1は、現処理時のメダルセンサ入力状態が前回処理時のメダルセンサ入力状態から変化したか否かを判別する（S 2 5 6）。

【0 6 2 1】

S 2 5 6において、メインCPU 1 0 1が、現処理時のメダルセンサ入力状態が前回処理時のメダルセンサ入力状態から変化していないと判別したとき（S 2 5 6がNO判定の場合）、メインCPU 1 0 1は、後述のS 2 6 1の処理を行う。40

【0 6 2 2】

一方、S 2 5 6において、メインCPU 1 0 1が、現処理時のメダルセンサ入力状態が前回処理時のメダルセンサ入力状態から変化したと判別したとき（S 2 5 6がYES判定の場合）、メインCPU 1 0 1は、前回処理時のメダルセンサ入力状態に基づいて、演算処理により、現処理時で得られるメダルセンサ入力状態の正常値（正常変化値）を生成する（S 2 5 7）。

【0 6 2 3】

なお、この処理において、前回処理時のメダルセンサ入力状態が「0 0 0 0 0 0 0 0 B」である場合（第 1 及び第 2 メダルセンサがともにメダル未検知である場合）には、メダル50

センサ入力状態の正常変化値として「００００００１Ｂ」（第１メダルセンサがメダル検知であり、第２メダルセンサがメダル未検知である場合）が生成され、前回処理時のメダルセンサ入力状態が「００００００１Ｂ」である場合には、メダルセンサ入力状態の正常変化値として「００００００１１Ｂ」（第１及び第２メダルセンサがともにメダル検知である場合）が生成される。また、この処理において、前回処理時のメダルセンサ入力状態が「００００００１１Ｂ」である場合には、メダルセンサ入力状態の正常変化値として「００００００１０Ｂ」（第１メダルセンサがメダル未検知であり、第２メダルセンサがメダル検知である場合）が生成され、前回処理時のメダルセンサ入力状態が「００００００１０Ｂ」である場合には、メダルセンサ入力状態の正常変化値として「００００００００Ｂ」（第１及び第２メダルセンサがともにメダル未検知である場合）が生成される。なお、メダルセンサ入力状態の正常変化値の生成（算出）手法については後で詳述する。

10

【０６２４】

次いで、メインＣＰＵ１０１は、現処理時のメダルセンサ入力状態がＳ２５７で生成された正常変化値と同じであるか否かを判別する（Ｓ２５８）。なお、この判定処理では、メダル逆行エラーの発生の有無が判定され、Ｓ２５８の判定条件が満たされない場合には、メインＣＰＵ１０１は、メダル逆行エラーが発生したと判定する。

【０６２５】

Ｓ２５８において、メインＣＰＵ１０１が、現処理時のメダルセンサ入力状態がＳ２５７で生成された正常変化値と同じでないと判別したとき（Ｓ２５８がＮＯ判定の場合）、メインＣＰＵ１０１は、後述のＳ２６２の処理を行う。

20

【０６２６】

一方、Ｓ２５８において、メインＣＰＵ１０１が、現処理時のメダルセンサ入力状態がＳ２５７で生成された正常変化値と同じであると判別したとき（Ｓ２５８がＹＥＳ判定の場合）、メインＣＰＵ１０１は、現処理時のメダルセンサ入力状態がメダル通過時の状態（「００００００００Ｂ」）であるか否かを判別する（Ｓ２５９）。Ｓ２５９において、メインＣＰＵ１０１が、現処理時のメダルセンサ入力状態がメダル通過時の状態であると判別したとき（Ｓ２５９がＹＥＳ判定の場合）、メインＣＰＵ１０１は、後述のＳ２６３の処理を行う。

【０６２７】

Ｓ２５９において、メインＣＰＵ１０１が、現処理時のメダルセンサ入力状態がメダル通過時の状態でないと判別したとき（Ｓ２５９がＮＯ判定の場合）、メインＣＰＵ１０１は、メダル通過チェックタイマーをセットする（Ｓ２６０）。この処理でメダル通過チェックタイマーにセットされる時間は、メダルがセレクト６６を通過したか否かを判別可能な時間であれば、任意の時間に設定することができる。また、この処理でセットされるタイマー値は、例えば、現処理時のメダルセンサ入力状態に応じて変化させてもよい。

30

【０６２８】

Ｓ２６０の処理後又はＳ２５６がＮＯ判定の場合、メインＣＰＵ１０１は、現処理時のメダルセンサ入力状態がメダル通過中の状態（「００００００１１Ｂ」）であり、かつ、メダル通過チェックタイマーが停止しているか否かを判別する（Ｓ２６１）。この判定処理では、メダル通過エラー（投入メダル通過時間エラー）の発生の有無が判定され、Ｓ２６１の判定条件が満たされた場合、メインＣＰＵ１０１は、メダル通過エラーが発生したと判定する。

40

【０６２９】

Ｓ２６１において、メインＣＰＵ１０１が、Ｓ２６１の判定条件が満たされないと判別したとき（Ｓ２６１がＮＯ判定の場合）、メインＣＰＵ１０１は、処理をＳ２５３の処理に戻し、Ｓ２５３以降の処理を繰り返す。

【０６３０】

一方、Ｓ２６１において、メインＣＰＵ１０１が、Ｓ２６１の判定条件が満たされると判別したとき（Ｓ２６１がＹＥＳ判定の場合）、又は、Ｓ２５８がＮＯ判定の場合、すなわち、メダル通過エラー又はメダル逆行エラーが発生したと判定された場合、メインＣＰＵ

50

101は、エラー処理を行う（S262）。この処理では、メインCPU101は、例えば、エラーコマンド生成格納処理等のエラー発生時の各種処理を行う。なお、エラー処理の詳細については、後述の図89を参照しながら後で説明する。そして、S262の処理後、メインCPU101は、処理をS253の処理に戻し、S253以降の処理を繰り返す。

【0631】

ここで再度、S259の処理に戻って、S259がYES判定の場合、メインCPU101は、規定数（本実施形態では3枚）のメダルが投入済みの状態であるか否かを判別する（S263）。

【0632】

S263において、メインCPU101が、規定数のメダルが投入済みの状態でないと判別したとき（S263がNO判定の場合）、メインCPU101は、図85で説明したメダル投入処理を行う（S264）。そして、S264の処理後、メインCPU101は、処理をS253の処理に戻し、S253以降の処理を繰り返す。

【0633】

一方、S263において、メインCPU101が、規定数のメダルが投入済みの状態であると判別したとき（S263がYES判定の場合）、メインCPU101は、クレジットカウンタの値に「1」を加算する（S265）。次いで、メインCPU101は、メダル投入コマンド生成格納処理を行う（S266）。この処理では、メインCPU101は、副制御回路200に送信するメダル投入コマンドのデータを生成し、該コマンドデータをメインRAM103に設けられた通信データ格納領域（図75B参照）に保存する。通信データ格納領域に保存されたメダル投入コマンドは、後述の図158で説明する割込処理内の通信データ送信処理により、主制御回路90から副制御回路200に送信される。

【0634】

次いで、メインCPU101は、クレジットカウンタの値に基づいて、メダルのクレジット枚数が上限値（本実施形態では50枚）であるか否かを判別する（S267）。

【0635】

S267において、メインCPU101が、メダルのクレジット枚数が上限値でないと判別したとき（S267がNO判定の場合）、メインCPU101は、処理をS253の処理に戻し、S253以降の処理を繰り返す。一方、S267において、メインCPU101が、メダルのクレジット枚数が上限値であると判別したとき（S267がYES判定の場合）、メインCPU101は、メダル投入チェック処理を終了し、処理をメダル受付・スタートチェック処理（図83参照）のS229に移す。

【0636】

本実施形態では、上述のようにしてメダル投入チェック処理が行われる。そして、上述したメダル投入チェック処理中のS255～S258の処理は、メインCPU101が、図88のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中で、S257のメダルセンサ入力状態の正常変化値の生成処理は、テーブルを参照して取得する処理ではなく、演算処理により行われる。具体的には、正常変化値の生成処理は、メインCPU101が図88に示すソースプログラム中のソースコード「RLA」及び「AND cBX__MDINSW」をこの順で実行することにより行われる。

【0637】

「RLA」命令は、Aレジスタに格納された1バイトのデータを、左（ビット0からビット7に向かう方向）に1回（1ビット分）シフトさせる命令コードである。図88に示す例では、「RLA」命令より前に実行されるソースコード「LD A, B」によりAレジスタに格納された前回のメダルセンサ入力状態を示す1バイトのデータが、「RLA」命令により、左に1回シフトされる。この際、ビット0に新たに格納されるビットデータは、「RLA」命令より前に実行されるソースコード「CP cBX__MDINSW2」の実行結果に基づいて決定される。

【0638】

10

20

30

40

50

「CP」命令は比較動作を実行する命令コードである。また、「cBX_MDISW2」は、1バイトのデータであり、本実施形態では「00000010B」である。ソースコード「CP cBX_MDISW2」が実行されると、Aレジスタに格納された前回のメダルセンサ入力状態を示す1バイトのデータが、「cBX_MDISW2 (00000010B)」と比較される。

【0639】

そして、ソースコード「CP cBX_MDISW2」を実行した結果、前回のメダルセンサ入力状態を示す1バイトのデータが「cBX_MDISW2 (00000010B)」未満であるという結果が得られた場合にはフラグ・レジスタFのキャリーフラグ(図11参照)に「1」がセットされ、「RLA」命令の実行時に、Aレジスタのビット0にフラグ・レジスタFのキャリーフラグの「1」が格納される。一方、ソースコード「CP cBX_MDISW2」を実行した結果、前回のメダルセンサ入力状態を示す1バイトのデータが「cBX_MDISW2 (00000010B)」以上であるという結果が得られた場合にはフラグ・レジスタFのキャリーフラグ(図11参照)に「0」がセットされ、「RLA」命令の実行時により、Aレジスタのビット0にフラグ・レジスタFのキャリーフラグの「0」が格納される。

【0640】

それゆえ、例えば、前回のメダルセンサ入力状態を示す1バイトのデータが「00000000B(メダル通過前又は通過後(通過時)の状態)」(<「cBX_MDISW2」)であれば、「RLA」命令の実行により、「00000001B」が生成され、前回のメダルセンサ入力状態を示す1バイトのデータが「00000001B(メダル通過開始時の状態)」(<「cBX_MDISW2」)であれば、「RLA」命令の実行により、「00000011B」が生成される。一方、例えば、前回のメダルセンサ入力状態を示す1バイトのデータが「00000011B(メダル通過中の状態)」(>「cBX_MDISW2」)であれば、「RLA」命令の実行により、「00000110B」が生成され、前回のメダルセンサ入力状態を示す1バイトのデータが「00000010B(メダル通過完了直前の状態)」(=「cBX_MDISW2」)であれば、「RLA」命令の実行により、「00000100B」が生成される。

【0641】

次いで、ソースコード「AND cBX_MDINSW」が実行されると、「RLA」命令の実行により生成された1バイトデータ(Aレジスタの格納データ)が、1バイトのデータ「cBX_MDINSW」と論理積され、メダルセンサ入力状態の正常変化値が算出される。なお、1バイトのデータ「cBX_MDINSW」は、本実施形態では「00000011B」である。それゆえ、ソースコード「AND cBX_MDINSW」が実行されれば、Aレジスタの格納データ中のビット0及びビット1のデータだけがマスクされ、その他のビットデータが「0」になる。

【0642】

その結果、例えば、前回のメダルセンサ入力状態を示す1バイトのデータが「00000000B(メダル通過前の状態)」であれば、メダルセンサ入力状態の正常変化値として「00000001B(メダル通過開始時の状態)」が生成され、前回のメダルセンサ入力状態を示す1バイトのデータが「00000001B(メダル通過開始時の状態)」であれば、メダルセンサ入力状態の正常変化値として「00000011B(メダル通過中の状態)」が生成される。一方、例えば、前回のメダルセンサ入力状態を示す1バイトのデータが「00000011B(メダル通過中の状態)」であれば、メダルセンサ入力状態の正常変化値として「00000010B(メダル通過完了直前の状態)」が生成され、前回のメダルセンサ入力状態を示す1バイトのデータが「00000010B(メダル通過完了直前の状態)」であれば、メダルセンサ入力状態の正常変化値として「00000000B(メダル通過後(通過時)の状態)」が生成される。

【0643】

上述のようにして、メダルセンサ入力状態の変化態様の検知処理をテーブル参照処理から

10

20

30

40

50

演算処理に変更することにより、メインROM102のテーブル格納領域の空き容量を増やすことができるとともに、プログラムの容量増を最小限に抑えることができる。それゆえ、上述した手法を採用することにより、メダル投入センサ状態の検知処理を効率化することができるのと同時に、メインROM102において増えた空き容量を活用して、遊技性を高めることが可能となる。

【0644】

〔エラー処理〕

次に、図89及び図90を参照して、例えば、メダル投入チェック処理（図87参照）中のS262で行うエラー処理について説明する。図89は、エラー処理の手順を示すフローチャートであり、図90は、エラー処理のソースプログラム上で、実際に参照されるエラーテーブルの構成の一例を示す図である。

10

【0645】

なお、図90に示すエラーテーブルでは、エラー要因の種別を示すポートのオン／オフ状態を表す1バイトデータ（図90中の例えばアドレス「dERR_HE」に格納されている1バイトデータ等）毎に、エラー表示データ（図90中の例えばアドレス「dERR_HE+1」及び「dERR_HE+2」に格納されている1バイトデータ等）が規定される。このエラー表示データは、情報表示器6に含まれる2桁の7セグLED（払出枚数表示用及びエラー表示用兼用）に出力される。

【0646】

まず、メインCPU101は、メダルソレノイドのオフ処理を行う（S271）。具体的には、メインCPU101は、セクタ66（図7参照）のソレノイドの駆動を停止する。次いで、メインCPU101は、メダルの払出枚数表示データの退避処理を行う（S272）。

20

【0647】

次いで、メインCPU101は、エラーテーブルのセット処理を行う（S273）。この処理により、図90に示すエラーテーブルの先頭アドレスがソースプログラム上にセットされる。

【0648】

次いで、メインCPU101は、エラー要因を取得する（S274）。なお、この処理で取得されるエラー要因は、現在処理中のエラー処理を読み出した処理に応じて変化する。なお、本実施形態で対象とするエラー要因としては、図90に示すように、「ホッパー EMPTYエラー」、「ホッパージャムエラー」、「投入メダル通過カウントエラー」、「投入メダル通過チェックエラー」、「投入メダル通過チェックエラー」、「投入メダル通過時間エラー」、「投入メダル逆行エラー」、「投入メダル補助収納庫満杯エラー」、「イリーガルヒットエラー」が規定される。例えば、メダル投入チェック処理中のS258の処理後にエラー処理が読み出された場合には、この処理において、エラー要因として図90中の「投入メダル逆行エラー（Cr）」が取得される。また、例えば、メダル投入チェック処理中のS261の処理後にエラー処理が読み出された場合には、この処理において、エラー要因として図90中の「投入メダル通過時間エラー（CE）」が取得される。

30

【0649】

次いで、メインCPU101は、エラーテーブルとエラー要因とから、エラー表示データを取得する（S275）。例えば、エラー要因が「投入メダル逆行エラー（Cr）」である場合、この処理において、2桁の7セグLEDのうち、上位桁の7セグLEDに出力するエラー表示データとして、図90に示すエラーテーブル中のアドレス「dERR_CR+1」に格納されている1バイトデータ「01001110B」が取得され、下位桁の7セグLEDに出力するエラー表示データとして、アドレス「dERR_CR+2」に格納されている1バイトデータ「00001001B」が取得される。この場合、2桁の7セグLEDには、「Cr」の2文字がエラー情報として表示される。

40

【0650】

次いで、メインCPU101は、エラーコマンド（発生）生成格納処理を行う（S276

50

）。この処理では、メインCPU101は、副制御回路200に送信する、エラー発生時のエラーコマンドのデータを生成し、該コマンドデータをメインRAM103に設けられた通信データ格納領域（図75B参照）に保存する。通信データ格納領域に保存されたエラー発生時のエラーコマンドは、後述の図158で説明する割込処理内の通信データ送信処理により、主制御回路90から副制御回路200に送信される。なお、エラー発生時のエラーコマンドには、エラー発生を示すパラメータを含んで構成される。

【0651】

次いで、メインCPU101は、1割込時間（1.1172ms）の待機処理を行う（S277）。次いで、メインCPU101は、エラーが解除されたか否かを判別する（S278）。

【0652】

S278において、メインCPU101が、エラーが解除されていないと判別したとき（S278がNO判定の場合）、メインCPU101は、処理をS277の処理に戻し、S277以降の処理を繰り返す。

【0653】

一方、S278において、メインCPU101が、エラーが解除されたと判別したとき（S278がYES判定の場合）、メインCPU101は、エラー要因のクリア処理を行う（S279）。なお、この処理は、メインRAM103の規定外作業領域で行われる。次いで、メインCPU101は、S272で退避させたメダルの払出枚数表示データの復帰処理を行う（S280）。

【0654】

次いで、メインCPU101は、エラーコマンド（解除）生成格納処理を行う（S281）。この処理では、メインCPU101は、副制御回路200に送信する、エラー解除時のエラーコマンドのデータを生成し、該コマンドデータをメインRAM103に設けられた通信データ格納領域（図75B参照）に保存する。通信データ格納領域に保存されたエラー解除時のエラーコマンドは、後述の図158で説明する割込処理内の通信データ送信処理により、主制御回路90から副制御回路200に送信される。なお、エラー解除時のエラーコマンドには、エラー解除を示すパラメータを含んで構成される。そして、S281の処理後、メインCPU101は、エラー処理を終了し、処理を例えばメダル投入チェック処理（図87参照）中のS253に移す。なお、エラー解除では、発生したエラー要因が解除され、リセットスイッチ76が押下されることにより、エラー状態が解除される。

【0655】

〔乱数取得処理〕

次に、図91を参照して、メインフロー（図82参照）中のS203で行う乱数取得処理について説明する。なお、図91は、乱数取得処理の手順を示すフローチャートである。

【0656】

まず、メインCPU101は、乱数回路の乱数レジスタ0のハードラッチ乱数（0～65535）を取得し、取得した乱数値を内部当籤役抽籤用の乱数値として、メインRAM103内の乱数値格納領域（不図示）に保存する（S291）。

【0657】

次いで、メインCPU101は、乱数回路の乱数レジスタ1～7のソフトラッチ乱数（0～65535：ART関連の抽籤処理で用いられる演出用乱数値、0～255：1バイト抽籤処理で乱数値）を生成するためのソフトラッチ乱数取得レジスタのセット処理を行う（S292）。次いで、メインCPU101は、ソフトラッチ乱数の取得個数（例えば、7）をセットする（S293）。

【0658】

次いで、メインCPU101は、取得個数分のソフトラッチ乱数を一括で取得し、取得個数分のソフトラッチ乱数を乱数値格納領域に保存する（S294）。なお、この際、乱数回路110の乱数レジスタ1から取得されるソフトラッチ乱数（演出用乱数値、2バイト乱数値）は、乱数値格納領域内において、乱数回路の乱数レジスタ0から取得されるハー

10

20

30

40

50

ドラッチ乱数（内部当籤役抽籤用の乱数値）が格納された領域とは異なる領域に保存される。そして、S294の処理後、メインCPU101は、乱数取得処理を終了し、処理をメインフロー（図82参照）のS204に移す。なお、本実施形態では、4つの2バイト乱数と、4つの1バイト乱数を格納するために、メインRAM103に12バイトの格納領域が乱数格納領域として割り当てられている。

【0659】

〔内部抽籤処理〕

次に、図92～図96を参照して、メインフロー（図82参照）中のS204で行う内部抽籤処理について説明する。なお、図92は、内部抽籤処理の手順を示すフローチャートであり、図93Aは、内部抽籤処理中のS302～S305の処理を実行するためのソースプログラムの一例を示す図であり、図93Bは、内部抽籤処理中のS308～S309の処理を実行するためのソースプログラムの一例を示す図である。

【0660】

また、図94は、内部抽籤処理のソースプログラム上で、実際に参照される内部抽籤テーブル（一般遊技中用）の構成の一例を示す図であり、図95は、内部抽籤処理のソースプログラム上で、実際に参照されるRT状態別抽籤値選択テーブルの構成の一例を示す図である。さらに、図96は、内部抽籤処理のソースプログラム上で、実際に参照される、内部抽籤値テーブル選択テーブル、1バイト内部抽籤値テーブル、2バイト内部抽籤値テーブル、1バイト設定別内部抽籤値テーブル及び2バイト設定別内部抽籤値テーブルの構成の一例を示す図である。なお、本実施形態では、RB（BB）中用の内部抽籤テーブルも設けられているが、ここでは、内部抽籤処理のソースプログラム上で参照されるRB中用の内部抽籤テーブルの構成の図示は省略する。

【0661】

まず、メインCPU101は、設定値・メダル投入枚数チェック処理を行う（S301）。この処理では、メインCPU101は、現遊技の設定値（1～6のいずれか）及びメダル投入枚数（本実施形態では3枚）のチェック処理を行う。

【0662】

次いで、メインCPU101は、一般遊技中用の内部抽籤テーブル及び抽籤回数（本実施形態では53回）をセットする（S302）。この処理では、図94に示す内部抽籤テーブル（一般遊技中用）中の「特賞当籤番号+小役当籤番号」の値（当り要求フラグステータス）がCレジスタにセットされ、「抽籤値選択テーブルor抽籤係数テーブル」の値（判定データ：アドレスに関するデータ）がAレジスタにセットされる。なお、当り要求フラグステータスは、図94に示すように、特賞当籤番号（「00H（はずれ）」、「01H（BB1）」、「02H（BB2）」：10進数で0、1、2）に「25H（16進数：10進数では37（後述の特賞番号）」）が乗算された値に、小役当籤番号（「00H」～「24H」：10進数で0～36）を加算した値である。

【0663】

次いで、メインCPU101は、RB作動中であるか否かを判別する（S303）。S303において、メインCPU101が、RB作動中でないと判別したとき（S303がNO判定の場合）、メインCPU101は、後述のS305の処理を行う。

【0664】

一方、S303において、メインCPU101が、RB作動中であると判別したとき（S303がYES判定の場合）、メインCPU101は、RB中用の内部抽籤テーブル及び抽籤回数（本実施形態では5回）をセットする（S304）。この処理では、S302でセットされた一般遊技中用の内部抽籤テーブル及び抽籤回数をRB中用の内部抽籤テーブル及び抽籤回数で上書きする。

【0665】

S304の処理後又はS303がNO判定の場合、メインCPU101は、セットされている内部抽籤テーブルから抽籤対象役の判定データ（アドレスに関するデータ）を取得し、抽籤テーブルアドレスを更新する（S305）。

【0666】

次いで、メインCPU101は、判定データがRT状態別データであるか否かを判別する（S306）。この処理では、メインCPU101は、現在取得されている抽籤対象役がRT状態に応じて抽籤値が変化する内部当籤役であるか否かを判別する。具体的には、メインCPU101は、現在取得されている抽籤対象役の判定データに規定されているアドレスが、図95に示すRT状態別抽籤値選択テーブル内のアドレスであるか否かを判別する。

【0667】

例えば、図94の内部抽籤テーブルにおいて内部当籤役「F__チリリプ」に対応付けられている判定データ「(dRPPTRO1-dRTRB__SEL)*2+001H」では、図95に示すRT状態別抽籤値選択テーブル内の内部当籤役「F__チリリプ」のアドレス「dRPPTRO1」が規定されているので、内部当籤役「F__チリリプ」に対応付けられている判定データは、RT状態別データに対応する。それゆえ、現在取得されている抽籤対象役が内部当籤役「F__チリリプ」である場合には、S306の処理において、メインCPU101は、判定データがRT状態別データであると判定する。

【0668】

S306において、メインCPU101が、判定データがRT状態別データでないと判別したとき（S306がNO判定の場合）、メインCPU101は、後述のS308の処理を行う。一方、S306において、メインCPU101が、判定データがRT状態別データであると判別したとき（S306がYES判定の場合）、メインCPU101は、判定データに基づいて、図95に示すRT状態抽籤値選択テーブルから選択データを取得し、該取得した選択データを判定データにセットする（S307）。

【0669】

S307の処理後又はS306がNO判定の場合、メインCPU101は、抽籤対象役の判定データが設定別データであるか否かを判別する（S308）。この処理では、メインCPU101は、現在取得されている抽籤対象役が、設定値に応じて抽籤値が変化する内部当籤役であるか否かを判別する。具体的には、メインCPU101は、現在取得されている抽籤対象役の判定データに規定されているアドレスが、図96に示す1バイト設定別内部抽籤値テーブル又は2バイト設定別内部抽籤値テーブル内のアドレスであるか否かを判別する。

【0670】

例えば、図94の内部抽籤テーブルにおいて内部当籤役「F__強チリ1」に対応付けられている判定データ「((dNMLB00F289-dPRB__DB__WV)/06H)*2+080H」では、図96に示す1バイト設定別内部抽籤値テーブル内の内部当籤役「F__強チリ1」のアドレス「dNMLB00F289」が規定されているので、内部当籤役「F__強チリ1」に対応付けられている判定データは、設定別データに対応する。それゆえ、現在取得されている抽籤対象役が内部当籤役「F__強チリ1」である場合には、S308の処理において、メインCPU101は、判定データが設定別データであると判定する。

【0671】

S308において、メインCPU101が、判定データが設定別データでないと判別したとき（S308がNO判定の場合）、メインCPU101は、後述のS310の処理を行う。一方、S308において、メインCPU101が、判定データが設定別データであると判別したとき（S308がYES判定の場合）、メインCPU101は、判定データに設定値データ（0～5にいずれか）を加算し、該加算した値を判定データにセットする（S309）。なお、この処理で判定データに加算される設定値データは、設定値に対応付けられたデータであるが、設定値そのものの値ではなく、設定値データ「0」～「5」は、それぞれ「設定1」～「設定6」に対応するデータである。

【0672】

S309の処理後又はS308がNO判定の場合、メインCPU101は、セットされて

10

20

30

40

50

いる判定データ（アドレスデータ）に基づいて、抽籤対象役の抽籤値が格納された領域のアドレスを算出し、該アドレスに格納された抽籤値を取得する（S310）。

【0673】

S310の処理において、例えば、抽籤対象役が、その抽籤値がRT状態及び設定値の両方に依存しない内部当籤役「F__サボ2」である場合には、図96に示す1バイト内部抽籤値テーブルからアドレス「dNM__B00F26」に格納された抽籤値「128」が取得される。また、例えば、抽籤対象役が、その抽籤値がRT状態により変化する内部当籤役「F__RT3リップ__1st」であり、RT状態がRT2状態である場合には、図96に示す2バイト内部抽籤値テーブルからアドレス「dRT2B00F13456」に格納された抽籤値「1800」が取得される。

10

【0674】

また、S310の処理において、例えば、抽籤対象役が、その抽籤値が設定値により変化する内部当籤役「F__強チリ1」である場合には、図96に示す1バイト設定別内部抽籤値テーブルに規定されている6種類の抽籤値「150（設定1）、150（設定2）、150（設定3）、150（設定4）、160（設定5）、170（設定6）」の中から設定値に対応する抽籤値が取得される。この際、設定値に対応する抽籤値の取得は、判定データに設定値データを加算（S309の処理）して求められたアドレスを指定することにより取得される。それゆえ、例えば、抽籤対象役が「F__強チリ1」であり、設定値が「6」（設定値データが「5」）である場合には、アドレス「dNMLB00F289+5」に格納された抽籤値「170」が取得される。

20

【0675】

なお、本実施形態では、例えば内部当籤役「F__維持リップA」のように、その抽籤値がRT状態及び設定値の両方に依存する役の場合には、内部抽籤値テーブル及び設定別内部抽籤値テーブルの両方を参照して、抽籤値が取得される。

【0676】

次いで、メインCPU101は、乱数格納領域に格納された内部当籤役抽籤用の乱数値（0～65535のいずれか）を取得する（S311）。

【0677】

次いで、メインCPU101は、抽籤実行処理を行う（S312）。この処理では、メインCPU101は、S310で取得された抽籤値に、S311で取得された乱数値を加算し、その加算結果を抽籤結果（抽籤対象役の当籤／非当籤）とする。なお、この抽籤実行処理において、抽籤値と乱数値との和が65535を超えた場合（オーバーフローした場合）、抽籤対象役が当籤した（抽籤対象役が内部当籤役として決定された）と判定される。

30

【0678】

次いで、メインCPU101は、乱数値に抽籤値を加算した値（抽籤実行後の乱数値）を新たな乱数値として、乱数格納領域に保存する（S313）。次いで、メインCPU101は、抽籤実行処理で当籤したか否か（オーバーフローが発生したか否か）を判別する（S314）。

【0679】

S314において、メインCPU101が、抽籤実行処理で当籤したと判別したとき（S314がYES判定の場合）、メインCPU101は、内部抽籤テーブルを参照して当籤した内部当籤役に対応する当り要求フラグステータス（「特賞当籤番号＋小役当籤番号」の値）を取得する（S315）。例えば、一般遊技中において、抽籤対象役が「F__確チリリップ」であるときの抽籤実行処理で当籤した場合、S315の処理では、当り要求フラグステータス「(00H×25H)+02H」（特賞当籤番号＝0、小役当籤番号＝2）が取得される。そして、S315の処理後、メインCPU101は、内部抽籤処理を終了し、処理をメインフロー（図82参照）のS205に移す。

40

【0680】

一方、S314において、メインCPU101が、抽籤実行処理で当籤していないと判別したとき（S314がNO判定の場合）、メインCPU101は、内部抽籤テーブルにお

50

いて抽籤対象役を次の役に更新し、抽籤回数を1減算する（S316）。次いで、メインCPU101は、減算後の抽籤回数が「0」であるか否かを判別する（S317）。

【0681】

S317において、メインCPU101が、減算後の抽籤回数が「0」でないと判別したとき（S317がNO判定の場合）、メインCPU101は、処理をS305の処理に戻し、S305以降の処理を繰り返す。

【0682】

一方、S317において、メインCPU101が、減算後の抽籤回数が「0」であると判別したとき（S317がYES判定の場合）、すなわち、内部当籤役が「はずれ」である場合、メインCPU101は、ハズレステータスをセットする（S318）。なお、「ハズレステータス」は、特賞当籤番号及び小役当籤番号のいずれもが「0」となる当り要求フラグステータスに対応する。そして、S318の処理後、メインCPU101は、内部抽籤処理を終了し、処理をメインフロー（図82参照）のS205に移す。

【0683】

本実施形態では、上述のようにして内部抽籤処理が行われる。なお、上述した内部抽籤処理中のS302～S305の処理は、メインCPU101が、図93Aのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中でも、S305の判定データの取得処理は、図93A中のソースコード「LDIN AC, (HL)」により実行される。

【0684】

ソースプログラム上において、例えば、ソースコード「LDIN ss, (HL)」が実行されると、HLレジスタ（ペアレジスタ）にセットされたアドレス及び該アドレスに「1」を加算したアドレスで指定されるメモリの内容（データ）が、ss（BC、DE、AC、AE又はBD）ペアレジスタにロードされるとともに、HLレジスタにセットされているアドレスが+2更新（2加算）される。それゆえ、図93A中のソースコード「LDIN AC, (HL)」が実行されると、HLレジスタ（ペアレジスタ）にセットされたアドレス及び該アドレスに1加算したアドレスで指定されるメモリの内容（データ）が、ACレジスタにロードされるとともに、HLレジスタにセットされているアドレスが+2更新（2加算）される。なお、S305の判定データの取得処理では、上述のように、この「LDIN」命令（所定の読み出し命令）により、Aレジスタに、判定データ（「抽籤値選択テーブルor抽籤係数テーブル」の値）が格納され、Cレジスタに当り要求フラグステータス（「特賞当籤番号+小役当籤番号」の値）が格納される。

【0685】

上述のように、内部抽籤処理中のS305の判定データの取得処理では、一つの命令コード（「LDIN」命令）により、データのロード処理及びアドレスの更新処理の両方を行うことができる。この場合、ソースプログラム上において、アドレス設定に係る命令コードを省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【0686】

また、上述した内部抽籤処理中のS308及びS309の処理は、メインCPU101が、図93Bのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中で、S309の設定値データ（0～5のいずれか）の加算処理は、メインCPU101が図93B中のソースコード「MUL A, 6」及び「ADDQ A, (.LOW.wWAVENUM)」をこの順で実行することにより行われる。なお、「MUL」命令及び「ADDQ」命令はともに、メインCPU101専用命令コードであり、「ADDQ」命令は、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行うメインCPU101専用命令コードである。

【0687】

10

20

30

40

50

ソースプログラム上において、例えば、ソースコード「MUL A, n」が実行されると、Aレジスタの格納データと、1バイトの整数nとを乗算し、その乗算結果をAレジスタに格納する。それゆえ、図93B中のソースコード「MUL A, 6」では、Aレジスタの内容（格納データ）に、1バイトの整数6が乗算され、その乗算結果がAレジスタに格納される。なお、この乗算処理は、マイクロプロセッサ91に含まれる演算回路107（図9参照）により実行される。すなわち、本実施形態のパチスロ1では、ソースプログラム上における乗算処理及び除算処理を実行するための演算専用回路（演算回路107）が設けられているので、乗算処理及び除算処理の効率化を図ることができる。

【0688】

また、ソースプログラム上において、例えば、ソースコード「ADDQ r, (k)」が実行されると、Qレジスタの格納データ（上位側アドレス値）及び1バイトの整数k（直値：下位側アドレス値）で指定されたアドレスのメモリの内容（格納データ）に、レジスタr（A、B、C、D、E、H又はLレジスタ）の格納データが加算され、該加算結果がレジスタrに格納される。それゆえ、図93B中のソースコード「ADDQ A, (.LOW.wWAVENUM)」が実行されると、Qレジスタの格納データ及び1バイトの整数値「.LOW.wWAVENUM」で指定されたアドレスのメモリの内容（設定値データ）にAレジスタの内容（格納データ）が加算され、該加算結果がAレジスタに格納される。

【0689】

すなわち、図93Bに示す例では、S309の設定値の加算処理において、抽籤テーブル選択用相対値に係数「6」を乗算して、その乗算値に設定値データを加算することにより、抽籤対象役の抽籤値が格納された抽籤テーブルのアドレスを算出している。

【0690】

上述のように、本実施形態では、内部抽籤処理において、Qレジスタ（拡張レジスタ）を用いたメインCPU101専用命令コード（「ADDQ」命令）が用いられており、この命令コードを用いれば、直値により、メインROM102、メインRAM103やメモリーマップI/Oにアクセスすることができる。それゆえ、内部抽籤処理のソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【0691】

〔図柄設定処理〕

次に、図97～図100を参照して、メインフロー（図82参照）中のS205で行う図柄設定処理について説明する。

【0692】

図97は、図柄設定処理の手順を示すフローチャートである。図98は、特賞（ボーナス）当籤番号及び小役当籤番号と、内部当籤役との対応表である。なお、図98では、「はずれ（00）」に対応する特賞当籤番号及び小役当籤番号の図示は省略している。また、図99は、図柄設定処理中のS324～S330の処理を実行するためのソースプログラムの一例を示す図であり、図100は、図柄設定処理のソースプログラム上で、実際に参照される当り要求フラグテーブル（フラグデータテーブル、当籤フラグテーブルデータ）の構成の一例を示す図である。

【0693】

まず、メインCPU101は、内部抽籤処理で取得された当り要求フラグステータスに基づいて、特賞当籤番号及び小役当籤番号を抽出し、該抽出された特賞当籤番号及び小役当籤番号をメインRAM103内の当籤番号格納領域（不図示）に保存する（S321）。

【0694】

本実施形態では、図98に示すように、特賞（ボーナス）当籤番号「1」及び「2」には、それぞれ、内部当籤役「F_BB1」及び「F_BB2」が対応付けられている。また

10

20

30

40

50

、小役当籤番号「1」～「36」には、それぞれ、内部当籤役「F__チリリブ」～「F__RB役4」が対応付けられている。そして、図94で説明したように、当り要求フラグステータスの値は、特賞当籤番号に特賞番号（本実施形態では「37（16進数では25H）」）を乗算した値に、小役当籤番号を加算した値である。それゆえ、S321の処理において、当り要求フラグステータスの値から特賞当籤番号及び小役当籤番号を抽出するため、本実施形態では、メインCPU101は、当り要求フラグステータスの値を特賞番号（「37」）で除算する。その結果、除算処理により生成された、商の値が特賞当籤番号（10進数で0～2のいずれか）となり、余りの値が小役当籤番号（10進数で0～36のいずれか）となる。

【0695】

次いで、メインCPU101は、抽出された小役当籤番号に基づいて、小役が当籤したか否かを判別する（S322）。この処理において、小役当籤番号が1～36のいずれかである場合には、メインCPU101は、小役が当籤したと判定し、小役当籤番号が0である場合には、メインCPU101は、小役が当籤しなかったと判定する。

【0696】

S322において、メインCPU101が、小役が当籤していないと判別したとき（S322がNO判定の場合）、メインCPU101は、後述のS331の処理を行う。一方、S322において、メインCPU101が、小役が当籤したと判別したとき（S322がYES判定の場合）、メインCPU101は、小役当籤番号を減算結果の初期値としてセットする（S323）。

【0697】

次いで、メインCPU101は、当り要求フラグテーブル（図100参照）をセットする（S324）。次いで、メインCPU101は、減算結果を1減算し、該減算結果を更新する（S325）。次いで、メインCPU101は、減算結果が「0」未満であるか否かを判別する（S326）。

【0698】

S326において、メインCPU101が、減算結果が「0」未満でないと判別したとき（S326がNO判定の場合）、メインCPU101は、ビット数算出処理を行う（S327）。なお、S327のビット数算出処理では、当り要求フラグテーブルに規定されている、小役当籤番号に対応する当り要求フラグデータの格納領域のブロック数を取得する。

【0699】

なお、本実施形態では、当り要求フラグ格納領域（内部当籤役格納領域）において、当り要求格納領域0～7のブロックと、当り要求格納領域8～11のブロックとが設けられている。それゆえ、S327のビット数算出処理で取得される当り要求フラグデータの格納領域のブロック数の最大値は「2」となる。例えば、内部当籤役が「F__確チリリブ」である場合には、当り要求フラグテーブル（図100参照）に示すように、当り要求格納領域0～7のブロックに含まれる格納領域7と、当り要求格納領域8～11のブロックに含まれる格納領域9にそれぞれ当り要求フラグデータが規定されているので、S327のビット数算出処理で取得される当り要求フラグデータの格納領域のブロック数は「2」となる。

【0700】

次いで、メインCPU101は、ビット数算出処理を行う（S328）。なお、S328のビット数算出処理では、当り要求フラグテーブル（図100参照）において規定されるブロック単位の当り要求フラグデータのバイト数を算出する。例えば、内部当籤役が「F__確チリリブ」である場合には、当り要求フラグテーブル（図100参照）に示すように格納領域7及び格納領域9ではともに1バイトの当り要求フラグデータが格納されるので、S328のビット数算出処理で取得されるブロック単位の当り要求フラグデータのバイト数は1バイトとなる。なお、図100に記載のテーブルにおいて、格納領域7に格納される当り要求フラグデータには「10000000B | 01000000B」と記載されているが、これは、格納領域7に格納される当り要求フラグデータが「10000000

10

20

30

40

50

0 B」又は（「|」は論理和の記号）「0 1 0 0 0 0 0 B」であることを意味する。

【0701】

なお、上述したS325～S328の処理は、小役当籤番号の回数だけ繰り返される。例えば、内部当籤役が「F__確チリリプ」（小役当籤番号が「2」）である場合には、上述したS325～S328の処理は、2回繰り返される。また、S325～S328の処理が複数回繰り返される場合には、S327及びS328のビット数算出処理でそれぞれ取得されるブロック数及びブロック単位の当り要求フラグデータのバイト数は、別の格納領域に保存される。また、上述したS325～S328の処理により得られたブロック数及びブロック単位の当り要求フラグデータのバイト数は、当り要求フラグデータの格納先を指定する情報（オンビット情報）となる。

10

【0702】

ここで再度、S326の処理に戻って、S326において、メインCPU101が、減算結果が「0」未満であると判別したとき（S326がYES判定の場合）、メインCPU101は、当り要求フラグ格納領域（内部当籤役格納領域）のセット処理を行う（S329）。この際、メインCPU101は、上述したS325～S328の処理により得られたブロック数及びブロック単位の当り要求フラグデータのバイト数（オンビット情報）に基づいて、チェック（更新）対象となる当り要求フラグ格納領域のみをセットする。具体的には、チェック（更新）対象となる当り要求フラグ格納領域のアドレスをDEレジスタに格納する（図99参照）。

【0703】

次いで、メインCPU101は、圧縮データ格納処理を行う（S330）。この処理では、メインCPU101は、主に、当り要求フラグデータをチェック（更新）対象となる当り要求フラグ格納領域内の所定の格納領域に転送（展開）する処理を行う。圧縮データ格納処理の詳細については、後述の図101を参照しながら後で説明する。

20

【0704】

S330の処理後又はS322がNO判定の場合、メインCPU101は、持越役格納領域（図31参照）を参照して、持越役があるか否かを判別する（S331）。S331において、メインCPU101が、持越役があると判別したとき（S331がYES判定の場合）、メインCPU101は、後述のS334の処理を行う。

【0705】

一方、S331において、メインCPU101が、持越役がないと判別したとき（S331がNO判定の場合）、メインCPU101は、S321の処理で抽出された特賞当籤番号に基づいて、ボーナス役（BB1又はBB2）が当籤したか否かを判別する（S332）。

30

【0706】

S332において、メインCPU101が、ボーナス役が当籤していないと判別したとき（S332がNO判定の場合）、メインCPU101は、図柄判定処理を終了し、処理をメインフロー（図82参照）のS206に移す。

【0707】

一方、S332において、メインCPU101が、ボーナス役が当籤したと判別したとき（S332がYES判定の場合）、メインCPU101は、当籤した特賞当籤番号を持越役格納領域に格納する（S333）。

40

【0708】

S333の処理後又はS331がNO判定の場合、メインCPU101は、特賞当籤番号を当籤番号格納領域（不図示）にセットし、当り要求フラグ格納領域に当り要求フラグデータをセットし、RT状態をRT5状態にセットし、RT遊技数（RT1状態の消化ゲーム数）をクリア（「0」）する（S334）。そして、S334の処理後、メインCPU101は、図柄設定処理を終了し、処理をメインフロー（図82参照）のS206に移す。

【0709】

本実施形態では、上述のようにして図柄設定処理が行われる。上述した図柄設定処理中の

50

S 3 2 4～S 3 3 0の処理（入賞に係るデータの圧縮・展開処理）は、メインCPU 1 0 1が、図9 9のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中で、S 3 3 0の圧縮データ格納処理は、メインCPU 1 0 1が図9 9中のソースコード「CALLF SB__BTEP__0 0」を実行することにより行われる。

【0 7 1 0】

「CALLF」命令は、上述のようにメインCPU 1 0 1専用の2バイト命令コードであり、図9 9中のソースコード「CALLF SB__BTEP__0 0」が実行されると、「SB__BTEP__0 0」で指定されているアドレスに、処理をジャンプさせ、圧縮データ格納処理が開始される。なお、S 3 3 0の圧縮データ格納処理では、上述のように、当り要求フラグテーブルに格納された当り要求フラグデータ（圧縮データ）が、対応する当り要求フラグ格納領域に展開（コピー）される。

【0 7 1 1】

また、上述した図柄設定処理中のS 3 2 9の当り要求フラグ格納領域のアドレスのセット処理は、メインCPU 1 0 1が図9 9中のソースコード「LDQ DE, .LOW.wWAVEBIT」を実行することにより行われる。すなわち、図柄設定処理中のS 3 2 9の処理は、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行うメインCPU 1 0 1専用の「LDQ」命令により行われる。この場合、図柄設定処理のソースプログラム上において、アドレス設定に係る命令コードを省略することができ、その分、図柄設定処理のソースプログラムの容量（メインROM 1 0 2の使用容量）を低減することができる。この結果、本実施形態では、メインROM 1 0 2において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【0 7 1 2】

さらに、本実施形態では、上述した図柄設定処理中のS 3 2 4～S 3 3 0で説明した処理手順で入賞に係るデータの圧縮・展開処理を行い、かつ、その処理の中で上述したメインCPU 1 0 1専用命令コードを用いることにより、入賞に係るデータの圧縮・展開処理の効率化を図ることができるとともに、限られたメインRAM 1 0 3の容量を有効活用することができる。

【0 7 1 3】

〔圧縮データ格納処理〕

次に、図1 0 1を参照して、例えば、図柄判定処理（図9 7参照）中のS 3 3 0で行う圧縮データ処理について説明する。図1 0 1は、圧縮データ格納処理の手順を示すフローチャートである。

【0 7 1 4】

なお、図1 0 1に示す圧縮データ格納処理は、図柄判定処理（図9 7参照）中のS 3 3 0だけでなく、後述の図柄コード取得処理（後述の図1 2 8参照）中のS 6 4 9においても実行される。図柄判定処理（図9 7参照）中のS 3 3 0で実行される圧縮データ格納処理では、処理対象となるフラグデータは当り要求フラグデータ（当籤役に係るフラグデータ）となるが、後述の図柄コード取得処理（後述の図1 2 8参照）中のS 6 4 9で実行される圧縮データ格納処理では、処理対象となるフラグデータは入賞作動フラグデータ（入賞役に係るフラグデータ）である。そして、処理対象となるフラグデータの種別が異なること以外は、両者の処理は同じ処理になる。

【0 7 1 5】

それゆえ、図1 0 1のフローチャートでは、処理対象とするフラグデータを「処理対象フラグデータ」と記し、処理対象となるフラグテーブルを「処理対象フラグテーブル」と記す。また、この記載に合わせて、以下の圧縮データ格納処理の説明においても、当り要求フラグデータ又は入賞作動フラグデータを「処理対象フラグデータ」と称し、当り要求フラグテーブル（図1 0 0参照）又は後述の図柄対応入賞作動テーブル（例えば、後述の図1 3 0 A等参照）を「処理対象フラグテーブル」と称す。

【0 7 1 6】

まず、メインCPU101は、格納先チェックビットをセットする（S341）。この処理では、格納先チェックビットはAレジスタ以外のレジスタに格納される。

【0717】

格納先チェックビットは、処理対象フラグデータの格納先（転送先）となるブロックを指定するための1バイトのデータである。本実施形態では、当り要求フラグ格納領域及び入賞作動フラグ格納領域はともに、2つのブロック（格納領域0～7のブロック及び格納領域8～11のブロック）で構成される。そして、例えば、内部当籤役「F__確チリリプ」が決定された場合には、図100の当り要求フラグテーブルに示すように格納領域7及び格納領域9のそれぞれに当り要求フラグデータが格納されるので（格納先のブロック数が「2」になるので）、S341の処理では、格納先チェックビットとして、「00000011B」がセットされる。なお、この1バイトのデータのビット0の値（1／0）が格納領域0～7のブロック内の格納先の有無に対応し、ビット1の値（1／0）が格納領域8～11のブロック内の格納先の有無に対応する。

10

【0718】

次いで、メインCPU101は、バイト単位の転送カウンタの値を「8」にセットする（S342）。本実施形態では、各ブロックのバイト数が「8」であるので、転送カウンタの初期値には「8」がセットされる。

【0719】

次いで、格納先チェックビットから転送指示ビットの値を抽出する（S343）。なお、転送指示ビットは、格納先チェックビット内のビット0のデータに対応し、S343の処理では、1バイトのレジスタに格納されている格納先チェックビットを1回（1ビット分）右シフトすることにより、転送指示ビットが抽出される。具体的には、格納先チェックビットが格納された1バイトのレジスタ（Aレジスタ以外のレジスタ）を1回右シフトすると、ビット7～ビット1に格納されているデータがそれぞれビット6～ビット0に移動するとともに、シフト前のビット0のデータが出力される。そして、このシフト処理により出力されたデータが転送指示ビットの値となる。

20

【0720】

次いで、メインCPU101は、抽出された転送指示ビットの値に基づいて、転送指示があるか否かを判別する（S344）。この処理では、メインCPU101は、抽出された転送指示ビットの値が「1」である場合に転送指示があると判定する。例えば、格納先チェックビットとして、「00000011B」がセットされた場合、1回目（格納領域の1ブロック目に対応）及び2回目（格納領域の2ブロック目に対応）のS344の判定処理では、転送指示ありの判定となるが、3回目以降のS344の判定処理では、転送指示なしの判定となる。

30

【0721】

S344において、メインCPU101が、転送指示がないと判別したとき（S344がNO判定の場合）、メインCPU101は、後述のS354の処理を行う。

【0722】

一方、S344において、メインCPU101が、転送指示があると判別したとき（S344がYES判定の場合）、メインCPU101は、処理対象フラグテーブルからバイト単位格納先指定情報を取得する（S345）。この処理では、バイト単位格納先指定情報として、処理対象フラグテーブル内の処理対象役（当籤役又は入賞役）のフラグデータが格納された領域の先頭アドレスに格納されている、転送先を示す1バイトのデータが取得される。例えば、内部当籤役が「F__確チリリプ」である場合には、図100に示す当り要求フラグテーブル内の「F__確チリリプ」のフラグデータが格納された領域の先頭アドレスに格納されている、格納領域7を転送先として指定する1バイトデータ「10000000B」がバイト単位格納先指定情報として取得される。

40

【0723】

次いで、メインCPU101は、処理対象フラグテーブル内で参照するアドレスの更新処理（アドレスを1加算する処理）を行う（S346）。また、この処理では、メインCP

50

U101は、処理対象フラグデータの格納（転送）先となるブロックの先頭格納領域を指定するアドレスを初期アドレスとしてセットする。例えば、1ブロック目の処理では、S346の処理において、初期アドレスとして格納領域0のアドレスがセットされ、2ブロック目の処理では、S346の処理において、初期アドレスとして格納領域8のアドレスがセットされる。

【0724】

次いで、メインCPU101は、バイト単位格納先指定情報から転送指示ビットの値を抽出する（S347）。なお、ここでいう転送指示ビットは、バイト単位格納先指定情報のビット0に対応し、S347の処理では、1バイトのレジスタに格納されているバイト単位格納先指定情報を1回右シフトすることにより、転送指示ビットの値を抽出する（ビット0のデータを出力する）。

10

【0725】

次いで、メインCPU101は、S347の処理で抽出された転送指示ビットの値に基づいて、転送指示があるか否かを判別する（S348）。この処理では、メインCPU101は、抽出された転送指示ビットの値が「1」である場合、転送指示があると判定する。例えば、バイト単位格納先指定情報として、「00000010B」がセットされた場合、2回目（1ブロック目の格納領域1又は2ブロック目の格納領域9）のS347の処理でビット1のデータ「1」が転送指示ビットの値として出力され転送指示ありの判定となるが、1回目及び3～8回目のS347の処理では、転送指示なしの判定となる。

【0726】

20

S348において、メインCPU101が、転送指示がないと判別したとき（S348がNO判定の場合）、メインCPU101は、後述のS351の処理を行う。

【0727】

一方、S348において、メインCPU101が、転送指示があると判別したとき（S348がYES判定の場合）、メインCPU101は、現在セットされている処理対象フラグテーブル内のアドレスに格納されている処理対象フラグデータ（当り要求フラグデータ又は入賞作動フラグデータ）を、指定された格納領域に転送（コピー）する（S349）。

【0728】

例えば、内部当籤役が「F__確チリリブ」であり、現在の処理が1ブロック目の格納領域（格納領域0～7）に対して行われている場合には、バイト単位格納先指定情報が「10000000B」（格納領域7を格納先として指定するデータ）となるので、8回目のS347の処理で転送指示があると判定され、その後のS349の処理で、当り要求フラグデータ「10000000B」、「01000000B」、「00100000B」及び「00010000B」のいずれかが、当り要求フラグ格納領域の格納領域7に転送（コピー）される。

30

【0729】

次いで、メインCPU101は、処理対象フラグテーブル内で参照するアドレスの更新処理（アドレスを1加算する処理）を行う（S350）。

【0730】

S350の処理後又はS348がNO判定の場合、メインCPU101は、処理対象フラグデータの格納先となる格納領域を指定するアドレスの更新処理（アドレスを1加算する処理）を行う（S351）。次いで、メインCPU101は、転送カウンタの値を1減算する（S352）。

40

【0731】

次いで、メインCPU101は、転送カウンタの値が「0」であるか否かを判別する（S353）。S353において、メインCPU101が、転送カウンタの値が「0」でないと判別したとき（S353がNO判定の場合）、メインCPU101は、処理をS347の処理に戻し、S347以降の処理を繰り返す。

【0732】

一方、S353において、メインCPU101が、転送カウンタの値が「0」とであると判

50

別したとき（S353がYES判定の場合）、メインCPU101は、現在の格納先チェックビットに転送指示対象が残っているか否かを判別する（S354）。この処理では、メインCPU101は、現処理時点において、格納先チェックビット内に「1」が格納されているビットが残っているか否かを判別する。そして、メインCPU101は、格納先チェックビット内に「1」が格納されているビットが残っている場合、すなわち、処理対象となるブロックが存在する場合には、現在の格納先チェックビットに転送指示対象が残っていると判定する。

【0733】

S354において、メインCPU101が、現在の格納先チェックビットに転送指示対象が残っていると判別したとき（S354がYES判定の場合）、メインCPU101は、処理をS342の処理に戻し、S342以降の処理を繰り返す。一方、S354において、メインCPU101が、現在の格納先チェックビットに転送指示対象が残っていないと判別したとき（S354がNO判定の場合）、メインCPU101は、圧縮データ格納処理を終了し、処理を例えば図柄判定処理（図97参照）中のS331に移す。

【0734】

〔第2インターフェースボード制御処理（規定外）〕

次に、図102を参照して、メインフロー（図82参照）中のS207で行う第2インターフェースボード制御処理について説明する。図102は、第2インターフェースボード制御処理の手順を示すフローチャートである。なお、この処理は、メインRAM103内の規定外作業領域（図12C参照）で行われる。また、この第2インターフェースボード制御処理で用いられるプログラムはメインROM102内の規定外エリアに格納されている（図12B参照）。

【0735】

まず、メインCPU101は、スタックポインタ（SP）にセットされているメインRAM103内のスタックエリアのアドレスデータを退避させる（S361）。次いで、メインCPU101は、メインRAM103内の規定外スタックエリアのアドレスデータをスタックポインタ（SP）にセットする（S362）。

【0736】

次いで、メインCPU101は、ナビデータを取得する（S363）。次いで、メインCPU101は、ナビ変換テーブルをメインRAM103内の規定外作業領域にセットする（S364）。

【0737】

次いで、メインCPU101は、ナビ変換テーブルを参照して第2インターフェース用押し順番号を取得する（S365）。次いで、メインCPU101は、取得した第2インターフェース用押し順番号を、規定外作業領域に設けられた規定外押し順番号格納領域（不図示）に格納する（S366）。次いで、メインCPU101は、規定外作業領域に設けられた押下位置テーブル選択カウンタの値に「0」をセットする（S367）。

【0738】

次いで、メインCPU101は、取得したナビデータが押し順ナビ（押し順小役用のナビデータ）であるか否かを判別する（S368）。S368において、メインCPU101が、取得したナビデータが押し順ナビであると判別したとき（S368がYES判定の場合）、メインCPU101は、後述のS372の処理を行う。

【0739】

一方、S368において、メインCPU101が、取得したナビデータが押し順ナビでないと判別したとき（S368がNO判定の場合）、メインCPU101は、取得したナビデータがBB1停止操作用のナビデータ（10）であるか否かを判別する（S369）。

【0740】

S369において、メインCPU101が、取得したナビデータがBB1停止操作用のナビデータであると判別したとき（S369がYES判定の場合）、メインCPU101は、後述のS371の処理を行う。一方、S369において、メインCPU101が、取得

10

20

30

40

50

したナビデータがBB1停止作用のナビデータでないと判別したとき（S369がNO判定の場合）、メインCPU101は、押下位置テーブル選択カウンタの値に「1」を加算する（S370）。S370において、押下位置テーブル選択カウンタの値に「1」を加算する処理は、押下位置テーブル（不図示）からBB2の押下位置を取得するために行われる処理である。

【0741】

S370の処理後又はS369がYES判定の場合、メインCPU101は、押下位置テーブル選択カウンタの値に「1」を加算する（S371）。S371において、押下位置テーブル選択カウンタの値に「1」を加算する処理は、押下位置テーブル（不図示）からBB1又はBB2の押下位置を取得するために行われる処理である。

10

【0742】

S371の処理後又はS368がYES判定の場合、メインCPU101は、押下位置テーブル選択カウンタの値に基づいて、押下位置テーブル（不図示）を選択する（S372）。次いで、メインCPU101は、選択した押下位置テーブルを参照して、3リール分（左リール3L、中リール3C及び右リール3R）の押下位置データを取得する（S373）。次いで、メインCPU101は、取得した押下位置データを規定外作業領域に設けられた規定外押下位置格納領域（不図示）に格納する（S374）。S367～S371の処理により、ナビデータが押し順ナビであれば、押下位置テーブル選択カウンタの値は「0」となり、ナビデータがBB1停止作用のナビデータであれば、押下位置テーブル選択カウンタの値は「1」となり、ナビデータがBB2停止作用のナビデータであれば、押下位置テーブル選択カウンタの値は「2」となる。すなわち、ナビデータに基づいて、押下位置データが取得される。

20

【0743】

次いで、メインCPU101は、第2インターフェースボード出力処理を行う（S375）。なお、第2インターフェースボード出力処理の詳細については、後述の図103を参照しながら後で説明する。

【0744】

次いで、メインCPU101は、全レジスタの復帰処理を行う（S376）。次いで、メインCPU101は、S361で退避させたスタックエリアのアドレスデータをスタックポインタ（SP）にセットする（S377）。そして、S377の処理後、メインCPU101は、第2インターフェースボード制御処理を終了し、処理をメインフロー（図82参照）のS208に移す。

30

【0745】

〔第2インターフェースボード出力処理〕

次に、図103を参照して、第2インターフェースボード制御処理（図102参照）中のS375で行う第2インターフェースボード出力処理について説明する。図103は、第2インターフェースボード出力処理の手順を示すフローチャートである。なお、この第2インターフェースボード出力処理は、メインRAM103の規定外作業領域で行われる。

【0746】

まず、メインCPU101は、第2インターフェース用シリアル回線（第2シリアル通信回路115：SCU2）を介して送信動作が行われているか否かを判別する（S381）。S381において、メインCPU101が、第2インターフェース用シリアル回線を介して送信動作が行われていると判別したとき（S381がYES判定の場合）、メインCPU101は、第2インターフェースボード出力処理を終了し、処理を第2インターフェースボード制御処理（図102参照）のS376に移す。

40

【0747】

一方、S381において、メインCPU101が、第2インターフェース用シリアル回線を介して送信動作が行われていないと判別したとき（S381がNO判定の場合）、メインCPU101は、規定外作業領域に設けられたループカウンタの値に「3」（リールの個数）をセットし、シリアル通信用サム値に初期値「1」をセットする（S382）。次

50

いで、メインCPU101は、第2インターフェース用シリアル回線（第2シリアル通信回路115：SCU2）を介して、送信開始データを送信する（S383）。

【0748】

次いで、メインCPU101は、所定のリール（回胴）の規定外押下位置格納領域を参照し、所定のリールの押下位置データを取得する（S384）。次いで、メインCPU101は、参照する規定外押下位置格納領域を次の対象リール（回胴）のそれに更新する（S385）。

【0749】

次いで、メインCPU101は、パルス変換データ（不図示）及び取得した押下位置データに基づいて、押下位置データ（図柄位置）に対応するパルス数データを取得する（S386）。なお、押下位置データ（図柄位置）とパルス数データとの対応関係の詳細については省略するが、例えば、取得した押下位置データ（図柄位置）が「3」（左リール3Lでは図柄「白7」）である場合には、パルス数データとして「38」が取得され、押下位置データ（図柄位置）が「10」（左リール3Lでは図柄「リプレイ」）である場合には、パルス数データとして「155」が取得される。また、例えば、取得した押下位置データ（図柄位置）が「12」（左リール3Lでは図柄「青7」）である場合には、パルス数データとして「189」が取得され、押下位置データ（図柄位置）が「15」（左リール3Lでは図柄「リプレイ」）である場合には、パルス数データとして「239」が取得される。

【0750】

次いで、メインCPU101は、取得したパルス数データを、第2インターフェース用シリアル回線（第2シリアル通信回路115：SCU2）を介して送信する（S387）。次いで、メインCPU101は、シリアル通信用サム値にパルス数データを加算する（S388）。次いで、メインCPU101は、ループカウンタの値を1減算する（S389）。

【0751】

次いで、メインCPU101は、ループカウンタの値が「0」であるか否かを判別する（S390）。S390において、メインCPU101が、ループカウンタの値が「0」でないと判別したとき（S390がNO判定の場合）、メインCPU101は、対象リールを次のリールに変更するとともに、処理をS384に戻し、S384以降の処理を繰り返す。

【0752】

一方、S390において、メインCPU101が、ループカウンタの値が「0」であると判別したとき（S390がYES判定の場合）、メインCPU101は、シリアル通信用サム値を、第2インターフェース用シリアル回線（第2シリアル通信回路115：SCU2）を介して送信する（S391）。そして、S391の処理後、メインCPU101は、第2インターフェースボード出力処理を終了し、処理を第2インターフェースボード制御処理（図102参照）のS376に移す。

【0753】

〔状態別制御処理〕

次に、図104を参照して、メインフロー（図82参照）中のS208で行う状態別制御処理について説明する。図104は、状態別制御処理の手順を示すフローチャートである。

【0754】

まず、メインCPU101は、サブフラグ変換処理を行う（S401）。この処理では、メインCPU101は、内部当籤役をサブフラグ（図36及び図37参照）に変換する処理を行う。なお、サブフラグ変換処理の詳細については、後述の図105を参照しながら後で説明する。

【0755】

次いで、メインCPU101は、ナビセット処理を行う（S402）。この処理では、メインCPU101は、RT状態、遊技状態及び小役当籤番号に基づいてナビデータを取得

10

20

30

40

50

する。なお、ナビセット処理の詳細については、後述の図108を参照しながら後で説明する。

【0756】

次いで、メインCPU101は、現在のRT状態がRT4状態であるか否かを判別する（S403）。S403において、メインCPU101が、現在のRT状態がRT4状態でないと判別したとき（S403がNO判定の場合）、メインCPU101は、後述のS406の処理を行う。

【0757】

一方、S403において、メインCPU101が、現在のRT状態がRT4状態であると判別したとき（S403がYES判定の場合）、メインCPU101は、フラグ変換処理を行う（S404）。この処理では、メインCPU101は、サブフラグをサブフラグEX（図36参照）に変換するためのフラグ変換抽籤処理（サブフラグデータの圧縮処理）を行う。このフラグ変換処理により、19種類（ハズレも含む）のサブフラグが、9種類（ハズレも含む）のサブフラグEXに変換（圧縮）される。なお、フラグ変換処理の詳細については、後述の図111を参照しながら後で説明する。

10

【0758】

次いで、メインCPU101は、サブフラグ圧縮処理を行う（S405）。この処理では、メインCPU101は、サブフラグEXをサブフラグD（図36参照）に変換し、サブフラグデータのさらなる圧縮処理を行う。このサブフラグ圧縮処理により、9種類（ハズレも含む）のサブフラグEXが、7種類（ハズレも含む）のサブフラグDに変換（圧縮）される。

20

【0759】

S405の処理後又はS403がNO判定の場合、メインCPU101は、現在の遊技状態が通常遊技状態であるか否かを判別する（S406）。

【0760】

S406において、メインCPU101が、現在の遊技状態が通常遊技状態であると判別したとき（S406がYES判定の場合）、メインCPU101は、通常中スタート時処理を行う（S407）。なお、通常中スタート時処理の詳細については、後述の図112を参照しながら後で説明する。そして、S407の処理後、メインCPU101は、状態別制御処理を終了し、処理をメインフロー（図82参照）のS209に移す。

30

【0761】

一方、S406において、メインCPU101が、現在の遊技状態が通常遊技状態でないと判別したとき（S406がNO判定の場合）、メインCPU101は、現在の遊技状態がCZであるか否かを判別する（S408）。

【0762】

S408において、メインCPU101が、現在の遊技状態がCZであると判別したとき（S408がYES判定の場合）、メインCPU101は、CZ中スタート時処理を行う（S409）。なお、CZ中スタート時処理の詳細については、後述の図113を参照しながら後で説明する。そして、S409の処理後、メインCPU101は、状態別制御処理を終了し、処理をメインフロー（図82参照）のS209に移す。

40

【0763】

一方、S408において、メインCPU101が、現在の遊技状態がCZでないと判別したとき（S408がNO判定の場合）、メインCPU101は、現在の遊技状態が通常ARTであるか否かを判別する（S410）。

【0764】

S410において、メインCPU101が、現在の遊技状態が通常ARTであると判別したとき（S410がYES判定の場合）、メインCPU101は、通常ART中スタート時処理を行う（S411）。なお、通常ART中スタート時処理の詳細については、後述の図117を参照しながら後で説明する。そして、S411の処理後、メインCPU101は、状態別制御処理を終了し、処理をメインフロー（図82参照）のS209に移す。

50

【0765】

一方、S410において、メインCPU101が、現在の遊技状態が通常ARTでないと判別したとき（S410がNO判定の場合）、メインCPU101は、現在の遊技状態がCTであるか否かを判別する（S412）。

【0766】

S412において、メインCPU101が、現在の遊技状態がCTであると判別したとき（S412がYES判定の場合）、メインCPU101は、CT中スタート時処理を行う（S413）。なお、CT中スタート時処理の詳細については、後述の図118を参照しながら後で説明する。そして、S413の処理後、メインCPU101は、状態別制御処理を終了し、処理をメインフロー（図82参照）のS209に移す。

10

【0767】

一方、S412において、メインCPU101が、現在の遊技状態がCTでないと判別したとき（S412がNO判定の場合）、メインCPU101は、現在の遊技状態がボーナス状態であるか否かを判別する（S414）。

【0768】

S414において、メインCPU101が、現在の遊技状態がボーナス状態であると判別したとき（S414がYES判定の場合）、メインCPU101は、BB中スタート時処理を行う（S415）。なお、BB中スタート時処理の詳細については、後述の図125を参照しながら後で説明する。そして、S415の処理後、メインCPU101は、状態別制御処理を終了し、処理をメインフロー（図82参照）のS209に移す。

20

【0769】

一方、S414において、メインCPU101が、現在の遊技状態がボーナス状態でないと判別したとき（S414がNO判定の場合）、メインCPU101は、その他処理を行う（S416）。この処理では、メインCPU101は、上記各種判定処理で対象となった遊技状態以外の遊技状態に応じた処理を行う。例えば、現在の遊技状態がART準備状態である場合には、ART準備状態に対応した処理を行う。そして、S416の処理後、メインCPU101は、状態別制御処理を終了し、処理をメインフロー（図82参照）のS209に移す。

【0770】

[サブフラグ変換処理]

次に、図105～図107を参照して、状態別制御処理（図104参照）中のS401で行うサブフラグ変換処理について説明する。図105は、サブフラグ変更処理の手順を示すフローチャートである。また、図106は、サブフラグ変更処理を実行するためのソースプログラムの一例を示す図であり、図107は、サブフラグ変換処理のソースプログラム上で、実際に参照されるサブフラグ変換テーブル（変換テーブル）の構成の一例を示す図である。

30

【0771】

まず、メインCPU101は、小役当籤番号（0～36）を取得する（S421）。次いで、メインCPU101は、現在、ボーナス作動中であるか否かを判別する（S422）。

【0772】

S422において、メインCPU101が、現在、ボーナス作動中であると判別したとき（S422がYES判定の場合）、メインCPU101は、小役当籤番号をボーナス作動中のサブフラグに変換して保存する（S423）。この小役当籤番号をボーナス作動中のサブフラグに変換する処理は、メインCPU101が図106中のソースコード「SUB（減算命令） cNHT__RBST-c7HT1__FLA」を実行することにより行われる。そして、本実施形態では、この「SUB」命令の実行により、一律、小役当籤番号をサブフラグ「サボテン（14）」に変換する。そして、S423の処理後、メインCPU101は、サブフラグ変換処理を終了し、処理を状態別制御処理（図104参照）のS402に移す。

40

【0773】

50

一方、S 4 2 2において、メインCPU 1 0 1が、現在、ボーナス作動中でないと判別したとき（S 4 2 2がNO判定の場合）、メインCPU 1 0 1は、図1 0 7に示すサブフラグ変換テーブルをセットする（S 4 2 4）。この処理では、判定対象とするサブフラグの初期値を「ハズレ（0 0）」にセットするとともに参照対象とする図1 0 7に示すサブフラグ変換テーブル内のブロックの初期アドレスとして、サブフラグ「ハズレ（0 0）」が格納されているアドレス（「d S B C V T B + 1」）をセットする。

【0 7 7 4】

次いで、メインCPU 1 0 1は、現在、参照対象となっているサブフラグ変換テーブル内のブロックに規定されている小役当籤番号のデータが、現ゲームで取得された小役当籤番号に対応するデータであるか否かを判別する（S 4 2 5）。

【0 7 7 5】

S 4 2 5において、メインCPU 1 0 1が、参照対象となっているサブフラグ変換テーブル内のブロックに規定されている小役当籤番号のデータが、現ゲームで取得された小役当籤番号に対応するデータでないと判別したとき（S 4 2 5がNO判定の場合）、メインCPU 1 0 1は、参照対象とするサブフラグ変換テーブル内のブロックを次のアドレスのブロックに更新する（S 4 2 6）。次いで、メインCPU 1 0 1は、サブフラグの値に「1」を加算する（S 4 2 7）。そして、S 4 2 7の処理後、メインCPU 1 0 1は、処理をS 4 2 5の処理に戻し、S 4 2 5以降の処理を繰り返す。

【0 7 7 6】

一方、S 4 2 5において、メインCPU 1 0 1が、参照対象となっているサブフラグ変換テーブル内のブロックに規定されている小役当籤番号のデータが、現ゲームで取得された小役当籤番号に対応するデータであると判別したとき（S 4 2 5がYES判定の場合）、メインCPU 1 0 1は、図1 0 7に示すサブフラグ変換テーブルを参照して、小役当籤番号に対応付けられたサブフラグ変換制御データ（小役当籤番号のアドレスの次のアドレスに格納された1バイトデータ）を取得し、該サブフラグ変換制御データをメインRAM 1 0 3に設けられたサブフラグ変換制御データ格納領域（不図示）に格納する（S 4 2 8）。この処理において、例えば、現ゲームで取得された小役当籤番号が「0 3」（内部当籤役「F __ 3連チリリプ」）である場合には、図1 0 7に示すサブフラグ変換テーブルを参照して、サブフラグ変換制御データ「0 0 0 0 0 0 1 1 B」が取得される。そして、S 4 2 8の処理後、メインCPU 1 0 1は、サブフラグ変換処理を終了し、処理を状態別制御処理（図1 0 4参照）のS 4 0 2に移す。

【0 7 7 7】

本実施形態では、上述のようにしてサブフラグ変換処理が行われる。なお、上述したサブフラグ変換処理は、メインCPU 1 0 1が、図1 0 6のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。また、サブフラグ変換処理のソースプログラム上で実際に参照される、図1 0 7に示すサブフラグ変換テーブルでは、各サブフラグに対してサブフラグ変換制御データ（制御ステータス）が対応付けられている。この際、同種のサブフラグに対しては、同じサブフラグ変換制御データ（制御ステータス）が対応付けられている。

【0 7 7 8】

例えば、サブフラグ「3連チリリプA」及び「3連チリリプB」に対しては、サブフラグ変換制御データ（制御ステータス）「0 0 0 0 0 0 1 1 B」が共通して割り付けられている。また、例えば、サブフラグ「リーチ目リプ1」～「リーチ目リプ4」に対しては、サブフラグ変換制御データ（制御ステータス）「0 0 0 0 0 0 0 1 B」が共通して割り付けられている。そして、上述した内部当籤役（サブフラグ）をサブフラグEXに変換する際のフラグ変換抽籤処理では、サブフラグ変換制御データ格納領域に格納されたサブフラグ変換制御データ（制御ステータス）に基づいて、抽籤が行われる。

【0 7 7 9】

メイン側で管理するフラグ（内部当籤役）をサブ側で管理可能なフラグに変換するためのサブフラグ変換テーブルにおいて、同種の内部当籤役（サブフラグ）に対して共通のサブ

10

20

30

40

50

フラグ変換制御データを設けることにより、該変換テーブルの汎用性が高くなり、機種変更に伴う変換プログラムの変更も軽微な変更で対応可能となるので、開発コストの増大を抑制することができる。

【0780】

〔ナビセット処理〕

次に、図108～図110を参照して、状態別制御処理（図104参照）中のS402で行うナビセット処理について説明する。図108は、ナビセット処理の手順を示すフローチャートである。また、図109は、ナビセット処理中の後述のS434～S436の処理を実行するためのソースプログラムの一例を示す図であり、図110は、ナビセット処理のソースプログラム上で、実際に参照されるナビデータテーブルの構成の一例を示す図である。

10

【0781】

まず、メインCPU101は、サブフラグ変換制御データ格納領域（不図示）にナビセットフラグがセットされているか否かを判別する（S431）。具体的には、メインCPU101は、サブフラグ変換制御データ格納領域を参照し、セットされているサブフラグ変換制御データが、押し順ナビを発生させる小役当籤番号（10～23）に対応するデータであるか否かを判別する。S431において、メインCPU101が、サブフラグ変換制御データ格納領域にナビセットフラグがセットされていないと判別したとき（S431がNO判定の場合）、メインCPU101は、ナビセット処理を終了し、処理を状態別制御処理（図104参照）のS403に移す。

20

【0782】

一方、S431において、メインCPU101が、サブフラグ変換制御データ格納領域にナビセットフラグがセットされていると判別したとき（S431がYES判定の場合）、メインCPU101は、RT状態がRT0又はRT1状態であるか否かを判別する（S432）。S432において、メインCPU101が、RT状態がRT0又はRT1状態でないと判別したとき（S432がNO判定の場合）、メインCPU101は、ナビセット処理を終了し、処理を状態別制御処理（図104参照）のS403に移す。

【0783】

一方、S432において、メインCPU101が、RT状態がRT0又はRT1状態であると判別したとき（S432がYES判定の場合）、メインCPU101は、遊技状態が一般遊技状態であるか否かを判別する（S433）。S433において、メインCPU101が、遊技状態が一般遊技状態であると判別したとき（S433がYES判定の場合）、メインCPU101は、ナビセット処理を終了し、処理を状態別制御処理（図104参照）のS403に移す。

30

【0784】

一方、S433において、メインCPU101が、遊技状態が一般遊技状態でないと判別したとき（S433がNO判定の場合）、メインCPU101は、小役当籤番号を取得する（S434）。次いで、メインCPU101は、図110に示すナビデータテーブルを参照し、小役当籤番号に基づいて、ナビデータ（1～9のいずれか）を取得する（S435）。

40

【0785】

次いで、メインCPU101は、取得したナビデータ（複数の表示列の変動表示の停止操作に関する情報）をメインRAM103内の図示しないナビデータ格納領域（停止操作指示情報格納領域）に格納する（S436）。そして、S436の処理後、メインCPU101は、ナビセット処理を終了し、処理を状態別制御処理（図104参照）のS403に移す。

【0786】

本実施形態では、上述のようにしてナビセット処理が行われる。なお、上述したナビセット処理中のS434～S436の処理は、メインCPU101が、図109のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。この一連の

50

処理では、図 109 に示すように、ソースプログラム上において、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行うメインCPU101専用の「LDQ」命令が用いられる。

【0787】

ソースプログラム上において、例えば、ソースコード「LDQ A, (k)」が実行されると、Qレジスタの格納データ（上位側アドレス値）と、1バイトの整数k（直値：下位側アドレス値）とで指定されたアドレスのメモリの内容（格納データ）がAレジスタにロードされる。それゆえ、例えば、図109中のソースコード「LDQ A, (WHITFRT)」が実行されると、Qレジスタの格納データと、整数値「WHITFRT」とで指定されるアドレスのメモリの内容がAレジスタにロードされる。

10

【0788】

また、ソースプログラム上において、例えば、ソースコード「LDQ (k), A」が実行されると、Aレジスタの格納データが、Qレジスタの格納データ（上位側アドレス値）と、1バイトの整数k（直値：下位側アドレス値）とで指定されたアドレスのメモリにロードされる。それゆえ、例えば、図109中のソースコード「LDQ (WNAV IPTN), A」の実行により、Aレジスタに格納されたデータ（ナビデータ）が、Qレジスタの格納データ（上位側アドレス値）と1バイトの整数値「WNAV IPTN」（下位側アドレス値）とで指定されたアドレスのナビデータ格納領域に格納される。

【0789】

上述のように、本実施形態では、ナビセット処理において、Qレジスタ（拡張レジスタ）を用いたメインCPU101専用命令コードが用いられ、直値により、メインROM102、メインRAM103やメモリーマップI/Oにアクセスすることができる。この場合、ナビセット処理のソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

20

【0790】

【フラグ変換処理】

次に、図111を参照して、状態別制御処理（図104参照）中のS404で行うフラグ変換処理について説明する。なお、図111は、フラグ変換処理の手順を示すフローチャートである。

30

【0791】

まず、メインCPU101は、CT開始時であるか否かを判別する（S441）。

【0792】

S441において、メインCPU101が、CT開始時でないと判別したとき（S441がNO判定の場合）、メインCPU101は、後述のS443の処理を行う。一方、S441において、メインCPU101が、CT開始時であると判別したとき（S441がYES判定の場合）、メインCPU101は、CT中のフラグ変換抽籤に用いるフラグ変換抽籤テーブルのテーブル番号を抽籤で決定し、セットする（S442）。

40

【0793】

S442の処理後又はS441がNO判定の場合、メインCPU101は、現在の状態に応じたフラグ変換抽籤テーブルをセットする（S443）。例えば、現在の状態が非ART中のRT4状態である場合には、非ART中フラグ変換抽籤テーブル（図62参照）がセットされ、現在の状態が通常ART中のRT4状態である場合には、ART中フラグ変換抽籤テーブル（図47A及び47B参照）がセットされ、現在の状態がCT中のRT4状態である場合には、CT中フラグ変換抽籤テーブル（図54参照）がセットされる。

【0794】

次いで、メインCPU101は、セットされたフラグ変換抽籤テーブルを参照し、内部当籤役に基づいてフラグ変換抽籤処理を行う（S444）。なお、実際、この処理では、メ

50

インCPU101は、内部当籤役に対応するサブフラグに基づいて、図107に示すサブフラグ変換テーブルから取得されるサブフラグ変換制御データを用いてフラグ変換抽籤処理を行う。

【0795】

次いで、メインCPU101は、S444のフラグ変換抽籤に当籤したか否かを判別する(S445)。

【0796】

S445において、メインCPU101が、フラグ変換抽籤に当籤したと判別したとき(S445がYES判定の場合)、メインCPU101は、サブフラグ変換処理を行う(S446)。この処理において、例えば、内部当籤役が「F__1確チリリブ」である場合、すなわち、サブフラグが「3連チリリブB(03)」である場合、フラグ変換抽籤処理に当籤すると、S446のサブフラグ変換処理により、サブフラグ「3連チリリブB(03)」が、サブフラグEX「確定役(06)」又はサブフラグEX「3連チリリブ(07)」に変換される(図36参照)。

10

【0797】

S446の処理後、メインCPU101は、現在の遊技状態が非ART状態であるか否かを判別する(S447)。S447において、メインCPU101が、現在の遊技状態が非ART状態でないと判別したとき(S447がNO判定の場合)、メインCPU101は、フラグ変換処理を終了し、処理を状態別制御処理(図104参照)のS405に移す。

【0798】

一方、S447において、メインCPU101が、現在の遊技状態が非ART状態であると判別したとき(S447がYES判定の場合)、メインCPU101は、ARTセット数に「1」を加算する(S448)。次いで、メインCPU101は、次遊技の遊技状態にART準備状態をセットする(S449)。そして、S449の処理後、メインCPU101は、フラグ変換処理を終了し、処理を状態別制御処理(図104参照)のS405に移す。

20

【0799】

ここで再度、S445の処理に戻って、S445において、メインCPU101が、フラグ変換抽籤に当籤しなかったと判別したとき(S445がNO判定の場合)、メインCPU101は、サブフラグ維持処理を行う(S450)。この処理において、例えば、内部当籤役が「F__1確チリリブ」である場合、すなわち、サブフラグが「3連チリリブB(03)」である場合、フラグ変換抽籤に非当籤であると、S450のサブフラグ維持処理により、サブフラグ「3連チリリブB(03)」が、サブフラグEX「リプレイ(01)」に変換(維持)される。そして、S450の処理後、メインCPU101は、フラグ変換処理を終了し、処理を状態別制御処理(図104参照)のS405に移す。

30

【0800】

〔通常中スタート時処理〕

次に、図112を参照して、状態別制御処理(図104参照)中のS407で行う通常中スタート時処理について説明する。なお、図112は、通常中スタート時処理の手順を示すフローチャートである。

40

【0801】

まず、メインCPU101は、CZ抽籤テーブル(図41A参照)を参照し、現在のCZの抽籤状態及び内部当籤役(サブフラグ)に基づいてCZ抽籤処理を行う(S461)。次いで、メインCPU101は、S461のCZ抽籤に当籤したか否かを判別する(S462)。

【0802】

S462において、メインCPU101が、CZ抽籤に当籤しなかったと判別したとき(S462がNO判定の場合)、メインCPU101は、後述のS465の処理を行う。

【0803】

一方、S462において、メインCPU101が、CZ抽籤に当籤したと判別したとき(

50

S 4 6 2 が Y E S 判定の場合)、メイン C P U 1 0 1 は、次遊技の遊技状態に当籤した種別の C Z をセットする (S 4 6 3)。次いで、メイン C P U 1 0 1 は、当籤した種別の C Z ゲーム数を C Z ゲーム数カウンタにセットする (S 4 6 4)。なお、C Z ゲーム数カウンタは、C Z の継続期間を計数するカウンタであり、メイン R A M 1 0 3 に設けられる。S 4 6 4 の処理において、例えば、C Z 1 が当籤している場合には、C Z ゲーム数カウンタ (前半部) に第 1 の所定ゲーム数 (例えば、「1 2」) がセットされ、C Z 2 が当籤している場合には、C Z ゲーム数カウンタ (前半部) に第 2 の所定ゲーム数 (例えば、「1 5」) がセットされ、C Z 3 が当籤している場合には、C Z ゲーム数カウンタに第 4 の所定ゲーム数 (例えば、「1 7」) がセットされる。

【0 8 0 4】

S 4 6 4 の処理後又は S 4 6 2 が N O 判定の場合、メイン C P U 1 0 1 は、通常中高確率抽籤テーブル (図 4 0 A 参照) を参照し、内部当籤役 (サブフラグ) に基づいて C Z の抽籤状態の移行抽籤を行う (S 4 6 5)。次いで、メイン C P U 1 0 1 は、移行抽籤の結果に基づいて、C Z の抽籤状態を更新する (S 4 6 6)。そして、S 4 6 6 の処理後、メイン C P U 1 0 1 は、通常中スタート時処理を終了するとともに、状態別制御処理 (図 1 0 4 参照) も終了する。

【0 8 0 5】

〔C Z 中スタート時処理〕

次に、図 1 1 3 を参照して、状態別制御処理 (図 1 0 4 参照) 中の S 4 0 9 で行う C Z 中スタート時処理について説明する。なお、図 1 1 3 は、C Z 中スタート時処理の手順を示すフローチャートである。

【0 8 0 6】

まず、メイン C P U 1 0 1 は、現在の遊技状態が C Z 1 であるか否かを判別する (S 4 7 1)。

【0 8 0 7】

S 4 7 1 において、メイン C P U 1 0 1 が、現在の遊技状態が C Z 1 であると判別したとき (S 4 7 1 が Y E S 判定の場合)、メイン C P U 1 0 1 は、C Z 1 (C Z 2) 中処理を行う (S 4 7 2)。なお、C Z 1 (C Z 2) 中処理の詳細については、後述の図 1 1 4 及び図 1 1 5 を参照しながら後で説明する。そして、S 4 7 2 の処理後、メイン C P U 1 0 1 は、C Z 中スタート時処理を終了するとともに、状態別制御処理 (図 1 0 4 参照) も終了する。

【0 8 0 8】

一方、S 4 7 1 において、メイン C P U 1 0 1 が、現在の遊技状態が C Z 1 でないと判別したとき (S 4 7 1 が N O 判定の場合)、メイン C P U 1 0 1 は、現在の遊技状態が C Z 2 であるか否かを判別する (S 4 7 3)。

【0 8 0 9】

S 4 7 3 において、メイン C P U 1 0 1 が、現在の遊技状態が C Z 2 であると判別したとき (S 4 7 3 が Y E S 判定の場合)、メイン C P U 1 0 1 は、C Z 1 (C Z 2) 中処理を行う (S 4 7 4)。C Z 1 (C Z 2) 中処理の詳細については、後述の図 1 1 4 及び図 1 1 5 を参照しながら後で説明する。そして、S 4 7 4 の処理後、メイン C P U 1 0 1 は、C Z 中スタート時処理を終了するとともに、状態別制御処理 (図 1 0 4 参照) も終了する。なお、本実施形態では、C Z 1 中処理と C Z 2 中処理との間では A R T 抽籤に当籤する期待度を示すランク (モード又はポイント) が異なるだけであり、基本的な処理内容は同じである。そこで、本実施形態では、C Z 1 中処理及び C Z 2 中処理を C Z 1 (C Z 2) 中処理として一つの処理で説明する。

【0 8 1 0】

一方、S 4 7 3 において、メイン C P U 1 0 1 が、現在の遊技状態が C Z 2 でないと判別したとき (S 4 7 3 が N O 判定の場合)、メイン C P U 1 0 1 は、C Z 3 中処理を行う (S 4 7 5)。なお、C Z 3 中処理の詳細については、後述の図 1 1 6 を参照しながら後で説明する。そして、S 4 7 5 の処理後、メイン C P U 1 0 1 は、C Z 中スタート時処理を

10

20

30

40

50

終了するとともに、状態別制御処理（図104参照）も終了する。

【0811】

〔CZ1（CZ2）中処理〕

次に、図114及び図115を参照して、CZ中スタート時処理（図113参照）中のS472又はS474で行うCZ1（CZ2）中処理について説明する。なお、図114及び図115は、CZ1（CZ2）中処理の手順を示すフローチャートである。

【0812】

まず、メインCPU101は、現遊技がCZ1（又はCZ2）の前半部の遊技であるか否かを判別する（S481）。S481において、メインCPU101が、現遊技がCZ1（又はCZ2）の前半部の遊技でないと判別したとき（S481がNO判定の場合）、メインCPU101は、後述のS490の処理を行う。

10

【0813】

一方、S481において、メインCPU101が、現遊技がCZ1の前半部の遊技であると判別したとき（S481がYES判定の場合）、メインCPU101は、CZ1中モードアップ抽籤テーブル（図42参照）を参照し、内部当籤役（サブフラグ）に基づいてモードアップ抽籤処理を行う（S482）。また、S481において、メインCPU101が、現遊技がCZ2の前半部の遊技であると判別したとき（S481がYES判定の場合）、メインCPU101は、CZ2中ポイント抽籤テーブル（図43参照）を参照し、内部当籤役（サブフラグ）に基づいてポイントアップ抽籤を行う（S482）。

20

【0814】

次いで、メインCPU101は、S482の抽籤結果に基づいて、ランク（モード又はポイント）を更新する（S483）。次いで、メインCPU101は、S482の抽籤においてフリーズに当籤したか否かを判別する（S484）。

【0815】

S484において、メインCPU101が、フリーズに当籤したと判別したとき（S484がYES判定の場合）、メインCPU101は、遊技の進行を一時的に停止するフリーズ処理を行うとともに、ARTセット数及びCTセット数に「1」を加算する（S485）。また、この処理では、メインCPU101は、ARTレベル決定テーブル（図48A参照）を参照してARTレベルを決定し、セットする。なお、フリーズ発生時には、ARTレベルとして「ARTレベル2」が決定される。

30

【0816】

次いで、メインCPU101は、次遊技の遊技状態にART準備状態をセットする（S486）。そして、S486の処理後、メインCPU101は、CZ1（CZ2）中処理を終了するとともに、CZ中スタート時処理（図113参照）も終了する。

【0817】

ここで再度、S484の処理に戻って、S484において、メインCPU101が、フリーズに当籤しなかったと判別したとき（S484がNO判定の場合）、メインCPU101は、CZゲーム数カウンタ（前半部）の値を1減算する（S487）。次いで、メインCPU101は、CZゲーム数カウンタ（前半部）の値が「0」であるか否かを判別する（S488）。

40

【0818】

S488において、メインCPU101が、CZゲーム数カウンタ（前半部）の値が「0」でないと判別したとき（S488がNO判定の場合）、メインCPU101は、CZ1（CZ2）中処理を終了するとともに、CZ中スタート時処理（図113参照）も終了する。

【0819】

一方、S488において、メインCPU101が、CZゲーム数カウンタ（前半部）の値が「0」とであると判別したとき（S488がYES判定の場合）、メインCPU101は、次遊技の遊技状態にCZ1又はCZ2の後半部をセットする（S489）。そして、S489の処理後、メインCPU101は、CZ1（CZ2）中処理を終了するとともに、

50

C Z 中スタート時処理（図 1 1 3 参照）も終了する。

【0820】

ここで再度、S 4 8 1 の処理に戻って、S 4 8 1 が N O 判定の場合、メイン C P U 1 0 1 は、現ゲームが C Z 1 又は C Z 2 の後半部の 1 ゲーム目であるか否かを判別する（S 4 9 0）。S 4 9 0 において、メイン C P U 1 0 1 が、現ゲームが C Z 1 又は C Z 2 の後半部の 1 ゲーム目でないと判別したとき（S 4 9 0 が N O 判定の場合）、メイン C P U 1 0 1 は、後述の S 4 9 5 の処理を行う。

【0821】

一方、S 4 9 0 において、メイン C P U 1 0 1 が、現ゲームが C Z 1 又は C Z 2 の後半部の 1 ゲーム目であると判別したとき（S 4 9 0 が Y E S 判定の場合）、メイン C P U 1 0 1 は、C Z 中 A R T 抽籤テーブル（図 4 4 A 及び 4 4 B 参照）を参照し、前半部のランク（モード又はポイント）に基づいて A R T 抽籤処理を行う（S 4 9 1）。

【0822】

次いで、メイン C P U 1 0 1 は、A R T 抽籤に当籤したか否かを判別する（S 4 9 2）。S 4 9 2 において、メイン C P U 1 0 1 が、A R T 抽籤に当籤しなかったと判別したとき（S 4 9 2 が N O 判定の場合）、メイン C P U 1 0 1 は、後述の S 4 9 4 の処理を行う。

【0823】

一方、S 4 9 2 において、メイン C P U 1 0 1 が、A R T 抽籤に当籤したと判別したとき（S 4 9 2 が Y E S 判定の場合）、メイン C P U 1 0 1 は、A R T セット数に「1」を加算し、A R T レベル決定テーブル（図 4 8 A 参照）を参照して A R T レベルの抽籤を行い、その抽籤結果をセットする（S 4 9 3）。

【0824】

S 4 9 3 の処理後又は S 4 9 2 が N O 判定の場合、メイン C P U 1 0 1 は、C Z ゲーム数カウンタ（後半部）に所定値をセットする（S 4 9 4）。なお、S 4 9 4 の処理において、例えば、A R T 抽籤に当籤している場合には、C Z ゲーム数カウンタ（後半部）に「4」がセットされ、A R T 抽籤に非当籤である場合には、C Z ゲーム数カウンタ（後半部）に「3」がセットされる。

【0825】

S 4 9 4 の処理後又は S 4 9 0 が N O 判定の場合、メイン C P U 1 0 1 は、C Z 中 A R T 抽籤テーブル（図 4 4 C 参照）を参照して、内部当籤役（サブフラグ）に基づいて A R T 抽籤処理を行う（S 4 9 5）。

【0826】

次いで、メイン C P U 1 0 1 は、A R T 抽籤に当籤したか否かを判別する（S 4 9 6）。S 4 9 6 において、メイン C P U 1 0 1 が、A R T 抽籤に当籤しなかったと判別したとき（S 4 9 6 が N O 判定の場合）、メイン C P U 1 0 1 は、後述の S 4 9 8 の処理を行う。

【0827】

一方、S 4 9 6 において、メイン C P U 1 0 1 が、A R T 抽籤に当籤したと判別したとき（S 4 9 6 が Y E S 判定の場合）、メイン C P U 1 0 1 は、A R T セット数に「1」を加算し、A R T レベル決定テーブル（図 4 8 A 参照）を参照して A R T レベルの抽籤を行い、その抽籤結果をセットする（S 4 9 7）。

【0828】

S 4 9 7 の処理後又は S 4 9 6 が N O 判定の場合、メイン C P U 1 0 1 は、C Z ゲーム数カウンタ（後半部）の値を 1 減算する（S 4 9 8）。次いで、メイン C P U 1 0 1 は、C Z ゲーム数カウンタ（後半部）の値が「0」であるか否かを判別する（S 4 9 9）。

【0829】

S 4 9 9 において、メイン C P U 1 0 1 が、C Z ゲーム数カウンタ（後半部）の値が「0」でないと判別したとき（S 4 9 9 が N O 判定の場合）、メイン C P U 1 0 1 は、C Z 1（C Z 2）中処理を終了するとともに、C Z 中スタート時処理（図 1 1 3 参照）も終了する。

【0830】

一方、S 4 9 9において、メインCPU 1 0 1が、C Zゲーム数カウンタ（後半部）の値が「0」であると判別したとき（S 4 9 9がYES判定の場合）、メインCPU 1 0 1は、ARTセット数が「1」以上であるか否かを判別する（S 5 0 0）。

【0 8 3 1】

S 5 0 0において、メインCPU 1 0 1が、ARTセット数が「1」以上であると判別したとき（S 5 0 0がYES判定の場合）、メインCPU 1 0 1は、次遊技の遊技状態にART準備状態をセットする（S 5 0 1）。そして、S 5 0 1の処理後、メインCPU 1 0 1は、C Z 1（C Z 2）中処理を終了するとともに、C Z中スタート時処理（図1 1 3参照）も終了する。

【0 8 3 2】

一方、S 5 0 0において、メインCPU 1 0 1が、ARTセット数が「1」以上でないと判別したとき（S 5 0 0がNO判定の場合）、メインCPU 1 0 1は、次遊技の遊技状態にC Z失敗時の状態をセットする（S 5 0 2）。そして、S 5 0 2の処理後、メインCPU 1 0 1は、C Z 1（C Z 2）中処理を終了するとともに、C Z中スタート時処理（図1 1 3参照）も終了する。

【0 8 3 3】

【C Z 3中処理】

次に、図1 1 6を参照して、C Z中スタート時処理（図1 1 3参照）中のS 4 7 5で行うC Z 3中処理について説明する。なお、図1 1 6は、C Z 3中処理の手順を示すフローチャートである。

【0 8 3 4】

まず、メインCPU 1 0 1は、C Z中ART抽籤テーブル（図4 5参照）を参照し、内部当籤役（サブフラグ）に基づいてART抽籤処理を行う（S 5 1 1）。

【0 8 3 5】

次いで、メインCPU 1 0 1は、ART抽籤に当籤したか否かを判別する（S 5 1 2）。S 5 1 2において、メインCPU 1 0 1が、ART抽籤に当籤しなかったと判別したとき（S 5 1 2がNO判定の場合）、メインCPU 1 0 1は、後述のS 5 1 8の処理を行う。

【0 8 3 6】

一方、S 5 1 2において、メインCPU 1 0 1が、ART抽籤に当籤したと判別したとき（S 5 1 2がYES判定の場合）、メインCPU 1 0 1は、ARTセット数に「1」を加算し、CTセット数に「1」を加算する（S 5 1 3）。次いで、メインCPU 1 0 1は、S 5 1 2のART抽籤においてフリーズに当籤したか否かを判別する（S 5 1 4）。

【0 8 3 7】

S 5 1 4において、メインCPU 1 0 1が、フリーズに当籤しなかったと判別したとき（S 5 1 4がNO判定の場合）、メインCPU 1 0 1は、後述のS 5 1 6の処理を行う。一方、S 5 1 4において、メインCPU 1 0 1が、フリーズに当籤したと判別したとき（S 5 1 4がYES判定の場合）、メインCPU 1 0 1は、遊技の進行を一時的に停止するフリーズ処理を行う（S 5 1 5）。

【0 8 3 8】

S 5 1 5の処理後又はS 5 1 4がNO判定の場合、メインCPU 1 0 1は、ARTレベル決定テーブル（図4 8 A参照）を参照してARTレベルの抽籤処理を行い、その抽籤結果（ARTレベル）をセットする（S 5 1 6）。次いで、メインCPU 1 0 1は、次遊技の遊技状態にART準備状態をセットする（S 5 1 7）。そして、S 5 1 7の処理後、メインCPU 1 0 1は、C Z 3中処理を終了するとともに、C Z中スタート時処理（図1 1 3参照）も終了する。

【0 8 3 9】

ここで再度、S 5 1 2の処理に戻って、S 5 1 2がNO判定の場合、メインCPU 1 0 1は、C Zゲーム数カウンタの値を1減算する（S 5 1 8）。次いで、メインCPU 1 0 1は、C Zゲーム数カウンタの値が「0」であるか否かを判別する（S 5 1 9）。

【0 8 4 0】

10

20

30

40

50

S 5 1 9において、メインCPU101が、C Zゲーム数カウンタの値が「0」でないと判別したとき（S 5 1 9がNO判定の場合）、メインCPU101は、C Z 3中処理を終了するとともに、C Z中スタート時処理（図113参照）も終了する。

【0841】

一方、S 5 1 9において、メインCPU101が、C Zゲーム数カウンタの値が「0」であると判別したとき（S 5 1 9がYES判定の場合）、メインCPU101は、ARTセット数に「1」を加算し、ARTレベル決定テーブル（図48A参照）を参照してARTレベルの抽籤を行い、その抽籤結果をセットする（S 5 2 0）。次いで、メインCPU101は、次遊技の遊技状態にART準備状態をセットする（S 5 2 1）。そして、S 5 2 1の処理後、メインCPU101は、C Z 3中処理を終了するとともに、C Z中スタート時処理（図113参照）も終了する。

10

【0842】

〔通常ART中スタート時処理〕

次に、図117を参照して、状態別制御処理（図104参照）中のS 4 1 1で行う通常ART中スタート時処理について説明する。なお、図117は、通常ART中スタート時処理の手順を示すフローチャートである。

【0843】

まず、メインCPU101は、ART継続ゲーム数カウンタの値に「1」を加算する（S 5 3 1）。なお、ART継続ゲーム数カウンタは、通常ARTが継続したゲーム数（消化ゲーム数）を計数するカウンタである。また、本実施形態では、ART継続ゲーム数カウンタの他に、通常ARTが継続可能なゲーム数を計数するART終了ゲーム数カウンタも設ける。そして、本実施形態のパチスロ1では、ART継続ゲーム数カウンタの値とART終了ゲーム数カウンタの値とを比較し、ART継続ゲーム数カウンタの値がART終了ゲーム数カウンタの値に到達すると、ART遊技状態が終了する。

20

【0844】

次いで、メインCPU101は、ART中CT抽籤テーブル（図50参照）を参照し、現在のCT抽籤状態及び内部当籤役（サブフラグ）に基づいてCT抽籤処理を行う（S 5 3 2）。次いで、メインCPU101は、CT抽籤に当籤したか否かを判別する（S 5 3 3）。S 5 3 3において、メインCPU101が、CT抽籤に当籤しなかったと判別したとき（S 5 3 3がNO判定の場合）、メインCPU101は、後述のS 5 3 6の処理を行う。

30

【0845】

一方、S 5 3 3において、メインCPU101が、CT抽籤に当籤したと判別したとき（S 5 3 3がYES判定の場合）、メインCPU101は、CTセット数に「1」を加算し、CTゲーム数カウンタの値に「8」をセットする（S 5 3 4）。次いで、メインCPU101は、次遊技の遊技状態に当籤した種別のCTをセットする（S 5 3 5）。

【0846】

S 5 3 5の処理後又はS 5 3 3がNO判定の場合、メインCPU101は、ARTレベル決定テーブル（図48B参照）を参照し、ART継続ゲーム数カウンタの値に基づいてARTレベルを抽籤し、その抽籤結果をセットする（S 5 3 6）。次いで、メインCPU101は、通常ART中高確率抽籤テーブル（図49参照）を参照し、現在のCT抽籤状態及び内部当籤役（サブフラグ）に基づいて、移行先のCT抽籤状態を抽籤し、その抽籤結果をセットする（S 5 3 7）。

40

【0847】

次いで、メインCPU101は、通常ART中上乗せ抽籤テーブル（図51参照）を参照し、内部当籤役（サブフラグ）に基づいてARTゲーム数の上乗せ抽籤処理を行う（S 5 3 8）。次いで、メインCPU101は、上乗せ抽籤に当籤したか否かを判別する（S 5 3 9）。

【0848】

S 5 3 9において、メインCPU101が、上乗せ抽籤に当籤しなかったと判別したとき（S 5 3 9がNO判定の場合）、メインCPU101は、後述のS 5 4 1の処理を行う。

50

一方、S 5 3 9において、メインCPU 1 0 1が、上乗せ抽籤に当籤したと判別したとき（S 5 3 9がYES判定の場合）、メインCPU 1 0 1は、当籤結果をART終了ゲーム数カウンタに加算する（S 5 4 0）。

【0 8 4 9】

S 5 4 0の処理後又はS 5 3 9がNO判定の場合、メインCPU 1 0 1は、ART継続ゲーム数カウンタの値が、ART終了ゲーム数カウンタの値に達したか否かを判定する（S 5 4 1）。S 5 4 1において、メインCPU 1 0 1が、ART継続ゲーム数カウンタの値が、ART終了ゲーム数カウンタの値に達していないと判別したとき（S 5 4 1がNO判定の場合）、メインCPU 1 0 1は、通常ART中スタート時処理を終了するとともに、状態別制御処理（図1 0 4参照）も終了する。

10

【0 8 5 0】

一方、S 5 4 1において、メインCPU 1 0 1が、ART継続ゲーム数カウンタの値が、ART終了ゲーム数カウンタの値に達したと判別したとき（S 5 4 1がYES判定の場合）、メインCPU 1 0 1は、ARTセット数を1減算する（S 5 4 2）。次いで、メインCPU 1 0 1は、ART終了時の状態をセットする（S 5 4 3）。そして、S 5 4 3の処理後、メインCPU 1 0 1は、通常ART中スタート時処理を終了するとともに、状態別制御処理（図1 0 4参照）も終了する。

【0 8 5 1】

〔CT中スタート時処理〕

次に、図1 1 8を参照して、状態別制御処理（図1 0 4参照）中のS 4 1 3で行うCT中スタート時処理について説明する。なお、図1 1 8は、CT中スタート時処理の手順を示すフローチャートである。

20

【0 8 5 2】

まず、メインCPU 1 0 1は、CT中上乗せ抽籤テーブル（図5 5参照）を参照し、内部当籤役（サブフラグ）に基づいてARTゲーム数の上乗せ抽籤を行う（S 5 5 1）。次いで、メインCPU 1 0 1は、上乗せ抽籤に当籤したか否かを判別する（S 5 5 2）。

【0 8 5 3】

S 5 5 2において、メインCPU 1 0 1が、上乗せ抽籤に当籤しなかったと判別したとき（S 5 5 2がNO判定の場合）、メインCPU 1 0 1は、後述のS 5 5 6の処理を行う。一方、S 5 5 2において、メインCPU 1 0 1が、上乗せ抽籤に当籤したと判別したとき（S 5 5 2がYES判定の場合）、メインCPU 1 0 1は、当籤結果をART終了ゲーム数カウンタに加算する（S 5 5 3）。なお、上述したように、本実施形態のパチスロ1では、同一のCT中にサブフラグ「3連チリリプ（3連チリリプA又は3連チリリプB）」に当籤した回数が増えるほど、1回の抽籤当りの上乗せ量が増える。

30

【0 8 5 4】

S 5 5 3の処理後、メインCPU 1 0 1は、内部当籤役がサブフラグEX「3連チリリプ」（又は「確定役」）に対応する役であるか否か、すなわち、内部当籤役「F__確チリリプ」又は「F__1確チリリプ」であり、かつ、図1 1 1中のS 4 4 4のフラグ変換抽籤処理に当籤したか否かを判別する（S 5 5 4）。

【0 8 5 5】

S 5 5 4において、メインCPU 1 0 1が、内部当籤役がサブフラグEX「3連チリリプ」に対応する役でないと判別したとき（S 5 5 4がNO判定の場合）、メインCPU 1 0 1は、CT中スタート時処理を終了するとともに、状態別制御処理（図1 0 4参照）も終了する。

40

【0 8 5 6】

一方、S 5 5 4において、メインCPU 1 0 1が、内部当籤役がサブフラグEX「3連チリリプ」に対応する役であると判別したとき（S 5 5 4がYES判定の場合）、メインCPU 1 0 1は、CTゲーム数カウンタの値に「1」を加算する（S 5 5 5）。そして、S 5 5 5の処理後、メインCPU 1 0 1は、CT中スタート時処理を終了するとともに、状態別制御処理（図1 0 4参照）も終了する。

50

【0857】

ここで再度、S552の処理に戻って、S552がNO判定の場合、メインCPU101は、CT中CT抽籤処理を行う（S556）。この処理では、メインCPU101は、主に、CTセット数の上乗せ抽籤を行う。なお、CT中CT抽籤処理の詳細については、後述の図119を参照しながら後で説明する。

【0858】

次いで、メインCPU101は、CTゲーム数カウンタの値を1減算する（S557）。次いで、メインCPU101は、CTゲーム数カウンタの値が「0」であるか否かを判別する（S558）。

【0859】

S558において、メインCPU101が、CTゲーム数カウンタの値が「0」でないと判別したとき（S558がNO判定の場合）、メインCPU101は、CT中スタート時処理を終了するとともに、状態別制御処理（図104参照）も終了する。一方、S558において、メインCPU101が、CTゲーム数カウンタの値が「0」であると判別したとき（S558がYES判定の場合）、メインCPU101は、CTセット数が「1」以上であるか否かを判別する（S559）。

【0860】

S559において、メインCPU101が、CTセット数が「1」以上であると判別したとき（S559がYES判定の場合）、メインCPU101は、CTセット数を1減算する（S560）。次いで、メインCPU101は、CTゲーム数カウンタの値に「8」をセットする（S561）。そして、S561の処理後、メインCPU101は、CT中スタート時処理を終了するとともに、状態別制御処理（図104参照）も終了する。

【0861】

一方、S559において、メインCPU101が、CTセット数が「1」以上でないと判別したとき（S559がNO判定の場合）、メインCPU101は、次遊技の遊技状態に通常ARTをセットする（S562）。そして、S562の処理後、メインCPU101は、CT中スタート時処理を終了するとともに、状態別制御処理（図104参照）も終了する。

【0862】

[CT中CT抽籤処理]

次に、図119を参照して、CT中スタート時処理（図118参照）中のS556で行うCT中CT抽籤処理について説明する。なお、図119は、CT中CT抽籤処理の手順を示すフローチャートである。

【0863】

まず、メインCPU101は、CT中CT抽籤テーブルをセットする（S571）。なお、ここで、セットされるCT中CT抽籤テーブルは、上記図56で説明したCT中セット数上乗せ抽籤テーブルであるが、ソースプログラム上で実際にセットされるCT中CT抽籤テーブル（CT中セット数上乗せ抽籤テーブル）の構成については、後述の図122を参照しながら後で説明する。

【0864】

次いで、メインCPU101は、テーブルデータ取得処理を行う（S572）。この処理では、メインCPU101は、CT中CT抽籤処理で参照する抽籤テーブルのアドレスを取得する。なお、テーブルデータ取得処理の詳細については、後述の図120を参照しながら後で説明する。

【0865】

次いで、メインCPU101は、1バイト抽籤処理を行う（S573）。この処理では、メインCPU101は、CTセットの上乗せ抽籤を行う。なお、1バイト抽籤処理の詳細については、後述の図123を参照しながら後で説明する。

【0866】

次いで、メインCPU101は、1バイト抽籤処理に当籤したか否かを判別する（S57

10

20

30

40

50

4)。S 5 7 4において、メインCPU101が、1バイト抽籤処理に当籤しなかったと判別したとき（S 5 7 4がNO判定の場合）、メインCPU101は、CT中CT抽籤処理を終了し、処理をCT中スタート時処理（図118参照）のS 5 5 7に移す。

【0867】

一方、S 5 7 4において、メインCPU101が、1バイト抽籤処理に当籤したと判別したとき（S 5 7 4がYES判定の場合）、メインCPU101は、CTセット数に「1」を加算する（S 5 7 5）。そして、S 5 7 5の処理後、メインCPU101は、CT中CT抽籤処理を終了し、処理をCT中スタート時処理（図118参照）のS 5 5 7に移す。

【0868】

〔テーブルデータ取得処理〕

次に、図120～図122を参照して、CT中CT抽籤処理（図119参照）中のS 5 7 2で行うテーブルデータ取得処理について説明する。図120は、テーブルデータ取得処理の手順を示すフローチャートである。また、図121は、テーブルデータ取得処理を実行するためのソースプログラムの一例を示す図であり、図122は、CT中CT抽籤処理及びテーブルデータ取得処理のソースプログラム上で、実際に参照されるCT中CT抽籤テーブル（図56で説明したCT中セット数上乘せ抽籤テーブルに対応）の構成の一例を示す図である。なお、図122に示すCT中CT抽籤テーブルに格納されている具体的な各種抽籤値は一例である。

【0869】

なお、本実施形態において、CT中CT抽籤処理で参照する抽籤値を取得する際、2段階のアドレス算出処理（1段階目及び2段階目のテーブルデータ取得処理）を経て、抽籤値が格納されているアドレスを算出する。まず、1段階目のテーブルデータ取得処理（後述のS 5 8 2及びS 5 8 3の処理）では、内部当籤役（実際にはサブフラグD）に対応付けられた「選択値（1バイト）」が取得される。なお、選択値には、内部当籤役の種別毎に設けられ、内部当籤役が抽籤対象であるか否かが判別可能であり且つ内部当籤役に対応付けられた抽籤テーブルの配置先を指定可能な値（相対値）が規定される。また、本実施形態では、CT中CT抽籤処理の抽籤結果が非当籤となる内部当籤役（実際にはサブフラグD）に対して予め選択値「0」を規定し、それらの内部当籤役を1段階目のテーブルデータ取得処理の時点で「ハズレ」として扱う。そして、2段階目のテーブルデータ取得処理（後述のS 5 8 5～S 5 8 7の処理）では、「0」以外の選択値が規定された内部当籤役の抽籤値が格納されたアドレスが算出される（抽籤テーブルの基準アドレス（2バイト）から相対値（選択値）を加算したアドレスが算出される）。

【0870】

まず、メインCPU101は、CT中CT抽籤選択テーブル（不図示）を参照して、CT中CT抽籤テーブルのアドレスを算出するための1段階目及び2段階目の加算選択データのアドレス、並びに、CT抽籤の抽籤回数（本実施形態では、2回）を取得する（S 5 8 1）。次いで、メインCPU101は、1段階目の加算選択データのアドレスをCT中CT抽籤テーブルのアドレスに加算して、1段階目の選択アドレスを算出する（S 5 8 2）。

【0871】

次いで、メインCPU101は、算出した1段階目の選択アドレスに格納されている選択値を取得する（S 5 8 3）。次いで、メインCPU101は、選択値が「0」であるか否かを判別する（S 5 8 4）。

【0872】

S 5 8 4において、メインCPU101が、選択値が「0」であると判別したとき（S 5 8 4がYES判定の場合）、メインCPU101は、テーブルデータ取得処理を終了し、処理をCT中CT抽籤処理（図119参照）のS 5 7 3に移す。

【0873】

一方、S 5 8 4において、メインCPU101が、選択値が「0」でないと判別したとき（S 5 8 4がNO判定の場合）、メインCPU101は、選択アドレスに選択値を加算して、2段階目の選択アドレスを算出する（S 5 8 5）。次いで、メインCPU101は、

10

20

30

40

50

2段階目の選択アドレスに2段階目の加算選択データのアドレスを加算して、選択アドレスを算出する（S586）。

【0874】

次いで、メインCPU101は、S586で算出した選択アドレスに格納されている選択値を取得し、該選択値を選択アドレスに加算して、CT中CT抽籤テーブル内において参照するアドレスを算出する（S587）。そして、S587の処理後、メインCPU101は、テーブルデータ取得処理を終了し、処理をCT中CT抽籤処理（図119参照）のS573に移す。

【0875】

本実施形態では、上述のようにしてテーブルデータ取得処理が行われる。そして、上述したテーブルデータ取得処理は、メインCPU101が、図121のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【0876】

その中で、1段階目のテーブルデータ取得処理（S581～S584の処理）では、図122に示すCT中CT当籤抽籤テーブル中のアドレス「dCTCTSTTB」～「dCTCTSRER-1」までの領域に格納されたテーブル（当籤役別テーブル選択相対テーブル）が参照される。また、当籤役別テーブル選択相対テーブル（抽籤テーブル選択テーブル）内において、CT当籤がハズレとなる各役（サブフラグD「ハズレ」、「サボテン」、「弱チェリー」、「強チェリー」、「確定役」、「3連チリリブ」）の選択テーブルのアドレスには、上述した選択値（相対値）として「0」が格納される。そして、1段階目のテーブルデータ取得処理（アドレス算出処理）では、選択値が「0」である場合には、抽籤結果を「ハズレ」にする（上記S584の判定処理参照）。

【0877】

上記構成のCT中CT当籤抽籤テーブルでは、1段階目のテーブルデータ取得処理で参照する当籤役別テーブル選択相対テーブルにおいて、役の種別だけで「ハズレ」を設定することができるので、抽籤テーブルに「ハズレ」役の抽籤値を規定する必要がなくなる。それゆえ、本実施形態では、CT中CT当籤抽籤テーブルにおいて、「ハズレ」役の抽籤値データ（「0」）を格納する必要がなくなり、メインROM102のテーブル領域の容量を節約することができる。

【0878】

また、2段階目（サブフラグD「リーチ目リブ」取得時）のテーブルデータ取得処理（S585～S587の処理）中のS587の処理では、算出された抽籤テーブルアドレスに基づいて、図122に示すCT中CT当籤抽籤テーブル（図56のCT中セット数上乘せ抽籤テーブルに対応）内から、通常CT状態時に用いる抽籤テーブル（先頭アドレス「dNMCTCTSRER」）又は高確率CT状態の抽籤テーブル（先頭アドレス「dSPCTCTSRER」）の一方が選択される。

【0879】

高確率CT状態時に用いる抽籤テーブルでは、図122に示すように、先頭アドレス「dSPCTCTSRER」の次のアドレス領域に1バイトデータからなる「判定ビット」（判定データ）が格納される。判定ビットには、抽籤対象の抽籤値が格納されたアドレスの範囲を示すデータが格納される。図122に示す例のように、高確率CT状態の抽籤テーブルの「判定ビット」の格納領域の次のアドレスにのみ抽籤対象（高確CT）の抽籤値が格納されている場合には、判定ビットの格納領域には、ビット0にのみ「1」が格納された1バイトデータ「00000001B」が判定ビットとして格納される。一方、通常CT状態時に用いる抽籤テーブルのように、判定ビットの格納領域の次のアドレス及び次々アドレスに抽籤対象（高確CT及び通常CT）の抽籤値が格納されている場合には、図122に示すように、判定ビットの格納領域には、ビット0及び1にのみ「1」が格納された1バイトデータ「00000011B」が判定ビットとして格納される。このような判定ビットを設けた場合、抽籤テーブルにおいて、判定ビットにおいてビットデータが「0」となるアドレスの領域に抽籤対象外の抽籤値データを格納する必要がなくなる。

10

20

30

40

50

【0880】

さらに、高確率CT状態時に用いる抽籤テーブルでは、図122に示すように、「判定ビット」の次のアドレス「dSPCTCTS_RER+2」に高確率CT当籤の抽籤値が格納され、高確率CT当籤の抽籤値としてアドレス「cABS_HIT」に規定されたデータが格納される。本実施形態では、このアドレス「cABS_HIT」に規定されているデータは、当籤確定（100%当籤）を示すデータ（以下、「確定データ」という）である。また、本実施形態では、CT中CT当籤抽籤テーブルにおいてハズレ用の抽籤値データ（「0」）を設ける必要がないので、図122に示すように、アドレス「cABS_HIT」に規定されている確定データに「0」を規定することができる。すなわち、上記構成のCT中CT当籤抽籤テーブルでは、抽籤値「0」を確定データとして使用することができる。

10

【0881】

なお、図56のCT中セット数上乘せ抽籤テーブルに説明したように、本実施形態では、高確率CT状態時のCT中セット数上乘せ抽籤では、必ず、「高確率CT当籤」が決定される。それゆえ、本実施形態では、ソースプログラム上では、図122に示すCT中CT当籤抽籤テーブルにおいて、高確率CT当籤の抽籤値として確定データを格納することができる。

【0882】

上述した2段階目（サブフラグD「リーチ目リブ」取得時）の抽籤テーブルのように、判定ビットを構成する各ビットデータの値により、CT抽籤の抽籤対象役及び抽籤対象外の役（サブフラグD）を判別することにより、抽籤対象外の役の抽籤値データ（ハズレデータ）をテーブルに格納する必要がなくなる。また、抽籤対象役の当籤確率が100%である確定データとしては、抽籤値「0」を用いることができる。これらのことから、本実施形態では、CT中CT当籤抽籤テーブル（CT中セット数上乘せ抽籤テーブル）の容量を圧縮することができ、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。また、本実施形態では、この技術をCT中CT当籤処理で使用する例を説明したが、本発明はこれに限定されず、ART抽籤（図115参照）、ARTゲーム数上乘せ抽籤（図117参照）、後述のCT抽籤（後述の図154参照）及び後述のCZの引き戻し抽籤（後述の図157参照）等で使用してもよい。

20

30

【0883】

[1バイト抽籤処理]

次に、図123及び図124を参照して、CT中CT抽籤処理（図119参照）中のS573で行う1バイト抽籤処理について説明する。図123は、1バイト抽籤処理の手順を示すフローチャートである。また、図124は、1バイト抽籤処理を実行するためのソースプログラムの一例を示す図である。

【0884】

まず、メインCPU101は、メインRAM103内の乱数格納領域（不図示）に格納されているCT中CT抽籤用の1バイト乱数値（0～255：乱数回路110の乱数レジスタ4のソフトラッチ乱数）をセットする（S591）。次いで、メインCPU101は、テーブルデータ取得処理中のS587で算出したアドレスに基づいて、CT中CT抽籤テーブルから抽籤判定データ（図122の2段階目に規定されている抽籤テーブル中の判定ビット）を取得する（S592）。また、この処理では、メインCPU101は、抽籤回数の初期値として、判定ビットのビット数「8」をセットする。

40

【0885】

次いで、メインCPU101は、抽籤判定データが抽籤対象であるか否かを判別する（S593）。この判定処理では、メインCPU101は、現在の抽籤回数に対応付けられた判定ビット内のビットデータを参照し、該ビットデータが「1」であれば、抽籤対象であると判定する。なお、本実施形態では、判定ビット内のビット0～ビット7が、抽籤回数「8」～「1」にそれぞれ対応付けられている。

50

【0886】

S593において、メインCPU101が、抽籤判定データが抽籤対象でないと判別したとき（S593がNO判定の場合）、メインCPU101は、後述のS599の処理を行う。一方、S593において、メインCPU101が、抽籤判定データが抽籤対象であると判別したとき（S593がYES判定の場合）、メインCPU101は、CT中CT抽籤テーブルから抽籤値を取得する（S594）。

【0887】

次いで、メインCPU101は、抽籤値が「0」（当籤確定データ）であるか否かを判別する（S595）。

【0888】

S595において、メインCPU101が、抽籤値が「0」とであると判別したとき（S595がYES判定の場合）、メインCPU101は、1バイト抽籤処理を終了し、処理をCT中CT抽籤処理（図119参照）のS574に移す。一方、S595において、メインCPU101が、抽籤値が「0」でないと判別したとき（S595がNO判定の場合）、メインCPU101は、CT抽籤（CTセット数の上乘せ抽籤）処理を行う（S596）。具体的には、メインCPU101は、乱数値（1バイト乱数値）から抽籤値を減算し、その減算結果を乱数値とする。

【0889】

次いで、メインCPU101は、S596のCT抽籤に当籤したか否かを判別する（S597）。なお、S596のCT抽籤では、メインCPU101は、S596の減算結果が「0」以下となった場合（いわゆる「桁かり」が生じた場合）に、当籤したと判定する。

【0890】

S597において、メインCPU101が、CT抽籤に当籤したと判別したとき（S597がYES判定の場合）、メインCPU101は、1バイト抽籤処理を終了し、処理をCT中CT抽籤処理（図119参照）のS574に移す。一方、S597において、メインCPU101が、CT抽籤に当籤しなかったと判別したとき（S597がNO判定の場合）、メインCPU101は、CT中CT抽籤テーブル内において参照する抽籤値の格納アドレス（抽籤アドレス）を次の抽籤アドレスに更新する（S598）。

【0891】

S598の処理後又はS593がNO判定の場合、メインCPU101は、抽籤回数を1減算する（S599）。次いで、メインCPU101は、抽籤回数が「0」であるか否かを判別する（S600）。

【0892】

S600において、メインCPU101が、抽籤回数が「0」でないと判別したとき（S600がNO判定の場合）、メインCPU101は、処理をS593に戻し、S593以降の処理を繰り返す。一方、S600において、メインCPU101が、抽籤回数が「0」とであると判別したとき（S600がYES判定の場合）、メインCPU101は、1バイト抽籤処理を終了し、処理をCT中CT抽籤処理（図119参照）のS574に移す。

【0893】

本実施形態では、上述のようにして1バイト抽籤処理が行われる。なお、上述した1バイト抽籤処理は、メインCPU101が、図124のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【0894】

1バイト抽籤処理中のS591の乱数取得処理では、図124に示すように、ソースプログラム上において、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行うメインCPU101専用命令コードである「LDQ」命令が用いられる。それゆえ、本実施形態では、1バイト抽籤処理においても、Qレジスタ（拡張レジスタ）を用いた命令コードが用いることにより、直値により、メインROM102、メインRAM103やメモリーマップE/Oにアクセスすることができるので、アドレス設定に係る命令コードを省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減する

10

20

30

40

50

ことができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【0895】

〔BB中スタート時処理〕

次に、図125を参照して、状態別制御処理（図104参照）中のS415で行うBB中スタート時処理について説明する。なお、図125は、BB中スタート時処理の手順を示すフローチャートである。

【0896】

まず、メインCPU101は、ボーナス中ARTゲーム数上乗せ抽籤テーブル（図59参照）を参照し、内部当籤役に基づいてARTゲーム数の上乗せ抽籤処理を行う（S611）。次いで、メインCPU101は、上乗せ抽籤に当籤したか否かを判別する（S612）。

10

【0897】

S612において、メインCPU101が、上乗せ抽籤に当籤しなかったと判別したとき（S612がNO判定の場合）、メインCPU101は、BB中スタート時処理を終了するとともに、状態別制御処理（図104参照）も終了する。一方、S612において、メインCPU101が、上乗せ抽籤に当籤したと判別したとき（S612がYES判定の場合）、メインCPU101は、当籤結果（上乗せゲーム数）をART終了ゲーム数カウンタに加算する（S613）。

20

【0898】

次いで、メインCPU101は、ARTセット数が「0」であるか否かを判別する（S614）。S614において、メインCPU101が、ARTセット数が「0」でないと判別したとき（S614がNO判定の場合）、メインCPU101は、BB中スタート時処理を終了するとともに、状態別制御処理（図104参照）も終了する。

【0899】

一方、S614において、メインCPU101が、ARTセット数が「0」とであると判別したとき（S614がYES判定の場合）、メインCPU101は、ARTセット数に「1」を加算する（S615）。そして、S615の処理後、メインCPU101は、BB中スタート時処理を終了するとともに、状態別制御処理（図104参照）も終了する。

30

【0900】

〔引込優先順位格納処理〕

次に、図126及び図127を参照して、メインフロー（図82参照）中のS212で行う引込優先順位格納処理について説明する。図126は、引込優先順位格納処理の手順を示すフローチャートである。また、図127は、引込優先順位格納処理中の後述のS625及びS626の処理を実行するためのソースプログラムの一例を示す図である。

【0901】

まず、メインCPU101は、検索リール数に「3」をセットする（S621）。次いで、メインCPU101は、引込優先順位テーブル選択処理を行う（S622）。この処理では、内部当籤役及び作動ストップボタンに基づいて、引込優先順位テーブル（図27参照）が選択される。

40

【0902】

次いで、メインCPU101は、引込優先順位格納領域選択処理を行う（S623）。この処理では、検索対象のリールの引込優先順位データ格納領域が選択される。次いで、メインCPU101は、図柄チェック数（回数）として「20」をセットする（S624）。

【0903】

次いで、メインCPU101は、図柄コード取得処理を行う（S625）。この処理では、図柄チェック数に対応した入賞作動フラグ格納領域及び図柄コード格納領域を参照して、図柄コードを取得する。なお、図柄コード取得処理の詳細については、後述の図128を参照しながら後で説明する。

50

【0904】

次いで、メインCPU101は、論理積演算処理を行う（S626）。この処理では、メインCPU101は、入賞作動フラグデータの生成処理を行う。論理積演算処理の詳細については、後述の図133を参照しながら後で説明する。

【0905】

次いで、メインCPU101は、引込優先順位取得処理を行う（S627）。この処理では、メインCPU101は、入賞作動フラグ（入賞役）格納領域（図28～図30参照）内においてビットが「1」にセットされており、かつ、当り要求フラグ格納領域でビットが「1」にされている役について、引込優先順位テーブル（図27参照）を参照して、引込優先順位データを取得する。なお、引込優先順位取得処理の詳細については、後述の図134及び図135を参照しながら後で説明する。

10

【0906】

次いで、メインCPU101は、取得した引込優先順位データをメインRAM103内の引込優先順位データ格納領域（不図示）に格納する（S628）。この際、引込優先順位データは、各優先順位の値と、格納領域のビットとが対応するように引込優先順位データ格納領域に格納される。

【0907】

なお、引込優先順位データ格納領域には、メインリールの種類毎に優先順位データの格納領域が設けられる。各引込優先順位データ格納領域には、対応するメインリールの各図柄位置「0」～「19」に応じて決定された引込優先順位データが格納される。本実施形態では、この引込優先順位データ格納領域を参照することにより、停止テーブルに基づいて決定された滑り駒数の他に、より適切な滑り駒数が存在するか否かを検索する。

20

【0908】

引込優先順位データ格納領域に格納される優先順位引込データの内容は、引込優先順位データを決定する際に参照された引込優先順位テーブル内の引込優先順位テーブル番号の種類によって異なる。また、引込優先順位データは、その値が大きいほど優先順位が高いことを表す。引込優先順位データを参照することにより、メインリールの周面に配された各図柄間における優先順位の相対的な評価が可能となる。すなわち、引込優先順位データとして最も大きい値が決定されている図柄が最も優先順位の高い図柄となる。したがって、引込優先順位データは、メインリールの周面に配された各図柄間の順位を示すものともいえる。なお、引込優先順位データの値が等しい図柄が複数存在する場合には、優先順序テーブルが規定する優先順序に従って1つの図柄が決定される。

30

【0909】

次いで、メインCPU101は、引込優先順位格納領域の更新処理を行う（S629）。この処理では、メインCPU101は、次のチェック図柄の引込優先順位データ格納領域をセットする。次いで、メインCPU101は、図柄チェック数を1減算する（S630）。次いで、メインCPU101は、図柄チェック数が「0」であるか否かを判別する（S631）。

【0910】

S631において、メインCPU101が、図柄チェック数が「0」でないと判別したとき（S631がNO判定の場合）、メインCPU101は、処理をS625の処理に戻し、S625以降の処理を繰り返す。一方、S631において、メインCPU101が、図柄チェック数が「0」であると判別したとき（S631がYES判定の場合）、メインCPU101は、検索対象リールの変更処理を行う（S632）。

40

【0911】

次いで、メインCPU101は、検索リール数を1減算する（S633）。次いで、メインCPU101は、検索リール数が「0」であるか否か、すなわち、全てのメインリールに対して上述した一連の処理が行われたか否かを判別する（S634）。

【0912】

S634において、メインCPU101が、検索リール数が「0」でないと判別したとき

50

(S 6 3 4 が N O 判定の場合)、メイン C P U 1 0 1 は、処理を S 6 2 2 の処理に戻し、S 6 2 2 以降の処理を繰り返す。一方、S 6 3 4 において、メイン C P U 1 0 1 が、検索リール数が「0」であると判別したとき (S 6 3 4 が Y E S 判定の場合)、メイン C P U 1 0 1 は、引込優先順位格納処理を終了し、処理をメインフロー (図 8 2 参照) の S 2 1 3 に移す。

【0913】

本実施形態では、上述のようにして引込優先順位格納処理が行われる。上述した引込優先順位格納処理中の S 6 2 5 及び S 6 2 6 の処理は、メイン C P U 1 0 1 が、図 1 2 7 のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【0914】

その中で、S 6 2 6 の論理積演算処理は、メイン C P U 1 0 1 が図 1 2 7 中のソースコード「CALLF SB__DAND__00」を実行することにより行われる。「CALLF」命令は、上述のようにメイン C P U 1 0 1 専用の 2 バイト命令コードであり、図 1 2 7 中のソースコード「CALLF SB__DAND__00」が実行されると、「SB__DAND__00」で指定されているアドレスに、処理をジャンプさせ、論理積演算処理が開始される。

【0915】

〔図柄コード取得処理〕

次に、図 1 2 8 ~ 図 1 3 2 を参照して、引込優先順位格納処理 (図 1 2 6 参照) 中の S 6 2 5 で行う図柄コード取得処理について説明する。図 1 2 8 は、図柄コード取得処理の手順を示すフローチャートであり、図 1 2 9 は、図柄コード取得処理を実行するためのソースプログラムの一例を示す図である。図 1 3 0 A は、図柄コード取得処理のソースプログラム上で、実際に参照される第 1 リール (左リール) 図柄配置テーブルの構成の一例を示す図であり、図 1 3 0 B は、第 1 リール図柄配置テーブルセット時に参照される図柄対応入賞作動テーブルの構成の一例を示す図である。図 1 3 1 A は、図柄コード取得処理のソースプログラム上で、実際に参照される第 2 リール (中リール) 図柄配置テーブルの構成の一例を示す図であり、図 1 3 1 B は、第 2 リール図柄配置テーブルセット時に参照される図柄対応入賞作動テーブルの構成の一例を示す図である。また、図 1 3 2 A は、図柄コード取得処理のソースプログラム上で、実際に参照される第 3 リール (右リール) 図柄配置テーブルの構成の一例を示す図であり、図 1 3 2 B は、第 3 リール図柄配置テーブルセット時に参照される図柄対応入賞作動テーブルの構成の一例を示す図である。

【0916】

まず、メイン C P U 1 0 1 は、入賞作動フラグ格納領域のクリア処理を行う (S 6 4 1)。この処理では、メイン C P U 1 0 1 は、入賞作動フラグ格納領域 (図 2 8 ~ 図 3 0 参照) 内の全ての格納領域に「0」をセットする。次いで、メイン C P U 1 0 1 は、第 1 リール図柄配置テーブル (図 1 3 0 A 参照) をセットする (S 6 4 2)。

【0917】

次いで、メイン C P U 1 0 1 は、第 1 リール (左リール 3 L) の停止時であるか否かを判別する (S 6 4 3)。

【0918】

S 6 4 3 において、メイン C P U 1 0 1 が、第 1 リール (左リール 3 L) の停止時であると判別したとき (S 6 4 3 が Y E S 判定の場合)、メイン C P U 1 0 1 は、後述の S 6 4 7 の処理を行う。一方、S 6 4 3 において、メイン C P U 1 0 1 が、第 1 リール (左リール 3 L) の停止時でないと判別したとき (S 6 4 3 が N O 判定の場合)、メイン C P U 1 0 1 は、第 2 リール図柄配置テーブル (図 1 3 1 A 参照) をセットする (S 6 4 4)。この処理では、S 6 4 2 の処理でセットされた第 1 リール図柄配置テーブルが、第 2 リール図柄配置テーブルで上書きされる。

【0919】

次いで、メイン C P U 1 0 1 は、第 2 リール (中リール 3 C) の停止時であるか否かを判別する (S 6 4 5)。

10

20

30

40

50

【0920】

S645において、メインCPU101が、第2リール（中リール3C）の停止時であると判別したとき（S645がYES判定の場合）、メインCPU101は、後述のS647の処理を行う。一方、S645において、メインCPU101が、第2リール（中リール3C）の停止時でないと判別したとき（S645がNO判定の場合）、メインCPU101は、第3リール図柄配置テーブル（図132A参照）をセットする（S646）。この処理では、S644の処理でセットされた第2リール図柄配置テーブルが、第3リール図柄配置テーブルで上書きされる。

【0921】

S646の処理後、又は、S643或いはS645がYES判定の場合、メインCPU101は、停止制御対象のリールに対する停止操作実行時の図柄チェック処理を行い、図柄チェック処理により取得された図柄に対応する図柄対応入賞作動テーブルを取得する（S647）。例えば、第1リール（左リール3L）停止時であり、停止操作時に有効ライン上に位置する図柄が「白7」である場合、メインCPU101は、図130B中のアドレス「dR1__SVN1」～アドレス「dR1__SVN2-1」の範囲のブロックに規定された図柄対応入賞作動テーブルの先頭アドレスを取得する。

【0922】

次いで、メインCPU101は、入賞作動フラグ格納領域をセットする（S648）。次いで、メインCPU101は、図101で説明した圧縮データ格納処理を行う（S649）。この処理では、メインCPU101は、主に、図柄対応入賞作動テーブルに格納された入賞可能な入賞作動フラグデータを、入賞作動フラグ格納領域内の対応する格納領域に転送（展開）する処理を行う。

【0923】

例えば、第1リール（左リール3L）停止時であり、停止操作時に有効ライン上に位置する図柄が「白7」である場合には、入賞可能な図柄組合せ（コンビネーション）は、図28～図30に示すように、第2格納領域に規定されるコンビネーション名称「C__2nd__A__01」、「C__2nd__A__01」及び「C__SP1__01」、第3格納領域に規定されるコンビネーション名称「C__9枚C__01」～「C__9枚C__03」、「C__9枚C__07」～「C__9枚C__09」及び「C__9枚E__01」、第4格納領域に規定されるコンビネーション名称「C__RB役A__01」、「C__RB役A__02」、「C__RB役B__01」～「C__RB役B__04」、「C__RB役C__01」及び「C__RB役C__02」、第6格納領域に規定されるコンビネーション名称「C__リーチ目リプC__01」～「C__リーチ目リプC__03」、「C__リーチ目リプD__01」、「C__リーチ目リプD__02」及び「C__リーチ目リプE__01」、並びに、第10格納領域に規定されるコンビネーション名称「C__BB1」である。

【0924】

この場合、図柄対応入賞作動テーブルに格納された入賞可能な入賞作動フラグデータの第1ブロック（第0～第7格納領域）の格納先は、図130Bに示すテーブル内のアドレス「dR1__SVN1+1」に格納されている1バイトの指定データ「01011100B」により指定される（図130B中のコメント「格納領域 +2, +3, +4, +6」欄参照）。また、図柄対応入賞作動テーブルに格納された入賞可能な入賞作動フラグデータの第2ブロック（第8～第11格納領域）の格納先は、図130Bに示すテーブル内のアドレス「dR1__SVN1+6」に格納されている1バイトの指定データ「10000000B」により指定される（図130B中のコメント「格納領域 +10」欄参照）。

【0925】

なお、本実施形態では、第1ブロックの指定データのビット0～ビット7が、格納先として、それぞれ第1ブロックの第0～第7格納領域を指定するビットであり、第2ブロックの指定データのビット0～ビット3が、格納先として、それぞれ第2ブロックの第8～第11格納領域を指定するビットである。そして、各ブロックの1バイトの指定データでは、入賞作動フラグデータの格納先となる入賞作動フラグ格納領域内の格納領域に対応する

10

20

30

40

50

ビットに「1」が格納される。

【0926】

それゆえ、例えば、第1リール（左リール3L）停止時であり、停止操作時に有効ライン上に位置する図柄が「白7」である場合、S649の処理において、図130Bに示すテーブル内のアドレス「dR1__SVN1+2」に格納されている入賞作動フラグデータ「00010000B」又は「00000100B」が、図柄対応入賞作動テーブルから入賞作動フラグ格納領域内の第2格納領域に転送され、アドレス「dR1__SVN1+3」に格納されている入賞作動フラグデータ「00100000B」又は「00000100B」が、図柄対応入賞作動テーブルから入賞作動フラグ格納領域内の第3格納領域に転送される。また、この場合、S649の処理において、図130Bに示すテーブル内のアドレス「dR1__SVN1+4」に格納されている入賞作動フラグデータ「10000000B」、「01000000B」又は「00100000B」が、図柄対応入賞作動テーブルから入賞作動フラグ格納領域内の第4格納領域に転送され、アドレス「dR1__SVN1+5」に格納されている入賞作動フラグデータ「00100000B」、「00010000B」又は「00001000B」が、図柄対応入賞作動テーブルから入賞作動フラグ格納領域内の第6格納領域に転送される。さらに、この場合、S649の処理において、図130Bに示すテーブル内のアドレス「dR1__SVN1+8」に格納されている入賞作動フラグデータ「10000000B」が、図柄対応入賞作動テーブルから入賞作動フラグ格納領域内の第10格納領域に転送される。

【0927】

S649の処理後、メインCPU101は、圧縮データ格納処理により更新された入賞作動フラグ格納領域をセットし、図柄コード格納領域をセットし、入賞作動フラグ格納領域のデータ長（本実施形態では12バイト）をセットする（S650）。そして、S650の処理後、メインCPU101は、図柄コード取得処理を終了し、処理を引込優先順位格納処理（図126参照）のS626に移す。

【0928】

本実施形態では、上述のようにして図柄コード取得処理が行われる。なお、上述した図柄コード取得処理は、メインCPU101が、図129のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中でも、S649の圧縮データ格納処理は、メインCPU101が図129中のソースコード「CALLF SB__BTEP__00」を実行することにより行われる。

【0929】

「CALLF」命令は、上述のようにメインCPU101専用の2バイト命令コードであり、図129中のソースコード「CALLF SB__BTEP__00」が実行されると、「SB__BTEP__00」で指定されているアドレスに、処理をジャンプさせ、圧縮データ格納処理が開始される。そして、この圧縮データ格納処理では、上述のように、各リールの図柄対応入賞作動フラグテーブルに格納された入賞作動フラグデータ（圧縮データ）が、入賞作動フラグ格納領域に展開（コピー）される。

【0930】

なお、本実施形態では、上述した図柄コード取得処理中のS647～S649で説明した処理手順で入賞に係るデータの圧縮・展開処理を行い、かつ、その処理の中で上述したメインCPU101専用命令コードを用いることにより、入賞に係るデータの圧縮・展開処理の効率化を図ることができるとともに、限られたメインRAM103の容量を有効活用することができる。

【0931】

また、本実施形態では、図柄コード取得処理中のS649の圧縮データ格納処理において、「CALLF」命令で指定するジャンプ先のアドレス「SB__BTEP__00」は、図97で説明した図柄設定処理中のS330の圧縮データ格納処理において、「CALLF」命令で指定するジャンプ先のアドレスと同じである。すなわち、本実施形態では、図柄コード取得処理で行う圧縮データ格納処理を実行するためのソースプログラムが、図柄設

定処理で行う圧縮データ格納処理を実行するためのソースプログラムと同じであり、S 6 4 9 及び S 3 3 0 の両処理において、圧縮データ格納処理のソースプログラムが共有化（モジュール化）されている。この場合、S 6 4 9 及び S 3 3 0 の両処理において、それぞれ別個に圧縮データ格納処理のソースプログラムを設ける必要が無くなるので、その分、ソースプログラムの容量（メインROM 1 0 2 の使用容量）を低減することができる。この結果、本実施形態では、メインROM 1 0 2 において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【0 9 3 2】

〔論理積演算処理〕

次に、図 1 3 3 を参照して、例えば、引込優先順位格納処理（図 1 2 6 参照）中の S 6 2 6 で行う論理積演算処理について説明する。図 1 3 3 は、論理積演算処理の手順を示すフローチャートである。なお、図 1 3 3 に示す論理積演算処理は、引込優先順位格納処理（図 1 2 6 参照）中の S 6 2 6 だけでなく、後述の引込優先順位取得処理（後述の図 1 3 4 及び図 1 3 5 参照）中の S 6 8 7 においても実行される。

【0 9 3 3】

引込優先順位格納処理（図 1 2 6 参照）中の S 6 2 6 で実行される論理積演算処理において、論理積演算される 2 つのデータは、上述した図柄コード取得処理中の S 6 5 0 でセットされた入賞作動フラグ格納領域のデータ、及び、図柄コード格納領域のデータである。そして、前者のデータが後述の「論理積先データ」に対応し、後者のデータが後述の「論理積元データ」に対応する。また、この場合、上述した図柄コード取得処理中の S 6 5 0 でセットされたデータ長（1 2 バイト）のバイト数「1 2」が後述の「論理積回数」に対応する。

【0 9 3 4】

一方、後述の引込優先順位取得処理（後述の図 1 3 4 及び図 1 3 5 参照）中の S 6 8 7 で実行される論理積演算処理において、論理積演算される 2 つのデータは、当り（引込）要求フラグ格納領域のデータ、及び、入賞作動フラグ格納領域のデータである。そして、前者のデータが後述の「論理積先データ」に対応し、後者のデータが後述の「論理積元データ」に対応する。また、この場合、後述の図 1 3 6 B 中に記載の R T 作動組み合わせ表示フラグのデータ長（1 バイト）のバイト数「1」が後述の「論理積回数」に対応する。

【0 9 3 5】

まず、メインCPU 1 0 1 は、論理積元データ（例えば、図柄コード格納領域のデータ）を取得する（S 6 6 1）。次いで、メインCPU 1 0 1 は、論理積元データと論理積先データ（例えば、入賞作動フラグ格納領域のデータ）との論理積演算を行い、その演算結果を論理積先データとして保存する（S 6 6 2）。

【0 9 3 6】

次いで、メインCPU 1 0 1 は、取得する論理積元データのアドレスを 1 加算する（S 6 6 3）。次いで、メインCPU 1 0 1 は、参照する論理積先データのアドレスを 1 加算する（S 6 6 4）。

【0 9 3 7】

次いで、メインCPU 1 0 1 は、論理積回数を 1 減算する（S 6 6 5）。次いで、メインCPU 1 0 1 は、論理積回数が「0」であるか否かを判別する（S 6 6 6）。

【0 9 3 8】

S 6 6 6 において、メインCPU 1 0 1 が、論理積回数が「0」でないと判別したとき（S 6 6 6 が N O 判定の場合）、メインCPU 1 0 1 は、処理を S 6 6 1 の処理に戻し、S 6 6 1 以降の処理を繰り返す。一方、S 6 6 6 において、メインCPU 1 0 1 が、論理積回数が「0」であると判別したとき（S 6 6 6 が Y E S 判定の場合）、メインCPU 1 0 1 は、論理積演算処理を終了し、処理を例えば引込優先順位格納処理（図 1 2 6 参照）の S 6 2 7 に移す。

【0 9 3 9】

〔引込優先順位取得処理〕

次に、図 1 3 4～図 1 3 7 を参照して、引込優先順位格納処理（図 1 2 6 参照）中の S 6 2 7 で行う引込優先順位取得処理について説明する。なお、図 1 3 4 及び図 1 3 5 は、引込優先順位取得処理の手順を示すフローチャートである。図 1 3 6 A は、引込優先順位取得処理中の後述の S 6 8 0～S 6 8 3 の処理を実行するためのソースプログラムの一例を示す図であり、図 1 3 6 B は、引込優先順位取得処理中の後述の S 6 8 6 の処理を実行するためのソースプログラムの一例を示す図であり、図 1 3 6 C は、引込優先順位取得処理中の後述の S 6 8 7 の処理を実行するためのソースプログラムの一例を示す図である。また、図 1 3 7 は、引込優先順位取得処理のソースプログラム上で、実際に参照される引込優先順位テーブルの構成の一例を示す図である。

【0940】

まず、メインCPU101は、右リール3R（特定の表示列）のチェック時であるか否かを判別する（S671）。

【0941】

S671において、メインCPU101が、右リール3Rのチェック時でないと判別したとき（S671がNO判定の場合）、メインCPU101は、後述のS674の処理を行う。一方、S671において、メインCPU101が、右リール3Rのチェック時であると判別したとき（S671がYES判定の場合）、メインCPU101は、内部当籤役に係る図柄組合せ（入賞役）に「ANY役」（所定の図柄の組合せ）が含まれるか否かを判別する（S672）。なお、ここでいう「ANY役」とは、少なくとも右リール3Rの停止図柄に関係なく入賞が確定する役（少なくとも右リール3Rの停止図柄が任意の図柄である入賞役）のことをいう。

【0942】

S672において、メインCPU101が、内部当籤役に係る図柄組合せに「ANY役」が含まれないと判別したとき（S672がNO判定の場合）、メインCPU101は、後述のS674の処理を行う。一方、S672において、メインCPU101が、内部当籤役に係る図柄組合せに「ANY役」が含まれると判別したとき（S672がYES判定の場合）、メインCPU101は、入賞作動フラグ格納領域内の「ANY役」に対応する格納領域をマスクする（S673）。具体的には、メインCPU101は、入賞作動フラグ格納領域内の「ANY役」に対応するビットに「1」をセットする。

【0943】

S673の処理後、又は、S671或いはS672がNO判定の場合、メインCPU101は、入賞作動フラグ格納領域（図28～図30参照）のアドレスとして、その最後尾の格納領域のアドレスに「1」を加算したアドレスをセットし、停止禁止データをセットし、入賞作動フラグデータ長（入賞作動フラグ格納領域のデータ長：本実施形態では、12バイト）をセットする（S674）。次いで、メインCPU101は、ストップボタン作動カウンタの値、及び、ストップボタン作動状態を取得する（S675）。なお、ストップボタン作動カウンタは、停止操作が検出されているストップボタンの数を管理するためのカウンタである。また、ストップボタン作動状態は、作動ストップボタン格納領域（図33参照）を参照することにより取得される。

【0944】

次いで、メインCPU101は、セットされている入賞作動フラグ格納領域のアドレスを1減算（-1更新）する（S676）。次いで、メインCPU101は、セットされている入賞作動フラグ格納領域とそれに対応する当り要求フラグ格納領域（図28～図30参照）とから当り要求フラグデータを生成し、該生成された当り要求フラグデータに基づいて禁止入賞作動位置を生成する（S677）。

【0945】

次いで、メインCPU101は、停止操作位置が禁止入賞作動位置であるか否かを判別する（S678）。

【0946】

S678において、メインCPU101が、停止操作位置が禁止入賞作動位置でないと判

10

20

30

40

50

別したとき（S 6 7 8 が N O 判定の場合）、メイン C P U 1 0 1 は、後述の S 6 8 4 の処理を行う。一方、S 6 7 8 において、メイン C P U 1 0 1 が、停止操作位置が禁止入賞作動位置であると判別したとき（S 6 7 8 が Y E S 判定の場合）、メイン C P U 1 0 1 は、ストップボタン作動カウンタの値が第 3 停止の値であるか否かを判別する（S 6 7 9）。

【0 9 4 7】

S 6 7 9 において、メイン C P U 1 0 1 が、ストップボタン作動カウンタの値が第 3 停止の値であると判別したとき（S 6 7 9 が Y E S 判定の場合）、メイン C P U 1 0 1 は、後述の S 7 0 5 の処理を行う。一方、S 6 7 9 において、メイン C P U 1 0 1 が、ストップボタン作動カウンタの値が第 3 停止の値でないと判別したとき（S 6 7 9 が N O 判定の場合）、メイン C P U 1 0 1 は、ストップボタン作動カウンタの値が第 2 停止の値であるか否かを判別する（S 6 8 0）。

10

【0 9 4 8】

S 6 8 0 において、メイン C P U 1 0 1 が、ストップボタン作動カウンタの値が第 2 停止の値でないと判別したとき（S 6 8 0 が N O 判定の場合）、メイン C P U 1 0 1 は、後述の S 6 8 4 の処理を行う。一方、S 6 8 0 において、メイン C P U 1 0 1 が、ストップボタン作動カウンタの値が第 2 停止の値であると判別したとき（S 6 8 0 が Y E S 判定の場合）、メイン C P U 1 0 1 は、右リール 3 R の停止後であるか否かを判別する（S 6 8 1）。

【0 9 4 9】

S 6 8 1 において、メイン C P U 1 0 1 が、右リール 3 R の停止後であると判別したとき（S 6 8 1 が Y E S 判定の場合）、メイン C P U 1 0 1 は、後述の S 6 8 4 の処理を行う。一方、S 6 8 1 において、メイン C P U 1 0 1 が、右リール 3 R の停止後でないと判別したとき（S 6 8 1 が N O 判定の場合）、メイン C P U 1 0 1 は、当り要求フラグが「A N Y 役」の干渉を受ける可能性があるフラグでないか否か（内部当籤役に係る図柄組合せ（入賞役）に「A N Y 役」が含まれないか否か）を判別する（S 6 8 2）。

20

【0 9 5 0】

S 6 8 2 において、メイン C P U 1 0 1 が、当り要求フラグが「A N Y 役」の干渉を受ける可能性があるフラグでないと判別したとき（S 6 8 2 が Y E S 判定の場合）、メイン C P U 1 0 1 は、後述の S 6 8 4 の処理を行う。一方、S 6 8 2 において、メイン C P U 1 0 1 が、当り要求フラグが「A N Y 役」の干渉を受ける可能性があるフラグであると判別したとき（S 6 8 2 が N O 判定の場合）、メイン C P U 1 0 1 は、現チェックが「A N Y 役」を含む当り要求フラグのチェック時であるか否かを判別する（S 6 8 3）。

30

【0 9 5 1】

S 6 8 3 において、メイン C P U 1 0 1 が、現チェックが「A N Y 役」を含む当り要求フラグのチェック時であると判別したとき（S 6 8 3 が Y E S 判定の場合）、メイン C P U 1 0 1 は、後述の S 7 0 5 の処理を行う。

【0 9 5 2】

一方、S 6 8 3 において、メイン C P U 1 0 1 が、現チェックが「A N Y 役」を含む当り要求フラグのチェック時でないと判別したとき（S 6 8 3 が N O 判定の場合）、S 6 7 8 或いは S 6 8 0 が N O 判定の場合、又は、S 6 8 1 或いは S 6 8 2 が Y E S 判定の場合、メイン C P U 1 0 1 は、入賞作動フラグデータ長を 1 減算する（S 6 8 4）。次いで、メイン C P U 1 0 1 は、入賞作動フラグデータ長が「0」であるか否かを判別する（S 6 8 5）。

40

【0 9 5 3】

S 6 8 5 において、メイン C P U 1 0 1 が、入賞作動フラグデータ長が「0」でないと判別したとき（S 6 8 5 が N O 判定の場合）、メイン C P U 1 0 1 は、処理を S 6 7 6 の処理に戻し、S 6 7 6 以降の処理を繰り返す。

【0 9 5 4】

一方、S 6 8 5 において、メイン C P U 1 0 1 が、入賞作動フラグデータ長が「0」であると判別したとき（S 6 8 5 が Y E S 判定の場合）、メイン C P U 1 0 1 は、停止制御用

50

引込要求フラグ設定処理を行う（S 6 8 6）。この処理は、メインCPU101により、図136Bのソースプログラムで規定されている各処理が順次実行されることにより行われる。それゆえ、この処理の中では、図133で説明した論理積演算処理が行われる。なお、S 6 8 6の処理内で実行される論理積演算処理では、上述のように、当り（引込）要求フラグ格納領域のデータが「論理積先データ」にセットされ、入賞作動フラグ格納領域のデータが「論理積元データ」にセットされ、「論理積回数」には、RT作動組み合わせ表示フラグのデータ長（1バイト）のバイト数「1」がセットされる。RT作動組み合わせ表示フラグは、入賞作動フラグ格納領域において、RT移行に係る図柄組合せが規定された格納領域のことであり、本実施形態では、図28～図30に示すように格納領域11のみとなる。

10

【0955】

次いで、メインCPU101は、引込優先順位テーブルアドレス格納領域を参照して、引込優先順位テーブルを取得する（S 6 8 7）。この処理は、メインCPU101により、図136Cのソースプログラムで規定されている各処理が順次実行されることにより行われる。それゆえ、この処理では、現在セットされているアドレスに、引込優先順位データの初期値「1（001H）」が設定されるとともに、図137に示す、先頭アドレスが「dPLVLTB00」～「dPLVLTB05」のいずれかとなるブロックに格納された引込優先順位テーブルが取得される。なお、図137に示す、先頭アドレスが「dPLVLTB00」～「dPLVLTB05」となるブロックに格納された引込優先順位テーブルは、それぞれ、図27に記載の引込優先順位テーブル番号「00」～「05」の引込優先順位テーブルに対応する。

20

【0956】

次いで、メインCPU101は、現在セットされているアドレスに格納されている引込優先順位テーブルのデータが、エンドコード（000H）であるか否かを判別する（S 6 8 8）。

【0957】

S 6 8 8において、メインCPU101が、現在セットされているアドレスに格納されている引込優先順位テーブルのデータが、エンドコードであると判別したとき（S 6 8 8がYES判定の場合）、メインCPU101は、後述のS 7 0 5の処理を行う。一方、S 6 8 8において、メインCPU101が、現在セットされているアドレスに格納されている引込優先順位テーブルのデータが、エンドコードでないと判別したとき（S 6 8 8がNO判定の場合）、メインCPU101は、入賞作動フラグ格納領域をセットする（S 6 8 9）。

30

【0958】

次いで、メインCPU101は、現在セットされているアドレスに基づいて、引込優先順位テーブルから引込優先順位データを取得する（S 6 9 0）。次いで、メインCPU101は、引込優先順位テーブルのブロックカウンタをセットする（S 6 9 1）。本実施形態では、この処理において、メインCPU101は、引込優先順位テーブルのブロックカウンタの値に「2」をセットする。

【0959】

次いで、メインCPU101は、引込優先順位テーブルのチェック回数をセットし、参照する引込優先順位テーブルのアドレスを1加算（+1更新）する（S 6 9 2）。本実施形態では、この処理において、メインCPU101は、引込優先順位テーブルのチェック回数に「8」（図137に示す引込優先順位テーブルに規定されているチェックデータのビット数）をセットする。

40

【0960】

次いで、メインCPU101は、更新された引込優先順位テーブルのアドレスに基づいて、チェックデータ（図137参照）を取得し、チェックデータからチェックビットを抽出する（S 6 9 3）。なお、本実施形態では、ここで抽出するチェックビットは、チェックデータのビット0に対応する。例えば、S 6 9 0の処理において、先頭アドレスが「dP

50

「L V L T B 0 0」であるブロックに規定された引込優先順位テーブルから引込優先順位データ「0 3 E H」が取得された場合、S 6 9 3の処理では、チェックデータとして「1 0 0 0 1 0 0 0 B」が取得され、チェックビットの値として「0」が抽出される。

【0 9 6 1】

次いで、メインCPU 1 0 1は、抽出されたチェックビットの値が「1」であるか否かを判別する（S 6 9 4）。

【0 9 6 2】

S 6 9 4において、メインCPU 1 0 1が、抽出されたチェックビットの値が「1」でないと判別したとき（S 6 9 4がNO判定の場合）、メインCPU 1 0 1は、後述のS 6 9 9の処理を行う。一方、S 6 9 4において、メインCPU 1 0 1が、抽出されたチェックビットの値が「1」であると判別したとき（S 6 9 4がYES判定の場合）、メインCPU 1 0 1は、参照する引込優先順位テーブルのアドレスを1加算（+1更新）し、更新後のアドレスに基づいて、引込優先順位テーブルから判定データ（図1 3 7中の「フラグ判定データ」）を取得する（S 6 9 5）。

【0 9 6 3】

次いで、メインCPU 1 0 1は、S 6 9 5で取得した判定データに基づいて、現在取得されている入賞作動フラグデータが判定対象であるか否かを判別する（S 6 9 6）。この処理では、メインCPU 1 0 1は、現在取得されている入賞作動フラグデータと、判定データとを比較し、前者が後者に対応するものである否かを判定し、前者が後者に対応するものである場合には、現在取得されている入賞作動フラグデータが判定対象であると判定する。

【0 9 6 4】

S 6 9 6において、メインCPU 1 0 1が、入賞作動フラグデータが判定対象でないと判別したとき（S 6 9 6がNO判定の場合）、メインCPU 1 0 1は、後述のS 6 9 9の処理を行う。一方、S 6 9 6において、メインCPU 1 0 1が、入賞作動フラグデータが判定対象であると判別したとき（S 6 9 6がYES判定の場合）、メインCPU 1 0 1は、引込優先順位データの更新処理を行う（S 6 9 7）。この処理では、メインCPU 1 0 1は、S 6 9 7で取得した判定データに対応付けられた引込優先順位データで、現在セットされている引込優先順位データを更新（上書き）する。

【0 9 6 5】

次いで、メインCPU 1 0 1は、チェックデータの更新処理を行う（S 6 9 8）。この処理では、メインCPU 1 0 1は、チェックデータを1ビットだけ右方向（ビット7からビット0に向かう方向）にシフトする。なお、この処理において、シフト後のチェックデータのビット7には、「0」がセットされる。

【0 9 6 6】

S 6 9 8の処理後、又は、S 6 9 4或いはS 6 9 6がNO判定の場合、メインCPU 1 0 1は、チェックデータにチェック対象のビット（「1」がセットされているビット）があるか否かを判別する（S 6 9 9）。

【0 9 6 7】

S 6 9 9において、メインCPU 1 0 1が、チェックデータにチェック対象のビットがないと判別したとき（S 6 9 9がNO判定の場合）、メインCPU 1 0 1は、後述のS 7 0 2の処理を行う。一方、S 6 9 9において、メインCPU 1 0 1が、チェックデータにチェック対象のビットがあると判別したとき（S 6 9 9がYES判定の場合）、メインCPU 1 0 1は、チェックする入賞作動フラグ格納領域のアドレスを1加算（+1更新）し、チェック回数を1減算する（S 7 0 0）。

【0 9 6 8】

次いで、メインCPU 1 0 1は、チェック回数が「0」であるか否かを判別する（S 7 0 1）。S 7 0 1において、メインCPU 1 0 1が、チェック回数が「0」でないと判別したとき（S 7 0 1がNO判定の場合）、メインCPU 1 0 1は、処理をS 6 9 8の処理に戻し、S 6 9 8以降の処理を繰り返す。

10

20

30

40

50

【0969】

一方、S701において、メインCPU101が、チェック回数が「0」とであると判別したとき（S701がYES判定の場合）、メインCPU101は、現在参照している入賞作動フラグ格納領域のアドレスにチェック回数の初期値「8」を加算して入賞作動フラグ格納領域のアドレスを更新し、ブロックカウンタの値を1減算する（S702）。次いで、メインCPU101は、ブロックカウンタの値が「0」とであるか否かを判別する（S703）。

【0970】

S703において、メインCPU101が、ブロックカウンタの値が「0」とでないと判別したとき（S703がNO判定の場合）、メインCPU101は、処理をS692の処理に戻し、S692以降の処理を繰り返す。

10

【0971】

一方、S703において、メインCPU101が、ブロックカウンタの値が「0」とであると判別したとき（S703がYES判定の場合）、メインCPU101は、参照する引込優先順位テーブルのアドレスを1加算（+1更新）する（S704）。例えば、現在参照している引込優先順位テーブルが、先頭アドレスが「dPLVLTB00」とであるブロックに規定された引込優先順位テーブルである場合、この処理により、参照する引込優先順位テーブルが、先頭アドレスが「dPLVLTB01」とであるブロックに規定された引込優先順位テーブルに変更される。そして、S704の処理後、メインCPU101は、処理をS688の処理に戻し、S688以降の処理を繰り返す。

20

【0972】

ここで再度、S679、S683又はS688の処理に戻って、S679、S683又はS688がYES判定の場合、メインCPU101は、この時点でセットされている引込順位データを、最終的な引込優先順位データとしてセットする（S705）。なお、S679又はS683がYES判定の場合、メインCPU101は、最終的な引込優先順位データとして「0（00H）」をセットする。この場合、引込優先順位データ「0（00H）」にはエンドコードが割り付けられているので、引込データ無し（停止禁止）がセットされる。そして、S705の処理後、メインCPU101は、引込優先順位取得処理を終了し、処理を引込優先順位格納処理（図126参照）のS628に移す。

【0973】

本実施形態では、上述のようにして引込優先順位取得処理が行われる。なお、上述した引込優先順位取得処理中のS680～S683の「ANY役」の引込優先対応処理は、メインCPU101が、図136Aのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中で、例えば、S683の判定処理は、ソースプログラム上において、「JCP」命令（所定の判定命令）により実行される。なお、「JCP」命令は、比較命令相当の動作を実行する命令であり、メインCPU101専用命令コードである。

30

【0974】

ソースプログラム上において、例えば、ソースコード「JCP cc, A, n, e」が実行されると、Aレジスタの内容（格納データ）と、整数nとを比較し、その比較結果が、ccの条件となれば、処理をeで指定されるアドレスにジャンプさせる。なお、「JCP」命令の「ccの条件」には、フラグ・レジスタF内のキャリーフラグの状態及びゼロフラグの状態の一方が指定される（図11参照）。例えば、ccに「C」が指定されていれば、ccの条件はキャリーフラグが「1」（オン状態）であることを意味し、ccに「NC」が指定されていれば、ccの条件はキャリーフラグが「0」（オフ状態）であることを意味する。また、例えば、ccに「Z」が指定されていれば、ccの条件はゼロフラグが「1」（オン状態）であることを意味し、ccに「NZ」が指定されていれば、ccの条件はゼロフラグが「0」（オフ状態）であることを意味する。

40

【0975】

「ANY役」の引込優先対応処理のソースプログラム上において、図136Aに示すよう

50

に、「JCP」命令を用いた場合、アドレス設定に係る命令を省略することができる（アドレス設定に係る命令を別途設ける必要がなくなる）ので、「ANY役」の引込優先対応処理の処理効率を高めることができるとともに、ソースプログラムの容量（メインROM 102の使用容量）を低減することができる。

【0976】

また、上述した引込優先順位取得処理中のS686の停止制御用引込要求フラグ設定処理は、メインCPU101が、図136Bのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。S686の停止制御用引込要求フラグ設定処理では、図136Bに示すように、メインCPU101専用命令コードである、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う「LDQ」命令、及び、「CALLF」命令が利用される。

10

【0977】

それゆえ、S686の停止制御用引込要求フラグ設定処理において、Qレジスタ（拡張レジスタ）を用いた「LDQ」命令を用いることにより、直値により、メインROM102、メインRAM103やメモリーマップI/Oにアクセスすることができる。この場合、ソースプログラム上において、アドレス設定に係る命令を省略することができ、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。また、「CALLF」命令は、上述のように、2バイトの命令コードである。それゆえ、停止制御用引込要求フラグ設定処理において、これらのメインCPU101専用命令コードを使用することにより、処理の効率化を図ることができ、限られたメインRAM103の容量を有効活用することができる。

20

【0978】

さらに、本実施形態では、優先引込順位取得処理中のS686の停止制御用引込要求フラグ設定処理において、「CALLF」命令で指定するジャンプ先の論理積演算処理のアドレス「SB__DAND__00」は、上記図126で説明した引込優先順位格納処理中のS626の論理積演算処理において「CALLF」命令で指定するジャンプ先のアドレスと同じである（図127参照）。すなわち、本実施形態では、優先引込順位取得処理中のS686の停止制御用引込要求フラグ設定処理で行う論理積演算処理を実行するためのソースプログラムが、引込優先順位格納処理中のS626で行う論理積演算処理を実行するためのソースプログラムと同じであり、S686及びS626の両処理において、論理積演算処理のソースプログラムが共有化（モジュール化）されている。この場合、S686及びS626の両処理において、それぞれ別個に論理積演算処理のソースプログラムを設ける必要がなくなるので、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

30

【0979】

また、上述した引込優先順位取得処理中のS687の引込優先順位テーブル（図137参照）の取得処理は、メインCPU101が、図136Cのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。そして、S687の引込優先順位テーブルの取得処理では、図136Cに示すように、メインCPU101専用命令コードである、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う「LDQ」命令が利用される。

40

【0980】

それゆえ、S687の引込優先順位テーブルの取得処理においても、「LDQ」命令の使用により、ソースプログラム上において、アドレス設定に係る命令を省略することができる。その結果、引込優先順位テーブルの取得処理の効率化を図ることができるとともに、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。

【0981】

上述のように、本実施形態の優先引込順位取得処理中の上記各種処理では、上述したメイ

50

ンCPU101専用の各種命令コードが適宜用いられ、対応する処理の効率化及びソースプログラムの容量の削減を実現している。その結果、本実施形態では、主制御回路90のプログラム処理速度の効率化と容量の削減とを図ることができ、削減した容量に対応する空き領域を活用して、遊技性を高めることが可能となる。

【0982】

〔リール停止制御処理〕

次に、図138～図140を参照して、メインフロー（図82参照）中のS213で行うリール停止制御処理について説明する。なお、図138は、リール停止制御処理の手順を示すフローチャートである。図139は、リール停止制御処理中の後述のS711～S716の処理を実行するためのソースプログラムの一例を示す図であり、図140は、リール停止制御処理中の後述のS726の処理を実行するためのソースプログラムの一例を示す図である。

10

【0983】

まず、メインCPU101は、リール停止可能信号OFF処理を行う（S711）。この処理では、メインCPU101は、主に、リール停止可能信号OFFデータのポート出力処理を行う。また、この処理は、メインRAM103の規定外作業領域を使用して行われる。なお、リール停止可能信号OFF処理の詳細については、後述の図141を参照しながら後で説明する。

【0984】

次いで、メインCPU101は、全リールの回転速度が所定の一定速度に到達したか否か（「定速」になったか否か）を判別する（S712）。S712において、メインCPU101が、全リールの回転速度が「定速」になっていないと判別したとき（S712がNO判定の場合）、メインCPU101は、S712の処理を繰り返す。

20

【0985】

一方、S712において、メインCPU101が、全リールの回転速度が「定速」になったと判別したとき（S712がYES判定の場合）、メインCPU101は、リール停止可能信号ON処理を行う（S713）。この処理では、メインCPU101は、主に、リール停止可能信号ONデータのポート出力処理を行う。また、この処理は、メインRAM103の規定外作業領域を使用して行われる。なお、リール停止可能信号ON処理の詳細については、後述の図142を参照しながら後で説明する。

30

【0986】

次いで、メインCPU101は、有効なストップボタンが押されたか否かを判別する（S714）。

【0987】

S714において、メインCPU101が、有効なストップボタンが押されていないと判別したとき（S714がNO判定の場合）、メインCPU101は、処理をS713の処理に戻し、S713以降の処理を繰り返す。一方、S714において、メインCPU101が、有効なストップボタンが押されたと判別したとき（S714がYES判定の場合）、メインCPU101は、作動ストップボタン格納領域（図33参照）を更新し、ストップボタン未作動カウンタの値を1減算する（S715）。

40

【0988】

次いで、メインCPU101は、作動ストップボタンから検索対象リールを決定する（S716）。また、この処理では、検索対象リールのリール制御管理情報が格納される回胴制御データ格納領域のアドレス（先頭アドレス）セット処理も行われる（図139中のソースコード「LDQ IX, WR1__CTRL - (WR2__CTRL - WR1__CTRL)」参照）。

【0989】

次いで、メインCPU101は、リール停止可能信号OFF処理を行う（S717）。この処理は、上記S711と同様に、メインRAM103の規定外作業領域を使用して行われる。なお、リール停止可能信号OFF処理の詳細については、後述の図141を参照し

50

ながら後で説明する。次いで、メインCPU101は、図柄カウンタの値に基づいて停止開始位置をメインRAM103に格納する（S718）。

【0990】

次いで、メインCPU101は、リール停止選択処理を行う（S719）。詳細な説明は省略するが、この処理では、メインCPU101は、滑り駒数の選択処理を行う。

【0991】

次いで、メインCPU101は、停止開始位置と、S719で決定された滑り駒数とに基づいて停止予定位置を決定し、該決定した停止予定位置をメインRAM103に格納する（S720）。この処理では、メインCPU101は、停止開始位置に滑り駒数を加算し、その加算結果を停止予定位置とする。

10

【0992】

次いで、メインCPU101は、図柄コード格納処理を実行する（S721）。この処理では、停止予定位置に対応する図柄コードが図柄コード格納領域に格納される。次いで、メインCPU101は、制御対象のリールが最終停止（第3停止）のリールであるか否かを判別する（S722）。この処理では、メインCPU101は、ストップボタン未作動カウンタの値に基づいて、制御対象のリールが最終停止（第3停止）のリールであるか否かを判別し、ストップボタン未作動カウンタの値が「0」であるときには、制御対象のリールが最終停止のリールであると判定する。

【0993】

S722において、メインCPU101が、制御対象のリールが最終停止のリールでないと判別したとき（S722がNO判定の場合）、メインCPU101は、制御変更処理を行う（S723）。この処理では、特定の停止位置にあった場合に、リールの停止に用いる停止情報群が更新される。次いで、メインCPU101は、図126で説明した引込優先順位格納処理を行う（S724）。

20

【0994】

次いで、メインCPU101は、停止間隔残時間待機処理を行う（S725）。この処理では、メインCPU101は、予め設定された所定のリール停止間隔時間が経過するまで、待機処理を行う。そして、S725の処理後、メインCPU101は、処理をS711の処理に戻し、S711以降の処理を繰り返す。

【0995】

ここで再度、S722の処理に戻って、S722において、メインCPU101が、制御対象のリールが最終停止のリールであると判別したとき（S722がYES判定の場合）、メインCPU101は、全リールの励磁が停止状態であるか否かを判別する（S726）。S726において、メインCPU101が、全リールの励磁が停止状態でないと判別したとき（S726がNO判定の場合）、メインCPU101は、S726の処理を繰り返す。

30

【0996】

一方、S726において、メインCPU101が、全リールの励磁が停止状態であると判別したとき（S726がYES判定の場合）、メインCPU101は、第3停止操作されたストップボタンがオン状態のままである（ストップボタンが放されていない）か否かを判別する（S727）。S727において、メインCPU101が、第3停止操作されたストップボタンがオン状態のままであると判別したとき（S727がYES判定の場合）、メインCPU101は、S727の処理を繰り返す。一方、S727において、メインCPU101が、第3停止操作されたストップボタンがオン状態でないと判別したとき（S727がNO判定の場合）、メインCPU101は、リール停止制御処理を終了し、処理をメインフロー（図82参照）のS214に移す。

40

【0997】

本実施形態では、上述のようにしてリール停止制御処理が行われる。なお、上述したリール停止制御処理中のS711～S716の処理は、メインCPU101が、図139のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。図

50

139に示すように、本実施形態のリール停止制御処理のソースプログラムでは、メインCPU101専用命令コードである、例えば、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う「LDQ」命令や、「CALLF」命令が用いられる。

【0998】

それゆえ、リール停止制御処理において、このようなメインCPU101専用命令コードを用いることにより、リール制御処理のソースプログラムの容量を削減することができる。すなわち、本実施形態では、主制御回路90におけるプログラム処理速度の効率化と容量の削減とを行うことが可能となり、削減した容量に応じて増加したメインROM102の空き領域を活用して、遊技性を高めることが可能となる。

10

【0999】

また、上述したリール停止制御処理中のS726の判定処理は、メインCPU101が、図140のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。この処理は、図140に示すように、ソースプログラム上では、「LDQ」命令、「ORQ」命令（所定の論理和演算命令）を用いて実行される。

【1000】

なお、「ORQ」命令は、論理和演算を行う命令コードであり、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行うメインCPU101専用命令コードである。そして、ソースプログラム上において、例えば、ソースコード「ORQ (k)」が実行されると、Qレジスタの格納データ（上位側アドレス値）及び1バイトの整数k（直値：下位側アドレス値）で指定されたアドレスのメモリの内容（格納データ）と、Aレジスタの内容（格納データ）との論理和演算が行われ、その演算結果がAレジスタに記憶される。

20

【1001】

それゆえ、リール停止制御処理中のS726の判定処理において、まず、図140中のソースコード「LDQ A, (.LOW.wr1__TIM)」が実行されると、Qレジスタの格納データと、整数値「.LOW.wr1__TIM」とで指定されるアドレスのメモリの内容（第1リールの励磁タイマー値）がAレジスタにロードされる。なお、本実施形態では、メインRAM103内における第1リールの励磁タイマー値が格納された領域のアドレスは、「F032h」である。そして、上述したS726の判定処理では、LDQ命令実行時に予めQレジスタに、アドレス「wr1__TIM (F032h)」の上位側アドレス値「F0h」がセットされ、kの値（直値）には、下位側アドレス値（「.LOW.wr2__TIM」=32h）が代入される。

30

【1002】

次いで、図140中のソースコード「ORQ (.LOW.wr2__TIM)」が実行されると、Qレジスタの格納データ（F0h）と、第2リールの励磁タイマー値が格納された領域のアドレス「wr2__TIM (F03Dh)」の下位側アドレス値（3Dh）で指定されたアドレスのメモリの内容（第2リールの励磁タイマー値）と、Aレジスタの内容（第1リールの励磁タイマー値）との論理和演算が行われ、その演算結果（第1リールの励磁タイマー値と第2リールの励磁タイマー値との合成結果）がAレジスタに記憶される。次いで、図140中のソースコード「ORQ (.LOW.wr3__TIM)」が実行されると、Qレジスタの格納データ（F0h）と、第3リールの励磁タイマー値が格納された領域のアドレス「wr3__TIM (F048h)」の下位側アドレス値（48h）で指定されたアドレスのメモリの内容（第3リールの励磁タイマー値）と、Aレジスタの内容（第1リールの励磁タイマー値と第2リールの励磁タイマー値との合成結果）との論理和演算が行われ、その演算結果（第1～第3リールの励磁タイマー値の合成結果）がAレジスタに記憶される。

40

【1003】

上述のように、本実施形態では、リール（回胴）の停止状態のチェック処理において、Qレジスタ（拡張レジスタ）を用いた各種メインCPU101専用命令コードが用いられる。それゆえ、これらのメインCPU101専用命令コードを用いることにより、直値によ

50

り、メインROM102、メインRAM103やメモリーマップI/Oにアクセスすることができ、ソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【1004】

上述のように、本実施形態のリール停止制御処理中の上記各種処理では、上述したメインCPU101専用の各種命令コードが適宜用いられ、対応する処理の効率化及びソースプログラムの容量の削減を実現している。その結果、本実施形態では、主制御回路90のプログラム処理速度の効率化と容量の削減とを図ることができ、削減した容量に対応する空き領域を活用して、遊技性を高めることが可能となる。

【1005】

〔リール停止可能信号OFF処理〕

次に、図141を参照して、リール停止制御処理（図138参照）中のS711又はS717で行うリール停止可能信号OFF処理について説明する。なお、図141は、リール停止可能信号OFF処理の手順を示すフローチャートである。

【1006】

まず、メインCPU101は、スタックポインタ（SP）にセットされているメインRAM103のスタックエリア（図12C参照）のアドレスを退避させる（S731）。次いで、メインCPU101は、スタックポインタ（SP）に規定外スタックエリアのアドレスをセットする（S732）。

【1007】

次いで、メインCPU101は、全てのレジスタにセットされているデータを退避させる（S733）。次いで、メインCPU101は、リール停止可能信号OFFデータのセット処理を行う（S734）。

【1008】

次いで、メインCPU101は、規定外ポート出力処理を行う（S735）。この処理では、メインCPU101は、リール停止可能信号OFFデータに基づいて、後述のOFF出力データ（出力オフモードデータ）の生成及び出力処理を行う。なお、この処理は、メインRAM103の規定外作業領域を使用して行われる。規定外ポート出力処理の詳細については、後述の図143を参照しながら後で説明する。

【1009】

次いで、メインCPU101は、S733で退避させた全レジスタのデータを復帰させる（S736）。次いで、メインCPU101は、S731で退避させたスタックエリアのアドレスをスタックポインタ（SP）にセットする（S737）。

【1010】

そして、S737の処理後、メインCPU101は、リール停止可能信号OFF処理を終了する。この際、実行したリール停止可能信号OFF処理がリール停止制御処理（図138参照）中のS711の処理である場合には、メインCPU101は、処理をリール停止制御処理中のS712の処理に移す。一方、実行したリール停止可能信号OFF処理がリール停止制御処理（図138参照）中のS717の処理である場合には、メインCPU101は、処理をリール停止制御処理中のS718の処理に移す。

【1011】

〔リール停止可能信号ON処理〕

次に、図142を参照して、リール停止制御処理（図138参照）中のS713で行うリール停止可能信号ON処理について説明する。なお、図142は、リール停止可能信号ON処理の手順を示すフローチャートである。

【1012】

まず、メインCPU101は、スタックポインタ（SP）にセットされているメインRAM

10

20

30

40

50

M103のスタックエリア（図12C参照）のアドレスを退避させる（S741）。次いで、メインCPU101は、スタックポインタ（SP）に規定外スタックエリアのアドレスをセットする（S742）。

【1013】

次いで、メインCPU101は、全てのレジスタにセットされているデータを退避させる（S743）。次いで、メインCPU101は、作動ストップボタン格納領域（図33参照）を参照し、ストップボタン状態を取得する（S744）。次いで、メインCPU101は、リール停止可能信号ONデータのセット処理を行う（S745）。

【1014】

次いで、メインCPU101は、規定外ポート出力処理を行う（S746）。この処理では、メインCPU101は、リール停止可能信号ONデータに基づいて、後述のON出力データ（出力オンモードデータ）の生成及び出力処理を行う。なお、この処理は、メインRAM103の規定外作業領域を使用して行われる。規定外ポート出力処理の詳細については、後述の図143を参照しながら後で説明する。

【1015】

次いで、メインCPU101は、S743で退避させた全レジスタのデータを復帰させる（S747）。次いで、メインCPU101は、S741で退避させたスタックエリアのアドレスをスタックポインタ（SP）にセットする（S748）。そして、S748の処理後、メインCPU101は、リール停止可能信号ON処理を終了し、処理をリール停止制御処理（図138参照）中のS714の処理に移す。

【1016】

〔規定外ポート出力処理〕

次に、図143及び図144を参照して、リール停止可能信号OFF処理（図141参照）中のS735及びリール停止可能信号ON処理（図142参照）中のS746で行う規定外ポート出力処理について説明する。なお、図143は、規定外ポート出力処理の手順を示すフローチャートである。また、図144は、規定外ポート出力処理を実行するためのソースプログラムの一例を示す図である。

【1017】

まず、メインCPU101は、ポート出力設定がON出力モードであるか否かを判別する（S751）。この処理において、メインCPU101は、リール停止可能信号ONデータがセットされている場合には、ポート出力設定がON出力モードであると判定し、リール停止可能信号OFFデータがセットされている場合には、ポート出力設定がON出力モードでないと判定する。

【1018】

S751において、メインCPU101が、ポート出力設定がON出力モードであると判別したとき（S751がYES判定の場合）、メインCPU101は、後述のS753の処理を行う。一方、S751において、メインCPU101が、ポート出力設定がON出力モードでないと判別したとき（S751がNO判定の場合）、メインCPU101は、OFF出力データ（出力オフモードデータ）の生成処理を行う（S752）。この処理では、現在、出力オン状態となっているポート（ビット）のうち、オフ状態にしたいポート（ビット）をオフ状態にするとともに、現在、出力オフ状態となっているポート（ビット）をオフ状態に維持するためのOFF出力データが生成される。

【1019】

S752の処理後又はS751がNO判定の場合、メインCPU101は、ON出力データ（出力オンモードデータ）の生成処理を行う（S753）。この処理では、現在、出力オフ状態となっているポート（ビット）のうち、オン状態にしたいポート（ビット）をオン状態にするとともに、現在、出力オン状態となっているポート（ビット）をオン状態に維持するためのON出力データが生成される。次いで、メインCPU101は、生成された出力データを指定ポートから出力する（S754）。

【1020】

10

20

30

40

50

そして、S 7 5 4 の処理後、メインCPU 1 0 1 は、規定外ポート出力処理を終了する。この際、実行した規定外ポート出力処理がリール停止可能信号OFF 処理（図 1 4 1 参照）中のS 7 3 5 の処理である場合には、メインCPU 1 0 1 は、処理をリール停止可能信号OFF 処理中のS 7 3 6 の処理に移す。一方、実行した規定外ポート出力処理がリール停止可能信号ON 処理（図 1 4 2 参照）中のS 7 4 6 の処理である場合には、メインCPU 1 0 1 は、処理をリール停止可能信号ON 処理中のS 7 4 7 の処理に移す。

【1 0 2 1】

本実施形態では、上述のようにして規定外ポート出力処理が行われる。そして、上述した規定外ポート出力処理は、メインCPU 1 0 1 が、図 1 4 4 のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【1 0 2 2】

その中で、上述した規定外ポート出力処理中のS 7 5 2 のOFF 出力データの生成処理は、図 1 4 4 中のソースコード「XOR （HL）」及び「AND （HL）」をこの順で実行することに行われる。また、上述した規定外ポート出力処理中のS 7 5 3 のON 出力データの生成処理は、図 1 4 4 中のソースコード「OR （HL）」を実行することに行われる。

【1 0 2 3】

ソースプログラム上において、このような各出力データの生成処理を行うことにより、S 7 5 2 のOFF 出力データの生成処理後に、S 7 5 3 のON 出力データの生成処理が行われてもS 7 5 2 で生成されたOFF 出力データは変化しない。

【1 0 2 4】

例えば、ポート出力設定がOFF 出力モードであり、今回の処理でオフ状態にしたい規定外ポートを示す出力データが「0 0 0 1 0 1 1 1」（「1」がオフ状態にしたいビット）であり、現在、規定外ポートに出力されている出力データ（バックアップデータ）が「0 1 0 1 0 0 1 1」（「1」が現在、オン状態のビット）である場合、バックアップデータのビット0、ビット1及びビット5のデータを「1」から「0」にするためのOFF 出力データが生成される。この場合、まず、図 1 4 4 中のソースコード「XOR （HL）」が実行されると、出力データ「0 0 0 1 0 1 1 1」と、バックアップデータ「0 1 0 1 0 0 1 1」との排他的論理和演算が行われ、演算結果として「0 1 0 0 0 1 0 0」が得られる。次いで、図 1 4 4 中のソースコード「AND （HL）」が実行されると、演算結果「0 1 0 0 0 1 0 0」とバックアップデータ「0 1 0 1 0 0 1 1」との論理積演算が行われ、演算結果「0 1 0 0 0 0 0 0」がOFF 出力データとして生成される。

【1 0 2 5】

その後、S 7 5 3 のON 出力データの生成処理が（図 1 4 4 中のソースコード「OR （HL）」）が実行されると、演算結果「0 1 0 0 0 0 0 0」（OFF 出力データ）と、今回の処理でオン状態にしたい規定外ポートを示す出力データ「0 0 0 0 0 0 0 0」（ポート出力設定がOFF 出力モードであるので、出力データの各ビットには「0」がセットされる）との論理和演算が行われ、演算結果として「0 1 0 0 0 0 0 0」が得られ、OFF 出力データは変化しない。定性的には、ポート出力設定がOFF 出力モードである場合、S 7 5 3 のON 出力データの生成処理では、OFF 出力データにおいて出力オン状態となっているポート（ビット）をオン状態に維持するための出力データが生成されるので、S 7 5 2 のOFF 出力データの生成処理後に、S 7 5 3 のON 出力データの生成処理が行われてもS 7 5 2 で生成されたOFF 出力データは変化しない。

【1 0 2 6】

〔入賞検索処理〕

次に、図 1 4 5 ～図 1 4 7 を参照して、メインフロー（図 8 2 参照）中のS 2 1 4 で行う入賞検索処理について説明する。なお、図 1 4 5 は、入賞検索処理の手順を示すフローチャートである。図 1 4 6 は、入賞検索処理を実行するためのソースプログラムの一例を示す図である。また、図 1 4 7 は、入賞検索処理のソースプログラム上で、実際に参照される払出枚数データテーブルの構成の一例を示す図である。

10

20

30

40

50

【1027】

まず、メインCPU101は、図柄コード格納領域（図35参照）に格納された各格納領域のデータを、入賞作動フラグ格納領域（図28～図30参照）の対応する格納領域に転送して保存する（S761）。そして、この処理終了時点では、DEレジスタに入賞作動フラグ格納領域の最後尾のアドレスがセットされる。

【1028】

次いで、メインCPU101は、払出枚数データテーブルのアドレス（図147に示す払出枚数データテーブルの先頭アドレス「dPAYNUMTB」）をHLレジスタにセットする（S762）。次いで、メインCPU101は、払出枚数テーブル数（本実施形態では「5」）を入賞検索カウンタの初期値とし、該初期値をBレジスタにセットする（S763）。

10

【1029】

次いで、メインCPU101は、HLレジスタにセットされたアドレスに基づいて、メダルの払出枚数（本実施形態では、1枚、2枚、3枚及び9枚のいずれか）のデータをCレジスタにセットし、判定対象データをAレジスタにセットし、HLレジスタにセットされているアドレスに「2」を加算（+2更新）する（S764）。なお、図147に示す払出枚数データテーブルにおいて、メダルの払出枚数のデータは、「払出枚数（1, 2, 3又は9）×2+0」であり、判定対象データは、払出枚数のデータの次のアドレスに格納されている1バイトのデータ（例えば「11111000B」等）である。また、以下では、Cレジスタにセットされたメダルの払出枚数のデータ「払出枚数（1, 2, 3又は9）×2+0」内のデータ「0」を「判定ビット」という。この判定ビットは入賞検索の判定対象ブロックであるか否かを示す情報である。

20

【1030】

次いで、メインCPU101は、Cレジスタにセットされたメダルの払出枚数のデータから判定ビットの値を抽出する（S765）。次いで、メインCPU101は、抽出した判定ビットの値に基づいて、判定対象ブロックであるか否かを判別する（S766）。この処理において、メインCPU101は、抽出した判定ビットの値が「1」である場合に、判定対象ブロックであると判定する。なお、本実施形態では、図147に示すように、メダルの払出枚数に関係なく、判定ビットの値は、常に「0」であるので、S766の処理は必ずNO判定となる。

30

【1031】

S766において、メインCPU101が、判定対象ブロックでないと判別したとき（S766がNO判定の場合）、メインCPU101は、後述のS768の処理を行う。一方、S766において、メインCPU101が、判定対象ブロックであると判別したとき（S766がYES判定の場合）、メインCPU101は、DEレジスタにセットされている入賞作動フラグ格納領域のアドレスを1減算（-1更新）する（S767）。

【1032】

S767の処理後又はS766がNO判定の場合、メインCPU101は、DEレジスタにセットされた入賞作動フラグ格納領域のアドレスで指定される格納領域のデータを判定データとして抽出する（S768）。

40

【1033】

次いで、メインCPU101は、S764でAレジスタにセットされた判定対象データと、S768で抽出した判定データとに基づいて、判定の結果が入賞であるか否かを判別する（S769）。この処理において、メインCPU101は、S764でAレジスタにセットされた判定対象データが、S768で抽出した判定データと同じであれば、判定の結果が入賞であると判定する。

【1034】

S769において、メインCPU101が、判定の結果が入賞でないと判別したとき（S769がNO判定の場合）、メインCPU101は、後述のS776の処理を行う。一方、S769において、メインCPU101が、判定の結果が入賞であると判別したとき（

50

S 7 6 9 が Y E S 判定の場合)、メインCPU 1 0 1 は、現遊技が 3 枚遊技 (メダルのベット枚数が 3 枚である遊技) であるか否かを判別する (S 7 7 0)。

【1 0 3 5】

S 7 7 0 において、メインCPU 1 0 1 が、現遊技が 3 枚遊技であると判別したとき (S 7 7 0 が Y E S 判定の場合)、メインCPU 1 0 1 は、後述の S 7 7 2 の処理を行う。一方、S 7 7 0 において、メインCPU 1 0 1 が、現遊技が 3 枚遊技でないと判別したとき (S 7 7 0 が N O 判定の場合)、メインCPU 1 0 1 は、2 枚遊技 (メダルのベット枚数が 2 枚である遊技) の払出枚数 (2 枚) を C レジスタにセットする (S 7 7 1)。

【1 0 3 6】

S 7 7 1 の処理後又は S 7 7 0 が Y E S 判定の場合、メインCPU 1 0 1 は、払出枚数の更新処理を行う (S 7 7 2)。具体的には、メインCPU 1 0 1 は、現在の入賞枚数カウンタの値に、C レジスタにセットされたメダルの払出枚数を加算し、加算後の値を払出枚数にセットする。

10

【1 0 3 7】

次いで、メインCPU 1 0 1 は、払出枚数の値が最大払出枚数「1 0」未満であるか否かを判別する (S 7 7 3)。

【1 0 3 8】

S 7 7 3 において、メインCPU 1 0 1 が、払出枚数の値が最大払出枚数「1 0」未満であると判別したとき (S 7 7 3 が Y E S 判定の場合)、メインCPU 1 0 1 は、後述の S 7 7 5 の処理を行う。一方、S 7 7 3 において、メインCPU 1 0 1 が、払出枚数の値が最大払出枚数「1 0」未満でないと判別したとき (S 7 7 3 が N O 判定の場合)、メインCPU 1 0 1 は、払出枚数に最大払出枚数「1 0」をセットする (S 7 7 4)。

20

【1 0 3 9】

S 7 7 4 の処理後又は S 7 7 3 が Y E S 判定の場合、メインCPU 1 0 1 は、払出枚数を入賞枚数カウンタに保存する (S 7 7 5)。

【1 0 4 0】

S 7 7 5 の処理後又は S 7 6 9 が N O 判定の場合、メインCPU 1 0 1 は、他の入賞があるか否かを判別する (S 7 7 6)。S 7 7 6 において、メインCPU 1 0 1 が、他の入賞があると判別したとき (S 7 7 6 が Y E S 判定の場合)、メインCPU 1 0 1 は、処理を S 7 6 9 の処理に戻し、S 7 6 9 以降の処理を繰り返す。

30

【1 0 4 1】

一方、S 7 7 6 において、メインCPU 1 0 1 が、他の入賞がないと判別したとき (S 7 7 6 が N O 判定の場合)、メインCPU 1 0 1 は、入賞検索カウンタの値を 1 減算 (-1 更新) する (S 7 7 7)。なお、本実施形態のように、有効ラインが 1 本である場合には、複数の小役が重複して入賞することがないので、S 7 7 6 の判定処理は必ず N O 判定となる。

【1 0 4 2】

次いで、メインCPU 1 0 1 は、入賞検索カウンタの値が「0」であるか否かを判別する (S 7 7 8)。

【1 0 4 3】

S 7 7 8 において、メインCPU 1 0 1 が、入賞検索カウンタの値が「0」でないと判別したとき (S 7 7 8 が N O 判定の場合)、メインCPU 1 0 1 は、処理を S 7 6 4 の処理に戻し、S 7 6 4 以降の処理を繰り返す。一方、S 7 7 8 において、メインCPU 1 0 1 が、入賞検索カウンタの値が「0」であると判別したとき (S 7 7 8 が Y E S 判定の場合)、メインCPU 1 0 1 は、入賞検索処理を終了し、処理をメインフロー (図 8 2 参照) 中の S 2 1 5 の処理に移す。

40

【1 0 4 4】

本実施形態では、上述のようにして入賞検索処理が行われる。そして、上述した入賞検索処理は、メインCPU 1 0 1 が、図 1 4 6 のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中でも、S 7 6 4 の払出枚数及び判定対

50

象データのセット処理は、メインCPU101がソースコード「LDIN AC, (HL)」を実行することにより行われる。

【1045】

ソースプログラム上において、例えば、ソースコード「LDIN ss, (HL)」が実行されると、HLレジスタ（ペアレジスタ）にセットされたアドレス及び該アドレスに1加算したアドレスで指定されるメモリの内容（データ）がss（BC、DE、AC、AE又はBD）ペアレジスタにロードされるとともに、HLレジスタにセットされているアドレスが+2更新（2加算）される。それゆえ、図146中のソースコード「LDIN AC, (HL)」が実行されると、HLレジスタ（ペアレジスタ）にセットされたアドレス及び該アドレスに1加算したアドレスで指定されるメモリの内容（払出枚数及び判定対象データ）が、ACレジスタにロードされるとともに、HLレジスタにセットされているアドレスが+2更新（2加算）される。なお、S764の処理では、この「LDIN」命令により、Aレジスタに払出枚数のデータが格納され、Cレジスタに判定対象データが格納され、HLレジスタにセットされているアドレスが+2更新される。

【1046】

上述のように、本実施形態の入賞検索処理では、一つの「LDIN」命令により、データのロード処理及びアドレスの更新処理の両方を行うことができる。この場合、ソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【1047】

また、上述した入賞検索処理中のS770の判定処理で参照するメダルカウンタの値の取得処理、S772の処理で参照する入賞枚数カウンタの値の取得処理、及び、S775の処理で行う入賞枚数カウンタの保存（更新）処理はいずれも、図146に示すように、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う「LDQ」命令（メインCPU101専用命令コード）により実行される。それゆえ、本実施形態の入賞検索処理では、「LDQ」命令を用いることにより、直値により、メインROM102、メインRAM103やメモリーマップE/Oにアクセスすることができるので、ソースプログラム上において、アドレス設定に係る命令を省略することができ（アドレス設定に係る命令を別途設ける必要がなくなる）、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【1048】

また、上述した入賞検索処理中のS769の判定処理は、図146に示すように、ソースプログラム上において、「JSLAA」命令（所定の判定命令）により実行される。なお、「JSLAA」命令は、左シフト（SLA）命令相当の動作を実行する命令である。

【1049】

ソースプログラム上において、例えば、ソースコード「JSLAA cc, e」が実行されると、ccの条件が成立すれば、処理をeで指定されるアドレスにジャンプさせる。なお、「JSLAA」命令で規定される「ccの条件」には、フラグ・レジスタF内のキャリーフラグの状態が指定される。例えば、ccに「C」が指定されていれば、ccの条件はキャリーフラグが「1」（オン状態）であることを意味し、ccに「NC」が指定されていれば、ccの条件はキャリーフラグが「0」（オフ状態）であることを意味する。それゆえ、図146中のソースコード「JSLAA NC, MN_CKLN_06」では、キャリーフラグが「0」（オフ状態）であれば、「MN_CKLN_06」で指定されるアドレスに処理がジャンプする。

【1050】

また、上述した入賞検索処理中のS770及びS773の判定処理は、図146に示すよ

うに、ソースプログラム上において、「JCP」命令により実行される。なお、「JCP」命令は、上述のように、比較命令相当の動作を実行する命令であり、メインCPU101専用命令コードである。

【1051】

それゆえ、入賞検索処理のソースプログラム上において、上述した「JSLAA」命令及び「JCP」命令を用いた場合、アドレス設定に係る命令を省略することができ（アドレス設定に係る命令を別途設ける必要がなくなる）、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

10

【1052】

【イリーガルヒットチェック処理】

次に、図148及び図149を参照して、メインフロー（図82参照）中のS215で行うイリーガルヒットチェック処理について説明する。なお、図148は、イリーガルヒットチェック処理の手順を示すフローチャートである。また、図149は、イリーガルヒットチェック処理を実行するためのソースプログラムの一例を示す図である。なお、イリーガルヒットとは、内部抽籤処理（図92参照）で抽籤され、図柄設定処理（図97参照）で当籤番号格納領域に格納されたBB当籤番号及び小役当籤番号（内部当籤役）に基づいて、左リール3L、中リール3C及び右リール3Rが、成立しえない図柄の組合せで有効ライン上に停止（図柄組合せ不成立）したことを示す用語である。

20

【1053】

まず、メインCPU101は、入賞作動フラグ格納領域（図28～図30参照）のアドレスをセットする（S781）。次いで、メインCPU101は、入賞作動フラグ格納領域のサイズ（バイト数、本実施形態では「12」）を、チェックカウンタの値にセットする（S782）。

【1054】

次いで、メインCPU101は、現在セットされている入賞作動フラグ格納領域のアドレスに基づいて、該アドレスに対応する当り要求フラグ格納領域（内部当籤役格納領域）内の格納領域に格納された内部当籤役のデータ（当り要求フラグデータ）を取得する（S783）。次いで、メインCPU101は、現在セットされている入賞作動フラグ格納領域のアドレスに格納された入賞役のデータ（入賞作動フラグデータ）と、内部当籤役のデータ（当り要求フラグデータ）とを合成する（S784）。

30

【1055】

なお、この合成処理では、まず、メインCPU101は、入賞役のデータ（入賞作動フラグデータ）と内部当籤役のデータ（当り要求フラグデータ）との排他的論理和を求める（図149に示すソースプログラム中のソースコード「XOR（HL）」）。次いで、メインCPU101は、求められた排他的論理和の算出結果と入賞役のデータ（入賞作動フラグデータ）との論理積を求め（図149に示すソースプログラム中のソースコード「AND（HL）」）、論理積の算出結果を合成結果とする。なお、イリーガルヒットエラーが発生していない場合、この合成結果の値は「0」となる。

40

【1056】

次いで、メインCPU101は、S784の合成処理の結果に基づいて、イリーガルヒットエラーが発生しているか否かを判別する（S785）。

【1057】

S785において、メインCPU101が、イリーガルヒットエラーが発生していないと判別したとき（S785がNO判定の場合）、メインCPU101は、参照する入賞作動フラグ格納領域のアドレスを+1更新する（S786）。次いで、メインCPU101は、チェックカウンタの値を1減算する（S787）。次いで、メインCPU101は、チェックカウンタの値が「0」であるか否かを判別する（S788）。

【1058】

50

S 7 8 8において、メインCPU101が、チェックカウンタの値が「0」でないと判別したとき（S 7 8 8がNO判定の場合）、メインCPU101は、処理をS 7 8 3の処理に戻し、S 7 8 3以降の処理を繰り返す。一方、S 7 8 8において、メインCPU101が、チェックカウンタの値が「0」であると判別したとき（S 7 8 8がYES判定の場合）、メインCPU101は、イリーガルヒットチェック処理を終了し、処理をメインフロー（図82参照）中のS 2 1 6の処理に移す。

【1059】

ここで再度、S 7 8 5の処理に戻って、S 7 8 5において、メインCPU101が、イリーガルヒットエラーが発生していると判別したとき（S 7 8 5がYES判定の場合）、メインCPU101は、図89で説明したエラー処理を行う（S 7 8 9）。この処理により、情報表示器6に含まれる2桁の7セグLED（払出枚数表示用及びエラー表示用兼用）に、イリーガルヒットエラーの発生を示す2文字「EE」をエラー情報として表示するためのエラー表示データが出力される。なお、イリーガルヒットエラーの発生状態（エラー状態）は、リセットスイッチ76（図7参照）を押下することにより解除される。

10

【1060】

次いで、メインCPU101は、入賞枚数カウンタの値及び当り要求フラグ格納領域のデータをクリアする（S 7 9 0）。そして、S 7 9 0の処理後、メインCPU101は、イリーガルヒットチェック処理を終了し、処理をメインフロー（図82参照）中のS 2 1 6の処理に移す。

【1061】

本実施形態では、上述のようにしてイリーガルヒットチェック処理が行われる。そして、上述したイリーガルヒットチェック処理は、メインCPU101が、図149のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

20

【1062】

なお、本実施形態では、図28～図30に示すように、入賞作動フラグ格納領域（表示役格納領域）の構成が当り要求フラグ格納領域（内部当籤役格納領域）のそれと同じであるので、入賞作動フラグ格納領域の役と内部当籤役との合成処理時にメインRAM103に配置される当り要求フラグ格納領域と入賞作動フラグ格納領域とを同一構成にすることができる。それゆえ、本実施形態のイリーガルヒットチェック処理におけるS 7 8 4の演算結果（入賞役のデータと内部当籤役のデータとを合成結果）は、上述のように、ソースプログラム上において、入賞役のデータと内部当籤役のデータとを単純に論理積（「AND」命令で実行する）することにより求められる。その結果、本実施形態では、イリーガルヒットチェック処理を効率化及び簡略化することができ、主制御プログラムの空き容量を確保する（増やす）ことができ、増えた空き容量を使用して遊技性を高めることが可能になる。

30

【1063】

〔入賞チェック・メダル払出処理〕

次に、図150及び図151を参照して、メインフロー（図82参照）中のS 2 1 6で行う入賞チェック・メダル払出処理について説明する。なお、図150は、入賞チェック・メダル払出処理の手順を示すフローチャートである。また、図151は、入賞チェック・メダル払出処理中の後述のS 8 0 4～S 8 0 8の処理を実行するためのソースプログラムの一例を示す図である。

40

【1064】

まず、メインCPU101は、入賞作動コマンド生成処理を行う（S 8 0 1）。この処理では、メインCPU101は、副制御回路200に送信する入賞作動コマンドに含まれる、種別データ及び各種通信パラメータを生成する。なお、入賞作動コマンドは、入賞作動フラグ（表示役）等を特定するパラメータを含んで構成される。

【1065】

次いで、メインCPU101は、図72で説明した通信データ格納処理を行う（S 8 0 2）。この処理により、入賞作動コマンドデータがメインRAM103に設けられた通信デ

50

一タ格納領域（図 7 5 B 参照）に保存される。なお、入賞作動コマンドは、後述の図 1 5 8 で説明する割込処理内の通信データ送信処理により、主制御回路 9 0 から副制御回路 2 0 0 に送信される。

【1 0 6 6】

次いで、メイン CPU 1 0 1 は、入賞枚数カウンタの値が「0」であるか否かを判別する（S 8 0 3）。S 8 0 3 において、メイン CPU 1 0 1 が、入賞枚数カウンタの値が「0」であると判別したとき（S 8 0 3 が Y E S 判定の場合）、メイン CPU 1 0 1 は、入賞チェック・メダル払出処理を終了し、処理をメインフロー（図 8 2 参照）中の S 2 1 7 の処理に移す。

【1 0 6 7】

一方、S 8 0 3 において、メイン CPU 1 0 1 が、入賞枚数カウンタの値が「0」でないと判別したとき（S 8 0 3 が N O 判定の場合）、メイン CPU 1 0 1 は、メダルのクレジット枚数（貯留枚数）がその上限枚数（本実施形態では 5 0 枚）以上であるか否かを判別する（S 8 0 4）。

【1 0 6 8】

S 8 0 4 において、メイン CPU 1 0 1 が、メダルのクレジット枚数がその上限枚数以上でないと判別したとき（S 8 0 4 が N O 判定の場合）、メイン CPU 1 0 1 は、クレジットカウンタの値に「1」を加算（+ 1 更新）する（S 8 0 5）。加算されたクレジットカウンタの値は、情報表示器 6 に含まれる貯留枚数表示用の 2 桁の 7 セグ L E D（不図示）により表示される。次いで、メイン CPU 1 0 1 は、メダル払出枚数チェック処理を行う（S 8 0 6）。なお、メダル払出枚数チェック処理の詳細については、後述の図 1 5 2 を参照しながら後で説明する。

【1 0 6 9】

次いで、メイン CPU 1 0 1 は、メダルの払い出しが終了したか否かを判別する（S 8 0 7）。S 8 0 7 において、メイン CPU 1 0 1 が、メダルの払い出しが終了したと判別したとき（S 8 0 7 が Y E S 判定の場合）、メイン CPU 1 0 1 は、入賞チェック・メダル払出処理を終了し、処理をメインフロー（図 8 2 参照）中の S 2 1 7 の処理に移す。

【1 0 7 0】

一方、S 8 0 7 において、メイン CPU 1 0 1 が、メダルの払い出しが終了していないと判別したとき（S 8 0 7 が N O 判定の場合）、メイン CPU 1 0 1 は、払出間隔待機処理を行う（S 8 0 8）。この処理では、メイン CPU 1 0 1 は、予め設定されたメダル払出間隔時間（本実施形態では 6 0 . 3 3 m s e c : 後述の図 1 5 8 で説明する割込処理（1 . 1 1 7 2 m s e c 周期）の 5 4 周期分）が経過するまでウェイトする。そして、S 8 0 8 の処理後、メイン CPU 1 0 1 は、処理を S 8 0 3 の処理に戻し、S 8 0 3 以降の処理を繰り返す。

【1 0 7 1】

ここで再度、S 8 0 4 の処理に戻って、S 8 0 4 において、メイン CPU 1 0 1 が、メダルのクレジット枚数がその上限枚数（5 0 枚）以上であると判別したとき（S 8 0 4 が Y E S 判定の場合）、メイン CPU 1 0 1 は、メダルの払出処理を行う（S 8 0 9）。この処理により、メダルが 1 枚、払い出される。そして、S 8 0 9 の処理後、メイン CPU 1 0 1 は、入賞チェック・メダル払出処理を終了し、処理をメインフロー（図 8 2 参照）中の S 2 1 7 の処理に移す。

【1 0 7 2】

本実施形態では、上述のようにして入賞チェック・メダル払出処理が行われる。なお、上述した入賞チェック・メダル払出処理中の S 8 0 4 ~ S 8 0 8 の処理は、メイン CPU 1 0 1 が、図 1 5 1 のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【1 0 7 3】

なお、本実施形態では、クレジットカウンタの更新（+ 1）後、払出動作を継続する場合、メイン CPU 1 0 1 は、S 8 0 8 の処理で 6 0 . 3 3 m s 間のウェイト（払出間隔待ち

10

20

30

40

50

）処理を行うが、この処理は、ソースプログラム上では、メインCPU101がソースコード「LD BC, cTM_PAYC」及び「RST SB_W1BC_00」をこの順で実行することにより実現されている。このように、入賞チェック・メダル払出処理において、クレジットカウンタの更新（+1）後、払出動作を継続するときに60.33ms間のウェイト（払出間隔待ち）を行った場合、無駄な待ち時間を減らすことができ、遊技者の精神的負担を軽減することができる。

【1074】

〔メダル払出枚数チェック処理〕

次に、図152及び図153を参照して、入賞チェック・メダル払出処理（図150参照）中のS806で行うメダル払出枚数チェック処理について説明する。なお、図152は、メダル払出枚数チェック処理の手順を示すフローチャートである。また、図153Aは、メダル払出枚数チェック処理中の後述のS811～S814の処理を実行するためのソースプログラムの一例を示す図であり、図153Bは、メダル払出枚数チェック処理中の後述のS816及びS817の処理を実行するためのソースプログラムの一例を示す図である。

【1075】

まず、メインCPU101は、メダルOUTカウンタの値に「1」を加算（+1更新）する（S811）。なお、メダルOUTカウンタは、メダルの払出回数を計数するためのカウンタである。次いで、メインCPU101は、払出枚数カウンタの値に「1」を加算（+1更新）する（S812）。なお、払出枚数カウンタは、メダルの払出枚数を計数するためのカウンタである。

【1076】

次いで、メインCPU101は、払出枚数7SEG表示処理を行う（S813）。この処理では、メインCPU101は、払出枚数カウンタの値を、情報表示器6に含まれる払出枚数表示用の2桁の7セグLED（不図示）により表示させる制御処理を行う。

【1077】

次いで、メインCPU101は、役連終了枚数カウンタの更新処理を行う（S814）。なお、役連終了枚数カウンタは、入賞役に対応するメダルの払出枚数の残り枚数を計数するためのカウンタである。この処理では、メインCPU101は、役連終了枚数カウンタの値とその下限値「0」とを比較し、役連終了枚数カウンタの値が下限値「0」より大きい場合には、役連終了枚数カウンタの値を1減算（-1更新）し、役連終了枚数カウンタの値が下限値「0」以下である場合には、役連終了枚数カウンタの値を「0」に保持する。

【1078】

次いで、メインCPU101は、入賞枚数カウンタの値を1減算（-1更新）する（S815）。

【1079】

次いで、メインCPU101は、クレジット情報コマンド生成処理を行う（S816）。この処理では、メインCPU101は、副制御回路200に送信するクレジット情報コマンドに含まれる、種別データ及び各種通信パラメータを生成する。なお、クレジット情報コマンドは、メダルのクレジット枚数を特定するパラメータを含んで構成される。

【1080】

次いで、メインCPU101は、図72で説明した通信データ格納処理を行う（S817）。この処理により、クレジット情報コマンドデータがメインRAM103に設けられた通信データ格納領域（図75B参照）に保存される。なお、クレジット情報コマンドは、後述の図158で説明する割込処理内の通信データ送信処理により、主制御回路90から副制御回路200に送信される。そして、S817の処理後、メインCPU101は、メダル払出枚数チェック処理を終了し、処理を入賞チェック・メダル払出処理（図150参照）中のS807の処理に移す。

【1081】

本実施形態では、上述のようにしてメダル払出枚数チェック処理が行われる。なお、上述

10

20

30

40

50

したメダル払出枚数チェック処理中のS 8 1 1～S 8 1 4の処理は、メインCPU 1 0 1が、図1 5 3 Aのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。その中で、S 8 1 4の役連終了枚数カウンタの更新処理は、図1 5 3 A中の「DCPLD」命令（所定の更新命令）により実行される。なお、「DCPLD」命令は、メインCPU 1 0 1専用命令コードである。

【1 0 8 2】

ソースプログラム上において、例えば、ソースコード「DCPLD (HL), n」が実行されると、HLレジスタで指定されたアドレスのメモリの内容（格納データ）と整数nとが比較され、メモリの内容が整数nより大きい場合には、メモリの内容が1減算され、メモリの内容が整数n以下である場合には、HLレジスタで指定されたアドレスのメモリに整数nが格納される。それゆえ、図1 5 3 A中のソースコード「DCPLD (HL), 0」が実行されると、HLレジスタで指定されたアドレスのメモリの内容（役連終了枚数カウンタの値）と整数0（下限値）とが比較され、メモリの内容（役連終了枚数カウンタの値）が整数0より大きい場合には、メモリの内容が1減算され、メモリの内容が整数0以下である場合には、メモリの内容（役連終了枚数カウンタの値）に「0」がセットされる。すなわち、現時点の役連終了枚数カウンタの値が「0」より大きい場合には、役連終了枚数カウンタの更新処理が行われ、現時点の役連終了枚数カウンタの値が「0」以下であれば、役連終了枚数カウンタの値を「0」に保持する処理が行われる。

【1 0 8 3】

上述のように、メダル払出枚数チェック処理中のS 8 1 4の処理では、一つの「DCPLD」命令（枚数管理カウンタの下限判定命令と、判断分岐命令が一体になっている命令）により、役連終了枚数カウンタの更新（減算）処理及び連終了枚数カウンタの値を「0」に保持する処理の両方を実行することができる。この場合、両処理を別個に実行するための命令コードを設ける必要がなくなる。例えば、連終了枚数カウンタの値が「0」であるか否かを判別するための判断分岐命令コードを省略することができる。それゆえ、本実施形態のメダル払出枚数チェック処理では、ソースプログラムの容量（メインROM 1 0 2の使用容量）を低減することができ、メインROM 1 0 2において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【1 0 8 4】

また、上述したメダル払出枚数チェック処理中のS 8 1 6及びS 8 1 7の処理は、メインCPU 1 0 1が、図1 5 3 Bのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【1 0 8 5】

その中で、S 8 1 6の処理では、図1 5 3 Bに示すように、クレジット情報コマンドの通信パラメータ1にはLレジスタを介して払出枚数カウンタの値がセットされ、通信パラメータ5にはCレジスタを介してクレジットカウンタの値がセットされる。しかしながら、クレジット情報コマンドを構成するその他の通信パラメータ2～4には、現時点においてHレジスタ、Eレジスタ及びDレジスタにそれぞれ格納されている値（不定値）がセットされる。それゆえ、クレジット情報コマンド送信時における通信パラメータ2～4の値は不定値となる。その結果、本実施形態では、クレジット情報コマンドのサム値（B C C）を送信毎に不定値にすることができ、ゴト等の不正行為を抑制することができる。

【1 0 8 6】

〔BBチェック処理〕

次に、図1 5 4を参照して、メインフロー（図8 2参照）中のS 2 1 7で行うBBチェック処理について説明する。なお、図1 5 4は、BBチェック処理の手順を示すフローチャートである。

【1 0 8 7】

まず、メインCPU 1 0 1は、現在の遊技状態がボーナス状態であるか否かを判別する（S 8 2 1）。S 8 2 1において、メインCPU 1 0 1が、現在の遊技状態がボーナス状態

10

20

30

40

50

でないと判別したとき（S 8 2 1 が N O 判定の場合）、メイン C P U 1 0 1 は、後述の S 8 3 2 の処理を行う。

【1 0 8 8】

一方、S 8 2 1 において、メイン C P U 1 0 1 が、現在の遊技状態がボーナス状態であると判別したとき（S 8 2 1 が Y E S 判定の場合）、メイン C P U 1 0 1 は、ボーナス状態中に払い出し可能なメダルの枚数を計数するための B B 中払出枚数カウンタの値から、入賞チェック・メダル払出処理において払い出されたメダルの払出枚数を減算する（S 8 2 2）。

【1 0 8 9】

次いで、メイン C P U 1 0 1 は、B B 中払出枚数カウンタの値が「0」未満であるか否かを判別する（S 8 2 3）。S 8 2 3 において、メイン C P U 1 0 1 が、B B 中払出枚数カウンタの値が「0」未満でないと判別したとき（S 8 2 3 が N O 判定の場合）、メイン C P U 1 0 1 は、B B チェック処理を終了し、処理をメインフロー（図 8 2 参照）中の S 2 1 8 の処理に移す。

【1 0 9 0】

一方、S 8 2 3 において、メイン C P U 1 0 1 が、B B 中払出枚数カウンタの値が「0」未満であると判別したとき（S 8 2 3 が Y E S 判定の場合）、メイン C P U 1 0 1 は、ボーナス終了時処理を行う（S 8 2 4）。この処理では、メイン C P U 1 0 1 は、ボーナス状態中の各種情報をクリアするとともに、R T 1 状態フラグをオン状態にセットする。

【1 0 9 1】

次いで、メイン C P U 1 0 1 は、ボーナス終了時 C T 抽籤テーブル（図 6 0 参照）を参照して、ボーナス終了時の C T 抽籤を行う（S 8 2 5）。次いで、メイン C P U 1 0 1 は、ボーナス終了時の C T 抽籤に当籤したか否かを判別する（S 8 2 6）。

【1 0 9 2】

S 8 2 6 において、メイン C P U 1 0 1 が、C T 抽籤に当籤しなかったと判別したとき（S 8 2 6 が N O 判定の場合）、メイン C P U 1 0 1 は、後述の S 8 2 8 の処理を行う。一方、S 8 2 6 において、メイン C P U 1 0 1 が、C T 抽籤に当籤したと判別したとき（S 8 2 6 が Y E S 判定の場合）、メイン C P U 1 0 1 は、C T セット数に「1」を加算する（S 8 2 7）。なお、A R T セット数が「0」であるときに C T 抽籤に当籤した場合には、S 8 2 7 の処理において、C T セット数に「1」を加算するとともに、A R T セット数にも「1」を加算する。

【1 0 9 3】

S 8 2 7 の処理後又は S 8 2 6 が N O 判定の場合、メイン C P U 1 0 1 は、A R T セット数又は C T セット数が「1」以上であるか否かを判別する（S 8 2 8）。

【1 0 9 4】

S 8 2 8 おいて、メイン C P U 1 0 1 が、A R T セット数又は C T セット数が「1」以上であると判別したとき（S 8 2 8 が Y E S 判定の場合）、メイン C P U 1 0 1 は、次遊技の遊技状態に A R T 準備状態をセットする（S 8 2 9）。そして、S 8 2 9 の処理後、メイン C P U 1 0 1 は、B B チェック処理を終了し、処理をメインフロー（図 8 2 参照）中の S 2 1 8 の処理に移す。

【1 0 9 5】

一方、S 8 2 8 において、メイン C P U 1 0 1 が、A R T セット数又は C T セット数が「1」以上でないと判別したとき（S 8 2 8 が N O 判定の場合）、メイン C P U 1 0 1 は、次遊技の遊技状態に通常遊技状態をセットする（S 8 3 0）。次いで、メイン C P U 1 0 1 は、通常中高確率抽籤テーブル（図 4 0 B 参照）を参照して、C Z の抽籤状態を抽籤し、抽籤結果をセットする（S 8 3 1）。そして、S 8 3 1 の処理後、メイン C P U 1 0 1 は、B B チェック処理を終了し、処理をメインフロー（図 8 2 参照）中の S 2 1 8 の処理に移す。

【1 0 9 6】

ここで再度、S 8 2 1 の処理に戻って、S 8 2 1 が N O 判定の場合、メイン C P U 1 0 1

10

20

30

40

50

は、BB役に係る図柄組合せ（コンビネーション「C__BB1」又は「C__BB2」の図柄組合せ）が表示されたか否かを判定する（S832）。S832において、メインCPU101が、BB役に係る図柄組合せが表示されなかったと判別したとき（S832がNO判定の場合）、メインCPU101は、BBチェック処理を終了し、処理をメインフロー（図82参照）中のS218の処理に移す。

【1097】

一方、S832において、メインCPU101が、BB役に係る図柄組合せが表示されたと判別したとき（S832がYES判定の場合）、メインCPU101は、ボーナス種別抽籤テーブル（図58参照）を参照して、ボーナス種別を抽籤し、抽籤結果をセットする（S833）。次いで、メインCPU101は、BB中払出枚数カウンタの値に所定値（ボーナス終了契機となる払出枚数：本実施形態では、「216」）をセットする（S834）。

10

【1098】

次いで、メインCPU101は、ボーナス開始時処理を行う（S835）。この処理では、メインCPU101は、例えば、次遊技の遊技状態にボーナス状態をセットするなどのボーナスの作動開始に必要な各種処理を行う。そして、S835の処理後、メインCPU101は、BBチェック処理を終了し、処理をメインフロー（図82参照）中のS218の処理に移す。

【1099】

〔RTチェック処理〕

20

次に、図155及び図156を参照して、メインフロー（図82参照）中のS218で行うRTチェック処理について説明する。なお、図155及び図156は、RTチェック処理の手順を示すフローチャートである。

【1100】

まず、メインCPU101は、RT状態がRT5状態であるか否かを判別する（S841）。S841において、メインCPU101が、RT状態がRT5状態であると判別したとき（S841がYES判定の場合）、メインCPU101は、RTチェック処理を終了し、処理をメインフロー（図82参照）中のS219の処理に移す。

【1101】

一方、S841において、メインCPU101が、RT状態がRT5状態でないと判別したとき（S841がNO判定の場合）、メインCPU101は、RT状態がRT0状態であるか否かを判別する（S842）。S842において、メインCPU101が、RT状態がRT0状態でないと判別したとき（S842がNO判定の場合）、メインCPU101は、後述のS845の処理を行う。

30

【1102】

一方、S842において、メインCPU101が、RT状態がRT0状態であると判別したとき（S842がYES判定の場合）、メインCPU101は、略称「ベルこぼし目」の図柄組合せ（図28参照）が表示されたか否かを判別する（S843）。S843において、メインCPU101が、略称「ベルこぼし目」の図柄組合せが表示されなかったと判別したとき（S843がNO判定の場合）、メインCPU101は、RTチェック処理を終了し、処理をメインフロー（図82参照）中のS219の処理に移す。

40

【1103】

一方、S843において、メインCPU101が、略称「ベルこぼし目」の図柄組合せが表示されたと判別したとき（S843がYES判定の場合）、メインCPU101は、RT2状態フラグをオン状態にセットする（S844）。この処理により、RT状態がRT0状態からRT2状態に移行する。そして、S844の処理後、メインCPU101は、RTチェック処理を終了し、処理をメインフロー（図82参照）中のS219の処理に移す。

【1104】

ここで再度、S842の処理に戻って、S842がNO判定の場合、メインCPU101

50

は、RT状態がRT1状態であるか否かを判別する（S845）。S845において、メインCPU101が、RT状態がRT1状態でないと判別したとき（S845がNO判定の場合）、メインCPU101は、後述のS850の処理を行う。

【1105】

一方、S845において、メインCPU101が、RT状態がRT1状態であると判別したとき（S845がYES判定の場合）、メインCPU101は、略称「ベルこぼし目」の図柄組合せが表示されたか否かを判別する（S846）。

【1106】

S846において、メインCPU101が、略称「ベルこぼし目」の図柄組合せが表示されたと判別したとき（S846がYES判定の場合）、メインCPU101は、RT1状態フラグをオフ状態にセットするとともに、RT2状態フラグをオン状態にセットする（S847）。この処理により、RT状態がRT1状態からRT2状態に移行する。そして、S847の処理後、メインCPU101は、RTチェック処理を終了し、処理をメインフロー（図82参照）中のS219の処理に移す。

10

【1107】

一方、S846において、メインCPU101が、略称「ベルこぼし目」の図柄組合せが表示されなかったと判別したとき（S846がNO判定の場合）、メインCPU101は、RT1状態の遊技が20ゲーム経過したか否かを判別する（S848）。

【1108】

S848において、メインCPU101が、RT1状態の遊技が20ゲーム経過していないと判別したとき（S848がNO判定の場合）、メインCPU101は、RTチェック処理を終了し、処理をメインフロー（図82参照）中のS219の処理に移す。一方、S848において、メインCPU101が、RT1状態の遊技が20ゲーム経過したと判別したとき（S848がYES判定の場合）、メインCPU101は、RT1状態フラグをオフ状態にセットする（S849）。この処理により、RT状態がRT1状態からRT0状態に移行する。そして、S849の処理後、メインCPU101は、RTチェック処理を終了し、処理をメインフロー（図82参照）中のS219の処理に移す。

20

【1109】

ここで再度、S845の処理に戻って、S845がNO判定の場合、メインCPU101は、RT状態がRT2状態であるか否かを判別する（S850）。S850において、メインCPU101が、RT状態がRT2状態でないと判別したとき（S850がNO判定の場合）、メインCPU101は、後述のS853の処理を行う。

30

【1110】

一方、S850において、メインCPU101が、RT状態がRT2状態であると判別したとき（S850がYES判定の場合）、メインCPU101は、略称「RT3移行リブ」の図柄組合せ（図28参照）が表示されたか否かを判別する（S851）。S851において、メインCPU101が、略称「RT3移行リブ」の図柄組合せが表示されなかったと判別したとき（S851がNO判定の場合）、メインCPU101は、RTチェック処理を終了し、処理をメインフロー（図82参照）中のS219の処理に移す。

【1111】

一方、S851において、メインCPU101が、略称「RT3移行リブ」の図柄組合せが表示されたと判別したとき（S851がYES判定の場合）、メインCPU101は、RT2状態フラグをオフ状態にセットするとともに、RT3状態フラグをオン状態にセットする（S852）。この処理により、RT状態がRT2状態からRT3状態に移行する。そして、S852の処理後、メインCPU101は、RTチェック処理を終了し、処理をメインフロー（図82参照）中のS219の処理に移す。

40

【1112】

ここで再度、S850の処理に戻って、S850がNO判定の場合、メインCPU101は、RT状態がRT3状態であるか否かを判別する（S853）。S853において、メインCPU101が、RT状態がRT3状態でないと判別したとき（S853がNO判定

50

の場合)、メインCPU101は、後述のS862の処理を行う。

【1113】

一方、S853において、メインCPU101が、RT状態がRT3状態であると判別したとき(S853がYES判定の場合)、メインCPU101は、略称「ベルこぼし目」又は「RT2移行リップ」の図柄組合せが表示されたか否かを判別する(S854)。

【1114】

S854において、メインCPU101が、略称「ベルこぼし目」又は「RT2移行リップ」の図柄組合せが表示されたと判別したとき(S854がYES判定の場合)、メインCPU101は、RT3状態フラグをオフ状態にセットするとともに、RT2状態フラグをオン状態にセットする(S855)。この処理により、RT状態がRT3状態からRT2状態に移行する。そして、S855の処理後、メインCPU101は、RTチェック処理を終了し、処理をメインフロー(図82参照)中のS219の処理に移す。

10

【1115】

一方、S854において、メインCPU101が、略称「ベルこぼし目」又は「RT2移行リップ」の図柄組合せが表示されていないと判別したとき(S854がNO判定の場合)、メインCPU101は、略称「RT4移行リップ」の図柄組合せ(図28参照)が表示されたか否かを判別する(S856)。

【1116】

S856において、メインCPU101が、略称「RT4移行リップ」の図柄組合せが表示されていないと判別したとき(S856がNO判定の場合)、メインCPU101は、RTチェック処理を終了し、処理をメインフロー(図82参照)中のS219の処理に移す。一方、S856において、メインCPU101が、略称「RT4移行リップ」の図柄組合せが表示されたと判別したとき(S856がYES判定の場合)、メインCPU101は、RT3状態フラグをオフ状態にセットするとともに、RT4状態フラグをオン状態にセットする(S857)。この処理により、RT状態がRT3状態からRT4状態に移行する。

20

【1117】

S857の処理後、メインCPU101は、遊技状態がART準備状態であるか否かを判別する(S858)。S858において、メインCPU101が、遊技状態がART準備状態でないと判別したとき(S858がNO判定の場合)、メインCPU101は、RTチェック処理を終了し、処理をメインフロー(図82参照)中のS219の処理に移す。

30

【1118】

一方、S858において、メインCPU101が、遊技状態がART準備状態であると判別したとき(S858がYES判定の場合)、メインCPU101は、CTセット数が「1」以上であるか否かを判別する(S859)。

【1119】

S859において、メインCPU101が、CTセット数が「1」以上であると判別したとき(S859がYES判定の場合)、メインCPU101は、次遊技の遊技状態にCTをセットし、CTゲーム数カウンタに「8」をセットする(S860)。そして、S860の処理後、メインCPU101は、RTチェック処理を終了し、処理をメインフロー(図82参照)中のS219の処理に移す。

40

【1120】

一方、S859において、メインCPU101が、CTセット数が「1」以上でないと判別したとき(S859がNO判定の場合)、メインCPU101は、次遊技の遊技状態に通常ARTをセットし、ART終了ゲーム数カウンタに所定値をセットする(S861)。そして、S861の処理後、メインCPU101は、RTチェック処理を終了し、処理をメインフロー(図82参照)中のS219の処理に移す。

【1121】

ここで再度、S853の処理に戻って、S853がNO判定の場合、メインCPU101は、略称「ベルこぼし目」又は「RT2移行リップ」の図柄組合せが表示されたか否かを判

50

別する（S 8 6 2）。S 8 6 2において、メインCPU 1 0 1が、略称「ベルこぼし目」又は「R T 2 移行リップ」の図柄組合せが表示されていないと判別したとき（S 8 6 2がN O判定の場合）、メインCPU 1 0 1は、R Tチェック処理を終了し、処理をメインフロー（図8 2参照）中のS 2 1 9の処理に移す。

【1 1 2 2】

一方、S 8 6 2において、メインCPU 1 0 1が、略称「ベルこぼし目」又は「R T 2 移行リップ」の図柄組合せが表示されたと判別したとき（S 8 6 2がY E S判定の場合）、メインCPU 1 0 1は、R T 4状態フラグをオフ状態にセットするとともに、R T 2状態フラグをオン状態にセットする（S 8 6 3）。この処理により、R T状態がR T 4状態からR T 2状態に移行する。そして、S 8 6 3の処理後、メインCPU 1 0 1は、R Tチェック処理を終了し、処理をメインフロー（図8 2参照）中のS 2 1 9の処理に移す。

10

【1 1 2 3】

〔C Z・A R T終了時処理〕

次に、図1 5 7を参照して、メインフロー（図8 2参照）中のS 2 1 9で行うC Z・A R T終了時処理について説明する。なお、図1 5 7は、C Z・A R T終了時処理の手順を示すフローチャートである。

【1 1 2 4】

まず、メインCPU 1 0 1は、現在の遊技状態がC Z失敗時及びA R T終了時のいずれかであるか否かを判別する（S 8 7 1）。S 8 7 1において、メインCPU 1 0 1が、現在の遊技状態がC Z失敗時及びA R T終了時のいずれかでないと判別したとき（S 8 7 1がN O判定の場合）、メインCPU 1 0 1は、C Z・A R T終了時処理を終了し、処理をメインフロー（図8 2参照）中のS 2 0 1の処理に移す。

20

【1 1 2 5】

一方、S 8 7 1において、メインCPU 1 0 1が、現在の遊技状態がC Z失敗時及びA R T終了時のいずれかであると判別したとき（S 8 7 1がY E S判定の場合）、メインCPU 1 0 1は、C Z抽籤テーブル（図4 1 B参照）を参照して、C Zの引き戻し抽籤を行う（S 8 7 2）。次いで、メインCPU 1 0 1は、C Zの引き戻し抽籤に当籤したか否かを判別す（S 8 7 3）。

【1 1 2 6】

S 8 7 3において、メインCPU 1 0 1が、C Zの引き戻し抽籤に当籤したと判別したとき（S 8 7 3がY E S判定の場合）、メインCPU 1 0 1は、次遊技の遊技状態に当籤した種別のC Zをセットする（S 8 7 4）。次いで、メインCPU 1 0 1は、当籤した種別のC Zに応じた値をC Zゲーム数カウンタにセットする（S 8 7 5）。そして、S 8 7 5の処理後、メインCPU 1 0 1は、C Z・A R T終了時処理を終了し、処理をメインフロー（図8 2参照）中のS 2 0 1の処理に移す。

30

【1 1 2 7】

一方、S 8 7 3において、メインCPU 1 0 1が、C Zの引き戻し抽籤に当籤しなかったと判別したとき（S 8 7 3がN O判定の場合）、メインCPU 1 0 1は、次遊技の遊技状態に通常遊技状態をセットする（S 8 7 6）。次いで、メインCPU 1 0 1は、通常中高確率抽籤テーブル（図4 0 B参照）を参照して、C Zの抽籤状態を抽籤し、抽籤結果をセットする（S 8 7 7）。そして、S 8 7 7の処理後、メインCPU 1 0 1は、C Z・A R T終了時処理を終了し、処理をメインフロー（図8 2参照）中のS 2 0 1の処理に移す。

40

【1 1 2 8】

〔メインCPUの制御による割込処理（1. 1 1 7 2 m s e c）〕

次に、図1 5 8を参照して、1. 1 1 7 2 m s e c周期で、メインCPU 1 0 1が行う割込処理について説明する。なお、図1 5 8は、割込処理の手順を示すフローチャートである。1. 1 1 7 2 m s e c周期で繰り返し実行される割込処理は、タイマー回路1 1 3（P T C）の初期化処理（図6 4中のS 2参照）で設定されたタイマー回路1 1 3のタイムアウト信号の出力タイミングに基づいて発生する割込みコントローラ1 1 2からの割込要求信号がメインCPU 1 0 1に入力された際に実行される処理である。

50

【1129】

まず、メインCPU101は、レジスタの退避処理を行う（S901）。次いで、メインCPU101は、入力ポートチェック処理を行う（S902）。この処理では、ストップスイッチ等の各種スイッチから入力される信号がチェックされる。

【1130】

次いで、メインCPU101は、リール制御処理を行う（S903）。この処理では、メインCPU101は、全リールの回転開始が要求されたときに、左リール3L、中リール3C及び右リール3Rの回転を開始し、その後、各リールが一定速度で回転するように、3つのステッピングモータを駆動制御する。また、滑り駒数が決定されたときは、メインCPU101は、該当するリールの図柄カウンタを滑り駒数分だけ更新する。そして、メインCPU101は、更新された図柄カウンタが停止予定位置に対応する値に一致する（停止予定位置の図柄が表示窓の有効ライン上の領域に到達する）のを待って、該当するリールの回転の減速及び停止が行われるように、対応するステッピングモータを駆動制御する。

10

【1131】

次いで、メインCPU101は、通信データ送信処理を行う（S904）。この処理では、主に、通信データ格納領域に格納された各種コマンドを主制御回路90の第1シリアル通信回路114（図9参照）を介して副制御回路200に送信する。メインCPU101は、副制御回路200にコマンドを送信した後、通信データポインタを1パケット分減算更新し（不図示）、通信データ格納領域の送信済みのコマンドデータをクリアする。なお、通信データ格納領域に複数のコマンドデータが格納されている場合には、格納された古い順で、コマンドデータを副制御回路200に送信する。また、通信データ格納領域にコマンドデータが格納されていない場合、すなわち、通信データポインタの値が「0」である場合には、無操作コマンドを生成して副制御回路200に送信する。次いで、メインCPU101は、投入メダル通過チェック処理を行う（S905）。この処理では、メインCPU101は、メダルセンサ（不図示）の検出結果（メダルセンサ入力状態）に基づいて、投入メダルがセクタ66を通過したか否かのチェック処理を行う。次いで、メインCPU101は、WDTのリスタート処理を行う（S906）。

20

【1132】

次いで、メインCPU101は、7セグLED駆動処理を行う（S907）。この処理では、メインCPU101は、情報表示器6に含まれる各種7セグLEDを駆動制御して、例えば、メダルの払出枚数やクレジット枚数、ストップボタンの押し順データなどを表示する。なお、7セグLED駆動処理の詳細については、後述の図159を参照しながら後で説明する。

30

【1133】

次いで、メインCPU101は、タイマー更新処理を行う（S908）。この処理では、メインCPU101は、セットされた各種タイマーのカウント（減算）処理を行う。なお、タイマー更新処理の詳細については、後述の図164を参照しながら後で説明する。

【1134】

次いで、メインCPU101は、エラー検知処理を行う（S909）。次いで、メインCPU101は、ドア開閉チェック処理を行う（S910）。ドア開閉チェック処理では、メインCPU101は、ドア開閉監視スイッチ67のオン（ドア閉）／オフ（ドア開）状態をチェックすることにより、フロントドア2b（図2参照）の開閉状態をチェックする。

40

【1135】

次いで、メインCPU101は、試射試験信号制御処理を行う（S911）。この処理では、第2インターフェースポート等を介して試験機に各種試験信号の出力する際の制御処理が行われる。また、この処理は、メインRAM103の規定外作業領域（図12C参照）を用いて実行される。なお、本実施形態では、この処理は、試射試験時以外のとき（パチスロ1が遊技店に設置された後）にも行われるが、この時には、主制御基板71が第2インターフェースポート等を介して試験機に接続されていないので、各種試験信号は生成

50

されても出力はされない。試射試験信号制御処理の詳細については、後述の図 1 6 6 を参照しながら後で説明する。

【1 1 3 6】

次いで、メイン CPU 1 0 1 は、レジスタの復帰処理を行う（S 9 1 2）。そして、S 9 1 2 の処理後、メイン CPU 1 0 1 は、割込処理を終了する。

【1 1 3 7】

〔7 セグ L E D 駆動処理〕

次に、図 1 5 9 及び図 1 6 0 を参照して、割込処理（図 1 5 8 参照）中の S 9 0 7 で行う 7 セグ L E D 駆動処理について説明する。なお、図 1 5 9 は、7 セグ L E D 駆動処理の手順を示すフローチャートである。また、図 1 6 0 A は、7 セグ L E D 駆動処理中の後述の S 9 2 3 ～ S 9 2 5 の処理を実行するためのソースプログラムの一例を示す図であり、図 1 6 0 B は、7 セグ L E D 駆動処理中の後述の S 9 3 1 ～ S 9 3 6 の処理を実行するためのソースプログラムの一例を示す図である。

【1 1 3 8】

まず、メイン CPU 1 0 1 は、割込カウンタの値に「1」を加算（+ 1 更新）する（S 9 2 1）。次いで、メイン CPU 1 0 1 は、割込カウンタの値が奇数であるか否かを判別する（S 9 2 2）。

【1 1 3 9】

S 9 2 2 において、メイン CPU 1 0 1 が、割込カウンタの値が奇数でないと判別したとき（S 9 2 2 が N O 判定の場合）、メイン CPU 1 0 1 は、7 セグ L E D 駆動処理を終了し、処理を割込処理（図 1 5 8 参照）中の S 9 0 8 の処理に移す。すなわち、本実施形態では、2 回の割込周期毎に、7 セグ L E D 駆動処理が行われる。なお、本実施形態では、7 セグ L E D 駆動処理を割込みカウンタの値が偶数の場合に実行する例を説明したが、本発明はこれに限定されず、割込みカウンタの値が奇数の場合に 7 セグ L E D 駆動処理を実行してもよいし、また、任意の整数で割込みカウンタの値を除算したときの商又は余りを用いて、7 セグ L E D 駆動処理の実行タイミングを決定してもよい。

【1 1 4 0】

一方、S 9 2 2 において、メイン CPU 1 0 1 が、割込カウンタの値が奇数であると判別したとき（S 9 2 2 が Y E S 判定の場合）、メイン CPU 1 0 1 は、ナビデータ格納領域からナビデータを取得する（S 9 2 3）。次いで、メイン CPU 1 0 1 は、7 セグ L E D の各カソードに出力される押し順表示データを格納するための押し順表示データ格納領域のアドレスをセットする（S 9 2 4）。

【1 1 4 1】

次いで、メイン CPU 1 0 1 は、7 セグ表示データ生成処理を行う（S 9 2 5）。この処理では、メイン CPU 1 0 1 は、ナビデータに基づいて、押し順表示データ（7 セグ表示データ）を作成し、生成された押し順表示データを押し順表示データ格納領域に格納する。なお、7 セグ表示データ生成処理の詳細については、後述の図 1 6 1 を参照しながら後で説明する。

【1 1 4 2】

次いで、メイン CPU 1 0 1 は、クレジットカウンタの値を取得する（S 9 2 6）。次いで、メイン CPU 1 0 1 は、7 セグ L E D の各カソードに出力されるクレジット表示データを格納するためのクレジット表示データ格納領域のアドレスをセットする（S 9 2 7）。

【1 1 4 3】

次いで、メイン CPU 1 0 1 は、7 セグ表示データ生成処理を行う（S 9 2 8）。この処理では、メイン CPU 1 0 1 は、クレジットカウンタの値に基づいて、クレジット表示データ（7 セグ表示データ）を生成し、生成されたクレジット表示データをクレジット表示データ格納領域に格納する。なお、7 セグ表示データ生成処理の詳細については、後述の図 1 6 1 を参照しながら後で説明する。

【1 1 4 4】

次いで、メイン CPU 1 0 1 は、後述の 7 セグコモンカウンタの値を格納するための 7 セ

10

20

30

40

50

グコモンカウンタ格納領域のアドレスをセットする（Ｓ９２９）。次いで、メインＣＰＵ１０１は、７セグコモンカウンタの値に「１」を加算（＋１更新）する（Ｓ９３０）。なお、この処理において、更新後の７セグコモンカウンタの値が「８」となった場合には、メインＣＰＵ１０１は、７セグコモンカウンタの値に「０」をセットする。本実施形態では、７セグＬＥＤをダイナミック制御するため、８回周期で７セグコモンカウンタの値が更新される。

【１１４５】

次いで、メインＣＰＵ１０１は、７セグコモンカウンタの値に基づいて、コモン選択データを作成し、対象のカソードデータ格納領域（押し順表示データ格納領域又はクレジット表示データ格納領域内の対象格納領域）のアドレスをセットする（Ｓ９３１）。次いで、メインＣＰＵ１０１は、７セグＬＥＤのカソードにクリアデータを出力する（Ｓ９３２）。この処理は、７セグＬＥＤを一旦消灯して、残像の影響を無くすために行われる。

【１１４６】

次いで、メインＣＰＵ１０１は、対象のカソードデータ格納領域から７セグカソード出力データを取得してセットする（Ｓ９３３）。次いで、メインＣＰＵ１０１は、７セグコモンバックアップデータとコモン選択データとから、７セグコモン出力データを生成する（Ｓ９３４）。

【１１４７】

次いで、メインＣＰＵ１０１は、７セグコモンバックアップデータ及び７セグカソードバックアップデータにそれぞれ７セグコモン出力データ及び７セグカソード出力データを保存する（Ｓ９３５）。次いで、メインＣＰＵ１０１は、７セグカソード出力データ及び７セグコモン出力データを出力する（Ｓ９３６）。そして、Ｓ９３６の処理後、メインＣＰＵ１０１は、７セグＬＥＤ駆動処理を終了し、処理を割込処理（図１５８参照）中のＳ９０８の処理に移す。

【１１４８】

本実施形態では、上述のようにして７セグＬＥＤ駆動処理が行われる。なお、上述した７セグＬＥＤ駆動処理中のＳ９２３～Ｓ９２５の処理は、メインＣＰＵ１０１が、図１６０Ａのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。また、上述した７セグＬＥＤ駆動処理中のＳ９３１～Ｓ９３６の処理は、メインＣＰＵ１０１が、図１６０Ｂのソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【１１４９】

その中で、Ｓ９３６の各７セグ出力データの出力処理は、図１６０Ｂに示すように、一つのソースコード「ＬＤ（ｃＰＡ＿ＳＥＧＣＯＭ），ＢＣ」により実行される。それゆえ、本実施形態の７セグＬＥＤ駆動処理では、２桁の７セグＬＥＤをダイナミック点灯制御する際に、７セグコモン出力（選択）データと、７セグカソード出力データとが同時に出力される。すなわち、指示モニタで押し順ナビを実施する際の７セグＬＥＤをダイナミック点灯制御、及び、２桁の７セグＬＥＤでクレジット情報を表示する際の７セグＬＥＤをダイナミック点灯制御では、７セグコモン出力（選択）データと、７セグカソード出力データとが同時に出力される。

【１１５０】

この場合、ソースプログラム上において、７セグＬＥＤのダイナミック点灯制御に必要な命令コード数を減らすことができる。それゆえ、本実施形態では、ソースプログラムの容量（メインＲＯＭ１０２の使用容量）を低減することができ、メインＲＯＭ１０２において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。また、本実施形態では、７セグＬＥＤ駆動処理を行う７セグ駆動回路（不図示）をカソードコモン回路で構成し、カソードで制御する例を説明したが、本発明はこれに限定されず、７セグ駆動回路をアノードコモン回路で構成し、アノードで７セグＬＥＤの制御を行ってもよい。

【１１５１】

また、上述した7セグLED駆動処理中のS923のナビデータの取得処理及びS924の押し表示データ格納領域のアドレスセット処理はいずれも、図160Aに示すように、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う「LDQ」命令（メインCPU101専用命令コード）により実行される。それゆえ、本実施形態の7セグLED駆動処理では、「LDQ」命令を用いることにより、直値により、メインROM102、メインRAM103やメモリーマップE/Oにアクセスすることができるので、ソースプログラム上において、アドレス設定に係る命令を省略することができ（アドレス設定に係る命令を別途設ける必要がなくなる）、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

10

【1152】

〔7セグ表示データ生成処理〕

次に、図161～図163を参照して、7セグLED駆動処理（図159参照）中のS925及びS928で行う7セグ表示データ生成処理について説明する。なお、図161は、7セグ表示データ生成処理の手順を示すフローチャートである。図162は、7セグ表示データ生成処理を実行するためのソースプログラムの一例を示す図である。また、図163は、7セグ表示データ生成処理のソースプログラム上で、実際に参照される7セグカソードテーブルの構成の一例を示す図である。

【1153】

なお、7セグLED駆動処理（図159参照）中のS925で行われる7セグ表示データ生成処理で生成される後述の「表示データ」は押し順表示データに対応し、7セグLED駆動処理（図159参照）中のS928で行われる7セグ表示データ生成処理で生成される後述の「表示データ」はクレジット表示データに対応する。

20

【1154】

まず、メインCPU101は、カソードデータ格納領域にセットされた表示データを「10」で除算し、その除算結果の商の値を、2桁の7セグLEDの上位桁の表示データとして取得し、除算結果の余の値を下位桁の表示データとして取得する（S941）。次いで、メインCPU101は、取得した上位桁の表示データに基づいて、上位桁表示を行うか否かを判別する（S942）。

30

【1155】

S942において、メインCPU101が、上位桁表示を行うと判別したとき（S942がYES判定の場合）、メインCPU101は、後述のS944の処理を行う。一方、S942において、メインCPU101が、上位桁表示を行わないと判別したとき（S942がNO判定の場合）、メインCPU101は、上位桁の表示無しをセットする（S943）。

【1156】

S943の処理後又はS942がYES判定の場合、メインCPU101は、7セグカソードテーブル（図163参照）を参照して、上位桁の表示データを取得する（S944）。次いで、メインCPU101は、上位桁の表示データ格納領域（不図示）に取得した上位桁の表示データを保存する（S945）。

40

【1157】

次いで、メインCPU101は、7セグカソードテーブル（図163参照）を参照して、下位桁の表示データを取得する（S946）。次いで、メインCPU101は、下位桁の表示データ格納領域（不図示）に取得した下位桁の表示データを保存する（S947）。

【1158】

そして、S947の処理後、メインCPU101は、7セグ表示データ生成処理を終了する。この際、実行した7セグ表示データ生成処理が7セグLED駆動処理（図158参照）中のS925の処理である場合には、メインCPU101は、処理を7セグLED駆動処理中のS926の処理に移す。一方、実行した7セグ表示データ生成処理が7セグLE

50

D駆動処理（図158参照）中のS928の処理である場合には、メインCPU101は、処理を7セグLED駆動処理中のS929の処理に移す。

【1159】

本実施形態では、上述のようにして7セグ表示データ生成処理が行われる。なお、上述した7セグ表示データ生成処理は、メインCPU101が、図162のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【1160】

〔タイマー更新処理〕

次に、図164及び図165を参照して、割込処理（図158参照）中のS908で行うタイマー更新処理について説明する。なお、図164は、タイマー更新処理の手順を示すフローチャートである。また、図165は、タイマー更新処理中の後述のS951～S954の処理を実行するためのソースプログラムの一例を示す図である。

【1161】

まず、メインCPU101は、HLレジスタに2バイトタイマー格納領域（不図示）の更新開始アドレスをセットし、Bレジスタに2バイトタイマー数をセットする（S951）。

【1162】

次いで、メインCPU101は、2バイトタイマー数とその下限値「0」とを比較し、2バイトタイマー数が下限値「0」より大きい場合には、2バイトタイマー数を1減算（－1更新）し、2バイトタイマー数が下限値「0」以下である場合には、2バイトタイマー数を「0」に保持する（S952）。さらに、S952の処理では、メインCPU101は、HLレジスタにセットされている2バイトタイマー格納領域の更新開始アドレスを2減算（－2更新）する。

【1163】

次いで、メインCPU101は、Bレジスタにセットされた2バイトタイマー数を1減算（－1更新）する（S953）。次いで、メインCPU101は、Bレジスタにセットされた2バイトタイマー数が「0」であるか否かを判別する（S954）。

【1164】

S954において、メインCPU101が、Bレジスタにセットされた2バイトタイマー数が「0」でないと判別したとき（S954がNO判定の場合）、メインCPU101は、処理をS952の処理に戻し、S952以降の処理を繰り返す。

【1165】

一方、S954において、メインCPU101が、Bレジスタにセットされた2バイトタイマー数が「0」であると判別したとき（S954がYES判定の場合）、メインCPU101は、HLレジスタに1バイトタイマー格納領域の更新開始アドレスをセットし、Bレジスタに1バイトタイマー数をセットする（S955）。

【1166】

次いで、メインCPU101は、1バイトタイマー数とその下限値「0」とを比較し、1バイトタイマー数が下限値「0」より大きい場合には、1バイトタイマー数を1減算（－1更新）し、1バイトタイマー数が下限値「0」以下である場合には、1バイトタイマー数を「0」に保持する（S956）。さらに、S956の処理では、メインCPU101は、HLレジスタにセットされている1バイトタイマー格納領域の更新開始アドレスを1減算（－1更新）する。

【1167】

次いで、メインCPU101は、Bレジスタにセットされた1バイトタイマー数を1減算（－1更新）する（S957）。次いで、メインCPU101は、Bレジスタにセットされた1バイトタイマー数が「0」であるか否かを判別する（S958）。

【1168】

S958において、メインCPU101が、Bレジスタにセットされた1バイトタイマー数が「0」でないと判別したとき（S958がNO判定の場合）、メインCPU101は、処理をS956の処理に戻し、S956以降の処理を繰り返す。

10

20

30

40

50

【1169】

一方、S958において、メインCPU101が、Bレジスタにセットされた1バイトタイマー数が「0」であると判別したとき（S958がYES判定の場合）、メインCPU101は、電磁カウンタ制御処理を行う（S959）。この処理では、メダルのIN/OUTを示す信号を外部集中端子板47に出力する際の出力制御処理が行われる。そして、S959の処理後、メインCPU101は、タイマー更新処理を終了し、処理を割込処理（図158参照）中のS909の処理に移す。

【1170】

本実施形態では、上述のようにしてタイマー更新処理が行われる。なお、上述したタイマー更新処理中のS951～S954の処理（2バイトタイマーの更新処理）は、メインCPU101が、図165のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。

【1171】

その中で、S952の処理（2バイトタイマーの更新処理）は、図165中の「DCPWLD」命令（所定の更新命令）により実行される。なお、「DCPWLD」命令は、メインCPU101専用命令コードである。

【1172】

ソースプログラム上において、例えば、ソースコード「DCPWLD（HL），n」が実行されると、HLレジスタで指定されたアドレスから2バイト分のメモリの内容（格納データ）と整数nとが比較され、2バイト分のメモリの内容が整数nより大きい場合には、2バイト分のメモリの内容が1減算され、2バイト分のメモリの内容が整数n以下である場合には、HLレジスタで指定されたアドレスから2バイト分のメモリに整数nが格納される。

【1173】

それゆえ、図165中のソースコード「DCPWLD（HL），0」では、HLレジスタで指定されたアドレスから2バイト分のメモリの内容（2バイトタイマー数）と整数「0」（下限値）とが比較され、2バイト分のメモリの内容が整数「0」より大きい場合には、2バイト分のメモリの内容が1減算され、2バイト分のメモリの内容が整数「0」以下である場合には、2バイト分のメモリの内容に「0」がセットされる。すなわち、現時点の2バイトタイマー数が「0」より大きい場合には、2バイトタイマーの更新処理が行われ、現時点の2バイトタイマー数が「0」以下であれば、2バイトタイマー数が「0」に保持される。

【1174】

上述のように、本実施形態のタイマー更新処理では、メインCPU101専用命令コードである「DCPWLD」命令により、タイマー数の更新（減算）処理及びタイマー数を「0」に保持する処理の両方を実行することができる。この場合、両処理を別個に実行するための命令コードを設ける必要がなくなる。また、タイマー数が「0」であるか否かを判別するための判断分岐命令コードも省略することができる。それゆえ、本実施形態では、ソースプログラムの容量（メインROM102の使用容量）を低減することができ、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。なお、本実施形態では、2バイトタイマーの更新処理においてのみ「DCPWLD」命令を使用する例を説明したが、本発明はこれに限定されず、1バイトタイマーの更新処理においても「DCPWLD」命令を使用してもよい。

【1175】

【試射試験信号制御処理（規定外）】

次に、図166を参照して、割込処理（図158参照）中のS911で行う試射試験信号制御処理について説明する。なお、図166は、試射試験信号制御処理の手順を示すフローチャートである。

【1176】

まず、メインCPU101は、メインRAM103のスタックエリアのアドレスを退避させる（S961）。次いで、メインCPU101は、スタックポインタ（SP）に規定外スタックエリアのアドレスをセットする（S962）。次いで、メインCPU101は、全レジスタのデータを退避させる（S963）。

【1177】

次いで、メインCPU101は、回胴制動信号生成処理を行う（S964）。この処理では、メインCPU101は、第2インターフェースポート等を介して試験機に出力される、各リールの回転制御信号（駆動信号）の生成及び出力処理を行う。なお、回胴制動信号生成処理の詳細については、後述の図167を参照しながら後で説明する。

【1178】

次いで、メインCPU101は、特賞信号制御処理を行う（S965）。この処理では、メインCPU101は、試験機に出力される、ボーナス（特賞）のON/OFF信号（試験信号）の出力処理を行う。なお、特賞信号制御処理の詳細については、後述の図168を参照しながら後で説明する。

【1179】

次いで、メインCPU101は、条件装置信号制御処理を行う（S966）。この処理では、メインCPU101は、条件装置信号制御フラグの状態に対応する制御信号の出力処理を行う。なお、条件装置信号制御処理の詳細については、後述の図169及び図170を参照しながら後で説明する。

【1180】

次いで、メインCPU101は、S963の処理で退避させた全レジスタのデータの復帰処理を行う（S967）。次いで、メインCPU101は、S961の処理で退避させたスタックエリアのアドレスをスタックポインタ（SP）にセットする（S968）。そして、S968の処理後、メインCPU101は、試射試験信号制御処理を終了し、処理を割込処理（図158参照）中のS912の処理に移す。

【1181】

〔回胴制動信号生成処理〕

次に、図167を参照して、試射試験信号制御処理（図166参照）中のS964で行う回胴制動信号生成処理について説明する。なお、図167は、回胴制動信号生成処理の手順を示すフローチャートである。

【1182】

まず、メインCPU101は、規定外作業領域に回胴制御データ格納領域（不図示）をセットする（S971）。次いで、メインCPU101は、リール数に「3」をセットし、回胴制動信号及びその生成状態（1バイトデータ）をクリアする（S972）。

【1183】

次いで、メインCPU101は、回胴制御データが「停止中未満」のデータであるか否かを判別する（S973）。なお、ここでいう「停止中未満」の回胴制御データとは、リールを停止するための回胴制御データ以外の回胴制御データ、すなわち、リールを回転駆動するための回胴制御データ（加速準備、加速中、定速待ち、定速中及び停止開始位置待ちのいずれかの状態）のことである。

【1184】

S973において、メインCPU101が、回胴制御データが「停止中未満」のデータであると判別したとき（S973がYES判定の場合）、メインCPU101は、後述のS975の処理を行う。一方、S973において、メインCPU101が、回胴制御データが「停止中未満」のデータでないと判別したとき（S973がNO判定の場合）、メインCPU101は、回胴制御データが「静定ホールド制御終了」のデータであるか否かを判別する（S974）。なお、ここでいう「静定ホールド制御終了」の回胴制御データとは、リールの全相全停止状態を示す回胴制御データのことである。

【1185】

S974において、メインCPU101が、回胴制御データが「静定ホールド制御終了」

10

20

30

40

50

のデータであると判別したとき（S 9 7 4 が Y E S 判定の場合）、メイン C P U 1 0 1 は、後述の S 9 7 6 の処理を行う。一方、S 9 7 4 において、メイン C P U 1 0 1 が、回胴制御データが「静定ホールド制御終了」のデータでないと判別したとき（S 9 7 4 が N O 判定の場合）、又は、S 9 7 3 が Y E S 判定の場合、メイン C P U 1 0 1 は、回胴制動信号の生成状態（1 バイトデータ）のビット 3 をオン状態（「1」）にする（S 9 7 5）。

【1 1 8 6】

S 9 7 5 の処理後又は S 9 7 4 が N O 判定の場合、メイン C P U 1 0 1 は、生成状態の各ビットのデータを 1 ビット分、右（ビット 7 からビット 0 に向かう方向）にシフトする（S 9 7 6）。次いで、メイン C P U 1 0 1 は、回胴制御データ格納領域のアドレスを次の制御対象のリールのアドレスに更新する（S 9 7 7）。

10

【1 1 8 7】

次いで、メイン C P U 1 0 1 は、リール数を 1 減算する（S 9 7 8）。次いで、メイン C P U 1 0 1 は、リール数が「0」であるか否かを判別する（S 9 7 9）。

【1 1 8 8】

S 9 7 9 において、メイン C P U 1 0 1 が、リール数が「0」でないと判別したとき（S 9 7 9 が N O 判定の場合）、メイン C P U 1 0 1 は、処理を S 9 7 3 の処理に戻し、S 9 7 3 以降の処理を繰り返す。

【1 1 8 9】

一方、S 9 7 9 において、メイン C P U 1 0 1 が、リール数が「0」であると判別したとき（S 9 7 9 が Y E S 判定の場合）、メイン C P U 1 0 1 は、生成状態のデータを、回胴制動信号出力ポートを介して試験機用第 1 インターフェースボード 3 0 1（図 7 参照）へ出力する（S 9 8 0）。そして、S 9 8 0 の処理後、メイン C P U 1 0 1 は、回胴制動信号生成処理を終了し、処理を試射試験信号制御処理（図 1 6 6 参照）中の S 9 6 5 の処理に移す。

20

【1 1 9 0】

〔特賞信号制御処理〕

次に、図 1 6 8 を参照して、試射試験信号制御処理（図 1 6 6 参照）中の S 9 6 5 で行う特賞信号制御処理について説明する。なお、図 1 6 8 は、特賞信号制御処理の手順を示すフローチャートである。

【1 1 9 1】

まず、メイン C P U 1 0 1 は、遊技状態フラグ格納領域（図 3 2 参照）を参照して、遊技状態フラグを取得する（S 9 9 1）。次いで、メイン C P U 1 0 1 は、遊技状態が R B 遊技状態であるか否かを判別する（S 9 9 2）。

30

【1 1 9 2】

S 9 9 2 において、メイン C P U 1 0 1 が、遊技状態が R B 遊技状態であると判別したとき（S 9 9 2 が Y E S 判定の場合）、メイン C P U 1 0 1 は、試験信号用の R B 中信号ポートから O N 信号を試験機用第 1 インターフェースボード 3 0 1（図 7 参照）へ出力する（S 9 9 3）。一方、S 9 9 2 において、メイン C P U 1 0 1 が、遊技状態が R B 遊技状態でないと判別したとき（S 9 9 2 が N O 判定の場合）、メイン C P U 1 0 1 は、試験信号用の R B 中信号ポートから O F F 信号を試験機用第 1 インターフェースボード 3 0 1（図 7 参照）へ出力する（S 9 9 4）。

40

【1 1 9 3】

S 9 9 3 又は S 9 9 4 の処理後、メイン C P U 1 0 1 は、遊技状態フラグ格納領域（図 3 2 参照）を参照して、遊技状態が B B 遊技状態であるか否かを判別する（S 9 9 5）。

【1 1 9 4】

S 9 9 5 において、メイン C P U 1 0 1 が、遊技状態が B B 遊技状態であると判別したとき（S 9 9 5 が Y E S 判定の場合）、メイン C P U 1 0 1 は、試験信号用の B B 中信号ポートから O N 信号を試験機用第 1 インターフェースボード 3 0 1（図 7 参照）へ出力する（S 9 9 6）。一方、S 9 9 5 において、メイン C P U 1 0 1 が、遊技状態が B B 遊技状態でないと判別したとき（S 9 9 5 が N O 判定の場合）、メイン C P U 1 0 1 は、試験信

50

号用の B B 中信号ポートから O F F 信号を試験機用第 1 インターフェースボード 3 0 1 (図 7 参照) へ出力する (S 9 9 7) 。

【 1 1 9 5 】

そして、 S 9 9 6 又は S 9 9 7 の処理後、メイン C P U 1 0 1 は、特賞信号制御処理を終了し、処理を試射試験信号制御処理 (図 1 6 6 参照) 中の S 9 6 6 の処理に移す。

【 1 1 9 6 】

〔条件装置信号制御処理〕

次に、図 1 6 9 及び図 1 7 0 を参照して、試射試験信号制御処理 (図 1 6 6 参照) 中の S 9 6 6 で行う条件装置信号制御処理について説明する。なお、図 1 6 9 及び図 1 7 0 は、条件装置信号制御処理の手順を示すフローチャートである。

【 1 1 9 7 】

まず、メイン C P U 1 0 1 は、条件装置信号制御フラグが初期状態であるか否かを判別する (S 1 0 0 1) 。 S 1 0 0 1 において、メイン C P U 1 0 1 が、条件装置信号制御フラグが初期状態であると判別したとき (S 1 0 0 1 が Y E S 判定の場合) 、メイン C P U 1 0 1 は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理 (図 1 6 6 参照) 中の S 9 6 7 の処理に移す。

【 1 1 9 8 】

一方、 S 1 0 0 1 において、メイン C P U 1 0 1 が、条件装置信号制御フラグが初期状態でないと判別したとき (S 1 0 0 1 が N O 判定の場合) 、メイン C P U 1 0 1 は、条件装置信号制御フラグが再遊技状態識別信号のオン状態を示すものであるか否かを判別する (S 1 0 0 2) 。

【 1 1 9 9 】

S 1 0 0 2 において、メイン C P U 1 0 1 が、条件装置信号制御フラグが再遊技状態識別信号のオン状態を示すものであると判別したとき (S 1 0 0 2 が Y E S 判定の場合) 、メイン C P U 1 0 1 は、条件装置信号制御状態に役物条件装置信号のオン状態をセットする (S 1 0 0 3) 。次いで、メイン C P U 1 0 1 は、条件装置 1 ~ 6 信号ポートから R T 状態の情報を試験機用第 1 インターフェースボード 3 0 1 (図 7 参照) へ出力する (S 1 0 0 4) 。そして、 S 1 0 0 4 の処理後、メイン C P U 1 0 1 は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理 (図 1 6 6 参照) 中の S 9 6 7 の処理に移す。

【 1 2 0 0 】

一方、 S 1 0 0 2 において、メイン C P U 1 0 1 が、条件装置信号制御フラグが再遊技状態識別信号のオン状態を示すものでないと判別したとき (S 1 0 0 2 が N O 判定の場合) 、メイン C P U 1 0 1 は、条件装置信号制御フラグが再遊技状態識別信号のオフ状態を示すものであるか否かを判別する (S 1 0 0 5) 。

【 1 2 0 1 】

S 1 0 0 5 において、メイン C P U 1 0 1 が、条件装置信号制御フラグが再遊技状態識別信号のオフ状態を示すものであると判別したとき (S 1 0 0 5 が Y E S 判定の場合) 、メイン C P U 1 0 1 は、条件装置 1 ~ 8 信号ポートから O F F 信号を試験機用第 1 インターフェースボード 3 0 1 (図 7 参照) へ出力する (S 1 0 0 6) 。そして、 S 1 0 0 6 の処理後、メイン C P U 1 0 1 は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理 (図 1 6 6 参照) 中の S 9 6 7 の処理に移す。

【 1 2 0 2 】

一方、 S 1 0 0 5 において、メイン C P U 1 0 1 が、条件装置信号制御フラグが再遊技状態識別信号のオフ状態を示すものでないと判別したとき (S 1 0 0 5 が N O 判定の場合) 、メイン C P U 1 0 1 は、条件装置信号制御状態が役物条件装置信号のオン状態であるか否かを判別する (S 1 0 0 7) 。

【 1 2 0 3 】

S 1 0 0 7 において、メイン C P U 1 0 1 が、条件装置信号制御状態が役物条件装置信号のオン状態であると判別したとき (S 1 0 0 7 が Y E S 判定の場合) 、メイン C P U 1 0 1 は、条件装置 1 ~ 8 信号ポートから特賞当籤番号の情報を試験機用第 1 インターフェー

10

20

30

40

50

スボード301（図7参照）へ出力し、この際、条件装置8信号ポートからON信号を試験機用第1インターフェースボード301（図7参照）へ出力する（S1008）。次いで、メインCPU101は、条件装置信号出力待ちタイマーに所定の待ち時間（本実施形態では、24.58ms）をセットする（S1009）。次いで、メインCPU101は、条件装置信号制御状態に条件装置信号出力待ちの状態をセットする（S1010）。そして、S1010の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図166参照）中のS967の処理に移す。

【1204】

一方、S1007において、メインCPU101が、条件装置信号制御状態が役物条件装置信号のオン状態でないとは判断したとき（S1007がNO判定の場合）、メインCPU101は、条件装置信号制御状態が条件装置信号出力待ちの状態であるか否かを判断する（S1011）。

【1205】

S1011において、メインCPU101が、条件装置信号制御状態が条件装置信号出力待ちの状態であると判断したとき（S1011がYES判定の場合）、メインCPU101は、条件装置信号出力待ちタイマーの値が「0」であるか否かを判断する（S1012）。

【1206】

S1012において、メインCPU101が、条件装置信号出力待ちタイマーの値が「0」でないと判断したとき（S1012がNO判定の場合）、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図166参照）中のS967の処理に移す。一方、S1012において、メインCPU101が、条件装置信号出力待ちタイマーの値が「0」であると判断したとき（S1012がYES判定の場合）、メインCPU101は、条件装置信号制御状態に小役条件装置信号のオン状態又は条件装置信号のオフ状態をセットする（S1013）。そして、S1013の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図166参照）中のS967の処理に移す。

【1207】

ここで再度、S1011の処理に戻って、S1011において、メインCPU101が、条件装置信号制御状態が条件装置信号出力待ちの状態でないとは判断したとき（S1011がNO判定の場合）、メインCPU101は、条件装置信号制御状態が小役条件装置信号のオン状態であるか否かを判断する（S1014）。

【1208】

S1014において、メインCPU101が、条件装置信号制御状態が小役条件装置信号のオン状態であると判断したとき（S1014がYES判定の場合）、メインCPU101は、条件装置1～8信号ポートから小役当籤番号の情報を試験機用第1インターフェースボード301（図7参照）へ出力し、この際、条件装置7信号ポートからON信号を試験機用第1インターフェースボード301（図7参照）へ出力する（S1015）。次いで、条件装置信号出力待ちタイマーに所定の待ち時間（本実施形態では、24.58ms）をセットする（S1016）。次いで、メインCPU101は、条件装置信号制御状態に条件装置信号出力待ちの状態をセットする（S1017）。そして、S1017の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図166参照）中のS967の処理に移す。

【1209】

一方、S1014において、メインCPU101が、条件装置信号制御状態が小役条件装置信号のオン状態でないとは判断したとき（S1014がNO判定の場合）、メインCPU101は、条件装置1～8信号ポートからOFF信号を試験機用第1インターフェースボード301（図7参照）へ出力する（S1018）。そして、S1018の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図166参照）中のS967の処理に移す。

10

20

30

40

50

【1210】

＜副制御回路の動作説明＞

次に、図171～図173を参照して、副制御回路200のサブCPU201が、プログラムを用いて実行する各種処理の内容について説明する。

【1211】

〔サブ側ナビ制御処理〕

最初に、図171を参照して、サブ側ナビ制御処理について説明する。なお、図171は、サブ側ナビ制御処理の手順を示すフローチャートである。

【1212】

まず、サブCPU201は、ナビデータを取得したか否かを判定する（S1101）。サブCPU201は、主制御基板71から受信したスタートコマンドデータの中から主制御基板71で決定されたナビデータを取得する。それゆえ、S1101の処理では、サブCPU201は、受信したスタートコマンドデータの中にナビデータが含まれていたか否かを判定する。

10

【1213】

S1101において、サブCPU201が、ナビデータを取得したと判別したとき（S1101がYES判定の場合）、サブCPU201は、ナビデータに応じたサブ側ナビデータをセットする（S1102）。例えば、サブCPU201がナビデータ「4」を取得した場合、図63に示すように、サブ側ナビデータとして押し順「左、中、右」を報知するためのナビデータが、この処理でセットされる。この結果、メイン側及びサブ側の双方において停止操作の内容を報知することができる。そして、S1102の処理後、サブCPU201は、サブ側ナビ制御処理を終了する。

20

【1214】

一方、S1101において、サブCPU201が、ナビデータを取得していないと判別したとき（S1101がNO判定の場合）、サブCPU201は、ナビ（停止操作の報知）の必要があるか否かを判定する（S1103）。本実施形態では、サブCPU201は、例えば、主制御基板71においてフラグ変換抽籤が行われた場合や、主制御基板71において所定の役が内部当籤役として決定された場合に、ナビの必要があると判定する。なお、フラグ変換抽籤の結果や、内部当籤役の種別は、スタートコマンドデータに含まれている。それゆえ、S1103の処理では、サブCPU201は、スタートコマンドデータに含まれるこれらの各種情報に基づいて、ナビの必要があるか否かを判定する。

30

【1215】

S1103において、サブCPU201が、ナビの必要がないと判別したとき（S1103がNO判定の場合）、サブCPU201は、サブ側ナビ制御処理を終了する。

【1216】

一方、S1103において、サブCPU201が、ナビの必要があると判別したとき（S1103がYES判定の場合）、サブCPU201は、各種抽籤結果に応じたサブ側ナビデータをセットする（S1104）。例えば、内部当籤役「F__確チリリプ」が決定され、かつ、フラグ変換抽籤に当籤している場合、サブCPU201は、この処理において、略称「3連チリリプ」に係る図柄組合せを表示するためのナビデータ（例えば、順押しでチリ図柄を狙わせるナビデータ）をセットする。また、例えば、内部当籤役「F__確チリリプ」が決定され、かつ、フラグ変換抽籤に非当籤であった場合、サブCPU201は、この処理において、略称「リプレイ」に係る図柄組合せを表示するためのナビデータ（例えば、順押し以外の押し順を示すナビデータ）をセットする。これらの処理により、メイン側で停止操作の内容を報知しない場合であっても、サブ側単独で停止操作の内容を報知することができる。そして、S1104の処理後、サブCPU201は、サブ側ナビ制御処理を終了する。

40

【1217】

〔遊技者登録処理〕

次に、図172を参照して、遊技者登録処理について説明する。なお、図172は、遊技

50

者登録処理の手順を示すフローチャートである。

【1218】

まず、サブCPU201は、登録操作を受け付けたか否かを判別する（S1111）。例えば、サブ表示装置18のメニュー画面222（図5B参照）において登録ボタン222bの操作を受け付け、その場合に表示される登録画面（不図示）において所定の操作を受け付けると、サブCPU201は、登録操作を受け付けたと判定する。

【1219】

S1111において、サブCPU201が、登録操作を受け付けたと判別したとき（S1111がYES判定の場合）、サブCPU201は、遊技者登録状態をセットする（S1112）。なお、遊技者登録状態がセットされている状況では、サブCPU201は、サブ表示装置18に遊技情報画面223，224，225（図5C～5E参照）が表示可能となるようにサブ表示装置18の表示画面を制御する。そして、S1112の処理後、サブCPU201は、遊技者登録処理を終了する。

10

【1220】

一方、S1111において、サブCPU201が、登録操作を受け付けていないと判別したとき（S1111がNO判定の場合）、サブCPU201は、登録削除操作を受け付けたか否かを判別する（S1113）。例えば、サブ表示装置18の登録画面において特定の操作を受け付けると、サブCPU201は、登録削除操作を受け付けたと判定する。

【1221】

S1113において、サブCPU201が、登録削除操作を受け付けていないと判別したとき（S1113がNO判定の場合）、サブCPU201は、遊技者登録処理を終了する。

20

【1222】

一方、S1113において、サブCPU201が、登録削除操作を受け付けたと判別したとき（S1113がYES判定の場合）、サブCPU201は、遊技者登録状態をクリアする（S1114）。なお、遊技者登録状態がクリアされている状況では、サブCPU201は、サブ表示装置18に遊技情報画面223，224，225（図5C～5E参照）が表示不可能となるようにサブ表示装置18の表示画面を制御する。そして、S1114の処理後、サブCPU201は、遊技者登録処理を終了する。

【1223】

〔履歴管理処理〕

30

次に、図173を参照して、履歴管理処理について説明する。なお、図173は、履歴管理処理の手順を示すフローチャートである。

【1224】

まず、サブCPU201は、主制御基板71から受信した各種コマンドデータから遊技結果を取得する（S1121）。例えば、サブCPU201は、この処理において、スタートコマンドデータから内部当籤役として決定された役の種類を把握することができる。また、例えば、サブCPU201は、この処理において、入賞作動コマンドデータから表示された図柄組合せ（すなわち、内部当籤役として決定された役の入賞の有無）を把握することができる。さらに、例えば、サブCPU201は、この処理において、スタートコマンドデータなどから現在の遊技状態や遊技状態の移行状況を把握することができる。

40

【1225】

次いで、サブCPU201は、取得した遊技結果に基づいて、遊技履歴の更新処理を行う（S1122）。この処理により、サブCPU201は、各種コマンドデータから取得した遊技結果に基づいて、例えば、ボーナス回数、ART回数、ゲーム数（遊技回数）、CZ回数、CZ成功回数、それぞれの役の当籤回数及び当籤確率などの様々な遊技履歴を管理することができる。そして、S1122の処理後、サブCPU201は、履歴管理処理を終了する。

【1226】

<各種効果>

本実施形態のパチスロ1では、その遊技性において、次のような各種効果が得られる。

50

【1227】

〔CT中の継続期間の管理〕

本実施形態のパチスロ1では、通常ARTの継続期間を延長可能な上乗せチャンスゾーンとしてCTを設け、このCT中の内部当籤役（サブフラグ）に基づいてARTゲーム数の上乗せを行う。なお、CTでは、1セット8回の遊技が行われるが、CT中にARTゲーム数の上乗せを行うことができた場合には遊技回数の減算を行わずに、上乗せできない場合に限り遊技回数を減算する。それゆえ、遊技者にとってみれば、CTがいつまで続くかわからず、また、上乗せが行われている限りCTが終了することがないため、CT中の遊技の興趣を高めることができる。

【1228】

また、本実施形態において、CT中にサブフラグEX「3連チリリブ」に当籤（サブフラグ変換抽籤に当籤）して上乗せが行われた場合には、1セット8回のCT遊技も再セット（ストック）される。この場合、例えば、CTの遊技期間が終了直前になっても、サブフラグEX「3連チリリブ」当籤時には、上乗せが行われ、CTが初めから再開されることになるので、CT中の遊技に対して強い関心を抱くことになり、退屈することなく遊技を継続できるとともに、CT中の遊技の興趣を高めることができる。

【1229】

また、サブフラグEX「3連チリリブ」当籤時の上乗せは、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が内部当籤役として決定され、かつ、フラグ変換抽籤に当籤した場合に限り行われる。それゆえ、遊技者に対して過大な利益を与えてしまうことを防止することができ、遊技者と遊技店との間の利益のバランスをとることができる。なお、本実施形態では、CT中に、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定された際にはフラグ変換抽籤に必ず当籤する例（図54参照）を説明したが、本発明はこれに限定されず、フラグ変換抽籤の当籤確率は、遊技者及び遊技店間の利益のバランスに応じて適宜設定することができる。

【1230】

〔CT中の「3連チリリブ」当籤時の上乗せゲーム数〕

本実施形態のパチスロ1では、CT中にサブフラグ「3連チリリブ」に当籤し、該サブフラグ「3連チリリブ」に基づく上乗せが行われた回数が所定回数を超えると、1回の上乗せ抽籤あたりに当籤するARTの上乗せゲーム数が増加する（図55参照）。また、上述のように、ARTゲーム数の上乗せが行われている限り、CTは終了することなく、また、サブフラグEX「3連チリリブ」に当籤するとCTの再セットが行われる。それゆえ、遊技者からすると、CTが継続するほど1回（1ゲーム）当りの上乗せ量が増えることについての期待を持つことができ、CT中の興趣が向上する。さらに、1回（1ゲーム）当りの上乗せ量を増やす契機となる回数は、CTの1セット分の基本遊技回数（8回）よりも多い回数（9回以上）であるため、遊技者に対して過大な利益を与えてしまうことを防止でき、遊技者及び遊技店間の利益のバランスをとる（良好に保つ）ことができる。

【1231】

〔メイン側で行うボーナス報知〕

本実施形態のパチスロ1では、情報表示器6の指示モニタ（不図示）に、ボーナス役（BB役）に係る図柄組合せを表示させるための停止操作の情報に一義的に対応付けた数値「10」（又は「11」）を表示することにより、メイン側でボーナス報知を行う（図63参照）。しかしながら、通常、パチスロでは、有効ライン上に引き込む（停止表示する）図柄の優先順位が定められており、ボーナス役とその他の役とが重複して内部当籤役として決定されている場合、優先順位により、ボーナス役に係る図柄組合せを引き込めることもあれば、引き込めないこともある。

【1232】

例えば、本実施形態のパチスロ1において、ボーナス役と、内部当籤役「はずれ」、「F__特殊役1」、「F__特殊役2」及び「F__特殊役3」のいずれかとが重複して決定されている場合には、ボーナス役に係る図柄組合せを引き込むことができるが、ボーナス役と

10

20

30

40

50

、内部当籤役「はずれ」、「F__特殊役1」、「F__特殊役2」及び「F__特殊役3」以外の役とが重複して決定されている場合には、ボーナス役に係る図柄組合せを引き込むことができない。それゆえ、本実施形態において、ボーナス役と、内部当籤役「はずれ」、「F__特殊役1」、「F__特殊役2」及び「F__特殊役3」のいずれかが重複して決定されている場合に限り、指示モニタに数値「10」（又は「11」）を表示する。この場合、メイン側のナビ（ボーナス報知）を、ボーナス役を入賞させることのできる適切なタイミングで行うことができる。

【1233】

また、本実施形態のパチスロ1において、ボーナス役と、内部当籤役「はずれ」、「F__特殊役1」、「F__特殊役2」及び「F__特殊役3」のいずれかが重複して決定されたとしても、ボーナス役が最初に当籤してから所定回数の遊技が経過するまでは、指示モニタに数値「10」（又は「11」）を表示せず（ナビをせず）、所定回数の遊技が経過したことを条件に、指示モニタに数値「10」（又は「11」）を表示する。

【1234】

なお、例えば、ボーナス役に当籤したことを契機として複数回の遊技にわたり行われる演出（いわゆる連続演出）が行われた場合、この連続演出の最中に、指示モニタに数値「10」（又は「11」）を表示すると、連続演出の意味が薄れてしまい、興趣を損ねてしまう可能性がある。それゆえ、本実施形態のパチスロ1では、所定回数の遊技が経過するまでは、指示モニタによる表示を行わず、所定回数の遊技が経過した後に、指示モニタによる表示を行う。その結果、演出効果を損ねることなく、メイン側でのボーナス報知を行うことができる。

【1235】

〔メイン側及びサブ側の両方で行う報知とサブ側単独で行う報知〕

本実施形態のパチスロ1では、停止操作の態様（押し順）に応じて表示される図柄組合せが異なる役を複数種類設けるとともに（図24参照）、これら複数種類の役には、表示される図柄組合せによって異なる特典が付与される役と、表示される図柄組合せが異なったとしても同一の特典が付与される役とが含まれる。

【1236】

例えば、略称「ベル」に係る図柄組合せが表示された場合と略称「ベルこぼし目」に係る図柄組合せが表示された場合とでは払い出されるメダルの枚数が異なり、これらの役は、表示される図柄組合せによって異なる特典が付与される役である。また、略称「リプレイ」に係る図柄組合せが表示された場合と略称「RT2移行リブ」に係る図柄組合せが表示された場合とでは再遊技の作動に加えてRT状態の移行が行われるか否かが異なるため、内部当籤役「F__3択ベル__1st」や「F__維持リブ__1st」もまた、表示される図柄組合せによって異なる特典が付与される役である。

【1237】

一方、略称「3連チリリブ」に係る図柄組合せが表示された場合と略称「リプレイ」に係る図柄組合せが表示された場合とでは、両者ともに再遊技の作動が行われるだけである。それゆえ、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」は、表示される図柄組合せが異なったとしても同一の特典が付与される役である。なお、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」は、上述のように、フラグ変換抽籤の結果に応じて付与する特典が異なる役であるが、表示される図柄組合せが付与する特典に影響を与える役ではない。

【1238】

本実施形態のパチスロ1では、表示される図柄組合せによって異なる特典が付与される役に対して、メイン側及びサブ側の両方で報知を行うが、表示される図柄組合せが異なったとしても同一の特典が付与される役に対しては、メイン側の指示モニタでは報知を行わずに、サブ側の表示装置11のみで報知を行う。このような報知機能を設けることにより、特典に影響する報知は、特典を管理するメイン側の指示モニタで適切に行いつつ、特典に影響しない報知は、サブ側の表示装置11で多様性のあるナビで行うことができる。

【1239】

〔遊技履歴の表示機能〕

本実施形態のパチスロ1では、表示装置11（プロジェクタ機構211及び表示ユニット212）とは別にサブ表示装置18を設け、このサブ表示装置18により遊技者に役立つ様々な情報を表示する。例えば、図5A～5Eに示すように、概要遊技履歴を表すトップ画面221、パチスロ1に対する様々な操作が可能なメニュー画面222、詳細遊技履歴を表す遊技情報画面223、224、225をサブ表示装置18に表示することができる。

【1240】

また、本実施形態のパチスロ1は、サブ表示装置18を介して、遊技を行う遊技者を登録可能にする機能、及び、登録された遊技者に対して固有のサービスを提供する機能を備える。例えば、本実施形態では、遊技者の登録を受け付けていない場合には、詳細遊技履歴を表す遊技情報画面223、224、225をサブ表示装置18に表示不可能にするが、遊技者の登録を受け付けている場合には、詳細遊技履歴を表す遊技情報画面223、224、225をサブ表示装置18に表示可能にする。より詳細な遊技履歴を確認できるようにすることは、遊技者の利便性の向上につながるもので、本実施形態では、遊技者の登録を受け付けている場合に、利便性を向上させることができる。

【1241】

また、本実施形態のパチスロ1では、サブ表示装置18は、演出を行う表示装置11とは別体に設けられる。サブ表示装置18は、液晶中継基板87を介して副制御基板72（サブCPU201）により制御される。また、表示装置11を構成する表示ユニット212は、役物中継基板（不図示）を介して副制御基板72（サブCPU201）により制御される。それゆえ、本実施形態では、サブ表示装置18を、表示装置11とは別個に制御することができる。具体的には、遊技中（すなわち、表示装置11による演出の実行中）であっても、サブ表示装置18の表示画面を切り替えることができる。それゆえ、遊技中であっても、表示装置11により実行されている演出を邪魔することなく、サブ表示装置18の表示を切り替えることにより、遊技者は様々な情報を取得することができる。

【1242】

また、本実施形態のパチスロ1の表示装置11では、プロジェクタ機構211からの照射光の照射により映像を出現させる複数のスクリーン機構（表示ユニット212）を切り替えることにより、平面状の映像表示を用いた演出、奥行き感（立体感）のある映像表示を用いた演出、及び、湾曲した映像表示を用いた演出を実行する場合、演出効果を著しく高めることができる。しかしながら、このような情報の表示形態は、演出中に遊技履歴などの演出とは関係ない情報を表示することには適さない。それゆえ、本実施形態のパチスロ1では、表示装置11とは別個に設けられたサブ表示装置18に演出とは関係ない情報を表示することができるので、演出効果を損なうことなく、かつ、遊技履歴などの各種情報を適切に表示することができる。

【1243】

ところで、一般的なパチスロでは、遊技者側から見て、台座部13の右側にメダル投入口14が設けられ、台座部13の左側にベットボタン15a、15bやスタートレバー16が設けられる。それゆえ、通常、遊技を進行させる際、遊技者は台座部13の右側又は左側（側方）の操作部を操作することになる。

【1244】

それに対して、本実施形態のパチスロ1では、台座部13から略垂直に立設する面の側方（左側）にサブ表示装置18を設け、このサブ表示装置18の画面上にサブ表示装置18の表示画面を切り替えるためのタッチセンサ19が設けられる。それゆえ、本実施形態では、遊技中に遊技者の手が位置する場所にサブ表示装置18やその表示を制御する入力装置（タッチセンサ19）が設けられることになるので、遊技者の操作性を向上させることができる。特に、本実施形態のように、タッチセンサ19付きのサブ表示装置18を、台座部13の水平面から立設する面に設けた場合には、遊技者は、台座部13に自身の手を置きながら、サブ表示装置18を操作することができる。この場合、遊技者の操作性が向

10

20

30

40

50

上するだけでなく、操作に伴う遊技者の疲労も軽減することができ、この結果、稼働率の向上も期待することができる。

【1245】

〔規定外ROM領域及び規定外RAM領域〕

本実施形態のパチスロ1では、図12Bに示すように、遊技者により実施される遊技の遊技性に直接関与しない各種処理（遊技性に影響を与えない各種処理）に使用される各種プログラム及び各種データ（テーブル）を、メインROM102内において、遊技用ROM領域とは異なるアドレスに配置された規定外ROM領域に格納する。

【1246】

このようなメインROM102の構成では、従来の規則上では、プログラム等の配置不可とされていたROM領域（規定外ROM領域）に、遊技者が実際に行う遊技そのものに不要なプログラム及びデータを配置することができる。それゆえ、本実施形態では、主制御基板71のメインROM102内において、遊技用ROM領域の容量の圧迫を回避することができるとともに、メインROM102内におけるプログラム及びテーブルの拡張性を高めることができる。

【1247】

〔電源投入（リセット割込）時処理により得られる効果〕

本実施形態のパチスロ1の電源投入（リセット割込）時処理では、図64に示すように、電源復帰直後（サムチェック前）に最初の1.1172ms周期の割込処理を行い（S7及びS8）、主制御回路90から副制御回路200に無操作コマンドが送信される。このように電源復帰直後に割込処理を許可することにより、電源復帰後、最短時間で無操作コマンドが送信され、主制御回路90及び副制御回路200間の通信接続を確立することができ、主制御回路90及び副制御回路200間の通信動作を安定化させることができる。

【1248】

また、電源復帰直後に送信される無操作コマンドを構成する通信パラメータ1～5には、電源復帰時に、それぞれLレジスタ、Hレジスタ、Eレジスタ、Dレジスタ及びCレジスタに格納されているデータがセットされる。それゆえ、本実施形態では、電源復帰直後の割込処理で送信される無操作コマンドのサム値（BCC）を、電源復帰毎に異ならせることができ、ゴト等の不正行為を抑制することができる。

【1249】

さらに、電源投入（リセット割込）時処理中のS13の処理において行われる、エラーコード「rr」を情報表示器6内の2桁の7セグLEDに表示する際の制御は、一つの「LDW」命令（所定の読み出し命令）により実行され、2桁の7セグLEDへの7セグコモン出力（選択）データの出力動作と7セグカソード出力データの出力動作とが同時に行われる（図65C参照）。すなわち、本実施形態のパチスロ1では、電源投入（リセット割込）時処理において、2桁の7セグLEDをダイナミック点灯制御する際に、7セグコモン出力（選択）データと、7セグカソード出力データとが同時に出力される。

【1250】

この場合、ソースプログラム上において、7セグLEDのダイナミック点灯制御に必要な命令コード数を減らすことができる。それゆえ、本実施形態では、ソースプログラムの容量（メインROM102の使用容量）を低減することができ、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1251】

〔遊技復帰処理で得られる効果〕

本実施形態のパチスロ1の遊技復帰処理では、図66に示すように、電断発生時の各ポートの入出力状態を電源復帰時に担保するとともに、電断時に回胴回転中の場合には、電源復帰時にリール制御管理情報を取得してリールの再回転開始に必要な処理も行う（S25～S32参照）。それゆえ、本実施形態では、リール回転中の電断から復帰したときであっても、安定して、リールの再回転制御を行うことが可能となり、遊技者に不快感を与え

10

20

30

40

50

ることが無くなる。

【1252】

また、本実施形態のパチスロ1は、上述のように、遊技機用のセキュリティ機能付きマイクロプロセッサ91を備える。そして、このマイクロプロセッサ91には、ソースプログラム上において規定可能な該マイクロプロセッサ91に特有の命令コード（メインCPU101専用命令コード）が各種設けられており、このメインCPU101専用命令コードを各種処理において用いることにより、処理の効率化やプログラム容量の削減などを可能にしている。

【1253】

例えば、遊技復帰処理では、図67に示すように、ソースプログラム上において、メインCPU101専用命令コードの一つである「LDQ」命令が用いられる。「LDQ」命令は、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う命令コードであり、上述のように、直値でメインROM102、メインRAM103やメモリーマップE/Oにアクセスすることができる。この場合、アドレス設定に係る命令コードを省略することができ、その分、遊技復帰処理のソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1254】

〔設定変更確認処理で得られる効果〕

本実施形態のパチスロ1の設定変更確認処理では、図69Aに示すように、ソースプログラム上において、メインCPU101専用命令コードである、「BITQ」命令及び「SETQ」命令（所定の命令）が用いられる。「BITQ」命令及び「SETQ」命令はいずれも、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う命令コードであり、これらの命令コードを使用した場合、上述のように、直値でメインRAM103やメモリーマップE/Oにアクセスすることができる。この場合、アドレス設定に係る命令コードを省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102の空き容量を増やすことが可能となるとともに、処理の高速化も図ることができる。

【1255】

また、設定変更確認処理中のS46の設定変更／設定確認開始時及びS57の設定変更／設定確認終了時で行う設定変更コマンド（初期化コマンド）の生成格納処理は、図69A及び69Bに示すように、ソースプログラム上において、メインCPU101専用命令コードである「CALLF」命令により実行される。そして、S46の「CALLF」命令で指定するジャンプ先のアドレスは、S57の「CALLF」命令で指定するジャンプ先のアドレスと同じである。すなわち、本実施形態では、設定変更時（遊技機起動時）、設定確認開始時（通常動作中）及び設定確認終了時に副制御回路200に送信する設定変更コマンド（初期化コマンド）の生成格納処理を実行するためのソースプログラムが、互いに同じであり、S46及びS57の両処理において、そのソースプログラムが共有化（モジュール化）されている。

【1256】

この場合、S46及びS57の両処理において、それぞれ別個に設定変更コマンドの生成格納処理のソースプログラムを設ける必要が無くなるので、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1257】

〔設定変更コマンド生成格納処理で得られる効果〕

本実施形態のパチスロ1の設定変更コマンド生成格納処理では、図70に示すように、設定値が通信パラメータ3としてEレジスタに格納され、RT情報が通信パラメータ5とし

10

20

30

40

50

てCレジスタに格納される。すなわち、設定変更コマンド（初期化コマンド）を構成する通信パラメータ1～5のうち、通信パラメータ3及び5は副制御回路200側で使用（解析）される通信パラメータ（使用パラメータ）であり、これらの通信パラメータには新たな情報がセットされる。一方、設定変更コマンド（初期化コマンド）を構成するその他の通信パラメータ1, 2及び4は、副制御回路200側で使用（解析）されない通信パラメータ（未使用パラメータ）であり、通信パラメータ1, 2及び4に対しては、現時点でLレジスタ、Hレジスタ及びDレジスタにそれぞれ格納されている値がセットされる。それゆえ、設定変更コマンド（初期化コマンド）送信時における通信パラメータ1, 2及び4の値は不定値となる。この場合、設定変更コマンドのサム値（BCC）を送信毎に不定値にすることができ、ゴト等の不正行為を抑制することができる。

10

【1258】

[通信データ格納処理で得られる効果]

本実施形態のパチスロ1の通信データ格納処理では、図72に示すように、Aレジスタに格納されたデータが通信コマンドの種別データとしてセットされ、Lレジスタ、Hレジスタ、Eレジスタ、Dレジスタ及びCレジスタに格納されたデータがそれぞれ通信コマンドの通信パラメータ1～5としてセットされ、Bレジスタに格納されたデータが通信コマンドの遊技状態フラグデータとしてセットされる。すなわち、本実施形態では、1パケット（8バイト）の通信データ（コマンドデータ）を作成する際に、各種パラメータをレジスタから転送して通信データ一時格納領域（通信バッファ）に格納する。

20

【1259】

この場合、未使用パラメータを含むコマンドデータを作成した時には、作成時毎に、未使用パラメータの値が不定値となる。すなわち、未使用パラメータを含むコマンドデータでは、同じ種別のコマンドデータあり、かつ、使用パラメータの値が同一であっても、コマンド作成毎に、コマンドデータのサム値（BCCデータ）が可変可能となる。それゆえ、本実施形態では、未使用パラメータを不定値とすることにより、通信データの解析を困難にしてゴト等の不正行為を抑止することができるとともに、不必要なゴト対策処理を加える必要がないため、ゴト対策処理の追加による、主制御回路90のプログラム容量の圧迫を抑制することができる。

【1260】

[通信データポインタ更新処理で得られる効果]

本実施形態のパチスロ1の通信データポインタ更新処理では、図75Aに示すように、ソースプログラム上において、メインCPU101専用命令コードである、「ICPLD」命令が用いられる。

30

【1261】

通信データポインタ更新処理において、「ICPLD」命令は、送信バッファの上限判定命令と、判断分岐命令とが一体になっている命令コードであるので、各命令処理を別個に実行するための命令コードを設ける必要がなくなる。それゆえ、「ICPLD」命令を用いることにより、通信データポインタ更新処理のソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

40

【1262】

[チェックサム生成処理及びサムチェック処理で得られる効果]

本実施形態のパチスロ1において、電断時に行われるチェックサム生成処理（規定外）では、図77に示すように、メインRAM103のデータを順次加算することにより、チェックサムが算出される。一方、電源投入時（電源復帰時）に行われるサムチェック処理（規定外）では、図79に示すように、電断発生時に生成されたチェックサムの値を、電源復帰時のメインRAM103に格納されたデータで順次減算し、最終的な減算結果が「0」であるか否かに基づいて、異常の発生の有無を判定する。すなわち、本実施形態では、電断発生時のチェックサムの生成処理は加算方式で行われ、電源復帰時のチェックサムの

50

判定処理は減算方式で行われる。

【1263】

このようなチェックサムの生成処理及び判定処理を採用した場合、電源復帰時に再度チェックサムを生成して、該チェックサムを電断発生時のチェックサムと照合する処理が不要となる。この場合、ソースプログラム上において、照合命令コードを省略することができ、ソースプログラムの容量を低減することができる。この結果、本実施形態では、メインROM102において、照合命令コードの省略分に対応する空き容量を確保する（増やす）ことができ、増えた空き容量を活用して、遊技性を高めることが可能になる。

【1264】

〔メダル受付・スタートチェック処理で得られる効果〕

本実施形態のパチスロ1のメダル受付・スタートチェック処理では、図83に示すように、設定変更確認処理（S233の処理）が行われるが、この処理は、遊技状態に関係なく実行される。それゆえ、本実施形態では、遊技状態がボーナス状態（特賞作動状態）であっても、設定値及びホールメニュー（各種履歴データ（エラー、電断履歴等））を確認することができ、ゴト等の不正行為を抑制することができる。

【1265】

〔メダル投入処理で得られる効果〕

本実施形態のパチスロ1のメダル投入処理では、図85に示すように、S244の処理において、メダル投入枚数表示用のLED点灯データが、テーブルを参照したループ処理でなく、演算処理により生成される。具体的には、図86に示すソースプログラム中の一連のソースコード「LD A, L」～「OR L」が順次実行されるにより、メダル投入枚数表示用のLED点灯データが生成される。

【1266】

メダル投入枚数表示用のLED点灯データを演算処理により生成した場合、メインROM102のテーブル領域の空き容量を増やすことができるとともに、プログラムの容量増を最小限に抑えることができる。すなわち、本実施形態のメダル投入処理では、メダル投入LED表示の処理を効率化することができるとともに、メインROM102の空き容量を確保し（増やし）、増えた空き領域を活用して、遊技性を高めることができる。

【1267】

〔メダル投入チェック処理で得られる効果〕

本実施形態のパチスロ1のメダル投入チェック処理（図87参照）において、S257のメダルセンサ入力状態の正常変化値の生成処理は、テーブルを参照して取得する処理ではなく、演算処理により行われる。具体的には、図88に示すソースプログラム中のソースコード「RLA」及び「AND cBX_MDISW」が順次実行されることにより、メダルセンサ入力状態正常変化値が算出される。

【1268】

メダルセンサ入力状態の変化態様の検知処理をテーブル参照処理から演算処理に変更することにより、メインROM102のテーブル格納領域の空き容量を増やすことができるとともに、プログラムの容量増を最小限に抑えることができる。それゆえ、上述した処理手法を採用することにより、メダル投入センサ状態の変化態様の検知処理を効率化することができるとともに、メインROM102において増えた空き容量を活用して、遊技性を高めることができる。

【1269】

〔内部抽籤処理で得られる効果〕

本実施形態のパチスロ1の内部抽籤処理（図92参照）において、S305の判定データの取得処理は、図93A中のソースコード「LDIN AC, (HL)」により実行される。この「LDIN」命令の実行により、S305の処理では、Aレジスタに、判定データ（「抽籤値選択テーブルor抽籤計数テーブル」の値）が格納され、Cレジスタに当り要求フラグステータス（「特賞当籤番号+小役当籤番号」の値）が格納される。また、「LDIN」命令の実行により、HLレジスタにセットされているアドレスが+2更新（2

10

20

30

40

50

加算)される。

【1270】

すなわち、内部抽籤処理中のS305の判定データの取得処理では、一つの命令コード(「LDIN」命令)により、データのロード処理及びアドレスの更新処理の両方を行うことができる。この場合、ソースプログラム上において、アドレス設定に係る命令コードを省略することができ、その分、ソースプログラムの容量(メインROM102の使用容量)を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する(増大させる)ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1271】

また、内部抽籤処理のS309の設定値データ(0~5のいずれか)の加算処理は、メインCPU101が図93B中のソースコード「MUL A, 6」及び「ADDQ A, (.LOW, wWAVENUM)」をこの順で実行することにより行われる。

【1272】

「MUL」命令は、メインCPU101専用の乗算処理の命令コードであり、この命令の実行は、マイクロプロセッサ91に含まれる演算回路107(図9参照)により実行される。すなわち、本実施形態のパチスロ1では、ソースプログラム上における乗算処理及び除算処理を実行するための演算専用回路(演算回路107)が設けられているので、乗算処理及び除算処理の効率化を図ることができる。

【1273】

また、「ADDQ」命令(所定の加算命令)は、メインCPU101専用命令コードであり、Qレジスタ(拡張レジスタ)を用いてアドレス指定を行う命令コードである。そして、この「ADDQ」命令を用いれば、直値により、メインROM102、メインRAM103やメモリマップE/Oにアクセスすることができる。それゆえ、「ADDQ」命令の使用により、内部抽籤処理のソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量(メインROM102の使用容量)を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する(増大させる)ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1274】

〔図柄設定処理で得られる効果〕

本実施形態のパチスロ1の図柄設定処理(図97参照)において、S330の圧縮データ格納処理は、メインCPU101が図99中のソースコード「CALLF SB__BTEP__00」を実行することにより行われる。「CALLF」命令は、上述のようにメインCPU101専用の2バイト命令コードであり、図99中のソースコード「CALLF SB__BTEP__00」が実行されると、「SB__BTEP__00」で指定されているアドレスに、処理をジャンプさせ、圧縮データ格納処理が開始される。

【1275】

また、図柄設定処理中のS329の当り要求フラグ格納領域のアドレスのセット処理は、メインCPU101が図99中のソースコード「LDQ DE, .LOW, wWAVEBIT」を実行することにより行われる。すなわち、S329の処理は、Qレジスタ(拡張レジスタ)を用いたメインCPU101専用の「LDQ」命令により行われる。この場合、図柄設定処理のソースプログラム上において、アドレス設定に係る命令コードを省略することができ、その分、図柄設定処理のソースプログラムの容量(メインROM102の使用容量)を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する(増大させる)ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1276】

また、本実施形態では、上述した図柄設定処理中のS324~S330で説明した処理手順で入賞に係るデータの圧縮・展開処理を行い、かつ、その処理の中で上述したメインC

10

20

30

40

50

P U 1 0 1 専用命令コードを用いることにより、入賞に係るデータの圧縮・展開処理の効率化を図ることができるとともに、限られたメイン R A M 1 0 3 の容量を有効活用することができる。

【 1 2 7 7 】

〔サブフラグ変換処理で得られる効果〕

本実施形態のパチスロ 1 において、サブフラグ変換処理のソースプログラム上で実際に参照される、図 1 0 7 に示すサブフラグ変換テーブルでは、各サブフラグに対してサブフラグ変換制御データ（制御ステータス）が対応付けられている。この際、同種のサブフラグに対しては、同じサブフラグ変換制御データ（制御ステータス）が対応付けられている。

【 1 2 7 8 】

例えば、サブフラグ「3 連チリリプ A」及び「3 連チリリプ B」に対しては、サブフラグ変換制御データ（制御ステータス）「0 0 0 0 0 0 1 1 B」が共通して割り付けられている。そして、内部当籤役（サブフラグ）をサブフラグ E X に変換する際のフラグ変換抽籤処理では、サブフラグに対応付けられたサブフラグ変換制御データ（制御ステータス）に基づいて、抽籤が行われる。

【 1 2 7 9 】

このように、メイン側で管理するサブフラグ変換テーブルにおいて、同種の内部当籤役（サブフラグ）に対して共通のサブフラグ変換制御データを設けることにより、該変換テーブルの汎用性が高くなり、機種変更に伴う変換プログラムの変更も軽微な変更で対応可能となるので、開発コストの増大を抑制することができる。

【 1 2 8 0 】

〔ナビセット処理で得られる効果〕

本実施形態のパチスロ 1 のナビセット処理では、図 1 0 9 に示すように、ソースプログラム上において、Q レジスタ（拡張レジスタ）を用いてアドレス指定を行う「L D Q」命令（メイン C P U 1 0 1 専用命令コード）が用いられる。

【 1 2 8 1 】

それゆえ、本実施形態のナビセット処理では、ソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量（メイン R O M 1 0 2 の使用容量）を低減することができる。この結果、本実施形態では、メイン R O M 1 0 2 において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【 1 2 8 2 】

〔テーブルデータ取得処理で得られる効果〕

本実施形態のパチスロ 1 のテーブルデータ取得処理（図 1 2 0 参照）において、S 5 8 1 ～ S 5 8 4 の 1 段階目のテーブルデータ取得処理では、C T 中 C T 当籤抽籤テーブル（図 1 2 2 参照）中の当籤役別テーブル選択相対テーブルが参照される。そして、この 1 段階目のテーブルデータ取得処理で参照する当籤役別テーブル選択相対テーブルでは、内部当籤役（サブフラグ D）の種別毎に設けられた選択値により C T 抽籤の「ハズレ」が設定されているので、抽籤テーブルに「ハズレ」役の抽籤値を規定する必要がなくなる。それゆえ、本実施形態では、C T 中 C T 当籤抽籤テーブルにおいて、「ハズレ」役の抽籤値データを格納する必要が無くなり、メイン R O M 1 0 2 のテーブル領域の容量を節約することができる。

【 1 2 8 3 】

また、C T 中 C T 当籤抽籤テーブル内の 2 段階目（サブフラグ D「リーチ目リプ」取得時）の抽籤テーブルでは、判定ビットを構成する各ビットの値により抽籤対象役又は抽籤対象外の役を判別することができ、抽籤対象外の役に対して抽籤値データ（ハズレデータ）をテーブルに格納する必要が無くなる。さらに、C T 中 C T 当籤抽籤テーブル内では、抽籤対象役の当籤確率が 1 0 0 % である確定データとしては、抽籤値「0」を用いることができる。これらのことから、本実施形態では、C T 中 C T 当籤抽籤テーブル（C T 中セット数上乗せ抽籤テーブル）の容量を圧縮することができ、メイン R O M 1 0 2 において、

10

20

30

40

50

空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1284】

〔図柄コード取得処理で得られる効果〕

本実施形態のパチスロ1の図柄コード取得処理（図128参照）では、S647～S649で説明した処理手順で入賞に係るデータの圧縮・展開処理を行い、かつ、その処理の中のメインCPU101専用命令コード（「CALLF」命令等）を用いることにより、入賞に係るデータの圧縮・展開処理の効率化を図ることができるとともに、限られたメインRAM103の容量を有効活用することができる。

【1285】

また、本実施形態では、図柄コード取得処理中のS649の圧縮データ格納処理において、「CALLF」命令で指定するジャンプ先のアドレスは、図柄設定処理（図97参照）中のS330の圧縮データ格納処理において、「CALLF」命令で指定するジャンプ先のアドレスと同じである。すなわち、本実施形態では、図柄コード取得処理及び図柄設定処理の両処理において、圧縮データ格納処理を実行するためのソースプログラムが共有化（モジュール化）されている。この場合、各処理において、それぞれ別個に圧縮データ格納処理のソースプログラムを設ける必要が無くなるので、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1286】

〔引込優先順位取得処理で得られる効果〕

本実施形態のパチスロ1の引込優先順位取得処理（図134及び図135参照）において、「ANY役」の引込優先対応処理中のS683の判定処理は、ソースプログラム上において、メインCPU101専用命令コードである「JCP」命令（比較命令）により実行される（図136A参照）。

【1287】

「ANY役」の引込優先対応処理のソースプログラム上において、「JCP」命令を用いた場合、上述のように、アドレス設定に係る命令を省略することができるので、「ANY役」の引込優先対応処理の処理効率を高めることができるとともに、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。

【1288】

また、引込優先順位取得処理中のS686の停止制御用引込要求フラグ設定処理では、ソースプログラム上において、図136Bに示すように、メインCPU101専用命令コードである、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う「LDQ」命令、及び、「CALLF」命令が利用される。

【1289】

それゆえ、S686の停止制御用引込要求フラグ設定処理では、「LDQ」命令を用いることにより、ソースプログラム上において、アドレス設定に係る命令を省略することができる。また、「CALLF」命令は、上述のように、2バイトの命令コードである。それゆえ、停止制御用引込要求フラグ設定処理において、これらのメインCPU101専用命令コードを使用することにより、処理の効率化を図ることができ、限られたメインRAM103の容量を有効活用することができる。

【1290】

さらに、本実施形態では、優先引込順位取得処理中のS686の停止制御用引込要求フラグ設定処理において、「CALLF」命令で指定するジャンプ先の論理積演算処理のアドレスは、引込優先順位格納処理（図126参照）中のS626の論理積演算処理において「CALLF」命令で指定するジャンプ先のアドレスと同じである（図127参照）。すなわち、本実施形態では、優先引込順位取得処理及び引込優先順位格納処理の両処理にお

10

20

30

40

50

いて、論理積演算処理を実行するためのソースプログラムが共有化（モジュール化）されている。

【1291】

この場合、優先引込順位取得処理及び引込優先順位格納処理の両処理において、それぞれ別個に論理積演算処理のソースプログラムを設ける必要が無くなるので、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1292】

また、引込優先順位取得処理中のS687の引込優先順位テーブル（図137参照）の取得処理では、図136Cに示すように、「LDQ」命令（メインCPU101専用命令コード）が利用される。それゆえ、S687の引込優先順位テーブルの取得処理においても、ソースプログラム上において、アドレス設定に係る命令を省略することができる。その結果、引込優先順位テーブルの取得処理の効率化を図ることができるとともに、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。

【1293】

【リール停止制御処理で得られる効果】

本実施形態のパチスロ1のリール停止制御処理（図138参照）において、S711～S715の処理では、図139に示すように、ソースプログラム上において、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う「LDQ」命令、及び、「CALLF」命令が利用される。

【1294】

それゆえ、本実施形態では、これらのメインCPU101専用命令コードを用いることにより、リール制御処理のソースプログラムの容量を削減することができるとともに、リール停止制御処理の処理効率を向上させることができる。すなわち、本実施形態では、主制御回路90におけるプログラム処理速度の効率化と容量の削減を行うことが可能となり、削減した容量に応じて増加したメインROM102の空き領域を活用して、遊技性を高めることができる。

【1295】

また、リール停止制御処理中のS726の判定処理（リール（回胴）の停止状態のチェック処理）では、図140に示すように、ソースプログラム上において、「LDQ」命令及び「ORQ」命令（Qレジスタ（拡張レジスタ）を用いてアドレス指定を行うメインCPU101専用命令コード）が用いられる。

【1296】

それゆえ、本実施形態では、リール停止制御処理のソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1297】

【入賞検索処理で得られる効果】

本実施形態のパチスロ1の入賞検索処理（図145参照）において、S764の払出枚数及び判定対象データのセット処理では、図146に示すように、ソースプログラム上において、「LDIN」命令が用いられる。

【1298】

それゆえ、本実施形態の入賞検索処理では、一つの「LDIN」命令により、データのロード処理及びアドレスの更新処理の両方を行うことができる。この場合、入賞検索処理のソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する（増大させ

10

20

30

40

50

る) ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1299】

また、入賞検索処理中のS770の判定処理で参照するメダルカウンタの値の取得処理、S772の処理で参照する入賞枚数カウンタの値の取得処理、及び、S775の処理で行う入賞枚数カウンタの保存(更新)処理では、いずれも、図146に示すように、Qレジスタ(拡張レジスタ)を用いてアドレス指定を行う「LDQ」命令が用いられる。それゆえ、本実施形態の入賞検索処理では、ソースプログラム上において、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量(メインROM102の使用容量)を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する(増大させる)ことができ、増えた空き容量を活用して、遊技性を高めることができる。

10

【1300】

さらに、入賞検索処理中のS769の判定処理では、図146に示すように、ソースプログラム上において、「JSLAA」命令が用いられ、S770及びS773の判定処理では、「JCP」命令が用いられる。入賞検索処理のソースプログラム上において、「JSLAA」命令及び「JCP」命令を用いた場合、上述のように、アドレス設定に係る命令を省略することができ、その分、ソースプログラムの容量(メインROM102の使用容量)を低減することができる。この結果、本実施形態では、メインROM102において、空き容量を確保する(増大させる)ことができ、増えた空き容量を活用して、遊技性を高めることができる。

20

【1301】

【イリーガルヒットチェック処理で得られる効果】

本実施形態では、図28～図30に示すように、入賞作動フラグ格納領域(表示役格納領域)の構成が当たり要求フラグ格納領域(内部当籤役格納領域)のそれと同じである。それゆえ、本実施形態のイリーガルヒットチェック処理におけるS784の演算処理では、ソースプログラム(図149参照)上において、入賞役のデータと内部当籤役のデータとを単純に論理積(「AND」命令で実行する)するだけで、入賞役のデータと内部当籤役のデータとの合成結果を得ることができる。

【1302】

それゆえ、本実施形態では、イリーガルヒットチェック処理を効率化及び簡略化することができ、その結果、主制御プログラムの空き容量を確保することができ、該空き容量を使用して遊技性を高めることができる。

30

【1303】

【入賞チェック・メダル払出処理で得られる効果】

本実施形態のパチスロ1の入賞チェック・メダル払出処理(図150参照)では、クレジットカウンタの更新(+1)後、払出動作を継続する場合、S808の処理において、60.33ms間のウェイト(払出間隔待ち)処理が行われる。この場合、無駄な待ち時間を減らすことができ、遊技者の精神的負担を軽減することができる。

【1304】

【メダル払出枚数チェック処理で得られる効果】

本実施形態のパチスロ1のメダル払出枚数チェック処理(図152参照)中のS814の役連終了枚数カウンタの更新処理では、図153Aに示すように、ソースプログラム上において、メインCPU101専用命令コードである「DCPLD」命令が用いられる。

40

【1305】

S814の処理において、「DCPLD」命令は、枚数管理カウンタの下限判定命令と、判断分岐命令とが一体になった命令コードでなるので、役連終了枚数カウンタの更新(減算)処理及び連終了枚数カウンタの値を「0」に保持する処理の両方を実行することができる。この場合、両処理を別個に実行するための命令コードを設ける必要がなくなる。それゆえ、本実施形態のメダル払出枚数チェック処理では、ソースプログラムの容量(メインROM102の使用容量)を低減することができ、メインROM102において、空き

50

容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1306】

また、メダル払出枚数チェック処理中のS816の処理では、図153Bに示すように、クレジット情報コマンドの通信パラメータ1には払出枚数カウンタの値がセットされ、通信パラメータ5にはクレジットカウンタの値がセットされる。しかしながら、クレジット情報コマンドを構成するその他の通信パラメータ2～4（未使用パラメータ）には、現時点においてHレジスタ、Eレジスタ及びDレジスタにそれぞれ格納されている値がセットされる。それゆえ、クレジット情報コマンド送信時における通信パラメータ2～4の値は不定値となる。その結果、本実施形態では、クレジット情報コマンドのサム値（BCC）を送信毎に不定値にすることができ、ゴト等の不正行為を抑制することができる。

10

【1307】

〔7セグLED駆動処理で得られる効果〕

本実施形態のパチスロ1の7セグLED駆動処理（図159参照）中のS936で行われる7セグコモン出力（選択）データ及び7セグカソード出力データの出力処理は、図160Bに示すように、一つのソースコード「LD（cPA__SEGC0M），BC」により実行される。すなわち、本実施形態では、7セグLED駆動処理において、2桁の7セグLEDをダイナミック点灯制御する際に、7セグコモン出力データと、7セグカソード出力データとが同時に出力される。この出力制御は、情報表示器6内の指示モニタに押し順表示データを表示する際にも行われる。

20

【1308】

この場合、7セグLED駆動処理のソースプログラム上において、7セグLEDのダイナミック点灯制御に必要な命令コード数を減らすことができる。それゆえ、本実施形態では、ソースプログラムの容量（メインROM102の使用容量）を低減することができ、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1309】

〔タイマー更新処理で得られる効果〕

本実施形態のパチスロ1のタイマー更新処理（図164参照）中のS952の処理（2バイトタイマーの更新処理）では、図165に示すように、ソースプログラム上において、メインCPU101専用命令コードである「DCPWL D」命令が用いられる。

30

【1310】

タイマー更新処理において、「DCPWL D」命令を実行した場合、上述のように、タイマー数（2バイトタイマー数）の更新（減算）処理及びタイマー数を「0」に保持する処理の両方を実行することができる。この場合、両処理を別個に実行するための命令コードを設ける必要がなくなる。それゆえ、本実施形態では、ソースプログラムの容量（メインROM102の使用容量）を低減することができ、メインROM102において、空き容量を確保する（増大させる）ことができ、増えた空き容量を活用して、遊技性を高めることができる。

【1311】

<第1実施形態における各種変形例>

以上、本発明の第1実施形態に係る遊技機の構成及び動作について、その作用効果も含めて説明した。しかしながら、本発明は、上述した第1実施形態に限定されるものではなく、特許請求の範囲に記載した本発明の要旨を逸脱しない限り、その他の種々の実施形態及び変形例が含まれる。

40

【1312】

〔変形例1：通常ART中のCT前兆遊技と報知抽籤〕

上記第1実施形態のパチスロ1では、理解を容易にするために、遊技者にとって有利な状態（例えば、CT）に当籤した場合に、遊技状態を次遊技（次ゲーム）から有利な状態に移行させる例を説明したが、本発明はこれに限定されない。例えば、遊技者に対して有利

50

な状態の遊技制御を行う場合に、いわゆる「前兆遊技」などと呼ばれている、所定回数の遊技を行った後に当該有利な状態の遊技制御を行ってもよい。

【1313】

ここでは、図174A及び174Bを参照して、その一例として、遊技状態が、通常ARTから所定回数の前兆遊技を介してCTに移行する例を説明する。なお、図174Aは、変形例1におけるCT抽籤当籤時の遊技フローを示す図であり、図174Bは、前兆遊技中に行われるフラグ変換抽籤で用いられるフラグ変換抽籤テーブルの構成図である。

【1314】

この例の通常ART中の遊技では、まず、図174Aに示すように、上記第1実施形態と同様に、ART中CT抽籤テーブル（図50参照）を用い、内部当籤役（サブフラグ）に基づいてCT抽籤を行う。このCT抽籤に当籤すると、遊技状態が、CT（上乘せチャンスゾーン）という遊技者にとって有利な状態に移行することが決定されるので、CTに当籤するまでの期間において、CT抽籤のための各種抽籤は、メイン側（主制御基板71）で行われる。例えば、主制御基板71（メインCPU101）は、内部当籤役を決めるための内部抽籤を行うとともに、内部当籤役として内部当籤役「F__確チリリプ」、「F__1確チリリプ」及び「F__リーチ目リプA」～「F__リーチ目リプD」のいずれかが決定された場合には、ART中フラグ変換抽籤テーブル（図47A及び47B参照）を用いてフラグ変換抽籤を行う。

【1315】

次いで、通常ART中の遊技においてCT抽籤に当籤すると、この例では、CT移行前に所定期間の前兆遊技（CT前兆遊技）が行われる。なお、この例では、CT前兆遊技において、例えば、フラグ変換抽籤により決定されたサブフラグEX（「3連チリリプ」、「リーチ目リプ」、「リプレイ」等）に基づくCT抽籤などの遊技者にとって特典を付与するような抽籤を行わないこととする。しかしながら、この例では、CT前兆遊技において、例えば、図174Bに示すフラグ変換抽籤テーブルを用いたフラグ変換抽籤をサブ側（副制御基板72側）で行い、このサブ側で行ったフラグ変換抽籤の結果に基づいて、サブ側で行う報知内容が制御（決定）される。例えば、サブ側で行ったフラグ変換抽籤に当籤した場合には、略称「リーチ目リプ」に係る図柄組合せを表示するための情報を表示装置11（プロジェクト機構211及び表示ユニット212）で報知し、フラグ変換抽籤に非当籤であった場合には、略称「リプレイ」に係る図柄組合せを表示するための情報を表示装置11で報知する。

【1316】

上述のように近年のパチスロでは、遊技者の利益（出玉）に影響を与える抽籤をメイン側で行うことが求められているが、この例のパチスロ1では、フラグ変換抽籤の抽籤結果が遊技者の利益に何ら影響を与えない期間（CT前兆遊技の期間）を設け、この期間に限り、サブ側でフラグ変換抽籤を行う。それゆえ、この例では、例えば、CT前兆中という特典の付与が決まっている状況において、例えば、略称「リーチ目リプ」に係る図柄組合せ等の特別な図柄組合せを表示する機会を増やすことができるとともに、図柄組合せの見せ方のバリエーションも増加させることができる。その結果、この例の構成によれば、遊技性をより向上させることが可能になる。

【1317】

なお、図174Bに示す抽籤テーブルは、サブ側で行うフラグ変換抽籤に用いられるCT前兆中のフラグ変換抽籤テーブルであり、ロムカートリッジ基板86に記憶されている。CT前兆中のフラグ変換抽籤テーブルは、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」と、サブ側で行うフラグ変換抽籤の抽籤結果（非当籤／当籤）と、各抽籤結果に対応付けられた抽籤値の情報との対応関係を規定する。

【1318】

図174Bに示すフラグ変換抽籤テーブルから明らかなように、この例では、CT前兆遊技において、内部当籤役「F__リーチ目リプA」～「F__リーチ目リプD」が非常に高い確率でサブフラグEX「リーチ目リプ」に変換される（フラグ変換抽籤に当籤する）。す

10

20

30

40

50

なわち、C T前兆遊技において、内部当籤役「F__リーチ目リブA」～「F__リーチ目リブD」のいずれかが決定されると、高い確率で略称「リーチ目リブ」の図柄組合せが停止表示されることになり、遊技者に対して、現在の遊技がC T前兆遊技中であることを示唆することができる。この際、上述のように、C T前兆遊技においてサブ側で行うフラグ変換抽籤に当籤した場合、略称「リーチ目リブ」に係る図柄組合せを表示するための報知は行われるものの、特典を付与することはない。

【1319】

なお、この例において、C T前兆遊技中のフラグ変換抽籤をメイン側で行ってもよい。しかしながら、図174A及び174Bで説明した例のように、遊技者の利益に何ら影響を与えない期間における抽籤をサブ側で行うことにより、メイン側のデータ容量や処理負荷を軽減することができる。

10

【1320】

〔変形例2：3連チリリブ表示用の押し順の別例〕

上記第1実施形態のパチスロ1では、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定された場合、ストップボタンの押し順が正解であると、略称「3連チリリブ」に係る図柄組合せが表示され、押し順が不正解であると、略称「リプレイ」に係る図柄組合せが表示される例を説明した（図24参照）。また、上記第1実施形態では、内部当籤役「F__確チリリブ」及び「F__1確チリリブ」に対応付けられた正解の押し順が「順押し」である例を説明した。しかしながら、本発明はこれに限定されない。

【1321】

20

例えば、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定された場合における、押し順が不正解である場合に表示される図柄組合せの種類を増やすとともに、内部当籤役「F__確チリリブ」が決定された際の正解の押し順を、内部当籤役「F__1確チリリブ」が決定された場合のそれと異ならせてもよい。その一例（変形例2）を、図175A及び175Bを参照して説明する。なお、図175Aは、変形例2における、内部当籤役と停止表示される図柄組合せ（停止図柄（略称））との対応関係を示す図であり、図175Bは、変形例2における、通常ART中のフラグ変換抽籤の結果とサブ側で行われるナビ種別との対応関係を示す図である。

【1322】

この例では、図175Aに示すように、内部当籤役「F__確チリリブ」が決定された場合における、正解の押し順は「順押し（左リール3Lを第1停止）」とし、不正解の押し順は「中リール3Cを第1停止（以下「中押し」という）又は右リール3Rを第1停止（以下、「逆押し」という）」、すなわち、「変則押し」とする。そして、この例では、内部当籤役「F__確チリリブ」が決定された場合に、順押しされると、略称「3連チリリブ」に係る図柄組合せが表示され、中押しされると、略称「リプレイ」に係る図柄組合せが表示され、逆押しされると、略称「2連チリリブ」に係る図柄組合せが表示される。

30

【1323】

また、この例では、内部当籤役「F__1確チリリブ」が決定された場合における、正解の押し順は逆押しとし、不正解の押し順は順押し及び中押しとする。そして、この例では、内部当籤役「F__1確チリリブ」が決定された場合に、逆押しされると、略称「3連チリリブ」に係る図柄組合せが表示され、中押しされると、略称「リプレイ」に係る図柄組合せが表示され、順押しされると、略称「2連チリリブ」に係る図柄組合せが表示される。

40

【1324】

この例のパチスロ1では、内部当籤役「F__確チリリブ」及び「F__1確チリリブ」のいずれかが決定された場合、順押しすると、内部当籤役の種類に応じて、略称「3連チリリブ」又は「2連チリリブ」に係る図柄組合せが表示される。また、内部当籤役「F__確チリリブ」及び「F__1確チリリブ」のいずれかが決定された場合に、逆押しすると、内部当籤役の種類に応じて、略称「3連チリリブ」又は「2連チリリブ」に係る図柄組合せが表示される。さらに、この例では、内部当籤役「F__確チリリブ」及び「F__1確チリリブ」のいずれかが決定された場合に、中押しすると、内部当籤役の種類に関係なく、略称

50

「リプレイ」に係る図柄組合せが表示される。

【1325】

内部当籤役と停止表示される図柄組合せ（停止図柄（略称））との対応関係を図175Aに示す関係に設定すると、例えば、内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定された場合に、遊技者にチリ図柄を狙わせるための様々なナビ（報知）を実施することが可能になる。

【1326】

例えば、「順押しでチリ図柄を狙わせるナビ」と「逆押しでチリ図柄を狙わせるナビ」との双方を行うことができる。なお、内部当籤役「F__確チリリブ」が決定されている場合には、「順押しでチリ図柄を狙わせるナビ」は略称「3連チリリブ」に係る図柄組合せを表示させるためのナビとなり、「逆押しでチリ図柄を狙わせるナビ」は略称「3連チリリブ」に係る図柄組合せを表示させないためのナビ（略称「2連チリリブ」に係る図柄組合せが表示されるナビ）となる。一方、内部当籤役「F__1確チリリブ」が決定されている場合には、「順押しでチリ図柄を狙わせるナビ」は略称「3連チリリブ」に係る図柄組合せを表示させないためのナビ（略称「2連チリリブ」に係る図柄組合せが表示されるナビ）となり、「逆押しでチリ図柄を狙わせるナビ」は略称「3連チリリブ」に係る図柄組合せを表示させるためのナビとなる。

【1327】

この例において、上述したナビを行うか否かの決定は、メイン側で行うフラグ変換抽籤により管理され、このメイン側のフラグ変換抽籤の結果に基づいて、サブ側の制御によりナビが実行される。この際、通常ART中にメイン側で行われるフラグ変換抽籤の抽籤結果と、サブ側で制御されるナビ種別との対応関係は、図175Bに示す対応関係となる。

【1328】

この例においても、通常ART中の遊技において内部当籤役「F__確チリリブ」又は「F__1確チリリブ」が決定されると、主制御基板71（メインCPU101）は、2段階のフラグ変換抽籤を行う（図47参照）。一方、サブ側では、この2段階のフラグ変換抽籤の結果を、スタートコマンドデータから取得し、2段階のフラグ変換抽籤の結果に基づいて、表示装置11で行うナビを決定する。

【1329】

そこで、この例において、1段階目のフラグ変換抽籤の時点で抽籤結果が非当籤であった場合には、副制御基板72（サブCPU201）は、図175Bに示すように、「リプレイナビ」と称するナビを行う。なお、「リプレイナビ」では、遊技者に対して中押しするように指示する情報が報知される。この例では、図174Aに示すように、内部当籤役が「F__確チリリブ」及び「F__1確チリリブ」のいずれであっても、中押し時には略称「リプレイ」に係る図柄組合せが表示されることになる。

【1330】

また、この例において、1段階目のフラグ変換抽籤に当籤し、かつ、2段階目のフラグ変換抽籤の抽籤結果が非当籤である場合には、副制御基板72（サブCPU201）は、図175Bに示すように、「チリリブ煽りナビ」と称するナビを行う。なお、「チリリブ煽りナビ」では、内部当籤役「F__確チリリブ」が決定されている場合、遊技者に対して逆押しでチリ図柄を狙わせるような指示情報が報知される。一方、「チリリブ煽りナビ」では、内部当籤役「F__1確チリリブ」が決定されている場合、遊技者に対して順押しでチリ図柄を狙わせるような指示情報が報知される。そして、このような「チリリブ煽りナビ」では、図174Aに示すように、内部当籤が「F__確チリリブ」である場合には、逆押し時に略称「2連チリリブ」に係る図柄組合せが表示され、内部当籤役が「F__1確チリリブ」である場合には、順押し時に略称「2連チリリブ」に係る図柄組合せが表示されることになる。

【1331】

また、この例において、1段階目及び2段階目のフラグ変換抽籤の両方に当籤した場合には、副制御基板72（サブCPU201）は、図175Bに示すように、「チリリブ揃い

10

20

30

40

50

ナビ」と称するナビを行う。なお、「チリリブ揃いナビ」では、内部当籤役「F__確チリリブ」が決定されている場合、遊技者に対して順押しでチリ図柄を狙わせるような指示情報が報知される。一方、「チリリブ揃いナビ」では、内部当籤役「F__1確チリリブ」が決定されている場合、遊技者に対して逆押しでチリ図柄を狙わせるような指示情報が報知される。そして、このような「チリリブ揃いナビ」では、図174Aに示すように、内部当籤が「F__確チリリブ」である場合には、順押し時に略称「3連チリリブ」に係る図柄組合せが表示され、内部当籤役が「F__1確チリリブ」である場合には、逆押し時に略称「3連チリリブ」に係る図柄組合せが表示されることになる。

【1332】

この例における、上述したフラグ変換抽籤の抽籤結果に基づく報知は、利益に影響を与えるものではない（フラグ変換抽籤自体は利益に影響を与えるものの、結果として表示される図柄組合せは利益に影響を与えない）ので、この例の上記報知動作は、メイン側（指示モニタ）では行わず、サブ側（表示装置11）でのみ行う。また、この例では、上述のように、「チリリブ揃いナビ」だけでなく、「チリリブ煽りナビ」も併せて行うことで、利益に影響しないナビを多様な態様でサブ側により制御することができる。その結果、遊技の興趣を向上させることができる。

【1333】

〔変形例3：フラグ間中及び非フラグ間中のベルナビ態様の別例〕

上記第1実施形態のパチスロ1では、上述のように、リールの停止操作の情報と一義的に対応する数値を、指示モニタ（不図示）に表示することにより、メイン側での報知を行う。この際、図63A～63Dで説明したナビデータの対応関係が参照される。そして、上記第1実施形態では、BBフラグ間状態（RT5状態）中に「白7ナビ」や「青7ナビ」を行うが、「ベルナビ」などを行わない例を説明したが、本発明はこれに限定されない。BBフラグ間状態（RT5）中に、「白7ナビ」や「青7ナビ」以外に「ベルナビ」などを行う構成にしてもよい。

【1334】

この場合、指示モニタに表示する数値は、BBフラグ間状態中と非BBフラグ間状態中とで互いに異ならせることとしてもよい。ここで、図176A及び176Bに、変形例3における、BBフラグ間状態中にメイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係を示す。なお、図176Aは、RT5状態中（BB1フラグ間）における、メイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係を示す図であり、図176Bは、RT5状態中（BB2フラグ間）における、メイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係を示す図である。

【1335】

また、この例において、非BBフラグ間状態中にメイン側で行う報知（ナビ）とサブ側で行う報知（ナビ）との対応関係は、上記第1実施形態と同様である（図63A及び63B参照）。それゆえ、非BBフラグ間状態において「ベルナビ」が行われる場合、指示モニタには「1」～「3」の数値（押し順役第2指示情報）が表示される。

【1336】

この例では、図176A及び176Bに示すように、BBフラグ間状態において「ベルナビ」が行われる場合、指示モニタに「12」～「14」の数値（押し順役第1指示情報）が表示される。なお、本発明はこれに限定されず、BBフラグ間状態において「ベルナビ」が行われる場合に、指示モニタに表示される数値は適宜変更することができる。なお、この例において、表示装置11を用いて行うサブ側のナビは、BBフラグ間状態及び非BBフラグ間状態の両方において共通のものが設けられていてもよい。

【1337】

〔変形例4：特典付与の別例〕

上記第1実施形態のパチスロ1では、メイン側で行われるフラグ変換抽籤の結果に基づいて報知内容が制御されるので、報知に従い停止操作を行った場合には表示される図柄組合せが異なる。すなわち、上記第1実施形態では、メイン側で行われるフラグ変換抽籤の結

10

20

30

40

50

果に基づいて特典を付与するか否かが決定される例を説明したが、本発明はこれに限定されない。例えば、メイン側（主制御基板 7 1）において、実際に表示された図柄組合せに基づいて特典を付与する構成にしてもよい。

【1338】

この場合、主制御基板 7 1（メイン CPU 1 0 1）は、フラグ変換抽籤の結果に応じて行われる報知に従って所定の図柄組合せが表示された場合には、特典を付与し、報知に従わなかった場合には、所定の図柄組合せが表示されても特典を付与しない構成にしてもよい。上記第 1 実施形態のパチスロ 1 では、押し順に応じて表示される図柄組合せが異なるが、遊技者が報知を無視して停止操作を行ってしまった場合にも、略称「3 連チリリプ」に係る図柄組合せ等の特別な図柄組合せが表示されてしまう可能性がある。それゆえ、この例では、報知を無視して特別な図柄組合せが表示されたとしても、特典を付与することなく、報知に従って特別な図柄組合せが表示された場合に限り、特典を付与するようにしてもよい。

10

【1339】

〔変形例 5：利益（特典）に影響を与えない報知制御の別例〕

上記変形例 1（図 1 7 4 A 及び 1 7 4 B 参照）のパチスロ 1 では、利益に影響を与える報知を行うか否かは、メイン側での報知抽籤（フラグ変換抽籤）により決定され、利益に影響を与えない報知を行うか否かは、サブ側での報知抽籤で決定される例を説明した。そして、この利益に影響を与えない報知の一例として、前兆遊技中に略称「リーチ目リプ」に係る図柄組合せを表示するための報知を行う例を説明したが、利益に影響を与えない報知は、この例に限定されるものではない。

20

【1340】

近年のパチスロでは、停止操作の順序に応じて遊技ロックを行い易い（難しい）状態に遷移させることがある。例えば、「左、中、右」の押し順である場合に遊技ロックを行い易い状態に遷移し、「右、中、左」の押し順である場合に遊技ロックを行い難い状態に遷移することがある。このようなパチスロでは、押し順を報知することで、遊技ロックを行い易い（難しい）状態に遷移させることができる。しかしながら、遊技ロックを行うか否かが利益に対して影響を与える場合には、この報知をメイン側で制御する必要がある。なお、遊技ロックを行うか否かが利益に対して影響を与えない場合には、この報知をサブ側で制御してもよい。

30

【1341】

また、遊技ロックを行うか否かが利益に対して影響を与える場合としては、例えば、遊技ロックが行われることで、ART 抽籤に当籤するといった場合が考えられる。一方、遊技ロックを行うか否かが利益に対して影響を与えない場合は、遊技ロックを演出として行う場合である。例えば、利益（特典）を付与することが決まっている前兆遊技中に遊技ロックを頻繁に行うようにすることにより、この後の遊技において利益が付与されることを演出で示すことが可能になる。

【1342】

ここで、図 1 7 7 A 及び 1 1 7 B を参照して、変形例 5 における、押し順とロック状態とを対応付ける構成例を説明する。なお、図 1 7 7 A は、押し順とロック状態との対応関係を示す図であり、図 1 7 7 B は、遊技ロックによる利益への影響の有無と、報知態様との関係を示す図である。

40

【1343】

図 1 7 7 A に示す例では、内部当籤役「F__維持リプ A」が決定された時に「左、中、右」の押し順で停止操作が行われると、ロック状態が「0」（ロックし難い状態）から「1」（ロックし易い状態）に遷移し、「左、右、中」、「中、左、右」又は「中、右、左」の押し順で停止操作が行われると、現在のロック状態が維持され、「右、左、中」又は「右、中、左」の押し順で停止操作が行われると、ロック状態が「1」から「0」に遷移する。また、内部当籤役「F__維持リプ B」が決定された時には、「中、左、右」の押し順で停止操作が行われると、ロック状態が「0」（ロックし難い状態）から「1」（ロック

50

し易い状態)に遷移し、それ以外の押し順で停止操作が行われると、現在のロック状態が維持される。

【1344】

なお、図117Aに示す例では、説明を単純にするために、ロック状態を「0（ロックし難い状態）」及び「1（ロックし易い状態）」の2段階にする例を説明するが、本発明はこれに限定されない。例えば、3段階以上のロック状態を設けてもよい。また、この場合には、ロック状態が1段階ずつ遷移するのではなく、多段階を1回で遷移する構成にしてもよい。

【1345】

そして、この例では、図117Bに示すように、遊技ロックが利益に影響を与える場合には、報知を行うか否かの報知抽籤をメイン（主制御基板71）側で行うとともに、その抽籤結果に基づいてメイン側及びサブ側の両方で所定の押し順を報知する。一方、遊技ロックが利益に影響を与えない場合には、報知を行うか否かの報知抽籤をサブ（副制御基板72）側で行うとともに、その抽籤結果に基づいてサブ側で所定の押し順を報知する。

【1346】

なお、パチスロとしては、遊技の全期間にわたり遊技ロックが利益に影響を与えるパチスロもあれば、遊技の全期間にわたり遊技ロックが利益に影響を与えないパチスロもある。また、パチスロとしては、遊技の所定期間では遊技ロックが利益に影響を与えるが、遊技の特定期間では遊技ロックが利益に影響を与えないパチスロもある。それゆえ、遊技ロックが利益に影響を与える期間では、報知を行うか否かの報知抽籤をメイン側で行い、その抽籤結果に基づいてメイン側及びサブ側の両方で所定の押し順を報知し、一方、遊技ロックが利益に影響を与えない期間では、報知を行うか否かの報知抽籤をサブ側で行い、その抽籤結果に基づいてサブ側で所定の押し順を報知するようにしてもよい。

【1347】

なお、遊技ロックの制御は、通常、メイン側（主制御基板71）で行われるので、図117Aに示す押し順に応じてロック状態を遷移する機能は、主制御基板71に設けられる。すなわち、主制御基板71は、検出した停止操作の順序に基づいてロック状態を設定するロック状態設定手段、ロック状態設定手段が設定したロック状態に応じた確率で遊技ロックを行うか否かを抽籤により決定する遊技ロック判定手段、及び、遊技ロック判定手段が遊技ロックを行うと判定すると、遊技の進行を一時的に停止するロック実行手段、としても機能する。また、主制御基板71及び／又は副制御基板72は、遊技ロックし易い（し難い）ロック状態をロック状態設定手段が設定する際に、押し順を報知するか否かの報知抽籤を行う報知抽籤手段、及び、報知抽籤手段の抽籤結果に基づいて所定の押し順を報知する報知手段としても機能する。

【1348】

【変形例6：通常ARTやCTの終了条件の別例】

通常ARTやCTの終了条件は、上記第1実施形態で説明した例に限定されず、任意の終了条件を採用することができる。例えば、通常ART中やCT中に付与されたメダル枚数、通常ART中やCT中の単位遊技の消化回数、通常ART中やCT中に行われた遊技者にとって有利な情報の報知の回数、所定ゲーム数（例えば50ゲーム）を1セットとする場合のセット数、1セット終了時の継続率等の終了条件を採用することができる。

【1349】

また、通常ART中やCT中に付与されたメダル枚数の計数手法としては、例えば、単位遊技において払い出されたメダル枚数を計数する手法を採用してもよいし、単位遊技において払い出されたメダル枚数から当該単位遊技に用いられたメダルのベット（掛け）枚数を減算した差枚数（純増枚数）を計数する手法を採用してもよい。また、通常ART中やCT中に付与されたメダル枚数の計数手法としては、実際に増加したメダルに基づき算出する手法（実値による算出）を採用してもよいし、実際に増加したか否かに関わらず、報知に従った場合に増加する予定のメダル枚数に基づき算出する手法（理想値による算出）を採用してもよい。

【1350】

また、内部当籤役の種別によって、付与されたメダルの枚数を増加しない構成、すなわち、付与されたメダルの枚数の終了条件となるメダルの枚数又は差枚数にはカウントしない構成を採用してもよい。例えば、内部当籤役として決定される確率が低い一部の役（レア役）や、停止操作のタイミングに応じて図柄組合せの表示／非表示が切り替わる役などが内部当籤役として決定されたとしても、付与されたメダルの枚数の増減（カウント）を行わないようにしてもよい。

【1351】

〔変形例7〕

また、上記第1実施形態及び各種変形例では、遊技機としてパチスロを例に挙げて説明したが、本発明はこれに限定されない。上記第1実施形態及び各種変形例で採用したリール制御に係る特徴や設定変更及び確認に係る特徴などのパチスロ1に特有の特徴以外の特徴は、「パチンコ」と呼ばれる遊技機にも適用可能であり、同様の効果が得られる。例えば、チェックサムの生成及び判定処理、メインCPU101専用命令コードを使用した各種処理（Qレジスタを用いたアドレスの指定処理、ソフトタイマーの更新処理、7セグLED駆動処理、通信データの生成格納処理等）、規定外ROM領域及び規定外RAM領域を使用した各種処理などの特徴は、「パチンコ」にも適用可能である。

【1352】

<主制御基板及び副制御基板が有する各種機能>

以上、本発明に係るパチスロ1の第1実施形態及び各種変形例について説明した。ここで、本発明の第1実施形態及び上記各種変形例に係るパチスロ1の主制御基板71（主制御回路90，メインCPU101）及び副制御基板72（副制御回路200，サブCPU201）が有する各種機能をまとめて説明する。

【1353】

主制御基板71は、スタートスイッチ79やストップスイッチ基板80と接続され、図1に示す遊技の進行を制御する。それゆえ、主制御基板72は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

【1354】

また、主制御基板71は、通常ART中に内部当籤役「F__確チリリプ」又は「F__1確チリリプ」が決定されると、フラグ変換抽籤を行い、このフラグ変換抽籤の結果やその他の内部当籤役に基づくCT抽籤に当籤すると、通常ARTの継続期間を延長するCTを開始する。それゆえ、主制御基板71は、変換抽籤手段及び上乗せ遊技開始手段としても機能する。

【1355】

また、主制御基板71は、CT中に内部当籤役に応じて通常ARTの継続期間を示すARTゲーム数を上乗せ（延長）する。具体的には、主制御基板71は、サブフラグ「サボテン」、「弱チェリー」又は「強チェリー」に対応する内部当籤役が決定された場合に、通常ARTのARTゲーム数を所定量上乗せし、サブフラグ「3連チリリプ（3連チリリプA又は3連チリリプB）」に対応する内部当籤役が決定された場合に、通常ARTのARTゲーム数を特定量上乗せする。さらに、主制御基板71は、サブフラグ「3連チリリプ」に基づくARTゲーム数の上乗せが行われた回数が、CTの1セットの基本ゲーム数である8回を超える9回以上になると、上乗せ1回あたりの上乗せ量を増加する。それゆえ、主制御基板71は、上乗せ制御手段としても機能する。

【1356】

また、主制御基板71は、CT中にARTゲーム数が上乗せされない遊技の回数を、CTゲーム数カウンタを用いて計数し、CTゲーム数カウンタが「8」を計数すると、CTを終了する。このとき、サブフラグEX「3連チリリプ」が当籤（すなわち、フラグ変換抽籤に当籤）すると、主制御基板71は、1セット8回（8ゲーム）のCT遊技を再セットする（CTゲーム数カウンタの値を初期値に戻す）。それゆえ、主制御基板71は、計数

10

20

30

40

50

手段及び上乗せ遊技終了手段としても機能する。

【1357】

また、主制御基板71は、ボーナス役（内部当籤役「F__BB1」，「F__BB2」）が内部当籤役として決定されると、遊技状態をBBフラグ間状態（RT5状態）に移行させるとともに、BBフラグ間状態ではボーナスが作動するまで（ボーナス役が入賞するまで）ボーナス役を内部当籤役として持ち越す。それゆえ、主制御基板71は、ボーナス持越手段としても機能する。また、主制御基板71は、BBフラグ間状態においてボーナス役が入賞すると、ボーナスを作動させ、遊技状態をボーナス状態に移行させる。それゆえ、主制御基板71は、ボーナス開始手段及び有利遊技手段としても機能する。

【1358】

また、主制御基板71は、図25に示すように、BBフラグ間状態において、ボーナス役と所定の内部当籤役（「はずれ」、「F__特殊1」～「F__特殊3」）とが重複して決定されている場合には、ボーナス役に係る図柄組合せ（「C__BB1」，「C__BB2」）を表示可能となるように停止制御を行い、ボーナス役と所定の内部当籤役以外の内部当籤役とが重複して決定されている場合には、ボーナス役に係る図柄組合せを表示不可能となるように停止制御を行う。そして、BBフラグ間状態中にボーナス役に係る図柄組合せが表示可能な場合には、主制御基板71は、情報表示器6の指示モニタ（不図示）を制御して、ボーナス役を入賞させるための停止操作の態様を一義的に示す数値「10」又は「11」を表示してボーナス指示情報を報知する。一方、BBフラグ間状態中にボーナス役に係る図柄組合せが表示不可能な場合には、主制御基板71は、指示モニタに数値「10」及び「11」を表示（報知）しない。それゆえ、主制御基板71及び情報表示器6の指示モニタは、指示情報報知手段として機能する。

【1359】

また、この際、主制御基板71は、ボーナス告知をした後に限り指示モニタを介して数値「10」又は「11」を報知する。具体的には、主制御基板71は、ボーナス役を持ち越していない状態でボーナス役を内部当籤役として決定すると、その後の遊技の回数を計数して、当該計数結果が所定回数になった後にボーナス役と所定の内部当籤役（「はずれ」、「F__特殊1」～「F__特殊3」）とが重複して決定されると、指示モニタを介して数値「10」又は「11」を報知する。それゆえ、主制御基板71は、ボーナス役を内部当籤役として決定してからの単位遊技の回数を計数する計数手段としても機能する。

【1360】

副制御基板72は、主制御基板71から受信した各種コマンドデータに基づいて遊技履歴を管理するとともに、遊技者からの登録操作を受け付けると、遊技を行う遊技者の登録を受け付ける。それゆえ、副制御基板72は、履歴管理手段及び登録受付手段として機能する。

【1361】

また、本発明に係るパチスロ1は、遊技の進行に応じた演出を行う表示装置11と、表示装置11とは別個に設けられ、トップ画面221や遊技情報画面223，224，225などを含む複数の表示画面を表示するサブ表示装置18と、サブ表示装置18の表示部上に設けられたタッチセンサ19とを有し、副制御基板72は、表示装置11及びサブ表示装置18の動作を制御する。具体的には、副制御基板72は、遊技者の登録を受け付けている場合には、サブ表示装置18に遊技情報画面223，224，225を表示可能となるようにサブ表示装置18を制御し、遊技者の登録を受け付けていない場合には、サブ表示装置18に遊技情報画面223，224，225を表示不可能とするよう制御する。それゆえ、副制御基板72は、制御手段としても機能する。

【1362】

また、主制御基板71は、遊技の結果に応じて様々な特典を付与するため、特典付与手段としても機能する。具体的には、主制御基板71は、有効ラインに沿って表示された図柄組合せに応じて特典を付与する。

【1363】

例えば、内部当籤役「F__3 択ベル__1 s t」が決定された遊技において、略称「ベル」に係る図柄組合せが有効ライン上に表示されると、主制御基板 7 1 は、9 枚のメダルを付与し、略称「ベルこぼし目」に係る図柄組合せが有効ライン上に表示されると、主制御基板 7 1 は、0 枚のメダルを付与する。また、例えば、内部当籤役「F__確チリリプ」が決定された遊技において、略称「3 連チリリプ」又は略称「リプレイ」に係る図柄組合せが有効ライン上に表示されると、主制御基板 7 1 は、再遊技の作動という同一の特典を付与する。また、主制御基板 7 1 は、有効ラインに沿って表示された図柄組合せではなく、フラグ変換抽籤の結果に基づいても特典を付与する。例えば、主制御基板 7 1 は、内部当籤役「F__確チリリプ」が決定されるとフラグ変換抽籤を行い、フラグ変換抽籤に当籤すると、C T 抽籤に当籤するなどの特典を付与する。

10

【1364】

また、副制御基板 7 2 は、表示装置 1 1 を介して停止操作の態様に応じた演出を実行する。それゆえ、副制御基板 7 2 及び表示装置 1 1 は、演出実行手段としても機能する。この際、有効ラインに沿って表示された図柄組合せによって付与する特典が異なる場合には、主制御基板 7 1 は、指示モニタを介して遊技者にとって有利な停止操作の態様を一義的に示す情報を報知し、有効ラインに沿って表示された図柄組合せによって付与する特典が同一である場合には、遊技者にとって有利な停止操作の態様を報知しない。これに対して、副制御基板 7 2 は、表示された図柄組合せによって付与する特典が同一／異なるかに関係なく、停止操作の順序を示す演出を実行する。

【1365】

20

また、主制御基板 7 1 は、C T を開始するか否かの C T 抽籤を行い、C T 抽籤に当籤した場合には、当籤してから例えば所定回数の前兆遊技が行われた後に C T を開始する。それゆえ、主制御基板 7 1 は、有利状態抽籤手段及び有利状態開始手段としても機能する。さらに、この際、C T 前兆遊技中に内部当籤役「F__確チリリプ」又は「F__1 確チリリプ」が決定されると、副制御基板 7 2 は、サブ側でのフラグ変換抽籤を行う。それゆえ、副制御基板 7 2 は、報知抽籤手段としても機能する。

【1366】

また、本発明に係るパチスロ 1 では、フラグ変換抽籤の結果に基づいて、主制御基板 7 1 は特典を付与するとともに、主制御基板 7 1 及び副制御基板 7 2 は、指示モニタ及び表示装置 1 1 を介して押し順を報知する。それゆえ、主制御基板 7 1 は、特典付与手段としても機能する。また、主制御基板 7 1、副制御基板 7 2、指示モニタ及び表示装置 1 1 は、報知手段としても機能する。

30

【1367】

さらに、本発明に係るパチスロ 1 では、主制御基板 7 1（主制御回路 9 0、メイン CPU 1 0 1）は、遊技動作全般に渡って、各種制御処理を行う。それゆえ、主制御基板 7 1 は演算制御手段としても機能する。また、本発明に係るパチスロ 1 では、主制御基板 7 1（主制御回路 9 0、メイン CPU 1 0 1）は、以下に示す各種処理の実行手段としても機能する。

【1368】

主制御基板 7 1（主制御回路 9 0、メイン CPU 1 0 1）は、電断発生時のチェックサム生成処理（図 7 7 参照）及び電源復帰時のサムチェック処理（図 7 9 及び図 8 0 参照）を行う。それゆえ、主制御基板 7 1 はサム値算出手段、サム値減算手段及びサム値判定手段としても機能する。

40

【1369】

主制御基板 7 1（主制御回路 9 0、メイン CPU 1 0 1）は、入賞検索処理（図 1 4 5 参照）を行う。それゆえ、主制御基板 7 1 は、特典付与判定手段及び入賞役決定手段としても機能する。

【1370】

主制御基板 7 1（主制御回路 9 0、メイン CPU 1 0 1）は、通信データ送信処理（図 1 5 8 の割込処理中の S 9 0 4）を行う。それゆえ、主制御基板 7 1 は、データ送信手段と

50

しても機能する。

【1371】

主制御基板71（主制御回路90、メインCPU101）は、設定変更確認処理（図68参照）を行う。それゆえ、主制御基板71は設定変更確認手段、開始時コマンド生成手段及び終了時コマンド生成手段としても機能する。

【1372】

主制御基板71（主制御回路90、メインCPU101）は、通信データ格納処理（図72参照）及び通信データポインタ更新処理（図74参照）を行う。それゆえ、主制御基板71は、通信データ生成手段及び通信データ生成格納手段としても機能する。

【1373】

主制御基板71（主制御回路90、メインCPU101）は、7セグLED駆動処理（図159参照）を行う。それゆえ、主制御基板71は、7セグLED駆動手段、LED駆動制御手段としても機能する。

【1374】

主制御基板71（主制御回路90、メインCPU101）は、遊技復帰処理（図68参照）を行う。それゆえ、主制御基板71は、遊技復帰手段としても機能する。

【1375】

主制御基板71（主制御回路90、メインCPU101）は、メダル受付・スタートチェック処理（図83参照）を行う。それゆえ、主制御基板71は、遊技開始判定手段及び設定確認手段（S233）としても機能する。

【1376】

主制御基板71（主制御回路90、メインCPU101）は、メダル投入チェック処理（図87参照）を行う。それゆえ、主制御基板71は、遊技媒体受付状態判別手段（S255～S258）としても機能する。

【1377】

主制御基板71（主制御回路90、メインCPU101）は、1.1172msec周期で割込処理（図158参照）を繰り返し実行する。それゆえ、主制御基板71は、割込処理実行手段、定周期処理手段としても機能する。

【1378】

主制御基板71（主制御回路90、メインCPU101）は、電源投入時処理（図64参照）を行う。それゆえ、主制御基板71は、電源復帰処理実行手段としても機能する。

【1379】

主制御基板71（主制御回路90、メインCPU101）は、内部抽籤処理（図92参照）を行う。それゆえ、主制御基板71は、内部抽籤手段としても機能する。

【1380】

主制御基板71（主制御回路90、メインCPU101）は、図柄設定処理（図97参照）を行う。それゆえ、主制御基板71は、内部当籤役生成手段（S321）、フラグテーブル展開手段、当籤フラグテーブル展開手段（S324）、フラグ格納領域指定手段、当籤フラグ格納領域指定手段（S329）及びフラグデータ格納手段、当籤フラグデータ格納手段（S330）としても機能する。

【1381】

主制御基板71（主制御回路90、メインCPU101）は、図柄コード取得処理（図28参照）を行う。それゆえ、主制御基板71は、入賞フラグ格納領域指定手段（S648）及び図柄コード格納領域設定手段（S650）としても機能する。

【1382】

主制御基板71（主制御回路90、メインCPU101）は、リール停止制御処理（図139参照）を行う。それゆえ、主制御基板71は、停止操作検出結果取得手段（S714）及び停止制御データ格納領域設定手段（図139中のソースコード「LDQ IX, wR1__CTRL - (wR2__CTRL - wR1__CTRL)」）としても機能する。

【1383】

10

20

30

40

50

主制御基板 7 1 (主制御回路 9 0、メイン CPU 1 0 1) は、引込優先順位取得処理 (図 1 3 4 及び図 1 3 5 参照) を行う。それゆえ、主制御基板 7 1 は、優先停止図柄決定手段、任意役対応処理手段 (S 6 8 0 ~ S 6 8 3)、当籤フラグ格納領域指定手段 (S 6 8 6)、入賞フラグ格納領域指定手段 (S 6 8 6)、論理積演算手段 (S 6 8 6) 及び優先順位データテーブル取得手段 (S 6 8 7) としても機能する。

【1 3 8 4】

主制御基板 7 1 (主制御回路 9 0、メイン CPU 1 0 1) は、イリーガルヒットチェック処理 (図 1 4 8 参照) を行う。それゆえ、主制御基板 7 1 は、エラー検出手段、エラー処理手段、入賞フラグ格納領域指定手段 (S 7 8 1)、論理積演算手段 (S 7 8 4) 及びエラー判定手段 (S 7 8 5) としても機能する。

10

【1 3 8 5】

主制御基板 7 1 (主制御回路 9 0、メイン CPU 1 0 1) は、入賞チェック・メダル払出処理 (図 1 5 0 参照) を行う。それゆえ、主制御基板 7 1 は、遊技媒体払出手段、遊技媒体加算手段 (S 8 0 5)、払出終了判定手段 (S 8 0 7) 及びウェイト発生手段 (S 8 0 8) としても機能する。

【1 3 8 6】

主制御基板 7 1 (主制御回路 9 0、メイン CPU 1 0 1) は、CT 中 CT 抽籤処理 (図 1 1 9 参照) を行う。それゆえ、主制御基板 7 1 は、特典付与決定手段としても機能する。

【1 3 8 7】

主制御基板 7 1 (主制御回路 9 0、メイン CPU 1 0 1) は、サブフラグ変換処理 (図 1 0 5 参照) を行う。それゆえ、主制御基板 7 1 は、第 1 サブフラグ変換手段としても機能する。

20

【1 3 8 8】

また、主制御基板 7 1 (主制御回路 9 0、メイン CPU 1 0 1) は、フラグ変換処理 (図 1 1 1 参照) を行う。それゆえ、主制御基板 7 1 は、第 2 サブフラグ変換手段としても機能する。

【1 3 8 9】

2. 第 2 実施形態

次に、本発明の第 2 実施形態に係るパチスロ 1 について説明する。なお、以下に示す第 2 実施形態のパチスロ 1 では、上記第 1 実施形態や上記各種変形例で説明したパチスロ 1 と遊技性が異なる。しかしながら、第 2 実施形態における例えばデータの圧縮・展開などデータの処理制御や各種処理内容などについては、基本的には、上記第 1 実施形態や上記各種変形例で説明した技術を同様に適用することができる。それゆえ、以下の第 2 実施形態の説明では、上記第 1 実施形態と同様の構成及び処理動作については、その説明を省略し、上記第 1 実施形態と異なる構成 (遊技性等)、動作、処理制御等についてのみ説明する。また、以下の説明において、上記第 1 実施形態と同様の構成には同じ符号を付して説明する。

30

【1 3 9 0】

第 2 実施形態のパチスロ 1 では、リール表示窓 4 の枠内に表示される 3 行×3 列の図柄を結ぶ擬似的なラインのうち、クロスダウンラインを有効ラインとして用い、その他のラインを疑似ラインとして用いる。すなわち、パチスロ 1 は、リール表示窓 4 において各リールの一つの図柄を表示する 3 行×3 列の単位図柄表示領域のうちの、左のリール 3 L の上段の単位図柄表示領域、中のリール 3 C の中段の単位図柄表示領域、及び右のリール 3 R の下段の単位図柄表示領域を夫々結ぶことにより形成されるクロスダウンラインを有効ラインとして用い、それ以外の単位図柄表示領域を結ぶことにより形成されるラインを疑似ラインとして用いる。なお、疑似ラインとしては、例えば、各リールの上段の単位図柄表示領域を結ぶトップライン、各リールの中段の単位図柄表示領域を結ぶセンターライン、各リールの下段の単位図柄表示領域を結ぶボトムライン、及びリール 3 L の下段、リール 3 C の中段及びリール 3 R の上段の単位図柄表示領域を結ぶクロスアップラインがある。

40

【1 3 9 1】

50

＜遊技状態の遷移フロー＞

初めに、図178を参照しながら、第2実施形態のパチスロ1の主制御回路90（メインCPU101）により管理される各種遊技状態及びその遷移フローについて説明する。

【1392】

〔基本的な遊技状態の遷移フロー〕

本実施形態のパチスロ1では、ボーナスゲームの種類として、ビッグボーナス（以下、「BB」と記す）が設けられる。BBは、第1種特別役物と呼ばれるレギュラーボーナス（以下、「RB」と記す）に係る役物連続作動装置であり、RBを連続して作動させる。

【1393】

それゆえ、本実施形態では、主制御回路90は、ボーナス役の当籤／作動（入賞）の有無に基づいて遊技状態を管理する。具体的には、図178に示すように、主制御回路90は、ボーナス役（後述の名称「F__クラウンBB」「F__赤BB」「F__青BB」の内部当籤役）の当籤／作動（入賞）の有無に基づいて、「ボーナス非当籤状態」、「フラグ間状態」及び「ボーナス状態」と称する3種類の遊技状態を管理する。

【1394】

なお、ボーナス非当籤状態は、ボーナスに非当籤であり、かつ、ボーナスが作動（入賞）していない状態であり、ボーナス状態は、ボーナスが作動している状態である。また、本実施形態では、ボーナス役が内部当籤役として決定されると、ボーナスが入賞するまで複数回の遊技に渡りボーナス役が内部当籤役として持ち越された状態が発生する。フラグ間状態は、ボーナス役が内部当籤役として持ち越されている状態、すなわち、ボーナス役が当籤し、かつ、ボーナスが作動していない状態である。

【1395】

また、本実施形態では、リプレイに係る内部当籤役の種別及びその当籤確率が互いに異なる、RT0遊技状態～RT7遊技状態の8種類の状態（以下、それぞれ「RT0状態」～「RT7状態」という）が設けられる。本実施形態では、ボーナス非当籤状態のRT状態は、RT0状態～RT5状態のいずれかとなり、フラグ間状態のRT状態は、RT6状態又はRT7状態となる。

【1396】

なお、RT0状態及びRT1状態は、リプレイに係る内部当籤役の当籤確率が低いRT状態であり、また、RT2状態、RT3状態、RT6状態及びRT7状態は、リプレイに係る内部当籤役の当籤確率が（RT0状態及びRT1状態よりは高い）中程度のRT状態であり、RT4状態及びRT5状態は、リプレイに係る内部当籤役の当籤確率が高いRT状態である。以下では、RT4状態及びRT5状態を「高RT状態」と呼ぶことがあり、また、RT0状態～RT3状態を「低RT状態」と呼ぶことがある。なお、後述するようにRT3状態は、特別なRT状態であるため、「低RT状態」と呼ぶ場合、RT0状態～RT2状態を示し、RT3状態を含まないこととしてもよい。

【1397】

また、RT0状態～RT7状態は、リプレイに係る内部当籤役の当籤確率だけでなく、自身から他のRT状態に移行することになる移行契機に応じて、無限RT状態又は有限RT状態に区別される。無限RT状態は、他のRT状態への移行契機として遊技回数を用いることのないRT状態であり、本実施形態では、RT0状態、RT2状態、RT4状態～RT7状態が該当する。有限RT状態は、他のRT状態への移行契機として遊技回数を用いるRT状態であり、本実施形態では、RT1状態及びRT3状態が該当する。

【1398】

図178に示すように、RT1状態又はRT3状態において、規定回数の遊技が行われると（移行条件（1）が成立すると）、主制御回路90は、遊技状態をRT1状態又はRT3状態からRT0状態に移行させる。なお、RT1状態の規定回数は、10回であり、また、RT3状態の規定回数は、20回である。

【1399】

また、RT2状態、RT4状態又はRT5状態において、略称「RT0移行図柄」に係る

10

20

30

40

50

図柄組合せ（後述の図 197～図 204 参照）が有効ライン上に表示されると（移行条件（2）が成立すると）、主制御回路 90 は、遊技状態を R T 2 状態、R T 4 状態又は R T 5 状態から R T 0 状態に移行させる。

【1400】

また、R T 0 状態、R T 2 状態又は R T 4 状態において、略称「R T 1 移行図柄」に係る図柄組合せ（後述の図 197～図 204 参照）が有効ライン上に表示されると（移行条件（3）が成立すると）、主制御回路 90 は、遊技状態を R T 0 状態、R T 2 状態又は R T 4 状態から R T 1 状態に移行させる。

【1401】

また、R T 0 状態、R T 4 状態又は R T 5 状態において、略称「R T 2 移行図柄」に係る図柄組合せ（後述の図 197～図 204 参照）が有効ライン上に表示されると（移行条件（4）が成立すると）、主制御回路 90 は、遊技状態を R T 0 状態、R T 4 状態又は R T 5 状態から R T 2 状態に移行させる。

【1402】

また、R T 2 状態において、略称「R T 3 移行図柄」に係る図柄組合せ（後述の図 197～図 204 参照）が有効ライン上に表示されると（移行条件（5）が成立すると）、主制御回路 90 は、遊技状態を R T 2 状態から R T 3 状態に移行させる。

【1403】

また、R T 2 状態又は R T 5 状態において、略称「R T 4 移行図柄」に係る図柄組合せ（後述の図 197～図 204 参照）が有効ライン上に表示されると（移行条件（6）が成立すると）、主制御回路 90 は、遊技状態を R T 2 状態又は R T 5 状態から R T 4 状態に移行させる。

【1404】

また、R T 4 状態において、略称「R T 5 移行図柄」に係る図柄組合せ（後述の図 197～図 204 参照）が有効ライン上に表示されると（移行条件（7）が成立すると）、主制御回路 90 は、遊技状態を R T 4 状態から R T 5 状態に移行させる。

【1405】

ここで、パチスロ 1 では、他の R T 状態への移行条件として、有効ライン上に表示される図柄組合せを用いる場合、無限 R T 状態中は、当該図柄組合せが有効ライン上に表示されると、対応する R T 状態に移行させなければならない。略称「R T 0 移行図柄」～「R T 5 移行図柄」は、無限 R T 状態中に有効ライン上に表示されると、対応する R T 状態に移行することになる図柄組合せである。

【1406】

一方、有限 R T 状態では、略称「R T 0 移行図柄」～「R T 5 移行図柄」が有効ライン上に表示されても、対応する R T 状態に移行させない制御が可能である。そのため、本実施形態のパチスロ 1 では、有限 R T 状態である R T 1 状態又は R T 3 状態では、略称「R T 0 移行図柄」～「R T 5 移行図柄」が有効ライン上に表示されても R T 状態を移行させることなく、規定回数の遊技が行われることを契機に R T 状態を移行させることで、図 178 に示す遊技状態の遷移フローを実現している。

【1407】

一方で、無限 R T 状態では、対応する図柄組合せが有効ライン上に表示されると他の R T 状態に移行しなければならないため、遊技性の観点から移行させたくない R T 状態が存在する場合には、対応する R T 状態中に、当該移行させたくない R T 状態への移行契機となっている図柄組合せが有効ライン上に表示されないように制御することで、図 178 に示す遊技状態の遷移フローを実現している。例えば、本実施形態では、R T 4 状態及び R T 5 状態（より詳細には、更に R T 0 状態も含む）では、R T 3 状態に移行することになる略称「R T 3 移行図柄」が有効ライン上に表示されないように制御することで、R T 3 状態への移行が R T 2 状態からしか行われなくないようにしている。

【1408】

また、ボーナス非当籤状態においてボーナス役が内部当籤役として決定されると、主制御

10

20

30

40

50

回路 90 は、遊技状態をボーナス非当籤状態からフラグ間状態に移行させる。具体的には、ボーナス役として名称「F__クラウンBB」又は「F__青BB」に係る内部当籤役が決定されると（移行条件（8）が成立すると）、主制御回路 90 は、遊技状態をボーナス非当籤状態からフラグ間状態の RT6 状態に移行させる。また、ボーナス役として名称「F__赤BB」に係る内部当籤役が決定されると（移行条件（9）が成立すると）、主制御回路 90 は、遊技状態をボーナス非当籤状態からフラグ間状態の RT7 状態に移行させる。

【1409】

また、フラグ間状態においてボーナス役が入賞すると、主制御回路 90 は、遊技状態をフラグ間状態からボーナス状態に移行させる。本実施形態のパチスロ 1 では、ボーナス状態として CBB と NBB とを有しており、名称「F__クラウンBB」が入賞する（略称「クラウンBB」に係る図柄組合せ（後述の図 197～図 204 参照）が有効ライン上に表示される）と（移行条件（10）が成立すると）、主制御回路 90 は、遊技状態をフラグ間状態から CBB に移行させる。また、名称「F__赤BB」又は「F__青BB」が入賞する（略称「BB」に係る図柄組合せ（後述の図 197～図 204 参照）が有効ライン上に表示される）と（移行条件（11）が成立すると）、主制御回路 90 は、遊技状態をフラグ間状態から NBB に移行させる。

【1410】

また、ボーナス状態において規定枚数を超えるメダルが払い出され、ボーナス状態が終了すると（移行条件（12）が成立すると）、主制御回路 90 は、遊技状態をボーナス状態からボーナス非当籤状態の RT0 状態に移行させる。ここで、本実施形態では、ボーナス状態の終了契機となる規定枚数は、CBB においては「396 枚」であり、NBB においては「232 枚」である。

【1411】

<メインROMに記憶されているデータテーブルの構成>

次に、メインROM 102 に記憶されている各種データテーブルの構成について説明する。

【1412】

[図柄配置テーブル]

まず、図 179 を参照して、図柄配置テーブルについて説明する。図柄配置テーブルは、左リール 3L、中リール 3C 及び右リール 3R のそれぞれの回転方向における各図柄の位置と、各位置に配置された図柄の種類を特定するデータ（以下、図柄コード（図 179 中の図柄コード表を参照）という）との対応関係を規定する。

【1413】

なお、第 2 実施形態のパチスロ 1 では、図柄コード表に示すように、各リールには、図柄「赤7」「青7」「BAR」「リプレイ」「ベル1」「ベル2」「ベル3」「スイカ1」「スイカ2」及び「チェリー」の 10 種類の図柄が表示されている。

【1414】

[内部抽籤テーブル]

次に、図 180～図 182 を参照して、内部当籤役を決定する際に参照される内部抽籤テーブルについて説明する。内部抽籤テーブルは、遊技状態毎に設けられ、各種内部当籤役と、各内部当籤役が決定されときの抽籤値との対応関係を規定する。なお、ボーナス非当籤状態の RT0 状態中は、RT0 用の内部抽籤テーブルが参照され、ボーナス非当籤状態の RT1 状態中は、RT1 用の内部抽籤テーブルが参照され、ボーナス非当籤状態の RT2 状態中は、RT2 用の内部抽籤テーブルが参照され、ボーナス非当籤状態の RT3 状態中は、RT3 用の内部抽籤テーブルが参照され、ボーナス非当籤状態の RT4 状態中は、RT4 用の内部抽籤テーブルが参照され、ボーナス非当籤状態の RT5 状態中は、RT5 用の内部抽籤テーブルが参照される。

【1415】

また、内部当籤役「F__クラウンBB」を持ち越しているフラグ間状態の RT6 状態中は、クラウンBBフラグ間用（RT6）の内部抽籤テーブルが参照され、内部当籤役「F__赤BB」を持ち越しているフラグ間状態の RT7 状態中は、赤BBフラグ間用（RT7）

の内部抽籤テーブルが参照され、内部当籤役「F__青BB」を持ち越しているフラグ間状態のRT6状態中は、青BBフラグ間用(RT6)の内部抽籤テーブルが参照される。また、CBBのボーナス状態中は、CBB用の内部抽籤テーブルが参照され、NBBのボーナス状態中は、NBB用の内部抽籤テーブルが参照される。

【1416】

〔内部当籤役と図柄組合せとの対応表(図柄組合せ決定テーブル)〕

次に、図183～図196を参照して、内部当籤役と図柄組合せとの対応表(図柄組合せ決定テーブル)について説明する。図柄組合せ決定テーブルは、各種内部当籤役と、各内部当籤役に対応付けられた、有効ライン上に表示可能な図柄組合せ(コンビネーション)との対応関係を規定する。すなわち、内部当籤役が決定されると、有効ライン上に表示可能な図柄組合せの種別(入賞可能な表示役の種別)が一義的に決定される。

10

【1417】

各図柄組合せ決定テーブル中の図柄組合せ欄に記載の各種データは、左リール3L、中リール3C及び右リール3Rに渡って設定された有効ラインに沿って表示を許可する図柄組合せを識別するためのデータである。なお、図柄組合せ(表示役)欄に記載の各名称と、具体的な図柄組合せとの関係は、後述の図197～図204において説明する。

【1418】

また、図柄組合せ決定テーブル中に記載の「○」印は、決定された内部当籤役において、有効ライン上に表示可能な図柄組合せ(コンビネーション)、すなわち、入賞可能となる表示役を示す。なお、図柄組合せ決定テーブルには、「内部当籤役」が「はずれ」となる場合が規定されていないが、これは、図183～図196に示した図柄組合せテーブルにより規定されている全ての図柄組合せの表示が許可されないことを示す。

20

【1419】

本実施形態のパチスロ1では、主制御回路90(メインCPU101)は、内部当籤役及び遊技状態に応じて停止制御を異ならせ、所定の役が内部当籤役として決定された場合に、図183～図196に示す対応関係の図柄組合せ(コンビネーション)を表示可能とするように左リール3L、中リール3C及び右リール3Rの回転停止制御を行う。なお、図183～図196に示す対応表では、決定された内部当籤役に対して表示可能な全ての図柄組合せを「○」印で列挙しているが、優先して引き込む図柄の関係上、「○」印が付された図柄組合せであっても表示されないことがある。

30

【1420】

〔図柄組合せの内容〕

次に、図197～図204を参照して、図柄組合せの具体的な内容について説明する。図197～図204において、コンビネーション欄は、各リール3L、3C、3Rが停止した場合に、有効ラインに沿って表示される各リール3L、3C、3Rの図柄を示し、名称欄は、コンビネーション欄が示す図柄の組合せの名称を意味する。また、払出欄は、対応する図柄の組合せ(コンビネーション)が表示された際に払い出されるメダルの枚数を規定する。また、略称欄は、それぞれの図柄の組合せの役割や、それぞれの図柄の組合せの特徴からつけられた略称を示す。

【1421】

コンビネーション名「R__3rd移行__01」～「R__1st移行__18」は、略称「RT2移行図柄」に係る図柄組合せであり、有効ラインに沿って表示されると、0枚のメダルが払い出される。また、上述したように、略称「RT2移行図柄」に係る図柄組合せは、無限RT状態中に有効ラインに沿って表示されると、遊技状態がRT2状態に移行する(図178参照)。

40

【1422】

また、コンビネーション名「T__停止制御A」～「T__停止制御L__02」は、略称「制御用出目」であり、有効ラインに沿って表示されると、0枚のメダルが払い出される。

【1423】

また、コンビネーション名「C__7RBR__01」「C__7RBR__02」は、第一種特

50

別役物の連続作動装置（ＢＢ）に係る図柄組合せであり、略称を「クラウンＢＢ」とする。略称「クラウンＢＢ」の図柄組合せが有効ラインに沿って表示されると、遊技状態がボーナス状態（ＣＢＢ）に移行する（図１７８参照）。

【１４２４】

また、コンビネーション名「Ｃ__７ＲＲＲ」～「Ｃ__７ＢＢＢ__０２」は、第一種特別役物の連続作動装置（ＢＢ）に係る図柄組合せであり、略称を「ＢＢ」とする。略称「ＢＢ」の図柄組合せが有効ラインに沿って表示されると、遊技状態がボーナス状態（ＮＢＢ）に移行する（図１７８参照）。

【１４２５】

また、コンビネーション名「Ｃ__ＣＤリプ」～「Ｃ__ＣＵリプ__０９」は、略称「斜めリプレイ」に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。なお、略称「斜めリプレイ」に係る図柄組合せは、３行×３列の図柄を結ぶ擬似的なラインのうちの斜めのライン（クロスダウンライン又はクロスアップライン）に図柄「リプレイ」が表示される図柄組合せである。

【１４２６】

また、コンビネーション名「Ｃ__ＲＴ４リプ__０１」～「Ｃ__ＲＴ４リプ__０３」は、略称「小山リプレイ」（別名を略称「ＲＴ４移行図柄」という）に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。また、上述したように、略称「小山リプレイ（＝ＲＴ４移行図柄）」に係る図柄組合せは、無限ＲＴ状態中に有効ラインに沿って表示されると、遊技状態がＲＴ４状態に移行する（図１７８参照）。なお、略称「小山リプレイ」に係る図柄組合せは、左のリール３Ｌの下段の単位図柄表示領域、中のリール３Ｃの中段の単位図柄表示領域、及び右のリール３Ｒの下段の単位図柄表示領域に図柄「リプレイ」が表示される図柄組合せである。

【１４２７】

また、コンビネーション名「Ｃ__ＲＴ３リプＡ__０１」～「Ｃ__ＲＴ３リプＤ__０４」は、略称「弱チェリプ」（別名を略称「ＲＴ３移行図柄」という）に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。また、上述したように、略称「弱チェリプ（＝ＲＴ３移行図柄）」に係る図柄組合せは、無限ＲＴ状態中に有効ラインに沿って表示されると、遊技状態がＲＴ３状態に移行する（図１７８参照）。なお、略称「弱チェリプ」に係る図柄組合せは、停止操作のタイミングが適切である場合に、左のリール３Ｌの上段又は下段の単位図柄表示領域に図柄「チェリー」が表示される図柄組合せである。

【１４２８】

また、コンビネーション名「Ｃ__ＢＢリプ__０１」～「Ｃ__ＢＢリプ__１８」は、略称「ＢＢ中リプレイ」に係る図柄組合せである。略称「ＢＢ中リプレイ」の図柄組合せが有効ラインに沿って表示されると、再遊技の作動が行われる。

【１４２９】

また、コンビネーション名「Ｃ__ＢＴリプ__０１」～「Ｃ__ＴＰリプ__０３」は、略称「平行リプレイ」（別名を略称「ＲＴ１移行図柄」という）に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。また、上述したように、略称「平行リプレイ（＝ＲＴ１移行図柄）」に係る図柄組合せは、無限ＲＴ状態中に有効ラインに沿って表示されると、遊技状態がＲＴ１状態に移行する（図１７８参照）。なお、略称「平行リプレイ」に係る図柄組合せは、３行×３列の図柄を結ぶ擬似的なラインのうちの平行なライン（ボトムライン、センターライン又はトップライン）に図柄「リプレイ」が表示される図柄組合せである。

【１４３０】

また、コンビネーション名「Ｃ__ＲＴ５__０１」～「Ｃ__ＲＴ５リプ__０９」は、略称「強チャンスリプ」（別名を略称「ＲＴ５移行図柄」という）に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。また、上述したように、略称「強チャンスリプ（＝ＲＴ５移行図柄）」に係る図柄組合せは、無限ＲＴ状態中に有効ラ

10

20

30

40

50

インに沿って表示されると、遊技状態が R T 5 状態に移行する（図 1 7 8 参照）。なお、略称「強チャンスリップ」に係る図柄組合せは、左のリール 3 L の中段の単位図柄表示領域、中のリール 3 C の中段の単位図柄表示領域、及び右のリール 3 R の上段の単位図柄表示領域に図柄「リプレイ」が表示される図柄組合せである。

【1 4 3 1】

また、コンビネーション名「C__CD7リップA__01」～「C__CU7リップC__04」は、略称「7揃いリップ」（別名を略称「R T 5 移行図柄」という）に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。また、上述したように、略称「7揃いリップ（＝R T 5 移行図柄）」に係る図柄組合せは、無限 R T 状態中に有効ラインに沿って表示されると、遊技状態が R T 5 状態に移行する（図 1 7 8 参照）。なお、略称「7揃いリップ」に係る図柄組合せは、停止操作のタイミングが適切である場合に、3 行×3 列の図柄を結ぶ擬似的なラインのうちの何れかに図柄「赤 7」又は図柄「青 7」が表示される図柄組合せである。

10

【1 4 3 2】

また、コンビネーション名「C__CUBARリップA」～「C__BTBARリップC__02」は、略称「BAR揃いリップ」（別名を略称「R T 5 移行図柄」という）に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。また、上述したように、略称「BAR揃いリップ（＝R T 5 移行図柄）」に係る図柄組合せは、無限 R T 状態中に有効ラインに沿って表示されると、遊技状態が R T 5 状態に移行する（図 1 7 8 参照）。なお、略称「BAR揃いリップ」に係る図柄組合せは、停止操作のタイミングが適切である場合に、3 行×3 列の図柄を結ぶ擬似的なラインのうちの何れかに図柄「BAR」が表示される図柄組合せである。

20

【1 4 3 3】

また、コンビネーション名「C__フェイクA__01」～「C__フェイクG__04」は、略称「フェイクリップ」に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。また、コンビネーション名「C__特殊リップA__01」～「C__特殊リップD__02」は、略称「リーチ目リップ」に係る図柄組合せであり、有効ラインに沿って表示されると、再遊技の作動が行われる。

【1 4 3 4】

また、コンビネーション名「C__CDベルAAA__01」～「C__BTベル__09」は、略称「ベル」に係る図柄組合せであり、有効ラインに沿って表示されると、8 枚のメダルが払い出される。なお、略称「ベル」に係る図柄組合せのうち、「C__CDベルAAA__01」～「C__CDベルBBB」に係る図柄組合せは、3 行×3 列の図柄を結ぶ擬似的なラインのうちのクロスダウン（CD）ラインに図柄「ベル」が表示される図柄組合せであり、以下では、「CDベル」と呼ぶことがある。同様に、略称「ベル」に係る図柄組合せのうち、「C__TPベル__01」～「C__TPベル__18」に係る図柄組合せは、トップ（TP）ラインに図柄「ベル」が表示される図柄組合せであり、以下では、「TPベル」と呼ぶことがあり、「C__CTベル__01」～「C__CTベル__06」に係る図柄組合せは、センター（CT）ラインに図柄「ベル」が表示される図柄組合せであり、以下では、「CTベル」と呼ぶことがあり、「C__CUベル__01」～「C__CUベル__18」に係る図柄組合せは、クロスアップ（CU）ラインに図柄「ベル」が表示される図柄組合せであり、以下では、「CUベル」と呼ぶことがあり、「C__BTベル__01」～「C__BTベル__09」に係る図柄組合せは、ボトム（BT）ラインに図柄「ベル」が表示される図柄組合せであり、以下では、「BTベル」と呼ぶことがある。

30

40

【1 4 3 5】

また、コンビネーション名「C__左1枚役A__01」～「C__右1枚役B__06」は、略称「1枚出目」に係る図柄組合せであり、有効ラインに沿って表示されると、1 枚のメダルが払い出される。

【1 4 3 6】

また、コンビネーション名「C__左移行役__01」～「C__右移行役__06」は、略称「

50

「R T O 移行図柄」に係る図柄組合せであり、有効ラインに沿って表示されると、1枚のメダルが払い出されるとともに、遊技状態がR T O 状態に移行する（図178参照）。

【1437】

また、コンビネーション名「C__スイカA__01」～「C__スイカF__02」は、略称「スイカ」に係る図柄組合せであり、有効ラインに沿って表示されると、3枚のメダルが払い出される。また、コンビネーション名「C__チェリーA__01」～「C__チェリーE__02」は、略称「チェリー」に係る図柄組合せであり、有効ラインに沿って表示されると、3枚のメダルが払い出される。

【1438】

また、コンビネーション名「C__SP役A__01」～「C__SP役G」は、略称「確定出目」に係る図柄組合せであり、有効ラインに沿って表示されると、1枚のメダルが払い出される。また、コンビネーション名「C__Vベル__01」～「C__Vベル__18」は、略称「C B B 中ベル」に係る図柄組合せであり、有効ラインに沿って表示されると、12枚のメダルが払い出される。

【1439】

〔当籤役と停止表示される図柄組合せとの対応関係〕

続いて、図205を参照して、内部当籤役と停止表示される図柄組合せとの対応関係について説明する。なお、図205は、決定され得る各種内部当籤役と、各内部当籤役決定時に停止表示される図柄組合せ（略称）との対応関係を示す図である。

【1440】

本実施形態のパチスロ1では、遊技者の停止操作の順序（押し順）に応じて表示される図柄組合せが異なる役、いわゆる「押し順役」を設ける。本実施形態のパチスロ1では、「F__6 択維持リブ123」～「F__6 択維持リブ321」、「F__6 択突入リブ123」～「F__6 択突入リブ321」、「F__6 択転落リブ123」～「F__6 択転落リブ321」、「F__3 択昇格リブ1xx」～「F__3 択昇格リブ3xx」、及び「F__123ベルA」～「F__321ベルB」が押し順役である。なお、以下では、「F__6 択維持リブ123」～「F__6 択維持リブ321」を「6 択維持リブ」と呼ぶことがあり、「F__6 択突入リブ123」～「F__6 択突入リブ321」を「6 択突入リブ」と呼ぶことがあり、「F__6 択転落リブ123」～「F__6 択転落リブ321」を「6 択転落リブ」と呼ぶことがあり、「F__3 択昇格リブ1xx」～「F__3 択昇格リブ3xx」を「3 択昇格リブ」と呼ぶことがあり、「F__123ベルA」～「F__321ベルB」を「押し順ベル」と呼ぶことがある。

【1441】

上述した通り、押し順役には、その名称の一部に、正解となる押し順を示す。例えば、名称の一部に「1xx」とある内部当籤役は、正解となる押し順が、第1停止操作が左リール3Lに対するものであることを意味し、名称の一部に「2xx」とある内部当籤役は、正解となる押し順が、第1停止操作が中リール3Cに対するものであることを意味し、名称の一部に「3xx」とある内部当籤役は、正解となる押し順が、第1停止操作が右リール3Rに対するものであることを意味する。また、内部当籤役の名称の一部に「123」とある内部当籤役は、正解となる押し順が「左、中、右」の順であることを意味し、内部当籤役の名称の一部に「132」とある内部当籤役は、正解となる押し順が「左、右、中」の順であることを意味し、内部当籤役の名称の一部に「213」とある内部当籤役は、正解となる押し順が「中、左、右」の順であることを意味し、内部当籤役の名称の一部に「231」とある内部当籤役は、正解となる押し順が「左、右、中」の順であることを意味し、内部当籤役の名称の一部に「312」とある内部当籤役は、正解となる押し順が「右、左、中」の順であることを意味し、内部当籤役の名称の一部に「321」とある内部当籤役は、正解となる押し順が「右、中、左」の順であることを意味する。

【1442】

図205（A）は、表示される図柄組合せが押し順に関係のない役（非押し順役）における内部当籤役と停止表示される図柄組合せとの対応関係である。パチスロ1では、非押し

10

20

30

40

50

順役が内部当籤役として決定されると、図 205 (A) において対応付けられた図柄組合せの略称のうち、図 183～図 196 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。なお、非押し順役のうち内部当籤役「F__クラウンBB」「F__赤BB」及び「F__青BB」は、所謂取りこぼしが生じる当籤役であり、停止操作のタイミングが正確である場合（所謂、目押しが正確である場合）に、図 205 (A) に示す図柄組合せの略称のうちの、図 183～図 196 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

【1443】

図 205 (B) は、表示される図柄組合せが押し順に関係のある役（押し順役）における内部当籤役と停止表示される図柄組合せとの対応関係である。なお、図 205 (B) では、「押し順ベル」の対応関係は省略しているが、「押し順ベル」が内部当籤役として決定された場合の表示される図柄組合せについては、後に後述する。

10

【1444】

図 205 (B) に示すように、内部当籤役「6 択維持リプ」は、押し順に応じて表示される図柄組合せが異なり、押し順が正解である場合には、コンビネーション名「C__CUリプ」に係る図柄組合せのうちの図 183～図 196 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。一方、押し順が正解でない場合には、コンビネーション名「C__BTリプ」又は「C__CTリプ」に係る図柄組合せのうちの図 183～図 196 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

【1445】

20

なお、コンビネーション名「C__CUリプ」とは、略称「斜めリプレイ」のうちのコンビネーション名「C__CDリプ」以外の「C__CUリプ__01」～「C__CUリプ__09」をいう（図 197～図 204 参照）。また、コンビネーション名「C__BTリプ」とは、略称「平行リプレイ（RT1 移行図柄）」のうちのコンビネーション名「C__BTリプ__01」～「C__BTリプ__09」をいい、また、コンビネーション名「C__CTリプ」とは、略称「平行リプレイ（RT1 移行図柄）」のうちのコンビネーション名「C__CTリプ__01」～「C__BTリプ__06」をいう。

【1446】

内部当籤役「6 択維持リプ」において押し順が正解の場合に有効ラインに沿って表示されるコンビネーション名「C__CUリプ」に係る図柄組合せは、RT 状態の移行契機とはならない図柄組合せであるため、「6 択維持リプ」が内部当籤役として決定され、かつ、押し順が正解の場合には、RT 状態が移行することなく維持される。反対に、押し順が正解でない場合に有効ラインに沿って表示されるコンビネーション名「C__BTリプ」又は「C__CTリプ」に係る図柄組合せは、RT1 移行図柄であるため、無限 RT 中に「6 択維持リプ」が内部当籤役として決定され、かつ、押し順が不正解の場合には、RT 状態が RT1 状態に移行することになる。

30

【1447】

また、内部当籤役「6 択突入リプ」は、押し順に応じて表示される図柄組合せが異なり、押し順が正解である場合には、略称「小山リプレイ」に係る図柄組合せのうちの図 183～図 196 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。一方、押し順が正解でない場合には、コンビネーション名「C__BTリプ」又は「C__CTリプ」に係る図柄組合せのうちの図 183～図 196 に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

40

【1448】

内部当籤役「6 択突入リプ」において押し順が正解の場合に有効ラインに沿って表示される略称「小山リプレイ」に係る図柄組合せは、RT4 移行図柄であり、また、押し順が正解でない場合に有効ラインに沿って表示されるコンビネーション名「C__BTリプ」又は「C__CTリプ」に係る図柄組合せは、RT1 移行図柄である。そのため、無限 RT 中に「6 択突入リプ」が内部当籤役として決定され、かつ、押し順が正解の場合には、RT 状態が RT4 状態に移行することになり、反対に、押し順が不正解の場合には、RT 状態が

50

R T 1 状態に移行することになる。

【1449】

また、内部当籤役「6 折転落リブ」は、押し順に応じて表示される図柄組合せが異なり、押し順が正解である場合には、略称「C__C Uリブ」に係る図柄組合せのうちの図183～図196に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。一方、押し順が正解でない場合には、コンビネーション名「小山リプレイ」に係る図柄組合せのうちの図183～図196に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

【1450】

内部当籤役「6 折転落リブ」において押し順が正解の場合に有効ラインに沿って表示されるコンビネーション名「C__C Uリブ」に係る図柄組合せは、R T 状態の移行契機とはならない図柄組合せであり、押し順が正解でない場合に有効ラインに沿って表示されるコンビネーション名「小山リプレイ」に係る図柄組合せは、R T 4 移行図柄である。そのため、無限R T 中に「6 折転落リブ」が内部当籤役として決定され、かつ、押し順が正解の場合には、R T 状態が移行することなく維持されることになり、反対に、押し順が不正解の場合には、R T 状態がR T 4 状態に移行することになる。

【1451】

また、内部当籤役「3 折昇格リブ」は、押し順に応じて表示される図柄組合せが異なり、押し順が正解である場合には、略称「強チャンスリブ」に係る図柄組合せのうちの図183～図196に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。一方、押し順が正解でない場合には、コンビネーション名「C__C Uリブ」又は「C__B Tリブ」に係る図柄組合せのうちの図183～図196に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

【1452】

内部当籤役「3 折昇格リブ」において押し順が正解の場合に有効ラインに沿って表示される略称「強チャンスリブ」に係る図柄組合せは、R T 5 移行図柄であるため、無限R T 中に「3 折昇格リブ」が内部当籤役として決定され、かつ、押し順が正解の場合には、R T 状態がR T 5 状態に移行する。一方、押し順が正解でない場合に有効ラインに沿って表示される図柄組合せのうち、コンビネーション名「C__C Uリブ」に係る図柄組合せは、R T 状態の移行契機とはならない図柄組合せであり、また、コンビネーション名「C__B Tリブ」又は「C__C Tリブ」に係る図柄組合せは、R T 1 移行図柄である。そのため、無限R T 中に「3 折昇格リブ」が内部当籤役として決定され、かつ、押し順が不正解の場合には、R T 状態が維持されることもあり、また、R T 状態がR T 1 状態に移行することもある。

【1453】

〔報知（ART）機能の作動の有無を考慮した遊技状態の遷移フロー〕

続いて、図206及び図207を参照して、報知（ART）機能の作動の有無を考慮した遊技状態の遷移フローについて説明する。本実施形態では、主制御回路90（メインCPU101）により、遊技者にとって有利な停止操作を報知する機能（ART機能）の作動の有無が決定される。それゆえ、本実施形態では、ボーナス非作動状態においてART機能の作動／非作動状態も遊技状態として管理する。

【1454】

本実施形態のパチスロ1では、主制御回路90は、非ボーナス作動状態において、報知（ART）の有無に基づいて「通常状態」「C Z（チャンスゾーン）」及び「ART状態」を別個の遊技状態として管理する（より詳細には、図206に示すように、それぞれの遊技状態への前兆期間や準備期間も別個の遊技状態として管理する）。

【1455】

通常状態は、遊技者にとって有利な停止操作の情報を報知しない遊技状態（非ナビ区間）であり、遊技者にとって不利な遊技状態である。通常状態は、R T 0～R T 5 状態のいずれかの状態である。

10

20

30

40

50

【1456】

CZ（チャンスゾーン）は、通常状態に比べてART抽籤に当籤する確率が高い遊技状態であり、通常状態に比べると遊技者にとって有利な遊技状態である。本実施形態のパチスロ1において、CZは、「初当りCZ」と「継続CZ」とからなり、また、継続CZは、更に「CZ（ART後）」と「特殊CZ」とからなる。

【1457】

CZは、1セット10ゲームの遊技期間であるが、CZからART状態に移行すると、残りの遊技期間が最低5ゲーム延長される。後述するようにパチスロ1では、CZ中にART抽籤に当籤すると、CZを一時中断してART状態に移行する。そして、その後ART状態が終了すると、中断していたCZを再開する。また、この再開したCZにおいてART抽籤に当籤すると、CZを一時中断してART状態に移行し、ART状態の終了後に中断していたCZを再開する。

【1458】

このときCZからART状態に移行するたびに、CZの残り遊技期間が延長されるため、パチスロ1では、CZ中のART抽籤に当籤する限り、CZとART状態とがループする。初当りCZは、このようなCZとART状態とのループにおける最初のCZであり、継続CZは、CZとART状態とのループ中に用いられるCZである。すなわち、CZとART状態とのループは、初当りCZから始まり、初当りCZ中にART抽籤に当籤すると、ART状態に移行し、このART状態が終了すると、その後、継続CZとART状態とがループすることになる。

【1459】

初当りCZは、通常状態から（CZ前兆を介して）移行するCZであり、遊技者にとって有利な停止操作の情報を報知しない遊技状態（非ナビ区間）である。初当りCZは、RT0～RT5状態のいずれかの状態である。

【1460】

継続CZは、ART状態から（前兆を介さずに）移行するCZであり、遊技者にとって有利な停止操作の情報を報知する遊技状態（ナビ区間）である。継続CZは、基本的には、RT4状態又はRT5状態のいずれかの状態である。継続CZのうち、CZ（ART後）と特殊CZとは、ART抽籤に当籤する確率がそれぞれ異なる状態であり、後述するように、特殊CZの方がCZ（ART後）よりもART抽籤に当籤する確率が高い。

【1461】

また、ART状態は、遊技者にとって有利な停止操作の情報を報知する遊技状態（ナビ区間）であり、遊技者にとって有利な遊技状態である。また、ART状態は、基本的には、RT4状態又はRT5状態のいずれかの状態である。なお、本実施形態では、ARTに当籤している状態で前兆遊技が終了し、RT状態がRT4状態又はRT4状態まで移行すると、ART状態が開始される。

【1462】

図206に示すように、ART状態は、「通常ART」と「EP（エピソード）」と「ランク決めART」とからなる。通常ARTは、1セットを30ゲームとするART状態であり、1セット終了後は必ずCZ（継続CZ）に再突入する。EPは、所謂上乘せ特化ゾーンであり、ART状態の継続期間の延長が通常ARTよりも高確率で行われる状態である。なお、本実施形態のパチスロ1では、基本的には、セット数によりART状態の継続期間を管理しているため、EP中は、高確率でART状態の権利が付与されることになる（なお、ART状態の権利は、メインRAM103に記憶されており、ART状態の権利が付与されると、付与された数の権利がメインRAM103に記憶され、また、遊技状態がCZや通常状態からART状態に移行すると、メインRAM103に記憶されているART状態の権利が1つ消去される）。

【1463】

ランク決めARTは、CZからART状態に移行した後の最初の1ゲームで完結する遊技状態であり、その後に行われる通常ART中のランクを決定する。なお、詳細は後述する

10

20

30

40

50

が、通常ART中のランクは、通常ART中における各種上乘せ（例えば、ART状態のセット数の上乘せやCZのゲーム数の上乘せ）に影響を与える情報であり、高ランクであるほど各種上乘せが優遇されている。

【1464】

図206及び図207に示すように、通常状態において後述するCZ抽籤に当籤し、かつ、1ゲーム以上のCZ前兆ゲーム数が決定されると（移行条件（A1）が成立すると）、主制御回路90は、遊技状態を通常状態からCZ前兆に移行させる。CZ前兆は、CZに移行する前に行われる前兆期間であり、CZ前兆においてCZ前兆ゲーム数が消化されると（移行条件（A2）が成立すると）、主制御回路90は、遊技状態をCZ前兆から初当りCZに移行させる。

10

【1465】

また、初当りCZにおいて後述するART抽籤に当籤すると、当籤した遊技におけるRT状態が低RT状態（RT0～RT3状態）である場合には（移行条件（B1）が成立すると）、主制御回路90は、遊技状態を初当りCZからART準備中に移行させる。ART準備中は、ART抽籤に当籤した後に遊技状態をART状態に移行させるための準備期間であり、RT状態を高RT状態（RT4又はRT5状態）に移行させるために必要な停止操作の情報が報知される。この報知の結果、RT状態が高RT状態まで移行すると（移行条件（B2）が成立すると）、主制御回路90は、遊技状態をART準備中からランク決めARTに移行させる。

【1466】

20

一方、初当りCZにおいて、ART抽籤に当籤した遊技におけるRT状態が高RT状態である場合には（移行条件（B3）が成立すると）、主制御回路90は、遊技状態を初当りCZからランク決めARTに移行させる。また、初当りCZがART抽籤に当籤することなく終了すると（移行条件（B4）が成立すると）、主制御回路90は、遊技状態を初当りCZから通常状態に移行させる。

【1467】

また、通常状態中もART抽籤を行っており、通常状態においてART抽籤に当籤し、かつ、1ゲーム以上のART前兆ゲーム数が決定されると（移行条件（C1）が成立すると）、主制御回路90は、遊技状態を通常状態からART前兆に移行させる。ART前兆は、ART状態に移行する前に行われる前兆期間である。ART前兆においてART前兆ゲーム数が消化されると、ART前兆ゲーム数が消化された遊技におけるRT状態が低RT状態である場合には（移行条件（C2）が成立すると）、主制御回路90は、遊技状態をART前兆からART準備中に移行させ、ART前兆ゲーム数が消化された遊技におけるRT状態が高RT状態である場合には（移行条件（C3）が成立すると）、主制御回路90は、遊技状態をART前兆からランク決めARTに移行させる。

30

【1468】

なお、図206及び図207に示すように、本実施形態のパチスロ1では、CZ前兆及びART前兆は、通常状態からCZ又はART状態に移行する際に経由することがある一方で、CZからART状態、又はART状態からCZに移行する場合には経由することがない。

40

【1469】

続いて、ランク決めARTは、1ゲームで完結し、1ゲームが行われるとランク決めART中に決定した通常ART中のランクに基づいて他の遊技状態に移行する。具体的には、ランク決めART中に決定したランクが3以下である場合には（移行条件（D）が成立すると）、主制御回路90は、遊技状態をランク決めARTから通常ARTに移行させる。

【1470】

また、ランク決めART中に決定したランクが4であり、かつ、ランク決めART終了時のRT状態がRT4状態である場合には（移行条件（E1）が成立すると）、主制御回路90は、遊技状態をランク決めARTからEP準備中に移行させる。EP準備中は、ランク決めARTからEPに移行させる際にRT状態をRT5状態に移行させるための準備期

50

間（なお、EPは、RT5状態中に行われる遊技状態である）であり、EP準備中にRT状態がRT5状態まで移行すると（移行条件（E2）が成立すると）、主制御回路90は、遊技状態をEP準備中からEPに移行させる。一方、ランク決めART中に決定したランクが4であり、かつ、ランク決めART終了時のRT状態がRT5状態である場合には（移行条件（E3）が成立すると）、主制御回路90は、遊技状態をランク決めARTからEPに移行させる。

【1471】

また、通常ARTにおいて、ナビ高確中に「3択昇格リブ」が内部当籤役として決定され、かつ、当該遊技においてRT5状態に移行すると（移行条件（F）が成立すると）、主制御回路90は、遊技状態を通常ARTからEPに移行させる。なお、「3択昇格リブ」が内部当籤役として決定された遊技では、押し順正解時に略称「強チャンスリブ（RT5移行図柄）」に係る図柄組合せが有効ラインに沿って表示され、RT状態がRT5状態に移行する一方で、押し順不正解時には、RT状態がRT5状態に移行しない。後述するように、通常ART中は、ナビ高確中は、「3択昇格リブ」が内部当籤役として決定された場合に正解の押し順を報知する一方で、非ナビ高確中は、「3択昇格リブ」が内部当籤役として決定された場合には不正解の押し順（コンビネーション名「C__C__Uリブ」に係る図柄組合せが表示される押し順）を報知する。そのため、ART状態中の押し順の報知に従う限り、非ナビ高確中にRT状態がRT5状態に移行することがなく、また、非ナビ高確中に押し順の報知に従わずにRT5状態に移行してしまった場合には、ナビ高確中ではないため移行条件（F）が成立することないため、非ナビ高確中に遊技状態がEPに移行することがない。

【1472】

また、EPにおいて、保証なし時の「6択転落リブ」が内部当籤役として決定され、かつ、当該遊技においてRT4状態に移行すると（移行条件（G）が成立すると）、主制御回路90は、遊技状態をEPから通常ARTに移行させる。なお、「6択転落リブ」が内部当籤役として決定された遊技では、押し順不正解時に略称「小山リブレイ（RT4移行図柄）」に係る図柄組合せが有効ラインに沿って表示され、RT状態がRT4状態に移行する一方で、押し順正解時には、RT状態がRT4状態に移行しない。後述するように、EP中は、RT4状態への転落回避の保証がある場合には、「6択転落リブ」の当籤時に正解の押し順を報知する一方で、保証がない場合には、「6択転落リブ」の当籤時に正解及び不正解の何れの押し順も報知しない（すなわち、押し順自体を報知しない）。そのため、保証なし時のEP中は、「6択転落リブ」の当籤時に自力で正解の押し順を当てることでEPを継続することができ、反対に押し順が不正解になってしまうと、EPから通常ARTに遊技状態が移行してしまう。

【1473】

また、通常ARTのゲーム数が消化され、1セットの通常ARTが終了すると（移行条件（H）が成立すると）、主制御回路90は、遊技状態を通常ARTから継続CZに移行させる。

【1474】

また、継続CZにおいて移行条件（I）が成立すると、主制御回路90は、遊技状態を継続CZから通常ARTに移行させる。ここで、継続CZ中はART抽籤を行っており、このART抽籤に当籤した場合に移行条件（I）が成立することになる。また、後述するように、本実施形態のパチスロ1では、ART状態のセット数を上乗せ（ストック）することがあり、継続CZの終了時にセット数のストックが残っている場合には、ストックの放出を行い、遊技状態が継続CZから通常ARTに移行する。そのため、継続CZから通常ARTに移行することになる移行条件（I）は、継続CZ中のART抽籤に当籤すること、又は、ART状態のセット数のストックが放出されること、の何れかが満たされた場合に成立する。

【1475】

一方で、移行条件（I）が成立することなく継続CZのゲーム数が消化してしまった場合

には（移行条件（J）が成立すると）、主制御回路90は、遊技状態を継続CZから通常状態に移行させる。なお、本実施形態のパチスロ1では、継続CZ（初当りCZも同様）から通常状態に移行する際に特殊な遊技性を有しており、この点の詳細は、図210で後述する。

【1476】

また、特定状態中にボーナスに係る役（「F__クラウンBB」「F__赤BB」又は「F__青BB」）が内部当籤役として決定されると（移行条件（K1）が成立すると）、主制御回路90は、遊技状態を特定状態から通常フラグ間に移行させる。なお、特定状態とは、通常状態、CZ前兆状態又はART前兆状態の何れかである。また、通常フラグ間において、ボーナスに係る役が入賞しボーナスが作動すると（移行条件（K2）が成立すると）、主制御回路90は、遊技状態を通常フラグ間から通常BBに移行させる。そして、通常BBにおいて規定枚数のメダルが払い出されボーナスの作動が終了すると（移行条件（K3）が成立すると）、主制御回路90は、遊技状態を通常BBから特定状態に移行させる。なお、通常BBから移行する特定状態は、移行条件（K1）が満たされたときに滞在していた遊技状態である。

【1477】

また、通常フラグ間において後述するART抽籤に当籤すると（移行条件（L）が成立すると）、主制御回路90は、遊技状態を通常フラグ間からART中フラグ間に移行させる。同様に、通常BBにおいて後述するART抽籤に当籤すると（移行条件（M）が成立すると）、主制御回路90は、遊技状態を通常BBからART中BBに移行させる。

【1478】

また、所定状態中にボーナスに係る役が内部当籤役として決定されると（移行条件（N1）が成立すると）、主制御回路90は、遊技状態を所定状態からART中フラグ間に移行させる。なお、所定状態とは、CZ（初当りCZ及び継続CZ）、ART準備中、ランク決めART、通常ART、EP前兆及びEPの何れかである。また、ART中フラグ間において、ボーナスに係る役が入賞しボーナスが作動すると（移行条件（N2）が成立すると）、主制御回路90は、遊技状態をART中フラグ間からART中BBに移行させる。そして、ART中BBにおいて規定枚数のメダルが払い出されボーナスの作動が終了すると（移行条件（N3）が成立すると）、主制御回路90は、遊技状態をART中BBからART準備中に移行させる。

【1479】

なお、図206に示すように、複数の遊技状態のうちの、通常状態、CZ前兆、ART前兆及び初当りCZは、遊技者に対して停止操作の情報を報知しない遊技状態（非ナビ区間）であり、また、ART準備中、ランク決めART、通常ART、EP前兆及び継続CZは、遊技者に対して停止操作の情報を報知する遊技状態（ナビ区間）である。また、EP中は、基本的にはナビ区間であるが、保証がない場合の「6択転落リプ」の当籤時に限り、非ナビ区間として扱われる。

【1480】

ここで、図205（B）を参照して、ナビ区間では、「6択維持リプ」又は「6択突入りリプ」が内部当籤役として決定されると、遊技者に対して正解の押し順が報知される。一方、ナビ区間であっても「3択昇格リプ」が内部当籤役として決定された場合は、ナビ高確中であるか非ナビ高確中であるかに応じて報知する停止操作の情報が異なる。具体的には、ナビ区間中のナビ高確中は、「3択昇格リプ」が内部当籤役として決定されると、正解の押し順を報知し略称「強チャンスリプ」に係る図柄組合せの表示を促し、ナビ区間中の非ナビ高確中は、「3択昇格リプ」が内部当籤役として決定されると、不正解の押し順のうちのコンビネーション名「C__CUリプ」に係る図柄組合せが表示される押し順を報知する。

【1481】

また、「6択転落リプ」当籤時に報知する押し順は、EP中の転落回避の保証の有無に応じて異なり、保証あり時（ナビ区間）では、正解の押し順を報知しコンビネーション名「

C__CUリブ」に係る図柄組合せの表示を促し、保証なし時（非ナビ区間）では、何らの押し順も報知しない。なお、EP中の転落回避の保証の有無は、後述する転落モードに基づき管理される。

【1482】

〔内部当籤役と抽籤用フラグとの対応関係〕

続いて、本実施形態のパチスロ1における遊技の流れの詳細について説明する。ここで、パチスロ1では、内部当籤役などに基づき各種の抽籤を行うが、以下では、内部当籤役を抽籤用フラグに変換した上で各種の抽籤を行う。図208は、内部当籤役と抽籤用フラグとの対応関係を示す図である。なお、本実施形態のパチスロ1では、各種抽籤をリール回転開始時（スタート時）又はリール停止時に行うことがある。図208（A）は、スタート時に行う各種抽籤に用いる抽籤用フラグと内部当籤役との対応関係を示し、図208（B）は、スタート時又はリール停止時に行う各種抽籤に用いる抽籤用フラグと内部当籤役との対応関係を示す。

【1483】

図208（A）に示すように、スタート時に行う各種抽籤に用いる抽籤用フラグは、内部当籤役から決定される。例えば、内部当籤役「押し順ベル」及び「F__共通ベル」は、抽籤用フラグ「ベル」に対応する。また、「BB（CBB又はNBB）中のはずれ」は、抽籤用フラグ「ベルはずれA」に対応し、内部当籤役「F__BB中弱チェリブ」は、抽籤用フラグ「ベルはずれB」に対応する。

【1484】

続いて、図208（B）に示すように、リール停止時に行う各種抽籤に用いる抽籤用フラグは、リール停止時に有効ラインに沿って表示された図柄組合せから決定される。上述したように、本実施形態のパチスロ1では、押し順に応じて表示される図柄組合せが異なる押し順役を設けており、このような押し順役では、遊技者が実際に停止操作を行うまで、パチスロ1側において表示される図柄組合せを把握することができない。そこで、パチスロ1では、押し順役の一部については、表示された図柄組合せ（言い換えると、押し順の正解の有無）に応じて抽籤用フラグを異ならせている。

【1485】

なお、ART状態などのナビ区間中は、遊技者に対して停止操作の情報（押し順）が報知されるため、押し順役の当籤時に表示されるべき図柄組合せを、パチスロ1側において予め把握することができる。そこで、図208（B）に示すように、パチスロ1では、押し順役の一部と対応する抽籤用フラグとを、ナビ区間と非ナビ区間で異ならせている。なお、パチスロ1では、表示される（表示されるべき）図柄組合せをパチスロ1側で予め把握できない非ナビ区間中は、各種抽籤をリール停止時に行い、パチスロ1側で予め把握できるナビ区間中は、各種抽籤をスタート時に行う。

【1486】

図208（B）に示すように、内部当籤役「6択突入リブ」は、非ナビ区間では、押し順正解時に抽籤用フラグ「弱チャンスリブ」が対応し、また、押し順不正解時に抽籤用フラグ「通常リブ」が対応する。また、内部当籤役「6択突入リブ」は、ナビ区間では、抽籤用フラグ「弱チャンスリブ」が対応する。なお、後述するように、各種抽籤では、抽籤用フラグ「通常リブ」よりも抽籤用フラグ「弱チャンスリブ」の方が、抽籤結果が優遇されている。

【1487】

ここで、内部当籤役「6択突入リブ」は、押し順正解時にRT2状態からRT4状態に移行する契機となる当籤役であるが、各種抽籤に用いる抽籤用フラグは、押し順正解時（RT4状態に移行）には「弱チャンスリブ」、押し順不正解時（RT2状態のまま維持）には「通常リブ」が対応するため、対応する抽籤用フラグもRT状態も押し順正解時の方が優遇されている。

【1488】

また、内部当籤役「6択転落リブ」は、非ナビ区間では、押し順正解時に抽籤用フラグ「

10

20

30

40

50

通常リブ」が対応し、また、押し順不正解時に抽籤用フラグ「弱チャンスリブ」が対応する。また、内部当籤役「6 択転落リブ」は、ナビ区間では、抽籤用フラグ「弱チャンスリブ」が対応する。

【1489】

ここで、内部当籤役「6 択転落リブ」は、押し順不正解時に R T 5 状態から R T 4 状態に移行（転落）する契機となる当籤役であるが、各種抽籤に用いる抽籤用フラグは、押し順正解時（R T 5 状態のまま維持）には「通常リブ」、押し順不正解時（R T 4 状態に転落）には「弱チャンスリブ」が対応するため、対応する抽籤用フラグとしては、押し順不正解時の方が押し順正解時よりも優遇されている。一方で、R T 状態が R T 5 状態から R T 4 状態に転落してしまうため、R T 状態の観点からは、押し順正解時の方が押し順不正解時よりも優遇されている。

10

【1490】

続いて、内部当籤役「3 択昇格リブ」は、非ナビ区間では、押し順正解時に抽籤用フラグ「強チャンスリブ」が対応し、また、押し順不正解時に抽籤用フラグ「通常リブ」が対応する。また、内部当籤役「3 択昇格リブ」は、ナビ区間中は、ナビ高確又は非ナビ高確に応じて対応する抽籤用フラグが異なり、ナビ区間中のナビ高確時は、抽籤用フラグ「強チャンスリブ」が対応し、また、ナビ区間中の非ナビ高確時は、抽籤用フラグ「通常リブ」が対応する。なお、後述するように、各種抽籤では、抽籤用フラグ「通常リブ」よりも抽籤用フラグ「強チャンスリブ」の方が、抽籤結果が優遇されている。

【1491】

ここで、内部当籤役「3 択昇格リブ」は、押し順正解時に R T 4 状態から R T 5 状態に移行（昇格）する契機となる当籤役であるが、各種抽籤に用いる抽籤用フラグは、押し順正解時（R T 5 状態に昇格）には「強チャンスリブ」、押し順不正解時（R T 4 状態のまま維持）には「通常リブ」が対応するため、対応する抽籤用フラグも R T 状態も押し順正解時の方が優遇されている。

20

【1492】

<各遊技状態中の遊技性>

続いて、図 209～図 214 を参照して、パチスロ 1 における代表的な遊技状態中の遊技の流れについて説明する。

【1493】

[通常状態中の遊技性]

初めに、図 209 を参照して、通常状態中の遊技の流れについて説明する。本実施形態のパチスロ 1 では、通常状態中は A R T 状態を目指して遊技を行うことになる。通常状態から A R T 状態への移行は、通常状態から C Z を経由して A R T 状態に移行するパターン（図 209（A）参照）と、通常状態から直接 A R T 状態に移行するパターン（図 209（B）参照）との 2 パターンがある。

30

【1494】

図 209（A）を参照して、通常状態から C Z を経由して A R T 状態に移行するパターンでは、主制御回路 90 は、通常状態中に C Z 抽籤を行い、C Z 抽籤に当籤すると遊技状態を通常状態から（C Z 前兆を介して）C Z に移行させ、C Z 抽籤に当籤しないと遊技状態を通常状態のまま維持する。パチスロ 1 では、主制御回路 90 は通常状態中に、抽籤用フラグ「ベル」に基づく C Z 抽籤（図 209（A-1）参照）、及びベル以外の抽籤用フラグに基づく C Z 抽籤（図 209（A-2）参照）を行う。

40

【1495】

図 209（A-1）に示すように、パチスロ 1 では、通常状態中のモードとして、モード 1、モード 2、モード 3 及びモード 4 の 4 段階のモードを有しており、主制御回路 90 は、抽籤用フラグが「ベル」である場合に、現在のモードに基づいて C Z 抽籤を行う。なお、4 段階のモードのうち、抽籤用フラグ「ベル」に基づく C Z 抽籤に当籤する確率は、モード 1 が最も低く、モード 2 が次に低く、モード 3 が次に低く、モード 4 が最も高い。

【1496】

50

また、図209(A-2)に示すように、パチスロ1では、通常状態中の抽籤状態として低確、高確及び超高確の3段階の抽籤状態を有しており、主制御回路90は、抽籤用フラグがベル以外である場合に、抽籤用フラグ及び現在の抽籤状態に基づいてCZ抽籤を行う。なお、3段階の抽籤状態のうち、ベル以外の抽籤用フラグに基づくCZ抽籤に当籤する確率は、低確が最も低く、高確が次に低く、超高確が最も高い。

【1497】

主制御回路90は、通常状態中にモード及び抽籤状態を様々な契機で移行させつつ、抽籤用フラグとモード又は抽籤状態とに基づきCZ抽籤を行う。その結果、通常状態中にCZ抽籤に当籤すると遊技状態を通常状態から(CZ前兆を介して)CZに移行させ、CZ抽籤に当籤しないと遊技状態を通常状態のまま維持する。

10

【1498】

また、通常状態から直接ART状態に移行するパターンも、図209(B-1)(B-2)に示すように、2つのパターンが設けられている。直接ART状態に移行する一つ目のパターンは、図209(B-1)に示すように、主制御回路90は、通常状態中に抽籤用フラグと抽籤状態等とに基づきART抽籤を行い、その結果、ART抽籤に当籤すると、遊技状態を通常状態から(ART前兆及びART準備中を介して)ART状態に移行させ、ART抽籤に当籤しないと遊技状態を通常状態のまま維持する。

【1499】

一方、直接ART状態に移行する二つ目のパターンでは、主制御回路90は、ART抽籤を行うことなく、RT状態がRT3状態に移行するとART状態のストックを付与して、その後、遊技状態を通常状態から(ART前兆及びART準備中を介して)ART状態に移行させる。

20

【1500】

ここで、上述のようにRT3状態へは、無限RT状態中にRT3移行図柄である略称「チェリプ」に係る図柄組合せが表示されると、移行する。図205を参照すると、略称「チェリプ」に係る図柄組合せは、内部当籤役として「F__弱チェリプ」が遊技において表示され、その他の役が内部当籤役として決定された場合には表示されない。また、図180を参照すると、無限RT状態では、「F__弱チェリプ」はRT2状態中に低い確率(32/65536)で内部当籤役として決定され、その他の無限RT状態(RT0状態、RT4状態及びRT5状態)中は、内部当籤役として決定されることがないため、RT3状態へは、RT2状態からのみ移行することがあり、その他のRT状態から移行することがない(図178参照)。

30

【1501】

そのため、直接ART状態に移行する二つ目のパターンでは、主制御回路90は、RT状態がRT2状態からRT3状態に移行すると、遊技状態を通常状態からART状態に移行させることになる。なお、有限RT状態であるRT1状態中は、「F__弱チェリプ」が中程度の確率(874/65536)で内部当籤役として決定されるが(図180参照)、RT1状態中に略称「弱チェリプ」に係る図柄組合せが表示されても、RT1状態は有限RT状態であるため、RT3状態に移行することはない。

【1502】

また、RT3状態中は、「F__弱チェリプ」が高確率(18752/65536)で内部当籤役として決定される(図180参照)。そのため、RT3状態に移行できた場合には略称「弱チェリプ」に係る図柄組合せが頻繁に表示されることになる。遊技者にとってみれば、略称「弱チェリプ」に係る図柄組合せが頻繁に表示されることで、RT3状態中であること、すなわち、ART状態のストックが付与されていることについて期待感を抱くことになる。このように本実施形態のパチスロ1では、特定の図柄組合せが頻繁に表示されると、ART状態のストックという特典が付与されている期待感を抱くことができる。

40

【1503】

なお、RT3状態に移行したことに伴いART状態に移行する場合の、ART前兆の開始タイミングは任意である。すなわち、RT3状態に移行した時点で遊技状態を通常状態か

50

らART前兆に移行することとしてもよく、また、RT3状態中に任意のゲーム数が経過した時点で遊技状態を通常状態からART前兆に移行することとしてもよく、また、RT3状態からRT0状態に移行した時点で遊技状態を通常状態からART前兆に移行することとしてもよい。

【1504】

〔CZ中の遊技性〕

続いて、図210及び図211を参照して、CZ中の遊技の流れについて説明する。

【1505】

図210(A)に示すように、本実施形態のパチスロ1では、CZ中にART抽籤を行っており、このART抽籤に当籤すると、CZを一時中断してART状態に移行する。そして、その後ART状態が終了すると、中断していたCZを再開し、再びART抽籤を行う。このようにパチスロ1では、CZ中は、CZゲーム数が残っている限りCZとART状態とがループする。

【1506】

また、CZの最終ゲームにおいてART抽籤に非当籤の場合には、CZは終了することになるが、ART状態のセット数のストック状況によっては、CZの最終ゲーム終了時にストックが残っていることがある。パチスロ1では、CZの最終ゲームにおいてART抽籤に非当籤の場合であっても、ART状態のセット数のストックが有るときは、ストックを1つ放出し、ART状態に遊技状態に移行する。このとき、CZからART状態への移行に伴い、CZの残りゲーム数が最低5ゲーム延長される結果、その後、CZとART状態とがループすることになり、また、ART状態のセット数のストックが残っている限り、CZの最終ゲーム終了時にストックの放出が行われる結果、CZとART状態とがループすることになる。

【1507】

一方で、CZの最終ゲームにおいてART抽籤に非当籤であり、かつ、ART状態のセット数のストックがない場合には、CZが終了し、遊技状態が通常状態に移行する。このとき、本実施形態のパチスロ1では、CZ終了後の通常状態の1ゲーム目に限り、2ゲーム目以降よりも高い確率で当籤するART抽籤（泣きの1回）を行う。このCZ終了後の1ゲーム目に行うART抽籤に当籤すると、ART状態に遊技状態に移行し、非当籤すると、通常状態のまま遊技状態を維持する。また、このCZ終了後の1ゲーム目に行うART抽籤に当籤することによりART状態に移行した場合にも、CZの残りゲーム数が最低5ゲーム延長され、結果、その後、CZとART状態とがループすることになり、また、その後のCZの最終ゲームのストック放出や、泣きの1回のART抽籤が行われる結果、CZとART状態とがループすることになる。

【1508】

ここで、上述したように、本実施形態では、通常状態から移行したCZを初当りCZと呼び、CZとART状態とのループ中にART状態から戻った後のCZを継続CZと呼ぶ。続いて、初当りCZの概要と継続CZの概要とについて説明する。

【1509】

図210(B)に示すように、初当りCZ中は、CZの残りゲーム数に応じてART状態への移行契機が異なり、初当りCZは、CZの残りゲーム数が1以上の場合の抽籤フェーズAと、CZの残りゲーム数が0である場合の抽籤フェーズBと、から構成される。また、図210(C)に示すように、継続CZは、ART抽籤によるART状態への移行とストックの放出によるART状態への移行という移行契機を有しており、継続CZは、継続CZ中に行う抽籤フェーズAと、CZの残りゲーム数が0である場合の放出フェーズDと、から構成される。

【1510】

なお、CZ終了後の1ゲーム目には泣きの1回のART抽籤が行われるが、この泣きの1回のART抽籤を便宜上、抽籤フェーズCと呼ぶ（抽籤フェーズCは、通常状態中のART抽籤であり、CZを構成するものではない）。

【1511】

続いて、図211（D）を参照して、これら各フェーズの概要について説明する。抽籤フェーズA、B、Cは、それぞれ抽籤用フラグなどに基づいてART抽籤を行うフェーズであり、このART抽籤に当籤した場合には、遊技状態がART状態に移行するとともに、CZの残りゲーム数を最低5ゲーム延長（加算）する。一方で、このART抽籤に非当籤の場合には、CZゲーム数が残っている限りCZが継続し、CZゲーム数が残っていない場合には通常状態に移行する。なお、本実施形態のパチスロ1では、抽籤フェーズC（CZ終了後の泣きの1回）においてART抽籤に当籤した場合にも、CZの残りゲーム数を最低5ゲーム延長（加算）する。その結果、抽籤フェーズCでART抽籤に当籤した場合には、その後、CZとART状態とのループが継続することになる。

10

【1512】

また、放出フェーズDは、継続CZの最終ゲームにおいてART抽籤に非当籤であり、かつ、ART状態のセット数のストックがある場合に、保有するストックを1つ放出するフェーズである。放出フェーズDでは、ART状態のセット数のストックが残っている場合には、ART状態に遊技状態に移行するとともに、CZの残りゲーム数を最低5ゲーム延長（加算）する。一方で、放出フェーズDにおいて、ART状態のセット数のストックが残っていない場合には（CZの残りゲーム数が0であるため）、次遊技に、泣きの1回の抽籤フェーズCに基づくART抽籤が行われる。

【1513】

なお、本実施形態のパチスロ1では、抽籤フェーズA、B、Cにおいて一度の当籤でART状態のセット数を複数ストックすることがある。この場合、一つ目のストックのみがART抽籤の当籤に応じて放出され、残りのストックは、放出フェーズDで放出されることになる。

20

【1514】

続いて、図211（E）を参照して、継続CZの種別の決定方法について説明する。上述したようにパチスロ1では、継続CZとして、CZ（ART後）と特殊CZとを有する。主制御回路90は、CZからART状態に移行した際に（より詳細には、ランク決めART中に（後述の図255参照））、当該ART状態から戻った後の継続CZの種別を抽籤する。その結果、特殊CZに非当籤すると、ART状態から戻った後の継続CZがCZ（ART後）となり、特殊CZに当籤すると、ART状態から戻った後の継続CZが特殊CZとなる。

30

【1515】

ここで、CZは、CZゲーム数分継続することになるが、特殊CZに当籤し、継続CZが特殊CZになった場合、主制御回路90は、それまでの残りCZゲーム数を特殊CZの残りCZゲーム数としてセットする。特殊CZ中は、CZ（ART後）よりも高い確率でART抽籤に当籤するため、パチスロ1では、継続CZの種別の抽籤において特殊CZに当籤すると、残りCZゲーム数の価値が上がることになる。

【1516】

なお、特殊CZに当籤した場合の「それまでの残りCZゲーム数を特殊CZの残りCZゲーム数としてセットする」とは、それまでの残りCZゲーム数を特殊CZの残りCZゲーム数としてセットした場合に、それまでの残りCZゲーム数をクリア（0にする）することであってもよく、それまでの残りCZゲーム数を維持しつつ、当該CZゲーム数を特殊CZの残りCZゲーム数としてセットすることであってもよい。前者の場合には、それまでの残りCZゲーム数をクリアされるため、特殊CZ当籤時のCZ全体の残りゲーム数は、特殊CZのCZゲーム数だけになり、後者の場合には、それまでの残りCZゲーム数も維持されるため、特殊CZ当籤時のCZ全体の残りゲーム数は、通常（CZ（ART後））のCZゲーム数と、特殊CZのCZゲーム数との和になる。

40

【1517】

続いて、図211（F）は、特殊CZ中の流れを示す図である。上述したように、継続CZは、抽籤フェーズAと放出フェーズDとから構成される。CZとART状態とのループ

50

中に一度、特殊C Zに移行すると、図211(F)に示すように、その後の特殊C Zにおいて抽籤フェーズAのART抽籤に伴いART状態に移行している限りは、継続C Zが特殊C Zのまま維持される。なお、特殊C Z中に抽籤フェーズAのART抽籤に伴いART状態に移行している場合、C Zゲーム数の上乗せ（最低5ゲーム）は、特殊C Zの残りゲーム数に対して行われる。

【1518】

一方で、特殊C Zにおいて放出フェーズDのストック放出に伴いART状態に移行した場合には、特殊C Zが終了し、継続C ZがC Z（ART後）に書き換わる。このように特殊C Z中に放出フェーズDのストック放出に伴いART状態に移行した場合、C Zゲーム数の上乗せ（最低5ゲーム）は、通常（C Z（ART後））の残りC Zゲーム数に対して行われる。

10

【1519】

〔ART中の遊技性〕

続いて、図212及び図213を参照して、ART状態中の遊技の流れについて説明する。

【1520】

図212(A)に示すように、ART状態は、ランク決めARTと通常ARTとEPとから構成される。ランク決めARTは、1ゲームで完結する遊技状態であり、通常ARTは、30ゲームを1セットとする遊技状態であり、EPは、所謂上乗せ特化ゾーンであり、RT4状態に転落するまで継続する遊技状態である。ART状態中は、ARTゲーム数が残っている間、通常ARTからEPへの移行を目指して遊技を行うことになる。一方で、ARTゲーム数を消化してしまうと、遊技状態は、ART状態（通常ART）から継続C Zに移行することになる。

20

【1521】

通常ART中は、ART中の抽籤状態として「低確」「高確」「超高確」が設けられており、これらART中の抽籤状態に応じてEPへの移行し易さが制御される。図212(B)に示すように、ART中の抽籤状態は、ランク決めARTにおいて決めるランクに応じて一義的に定まる。ここで、図212(B)を参照して、ART中の抽籤状態、すなわち、ARTのランクの決定方法について説明する。

【1522】

図212(B)に示すように、主制御回路90は、ART抽籤の当籤時にARTのランクを仮決めする。ART当籤時のランクの仮決めは、ART抽籤に当籤した際の抽籤用フラグ（内部当籤役）に基づいて行われ、例えば、内部当籤役として決定される確率の低い抽籤用フラグ（所謂、強レア役）に基づいてART抽籤に当籤した場合には、高いランクが決定され易くなっている。

30

【1523】

上述のように、本実施形態のパチスロ1では、一度の当籤でART状態のセット数を複数ストックすることがある。主制御回路90は、一度の当籤でART状態のセット数を複数のストックした場合、それぞれのストックに対してランクを決定する。なお、後述するように、主制御回路90は、一つ目のストックについては、ART当籤時の抽籤用フラグに基づいてランクを決定し、二つ目以降のストックについては、抽籤用フラグを用いることなくランクを決定するが、これに限られるものではなく、二つ目以降のストックについても、ART当籤時の抽籤用フラグに基づいてランクを決定することとしてもよい。

40

【1524】

ART当籤時にそれぞれのストックに対してランクを決定（仮決め）すると、主制御回路90は、当該ストックの放出時に移行するランク決めARTにおいて、当籤時に仮決めしたランクの昇格抽籤を行い、ARTのランクを確定する。このランクの昇格抽籤は、1ゲームで完結するランク決めART中の抽籤用フラグに基づいて行われ、例えば、内部当籤役として決定される確率の低い抽籤用フラグ（所謂、強レア役）に基づいてART抽籤に当籤した場合には、ランクの昇格が行われ易くなっている。

【1525】

50

ランク決めART中の昇格抽籤は、「ランク維持」「ランク+1」「ランク+2」「ランク+3」の何れかが決定され、主制御回路90は、ART当籤時に仮決めしていたランクを昇格抽籤の抽籤結果に応じて更新する。なお、本実施形態では、ランクの最大は「ランク4」であるため、昇格抽籤の結果が「ランク4」以上である場合は「ランク4」とする。

【1526】

このように本実施形態のパチスロ1では、ART当籤時にARTのランクを仮決めしておき、ART状態の開始1ゲーム目（ランク決めART）に、仮決めしたランクからART中のランク（ART中の抽籤状態）を確定させる。言い換えると、本実施形態のパチスロ1では、ART当籤時は、ART中のランクとして幅を持たせたランクを決めておき、その後、ART状態の開始1ゲーム目に幅を持ったランクから具体的なランクを決定する。なお、ランク決めART中の昇格抽籤では、最低でも「ランク維持」が決定されるため、ART当籤時に仮決めされるランクは、本実施形態では、上方向のみの幅を持つことになる。もちろん、昇格抽籤においてランクを下げる抽籤結果（例えば、「ランク-1」など）を採用すれば、ART当籤時に仮決めされるランクに、下方向の幅も持たせることができる。

10

【1527】

ランク決めART中の昇格抽籤の結果、ART中のランクが確定すると、確定したランクからART中の抽籤状態が一義的に定まる。具体的には、確定したランクが「ランク1」である場合はART中の抽籤状態が「低確」となり、確定したランクが「ランク2」である場合はART中の抽籤状態が「高確」となり、確定したランクが「ランク3」である場合はART中の抽籤状態が「超高確」となり、確定したランクが「ランク4」である場合はART中の抽籤状態が「超高確」、かつ、EPのストックが1つ付与される。なお、確定したランクが「ランク4」である場合には、ランク決めARTの後にEPに移行することになる（図206の移行条件（E1）（E2）又は（E3）参照）。

20

【1528】

なお、本実施形態のパチスロ1では、ARTのランクや抽籤状態を示唆し、遊技者に対して、その後の遊技（1セットの通常ART）に興味を抱かせることとしている。この点、図212（C）を参照して、ARTのランクや抽籤状態を示唆する演出の概要について説明する。

【1529】

パチスロ1では、副制御回路200は、ART状態への移行時に武器ランクに応じた武器を表示することで、ART当籤時の幅を持ったランクを示唆する。具体的には、本実施形態では、武器ランクに応じた武器として、武器ランクが高い順に「武器S」「武器A」「武器B」「武器C」「武器D」に応じた映像データを有しており、ART当籤時のランクから武器ランクを決定し、当該武器ランクに応じた画像を表示する。なお、ART当籤時のランクと武器ランクとの対応関係の詳細は省略するが、副制御回路200は、ART当籤時のランクに応じた確率で、武器ランクを抽籤し、当籤した武器ランクに応じた映像を表示することで、ART当籤時ランクをART状態への移行時に示唆する。

30

【1530】

例えば、副制御回路200は、ART当籤時に「ランク1」が仮決定されている場合には、武器ランクの低い「武器C」又は「武器D」を高い確率で抽籤し、また、ART当籤時に「ランク4」が仮決定されている場合には、反対に武器ランクの高い「武器S」を高い確率で抽籤する。

40

【1531】

また、図212（C）に示すように、示唆するランクが不明な武器ランク「武器？」及び「武器？」に応じた映像データを設け、ART当籤時のランクを「武器？」で表示する（すなわち、ART当籤時のランクを示唆しない）こととしてもよい。

【1532】

この点、本実施形態のパチスロ1では、副制御回路200は、ART抽籤に当籤することを契機にART状態に移行（抽籤フェーズA，B，CからART状態に移行）する場合、

50

「保証なし」時は、「6 択転落リブ」の当籤時に自力で正解の押し順を当てなければ、（R T 状態が R T 4 状態に転落してしまう結果）E P から通常 A R T に遊技状態が移行してしまう。一方で、転落モード「ショート」又は「ロング」時は、「6 択転落リブ」の当籤時に正解の押し順が報知されるため、通常 A R T に転落することなく E P が継続する。

【1539】

後述するように、転落モード「ショート」又は「ロング」は、抽籤用フラグが「7 揃い」「B A R 揃い」又は「フェイク 7」である場合に、転落モード「保証なし」に移行することがあり、その他の抽籤用フラグである場合には、転落モード「保証なし」に移行することはない（後述の図 2 4 7 参照）。ただし、E P 中に抽籤用フラグが一度「7 揃い」又は「B A R 揃い」となるまでは、転落モード「保証なし」に移行することはない（すなわち、E P に移行すると、抽籤用フラグ「7 揃い」又は「B A R 揃い」に基づく A R T 状態のストックの付与が少なくとも一度は行われる）。また、後述するように、転落モードは、所謂レア役や転落モード「保証なし」時の自力の押し順正解時に昇格することもある（後述の図 2 4 7 参照）。

【1540】

〔B B 中の遊技性〕

続いて、図 2 1 4 を参照して、B B 中の遊技の流れについて説明する。図 2 1 4 に示すように、B B 中において主制御回路 9 0 は、抽籤用フラグが「7 揃い」又は「B A R 揃い」である場合に、各種ストックを付与し、また、抽籤用フラグが「ベルはずれ A」又は「ベルはずれ B」である場合に各種抽籤を行い、当籤すると各種ストックを付与し、非当籤するとストックを付与しない。なお、抽籤用フラグ「7 揃い」又は「B A R 揃い」に基づく各種ストックの付与は、C B B 中に行い、また、抽籤用フラグ「ベルはずれ A」又は「ベルはずれ B」に基づく各種抽籤は、N B B 中に行う。また、主制御回路 9 0 は、B B 中に規定枚数のメダルが払い出されると、B B を終了する。

【1541】

ここで、抽籤用フラグ「ベルはずれ A」又は「ベルはずれ B」に基づく各種抽籤の種別、及び、抽籤用フラグ「7 揃い」又は「B A R 揃い」に基づき付与するストックの種別は、状態に応じて異なる。具体的には、主制御回路 9 0 は、通常 B B（図 2 0 6 参照）中は、抽籤用フラグ「ベルはずれ A」又は「ベルはずれ B」に基づき C Z 抽籤及び A R T 抽籤を行い、また、抽籤用フラグ「7 揃い」又は「B A R 揃い」に基づき A R T 状態のセット数を付与（ストック）する。また、主制御回路 9 0 は、A R T 中 B B（図 2 0 6 参照）中は、抽籤用フラグ「ベルはずれ A」又は「ベルはずれ B」に基づき A R T 抽籤及び E P の付与抽籤を行い、また、抽籤用フラグ「7 揃い」又は「B A R 揃い」に基づき A R T 状態のセット数及び E P を付与（ストック）する（後述の図 2 4 9 参照）。なお、A R T 中 B B において、E P が付与された場合には、B B 終了後に A R T 準備中（及び必要に応じてランク決め A R T）を経由して、E P 準備中に移行する。

【1542】

＜各種データテーブル＞

続いて、図 2 1 5 ～図 2 4 9 を参照して、メイン R O M 1 0 2 に記憶される各種のデータテーブルについて説明する。なお、主制御回路 9 0 は、0 ～2 5 5 の範囲の乱数値（すなわち、確率分母 2 5 6）を用いて以下に示すデータテーブルを用いた抽籤を行う。

【1543】

〔モード移行抽籤テーブル〕

図 2 1 5 は、通常状態中のモードを決定するために用いるモード移行抽籤テーブルである。上述したように通常状態中は、抽籤用フラグが「ベル」である場合に現在のモードに基づき C Z 抽籤を行う（図 2 0 9（A-1）参照）。モード移行抽籤テーブルは、C Z 抽籤に用いる通常状態中のモードを決定するためのテーブルであり、図 2 1 5（A）は、N B B の終了時にモードを決定するためのモード移行抽籤テーブルであり、図 2 1 5（B）は、C B B の終了時にモードを決定するためのモード移行抽籤テーブルであり、図 2 1 5（C）は、A R T 状態に移行することなく終了した初当り C Z の終了（初当り C Z から通常

状態に移行)時にモードを決定するためのモード移行抽籤テーブルであり、図215(D)は、継続CZの終了(継続CZから通常状態に移行)時にモードを決定するためのモード移行抽籤テーブルであり、図215(E)は、設定変更時にモードを決定するためのモード移行抽籤テーブルである。

【1544】

モード移行抽籤テーブルは、移行先のモード(抽籤結果)を決定するための抽籤値の情報を規定する。主制御回路90は、モード移行抽籤テーブルを参照して、移行先のモードを決定する。なお、NBB、CBB又は初当りCZの終了時は、移行前のモード(モード1~4)に応じて移行先のモードを決定し(図215(A)~(C)参照)、継続CZの終了時又は設定変更時は、移行前のモードに関係なく移行先のモードを決定する(図215(D),(E)参照)。

10

【1545】

また、モード移行抽籤テーブルで規定される抽籤値は、図215(A)~(E)に示すように、設定値の偶奇情報に応じて異なる。本実施形態のパチスロ1では、設定値(遊技者にとっての有利度合)として、「1」、「2」、「5」及び「6」の4段階の設定値が設けられている。なお、内部的には、設定値「1」、「2」、「5」及び「6」はそれぞれ、値「0」、「1」、「2」及び「3」(以下、「設定値の内部値」という)で管理される。ただし、設定値の内部値(「0」~「3」)は、本発明に係る設定値に関する情報の一具体例を示すものである。

【1546】

20

本実施形態のモード移行抽籤テーブルでは、設定値が偶数設定(「2」又は「6」)である場合におけるモードの昇格確率(移行前のモードより移行先のモードが高くなる確率)は、設定値が奇数設定(「1」又は「5」)である場合のそれにより高くなる。なお、設定値の偶奇情報は、後述するように、予め、設定変更時において設定値の内部値(「0」~「3」)に対して所定の演算を施すことにより算出され、メインRAM103内の所定領域(後述の設定値偶奇格納領域)に格納されている(後述の図282及び図283参照)。

【1547】

ここで、図216に、ソースプログラム上において実際に参照される、図215(D)の継続CZ終了時用のモード移行抽籤テーブル及び図215(E)の設定変更時用のモード移行抽籤テーブルの一構成例を示す。なお、図216に示すモード移行抽籤テーブルでは、図215(D)に示す継続CZ終了時用のモード移行抽籤テーブルと図215(E)に示す設定変更時用のモード移行抽籤テーブルとが一つのテーブルで構成される。

30

【1548】

ソースプログラム上において実際に参照される、図216のモード移行抽籤テーブル内では、データ「.LOW.wDDMPAR_X」が遊技状況(継続CZ終了時又は設定変更時)に関するパラメータであり、データ「.LOW.wEVN_ODD」が設定値の偶奇情報である。そして、このモード移行抽籤テーブルを用いた抽籤処理では、データ「.LOW.wDDMPAR_X」(遊技状況)及びデータ「.LOW.wEVN_ODD」(設定値の偶奇情報)を含む各種抽籤選択情報(モード移行抽籤テーブルの先頭アドレスから8バイト先のアドレスまでの領域に格納されている各種データ)に基づいて、所定の抽籤値群が選択される。

40

【1549】

例えば、遊技状況が設定変更時であり且つ設定値が奇数設定(「1」又は「5」)である場合には、データ「(80H)+dVCMD1_16-\$」(遊技状況が設定変更時であり且つ設定値が奇数設定である場合に使用される抽籤値群の格納領域の先頭アドレス)を参照し、ラベル「dVCMD_16」が付された先頭アドレスから2バイト先のアドレスまでの領域(3バイトの領域)に格納されている抽籤値群(この例では、抽籤値「16」及び「64」)が選択される。なお、本実施形態では、遊技状況が設定変更時であり且つ設定値が奇数設定である場合のモード移行抽籤において、モード2及びモード3に非当籤

50

となると、自動的にモード1が当籤することになる（図215（E）参照）ので、このときには、モード1の当否を抽籤により決定する必要がない。それゆえ、ラベル「d V C M D__16」が付された先頭アドレスから始まる抽籤値群の格納領域には、モード1に対する抽籤値が格納されない。

【1550】

〔状態移行抽籤テーブル〕

図217及び図218は、通常状態中の抽籤状態を決定するために用いる状態移行抽籤テーブルである。通常状態中は、抽籤用フラグが「ベル」以外である場合に現在の抽籤状態及び抽籤用フラグに基づきCZ抽籤を行う（図209（A-2）参照）。状態移行抽籤テーブルは、CZ抽籤に用いる通常状態中の抽籤状態を決定するためのテーブルであり、図217（A）は、現在の抽籤状態が「低確」である場合に移行先の抽籤状態を決定するための状態移行抽籤テーブルであり、図217（B）は、現在の抽籤状態が「高確」であり、かつ、後述する高確保証ゲーム数が10未満である場合に移行先の抽籤状態を決定するための状態移行抽籤テーブルであり、図217（C）は、現在の抽籤状態が「高確」であり、かつ、後述する高確保証ゲーム数が10以上である場合に移行先の抽籤状態を決定するための状態移行抽籤テーブルであり、図218（D）は、現在の抽籤状態が「超高確」である場合に移行先の抽籤状態を決定するための状態移行抽籤テーブルである。また、図218（E）は、BB終了時に移行先の抽籤状態を決定するための状態移行抽籤テーブルであり、図218（F）は、設定変更時に移行先の抽籤状態を決定するための状態移行抽籤テーブルであり、図218（G）は、CZ終了時に移行先の抽籤状態を決定するための状態移行抽籤テーブルである。

【1551】

状態移行抽籤テーブルは、移行先の抽籤状態（抽籤結果）を決定するための抽籤値の情報を規定する。主制御回路90は、状態移行抽籤テーブルを参照して、移行先の抽籤状態を決定する。なお、移行先の抽籤状態として「高確A」又は「高確B」が決定された場合、後述する高確保証ゲーム数抽籤テーブル（後述の図219）を参照して、高確保証ゲーム数が付与される。この高確保証ゲーム数が「1」以上である場合、図217（B）に示す状態移行抽籤テーブルにおいて、移行先の抽籤状態として「低確」が決定されても、抽籤状態は移行せずに「高確」のまま維持される。一方で、移行先の抽籤状態として「高確保証なし」が決定された場合には、移行先の抽籤状態は「高確」になるものの、高確保証ゲーム数は付与されない。

【1552】

また、高確保証ゲーム数が残っている状態で移行先の抽籤状態として「超高確」が決定された場合、高確保証ゲーム数は、クリア（0にする）こととしてもよく、また、保持することとしてもよい。なお、図218（D）に示す「超高確」時用の状態移行抽籤テーブルによると、移行先の抽籤状態として「低確」が決定されることがあるが、高確保証ゲーム数を保持する場合には、移行前の「超高確」から移行先の抽籤状態として「低確」が決定されても、「低確」に移行させることなく「超高確」のまま維持することとしてもよく、また、「低確」の代わりに「高確」に移行させることとしてもよい。

【1553】

図217（A）～図218（D）を参照すると、通常状態中に毎ゲーム参照される状態移行抽籤テーブルでは、移行前の抽籤状態と抽籤用フラグとに基づいて移行先の抽籤状態が決定され、また、他の遊技状態から通常状態に移行した時などに参照される状態移行抽籤テーブル（図218（E）～（G））では、抽籤用フラグを用いることなく移行先の抽籤状態が決定される。

【1554】

〔高確保証ゲーム数抽籤テーブル〕

図219は、移行先の抽籤状態として「高確A」又は「高確B」が決定された場合に、高確保証ゲーム数を決定するために用いる高確保証ゲーム数抽籤テーブルである。高確保証ゲーム数抽籤テーブルは、移行先の抽籤状態として決定された高確の種別毎に、付与する

高確保証ゲーム数の抽籤結果についての抽籤値の情報を規定する。主制御回路 90 は、移行先の抽籤状態として「高確 A」又は「高確 B」を決定すると、高確保証ゲーム数抽籤テーブルを参照して、高確保証ゲーム数を決定し、付与する。

【1555】

〔モード別 C Z 抽籤テーブル〕

図 220 は、通常状態中にモードに基づき行う C Z 抽籤に用いるモード別 C Z 抽籤テーブルである。モード別 C Z 抽籤テーブルは、現在のモード毎に、C Z 抽籤の抽籤結果についての抽籤値の情報を規定する。主制御回路 90 は、通常状態において抽籤用フラグが「ベル」である場合、モード別 C Z 抽籤テーブルを参照して、現在のモードに基づき C Z 抽籤を行い、この C Z 抽籤に当籤した場合には、遊技状態を（C Z 前兆を介して）C Z に移行させる。なお、同遊技において行う通常状態中の A R T 抽籤に当籤している場合には、C Z 抽籤は行わない。

10

【1556】

また、モード別 C Z 抽籤テーブルで規定される抽籤値は、図 220 に示すように、設定値の高低情報に応じて異なる。本実施形態のモード別 C Z 抽籤テーブルでは、設定値が高設定（「5」又は「6」）である場合における C Z 抽籤の当籤確率は、設定値が低設定（「1」又は「2」）である場合のそれにより若干高くなる。具体的には、設定値が高設定（「5」又は「6」）である場合におけるモード 2 時の C Z 抽籤の当籤確率が、設定値が低設定（「1」又は「2」）である場合のそれにより若干高くなる。その他のモード時では、C Z 抽籤の当籤確率は、設定値の高低に応じて変化しない。なお、設定値の高低情報は、後述するように、予め、設定変更時において設定値の内部値（「0」～「3」）に対して所定の演算を施すことにより算出され、メイン R A M 103 内の特定領域（後述の設定値高低格納領域）に格納されている（後述の図 282 及び図 283 参照）。

20

【1557】

ここで、図 221 に、ソースプログラム上において実際に参照されるモード別 C Z 抽籤テーブルの一構成例を示す。

【1558】

ソースプログラム上において実際に参照される図 221 のモード別 C Z 抽籤テーブル内では、データ「.LOW.wLOW_H I H」が設定値の高低情報であり、データ「.LOW.wVC__MODE」が現在のモード情報である。そして、このモード別 C Z 抽籤テーブルを用いた抽籤処理では、データ「.LOW.wLOW_H I H」（高低情報）及びデータ「.LOW.wVC__MODE」（現在のモード情報）を含む各種抽籤選択情報（モード別 C Z 抽籤テーブルの先頭アドレスから 4 バイト先のアドレスまでの領域に格納されている各種データ）に基づいて、所定の抽籤値群が選択される。

30

【1559】

例えば、設定値が高設定（「5」又は「6」）であり且つ現在のモードがモード 3 である場合には、まず、データ「(80H)+dNRMVC__H-\$」（設定値が高設定である場合に使用される抽籤値群の格納領域の先頭アドレス）を参照し、ラベル「dNRMVC__H」で示される先頭アドレスから 3 バイト先のアドレスまでの領域（4 バイトの領域）に格納されている抽籤値群が選択される。次いで、その選択された抽籤値群の中からモード 3 に対応するデータ「cPRB__6」が選択される。なお、データ「cPRB__6」は、抽籤値「6」が格納されている領域のアドレスに対応付けられたラベルデータであり、このデータが選択されると、抽籤値「6」が取得される。

40

【1560】

〔B B 当籤時状態別 C Z 抽籤テーブル及び状態別 C Z 抽籤テーブル〕

続いて、図 222 及び図 223 は、通常状態中に抽籤状態に基づき行う C Z 抽籤に用いる状態別 C Z 抽籤テーブルであり、図 222 は、通常状態中の B B 当籤時に B B と重複して当籤した当籤役に応じた抽籤用フラグを参照して C Z 抽籤を行うための B B 当籤時状態別 C Z 抽籤テーブルであり、図 223 は、通常状態中の B B 非当籤時に抽籤用フラグを参照して C Z 抽籤を行うための状態別 C Z 抽籤テーブルである。

50

【1561】

図222のBB当籤時状態別CZ抽籤テーブルは、現在の抽籤確率毎（低確、高確、超高確）に設けられ、抽籤用フラグ毎にCZ抽籤の抽籤結果についての抽籤値の情報を規定する。また、図223の状態別CZ抽籤テーブルは、現在の抽籤確率と設定値の高低情報との組み合わせ毎に設けられ、抽籤用フラグ毎にCZ抽籤の抽籤結果についての抽籤値の情報を規定する。主制御回路90は、通常状態中に、現在の抽籤確率（抽籤状態）及び抽籤用フラグ及び設定値に基づいてCZ抽籤を行い、このCZ抽籤に当籤した場合には、遊技状態を（CZ前兆を介して）CZに移行させる。なお、同遊技において行う通常状態中のART抽籤に当籤している場合には、CZ抽籤は行わない。

【1562】

なお、図223の状態別CZ抽籤テーブルを用いたCZ抽籤において、設定値が高設定（「5」又は「6」）であり且つ抽籤用フラグが「強スイカ」、「強レア（強チェリー、強チャンスリップ又は強ベル）」である場合におけるCZ抽籤の当籤確率は、設定値が低設定（「1」又は「2」）であり且つ抽籤用フラグが「強スイカ」、「強レア」である場合のそれより高くなる。

【1563】

[CZ前兆ゲーム数抽籤テーブル]

続いて、図224は、通常状態中のCZ抽籤に当籤した際に、通常状態からCZに移行するまでの期間（CZ前兆ゲーム数）を抽籤するために用いるCZ前兆ゲーム数抽籤テーブルである。CZ前兆ゲーム数抽籤テーブルは、CZ抽籤に当籤した際のBB当籤の有無に応じてCZ前兆ゲーム数の範囲（抽籤結果）についての抽籤値の情報を規定する。主制御回路90は、通常状態中にCZ抽籤に当籤すると、CZ前兆ゲーム数の抽籤を行い、当籤したCZ前兆ゲーム数をセットし、遊技状態をCZ前兆に移行する。

【1564】

なお、CZ前兆ゲーム数抽籤テーブルでは、CZ前兆ゲーム数の抽籤結果として一定の範囲を持たせているが、主制御回路90は、当籤した範囲の中から均等にCZ前兆ゲーム数を決定する。また、BB当籤時にCZ抽籤に当籤していた場合には、主制御回路90は、BBの終了後に当籤したCZ前兆ゲーム数をセットする。すなわち、通常状態中にCZ抽籤に当籤し、かつ、CZ前兆ゲーム数として「1」以上が決定された場合、BB非当籤時のCZ当籤時は、その後、通常状態からCZ前兆に遊技状態を移行し、また、BB当籤時のCZ当籤時は、その後、BBが終了した後にCZ前兆に遊技状態を移行する。

【1565】

[BB当籤時ART抽籤テーブル及び通常時ART抽籤テーブル]

続いて、図225及び図226は、通常状態中に抽籤用フラグ等に基づき行うART抽籤に用いるART抽籤テーブルであり、図225は、通常状態中のBB当籤時にBBと重複して当籤した当籤役に応じた抽籤用フラグを参照してART抽籤を行うためのBB当籤時ART抽籤テーブルであり、図226は、通常状態中のBB非当籤時に抽籤用フラグを参照してART抽籤を行うための通常時ART抽籤テーブルである。

【1566】

なお、BB当籤時は、NBB当籤、かつ、抽籤状態が「通常」又は「高確」である場合に、図225（A）に示すBB当籤時ART抽籤テーブルが参照され、また、CBB当籤、かつ、抽籤状態が「通常」又は「高確」である場合に、図225（B）に示すBB当籤時ART抽籤テーブルが参照され、また、抽籤状態が「超高確率」又は遊技状態が「CZ前兆」「ART前兆中」の何れかである場合に、BB（CBB又はNBB）の種別に関わらず、図225（C）に示すBB当籤時ART抽籤テーブルが参照される。

【1567】

また、BB非当籤時において、抽籤状態が「低確」又は「高確」であり且つ設定値が低設定（「1」又は「2」）である場合には、図226（A）に示す通常時ART抽籤テーブルが参照され、また、抽籤状態が「超高確率」又は遊技状態が「CZ前兆」及び「ART前兆中」のいずれかであり且つ設定値が低設定である場合には、図226（B）に示す通

10

20

30

40

50

常時ART抽籤テーブルが参照され、また、通常フラグ間中であり且つ設定値が低設定である場合には、図226(C)に示す通常時ART抽籤テーブルが参照される。さらに、BB当籤時において、抽籤状態が「低確」又は「高確」であり且つ設定値が高設定(「5」又は「6」)である場合には、図226(D)に示す通常時ART抽籤テーブルが参照され、また、抽籤状態が「超高確率」又は遊技状態が「CZ前兆」及び「ART前兆中」のいずれかであり且つ設定値が高設定である場合には、図226(E)に示す通常時ART抽籤テーブルが参照され、また、通常フラグ間中であり且つ設定値が高設定である場合には、図226(F)に示す通常時ART抽籤テーブルが参照される。

【1568】

BB当籤時ART抽籤テーブルは、現在の抽籤状態や遊技状態に応じて設けられ、抽籤用フラグ毎にART抽籤の抽籤結果についての抽籤値の情報を規定する。また、通常時ART抽籤テーブルは、現在の抽籤状態や遊技状態と設定値の高低情報との組み合わせ毎に設けられ、抽籤用フラグ毎にART抽籤の抽籤結果についての抽籤値の情報を規定する。なお、主制御回路90は、通常状態中に抽籤用フラグ等に基づいてART抽籤を行い、このART抽籤に当籤した場合には、遊技状態を(ART前兆を介して)ART状態に移行させる。また、本実施形態の通常時ART抽籤テーブルを用いたART抽籤では、設定値が高設定であり且つ抽籤用フラグが「強スイカ」、「強レア(強チェリー、強チャンスリブ又は強ベル)」である場合におけるART抽籤の当籤確率は、設定値が低設定であり且つ抽籤用フラグが「強スイカ」、「強レア」である場合のそれより高くなる。

【1569】

【ART前兆ゲーム数抽籤テーブル】

続いて、図227は、通常状態中のART抽籤に当籤した際に、通常状態からART状態に移行するまでの期間(ART前兆ゲーム数)を抽籤するために用いるART前兆ゲーム数抽籤テーブルである。ART前兆ゲーム数抽籤テーブルは、ART前兆ゲーム数の範囲(抽籤結果)についての抽籤値の情報を規定する。主制御回路90は、通常状態中にART抽籤に当籤すると、ART前兆ゲーム数の抽籤を行い、当籤したART前兆ゲーム数をセットし、遊技状態をART前兆に移行する。

【1570】

なお、ART前兆ゲーム数抽籤テーブルでは、ART前兆ゲーム数の抽籤結果として一定の範囲を持たせているが、主制御回路90は、当籤した範囲の中から均等にART前兆ゲーム数を決定する。なお、BB当籤時や通常フラグ間中にART抽籤に当籤した場合には、遊技状態がART中フラグ間に移行し、ART中BB終了後にART準備中に移行することから(図206参照)、この場合、主制御回路90は、ART前兆ゲーム数を抽籤しない。すなわち、主制御回路90は、図226(A)(B)に示す通常時ART抽籤テーブルを用いたART抽籤に当籤した場合に、ART前兆ゲーム数を抽籤し、図225(A)～(C)に示すBB当籤時ART抽籤テーブル、及び図226(C)に示す通常時ART抽籤テーブルを用いたART抽籤に当籤した場合には、ART前兆ゲーム数を抽籤しない。

【1571】

【CZ中ART抽籤テーブル】

続いて、図228及び図229は、CZ中のART抽籤に用いるCZ中ART抽籤テーブルである。図228(A)は、初当たりCZ、かつ、CZの残りゲーム数が1以上である場合に参照されるCZ中ART抽籤テーブルであり、図228(B)は、初当たりCZ、かつ、CZの残りゲーム数が0である場合に参照されるCZ中ART抽籤テーブルであり、図228(C)は、初当たりCZの終了の次遊技(泣きの1回)に参照されるCZ中ART抽籤テーブル(なお、初当たりCZの終了の次遊技は、詳細には通常状態中である)である。また、図228(D)は、CZ(ART後)、かつ、非ナビ高確時に参照されるCZ中ART抽籤テーブルであり、図229(E)は、CZ(ART後)、かつ、ナビ高確時に参照されるCZ中ART抽籤テーブルであり、図229(F)は、CZ(ART後)の終了の次遊技(泣きの1回)に参照されるCZ中ART抽籤テーブル(なお、CZ(ART後

10

20

30

40

50

）の終了の次遊技は、詳細には通常状態中である）であり、図 229（G）は、特殊 C Z、かつ、非ナビ高確時に参照される C Z 中 A R T 抽籤テーブルであり、図 229（H）は、特殊 C Z、かつ、ナビ高確時に参照される C Z 中 A R T 抽籤テーブルである。

【1572】

C Z 中 A R T 抽籤テーブルは、C Z の種別など毎に設けられ、抽籤用フラグ毎に A R T 抽籤の抽籤結果についての抽籤値の情報を規定する。主制御回路 90 は、C Z 中に抽籤用フラグ等に基づいて A R T 抽籤を行い、この A R T 抽籤に当籤した場合には、遊技状態を（必要に応じて A R T 準備中を介して）A R T 状態に移行させる。なお、C Z 中の B B 当籤時は、図 228 及び図 229 に示す C Z 中 A R T 抽籤テーブルではなく、後述の A R T 中 B B 当籤時ストック抽籤テーブル（後述の図 233（D））を参照して、A R T 抽籤（必ず当籤）を行う。

10

【1573】

〔A R T 中ストック抽籤テーブル〕

続いて、図 230～図 232 は、A R T 状態中に行う A R T 状態のセット数の上乗せ抽籤に用いる A R T 中ストック抽籤テーブルである。図 230（A）は、通常 A R T、かつ、A R T 中の抽籤状態「低確」又は「高確」、かつ、非ナビ高確中に参照される A R T 中ストック抽籤テーブルであり、図 230（B）は、通常 A R T、かつ、A R T 中の抽籤状態「低確」又は「高確」、かつ、ナビ高確中に参照される A R T 中ストック抽籤テーブルであり、図 230（C）は、通常 A R T、かつ、A R T 中の抽籤状態「超高確」、かつ、非ナビ高確中に参照される A R T 中ストック抽籤テーブルであり、図 230（D）は、通常 A R T、かつ、A R T 中の抽籤状態「超高確」、かつ、ナビ高確中に参照される A R T 中ストック抽籤テーブルである。また、図 231（E）は、E P 中、かつ、転落モード「保証なし」時に参照される A R T 中ストック抽籤テーブルであり、図 231（F）は、E P 中、かつ、転落モード「ショート」時に参照される A R T 中ストック抽籤テーブルであり、図 231（G）は、E P 中、かつ、転落モード「ロング」時に参照される A R T 中ストック抽籤テーブルである。また、図 232（H）は、ランク決め A R T 又は E P 準備中、かつ、非ナビ高確中に参照される A R T 中ストック抽籤テーブルであり、図 232（I）は、ランク決め A R T 又は E P 準備中、かつ、ナビ高確中に参照される A R T 中ストック抽籤テーブルである。また、図 232（J）は、A R T 準備中に参照される A R T 中ストック抽籤テーブルであり、図 232（K）は、A R T 中フラグ間に参照される A R T 中ストック抽籤テーブルである。

20

30

【1574】

A R T 中ストック抽籤テーブルは、現在の遊技状態など毎に設けられ、抽籤用フラグ毎に A R T 状態のセット数の上乗せ抽籤の抽籤結果についての抽籤値の情報を規定する。主制御回路 90 は、対応する遊技状態中に抽籤用フラグ等に基づいて上乗せ抽籤を行い、この A R T 抽籤に当籤した場合には、A R T 状態のセット数を所定数だけ上乗せする。

【1575】

〔A R T 中 B B 当籤時ストック抽籤テーブル〕

続いて、図 233 は、A R T 状態中の B B 当籤時に行う A R T 状態のセット数の上乗せ抽籤に用いる A R T 中 B B 当籤時ストック抽籤テーブルである。図 233（A）は、通常 A R T 中の抽籤状態「低確」若しくは「高確」時、又は、E P 中の転落モード「保証なし」時に参照される A R T 中 B B 当籤時ストック抽籤テーブルであり、図 233（B）は、通常 A R T 中の抽籤状態「超高確」時、又は、E P 中の転落モード「ショート」時に参照される A R T 中 B B 当籤時ストック抽籤テーブルであり、図 233（C）は、ランク決め A R T 中、又は、E P 準備中に参照される A R T 中 B B 当籤時ストック抽籤テーブルであり、図 233（D）は、C Z 若しくは C Z 終了の次遊技、又は、E P 中の転落モード「ロング」時に参照される A R T 中 B B 当籤時ストック抽籤テーブルであり、図 233（E）は、A R T 準備中に参照される A R T 中 B B 当籤時ストック抽籤テーブルである。

40

【1576】

A R T 中 B B 当籤時ストック抽籤テーブルは、現在の遊技状態など毎に設けられ、抽籤用

50

フラグ毎にART状態のセット数の上乗せ抽籤の抽籤結果についての抽籤値の情報を規定する。主制御回路90は、対応する遊技状態中に抽籤用フラグ等に基づいて上乗せ抽籤を行い、このART抽籤に当籤した場合には、ART状態のセット数を所定数だけ上乗せする。

【1577】

〔ART当籤時ストック数抽籤テーブル〕

続いて、図234は、ART抽籤や、ART状態のセット数の上乗せ抽籤に当籤した場合（すなわち、ART当籤時）に、付与するART状態のセット数を抽籤するために用いるART当籤時ストック数抽籤テーブルである。ART当籤時ストック数抽籤テーブルは、ART当籤時の当籤契機毎に、付与するART状態のセット数についての抽籤値の情報を規定する。主制御回路90は、ART当籤時に対応する当籤契機に基づいてART状態のセット数を抽籤し、当籤したセット数を付与する。

10

【1578】

なお、当籤契機「初当り」とは、通常状態中にART抽籤に当籤した場合（通常状態中にRT3状態への移行に伴うART状態のストック付与を含む）、ART前兆中にART抽籤に当籤した場合、初当りCZ中にART抽籤に当籤した場合、通常BBフラグ間中にART抽籤に当籤した場合、又は通常BB中にART抽籤に当籤した場合の何れかをいい、また、当籤契機「上乗せA」とは、EP中にART状態のセット数の上乗せ抽籤に当籤した場合をいい、また、当籤契機「上乗せB」とは、継続CZ中にART抽籤に当籤した場合をいい、また、当籤契機「上乗せC」とは、当籤契機「初当り」「上乗せA」「上乗せB」以外の状況でART抽籤や、ART状態のセット数の上乗せ抽籤に当籤した場合をいう。

20

【1579】

また、ART当籤時ストック数抽籤テーブルで規定される抽籤値は、図234に示すように、設定値の高低情報に応じて異なる。本実施形態のART当籤時ストック数抽籤テーブルでは、設定値が高設定（「5」又は「6」）である場合に、より多くのART状態のセット数が当籤するような構成になっている。具体的には、当籤契機が「初当り」時であり且つ設定値が高設定（「5」又は「6」）である場合に3セットのART状態が上乗せされる確率は、当籤契機が「初当り」時であり且つ設定値が低設定（「1」又は「2」）である場合のそれより高くなる。また、その他の当籤契機では、高設定時における2セットのART状態の上乗せ確率が、低設定時のそれより高くなる。

30

【1580】

〔特殊CZ移行抽籤テーブル〕

続いて、図235は、継続CZとして特殊CZを用いるか否かを抽籤するために用いる特殊CZ移行抽籤テーブルである。特殊CZ移行抽籤テーブルは、ART状態のセット数のストックの有無毎に、抽籤結果についての抽籤値の情報を規定する。主制御回路90は、ランク決めARTの開始時にART状態のセット数のストックの有無に応じて特殊CZを用いるか否かを抽籤し、当籤した場合には、次の継続CZを特殊CZとする一方で、非当籤した場合には、次の継続CZをCZ（ART後）とする。

【1581】

なお、主制御回路90は、次の継続CZを特殊CZとした後は、特殊CZが終了するまで特殊CZ移行抽籤テーブルを用いた抽籤を行わない。また、ART状態のセット数のストックがない状況で、本抽籤に当籤した場合には、主制御回路90は、ART状態のセット数を1つ付与する。

40

【1582】

〔ART当籤時ランク抽籤テーブル〕

続いて、図236～図240は、ART当籤時に当籤したART状態のランクを決定するために用いるART当籤時ランク抽籤テーブルである。なお、ART当籤時には、ART状態のセット数が複数付与されることがあるが、2つ目以降のART状態のランクは、図240（O）に示すART当籤時ランク抽籤テーブルにより決定され、図236（A）～

50

図 2 4 0 (N) に示す A R T 当籤時ランク抽籤テーブルは、1 つ目の A R T 状態のランクを決定するために用いられる。

【1 5 8 3】

図 2 3 6 (A) は、A R T 当籤の当籤タイミングが通常状態、C Z 前兆、A R T 前兆、又は E P 以外の A R T 状態中であり、かつ、B B 非当籤時に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 3 6 (B) は、A R T 当籤の当籤タイミングが通常状態、C Z 前兆、A R T 前兆、又は E P 以外の A R T 状態中であり、かつ、N B B の当籤時に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 3 6 (C) は、A R T 当籤の当籤タイミングが通常状態、C Z 前兆、A R T 前兆、又は E P 以外の A R T 状態中であり、かつ、C B B の当籤時に用いられる A R T 当籤時ランク抽籤テーブルである。

10

【1 5 8 4】

また、図 2 3 7 (D) は、A R T 当籤の当籤タイミングが A R T 準備中であり、かつ、B B 非当籤時に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 3 7 (E) は、A R T 当籤の当籤タイミングが A R T 準備中であり、かつ、N B B 当籤時に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 3 7 (F) は、A R T 当籤の当籤タイミングが A R T 準備中であり、かつ、C B B 当籤時に用いられる A R T 当籤時ランク抽籤テーブルである。

【1 5 8 5】

また、図 2 3 8 (G) は、A R T 当籤の当籤タイミングが E P 中であり、かつ、B B 非当籤時に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 3 8 (H) は、A R T 当籤の当籤タイミングが E P 中であり、かつ、N B B 当籤時に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 3 8 (I) は、A R T 当籤の当籤タイミングが E P 中であり、かつ、C B B 当籤時に用いられる A R T 当籤時ランク抽籤テーブルである。

20

【1 5 8 6】

また、図 2 3 9 (J) は、A R T 当籤の当籤タイミングが C Z 中であり、かつ、B B 非当籤時に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 3 9 (K) は、A R T 当籤の当籤タイミングが C Z 中であり、かつ、N B B 当籤時に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 3 9 (L) は、A R T 当籤の当籤タイミングが C Z 中であり、かつ、C B B 当籤時に用いられる A R T 当籤時ランク抽籤テーブルである。

【1 5 8 7】

また、図 2 4 0 (M) は、A R T 当籤の当籤タイミングがフラグ間状態（通常フラグ間又は A R T 中フラグ間）である場合に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 4 0 (N) は、A R T 当籤の当籤タイミングが B B 中である場合に用いられる A R T 当籤時ランク抽籤テーブルであり、図 2 4 0 (O) は、複数ストック時の 2 つ目以降の A R T 状態のランクを決定するために用いられる A R T 当籤時ランク抽籤テーブルである。

30

【1 5 8 8】

A R T 当籤時ランク抽籤テーブルは、A R T 当籤時の遊技状態毎に設けられ、A R T 当籤時の抽籤用フラグ毎に、抽籤結果についての抽籤値の情報を規定する。主制御回路 9 0 は、A R T 抽籤や、A R T 状態のセット数の上乗せ抽籤に当籤した場合、当籤契機となった抽籤用フラグを用いて、当籤した A R T 状態のランクを決定する。A R T 当籤時ランク抽籤テーブルを用いて決定されたランクは、ランク決め A R T 中の昇格抽籤に基づき昇格し、通常 A R T 中のランク（抽籤状態）となる（図 2 1 2 (B) 参照）。

40

【1 5 8 9】

〔ストック放出順序抽籤テーブル〕

続いて、図 2 4 1 は、上述の放出フェーズ D において放出する A R T 状態のストックの順序を抽籤するために用いるストック放出順序抽籤テーブルである。ストック放出順序抽籤テーブルは、放出する A R T 状態のランクの順序（抽籤結果）についての抽籤値の情報を規定する。主制御回路 9 0 は、放出フェーズ D では、ストック放出順序抽籤テーブルを参照して決定した順序に応じて、保有する A R T 状態のストックの中から 1 つのストックを放出する。例えば、放出順序「4, 3, 2, 1」である場合、主制御回路 9 0 は、放出フ

50

フェーズDになると、保有するストックの中のランク4のストックを最も優先して放出し、ランク3のストックを次に優先して放出し、ランク2のストックを次に優先して放出し、ランク1のストックを次に優先して放出する。

【1590】

なお、放出順序の抽籤は、CZとART状態とのループ中に一度のみ行うこととしてもよく、また、放出フェーズDになるたびに行うこととしてもよい。また、放出順序の抽籤を、CZとART状態とのループ中に一度のみ行う場合の抽籤タイミングは任意であり、例えば、ループ中に初めて放出フェーズDになったタイミングでもよく、また、CZから初めてART状態に移行したタイミングであってもよい。

【1591】

〔ランク決めART中ランク昇格抽籤テーブル〕

続いて、図242は、ランク決めART中の昇格抽籤に用いるランク決めART中ランク昇格抽籤テーブルであり、図242(A)は、非ナビ高確時のランク決めART中に参照されるランク決めART中ランク昇格抽籤テーブルであり、図242(B)は、ナビ高確時のランク決めART中に参照されるランク決めART中ランク昇格抽籤テーブルであり、図242(C)は、1ゲームで完結するランク決めART中にBBに当籤した場合に参照されるランク決めART中ランク昇格抽籤テーブルである。

【1592】

ランク決めART中ランク昇格抽籤テーブルは、ランク決めART中の状況毎に設けられ、抽籤用フラグ毎に昇格抽籤の抽籤結果についての抽籤値の情報を規定する。主制御回路90は、ART当籤時に決定していたランクを、昇格抽籤の抽籤結果に基づいて更新する。なお、昇格抽籤の結果ランク4以上になる場合には、主制御回路90は、更新後のランクをランク4とする。

【1593】

〔ナビ高確ゲーム数獲得抽籤テーブル〕

続いて、図243は、通常ART中のナビ高確ゲーム数の獲得抽籤に用いるナビ高確ゲーム数獲得抽籤テーブルであり、図243(A)は、ART状態中の抽籤状態が「低確（ランク1）」時の通常ART中に参照されるナビ高確ゲーム数獲得抽籤テーブルであり、図243(B)は、ART状態中の抽籤状態が「高確（ランク2）」時の通常ART中に参照されるナビ高確ゲーム数獲得抽籤テーブルであり、図243(C)は、ART状態中の抽籤状態が「超高確（ランク3、4）」時の通常ART中に参照されるナビ高確ゲーム数獲得抽籤テーブルである。

【1594】

ナビ高確ゲーム数獲得抽籤テーブルは、ART状態中の抽籤状態毎に設けられ、抽籤用フラグ毎にナビ高確ゲーム数の獲得抽籤の抽籤結果についての抽籤値の情報を規定する。主制御回路90は、通常ART中に抽籤状態及び抽籤用フラグに基づいてナビ高確ゲーム数を付与する。その結果、ナビ高確ゲーム数が1以上になるとナビ高確中となる。ナビ高確ゲーム数は、通常ART中の1回の遊技毎に更新（1減算）されることになるが、この点は後述する。

【1595】

〔CZゲーム数獲得抽籤テーブル〕

続いて、図244は、CZゲーム数の上乗せ抽籤に用いるCZゲーム数獲得抽籤テーブルであり、図244(A)は、ART準備中に参照されるCZゲーム数獲得抽籤テーブルであり、図244(B)は、通常ART、かつ、非ナビ高確時に参照されるCZゲーム数獲得抽籤テーブルであり、図244(C)は、通常ART、かつ、ナビ高確時に参照されるCZゲーム数獲得抽籤テーブルである。また、図244(D)は、CZ中の抽籤フェーズA～CにおいてART抽籤に当籤した場合、又は、CZ中の放出フェーズDにおいてART状態のストックの放出が行われた場合に参照されるCZゲーム数獲得抽籤テーブルである。

【1596】

10

20

30

40

50

図 2 4 4 (A) ~ (C) に示す C Z ゲーム数獲得抽籤テーブルは、遊技状態や抽籤用フラグ毎に上乗せする C Z ゲーム数の抽籤結果についての抽籤値の情報を規定する。また、図 2 4 4 (D) に示す C Z ゲーム数獲得抽籤テーブルは、(抽籤用フラグに関わらず) 上乗せする C Z ゲーム数の抽籤結果についての抽籤値の情報を規定する。主制御回路 9 0 は、通常 A R T 中に、抽籤用フラグに基づいて C Z ゲーム数の上乗せを行うとともに、C Z から A R T 状態への移行時に C Z ゲーム数の上乗せを行う。なお、図 2 4 4 (D) に示すように、C Z から A R T 状態への移行時には、最低でも 5 ゲームの上乗せが行われる。

【1 5 9 7】

〔A R T ゲーム数獲得抽籤テーブル〕

続いて、図 2 4 5 は、A R T ゲーム数の上乗せ抽籤に用いる A R T ゲーム数獲得抽籤テーブルである。A R T ゲーム数獲得抽籤テーブルは、A R T 状態(ランク決め A R T、通常 A R T、E P 準備又は E P)、A R T 準備中又は継続 C Z 中に抽籤用フラグとして「N B B」又は「C B B」が決定された場合に参照され、抽籤用フラグ毎に上乗せする A R T ゲーム数の抽籤結果についての抽籤値の情報を規定する。なお、図 2 4 4 に示す C Z ゲーム数獲得抽籤テーブルと図 2 4 5 に示す A R T ゲーム数獲得抽籤テーブルとを比較すると分かるように、本実施形態のパチスロ 1 では、通常 A R T 中等は、B B 以外の抽籤用フラグに基づいて C Z ゲーム数の上乗せを行い、B B の抽籤用フラグに基づいて A R T ゲーム数の上乗せを行う。

【1 5 9 8】

主制御回路 9 0 は、A R T 状態、A R T 準備中又は継続 C Z 中に B B が当籤すると、A R T ゲーム数の上乗せを行う。その結果、B B 終了後に移行することになる通常 A R T の遊技期間が延長されることになる。

【1 5 9 9】

〔転落モード抽籤テーブル〕

続いて、図 2 4 6 は、E P 当籤時に転落モードを抽籤するための転落モード抽籤テーブルである。転落モード抽籤テーブルは、E P の当籤タイミング毎に設けられ、転落モードの抽籤結果についての抽籤値の情報を規定する。なお、E P は、ランク決め A R T においてランク 4 が決定されたタイミング、ナビ高確中の通常 A R T 中に「3 択昇格リブ」が内部当籤役として決定されたタイミング、A R T 中 B B 中の E P ストック抽籤に当籤したタイミングにおいてストックが付与され、ランク決め A R T においてランク 4 が決定された場合は「ランク決め A R T 中」欄の抽籤値を用いて転落モードを抽籤し、それ以外の場合は「ランク決め A R T 中以外」欄の抽籤値を用いて転落モードを抽籤する。

【1 6 0 0】

主制御回路 9 0 は、E P のストックが付与されると、そのタイミングに応じて転落モードを抽籤し、セットする。なお、セットした転落モードは、後述の図 2 4 7 の転落モード移行抽籤テーブルに示すように、E P 中に移行することになる。

【1 6 0 1】

〔転落モード移行抽籤テーブル〕

続いて、図 2 4 7 は、E P 中に転落モードを移行させるために用いる転落モード移行抽籤テーブルであり、図 2 4 7 (A) は、移行前の転落モードが「保証なし」である場合に参照される転落モード移行抽籤テーブルであり、図 2 4 7 (B) は、移行前の転落モードが「ショート」である場合に参照される転落モード移行抽籤テーブルであり、図 2 4 7 (C) は、移行前の転落モードが「ロング」である場合に参照される転落モード移行抽籤テーブルである。

【1 6 0 2】

転落モード移行抽籤テーブルは、現在の転落モード毎に設けられ、抽籤用フラグ毎に移行先の転落モードについての抽籤値の情報を規定する。主制御回路 9 0 は、E P 中は現在の転落モード及び抽籤用フラグに基づいて、転落モードを移行させる。なお、図 2 4 7 (A) に示す「保証なし」時の転落モード移行抽籤テーブルでは、「E P 継続時」という欄がある。この「E P 継続時」は、「6 択転落リブ」当籤時に自力で押し順に正解したときに

10

20

30

40

50

参照される。

【1603】

また、転落モード「ショート」又は「ロング」時に抽籤用フラグが「フェイク7」である場合、所定の確率で転落モード「保証なし」に移行することがある。しかしながら、上述したように、EP中は、少なくとも抽籤用フラグが1回「7揃い」又は「BAR揃い」になるまでは継続するため、EP中に抽籤用フラグ「7揃い」又は「BAR揃い」が一度も当籤していない場合には、抽籤用フラグ「フェイク7」が当籤しても転落モードの移行を行わない。

【1604】

〔EP中BB当籤時転落モード移行抽籤テーブル〕

続いて、図248は、EP中のBB当籤時に転落モードの移行を行うために用いるEP中BB当籤時転落モード移行抽籤テーブルである。EP中BB当籤時転落モード移行抽籤テーブルは、EP中に当籤したBBの種別、及び、移行前の転落モード毎に、移行先の転落モードについての抽籤値の情報を規定する。主制御回路90は、EP中にBBが当籤すると、当籤したBBの種別、及び、現在の転落モード及び抽籤用フラグに基づいて移行先の転落モードを決定する。EP中にBBに当籤すると、BBの終了後にEPに復帰するが、決定した転落モードは、BBの終了後に復帰したEPにおいて用いられる。

【1605】

〔BB中の各種抽籤テーブル〕

続いて、図249は、BB中の各種抽籤に用いる抽籤テーブルであり、図249(A)は、通常BB又はART中BB中に抽籤用フラグに基づいてART抽籤を行うためのBB中ART抽籤テーブルであり、図249(B)は、通常BB中に抽籤用フラグに基づいてCZ抽籤を行うためのBB中CZ抽籤テーブルであり、図249(C)は、ART中BB中に抽籤用フラグに基づいてEPストック抽籤を行うためのBB中EPストック抽籤テーブルである。

【1606】

図249に示すように、これら各種の抽籤テーブルは、抽籤用フラグ毎に抽籤結果についての抽籤値の情報を規定する。主制御回路90は、BB中のART抽籤に当籤した場合、ART状態のセット数のストックを所定数付与し、BB終了後に遊技状態をART準備中に移行する。また、主制御回路90は、BB中のCZ抽籤に当籤した場合、CZのストックを1つ付与し、BB終了後に遊技状態をCZ前兆又はCZに移行する。また、主制御回路90は、BB中のEPストック抽籤に当籤した場合、EPのストックを1つ付与し、BB終了後に遊技状態を（ART準備及びEP準備中を介して）EPに移行する。

【1607】

＜遊技状態毎の抽籤内容＞

次に、図250～図263を参照して各遊技状態において行われる各種抽籤の内容について説明する。なお、以下に説明する抽籤は、主制御回路90のメインCPU101により実行される後述のスタート時出玉関連抽籤処理及び停止時出玉関連抽籤処理（後述の図286参照）において行われる。また、本実施形態では、主制御回路90は、内部当籤役（抽籤用フラグ）に応じて、リール回転開始時（スタート時）又はリール停止時に各種抽籤を行うが、各種抽籤を行うタイミングは既に説明しているため（図208参照）、以下では、抽籤を行うタイミングについての詳細は省略する。

【1608】

〔通常状態中に行う抽籤内容〕

初めに、図250を参照して、主制御回路90のメインCPU101が通常状態中に行う各種の抽籤の詳細について説明する。

【1609】

通常状態中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図208参照）、メインCPU101は、(A)のART抽籤を行う。この抽籤では、メインCPU101は、BB当籤時には図225に示すBB当籤時ART抽籤テーブルを参照して、BB

10

20

30

40

50

と重複して当籤した内部当籤役に対応する抽籤用フラグ、及び現在の抽籤状態に基づきART抽籤を行う。また、BB非当籤時には、メインCPU101は、通常状態中の抽籤状態が「低確」又は「高確」である場合には、図226(A)に示す通常時ART抽籤テーブルを参照して、抽籤用フラグに基づきART抽籤を行い、また、通常状態中の抽籤状態が「超高確」である場合には、図226(B)に示す通常時ART抽籤テーブルを参照して、抽籤用フラグに基づきART抽籤を行う。

【1610】

ここで、通常時ART抽籤テーブルを参照すると、抽籤用フラグが「7揃い」又は「BAR揃い」である場合、通常状態中のART抽籤に必ず当籤することが分かり、また、図180の内部抽籤テーブル及び図208の内部当籤役と抽籤用フラグとの対応関係を参照すると、RT5状態中は、通常状態中のART抽籤に必ず当籤する抽籤用フラグ「7揃い」又は「BAR揃い」に対応する「F_7リブA」「F_7リブB」又は「F_BARリブ」が内部当籤役として決定される確率が、他のRT状態に比べて非常に高いことが分かる。

10

【1611】

そのため、本実施形態のパチスロ1では、非ナビ区間である通常状態中にRT状態がRT5状態に移行すると、その後のRT5状態中の遊技においてART抽籤に当籤する期待を持つことができる。

【1612】

また、通常時ART抽籤テーブルを参照すると、抽籤用フラグが「強チャンスリブ」である場合、通常状態中のART抽籤に所定の確率で当籤することが分かり、また、図208を参照すると、通常状態（非ナビ区間）では、内部当籤役として「強チェリブ」が決定された場合の他、内部当籤役として「3択昇格リブ」が決定され、かつ、押し順が正解である場合に、抽籤用フラグが「強チャンスリブ」となることが分かる。そして、内部当籤役として「強チェリブ」が決定された場合、又は、「3択昇格リブ」当籤時の押し順に正解した場合は、略称「強チェリブ(RT5移行図柄)」に係る図柄組合せが表示され、RT状態がRT5状態に移行するが分かる(図205参照)。

20

【1613】

そのため、本実施形態のパチスロ1では、非ナビ区間である通常状態中にRT状態を自力でRT5状態まで移行できた場合には、RT5状態に移行した遊技でもART抽籤の当籤に期待を持てることになる。すなわち、本実施形態のパチスロ1では、通常状態中は、抽籤用フラグとして「7揃い」又は「BAR揃い」が決定される確率が高いRT5状態中だけでなく、RT5状態に移行したタイミング、言い換えると、抽籤用フラグ「7揃い」又は「BAR揃い」のチャンス期間であるRT5状態中だけでなく、そのチャンス期間の入り口でも、ART抽籤の当籤に期待を持てることになる。

30

【1614】

また、上述のように通常状態中にRT状態がRT3状態に移行した場合には、ART状態のセット数が付与される(図209(B-2)参照)。ここで、通常状態中にRT3状態へ移行した場合には、メインCPU101は、(A)のART抽籤に当籤したものとして続く抽籤を行う。

【1615】

図250に戻り、(A)のART抽籤が非当籤の場合、メインCPU101は、続いて、(B)のCZ抽籤を行う。この抽籤では、メインCPU101は、BB当籤時には図222に示すBB当籤時状態別CZ抽籤テーブルを参照して、BBと重複して当籤した内部当籤役に対応する抽籤用フラグ、及び現在の抽籤状態に基づきCZ抽籤を行う。一方で、BB非当籤時には、メインCPU101は、抽籤用フラグが「ベル」である場合には、図220に示すモード別CZ抽籤テーブルを参照して、現在のモードに基づきCZ抽籤を行う。また、BB非当籤時、かつ、抽籤用フラグが「ベル」以外の場合は、メインCPU101は、図223に示す状態別CZ抽籤テーブルを参照して、抽籤用フラグ及び現在の抽籤状態に基づきCZ抽籤を行う。

40

【1616】

50

なお、ＢＢ当籤時、かつ、抽籤用フラグが「ベル」である場合、本実施形態では、図２２０に示すモード別ＣＺ抽籤テーブルを参照したＣＺ抽籤のみを行うが、これに限られるものではなく、メインＣＰＵ１０１は、ＢＢ当籤時、かつ、抽籤用フラグが「ベル」である場合に、図２２０に示すモード別ＣＺ抽籤テーブルを参照したＣＺ抽籤と、図２２０に示すモード別ＣＺ抽籤テーブルを参照したＣＺ抽籤の双方を行うこととしてもよい。

【１６１７】

続いて、（Ｂ）のＣＺ抽籤が非当籤の場合、メインＣＰＵ１０１は、続いて、（Ｃ）の状態移行抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２１７（Ａ）～図２１８（Ｄ）に示す状態移行抽籤テーブルを参照して、現在（移行前）の抽籤状態及び抽籤用フラグに基づいて移行先の抽籤状態を決定し、次遊技の抽籤状態としてセットする。

10

【１６１８】

続いて、（Ｃ）の状態移行抽籤において移行先の抽籤状態として「高確Ａ」又は「高確Ｂ」を決定すると、メインＣＰＵ１０１は、続いて、（Ｄ）の高確保証ゲーム数抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２１９に示す高確保証ゲーム数抽籤テーブルを参照して、移行先として決定した抽籤状態に基づき高確保証ゲーム数を決定し、付与する。

【１６１９】

（Ａ）のＡＲＴ抽籤及び（Ｂ）のＣＺ抽籤の双方が非当籤の場合には、（Ｃ）の状態移行抽籤及び（Ｄ）の高確保証ゲーム数抽籤を行うと、今回の遊技において行うべき抽籤が終了する。なお、次遊技の遊技状態（報知の有無を考慮した遊技状態）は、それぞれの抽籤結果に応じて異なるが、詳しくは後述する。

20

【１６２０】

また、（Ｂ）のＣＺ抽籤に当籤した場合、メインＣＰＵ１０１は、続いて、（Ｅ）の前兆ゲーム数抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２２４に示すＣＺ前兆ゲーム数抽籤テーブルを参照して、ＢＢ当籤の有無に基づきＣＺの前兆ゲーム数を決定、かつ、セットし、今回の遊技において行うべき抽籤を終了する。

【１６２１】

また、（Ａ）のＡＲＴ抽籤に当籤した場合、メインＣＰＵ１０１は、続いて、（Ｆ）のＡＲＴストック数抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３４に示すＡＲＴ当籤時ストック数抽籤テーブルを参照して、ＡＲＴ抽籤の当籤契機に基づきＡＲＴ状態のセット数を決定し、付与する。なお、図２３４に示すＡＲＴ当籤時ストック数抽籤テーブルを参照すると、通常状態中のＡＲＴ抽籤に当籤した場合には、ＡＲＴ状態のセット数として２つ以上が決定されることが分かる。

30

【１６２２】

メインＣＰＵ１０１は、続いて、（Ｇ）のランク抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３６～図２４０に示すＡＲＴ当籤時ランク抽籤テーブルを参照して、当籤したＡＲＴ状態のランクを決定する。具体的には、メインＣＰＵ１０１は、１つ目のＡＲＴ状態のストックについては、図２３６（Ａ）～（Ｃ）に示すＡＲＴ当籤時ランク抽籤テーブルを参照して、ＡＲＴ当籤時の抽籤用フラグに基づいてＡＲＴ状態のランクを決定し、２つ目以降のＡＲＴ状態のストックについては、図２４０（Ｏ）に示すＡＲＴ当籤時ランク抽籤テーブルを参照して、ＡＲＴ状態のランクを決定する。

40

【１６２３】

メインＣＰＵ１０１は、続いて、（Ｈ）の前兆ゲーム数抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２２７に示すＡＲＴ前兆ゲーム数抽籤テーブルを参照して、ＡＲＴ状態の前兆ゲーム数を決定、かつ、セットし、今回の遊技において行うべき抽籤を終了する。

【１６２４】

〔通常状態から移行する遊技状態〕

メインＣＰＵ１０１は、通常状態中にこれらの抽籤を行うと、その抽籤結果に応じて次遊技の遊技状態（報知の有無を考慮した遊技状態）をセットする。なお、報知の有無を考慮

50

した遊技状態の移行は、図206及び図207において既に説明しているが、通常状態から移行可能な遊技状態の種別、及びその条件を図250にも示している。

【1625】

具体的には、通常状態中に（A）のART抽籤に当籤し、かつ、今回の遊技においてBBが作動した場合（すなわち、今回の遊技においてBBに当籤し、かつ、BB当籤時のART抽籤に当籤し、かつ、BBに応じた図柄組合せが有効ラインに沿って表示された場合）、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。なお、この場合、メインCPU101は、通常状態中に用いるモード及び抽籤状態をクリアする。

【1626】

また、通常状態中に（A）のART抽籤に非当籤し、かつ、今回の遊技においてBBが作動した場合（すなわち、今回の遊技においてBBに当籤し、かつ、BB当籤時のART抽籤に非当籤し、かつ、BBに応じた図柄組合せが有効ラインに沿って表示された場合）、メインCPU101は、次遊技の遊技状態として通常BBをセットする。

【1627】

また、通常状態中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態として通常フラグ間をセットする。また、通常状態中に（A）のART抽籤に当籤した場合、メインCPU101は、次遊技の遊技状態としてART前兆をセットする。なお、この場合、メインCPU101は、通常状態中に用いるモード及び抽籤状態をクリアする。また、通常状態中に（B）のCZ抽籤に当籤した場合、メインCPU101は、次遊技の遊技状態としてCZ前兆をセットする。なお、この場合、メインCPU101は、通常状態中に用いる抽籤状態をクリアする。

【1628】

〔CZ前兆中に行う抽籤内容〕

続いて、図251を参照して、主制御回路90のメインCPU101がCZ前兆中に行う各種の抽籤の詳細について説明する。

【1629】

CZ前兆中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図208参照）、メインCPU101は、（A）のART抽籤を行う。この抽籤では、メインCPU101は、BB当籤時には図225（C）に示すBB当籤時ART抽籤テーブルを参照して、BBと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきART抽籤を行う。また、BB非当籤時には、メインCPU101は、図226（B）に示す通常時ART抽籤テーブルを参照して、抽籤用フラグに基づきART抽籤を行う。

【1630】

（A）のART抽籤が非当籤の場合、メインCPU101は、BB当籤の有無に基づいて処理を異ならせ、BB（「F__クラウンBB」「F__赤BB」又は「F__青BB」）が内部当籤役として決定されていない場合には、今回の遊技において行うべき抽籤を終了する。一方で、BBが内部当籤役として決定されている場合には、メインCPU101は、続いて、（B）のCZの前兆ゲーム数の抽籤を行う。この抽籤では、メインCPU101は、図224に示すCZ前兆ゲーム数抽籤テーブルの「BB当籤時」欄を参照して、CZの前兆ゲーム数を決定し、現在セットされているCZの前兆ゲーム数を決定したCZの前兆ゲーム数に書き換え、今回の遊技において行うべき抽籤を終了する。

【1631】

また、（A）のART抽籤に当籤した場合、メインCPU101は、続いて、（C）のARTストック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、（D）のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定する。続いて、メインCPU101は、CZの前兆ゲーム数をARTの前兆ゲーム数に書き換えて、今回の遊技において行うべき抽籤を終了する。

【1632】

〔CZ前兆から移行する遊技状態〕

CZ前兆中に(A)のART抽籤に当籤し、かつ、今回の遊技においてBBが作動した場合(すなわち、今回の遊技においてBBに当籤し、かつ、BB当籤時のART抽籤に当籤し、かつ、BBに応じた図柄組合せが有効ラインに沿って表示された場合)、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。なお、この場合、メインCPU101は、モードをクリアするとともに、CZの前兆ゲーム数をARTの前兆ゲーム数に書き換える。

【1633】

また、CZ前兆中に(A)のART抽籤に非当籤し、かつ、今回の遊技においてBBが作動した場合(すなわち、今回の遊技においてBBに当籤し、かつ、BB当籤時のART抽籤に非当籤し、かつ、BBに応じた図柄組合せが有効ラインに沿って表示された場合)、メインCPU101は、次遊技の遊技状態として通常BBをセットする。

【1634】

また、CZ前兆中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態として通常フラグ間をセットする。また、CZ前兆中に、(A)のART抽籤に当籤し、かつ、前兆ゲーム数が今回の遊技で0に更新され、かつ、RT状態が高RT状態(RT4状態又はRT5状態)である場合、メインCPU101は、次遊技の遊技状態としてランク決めARTをセットする。なお、この場合、メインCPU101は、モードをクリアする。

【1635】

また、CZ前兆中に(A)のART抽籤に当籤し、かつ、前兆ゲーム数が今回の遊技で0に更新され、かつ、RT状態が低RT状態(RT0状態~RT3状態)である場合、メインCPU101は、次遊技の遊技状態としてART準備中をセットする。なお、この場合、メインCPU101は、モードをクリアする。

【1636】

また、CZ前兆中に(A)のART抽籤に当籤し、かつ、前兆ゲーム数が今回の遊技の更新後においても1以上である場合には、メインCPU101は、次遊技の遊技状態としてART前兆をセットする。なお、この場合、メインCPU101は、モードをクリアする。

【1637】

また、CZ前兆中に(A)のART抽籤に当籤することなく、CZの前兆ゲーム数が0に更新された場合、メインCPU101は、次遊技の遊技状態として初当りCZをセットする。

【1638】

〔ART前兆中に行う抽籤内容〕

続いて、図252を参照して、主制御回路90のメインCPU101がART前兆中に行う各種の抽籤の詳細について説明する。

【1639】

ART前兆中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると(図208参照)、メインCPU101は、(A)のART抽籤を行う。この抽籤では、メインCPU101は、BB当籤時には図225(C)に示すBB当籤時ART抽籤テーブルを参照して、BBと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきART抽籤を行う。また、BB非当籤時には、メインCPU101は、図226(B)に示す通常時ART抽籤テーブルを参照して、抽籤用フラグに基づきART抽籤を行う。

【1640】

(A)のART抽籤が非当籤の場合、メインCPU101は、今回の遊技において行うべき抽籤を終了する。一方で、(A)のART抽籤に当籤した場合、メインCPU101は、続いて、(B)のARTストック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、

10

20

30

40

50

(C) のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定し、今回の遊技において行うべき抽籤を終了する。

【1641】

〔ART前兆から移行する遊技状態〕

ART前兆中にBBが作動した場合（すなわち、今回の遊技においてBBに当籤し、かつ、BBに応じた図柄組合せが有効ラインに沿って表示された場合）、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。また、ART前兆中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態として通常フラグ間をセットする。

【1642】

また、ART前兆中にARTの前兆ゲーム数が0に更新され、かつ、RT状態が高RT状態（RT4状態又はRT5状態）である場合、メインCPU101は、次遊技の遊技状態としてランク決めARTをセットし、また、ART前兆中にARTの前兆ゲーム数が0に更新され、かつ、RT状態が低RT状態（RT0状態～RT3状態）である場合、メインCPU101は、次遊技の遊技状態としてART準備中をセットする。なお、これらの場合、メインCPU101は、CZゲーム数に10をセットする。

【1643】

通常、ART状態とCZとのループは、初当りCZから始まるため、ART状態に移行した時点でCZゲーム数が既にセットされているが、ART前兆からART状態に移行（すなわち、通常状態からART前兆を介してART状態に移行）した場合、CZを経由していないためCZゲーム数がセットされていない。そこで、ART前兆中にARTの前兆ゲーム数が0に更新された場合には、CZゲーム数として10をセットし、その後のART状態とCZとのループを可能にしている。

【1644】

〔初当りCZ、及び、初当りCZ終了の次遊技に行う抽籤内容〕

続いて、図253を参照して、主制御回路90のメインCPU101が初当りCZ、及び、初当りCZ終了の次遊技に行う各種の抽籤の詳細について説明する。

【1645】

初当りCZ、及び、初当りCZ終了の次遊技では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図208参照）、メインCPU101は、(A)のART抽籤を行う。この抽籤では、メインCPU101は、BB当籤時には、図233(D)に示すART中BB当籤時ストック抽籤テーブルを参照して、BBと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきART抽籤を行い、また、BB非当籤時には、図228(A)～(C)に示すCZ中ART抽籤テーブルを参照して、抽籤用フラグに基づきART抽籤を行う。

【1646】

(A)のART抽籤が非当籤の場合、メインCPU101は、今回の遊技において行うべき抽籤を終了する。一方で、(A)のART抽籤に当籤した場合、メインCPU101は、続いて、(B)のARTストック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、(C)のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定する。

【1647】

続いて、メインCPU101は、(D)のCZゲーム数の獲得抽籤を行う。この抽籤では、メインCPU101は、図244(D)に示すCZゲーム数獲得抽籤テーブルを参照して、上乘せするCZゲーム数を決定し、当該上乘せするゲーム数を現在のCZゲーム数に加算し、今回の遊技において行うべき抽籤を終了する。

10

20

30

40

50

【1648】

〔初当りCZ、及び、初当りCZ終了の次遊技から移行する遊技状態〕

初当りCZ、及び、初当りCZ終了の次遊技にBBが作動した場合、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。また、ART前兆中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態としてART中フラグ間をセットする。なお、これらの場合、メインCPU101は、モードをクリアする。

【1649】

また、初当りCZ、及び、初当りCZ終了の次遊技において、ART抽籤に当籤し、かつ、RT状態が高RT状態（RT4状態又はRT5状態）である場合、メインCPU101は、次遊技の遊技状態としてランク決めARTをセットし、また、ART抽籤に当籤し、かつ、RT状態が低RT状態（RT4状態又はRT5状態）である場合、メインCPU101は、次遊技の遊技状態としてART準備中をセットする。なお、これらの場合、メインCPU101は、モードをクリアする。

【1650】

また、初当りCZ終了の次遊技において、ART抽籤に非当籤し、かつ、CZのストックがある場合には、メインCPU101は、次遊技の遊技状態としてCZ前兆をセットする。なお、この場合、メインCPU101は、モード及びCZの前兆ゲーム数を抽籤し、セットする。

【1651】

また、初当りCZ終了の次遊技において、ART抽籤に非当籤し、かつ、CZのストックがない場合には、メインCPU101は、次遊技の遊技状態として通常状態をセットする。なお、この場合、メインCPU101は、モード、通常状態中の抽籤状態、及び高確保証ゲーム数を抽籤し、セットする。

【1652】

〔ART準備中に行う抽籤内容〕

続いて、図254を参照して、主制御回路90のメインCPU101がART準備中に行う各種の抽籤の詳細について説明する。

【1653】

ART準備中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図208参照）、メインCPU101は、（A）のART抽籤を行う。この抽籤では、メインCPU101は、BB当籤時には、図233（E）に示すART中BB当籤時ストック抽籤テーブルを参照して、BBと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきART抽籤を行い、また、BB非当籤時には、図232（J）に示すART中ストック抽籤テーブルを参照して、抽籤用フラグに基づきART抽籤を行う。

【1654】

（A）のART抽籤に当籤した場合、メインCPU101は、続いて、（B）のARTストック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、（C）のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定する。

【1655】

（A）のART抽籤が非当籤の場合、又は、（C）のランク抽籤に続いて、メインCPU101は、（D）のCZゲーム数の獲得抽籤を行う。この抽籤では、メインCPU101は、図244（A）に示すCZゲーム数獲得抽籤テーブルを参照して、抽籤用フラグに基づき上乗せするCZゲーム数を決定し、当該上乗せするゲーム数を現在のCZゲーム数に加算する。

【1656】

続いて、メインCPU101は、（E）のARTゲーム数の獲得抽籤を行う。この抽籤で

10

20

30

40

50

は、メインCPU101は、図245に示すARTゲーム数獲得抽籤テーブルを参照して、当籤したBBの種別に応じて上乘せするARTゲーム数を決定し、当該上乘せするゲーム数を現在のARTゲーム数に加算し、今回の遊技において行うべき抽籤を終了する。

【1657】

〔ART準備から移行する遊技状態〕

ART準備中にBBが作動した場合、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。また、ART前兆中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態としてART中フラグ間をセットする。

【1658】

また、ART準備中にRT状態がRT4状態に移行すると、メインCPU101は、次遊技の遊技状態として以下の遊技状態をセットする。具体的には、通常ART又はランク決めART中に当籤したBBの終了後のART準備中にRT状態がRT4状態に移行した場合、メインCPU101は、次遊技の遊技状態として通常ARTをセットする。また、EP中に当籤したBBの終了後のART準備中にRT状態がRT4状態に移行した場合、メインCPU101は、次遊技の遊技状態としてEPをセットする。また、EPのストックが有るART準備中にRT状態がRT4状態に移行した場合、メインCPU101は、次遊技の遊技状態としてEPをセットする（なお、この場合、メインCPU101は、初回EP処理も行う）。また、メインCPU101は、これ以外の状況でART準備中にRT状態がRT4状態に移行すると、次遊技の遊技状態としてランク決めARTをセットする。

【1659】

なお、以下において、初回EP処理とは、EP中の転落モードを抽籤し、セットすること、及び、ナビ高確ゲーム数をクリア（0にする）すること、のうちの未だ終了していない処理を行うことをいう。すなわち、メインCPU101は、いずれの処理も未だ終了していない場合には、初回EP処理において両処理を行い、いずれか一方の処理が終了している場合には、初回EP処理において終了していない処理を行い、いずれの処理も終了している場合には、初回EP処理においていずれの処理も行わない。

【1660】

〔ランク決めART中に行う抽籤内容〕

続いて、図255を参照して、主制御回路90のメインCPU101がランク決めART中に行う各種の抽籤の詳細について説明する。

【1661】

ランク決めART中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図208参照）、メインCPU101は、（A）のランク昇格抽籤を行う。この抽籤では、メインCPU101は、図242に示すランク決めART中ランク昇格抽籤テーブルを参照して、抽籤用フラグやナビ高確中か否か等に基づきART状態のランクの昇格抽籤を行い、その抽籤結果に基づきART当籤時に仮決めしておいたランクを更新する。

【1662】

このランク昇格抽籤の結果、ART状態のランクがランク4まで昇格した場合、メインCPU101は、続いて、（B）の転落モード抽籤を行う。この抽籤では、メインCPU101は、図246に示す転落モード抽籤テーブルを参照して、転落モードの初期値を決定し、セットする。

【1663】

続いて、メインCPU101は、（C）のART抽籤を行う。この抽籤では、メインCPU101は、BB当籤時には、図233（C）に示すART中BB当籤時ストック抽籤テーブルを参照して、BBと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきART抽籤を行い、また、BB非当籤時には、図232（H）（I）に示すART中ストック抽籤テーブルを参照して、抽籤用フラグ及びナビ高確中か否かに基づきART抽籤を行う。

【1664】

（C）のART抽籤に当籤した場合、メインCPU101は、続いて、（D）のARTス

10

20

30

40

50

トック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、(E)のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定する。

【1665】

(C)のART抽籤が非当籤の場合、又は、(E)のランク抽籤に続いて、メインCPU101は、(F)ARTゲーム数の獲得抽籤を行う。この抽籤では、メインCPU101は、図245に示すARTゲーム数獲得抽籤テーブルを参照して、当籤したBBの種別に応じて上乗せするARTゲーム数を決定し、当該上乗せするゲーム数を現在のARTゲーム数に加算する。

10

【1666】

続いて、メインCPU101は、(G)の特殊CZの発動抽籤を行う。この抽籤では、メインCPU101は、図235に示す特殊CZ移行抽籤テーブルを参照して、通常ARTの終了後に移行する継続CZとして特殊CZを用いるか否かを抽籤し、今回の遊技において行うべき抽籤を終了する。

【1667】

[ランク決めARTから移行する遊技状態]

ランク決めART中にBBが作動した場合、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。また、ランク決めART中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態としてART中フラグ間をセットする。

20

【1668】

また、ランク決めART中の(A)のランク昇格抽籤でランク4まで昇格し、かつ、RT状態がRT5状態である場合、メインCPU101は、次遊技の遊技状態としてEPをセットする。また、ランク決めART中の(A)のランク昇格抽籤でランク4まで昇格し、かつ、RT5状態以外のRT状態である場合、メインCPU101は、次遊技の遊技状態としてEP準備中をセットする。なお、これらの場合、メインCPU101は、初回EP処理を行う。

【1669】

また、上述した状況以外の場合、メインCPU101は、次遊技の遊技状態として通常ARTをセットする。

30

【1670】

[通常ART中に行う抽籤内容]

続いて、図256を参照して、主制御回路90のメインCPU101が通常ART中に行う各種の抽籤の詳細について説明する。

【1671】

通常ART中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると(図208参照)、メインCPU101は、(A)のART抽籤を行う。この抽籤では、メインCPU101は、BB当籤時には、図233(A)(B)に示すART中BB当籤時ストック抽籤テーブルを参照して、BBと重複して当籤した内部当籤役に対応する抽籤用フラグ及びART状態中の抽籤状態に基づきART抽籤を行い、また、BB非当籤時には、図230に示すART中ストック抽籤テーブルを参照して、抽籤用フラグ、ART状態中の抽籤状態及びナビ高確中か否かに基づきART抽籤を行う。

40

【1672】

(A)のART抽籤に当籤した場合、メインCPU101は、続いて、(B)のARTストック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、(C)のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽

50

籤テーブルを参照して、当籤したART状態のランクを決定する。

【1673】

(A)のART抽籤が非当籤の場合、又は、(C)のランク抽籤に続いて、メインCPU101は、(D)CZゲーム数の獲得抽籤を行う。この抽籤では、メインCPU101は、図244(B)(C)に示すCZゲーム数獲得抽籤テーブルを参照して、抽籤用フラグ及びナビ高確中か否かに基づき上乘せするCZゲーム数を決定し、当該上乘せするゲーム数を現在のCZゲーム数に加算する。

【1674】

続いて、メインCPU101は、(E)のARTゲーム数の獲得抽籤を行う。この抽籤では、メインCPU101は、図245に示すARTゲーム数獲得抽籤テーブルを参照して、当籤したBBの種別に応じて上乘せするARTゲーム数を決定する。メインCPU101は、ARTゲーム数の獲得抽籤に当籤した場合(通常ART中のBB当籤時には、必ずARTゲーム数の獲得抽籤に当籤するため、通常ART中のBB当籤時と同じである)、当該上乘せするゲーム数を現在のARTゲーム数に加算し、今回の遊技において行うべき抽籤を終了する。

【1675】

一方で、(E)のARTゲーム数の獲得抽籤に非当籤した場合(BB当籤時以外は必ず非当籤であるため、通常ART中のBB非当籤時と同じである)、メインCPU101は、続いて、(F)のナビ高確ゲーム数の獲得抽籤を行う。この抽籤では、メインCPU101は、図243に示すナビ高確ゲーム数獲得抽籤テーブルを参照して、抽籤用フラグ及び(ART状態のランクから一義的に定まる)ART状態中の抽籤状態に基づき、上乘せするナビ高確ゲーム数を決定し、当該上乘せするゲーム数を現在のナビ高確ゲーム数に加算する。

【1676】

続いて、今回の遊技がナビ高確中であり、かつ、今回の遊技において「3択昇格リブ」が内部当籤役として決定された場合には、メインCPU101は、続いて、(G)の転落モード抽籤を行う。この抽籤では、メインCPU101は、図246に示す転落モード抽籤テーブルを参照して、転落モードの初期値を決定し、セットし、今回の遊技において行うべき抽籤を終了する。一方で、上述の何れかの条件を満たさない場合には、メインCPU101は、今回の遊技において行うべき抽籤を終了する。

【1677】

[通常ARTから移行する遊技状態]

通常ART中にBBが作動した場合、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。また、通常ART中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態としてART中フラグ間をセットする。

【1678】

また、通常ART中のナビ高確中にRT状態がRT5状態に移行すると、メインCPU101は、次遊技の遊技状態としてEPをセットする。なお、この場合、メインCPU101は、初回EP処理を行う。

【1679】

また、通常ART中にART状態の残りゲーム数が0に更新され、かつ、特殊CZに非当籤の場合、メインCPU101は、次遊技の遊技状態としてCZ(ART後)をセットする。また、通常ART中にART状態の残りゲーム数が0に更新され、かつ、特殊CZに当籤している場合、メインCPU101は、次遊技の遊技状態として特殊CZをセットする。

【1680】

[EP準備中に行う抽籤内容]

続いて、図257を参照して、主制御回路90のメインCPU101がEP準備中に行う各種の抽籤の詳細について説明する。

【1681】

ＥＰ準備中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図２０８参照）、メインＣＰＵ１０１は、（Ａ）のＡＲＴ抽籤を行う。この抽籤では、メインＣＰＵ１０１は、ＢＢ当籤時には、図２３３（Ｃ）に示すＡＲＴ中ＢＢ当籤時ストック抽籤テーブルを参照して、ＢＢと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきＡＲＴ抽籤を行い、また、ＢＢ非当籤時には、図２３２（Ｈ）（Ｉ）に示すＡＲＴ中ストック抽籤テーブルを参照して、抽籤用フラグ及びナビ高確中か否かに基づきＡＲＴ抽籤を行う。

【１６８２】

（Ａ）のＡＲＴ抽籤に当籤した場合、メインＣＰＵ１０１は、続いて、（Ｂ）のＡＲＴストック数抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３４に示すＡＲＴ当籤時ストック数抽籤テーブルを参照して、ＡＲＴ抽籤の当籤契機に基づきＡＲＴ状態のセット数を決定し、付与する。続いて、メインＣＰＵ１０１は、（Ｃ）のランク抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３６～図２４０に示すＡＲＴ当籤時ランク抽籤テーブルを参照して、当籤したＡＲＴ状態のランクを決定する。

【１６８３】

（Ａ）のＡＲＴ抽籤が非当籤の場合、又は、（Ｃ）のランク抽籤に続いて、メインＣＰＵ１０１は、（Ｄ）のＡＲＴゲーム数の獲得抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２４５に示すＡＲＴゲーム数獲得抽籤テーブルを参照して、当籤したＢＢの種別に応じて上乘せするＡＲＴゲーム数を決定する。この抽籤に当籤した場合には、メインＣＰＵ１０１は、上乘せするゲーム数を現在のＡＲＴゲーム数に加算し、今回の遊技において行うべき抽籤を終了する一方で、この抽籤に非当籤した場合には、メインＣＰＵ１０１は、今回の遊技において行うべき抽籤を終了する。

【１６８４】

〔ＥＰ準備から移行する遊技状態〕

ＥＰ準備中にＢＢが作動した場合、メインＣＰＵ１０１は、次遊技の遊技状態としてＡＲＴ中ＢＢをセットする。また、ＥＰ準備中にＢＢに当籤したもののＢＢが作動しない場合、メインＣＰＵ１０１は、次遊技の遊技状態としてＡＲＴ中フラグ間をセットする。

【１６８５】

また、ＥＰ準備中にＲＴ状態がＲＴ５状態に移行すると、メインＣＰＵ１０１は、次遊技の遊技状態としてＥＰをセットする。なお、この場合、メインＣＰＵ１０１は、初回ＥＰ処理を行う。

【１６８６】

〔ＥＰ中に行う抽籤内容〕

続いて、図２５８を参照して、主制御回路９０のメインＣＰＵ１０１がＥＰ中に行う各種の抽籤の詳細について説明する。

【１６８７】

ＥＰ中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図２０８参照）、メインＣＰＵ１０１は、（Ａ）のＡＲＴ抽籤を行う。この抽籤では、メインＣＰＵ１０１は、ＢＢ当籤時には、図２３３（Ａ）（Ｂ）（Ｄ）に示すＡＲＴ中ＢＢ当籤時ストック抽籤テーブルを参照して、ＢＢと重複して当籤した内部当籤役に対応する抽籤用フラグ、及びＥＰ中の転落モードに基づきＡＲＴ抽籤を行い、また、ＢＢ非当籤時には、図２３１に示すＡＲＴ中ストック抽籤テーブルを参照して、抽籤用フラグ及びＥＰ中の転落モードに基づきＡＲＴ抽籤を行う。

【１６８８】

（Ａ）のＡＲＴ抽籤に当籤した場合、メインＣＰＵ１０１は、続いて、（Ｂ）のＡＲＴストック数抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３４に示すＡＲＴ当籤時ストック数抽籤テーブルを参照して、ＡＲＴ抽籤の当籤契機に基づきＡＲＴ状態のセット数を決定し、付与する。続いて、メインＣＰＵ１０１は、（Ｃ）のランク抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３６～図２４０に示すＡＲＴ当籤時ランク抽籤テーブルを参照して、当籤したＡＲＴ状態のランクを決定する。

【１６８９】

(A) のART抽籤が非当籤の場合、又は、(C) のランク抽籤に続いて、メインCPU101は、(D) のARTゲーム数の獲得抽籤を行う。この抽籤では、メインCPU101は、図245に示すARTゲーム数獲得抽籤テーブルを参照して、当籤したBBの種別に応じて上乗せするARTゲーム数を決定する。

【1690】

(C) のARTゲーム数の獲得抽籤に非当籤した場合、メインCPU101は、(E) の転落モード移行抽籤を行う。この抽籤では、メインCPU101は、図247に示す転落モード移行抽籤テーブルを参照して、抽籤用フラグ及び現在の転落モードに基づき移行先の転落モードを決定し、セットする。続いて、メインCPU101は、今回の遊技において行うべき抽籤を終了する。

10

【1691】

一方で、(C) のARTゲーム数の獲得抽籤に当籤した場合、メインCPU101は、上乗せするゲーム数を現在のARTゲーム数に加算し、続いて、(F) の転落モード移行抽籤を行う。この抽籤では、メインCPU101は、図248に示すEP中BB当籤時転落モード移行抽籤テーブルを参照して、EP中に当籤したBBの種別、及び、移行前の転落モードに基づき、移行先の転落モードを決定し、セットする。続いて、メインCPU101は、今回の遊技において行うべき抽籤を終了する。

【1692】

〔EPから移行する遊技状態〕

EP中にBBが作動した場合、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。また、EP中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態としてART中フラグ間をセットする。

20

【1693】

また、EP中にRT5状態以外のRT状態に移行し、かつ、EPのストックがある場合、メインCPU101は、次遊技の遊技状態としてEP準備中をセットする。また、EP中にRT5状態以外のRT状態に移行し、かつ、EPのストックがない場合、メインCPU101は、次遊技の遊技状態として通常ARTをセットする。

【1694】

〔CZ (ART後)、CZ (ART後) 終了の次遊技、及び特殊CZ中に行う抽籤内容〕
続いて、図259を参照して、主制御回路90のメインCPU101がCZ (ART後)、CZ (ART後) 終了の次遊技、及び特殊CZ中に行う各種の抽籤の詳細について説明する。

30

【1695】

CZ (ART後)、CZ (ART後) 終了の次遊技、及び特殊CZ中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると(図208参照)、メインCPU101は、(A) のART抽籤を行う(上述の抽籤フェーズA～C)。この抽籤では、メインCPU101は、BB当籤時には、図233(D) に示すART中BB当籤時ストック抽籤テーブルを参照して、BBと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきART抽籤を行い、また、BB非当籤時には、図228(D)～図229(H) に示すCZ中ART抽籤テーブルを参照して、抽籤用フラグ、CZの種別(CZ (ART後)、CZ (ART後) 終了の次遊技、又は特殊CZ)、及びナビ高確中か否かに基づきART抽籤を行う。

40

【1696】

(A) のART抽籤が非当籤の場合、メインCPU101は、ART状態のストックの放出があるか否かに応じてその後の処理を異ならせる。上述のように、ART状態のストックの放出は、継続CZ中のCZの残りゲーム数が0になったタイミングで行われるため(上述の放出フェーズD)、今回の遊技がCZ (ART後) 終了の次遊技、又はCZの残りゲーム数が1以上のCZ (ART後) 若しくは特殊CZ中は、ART状態のストックの放出は行わず、今回の遊技において行うべき抽籤を終了する。また、ART状態のストックがない場合にはストックを放出することができないため、CZの残りゲーム数が0のCZ

50

(ART後)中であっても、ART状態のストックがない場合には、ART状態のストックの放出は行わず、今回の遊技において行うべき抽籤を終了する(なお、特殊CZは、放出フェーズDのストックが放出されると終了するため、特殊CZ中は常にART状態のセット数が1つ以上ストックされている)。

【1697】

一方で、CZの残りゲーム数が0のCZ(ART後)又は特殊CZにおいてART状態のストックがある場合、メインCPU101は、続いて、(B)の放出順序抽籤を行う。この抽籤では、メインCPU101は、図241に示すストック放出順序抽籤テーブルを参照して、ストックしているART状態に対して設定されたランクに基づいて、ストックしているART状態の中から、放出フェーズDで放出するストック分のART状態を決定する。この抽籤の後、メインCPU101は、決定したART状態を、ART状態のストックの中から消し去るとともに、決定したART状態に対して設定されたランクに基づいて、次遊技のランク決めART中に昇格抽籤を行う。

10

【1698】

一方で、(A)のART抽籤に当籤した場合、メインCPU101は、続いて、(C)のARTストック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、(D)のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定する。

20

【1699】

(B)の放出順序抽籤、又は(D)のランク抽籤に続いて、メインCPU101は、(E)のCZゲーム数の獲得抽籤を行う。この抽籤では、メインCPU101は、図244(D)に示すCZゲーム数獲得抽籤テーブルを参照して、上乘せするCZゲーム数を決定し、当該上乘せするゲーム数を現在のCZゲーム数に加算し、今回の遊技において行うべき抽籤を終了する。

【1700】

[CZ(ART後)、CZ(ART後)終了の次遊技、及び特殊CZから移行する遊技状態]

CZ(ART後)、CZ(ART後)終了の次遊技、及び特殊CZ中にBBが作動した場合、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。また、ART前兆中にBBに当籤したもののBBが作動しない場合、メインCPU101は、次遊技の遊技状態としてART中フラグ間をセットする。

30

【1701】

また、CZ(ART後)、CZ(ART後)終了の次遊技、及び特殊CZ中に、ART抽籤に当籤、又はART状態のストックの放出が行われると、メインCPU101は、次遊技の遊技状態としてランク決めARTをセットする。

【1702】

また、CZ(ART後)の次遊技において、ART抽籤に非当籤し、かつ、CZのストックがある場合には、メインCPU101は、次遊技の遊技状態としてCZ前兆をセットする。なお、この場合、メインCPU101は、モード及びCZの前兆ゲーム数を抽籤し、セットする。

40

【1703】

また、CZ(ART後)の次遊技において、ART抽籤に非当籤し、かつ、CZのストックがない場合には、メインCPU101は、次遊技の遊技状態として通常状態をセットする。なお、この場合、メインCPU101は、モード、通常状態中の抽籤状態、及び高確保証ゲーム数を抽籤し、セットする。

【1704】

[通常フラグ間中に行う抽籤内容]

続いて、図260を参照して、主制御回路90のメインCPU101が通常フラグ間中に

50

行う各種の抽籤の詳細について説明する。

【1705】

通常フラグ間中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図208参照）、メインCPU101は、（A）のART抽籤を行う。この抽籤では、メインCPU101は、図226（C）に示す通常時ART抽籤テーブルを参照して、持ち越しているBBと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきART抽籤を行う。（A）のART抽籤に非当籤した場合、メインCPU101は、今回の遊技において行うべき抽籤を終了する。

【1706】

一方で、（A）のART抽籤に当籤した場合、メインCPU101は、続いて、（B）のARTストック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、（C）のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定し、今回の遊技において行うべき抽籤を終了する。

【1707】

〔通常フラグ間から移行する遊技状態〕

通常フラグ間中にBBが作動し、かつ、BBの作動時にART状態のセット数のストックがない場合、メインCPU101は、次遊技の遊技状態として通常BBをセットする。また、通常フラグ間中にBBが作動し、かつ、BBの作動時にART状態のセット数のストックがある場合、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。

【1708】

〔通常BB中に行う抽籤内容〕

続いて、図261を参照して、主制御回路90のメインCPU101が通常BB中に行う各種の抽籤の詳細について説明する。

【1709】

通常BB中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図208参照）、メインCPU101は、（A）のART抽籤を行う。この抽籤では、メインCPU101は、図249（A）に示すBB中ART抽籤テーブルを参照して、抽籤用フラグに基づきART抽籤を行う。

【1710】

（A）のART抽籤に非当籤した場合、メインCPU101は、続いて、（B）のCZ抽籤を行う。この抽籤では、メインCPU101は、図249（B）に示すBB中CZ抽籤テーブルを参照して、抽籤用フラグに基づきCZ抽籤を行い、今回の遊技において行うべき抽籤を終了する。なお、このCZ抽籤に当籤した場合には、メインCPU101は、CZの権利を1つ付与（ストック）する。

【1711】

一方で、（A）のART抽籤に当籤した場合、メインCPU101は、続いて、（C）のARTストック数抽籤を行う。この抽籤では、メインCPU101は、図234に示すART当籤時ストック数抽籤テーブルを参照して、ART抽籤の当籤契機に基づきART状態のセット数を決定し、付与する。続いて、メインCPU101は、（D）のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定し、今回の遊技において行うべき抽籤を終了する。

【1712】

〔通常BBから移行する遊技状態〕

通常BB中に（A）のART抽籤に当籤すると、メインCPU101は、次遊技の遊技状態としてART中BBをセットする。また、通常BB中に払い出されたメダルの枚数が規

10

20

30

40

50

定枚数に達し、ＢＢの作動が終了すると、メインＣＰＵ１０１は、各種のストックの有無に応じて次遊技の遊技状態をセットする。具体的には、ＡＲＴ状態のセット数のストックがあり、かつ、ＡＲＴ状態の前兆ゲーム数がない場合には、メインＣＰＵ１０１は、次遊技の遊技状態としてＡＲＴ準備をセットする。

【１７１３】

また、ＣＺのストックがあり、かつ、ＣＺの前兆ゲーム数がない場合には、メインＣＰＵ１０１は、次遊技の遊技状態として初当りＣＺをセットする。なお、この場合、メインＣＰＵ１０１は、モード、及び通常状態中の抽籤状態をセットする。また、ＣＺのストックがあり、かつ、ＣＺの前兆ゲーム数がある場合には、メインＣＰＵ１０１は、次遊技の遊技状態としてＣＺ前兆をセットする。なお、この場合、メインＣＰＵ１０１は、モード、及び通常状態中の抽籤状態をセットする。また、ＡＲＴ状態及びＣＺのいずれのストックもない場合には、メインＣＰＵ１０１は、次遊技の遊技状態として通常状態をセットする。なお、この場合、メインＣＰＵ１０１は、モード、及び通常状態中の抽籤状態をセットする。

10

【１７１４】

〔ＡＲＴ中フラグ間中に行う抽籤内容〕

続いて、図２６２を参照して、主制御回路９０のメインＣＰＵ１０１がＡＲＴ中フラグ間中に行う各種の抽籤の詳細について説明する。

【１７１５】

ＡＲＴ中フラグ間中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図２０８参照）、メインＣＰＵ１０１は、（Ａ）のＡＲＴ抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３２（Ｋ）に示すＡＲＴ中ストック抽籤テーブルを参照して、持ち越しているＢＢと重複して当籤した内部当籤役に対応する抽籤用フラグに基づきＡＲＴ抽籤を行う。（Ａ）のＡＲＴ抽籤に非当籤した場合、メインＣＰＵ１０１は、今回の遊技において行うべき抽籤を終了する。

20

【１７１６】

一方で、（Ａ）のＡＲＴ抽籤に当籤した場合、メインＣＰＵ１０１は、続いて、（Ｂ）のＡＲＴストック数抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３４に示すＡＲＴ当籤時ストック数抽籤テーブルを参照して、ＡＲＴ抽籤の当籤契機に基づきＡＲＴ状態のセット数を決定し、付与する。続いて、メインＣＰＵ１０１は、（Ｃ）のランク抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３６～図２４０に示すＡＲＴ当籤時ランク抽籤テーブルを参照して、当籤したＡＲＴ状態のランクを決定し、今回の遊技において行うべき抽籤を終了する。

30

【１７１７】

〔ＡＲＴ中フラグ間から移行する遊技状態〕

ＡＲＴ中フラグ間中にＢＢが作動すると、メインＣＰＵ１０１は、次遊技の遊技状態としてＡＲＴ中ＢＢをセットする。

【１７１８】

〔ＡＲＴ中ＢＢ中に行う抽籤内容〕

続いて、図２６３を参照して、主制御回路９０のメインＣＰＵ１０１がＡＲＴ中ＢＢ中に行う各種の抽籤の詳細について説明する。

40

【１７１９】

ＡＲＴ中ＢＢ中では、内部当籤役や押し順に基づいて抽籤用フラグが決まると（図２０８参照）、メインＣＰＵ１０１は、（Ａ）のＡＲＴ抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２４９（Ａ）に示すＢＢ中ＡＲＴ抽籤テーブルを参照して、抽籤用フラグに基づきＡＲＴ抽籤を行う。

【１７２０】

（Ａ）のＡＲＴ抽籤に当籤した場合、メインＣＰＵ１０１は、続いて、（Ｂ）のＡＲＴストック数抽籤を行う。この抽籤では、メインＣＰＵ１０１は、図２３４に示すＡＲＴ当籤時ストック数抽籤テーブルを参照して、ＡＲＴ抽籤の当籤契機に基づきＡＲＴ状態のセッ

50

ト数を決定し、付与する。続いて、メインCPU101は、(C)のランク抽籤を行う。この抽籤では、メインCPU101は、図236～図240に示すART当籤時ランク抽籤テーブルを参照して、当籤したART状態のランクを決定し、今回の遊技において行うべき抽籤を終了する。

【1721】

また、(A)のART抽籤に非当籤した場合、又は(C)のランク抽籤に続いて、メインCPU101は、(D)のEPストック抽籤を行う。この抽籤では、メインCPU101は、図249(C)に示すBB中EPストック抽籤テーブルを参照して、抽籤用フラグに基づきEPのストック抽籤を行う。(D)のEPストック抽籤に非当籤した場合には、メインCPU101は、今回の遊技において行うべき抽籤を終了する。

10

【1722】

一方、(D)のEPストック抽籤に当籤した場合には、メインCPU101は、続いて、(E)の転落モード抽籤を行う。この抽籤では、メインCPU101は、図246に示す転落モード抽籤テーブルを参照して、EP中の転落モードを抽籤し、セットし、今回の遊技において行うべき抽籤を終了する。

【1723】

[ART中BBから移行する遊技状態]

ART中BB中に払い出されたメダルの枚数が規定枚数に達し、BBの作動が終了すると、メインCPU101は、次遊技の遊技状態としてART準備をセットする。

【1724】

20

<パチスロ1に特有な制御>

以上、本実施形態のパチスロ1の遊技性に関して説明した。続いて、本実施形態のパチスロ1に特有の制御に関して個別に説明する。

【1725】

[ART抽籤及びランク抽籤]

本実施形態のパチスロ1では、図228(A)(B)のCZ中ART抽籤テーブルに示すように、CZ中のART抽籤は、CZの残りゲーム数に応じて当籤する確率が異なるものの、図239のART当籤時ランク抽籤テーブルに示すように、CZ中のランク抽籤は、決定されるランクの傾向は、CZの残りゲーム数に関わらず同一であるが、これに限られるものではない。すなわち、CZ(初当りCZ及び継続CZ)中に行うART抽籤及びランク抽籤の双方ともに、CZの残りゲーム数に応じて異ならせることとしてもよい。

30

【1726】

ここで、図264及び図265は、CZ中に行うART抽籤とランク抽籤との概念図であり、図264(A)(B)は、CZの残りゲーム数とCZ中に行う各抽籤との関係例を示している。また、図264及び図265に概念的に示すテーブルのうち、ART抽籤に用いるテーブルは、図228等 to 示すCZ中ART抽籤テーブルに相当するテーブルであり、また、ランク抽籤に用いるテーブルは、図236等 to 示すART当籤時ランク抽籤テーブルに相当するテーブルである。

【1727】

パチスロ1において主制御回路90は、CZの残りゲーム数に応じて異なる確率で当籤するART抽籤を行うこととしてもよい。具体的には、図264(A)に示すように、ART抽籤に用いる抽籤テーブルを、CZの残りゲーム数毎に有することで、CZの残りゲーム数に応じて異なる確率で当籤するART抽籤を行うことができる。

40

【1728】

なお、図264(A)に示す例では、CZの残りゲーム数が「7, 8ゲーム」である場合には、テーブルAを用いてART抽籤を行い、CZの残りゲーム数が「5, 6ゲーム」である場合には、テーブルBを用いてART抽籤を行い、CZの残りゲーム数が「3, 4ゲーム」である場合には、テーブルCを用いてART抽籤を行い、CZの残りゲーム数が「1, 2ゲーム」である場合には、テーブルDを用いてART抽籤を行い、CZの残りゲーム数が「0ゲーム」である場合には、テーブルEを用いてART抽籤を行うこととしてい

50

る。

【1729】

この場合において、C Zの残りゲーム数とART抽籤に当籤する確率との関係は、参照するテーブルに応じて任意に設定することができ、例えば、C Zの残りゲーム数が長いほどART抽籤に当籤する確率を高くしてもよく、また、反対にC Zの残りゲーム数が短いほどART抽籤に当籤する確率を高くしてもよい。このようにすることで、C Z中は、C Zの消化に伴い残りゲーム数が減ることで、ART抽籤に当籤する確率が徐々に低く又は高くなっていくことになる。なお、図264(A)に示す例の場合、テーブルA、B、C、D、Eの順で、ART抽籤に当籤する確率を高く又は低くすることで実現することができる。

10

【1730】

また、特定の残りゲーム数において、ART抽籤に当籤する確率を高く又は低くしてもよい。図264(A)に示す例の場合、例えば、テーブルBをART抽籤に当籤する確率が最も高く設定し、テーブルDをART抽籤に当籤する確率が次に高く設定し、テーブルEをART抽籤に当籤する確率が次に高く設定し、テーブルAをART抽籤に当籤する確率が次に高く設定し、テーブルBをART抽籤に当籤する確率が次に高く設定することで、ART抽籤に当籤する確率が高いゲーム数と低いゲーム数とを混在させることができる。

【1731】

また、パチスロ1において主制御回路90は、C Zの残りゲーム数に応じてランク抽籤の抽籤結果を異ならせることとしてもよい。具体的には、図264(A)に示すように、ランク抽籤に用いる抽籤テーブルを、C Zの残りゲーム数毎に有することで、C Zの残りゲーム数に応じてランク抽籤の抽籤結果を異ならせることができる。

20

【1732】

なお、図264(A)に示す例では、C Zの残りゲーム数が「7、8ゲーム」である場合には、テーブルVを用いてランク抽籤を行い、C Zの残りゲーム数が「5、6ゲーム」である場合には、テーブルWを用いてランク抽籤を行い、C Zの残りゲーム数が「3、4ゲーム」である場合には、テーブルXを用いてランク抽籤を行い、C Zの残りゲーム数が「1、2ゲーム」である場合には、テーブルYを用いてランク抽籤を行い、C Zの残りゲーム数が「0ゲーム」である場合には、テーブルZを用いてランク抽籤を行うこととしている。

30

【1733】

この場合において、C Zの残りゲーム数とランク抽籤の抽籤結果との関係は、参照するテーブルに応じて任意に設定することができ、例えば、C Zの残りゲーム数が長いほどART当籤時のART状態のランクとして高ランクが決定され易くしてもよく、また、反対にC Zの残りゲーム数が短いほどART当籤時のART状態のランクとして高ランクが決定され易くしてもよい。このようにすることで、C Z中は、C Zの消化に伴い残りゲーム数が減ることで、ART当籤時のART状態のランクが徐々に低く又は高くなっていくことになる。なお、図264(A)に示す例の場合、テーブルV、W、X、Y、Zの順で、ART状態のランクとして高ランクが決定され易く又は決定され難くすることで実現することができる。

40

【1734】

また、特定の残りゲーム数では、ART状態のランクとして高ランクが決定され易く又は決定され難くしてもよい。図264(A)に示す例の場合、例えば、テーブルX、W、Z、V、Yの順で、ART状態のランクとして高ランクが決定され易く又は決定され難くすることで、ART当籤時にART状態のランクとして高ランクが決定され易いゲーム数と決定され難いゲーム数とを混在させることができる。

【1735】

なお、図264(A)では、C Z中に行うART抽籤とランク抽籤との間で、C Zの残りゲーム数を揃えているが、これに限られるものではない。例えば、図264(B)に示すように、C Z中のART抽籤は、C Zの残りゲーム数「7、8ゲーム」「5、6ゲーム」

50

「3, 4ゲーム」「1, 2ゲーム」「0ゲーム」に応じて異ならせる一方で、C Z中のランク抽籤は、C Zの残りゲーム数「6, 7, 8ゲーム」「4, 5ゲーム」「0, 1, 2, 3ゲーム」に応じて異ならせている。このようにすることで、C Z中の各種抽籤をより多様化することができ、遊技性に富んだものにすることができる。

【1736】

また、パチスロ1において主制御回路90は、ART状態のセット数のストックの有無に応じて、C Z中のART抽籤及びランク抽籤を行うこととしてもよい。具体的には、図265(A)に示すように、ART抽籤に用いる抽籤テーブル及びランク抽籤に用いる抽籤テーブルを、ART状態のセット数のストックの有無毎に有することで、C Z中に主制御回路90は、ART状態のセット数のストックの有無に応じて抽籤結果が異なるART抽籤及びランク抽籤を行うことができる。

10

【1737】

なお、図265(A)に示す例では、ART状態のセット数のストックがある場合には、テーブルAを用いてART抽籤を行い、また、テーブルZを用いてランク抽籤を行い、ART状態のセット数のストックがない場合には、テーブルBを用いてART抽籤を行い、また、テーブルYを用いてランク抽籤を行うこととしている。

【1738】

この場合において、ART状態のセット数のストックの有無と、ART抽籤及びランク抽籤の抽籤結果との関係は、参照するテーブルに応じて任意に設定することができ、例えば、ART状態のセット数のストックがある場合にはストックがない場合に比べて、ART抽籤に当籤する確率が低く、ART状態のランクとして低いランクが決定され易くしてもよく、また反対に、ART抽籤に当籤する確率が高く、ART状態のランクとして高いランクが決定され易くしてもよい。また、例えば、ART状態のセット数のストックがある場合にはストックがない場合に比べて、ART抽籤に当籤する確率は低いものの、ART状態のランクとしては高いランクが決定され易くしてもよく、また反対に、ART抽籤に当籤する確率は高いものの、ART状態のランクとしては低いランクが決定され易くしてもよい。

20

【1739】

なお、ART状態のセット数のストックの有無に応じてART抽籤及びランク抽籤の抽籤結果を異ならせる制御は、C Z中に限るものではなく、通常状態、ART状態又はBB中のいずれの状態であっても適用することができる。

30

【1740】

また、パチスロ1において主制御回路90は、C Zの残りゲーム数と、ART状態のセット数のストックの有無との双方を考慮して、C Z中のART抽籤及びランク抽籤を行うこととしてもよい。具体的には、図265(B)に示すように、ART抽籤に用いる抽籤テーブル及びランク抽籤に用いる抽籤テーブルを、C Zの残りゲーム数及びART状態のセット数のストックの有無毎に有することで、C Z中に主制御回路90は、C Zの残りゲーム数及びART状態のセット数のストックの有無に応じて抽籤結果が異なるART抽籤及びランク抽籤を行うことができる。

【1741】

なお、図265(B)に示す例では、ART状態のセット数のストックがあるときは、C Zの残りゲーム数が「7, 8ゲーム」である場合に、テーブルAを用いてART抽籤を行うとともにテーブルVを用いてランク抽籤を行い、C Zの残りゲーム数が「5, 6ゲーム」である場合に、テーブルBを用いてART抽籤を行うとともにテーブルWを用いてランク抽籤を行い、C Zの残りゲーム数が「3, 4ゲーム」である場合には、テーブルCを用いてART抽籤を行うとともにテーブルXを用いてランク抽籤を行い、C Zの残りゲーム数が「1, 2ゲーム」である場合には、テーブルDを用いてART抽籤を行うとともにテーブルYを用いてランク抽籤を行い、C Zの残りゲーム数が「0ゲーム」である場合には、テーブルEを用いてART抽籤を行うとともにテーブルZを用いてランク抽籤を行うこととしている。

40

50

【1742】

一方で、ART状態のセット数のストックがないときは、CZの残りゲーム数が「7, 8ゲーム」である場合に、テーブルFを用いてART抽籤を行うとともにテーブルQを用いてランク抽籤を行い、CZの残りゲーム数が「5, 6ゲーム」である場合に、テーブルGを用いてART抽籤を行うとともにテーブルRを用いてランク抽籤を行い、CZの残りゲーム数が「3, 4ゲーム」である場合には、テーブルHを用いてART抽籤を行うとともにテーブルSを用いてランク抽籤を行い、CZの残りゲーム数が「1, 2ゲーム」である場合には、テーブルIを用いてART抽籤を行うとともにテーブルTを用いてランク抽籤を行い、CZの残りゲーム数が「0ゲーム」である場合には、テーブルJを用いてART抽籤を行うとともにテーブルUを用いてランク抽籤を行うこととしている。

10

【1743】

この場合においても、CZの残りゲーム数及びART状態のセット数のストックの有無と、ART抽籤に当籤する確率との関係、並びに、CZの残りゲーム数及びART状態のセット数のストックの有無と、ランク抽籤の抽籤結果との関係を参照するテーブルに応じて任意に設定できる点は、上述の通りである。なお、図264の例に比べて図265の例では、ART状態のセット数のストックの有無を更に考慮することとしているため、ART状態のセット数のストックがある場合とストックがない場合とで、ART抽籤及びランク抽籤の傾向も異ならせることができる。

【1744】

例えば、ART状態のセット数のストックがある場合は、CZの消化に伴い残りゲーム数が減るにつれて、ART抽籤に当籤する確率が低く（又は高く）なる一方で、ART状態のセット数のストックがない場合は、CZの消化に伴い残りゲーム数が減るにつれて、ART抽籤に当籤する確率が高く（又は低く）なるように設定することもできる。また、例えば、ART状態のセット数のストックがある場合とない場合とで、ART抽籤に当籤し易いゲーム数を異ならせることもできる。

20

【1745】

もちろん、ART状態のランク抽籤も同様であり、例えば、ART状態のセット数のストックがある場合は、CZの消化に伴い残りゲーム数が減るにつれて、ART状態のランクとして低い（又は高い）ランクが決定され易くなる一方で、ART状態のセット数のストックがない場合は、CZの消化に伴い残りゲーム数が減るにつれて、ART状態のランクとして高い（又は低い）ランクが決定され易くなるように設定することもできる。また、例えば、ART状態のセット数のストックがある場合とない場合とで、ART状態のランクとして高いランクが決定され易いゲーム数を異ならせることもできる。

30

【1746】

また、図265（A）（B）では、CZ中に行うART抽籤とランク抽籤との間で、CZの残りゲーム数を揃えているが、これに限られるものではなく、図264（B）に例示するように、CZ中に行うART抽籤とランク抽籤との間で、CZの残りゲーム数を異ならせることとしてもよい。

【1747】

同様に、図265（A）（B）では、ART状態のストックがある場合に行うART抽籤と、ART状態のストックがない場合に行うART抽籤との間でCZの残りゲーム数を揃え、同様に、ART状態のストックがある場合に行うランク抽籤と、ART状態のストックがない場合に行うランク抽籤との間でCZの残りゲーム数を揃えているが、これに限られるものではない。

40

【1748】

例えば、図265（C）に示すように、ART状態のストックがない場合には、CZ中のART抽籤は、CZの残りゲーム数「7, 8ゲーム」「5, 6ゲーム」「3, 4ゲーム」「1, 2ゲーム」「0ゲーム」に応じて異ならせる一方で、CZ中のランク抽籤は、CZの残りゲーム数「6, 7, 8ゲーム」「4, 5ゲーム」「0, 1, 2, 3ゲーム」に応じて異ならせ、また、ART状態のストックがある場合には、CZ中のART抽籤は、CZ

50

の残りゲーム数「5, 6, 7, 8ゲーム」「2, 3, 4ゲーム」「0, 1ゲーム」に応じて異ならせる一方で、C Z中のランク抽籤は、C Zの残りゲーム数「6, 7, 8ゲーム」「3, 4, 5ゲーム」「0, 1, 2ゲーム」に応じて異ならせることとしてもよい。このようにすることで、C Z中の各種抽籤をより多様化することができ、遊技性に富んだものにする事ができる。

【1749】

以上説明したように本実施形態のパチスロ1では、ART抽籤に当籤するとランク抽籤を行い、ART状態中のランクを決定する。なお、ART状態中は、このランク（より詳細には、ランクから定まる抽籤状態）に基づき、C Zゲーム数の上乘せやEPへの移行（より詳細には、ナビ高確ゲーム数の付与）といった特典が付与されることになるため、低いランクは小さな特典しか付与されず、反対に高いランクは大きな特典が付与されることになる。そのため、ART状態のランクは、遊技者に対して付与する特典の大きさということができる。

【1750】

このとき、パチスロ1では、ART状態のセット数のストックの有無に応じてランク抽籤の抽籤結果を異ならせる。例えば、ART状態のセット数のストックがある場合には、ランク抽籤において低いランクを決定し易く、反対に、ART状態のセット数のストックがない場合には、ランク抽籤において高いランクを決定し易くすることができる。これにより、ART状態のセット数のストックがある状態で更にART抽籤に当籤した場合には、ART状態のセット数のストックがない状態でART抽籤に当籤した場合よりも、低いランクが期待できることになり、その後のART状態における特典への期待が抑制される。

【1751】

このようにパチスロ1では、ART状態のセット数のストックがある場合には、ART状態のランクを相対的に低く決定するため、ストックがある状態での更なるART当籤時に遊技者に対して過度な利益を付与してしまうことを抑制することができる。

【1752】

なお、ART状態のセット数のストックがある場合には、ランク抽籤において高いランクを決定し易く、反対に、ART状態のセット数のストックがない場合には、ランク抽籤において低いランクを決定し易くすることもできる。このような場合であっても、2つ目以降のストック分のランクは高くなってしまふものの、1つ目のストック分のランクは低く抑えることができるため、遊技者に対して過度な利益を付与してしまうことを抑制することができる。

【1753】

また、パチスロ1では、ART状態のセット数のストックの有無に応じてART抽籤の抽籤結果も異ならせる。例えば、ART状態のセット数のストックがある場合にART抽籤に当籤し易くすることで、ストックがある場合には、ART抽籤に当籤し易いものの低いランクが決定され易くすることができ、また、ART状態のセット数のストックがある場合にART抽籤に当籤し難くすることで、ストックがある場合には、ART抽籤に当籤し難く、かつ、当籤しても低いランクが決定され易くすることができる。これにより、C Z中の各種抽籤をより多様化することができ、遊技性に富んだものにする事ができる。

【1754】

また、パチスロ1では、C Z中の残りゲーム数に応じてART抽籤及びランク抽籤の抽籤結果を異ならせる。これにより、C Z中の各種抽籤をより多様化することができ、遊技性に富んだものにする事ができる。

【1755】

なお、パチスロ1において、ART状態のランクは、C Zゲーム数の上乘せに影響を与える情報であるため、高いランクが決定された場合には、C Zゲーム数の上乘せに期待でき、結果、C ZとART状態とのループが継続することが期待でき、遊技性が向上する。

【1756】

〔主制御基板及び副制御基板が有する各種機能〕

以上のような本実施形態のパチスロ１に特有の制御を実現するために、本実施形態のパチスロ１の主制御基板（主制御回路９０，メインＣＰＵ１０１）及び副制御基板（副制御回路２００，サブＣＰＵ２０１）は、次のような機能を有する。

【１７５７】

主制御基板７１は、スタートスイッチ７９やストップスイッチ基板８０と接続され、図１に示す遊技の進行を制御する。それゆえ、主制御基板７１は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

【１７５８】

また、主制御基板７１は、ＡＲＴ抽籤を行いＡＲＴ状態に移行するか否か、及びＡＲＴ状態に移行する場合にＡＲＴ状態中の有利度合（すなわち、ＡＲＴ状態中のランク）を決定するため、特典決定手段、移行決定手段及びモード決定手段として機能する。

なお、上述のようにＡＲＴ状態中のランクは、ＡＲＴ状態中に付与する特典（例えば、ＣＺゲーム数の上乘せなど）に影響を与える情報であるため、ＡＲＴ状態という特典を付与する遊技機においては、ＡＲＴ状態のランクを決定することは、ＡＲＴ状態という特典の大きさを決定することと同じである（すなわち、ランクが高いほど、有利なＡＲＴ状態になるため、付与する特典も大きくなる）。

【１７５９】

また、主制御基板７１は、ＡＲＴ抽籤に当籤すると、遊技状態をＡＲＴ状態に移行させるとともに、このＡＲＴ状態中の遊技を、ランク抽籤で決定したランクに基づき制御（すなわち、ランクに基づいてＣＺゲーム数の上乘せなどの特典を付与）するため、特典付与手段及び有利状態制御手段として機能する。

【１７６０】

また、主制御基板７１は、ＡＲＴ抽籤に当籤すると、付与するＡＲＴ状態の権利数を決定して、メインＲＡＭ１０３に記憶するとともに、遊技状態をＡＲＴ状態に移行させると、メインＲＡＭ１０３に記憶されているＡＲＴ状態の権利を１つ消し去るため、権利管理手段として機能する。

【１７６１】

また、主制御基板７１は、ＣＺ中の遊技を制御するため、状態制御手段及び高確率状態制御手段として機能する。なお、主制御基板７１は、ＣＺ中にＡＲＴ抽籤に当籤すると、ＣＺを中断して遊技状態をＡＲＴ状態に移行させ、その後、このＡＲＴ状態が終了すると、中断していたＣＺを再開することで、ＣＺとＡＲＴ状態とのループを実現する。

【１７６２】

〔ＣＺ終了後の泣きの１回のＡＲＴ抽籤〕

本実施形態のパチスロ１では、通常状態中であっても、ＣＺが終了した次の１ゲームでは、当籤する確率が高いＡＲＴ抽籤を行う（図２２８、図２２９参照）。遊技者にとってみると、ＣＺが終了した後にもＡＲＴ状態への移行の期待度を持つことができるため、遊技の興趣の低下を抑制することができる。

【１７６３】

特に、本実施形態のパチスロ１では、ＣＺからＡＲＴ状態に移行すると、ＣＺの残りゲーム数が上乘せされるが、ＣＺが終了した次の１ゲームにおいてＡＲＴ抽籤に当籤した場合にも、ＣＺの残りゲーム数が上乘せされ、結果、ＣＺの終了に伴い終了したＣＺとＡＲＴ状態とのループが再開することになるため、遊技の興趣が向上する。なお、ＣＺの最終ゲーム（残りゲーム数「０」）に遊技状態がＡＲＴ状態に移行し、かつ、ＣＺゲーム数に「５ゲーム」が上乘せされた場合、ＡＲＴ状態終了時のＣＺゲーム数の残りは「５ゲーム」である。また、ＣＺが終了した次の１ゲームにおいてＡＲＴ抽籤に当籤し、遊技状態がＡＲＴ状態に移行し、かつ、ＣＺゲーム数に「５ゲーム」が上乘せされた場合も、ＡＲＴ状態終了時のＣＺゲーム数の残りは「５ゲーム」である。

【１７６４】

また、ＣＺが終了した次の１ゲームにおける演出内容は、任意である。例えば、ＣＺが終

10

20

30

40

50

了した次の1ゲームでは、CZとART状態とのループによる有利区間の結果を表示する終了画面を表示しておき、CZが終了した次の1ゲームにおけるART抽籤に当籤した場合には、この終了画面から逆転させる形で演出画面を切り替えることとしてもよい。また、CZとART状態とのループによる有利区間の結果は、CZの最終ゲームにおいて表示しておき、次のゲーム（CZが終了した次の1ゲーム）のスタート時には、通常状態中の通常の演出画面を表示しておくこととしてもよい。この場合、CZが終了した次の1ゲームにおいてART抽籤に当籤すると、例えば、第1停止操作～第3停止操作等の任意のタイミングで、通常の演出画面が確定画面に演出画面を切り替えることとしてもよい。

【1765】

また、本実施形態のパチスロ1では、CZが終了した次の1ゲームにおいて当籤確率の高いART抽籤を行うこととしているが、これに限られるものではない。継続CZ及びART状態は、高RT状態中に行われるため、例えば、CZが終了した後の低RT状態に移行したタイミングの1ゲームに限り、当籤確率の高いART抽籤を行うこととしてもよい。すなわち、主制御回路90は、継続CZの終了時は、RT状態がRT0～RT2状態の何れかに移行した次の1ゲームに限り、当籤確率の高いART抽籤を行い、また、初当りCZの終了時には、初当りCZの終了時のRT状態が既にRT0～RT2状態である場合は初当りCZが終了した次の1ゲームにおいて当籤確率の高いART抽籤を行い、また、初当りCZの終了時のRT状態がRT4状態又はRT5状態である場合は、初当りCZが終了した後にRT状態がRT0～RT2状態の何れかに移行した次の1ゲームに限り、当籤確率の高いART抽籤を行うこととしてもよい。

【1766】

また、ART抽籤は、抽籤フラグに基づいて行われるところ、抽籤用フラグは、内部当籤役とナビ区間の有無とに応じて決定されるため（図208（B）参照）、CZが終了した次の1ゲームをナビ区間とするか否かに応じて、CZが終了した次の1ゲームにおけるART抽籤の抽籤結果が異なることになる。この点、CZが終了した次の1ゲームをナビ区間としてもよく、また、非ナビ区間としてもよく、更には、ナビ高確中のナビ区間としてもよく、また非ナビ高確中のナビ区間としてもよい。

【1767】

【主制御基板及び副制御基板が有する各種機能】

以上のような本実施形態のパチスロ1に特有の制御を実現するために、本実施形態のパチスロ1の主制御基板（主制御回路90，メインCPU101）及び副制御基板（副制御回路200，サブCPU201）は、次のような機能を有する。

【1768】

主制御基板71は、スタートスイッチ79やストップスイッチ基板80と接続され、図1に示す遊技の進行を制御する。それゆえ、主制御基板71は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

【1769】

また、主制御基板71は、ART抽籤を行い、ART抽籤に当籤すると、遊技状態をART状態に移行させるため、付与決定手段及び特典付与手段として機能する。

また、主制御基板71は、ART抽籤に当籤すると、付与するART状態の権利数を決定して、メインRAM103に記憶するとともに、遊技状態をART状態に移行させると、メインRAM103に記憶されているART状態の権利を1つ消し去るため、特典管理手段として機能する。

【1770】

また、主制御基板71は、CZ中の遊技を制御するため、状態制御手段として機能する。なお、主制御基板71は、CZ中にART抽籤に当籤すると、CZを中断して遊技状態をART状態に移行させ、その後、このART状態が終了すると、中断していたCZを再開することで、CZとART状態とのループを実現するが、この際、CZからART状態への移行に伴いCZゲーム数の上乗せを行うため、高確率状態加算手段として機能する。

【1771】

[ART当籤時のランク抽籤]

本実施形態のパチスロ1では、ART抽籤に当籤した場合に、ランク抽籤を行う。例えば、主制御回路90は、ART抽籤において当籤したART状態の1つ目のストックについては、図236(A)～図240(N)に示すART当籤時ランク抽籤テーブルを参照して、ART当籤時の内部当籤役（より詳細には、抽籤用フラグ）に基づいてART当籤時のランクを決定し、また、2つ目以降のストックについては、図240(O)に示すART当籤時ランク抽籤テーブルを参照してART当籤時のランクを決定する。

【1772】

ART当籤時のランクは、ART状態の抽籤状態を決定するために参照されることから、ART当籤時に高ランクが決定されている場合には、その後のART状態においてCZゲーム数の上乘せ等が行われ易く、反対にART当籤時に低ランクが決定されている場合には、その後のART状態においてCZゲーム数の上乘せ等が行われ難くなる可能性があり、ART状態中の遊技が画一化してしまうことを防止できる。また、ART当籤時のランクは、ART当籤時の内部当籤役に基づいて決定されることから、CZ中に強い役を契機にART抽籤に当籤した場合には、その後のART状態に期待を持つことができ、遊技の興趣が向上する。

【1773】

また、このART当籤時のランクは、副制御回路200の制御により表示装置11を介して示唆される（図212(C)参照）。具体的には、副制御回路200は、1つ目のストック（すなわち、内部当籤役に基づいてART当籤時のランクが決定されるストック）については、「武器S」「武器A」「武器B」「武器C」「武器D」に応じた映像を表示することで、当該1つ目のストック分のART当籤時のランクを示唆する演出を行う。一方で、2つ目以降のストック分については、副制御回路200は、「武器？」に応じた映像を表示することで、ART当籤時のランクを示唆しない。

【1774】

なお、副制御回路200がART当籤時のランクを示唆する又は示唆しない演出を行うタイミングは、主制御回路90がART状態のストックを放出するタイミング（すなわち、付与したART状態のストックに基づき、遊技状態をART状態に移行させるタイミング）である。ここで、CZ中のART抽籤において一度に複数のART状態のストックが付与された場合、主制御回路90は、ART当籤後に遊技状態を一度CZからART状態に移行させることで、1つ目のストックの放出を行い、また、その後、CZの残りゲーム数がなくなったタイミングで、遊技状態をCZからART状態に移行させることで、2つ目のストックを放出する。そのため、副制御回路200は、1つ目のストック分は、ART当籤時に武器ランクに応じた映像を表示することで、ランクを示唆し、2つ目以降のストック分は、CZの残りゲーム数がなくなった時（放出フェーズD）に「武器？」に応じた映像を表示することで、ART当籤時のランクを示唆しない。

【1775】

また、ART状態のランク（抽籤状態）は、ART当籤時のランク抽籤と、ランク決めART中の昇格抽籤という2段階の抽籤を経て決定される。副制御回路200は、ART当籤時の（仮の）ランクを、武器ランクに応じた映像を表示して示唆する一方で、ART状態中に抽籤状態に応じた演出ステージを用いて演出を行うことで、昇格抽籤後の確定したランク（抽籤状態）を報知する（図212(C)参照）。遊技者にとってみれば、ART状態中の演出ステージから、今回のART状態の有利度合を把握できるため、その後の遊技に対して様々期待を持つことができ、興趣が向上する。

【1776】

[主制御基板及び副制御基板が有する各種機能]

以上のような本実施形態のパチスロ1に特有の制御を実現するために、本実施形態のパチスロ1の主制御基板（主制御回路90，メインCPU101）及び副制御基板（副制御回路200，サブCPU201）は、次のような機能を有する。

【1777】

主制御基板71は、スタートスイッチ79やストップスイッチ基板80と接続され、図1に示す遊技の進行を制御する。それゆえ、主制御基板71は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

【1778】

また、主制御基板71は、ART抽籤を行いART状態に移行するか否かを決定し、ART抽籤に当籤した場合に遊技状態をART状態に移行するため、移行決定手段、有利状態制御手段、特典決定手段及び特典付与手段として機能する。また、主制御基板71は、ART抽籤に当籤した場合にランク抽籤を行い、ART当籤時の内部当籤役から定まる抽籤用フラグに基づきART当籤時のランクを決定するとともに、ランク決めART中に昇格抽籤を行うことでART当籤時のランクからART状態の抽籤状態を決定するため、モード決定手段、モード決定情報決定手段及び特典決定手段（なお、この場合における特典は、ART状態への移行という特典を意味する）として機能する。

10

【1779】

また、主制御基板71は、ART状態中に、ART当籤時のランク（より詳細には、当該ランクから決定される抽籤状態）に基づき、ART状態中にCZゲーム数の上乗せなどの特典を付与するため、特典付与手段として機能する。

【1780】

また、主制御基板71は、ART抽籤に当籤すると、付与するART状態の権利数を決定して、メインRAM103に記憶するとともに、遊技状態をART状態に移行させると、メインRAM103に記憶されているART状態の権利を1つ消し去るため、権利管理手段として機能する。

20

【1781】

また、主制御基板71は、CZ中の遊技を管理してCZの残りゲーム数が0になるとCZを終了することから、高確率状態制御手段及び状態制御手段として機能する。

【1782】

また、副制御基板200及び表示装置11は、ART当籤時のランクを示唆する演出を行うとともに、一度のART当籤時に複数ストックした場合に、1つ目のストック分のART当籤時のランクは示唆する一方で、2つ目以降のストック分のART当籤時のランクは示唆しないため、演出手段として機能する。

30

【1783】

[ランク決めART中のランク昇格抽籤]

本実施形態のパチスロ1では、ART状態に移行した最初の遊技（ランク決めART）において、ART当籤時に決定していたランクを昇格して、ART状態中の抽籤状態を決定する。そして、パチスロ1では、ART状態中に、この抽籤状態に応じてCZゲーム数の上乗せやナビ高確ゲーム数を付与するため、抽籤状態に応じてART状態中の遊技性が異なることになる。

【1784】

ここで、パチスロ1では、抽籤状態を、ART当籤時のランク抽籤と、ランク決めART中の昇格抽籤という二段階の抽籤を経て決定するため、例えば、ランク抽籤の抽籤結果が好ましくない場合であっても、昇格抽籤の抽籤結果により好ましい結果となることがあり、ART状態中の遊技性を多様化することができる。また、ART当籤時のランク抽籤は、ART当籤時の内部当籤役（より詳細には、内部当籤役から定まる抽籤用フラグ）に基づき行われ、昇格抽籤は、ランク決めART中の内部当籤役（より詳細には、内部当籤役から定まる抽籤用フラグ）に基づき行われるため、例えば、強い役を契機にART抽籤に当籤した場合には、その後のART状態に期待を持てるだけでなく、弱い役を契機にART抽籤に当籤した場合であっても、ランク決めART中に強い役が当籤することでその後のART状態に期待を持てることができ、遊技の興味が向上する。

40

【1785】

50

また、ART状態中に抽籤状態に基づいてCZゲーム数の上乘せといった特典が付与された場合には、その後、CZとART状態とのループが期待でき、遊技の興趣が更に向上する。

【1786】

〔主制御基板及び副制御基板が有する各種機能〕

以上のような本実施形態のパチスロ1に特有の制御を実現するために、本実施形態のパチスロ1の主制御基板（主制御回路90，メインCPU101）及び副制御基板（副制御回路200，サブCPU201）は、次のような機能を有する。

【1787】

主制御基板71は、スタートスイッチ79やストップスイッチ基板80と接続され、図1に示す遊技の進行を制御する。それゆえ、主制御基板71は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

【1788】

また、主制御基板71は、ART抽籤を行いART状態に移行するか否かを決定し、ART抽籤に当籤した場合に遊技状態をART状態に移行するため、移行決定手段及び有利状態制御手段として機能する。また、主制御基板71は、ART抽籤に当籤した場合にランク抽籤を行い、ART当籤時の内部当籤役から定まる抽籤用フラグに基づきART当籤時のランクを決定するため、モード決定情報決定手段として機能する。

【1789】

また、主制御基板71は、ランク決めART中に内部当籤役から定まる抽籤用フラグに基づき昇格抽籤を行い、ART当籤時のランクを昇格させてART状態中の抽籤状態を決定するため、モード決定手段として機能する。なお、本実施形態のパチスロ1では、ART状態の最初の1ゲームをランク決めARTとして昇格抽籤を行っているが、これに限られるものではなく、ART抽籤に当籤した遊技よりも後の遊技において昇格抽籤を行うこととしてもよい。

【1790】

また、主制御基板71は、ART状態中に、昇格抽籤において決定された抽籤状態に基づき、CZゲーム数の上乘せなどの特典を付与するため、特典付与手段として機能する。

【1791】

〔通常状態中の自力RT5移行時のART抽籤〕

本実施形態のパチスロ1では、RT移行図柄に基づきRT状態を移行しており、通常状態中に押し順に正解した場合、通常状態中であっても高RT状態であるRT4状態やRT5状態に移行することがある。例えば、図266（A）に示すようにRT4状態において「3択昇格リブ」が内部当籤役として決定された場合に、当籤した「3択昇格リブ」の押し順に正解すると、略称「強チャンスリブ（RT5移行図柄）」に係る図柄組合せが停止表示され（図205参照）、RT状態がRT4状態からRT5状態に移行する。

【1792】

ここで、図226に示す通常時ART抽籤テーブルを参照すると、略称「強チャンスリブ（RT5移行図柄）」に係る図柄組合せが停止表示されたときの抽籤用フラグ「強チャンスリブ」は、通常時のART抽籤に当籤し易い抽籤用フラグであることが分かる。

【1793】

また、RT5状態では、通常時のART抽籤に必ず当籤する抽籤用フラグ「7揃い」「BAR揃い」に対応する「F__7リブA」「F__7リブB」「F__BARリブ」が内部当籤役として決定される確率が高い（図180参照）。そのため、通常状態中のRT5状態は、通常状態中のRT4状態に比べてART抽籤に当籤する確率が高い状態であるといえる。

【1794】

また、図266（A）に示すようにRT5状態において「6択転落リブ」が内部当籤役として決定された場合に、当籤した「6択転落リブ」の押し順に不正解すると、略称「小山リプレイ（RT4移行図柄）」に係る図柄組合せが停止表示され、RT状態がRT5状態

10

20

30

40

50

から R T 4 状態に移行（転落）してしまうが、当籤した「6 択転落リプ」の押し順に正解するとコンビネーション名「C__CUリプ」に係る図柄組合せが停止表示され（図 2 0 5 参照）、結果、R T 状態が R T 5 状態のまま維持される。

【1 7 9 5】

R T 状態を R T 5 状態のまま維持できた場合には、R T 5 状態中の高確率で当籤する A R T 抽籤をその後も受けることができ好適であるが、R T 状態が R T 4 状態に転落してしまった場合であっても、図 2 2 3 の状態別 C Z 抽籤テーブルを参照すると、転落時に表示される略称「小山リプレイ」に係る図柄組合せに対応する抽籤用フラグ「弱チャンスリプ」は、R T 状態の維持時に表示されるコンビネーション名「C__CUリプ」に係る図柄組合せに対応する「通常リプ」よりも通常時の C Z 抽籤に当籤し易い。

10

【1 7 9 6】

このように本実施形態のパチスロ 1 では、R T 4 状態から R T 5 状態に移行したタイミング（すなわち、R T 5 状態の入り口）で A R T 抽籤の当籤に期待を持つことができ、また、R T 5 状態から R T 4 状態に転落したタイミング（すなわち、R T 5 状態の出口）で C Z 抽籤の当籤に期待を持つことができる。言い換えると、パチスロ 1 では、A R T 抽籤に高確率で当籤する R T 5 状態への入り口と出口との双方において、特典（A R T 状態や C Z）の付与が期待できる。

【1 7 9 7】

なお、本実施形態のパチスロ 1 では、通常状態中は何らの報知（ナビ）を行わない例を示しているが、通常状態中も所定の報知を行うこととしてもよい。図 2 6 6（B）は、通常状態中のナビ制御の一例を示す図である。図 2 6 6（B）に示すように、パチスロ 1 は、通常状態中のナビ状態として、通常ナビ状態と非ナビ状態とを備えることができる。

20

【1 7 9 8】

通常ナビ状態とは、一定の範囲で遊技者に対して押し順を報知する状態であり、非ナビ状態とは、何らの報知も行わない状態である。一定の範囲は任意であるが、パチスロ 1 では、例えば、R T 5 状態に移行するための「3 択昇格リプ」の当籤時に正解の押し順を報知することができる。このようにすることで、通常ナビ状態中は、A R T 抽籤に高確率で当籤する R T 5 状態への移行が容易になる。

【1 7 9 9】

また、通常ナビ状態中は、「3 択昇格リプ」の当籤時の報知に代えて又は加えて、R T 4 状態への転落を回避するための「6 択転落リプ」の当籤時に正解の押し順を報知することとしてもよい。このようにすることで、通常ナビ状態中は、一度 R T 5 状態に移行できると、その後、R T 4 状態への転落を回避しつつ、R T 5 状態の A R T 抽籤を受けることができる。

30

【1 8 0 0】

なお、通常状態中に非ナビ状態から通常ナビ状態に移行させるための方法は任意であり、例えば、抽籤用フラグに基づいて抽籤により決定することとしてもよく、また、R T 5 状態へ唯一移行することができる R T 4 状態に移行したタイミングで所定の確率で通常ナビ状態にすることとしてもよい。ここで、図 1 7 8 を参照すると、R T 4 状態へは、R T 2 状態中の「6 択突入リプ」の当籤時に押し順に正解することで（略称「小山リプレイ」に係る図柄組合せが停止表示され）移行する。そのため、例えば、パチスロ 1 の主制御回路 9 0 は、R T 2 状態から R T 4 状態に移行したタイミングで通常ナビ状態に移行するか否かを決定することとしてもよい。

40

【1 8 0 1】

また、通常ナビ状態から非ナビ状態に移行させるための方法も任意であり、例えば、抽籤用フラグに基づいて抽籤により決定することとしてもよく、また、通常ナビ状態中の遊技が所定回数行われることを条件に通常ナビ状態から非ナビ状態に移行させることとしてもよく、また、通常ナビ状態中に行った報知の回数が特定回数に達することを条件に通常ナビ状態から非ナビ状態に移行させることとしてもよい。

【1 8 0 2】

50

また、本実施形態のパチスロ1では、「F__7リプA」「F__7リプB」「F__BARリプ」が内部当籤役として決定されると、停止操作の態様に関わらず略称「7揃いリプ」「BAR揃いリプ」に係る図柄組合せが停止表示される（図205参照）が、これに限られるものではない。すなわち、「F__7リプA」「F__7リプB」「F__BARリプ」が内部当籤役として決定された場合に、停止操作の態様が予め定められた態様である場合（例えば、押し順に正解すると）略称「7揃いリプ」「BAR揃いリプ」に係る図柄組合せが停止表示され、停止操作の態様が予め定められた態様ではない場合（例えば、押し順に不正解すると）略称「7揃いリプ」「BAR揃いリプ」に係る図柄組合せが停止表示されない（例えば、略称「フェイクリプ」に係る図柄組合せが停止表示される）こととしてもよい。

10

【1803】

この場合、RT5状態中に「F__7リプA」「F__7リプB」「F__BARリプ」が内部当籤役として決定されると、リール停止時に抽籤用フラグを決定し、当該抽籤用フラグによりART抽籤を行うことになる。すなわち、主制御回路90は、RT5状態中に「F__7リプA」「F__7リプB」「F__BARリプ」が内部当籤役として決定された場合、略称「7揃いリプ」「BAR揃いリプ」に係る図柄組合せが停止表示されると、抽籤用フラグ「7揃い」「BAR揃い」に基づいてART抽籤を行う。

【1804】

また、このように「F__7リプA」「F__7リプB」「F__BARリプ」を押し順役として用いる場合、上述の通常ナビ状態中に正解の押し順を報知することとしてもよい。これにより、通常ナビ状態中は、RT5状態中に高確率でART抽籤に当籤することになるが、非ナビ状態中は、RT5状態中に押し順に正解しないとART抽籤に当籤なくなり、遊技性が多様化する。

20

【1805】

なお、本実施形態のパチスロ1では、通常状態中にART抽籤に当籤すると、遊技状態をART前兆に移行し、その後、ART準備中を経て遊技状態をART状態に移行することになっている。この点、RT5状態は、ART状態が行われる高RT状態であるため、RT5状態の入り口である、抽籤用フラグ「強チャンスリプ」に基づく通常時のART抽籤に当籤した場合、又はRT5状態中に抽籤用フラグ「7揃い」「BAR揃い」に基づく通常所のART抽籤に当籤した場合には、ART前兆に移行することなく、ART状態に移行させることとしてもよい。

30

【1806】

〔主制御基板及び副制御基板が有する各種機能〕

以上のような本実施形態のパチスロ1に特有の制御を実現するために、本実施形態のパチスロ1の主制御基板（主制御回路90，メインCPU101）及び副制御基板（副制御回路200，サブCPU201）は、次のような機能を有する。

【1807】

主制御基板71は、スタートスイッチ79やストップスイッチ基板80と接続され、図1に示す遊技の進行を制御する。それゆえ、主制御基板71は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

40

【1808】

また、主制御基板71は、RT4状態中に略称「強チャンスリプ（RT5移行図柄）」に係る図柄組合せが停止表示されると、RT状態をRT5状態に移行させ、RT5状態中に略称「小山リプレイ（RT4移行図柄）」に係る図柄組合せが停止表示されると、RT状態をRT4状態に移行させるため、RT移行手段として機能する。

【1809】

また、主制御基板71は、抽籤用フラグに基づきART抽籤やCZ抽籤を行っており、この抽籤に当籤すると、遊技状態をART状態やCZ状態に移行するため、付与決定手段及び特典付与手段として機能する。なお、主制御基板71が行うART抽籤は、「3択昇格

50

リップ」の当籤時に略称「強チャンスリップ」に係る図柄組合せが停止表示されると、所定の確率で当籤し、また、「F__7リップA」「F__7リップB」「F__BARリップ」の当籤時（押し順役として用いる場合には、略称「7揃いリップ」「BAR揃いリップ」に係る図柄組合せの停止表示時）には、必ず当籤する。また、主制御基板71が行うCZ抽籤は、「6折転落リップ」当籤時に略称「小山リップ」に係る図柄組合せが停止表示されると、コンビネーション名「C__CUリップ」に係る図柄組合せが停止表示された場合よりも高い確率で当籤する。

【1810】

また、主制御基板71は、通常状態中に通常ナビ状態に移行するか否かを決定し、通常ナビ状態中は、一定の範囲で押し順を報知するため、報知決定手段、通常時報知制御手段及び報知手段として機能する。なお、押し順の報知は、液晶などのサブ側で制御する装置により行うこともでき、この場合には、副制御基板72も報知手段として機能する。

【1811】

〔RT3状態への移行を契機としたART状態への移行〕

本実施形態のパチスロ1では、RT移行図柄に基づきRT状態を移行しており、RT2状態に略称「弱チェリップ（RT3移行図柄）」に係る図柄組合せが停止表示されると、RT状態がRT3状態に移行し、また、ART状態のセット数のストックという特典が付与される（図209（B-2）参照）。なお、略称「弱チェリップ（RT3移行図柄）」に係る図柄組合せは、「F__弱チェリップ」が内部当籤役として決定された場合に停止表示されるため、RT状態がRT3状態に移行することと、RT2状態中に略称「弱チェリップ（RT3移行図柄）」に係る図柄組合せが停止表示されることと、RT2状態中に「F__弱チェリップ」が内部当籤役として決定されることとは、同じことを意味する。

【1812】

また、本実施形態では、RT状態がRT3状態に移行することを契機に、必ずART状態のセット数を付与することとしているが、これに限られるものではなく、RT状態がRT3状態に移行した場合に、所定の確率でART状態のセット数を付与することとしてもよい。

【1813】

ここで、RT3状態中は、「F__弱チェリップ」が高確率（18752／65536）で内部当籤役として決定されるため（図180参照）、RT3状態に移行できた場合には略称「弱チェリップ」に係る図柄組合せが頻繁に表示されることになる。遊技者にとってみれば、略称「弱チェリップ」に係る図柄組合せが頻繁に表示されることで、RT3状態中であること、すなわち、ART状態のストックが付与されていることについて期待感を抱くことになる。このように本実施形態のパチスロ1では、「F__弱チェリップ」が連続して内部当籤役として決定されている場合に、ART状態のストックという特典が付与されている期待感を抱くことができる。

【1814】

また、パチスロ1では、「F__弱チェリップ」を所定の確率で内部当籤役として決定するものの、略称「弱チェリップ」に係る図柄組合せが停止表示されても、他のRT状態に移行しないRT状態として、RT1状態を有する。ここで、このようなRT1状態を有しない場合、略称「弱チェリップ」に係る図柄組合せが停止表示されると、RT3状態に移行してしまうことになるため、「F__弱チェリップ」が連続して当籤するか否かに関わらず、一度の略称「弱チェリップ」に係る図柄組合せの停止表示から、ART状態のストックという特典が付与されている期待感を抱くことができてしまう。これに対して、略称「弱チェリップ」に係る図柄組合せが停止表示されてもRT3状態に移行しないRT状態を設けることで、「F__弱チェリップ」の連続当籤から初めてRT3状態に滞在中であることを期待できるようになり、結果、「F__弱チェリップ」の連続当籤から初めてART状態のストックという特典が付与されている期待感を抱くことができる。

【1815】

また、RT3状態への移行を契機に付与される特典は、高RT状態中に行うART状態で

あるところ、高 R T 状態中は、略称「弱チェリプ」に係る図柄組合せが停止表示されることになる「F__弱チェリプ」を内部当籤役として決定することがなく（図 180 参照）、パチスロ 1 では、高 R T 状態から R T 3 状態に移行することがない。これにより、それぞれの R T 状態の役割を明確にすることができる。

【1816】

なお、R T 3 状態中は、「F__弱チェリプ」が内部当籤役として決定される確率が高いが、R T 3 状態中の「F__弱チェリプ」当籤に伴う各種の抽籤は、任意に設定することができる。すなわち、R T 3 状態中の「F__弱チェリプ」の当籤時には、A R T 抽籤及び C Z 抽籤に必ず非当籤（又は抽籤自体を行わない）することとしてもよく、また、R T 3 状態中であっても「F__弱チェリプ」の当籤時に A R T 抽籤及び C Z 抽籤に所定の確率で当籤することとしてもよい。なお、R T 3 状態中の「F__弱チェリプ」の当籤時における A R T 抽籤及び C Z 抽籤の当籤確率は、図 223 に示す状態別 C Z 抽籤テーブル又は図 226 に示す通常時 A R T 抽籤テーブルにおける抽籤用フラグ「弱チェリー」に対応する抽籤値を規定することで、任意に設定することができる。

10

【1817】

〔主制御基板及び副制御基板が有する各種機能〕

以上のような本実施形態のパチスロ 1 に特有の制御を実現するために、本実施形態のパチスロ 1 の主制御基板（主制御回路 90，メイン C P U 101）及び副制御基板（副制御回路 200，サブ C P U 201）は、次のような機能を有する。

【1818】

主制御基板 71 は、スタートスイッチ 79 やストップスイッチ基板 80 と接続され、図 1 に示す遊技の進行を制御する。それゆえ、主制御基板 71 は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

20

【1819】

また、主制御基板 71 は、無限 R T 状態中に略称「弱チェリプ（R T 3 移行図柄）」に係る図柄組合せが停止表示されると、R T 状態を R T 3 状態に移行させる一方で、有限 R T 状態中に略称「弱チェリプ（R T 3 移行図柄）」に係る図柄組合せが停止表示されても、R T 状態を移行させずに維持し、また、R T 3 状態中に 20 回の遊技が行われると、R T 状態を R T 3 状態から R T 0 状態に移行させるため、R T 移行手段として機能する。

30

【1820】

また、主制御基板 71 は、R T 状態が R T 3 状態に移行すると、A R T 状態のセット数を付与するとともに、その後、R T 状態が R T 3 状態から高 R T 状態に移行すると遊技状態を A R T 状態に移行することから、付与決定手段及び特典付与手段として機能する。

【1821】

〔押し順ベル当籤時の制御〕

本実施形態のパチスロ 1 では、押し順ベルの当籤時に停止操作の順序（押し順）に応じて有効ライン上に停止表示する図柄組合せを異ならせる。通常の A R T 機では、押し順ベル当籤時の押し順が不正解である場合、何らかの不利益（例えば、R T 状態の移行）を受けることになるが、本実施形態のパチスロ 1 では、押し順が不正解の場合であっても、何らかの不利益も受けないことがある。以下、図 267～図 269 を参照して、本実施形態のパチスロ 1 における押し順ベル当籤時の制御について説明する。

40

【1822】

なお、以下において「1 停ミス」とは、第 1 停止操作で操作すべきストップボタンの種別を間違えること（言い換えると、1 番目の押し順を間違えること）をいい、同様に「2 停ミス」は、第 2 停止操作で操作すべきストップボタンの種別を間違えること（言い換えると、2 番目の押し順を間違えること）をいい、また、「3 停ミス」は、第 3 停止操作で操作すべきストップボタンの種別を間違えること（言い換えると、3 番目の押し順を間違えること）をいう。また、反対に、第 1 停止操作で操作すべきストップボタンの種別が正しいこと（言い換えると、1 番目の押し順が正しいこと）を、「1 停正解」といい、第 2 停

50

止操作で操作すべきストップボタンの種別が正しいこと（言い換えると、2番目の押し順が正しいこと）を、「2 停止解」といい、第3 停止操作で操作すべきストップボタンの種別が正しいこと（言い換えると、3番目の押し順が正しいこと）を、「3 停止解」という。

【1823】

また、図267において、コンビネーション名「C__TPベル」とは、略称「ベル」に係る図柄組合せのうちのコンビネーション名「C__TPベル__01」～「C__TPベル__18」をいい、コンビネーション名「C__CTベル」とは、略称「ベル」に係る図柄組合せのうちのコンビネーション名「C__CTベル__01」～「C__CTベル__06」をいい、コンビネーション名「C__CUベル」とは、略称「ベル」に係る図柄組合せのうちのコンビネーション名「C__CUベル__01」～「C__CUベル__18」をいい、コンビネーション名「C__BTベル」とは、略称「ベル」に係る図柄組合せのうちのコンビネーション名「C__BTベル__01」～「C__BTベル__09」をいい、コンビネーション名「C__CDベル」とは、略称「ベル」に係る図柄組合せのうちのコンビネーション名「C__CDベル__01」～「C__CDベル__BB」をいう。

【1824】

また、コンビネーション名「C__左移行役」とは、略称「RT0 移行図柄」に係る図柄組合せのうちのコンビネーション名「C__左移行役__01」～「C__左移行役__03」をいい、コンビネーション名「C__中移行役」とは、略称「RT0 移行図柄」に係る図柄組合せのうちのコンビネーション名「C__中移行役A__01」～「C__中移行役B__06」をいい、コンビネーション名「C__右移行役」とは、略称「RT0 移行図柄」に係る図柄組合せのうちのコンビネーション名「C__右移行役__01」～「C__右移行役__06」をいう。

【1825】

また、コンビネーション名「R__3rd移行」とは、略称「RT2 移行図柄」に係る図柄組合せのうちのコンビネーション名「R__3rd移行__01」～「R__3rd移行__18」をいい、コンビネーション名「R__2nd移行」とは、略称「RT2 移行図柄」に係る図柄組合せのうちのコンビネーション名「R__2nd移行__01」～「R__2nd移行__09」をいい、コンビネーション名「R__1st移行」とは、略称「RT2 移行図柄」に係る図柄組合せのうちのコンビネーション名「R__1st移行__01」～「R__1st移行__18」をいう（図197～図204参照）。

【1826】

図267は、押し順ベル当籤時の押し順と停止表示される図柄組合せとの対応関係を示す図である。図267に示すように、内部当籤役「押し順ベル（「F__123ベルA」～「F__321ベルB」）」は、押し順に応じて表示される図柄組合せが異なり、押し順が正解である場合には、コンビネーション名「C__TPベル」に係る図柄組合せのうちの図183～図196に示した表示可能な図柄組合せのいずれかが有効ラインに沿って表示される。

【1827】

一方、押し順が正解でない場合には、押し順を間違えた順番に応じて停止表示される図柄組合せが異なり、1 停ミス時には、コンビネーション名「C__CDベル」又は略称「RT2 移行図柄（「R__3rd移行」「R__2nd移行」「R__1st移行」）」に係る図柄組合せのうちの、図267に示す図柄組合せが停止表示される。例えば、内部当籤役が「F__123ベルA」～「F__132ベルB」である場合、1 番目の正解の押し順は、左であるため、押し順が「123」「132」であるときは1 停止解となり、押し順が「213」「231」「312」「321」であるときは1 停ミスとなる。

【1828】

また、2 停ミス時には、略称「RT0 移行図柄（「C__左移行役」「C__中移行役」「C__右移行役」）」に係る図柄組合せのうちの、図267に示す図柄組合せが停止表示される。例えば、内部当籤役が「F__123ベルA」「F__123ベルB」である場合、2 番目の正解の押し順は、中であるため、押し順が「123」であるときは2 停止解となり、

押し順が「1 3 2」であるときは2 停ミスとなり、押し順が「2 1 3」「2 3 1」「3 1 2」「3 2 1」であるときは1 停ミスとなる。

【1 8 2 9】

このように本実施形態のパチスロ1では、押し順ベル当籤時の押し順に不正解の場合、1 停ミス時には、コンビネーション名「C __ C Dベル」又は略称「R T 2 移行図柄」に係る図柄組合せが停止表示され、また、2 停ミス時には、略称「R T 0 移行図柄」に係る図柄組合せが停止表示される。なお、コンビネーション名「C __ C Dベル」は、R T 移行図柄ではないため、本実施形態のパチスロ1では、押し順不正解時であってもR T 状態が移行しないことがある。また、略称「R T 2 移行図柄」と略称「R T 0 移行図柄」とでは、移行先のR T 状態が異なるため、本実施形態のパチスロ1では、押し順不正解時の移行先のR T 状態が異なることになる。一方で、略称「ベル」は、R T 移行図柄ではないため、略称「ベル」に係る図柄組合せが停止表示されてもR T 状態は移行しない（すなわち、主制御回路90は、R T 状態を現在のR T 状態のまま維持する）。

【1 8 3 0】

また、1 停ミス時には、コンビネーション名「C __ C Dベル」又は略称「R T 2 移行図柄」に係る図柄組合せが停止表示されることになるが、本実施形態のパチスロ1では、停止操作のタイミングに応じて停止表示する図柄組合せを異ならせている。ここで、図267を参照すると、パチスロ1では、4つの押し順ベルを1つのグループにしていることが分かる。具体的には、パチスロ1では、「F __ 1 2 3ベルA」「F __ 1 2 3ベルB」「F __ 1 3 2ベルA」「F __ 1 3 2ベルB」を1番目の正解の押し順が左である押し順ベルのグループとし、同様に、「F __ 2 1 3ベルA」「F __ 2 1 3ベルB」「F __ 2 3 1ベルA」「F __ 2 3 1ベルB」を1番目の正解の押し順が中である押し順ベルのグループとし、「F __ 3 1 2ベルA」「F __ 3 1 2ベルB」「F __ 3 2 1ベルA」「F __ 3 2 1ベルB」を1番目の正解の押し順が右である押し順ベルのグループとしている。

【1 8 3 1】

パチスロ1では、これら4つの押し順ベルのそれぞれにおいて、1 停ミス時にコンビネーション名「C __ C Dベル」に係る図柄組合せが停止表示されることになる停止操作のタイミングを異ならせている。一例として、「F __ 1 2 3ベルA」～「F __ 1 3 2ベルB」のグループにおける1 停ミス時の停止制御について説明すると、内部当籤役が「F __ 1 2 3ベルA」である場合、有効ライン上に図柄位置「0」～「4」の図柄が位置しているタイミングで停止操作が行われるとコンビネーション名「C __ C Dベル」に係る図柄組合せが停止表示され、有効ライン上に図柄位置「5」～「20」の図柄が位置しているタイミングで停止操作が行われると略称「R T 2 移行図柄」が停止表示される。また、内部当籤役が「F __ 1 2 3ベルB」である場合、有効ライン上に図柄位置「5」～「9」の図柄が位置しているタイミングで停止操作が行われるとコンビネーション名「C __ C Dベル」に係る図柄組合せが停止表示され、有効ライン上に図柄位置「0」～「4」「10」～「20」の図柄が位置しているタイミングで停止操作が行われると略称「R T 2 移行図柄」が停止表示される。

【1 8 3 2】

また、内部当籤役が「F __ 1 3 2ベルA」である場合、有効ライン上に図柄位置「10」～「14」の図柄が位置しているタイミングで停止操作が行われるとコンビネーション名「C __ C Dベル」に係る図柄組合せが停止表示され、有効ライン上に図柄位置「0」～「9」「15」～「20」の図柄が位置しているタイミングで停止操作が行われると略称「R T 2 移行図柄」が停止表示される。また、内部当籤役が「F __ 1 3 2ベルB」である場合、有効ライン上に図柄位置「15」～「20」の図柄が位置しているタイミングで停止操作が行われるとコンビネーション名「C __ C Dベル」に係る図柄組合せが停止表示され、有効ライン上に図柄位置「0」～「14」の図柄が位置しているタイミングで停止操作が行われると略称「R T 2 移行図柄」が停止表示される。

【1 8 3 3】

このようにすることで、本実施形態のパチスロ1では、1 停ミス時であって4分の1の確

率で R T 状態が移行しないコンビネーション名「C __ C D ベル」に係る図柄組合せを停止表示することができる。

【1834】

ここで、押し順ベル当籤時に表示される図柄組合せと R T 制御の概念図を図 268 及び図 269 に示す。図 268 (A) は、本実施形態のパチスロ 1 における押し順ベル当籤時に表示される図柄組合せと R T 制御を示し、図 268 (B) ~ 図 269 (D) は、変形例に係るパチスロ 1 における押し順ベル当籤時に表示される図柄組合せと R T 制御を示す。

【1835】

図 268 (A) に示すように、本実施形態のパチスロ 1 において、主制御回路 90 は、押し順正解時には、略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8 枚のメダルを払い出す。また、主制御回路 90 は、1 停ミス時は、停止操作のタイミングに応じて図柄組合せを異ならせ、4 分の 1 の確率でコンビネーション名「C __ C D ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8 枚のメダルを払い出し、4 分の 3 の確率で略称「R T 2 移行図柄」に係る図柄組合せが停止表示されるように停止制御を行い、結果、R T 状態を R T 2 状態に移行する。また、主制御回路 90 は、2 停ミス時は、停止操作のタイミングに関わらず略称「R T 0 移行図柄」に係る図柄組合せが停止表示されるように製紙制御を行い、結果、R T 状態を R T 0 状態に移行する。

【1836】

なお、本実施形態のパチスロ 1 (以下に説明する別例のパチスロ 1 も同様) では、押し順不正解時に停止操作のタイミングに応じて停止表示される図柄組合せを異ならせる場合に、4 分の 1 の確率を採用しているが、4 分の 1 に限られるものではなく、グループ化する押し順ベルの数を任意に設定することで、2 分の 1、3 分の 1 又は 5 分の 1 等の任意の確率を採用することができる。

【1837】

また、本実施形態のパチスロ 1 (以下に説明する別例のパチスロ 1 も同様) では、1 停ミス時と 2 停ミス時とを比較した停止制御の例について説明しているが、これに限られるものではなく、1 停ミス時と 3 停ミス時とを比較することとしてもよく、また、2 停ミス時と 3 停ミス時とを比較することとしてもよく、また、停止対象のリール数が 4 つ以上である場合には、更に任意の順序を比較することとしてもよい。また、比較する順序も 2 つに限るものではなく、3 つ以上の順序を比較して停止制御を行うこととしてもよい。すなわち、パチスロ 1 では、所定番目の押し順の正否と所定番目よりも後の特定番目の押し順の正否とを比較して停止制御を行うものである。

【1838】

また、本実施形態のパチスロ 1 (以下に説明する別例のパチスロ 1 も同様) では、1 停ミス時は R T 2 状態に移行し、2 停ミス時は R T 0 状態に移行するというように、1 停ミス時と 2 停ミス時とで移行する R T 状態を異ならせているが、移行先の R T 状態は任意に設定することができる。また、移行先の R T 状態は、必ずしも異なる R T 状態である必要もなく、同じ R T 状態であってもよい。

【1839】

また、本実施形態のパチスロ 1 (以下に説明する別例のパチスロ 1 も同様) では、押し順不正解時に停止操作のタイミングに応じて停止表示される図柄組合せを異ならせている。ここで、上述の例では、停止操作のタイミングを参照する対象となる停止操作の種別を省略しているが、パチスロ 1 では、任意の順序の停止操作のタイミングを参照して、上述の停止制御を行うことができる。

【1840】

例えば、本実施形態のパチスロ 1 のように、1 停ミス時に停止操作のタイミングに応じて停止表示される図柄組合せを異ならせる場合、押し順をミスした第 1 停止操作のタイミングに応じて停止表示される図柄組合せを異ならせることとしてもよく、また、押し順をミスした第 1 停止操作よりも後の第 2 停止操作 (又は第 3 停止操作) のタイミングに応じて

10

20

30

40

50

停止表示される図柄組合せを異ならせることとしてもよい。また、後述するように、2 停ミス時に停止操作のタイミングに応じて停止表示される図柄組合せを異ならせる場合には、押し順をミスした第2 停止操作のタイミングに応じて停止表示される図柄組合せを異ならせることとしてもよく、また、押し順をミスした第2 停止操作よりも後の第3 停止操作のタイミングに応じて停止表示される図柄組合せを異ならせることとしてもよい。

【1841】

続いて、押し順ベル当籤時の制御の別例について説明する。本実施形態のパチスロ1では、上述のように1 停ミス時に停止操作のタイミングに応じて停止表示される図柄組合せを異ならせ、2 停ミス時には停止操作のタイミングに関わらず同じ図柄組合せを停止表示させている。この点、図268 (B) に示す別例1のように、1 停ミス時には停止操作のタイミ

10

【1842】

図267と考え方は同じであるため図示は省略するが、この別例1では、押し順正解時には、略称「ベル」に係る図柄組合せが停止表示され、また、1 停ミス時には、略称「RT0 移行図柄」に係る図柄組合せが停止表示され、また、2 停ミス時には停止操作のタイミングに応じて略称「ベル」又は略称「RT2 移行図柄」に係る図柄組合せが停止表示される。

【1843】

ここで、図268 (B) に示すように、この別例1では、2 停ミス時に停止表示される図柄組合せを異ならせるために、押し順ベルとして「F__123ベルC」「F__123ベルD」等を新たに追加して、それぞれの押し順毎に対応する押し順ベルを4つ設ける。そして、この4つの押し順ベル「F__123ベルA」「F__123ベルB」「F__123ベルC」「F__123ベルD」を1つのグループにしている。

20

【1844】

押し順ベル「F__123ベルA」～「F__123ベルD」では、押し順「132」の場合に2 停ミスとなり、押し順「213」「231」「312」「321」の場合に1 停ミスとなるため、上述のように押し順「132」の場合の停止制御を押し順ベル「F__123ベルA」～「F__123ベルD」において異ならせることで、2 停ミス時に4分の1の確率で略称「ベル」に係る図柄組合せが停止表示され、4分の3の確率で略称「RT2 移行図柄」に係る図柄組合せが停止表示されるように制御することができる。

30

【1845】

図268 (B) に示すように、別例1の停止制御によると、主制御回路90は、押し順正解時には、略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8枚のメダルを払い出す。また、主制御回路90は、1 停ミス時は、停止操作のタイミングに関わらず略称「RT0 移行図柄」に係る図柄組合せが停止表示されるように停止制御を行い、結果、RT状態をRT0状態に移行する。また、主制御回路90は、2 停ミス時は、停止操作のタイミングに応じて図柄組合せを異ならせ、4分の1の確率で略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8枚のメダルを払い出し、4分の3の確率で略称「RT2 移行図柄」に係る図柄組合せが停止表示されるように停止制御を行い、結果、RT状態をRT2状態に移行する。

40

【1846】

続いて、押し順ベル当籤時の制御の別例2について説明する。本実施形態のパチスロ1及び別例1のパチスロ1では、1 停ミス時又は2 停ミス時の何れか一方に限り、停止操作のタイミングに応じて停止表示される図柄組合せを異ならせ、もう一方は、停止操作のタイミングに関わらず同じ図柄組合せを停止表示させている。この点、図269 (C) に示す別例2では、1 停ミス時も2 停ミス時も停止操作のタイミングに応じて停止表示される図柄組合せを異ならせる。

【1847】

基本的な考え方は本実施形態のパチスロ1及び別例のパチスロ1と同じであるため詳細な

50

説明は説明する。

図 269 (C) に示すように、別例 2 の停止制御によると、主制御回路 90 は、押し順正解時には、略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8 枚のメダルを払い出す。また、主制御回路 90 は、1 停ミス時は停止操作のタイミングに応じて図柄組合せを異ならせ、4 分の 1 の確率で略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8 枚のメダルを払い出し、4 分の 3 の確率で略称「R T 0 移行図柄」に係る図柄組合せが停止表示されるように停止制御を行い、結果、R T 状態を R T 0 状態に移行する。また、主制御回路 90 は、2 停ミス時も停止操作のタイミングに応じて図柄組合せを異ならせ、4 分の 1 の確率で略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8 枚のメダルを払い出し、4 分の 3 の確率で略称「R T 2 移行図柄」に係る図柄組合せが停止表示されるように停止制御を行い、結果、R T 状態を R T 2 状態に移行する。

【1848】

なお、別例 2 では、1 停ミス時と 2 停ミス時の双方において、4 分の 1 の確率で略称「ベル」に係る図柄組合せを停止表示させることとしているが、1 停ミス時に略称「ベル」に係る図柄組合せを停止表示可能な確率（すなわち、1 停ミス時に略称「ベル」に係る図柄組合せを停止表示するために要求される停止操作のタイミング）と、2 停ミス時に略称「ベル」に係る図柄組合せを停止表示可能な確率（すなわち、2 停ミス時に略称「ベル」に係る図柄組合せを停止表示するために要求される停止操作のタイミング）とを異ならせることとしてもよい。

【1849】

続いて、押し順ベル当籤時の制御の別例 3 について説明する。第 2 停止操作で操作すべきストップボタンの種別を間違えた場合（2 停ミス）や、第 3 停止操作で操作すべきストップボタンの種別を間違えた場合（3 停ミス）、それ以前の停止操作の順序は正しいこともあれば、それ以前の停止操作の順序も間違っていることもある。例えば、正解の押し順が「1 2 3」である場合、第 2 停止操作で操作すべきストップボタンの種別が中のストップボタン 17 C 以外のストップボタンである場合に 2 停ミスとなるため、押し順「1 3 2」「2 1 3」「2 3 1」「3 1 2」の何れも 2 停ミスとなる。このうち押し順「1 3 2」は、第 1 停止操作で操作すべきストップボタンの種別は正しく、また、それ以外の押し順「2 1 3」「2 3 1」「3 1 2」は、第 1 停止操作及び第 2 停止操作のいずれも操作すべきストップボタンの種別が正しくない。

【1850】

別例 3 では、2 停ミス時の第 1 停止操作の種別に応じて停止制御を異ならせる制御について説明する。なお、別例 3 では、2 停ミス時に着目して例示しているが、これに限られるものではなく、所定番目の押し順よりも後の特定番目の押し順不正解時における、所定番目の押し順の正否に応じて停止制御を異ならせることを含むものである。

【1851】

基本的な考え方は本実施形態のパチスロ 1 及び別例のパチスロ 1 と同じであるため詳細な説明は説明する。

図 269 (D) に示すように、別例 3 の停止制御によると、主制御回路 90 は、押し順正解時には、略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8 枚のメダルを払い出す。また、主制御回路 90 は、1 停ミス、かつ、2 停ミス時（すなわち、正解の押し順「1 2 3」に対する押し順「2 1 3」「2 3 1」「3 1 2」）は停止操作のタイミングに応じて図柄組合せを異ならせ、4 分の 1 の確率で略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8 枚のメダルを払い出し、4 分の 3 の確率で略称「R T 0 移行図柄」に係る図柄組合せが停止表示されるように停止制御を行い、結果、R T 状態を R T 0 状態に移行する。また、主制御回路 90 は、1 停正解、かつ、2 停ミス時（すなわち、正解の押し順「1 2 3」に対する押し順「1 3 2」）も停止操作のタイミングに応じて図柄組合せを異ならせ、2 分の 1 の確率で略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、結果、8 枚のメダルを払い出

し、2分の1の確率で略称「RT2移行図柄」に係る図柄組合せが停止表示されるように停止制御を行い、結果、RT状態をRT2状態に移行する。また、主制御回路90は、1停ミス、かつ、2停正解時（すなわち、正解の押し順「123」に対する押し順「321」）は、停止操作のタイミングに関わらず略称「1枚出目」に係る図柄組合せが停止表示されるように停止制御を行い、結果、1枚のメダルを払い出す。

【1852】

なお、別例3では、1停ミス、かつ、2停ミス時と、1停正解、かつ、2停ミス時とで、略称「ベル」に係る図柄組合せを停止表示するために要求される停止操作のタイミングを異ならせているが、これに限られるものではなく、両ケースにおいて略称「ベル」に係る図柄組合せを停止表示するために要求される停止操作のタイミングを同じにしてもよい。また、別例3では、1停正解、かつ、2停ミス時よりも、1停ミス、かつ、2停ミス時の方が、略称「ベル」に係る図柄組合せを停止表示するために要求される停止操作のタイミングが厳しい（2分の1と4分の1）が、これに限られるものではない。主制御回路90は、1停正解、かつ、2停ミス時に4分の1の確率で略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、4分の3の確率で略称「RT0移行図柄」に係る図柄組合せが停止表示されるように停止制御を行い、また、1停ミス、かつ、2停ミス時に2分の1の確率で略称「ベル」に係る図柄組合せが停止表示されるように停止制御を行い、2分の1の確率で略称「RT2移行図柄」に係る図柄組合せが停止表示されるように停止制御を行うこととしてもよい。

【1853】

〔主制御基板及び副制御基板が有する各種機能〕

以上のようなパチスロ1に特有の制御を実現するために、パチスロ1の主制御基板（主制御回路90，メインCPU101）及び副制御基板（副制御回路200，サブCPU201）は、次のような機能を有する。

【1854】

主制御基板71は、スタートスイッチ79やストップスイッチ基板80と接続され、図1に示す遊技の進行を制御する。それゆえ、主制御基板71は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

【1855】

また、主制御基板71は、無限RT状態中にRT移行図柄が停止表示されると、RT状態を移行させるため、RT移行手段として機能する。

【1856】

〔ART状態中のナビ高確〕

また、本実施形態のパチスロ1では、通常ART中にナビ高確ゲーム数が付与されるとナビ高確中の通常ARTになり、「3択昇格リプ」が内部当籤役として決定された場合に、略称「強チャンスリプ（RT5移行図柄）」に係る図柄組合せを停止表示するための報知を行う。その結果、ナビ高確中の通常ARTでは、「3択昇格リプ」の当籤時にRT5状態に移行し、遊技状態が通常ARTからEPに移行する。なお、非ナビ高確中の通常ARTでは、「3択昇格リプ」が内部当籤役として決定された場合に、略称「斜めリプレイ（C__CUリプ）」に係る図柄組合せを停止表示するための報知を行うため、RT5状態に移行することがない。

【1857】

ここで、図270を参照して、ナビ高確ゲーム数の管理方法について説明する。ナビ高ゲーム数は、メインRAM103の所定領域に記憶されており、通常ART中のみ行われる獲得抽籤（図243参照）に当籤すると、当籤したナビ高確ゲーム数が加算される。図270（A）に示すように、ナビ高確ゲーム数は、通常ART中に毎ゲーム1ずつ減算される。図270（A）に示す例では、2ゲーム目の抽籤用フラグ「強チェリー」に基づきナビ高確ゲーム数「5ゲーム」が付与され、続く3～5ゲーム目において、1ずつ減算されている。また、ナビ高確ゲーム数は、上乗せが可能であり、図270（A）に示す例では

、6ゲーム目の抽籤用フラグ「強チェリー」に基づき「5ゲーム」上乘せされている。

【1858】

主制御回路90は、ナビ高確ゲーム数が0である場合、非ナビ高確中として遊技を制御し、ナビ高確ゲーム数が1以上である場合、ナビ高確中として遊技を制御する。そのため、ナビ高確ゲーム数が0である状況でナビ高確ゲーム数が付与されると、主制御回路90は、次遊技をナビ高確中として管理し、また、ナビ高確ゲーム数が0になると、主制御回路90は、次遊技を非ナビ高確中として管理する。

【1859】

続いて、図270(B)に示すように、主制御回路90は、ナビ高確中に遊技状態が通常ARTからEPに移行すると、ナビ高確ゲーム数をクリア(0にする)する。図270(B)に示す例では、4ゲーム目(ナビ高確中)に「3択昇格リブ」が内部当籤役として決定されている。ナビ高確中は、「3択昇格リブ」当籤時に正解の押し順を報知するため、図270(B)に示す例では、4ゲーム目に略称「強チャンスリブ」に係る図柄組合せが表示され、結果、遊技状態が通常ARTからEPに移行している。図270(B)に示す例では、4ゲーム目の時点でナビ高確ゲーム数の残りが「4ゲーム」であったのが、遊技状態がEPに移行したことから、ナビ高確ゲーム数がクリアされている。

【1860】

また、上述のように、パチスロ1では、ナビ高確ゲーム数の付与は通常ART中しか行わないものの、ナビ高確ゲーム数を継続CZやランク決めARTに持ち越すことがある。図270(C)に示す例では、2ゲーム目に付与されたナビ高確ゲーム数が、4ゲーム目の継続CZに持ち越されている。主制御回路90は、ナビ高確ゲーム数を継続CZに持ち越した場合、継続CZ中もナビ高確ゲーム数を毎ゲーム1ずつ減算する。一方で、遊技状態が継続CZからランク決めARTに移行した場合、主制御回路90は、ランク決めARTの終了後(すなわち、ランク決めART中はナビ高確中)にナビ高確ゲーム数をクリア(0にする)する。

【1861】

なお、主制御回路90は、RT5状態中は、ナビ高確ゲーム数を減算しないこととしてもよい。RT5状態中は、ナビ高確中に特有のRT5状態に移行するための押し順報知を行う機会がないため、せっかく付与されたナビ高確ゲーム数を無駄にせずに済む。ここで、RT5状態へは、基本的には「3択昇格リブ」の押し順正解時に移行し、この場合、遊技状態がEPに移行する結果、ナビ高確ゲーム数がクリアされるが、ナビ高確中の通常ART(RT4状態)において「F__強チェリー」「F__7リブA」「F__7リブB」「F__BARリブ」が内部当籤役として決定されたことに基づきRT5状態に移行することもあり、また、ナビ高確ゲーム数を持ち越した継続CZ中にRT5状態に移行し、その後の非ナビ高確中の通常ART(RT5状態)中に、ナビ高確ゲーム数が付与されることもある。このような場合には、ナビ高確中の通常ART(RT5状態)になるため、主制御回路90は、RT5状態中にナビ高確ゲーム数を減算しない。

【1862】

続いて、図271を参照して、ナビ高確ゲーム数を継続CZやランク決めARTに持ち越した場合の各種抽籤について説明する。図271(A)は、継続CZにおいて「3択昇格リブ」が内部当籤役として決定された場合の継続CZ中のART抽籤の概要を示す図である。非ナビ高確中の継続CZでは「3択昇格リブ」に対応する抽籤用フラグが「通常リブ」であるため(図208参照)、継続CZ中のART抽籤に当籤する確率は低い(図228(D)参照)。一方で、ナビ高確中の継続CZでは「3択昇格リブ」に対応する抽籤用フラグが「強チャンスリブ」であるため(図208参照)、継続CZ中のART抽籤に必ず当籤する(図229(E)参照)。なお、ナビ高確中の「3択昇格リブ」当籤時には、正解の押し順が報知されるため、ナビ高確中の継続CZにおいて「3択昇格リブ」が内部当籤役として決定された場合、ART抽籤に必ず当籤するとともに、次遊技(ランク決めART)のRT状態がRT5状態になる。

【1863】

10

20

30

40

50

また、図 271 (B) (C) は、ランク決め A R T における昇格抽籤の概要を示す図である。図 271 (B) に示すように、ランク決め A R T 中も「3 択昇格リブ」が内部当籤役として場合、非ナビ高確中の抽籤用フラグ「通常リブ」に基づく昇格抽籤よりも、ナビ高確中の抽籤用フラグ「強チャンスリブ」に基づく昇格抽籤の方が、ランクが昇格し易い (図 242 参照)。また、図 271 (C) に示すように、内部当籤役が「はずれ」である場合も、非ナビ高確中の昇格抽籤よりも、ナビ高確中の昇格抽籤の方が、ランクが昇格し易い (図 242 参照)。

【1864】

また、継続 C Z にナビ高確ゲーム数を持ち越した場合、ランク決め A R T 中の R T 状態が R T 5 状態になることがある。図 271 (D) は、R T 4 状態のランク決め A R T 中の昇格抽籤と、R T 5 状態のランク決め A R T 中の昇格抽籤とを比較した図である。図 242 に示すランク決め A R T 中ランク昇格抽籤テーブルを参照すると抽籤用フラグ「フェイク 7」「7 揃い」「B A R 揃い」では、A R T 当籤時のランクが昇格し易いことが分かる。また、図 180 の内部抽籤テーブルを参照すると、R T 4 状態と R T 5 状態とでは、この抽籤用フラグ「フェイク 7」「7 揃い」「B A R 揃い」に対応する「F __ フェイクリブ」「F __ 7 リブ A」「F __ 7 リブ B」「F __ B A R リブ」が内部当籤役として決定される確率は、R T 4 状態よりも R T 5 状態の方が高いことが分かる。そのため、R T 5 状態のランク決め A R T 中は、R T 4 状態のランク決め A R T よりも A R T 当籤時のランクが昇格し易くなる。

【1865】

図 271 (A) ~ (D) に例示するように、ナビ高確ゲーム数を継続 C Z やランク決め A R T に持ち越した場合、継続 C Z 中の A R T 抽籤、及びランク決め A R T 中の昇格抽籤のいずれも遊技者にとって好ましい結果になり易い。特に、「3 択昇格リブ」という 1 つの役に基づく A R T 抽籤及び昇格抽籤の抽籤結果がナビ高確中であるか否かに応じて異なることから、遊技性が多様になる。また、ナビ高確中の通常 A R T において「3 択昇格リブ」が内部当籤役として決定されなかった場合であっても、ナビ高確ゲーム数が残っている場合には、その後の継続 C Z やランク決め A R T が有利になることから、遊技者が不満を感じることがない。

【1866】

【主制御基板及び副制御基板が有する各種機能】

以上のような本実施形態のパチスロ 1 に特有の制御を実現するために、本実施形態のパチスロ 1 の主制御基板 (主制御回路 90, メイン C P U 101) 及び副制御基板 (副制御回路 200, サブ C P U 201) は、次のような機能を有する。

【1867】

主制御基板 71 は、スタートスイッチ 79 やストップスイッチ基板 80 と接続され、図 1 に示す遊技の進行を制御する。それゆえ、主制御基板 71 は、開始操作検出手段、図柄変動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段 (停止制御手段) 及び入賞判定手段として機能する。

【1868】

また、主制御基板 71 は、A R T 抽籤を行い A R T 状態に移行するか否かを決定し、A R T 抽籤に当籤した場合に遊技状態を A R T 状態に移行するため、移行決定手段及び有利状態制御手段として機能する。また、主制御基板 71 は、C Z 中の遊技を管理して C Z の残りゲーム数が 0 になると C Z を終了することから、高確率状態制御手段として機能する。

【1869】

また、主制御基板 71 は、ナビ高確中の A R T 状態中に「3 択昇格リブ」が内部当籤役として決定され、かつ、ナビ高確中の報知に従い略称「強チャンスリブ」に係る図柄組合せが表示されると、遊技状態を通常 A R T から E P に移行させるため、特典付与手段として機能する。

【1870】

また、主制御基板 71 は、ナビ高確ゲーム数に基づいてナビ高確又は非ナビ高確を制御し

、また、ナビ高確中に遊技状態がE Pに移行するとナビ高確から非ナビ高確に切り替えるため、報知モード制御手段として機能する。

【1871】

〔継続C Zの昇格制御〕

また、本実施形態のパチスロ1では、C Z中のART抽籤に当籤し、C ZからART状態に移行する際に（より詳細には、ランク決めART中に）、継続C Zとして特殊C Zを用いるか否か抽籤し、この抽籤に当籤した場合には、その後の継続C Zとして特殊C Zを用いる（図235参照）。これにより、C ZとART状態とのループ中にC Zの種別が切り替わることがあるため、C Z中の遊技が単調になることを防止できる。

【1872】

また、本実施形態のパチスロ1では、特殊C Zへの移行に当籤したときのART状態のセット数のストックがない場合、ART状態のセット数が1つ付与される。これにより、特殊C Zに移行した場合には、必ずART状態に移行することになるため、安心して遊技を行うことができる。なお、特殊C Z移行時に付与するART状態のストック数は任意であり、主制御回路90は、2つ以上のストックを付与することとしてもよい。また、特殊C Zへの移行時に固定数のストックを付与するのではなく、ART状態のセット数のストックが所定数に達するまでに必要な数のストックを付与することとしてもよい。例えば、特殊C Zへの移行時にART状態のセット数のストックが3つになるまでART状態のセット数のストックを付与する場合、特殊C Zへの移行時に保有しているストック数が0である場合には3つのストックを付与し、特殊C Zへの移行時に保有しているストック数が1である場合には2つのストックを付与し、特殊C Zへの移行時に保有しているストック数が2である場合には1つのストックを付与し、特殊C Zへの移行時に保有しているストック数が3以上である場合にはストックを付与しないこととしてもよい。

【1873】

また、継続C Zの遊技期間は、C Zゲーム数により管理されるが、このC Zゲーム数として、特殊C Z用のC Zゲーム数とC Z（ART後）用のC Zゲーム数とを別々に設けることとしてもよく、また、共通のC Zゲーム数を用いることとしてもよい。別々のC Zゲーム数を用いる場合、特殊C Zに移行すると、主制御回路90は、C Z（ART後）用のC Zゲーム数を、特殊C Z用のC Zゲーム数としてセットする。このとき、主制御回路90は、C Z（ART後）用のC Zゲーム数を維持したままとしてもよく、また、クリア（0にする）することとしてもよい。

【1874】

また、別々のC Zゲーム数を用いる場合、主制御回路90は、C Z（ART後）からART状態に移行すると、C Z（ART後）用のC Zゲーム数を上乗せし、また、特殊C Z中にART抽籤に当籤したことに応じてART状態に移行すると、特殊C Z用のC Zゲーム数を上乗せする。一方で、特殊C Z中に放出フェーズDに基づきART状態に移行した場合には、C Z（ART後）用のC Zゲーム数を上乗せする。

【1875】

一方で、共通のC Zゲーム数を用いる場合には、主制御回路90は、C Zの種別を考慮することなく、ART状態への移行時にC Zゲーム数の上乗せを行えばよい。なお、別々のC Zゲーム数を用いる場合であっても、共通のC Zゲーム数を用いる場合であっても、C Zゲーム数は、メインRAM103において管理される。

【1876】

〔主制御基板及び副制御基板が有する各種機能〕

以上のような本実施形態のパチスロ1に特有の制御を実現するために、本実施形態のパチスロ1の主制御基板（主制御回路90，メインCPU101）及び副制御基板（副制御回路200，サブCPU201）は、次のような機能を有する。

【1877】

主制御基板71は、スタートスイッチ79やストップスイッチ基板80と接続され、図1に示す遊技の進行を制御する。それゆえ、主制御基板71は、開始操作検出手段、図柄変

10

20

30

40

50

動手段、内部当籤役決定手段、停止操作検出手段、リール停止制御手段（停止制御手段）及び入賞判定手段として機能する。

【1878】

また、主制御基板71は、ART抽籤を行いART状態に移行するか否かを決定し、ART抽籤に当籤した場合に遊技状態をART状態に移行するため、付与決定手段及び特典付与手段として機能する。

【1879】

また、主制御基板71は、ART抽籤に当籤すると、付与するART状態の権利数を決定して、メインRAM103に記憶するとともに、遊技状態をART状態に移行させると、メインRAM103に記憶されているART状態の権利を1つ消し去るため、権利管理手段として機能する。

10

【1880】

また、主制御基板71は、CZ中の遊技を管理してCZの残りゲーム数が0になるとCZを終了することから、第1高確率状態制御手段及び第2高確率状態制御手段として機能する。

【1881】

また、主制御基板71は、CZ（ART後）と特殊CZとで共通のCZゲーム数を用いる場合には、期間管理手段として機能し、別々のCZゲーム数を用いる場合には、特殊CZの移行時にCZ（ART後）用のCZゲーム数を特殊CZ用のCZゲーム数としてセットすることから、期間設定手段として機能する。

20

また、主制御基板71は、CZ中にART状態に移行すると、CZゲーム数の上乗せを行うため、高確率状態加算手段として機能する。

【1882】

<乱数値格納領域の構成>

次に、図272及び図273を参照して、各種抽籤処理に用いられる各種乱数値の格納領域の構成を説明する。図272は、本実施形態におけるメインRAM103のメモリマップを示す図であり、図273は、メインRAM103内に設けられた乱数格納領域の構成を説明するためにクロスリファレンスファイルから乱数格納領域が記載された部分を抜粋した図である。なお、クロスリファレンスファイルとは、プログラムソースファイルをメインROM102に格納するための変換（アセンブル等と呼ばれる）を行った際に出力される、アドレスやデータが16進数で付加されたファイルのことである。

30

【1883】

本実施形態のメインRAM103（書き換え可能な不揮発性格納領域）のメモリマップでは、図272に示すように、上記第1実施形態と同様に、メインRAM103の先頭アドレス（F000H）側から、遊技用RAM領域（所定格納領域、遊技用一時記憶領域）及び規定外RAM領域（規定外一時記憶領域）が、この順で、それぞれ所定のアドレスに配置される。そして、乱数値格納領域は、メインRAM103内において、「設定変更開始時」の初期化開始アドレスと、「ボーナス終了時」の初期化開始アドレスとの間の領域に配置される。

【1884】

40

具体的には、図272及び図273に示すように、乱数値格納領域は、アドレス「F009H」（ラベル「wRANDOMX」が付されたアドレス）～「F016H」（ラベル「wRANDOMS」が付されたアドレス）の範囲（所定のアドレス範囲）に配置される。なお、アドレス「F009H」（ラベル「wRANDOMX」が付されたアドレス）の乱数値X格納領域には、内部当籤役抽籤用乱数（ハードラッチ乱数：第1の乱数値）が格納され、アドレス「F00BH」（ラベル「wRANDOM1BS」及び「wRANDOMY」が付されたアドレス）の乱数値Y格納領域～「F016H」（ラベル「wRANDOMS」が付されたアドレス）の乱数値S格納領域にはそれぞれ、ART等の出玉（演出）関連抽籤用の12種類の乱数値（ソフトラッチ乱数：第2の乱数値）が格納される。

【1885】

50

＜ソースプログラム上での各種抽籤テーブルの構成＞

本実施形態のパチスロ1のメインROM102のデータエリア（図12B参照）に格納された各種抽籤テーブルでは、テーブルやプログラムのデータ容量を減らすための様々な工夫が施されている。以下では、そのような工夫が施された抽籤テーブルの構成例を説明する。

【1886】

〔状態移行抽籤テーブル（BB終了時用）〕

図274は、ソースプログラム上で実際に参照される、図218（E）の状態移行抽籤テーブル（BB終了時用）の構成例を示す図である。なお、図215、図217、図218（A）～218（D）、図218（F）、図218（G）、図219、図224、図234、図236～図240、図242～図248の抽籤テーブルの構成も図274に示す構成と同様となる。

【1887】

ソースプログラム上で実際に参照される状態移行抽籤テーブル（BB終了時用）は、図274に示すように、ラベル「dBDDMC GTB」が付された先頭アドレスから9バイト先のアドレスまでの格納領域（10バイトの格納領域）に配置される。この状態移行抽籤テーブル（BB終了時用）内において、先頭アドレスから9バイト先のアドレスまでの各アドレスには、データ「.LOW. (wRAND1BS + cRNDAT04) + 20H * (3)」、「.LOW. wDDM_STS」、「(80H) + dBDDMC_L-\$」、「(80H) + dBDDMC_H-\$」、「00000101B」、「124」、「3」、「00000101B」、「224」及び「cHITABS」がそれぞれ格納される。なお、各データ（「DB」はデータバイトを表す疑似命令）は1バイト（8ビット）のデータで構成される。以下、各データの内容を説明する。

【1888】

（1）複合データの構成

状態移行抽籤テーブル（BB終了時用）の先頭アドレスに格納されているデータ「.LOW. (wRAND1BS + cRNDAT04) + 20H * (3)」は、状態移行抽籤の抽籤回数（この例では3回）と、状態移行抽籤で使用する出玉抽籤用乱数値が格納された乱数格納領域のアドレスを指定するためのオフセット値（乱数格納領域の先頭アドレスからのオフセット値）とが格納された1バイトの複合データ（複合情報）である。

【1889】

ここで、図275に、複合データ「.LOW. (wRAND1BS + cRNDAT04) + 20H * (3)」のビット構成を示す。複合データ「.LOW. (wRAND1BS + cRNDAT04) + 20H * (3)」では、抽籤回数（3回）に「20H」が乗算されているので、抽籤回数の情報は、複合データのビット5～ビット7に格納され、2バイトラベル「wRAND1BS + cRNDAT04」の下位側1バイトの値（乱数格納領域の先頭アドレスからのオフセット値）が複合データのビット0～ビット4に格納される。

【1890】

なお、この例の2バイトラベル「wRAND1BS + cRNDAT04」のうち、ラベル「wRAND1BS」は、図273に示すように、出玉関連抽籤用の乱数値格納領域の先頭アドレス「F00BH」を示す。また、ラベル「cRNDAT04」は、乱数指定用オフセット番号を示す。

【1891】

ここで、図276に、ソースプログラム上で実際に参照される乱数指定用オフセット番号の値とラベルとの対応テーブルを示す。例えば、ラベル「cRNDAT04」の乱数指定用オフセット番号の値は「0000H」となる。それゆえ、2バイトラベル「wRAND1BS + cRNDAT04」の値は「F00BH (=「F00BH」 + 「0000H」)」となり、複合データのビット0～ビット4には、この2バイトラベルの下位側1バイトの値「0BH」が格納される。したがって、メインROM102には、複合データ「.LOW. (wRAND1BS + cRNDAT04) + 20H * (3)」として、1バイトの値

10

20

30

40

50

「01101011B」=「6BH」が格納される。

【1892】

なお、抽籤処理において、上述した複合データに含まれる抽籤回数及びオフセット値を使用する際には、複合データから抽籤回数及びオフセット値をそれぞれ別個に分離する処理が施され、これにより各データが複合データから抽出される。この分離（抽出）処理は、図示しないが、ソースプログラム上においてメインCPU101により実行される。

【1893】

（2）出玉ステータスデータの構成

状態移行抽籤テーブル（BB終了時用）の先頭アドレス（複合データの格納アドレス）から1バイト先のアドレスに格納されているデータ「.LOW.wDDM_STS」は、現在の通常状態中の抽籤状態に関するデータ（出玉ステータスデータ）である。具体的には、データ「.LOW.wDDM_STS」は、現在の抽籤状態に対応する状態移行抽籤テーブルを指定するためのデータである。なお、現在の抽籤状態が「低確」である場合には、「wDDM_STS」は「0」となり、現在の抽籤状態が「高確」である場合には、「wDDM_STS」は「1」となり、現在の抽籤状態が「超高確」である場合には、「wDDM_STS」は「2」なる。

【1894】

例えば、現在の抽籤状態が「低確」である場合には、データ「.LOW.wDDM_STS」として、1バイト先（先頭アドレスから2バイト先）のアドレスに格納されたデータ「(80H)+dBDMC_L-\$」を指定するためのデータが格納される。また、現在の抽籤状態が「高確」である場合には、データ「.LOW.wDDM_STS」として、2バイト先（先頭アドレスから3バイト先）のアドレスに格納されているデータ「(80H)+dBDMC_H-\$」を指定するためのデータが格納される。

【1895】

（3）低確時用抽籤値格納領域の指定データの構成

状態移行抽籤テーブル（BB終了時用）の先頭アドレスから2バイト先のアドレスに格納されているデータ「(80H)+dBDMC_L-\$」は、現在の抽籤状態が低確時である場合に参照される抽籤値（抽籤係数）群が格納されている領域（以下、「低確時用抽籤値格納領域」という）の先頭アドレス、及び、抽籤指定有りを指定するためのデータである。

【1896】

データ「(80H)+dBDMC_L-\$」では、「(80H)」により、ビット7に「1」を格納され、これにより抽籤指定有りが示され、このデータは、上述した複合データに含まれる抽籤回数分（この例では、3回）の抽籤が実行されることを示す指定データとなる。本実施形態では、複合データに含まれる抽籤回数が1以上であれば、抽籤指定が有りとなる。そして、抽籤指定有りがセットされている場合には、抽籤回数分の抽籤処理の中で実際に抽籤値データを取得して抽籤を行うタイミングが指定される。この実際に抽籤が行われるタイミングは、後述の1バイトの抽籤指定データにより指定される。具体的には、抽籤処理毎に抽籤指定データのビット0からビット7に向かって各ビットの値が順次検索され、検索対象のビットに「1」が格納されていれば、抽籤値データを取得して抽籤を実行する。なお、抽籤指定無しの場合には、低確時用抽籤値格納領域の指定データ内において、「(80H)」が規定されない（ビット7には「0」が格納される）。

【1897】

また、データ「(80H)+dBDMC_L-\$」内の「dBDMC_L-\$」には、低確時用抽籤値格納領域の先頭アドレス（ラベル「dBDMC_L」が付されたアドレス）を示すデータが格納される。なお、本実施形態では、データ「(80H)+dBDMC_L-\$」が格納されているアドレスから1バイト先のアドレス（ラベル「dBDMC_L」が付されたアドレス）から3バイト分の領域に低確時用抽籤値格納領域が配置される。

【1898】

（4）高確時用抽籤値格納領域の指定データの構成

状態移行抽籤テーブル（ＢＢ終了時用）の先頭アドレスから３バイト先のアドレスに格納されているデータ「（８０Ｈ）＋ｄＢＤＭＣ＿Ｈ－\$」は、現在の抽籤状態が高確時である場合に参照される抽籤値（抽籤係数）群が格納されている領域（以下、「高確時用抽籤値格納領域」という）の先頭アドレス、及び、抽籤指定有りを指定するためのデータである。

【１８９９】

データ「（８０Ｈ）＋ｄＢＤＭＣ＿Ｈ－\$」においても、「（８０Ｈ）」により、ビット７に「１」を格納され、これにより抽籤指定有りが示される。また、データ「（８０Ｈ）＋ｄＢＤＭＣ＿Ｈ－\$」内の「ｄＢＤＭＣ＿Ｈ－\$」には、高確時用抽籤値格納領域の先頭アドレス（ラベル「ｄＢＤＭＣ＿Ｈ」が付されたアドレス）を示すデータが格納される。なお、本実施形態では、データ「（８０Ｈ）＋ｄＢＤＭＣ＿Ｈ－\$」が格納されているアドレスから４バイト先のアドレス（ラベル「ｄＢＤＭＣ＿Ｈ」が付されたアドレス）から３バイト分の領域に高確時用抽籤値格納領域が配置される。

【１９００】

なお、状態移行抽籤テーブル（ＢＢ終了時用）内に格納されている複合データ「.LOW.（wRAND1BS+cRNDAT04）+20H*（３）」、データ「.LOW.wDDM_STS」、データ「（８０Ｈ）＋ｄＢＤＭＣ＿Ｌ－\$」及びデータ「（８０Ｈ）＋ｄＢＤＭＣ＿Ｈ－\$」は、本発明に係る抽籤選択情報の一具体例を示すものである。また、これらのデータが格納されている領域は、本発明に係る抽籤選択格納領域の一具体例を示すものである。また、複合データに含まれる抽籤回数及びオフセット値はそれぞれ、本発明に係る抽籤回数情報及び乱数選択情報の一具体例を示すものである。さらに、これらのデータに基づく抽籤処理は、主制御基板７１（メインＣＰＵ１０１）により実行されるので、主制御基板７１（メインＣＰＵ１０１）は、抽籤手段及び出玉抽籤手段としても機能する。

【１９０１】

（５）低確時用抽籤値格納領域の構成

状態移行抽籤テーブル（ＢＢ終了時用）内において、ラベル「ｄＢＤＭＣ＿Ｌ」が付された低確時用抽籤値格納領域の先頭アドレスには、抽籤指定データ「０００００１０１Ｂ」が格納される。この例では、抽籤回数が３回であるので、抽籤処理毎に抽籤指定データのビット０からビット２まで各ビットの値が順次検索され、検索対象のビットに「１」が格納されていれば、抽籤値データを用いた抽籤を実行し、検索対象のビットに「０」が格納されていれば、抽籤を行わない。それゆえ、この例の低確時用の状態移行抽籤テーブル（ＢＢ終了時用）を用いた状態移行抽籤では、１回目及び３回目の抽籤処理において抽籤値データを用いた抽籤が実行される。

【１９０２】

ここで、図２７７に、ＢＢ終了時の状態移行抽籤で参照される抽籤指定データの各ビットと、該ビットに対応する抽籤の内容との関係を示す。この例では、抽籤指定データのビット０が抽籤結果「高確Ｂ」が決定されるか否かの抽籤の有無を表し、ビット１が抽籤結果「高確Ａ」が決定されるか否かの抽籤の有無を表し、ビット２が抽籤結果「超高確」が決定されるか否かの抽籤の有無を表す。この例の低確時用の状態移行抽籤テーブル（ＢＢ終了時用）では、抽籤結果「高確Ｂ」及び「超高確」に対して１以上の抽籤値が規定されているので（図２１８（Ｅ）参照）、抽籤指定データのビット０及びビット２に「１」がセットされる。一方、この例の低確時用の状態移行抽籤テーブル（ＢＢ終了時用）では、抽籤結果「高確Ａ」に対して抽籤値「０」が規定されている（非当籤が確定している）ので（図２１８（Ｅ）参照）、抽籤指定データのビット１に「０」がセットされ、抽籤結果「高確Ａ」が決定されるか否かの抽籤は行われない。

【１９０３】

低確時用抽籤値格納領域の先頭アドレス（ラベル「ｄＢＤＭＣ＿Ｌ」が付されたアドレス）から１バイト先のアドレスに格納されるデータ「１２４」は、１回目の状態移行抽籤（抽籤指定データのビット０に対応する抽籤：抽籤結果「高確Ｂ」が決定されるか否かの抽

10

20

30

40

50

籤)で使用される抽籤値データである。また、低確時用抽籤値格納領域の先頭アドレスから2バイト先のアドレスに格納されるデータ「3」は、3回目の状態移行抽籤(抽籤指定データのビット2に対応する抽籤:抽籤結果「超高確」が決定されるか否かの抽籤)で使用する抽籤値データである。

【1904】

なお、図218(E)に示す低確時用の状態移行抽籤テーブル(BB終了時用)では、抽籤結果「低確」に対して抽籤値「128」が設定されているが、この抽籤値と、抽籤結果「高確B」及び「超高確」の抽籤値とを加算すると「256」となる。それゆえ、BB終了時であり且つ現在の抽籤状態が低確時である場合の状態移行抽籤において、抽籤結果「高確B」及び「超高確」に対するいずれの抽籤においても非当籤が決定された時には、抽籤をしなくても自動的に、抽籤結果「低確」が決定される。したがって、この例では、抽籤指定データ内で抽籤結果「低確」が決定されるか否かの抽籤の有無を表すビットを設けず、抽籤結果「低確」に対する状態移行抽籤は行われない。また、低確時用抽籤値格納領域には、抽籤結果「低確」の抽籤値も格納されない。

【1905】

(6) 高確時用抽籤値格納領域の構成

状態移行抽籤テーブル(BB終了時用)内において、ラベル「dBDMC_H」が付された高確時用抽籤値格納領域の先頭アドレスには、抽籤指定データ「00000101B」が格納される。それゆえ、この例の高確時用の状態移行抽籤テーブル(BB終了時用)を用いた状態移行抽籤では、1回目及び3回目の抽籤処理において抽籤値データを用いた抽籤が実行される。

【1906】

なお、図218(E)に示す高確時用の状態移行抽籤テーブル(BB終了時用)では、抽籤結果「高確B」及び「超高確」に対して1以上の抽籤値が規定されているので、抽籤指定データのビット0及びビット2に「1」がセットされる。一方、この例の高確時用の状態移行抽籤テーブル(BB終了時用)では、抽籤結果「高確A」に対して抽籤値「0」が規定されている(非当籤が確定している)ので(図218(E)参照)、抽籤指定データのビット1に「0」がセットされ、抽籤結果「高確A」が決定されるか否かの抽籤は行われない。

【1907】

また、図218(E)に示す高確時用の状態移行抽籤テーブル(BB終了時用)では、抽籤結果「低確」に対して抽籤値「0」が規定されているので、抽籤結果「低確」に対する状態移行抽籤を行わなくても、非当籤が確定している。それゆえ、BB終了時であり且つ現在の抽籤状態が高確時である場合の状態移行抽籤では、ソースプログラム上において、抽籤結果「低確」に対する状態移行抽籤は行われず、高確時用抽籤値格納領域にも抽籤結果「低確」の抽籤値は格納されない。

【1908】

高確時用抽籤値格納領域の先頭アドレス(ラベル「dBDMC_H」が付されたアドレス)から1バイト先のアドレスに格納されるデータ「124」は、1回目の状態移行抽籤(抽籤指定データのビット0に対応する抽籤:抽籤結果「高確B」が決定されるか否かの抽籤)で使用する抽籤値データである。また、高確時用抽籤値格納領域の先頭アドレスから2バイト先のアドレスに格納されるデータ「cHITABS」は、3回目の状態移行抽籤(抽籤指定データのビット2に対応する抽籤:抽籤結果「超高確」が決定されるか否かの抽籤)で使用する抽籤値データであり、抽籤結果「超高確」が無条件で当籤確定となることを示すデータである。

【1909】

なお、図218(E)に示す高確時用の状態移行抽籤テーブル(BB終了時用)では、抽籤結果「高確B」及び「超高確」に対してのみ抽籤値が割り振られており、両抽籤値を加算すると「256」となる。それゆえ、高確時用の状態移行抽籤テーブルを用いた状態移行抽籤(1回目の抽籤)において、抽籤結果「高確B」が決定されなかった場合には、抽

10

20

30

40

50

籤をしなくても自動的に、抽籤結果「超高確」が決定される。それゆえ、この例では、3回目の状態移行抽籤（抽籤結果「超高確」に対する抽籤）で使用される抽籤値データの格納領域には、無条件で当籤が確定するデータ「c H I T A B S」が格納され、3回目の状態移行抽籤は実際には行われない。

【1910】

また、本実施形態では、図218（E）の状態移行抽籤テーブル（BB終了時用）に示すように、現在の抽籤状態が「超高確」である場合には、状態移行抽籤の結果として必ず「超高確」が決定され、抽籤状態が「超高確」に維持される（抽籤状態が移行しない）。それゆえ、現在の抽籤状態が「超高確」である場合には、ソースプログラム上では、状態移行抽籤は行われず、状態移行抽籤テーブル（BB終了時用）内にも、超高確時用の抽籤値群の格納領域及びそれを指定するための各種データは格納されない。

10

【1911】

なお、状態移行抽籤テーブル（BB終了時用）内で規定されている低確時用抽籤値格納領域及び高確時用抽籤値格納領域は、本発明に係る抽籤値格納領域の一具体例を示すものである。

【1912】

上述した状態移行抽籤テーブル（BB終了時用）の構成では、次のような効果が得られる。ソースプログラム上で実際に参照される状態移行抽籤テーブル（BB終了時用）において、図218（E）に示す構成の抽籤テーブルを忠実に構成しようとする、現在の抽籤状態毎に5つの抽籤値をテーブル内に格納しなければならない。それゆえ、この場合には、15バイトの格納領域が必要となる。それに対して、図274に示すような態様、すなわち、抽籤結果が確定している抽籤対象の抽籤値（抽籤係数）を格納しない態様で、状態移行抽籤テーブル（BB終了時用）を構成した場合には、上述のように、10バイトの格納領域で状態移行抽籤テーブル（BB終了時用）を構成することができる。それゆえ、図274に示す構成の抽籤テーブルを用いることにより、メインROM102に格納されるテーブルデータの容量を低減することができる。

20

【1913】

〔BB中ART抽籤テーブル（通常BB又はART中BB中用）〕

図278は、ソースプログラム上で実際に参照される、図249（A）のBB中ART抽籤テーブル（通常BB又はART中BB中用）の構成例を示す図である。なお、図220、図223、図225、図228～図233、図235、図241、図249（B）、図249（C）の抽籤テーブルの構成も図278に示す構成と同様となる。

30

【1914】

ソースプログラム上で実際に参照されるBB中ART抽籤テーブル（通常BB又はART中BB中用）は、図278に示すように、ラベル「d B B S V O T B」が付された先頭アドレスから4バイト先のアドレスまでの格納領域（5バイトの格納領域）に配置される。このBB中ART抽籤テーブル（通常BB又はART中BB中用）内において、先頭アドレスから4バイト先のアドレスまでの各アドレスには、データ「.LOW. (w R A N D 1 B S + c R N D A T 1 5) + 2 0 H * (0)」、「.LOW. w D D M F L G _ C」、「c P R B _ 0」、「c P R B _ 4」及び「c P R B _ H I T」がそれぞれ格納される。なお、各データ（DB）は1バイト（8ビット）のデータで構成される。以下、各データの内容を説明する。

40

【1915】

（1）複合データの構成

BB中ART抽籤テーブル（通常BB又はART中BB中用）の先頭アドレスに格納されているデータ「.LOW. (w R A N D 1 B S + c R N D A T 1 5) + 2 0 H * (0)」は、通常BB又はART中BB中時のART抽籤の抽籤回数と、ART抽籤で使用する出玉抽籤用乱数値が格納された乱数格納領域のアドレスを指定するためのオフセット値とが格納された1バイトの複合データである。

【1916】

50

この例では、抽籤回数として「0」が複合データのビット5～ビット7に格納され、2バイトラベル「w R A N D 1 B S + c R N D A T 1 5」の下位側1バイトの値（オフセット値「1 0 H」（「F 0 1 0 H」の下位側1バイトの値））がビット0～ビット4に格納される。なお、ここでいう、「抽籤回数が0回」とは、抽籤を行わないという意味ではなく、抽籤指定を行わない（上述した抽籤指定データを用いた抽籤を行わない）という意味であり、この場合には、指定されたアドレスに格納された抽籤値データを用いて1回抽籤が行われる。

【1917】

（2）フラググループ種別の指定データの構成

BB中ART抽籤テーブル（通常BB又はART中BB中用）の先頭アドレスから1バイト先のアドレスに格納されているデータ「.LOW.wDDMF LG_C」は、現ゲームで取得された抽籤用フラグの種別に対応するフラググループ（フラググループC）の種別に関するデータ（フラググループCの指定データ）である。

【1918】

なお、本実施形態では、図249（A）に示すように、BB中ART抽籤テーブル（通常BB又はART中BB中用）で用いられる抽籤用フラグの種別は、「CBBベル」、「中段ベル」、「斜めベル」、「ベルハズレA」、「ベルハズレB」、「フェイク7」、「7揃い」及び「BAR揃い」の8種類が設けられているが、ソースプログラム上では、これらの抽籤用フラグが3種類のフラググループCに振り分けられる。そして、BB中ART抽籤テーブル（通常BB又はART中BB中用）を使用したART抽籤では、ソースプログラム上において、8種類の抽籤用フラグの代わりに3種類のフラググループCが抽籤パラメータとして用いられる。

【1919】

具体的には、当籤の抽籤値が「0」である抽籤用フラグ「CBBベル」、「中段ベル」、「斜めベル」及び「フェイク7」が「ハズレ」と称するフラググループCでまとめられ、当籤の抽籤値が「4」である抽籤用フラグ「ベルハズレA」及び「ベルハズレB」が「EP役」と称するフラググループCでまとめられ、当籤の抽籤値が「256」である抽籤用フラグ「7揃い」及び「BAR揃い」が「プレミア系」と称するフラググループCでまとめられる。

【1920】

そして、データ「.LOW.wDDMF LG_C」には、フラググループCの種別に対応する抽籤値データが格納された領域を指定するためのデータが格納される。具体的には、フラググループCが「ハズレ」である場合には、データ「.LOW.wDDMF LG_C」として、1バイト先に格納されているデータ「cPRB_0」（非当籤確定を指定するラベルデータ）を指定するためのデータが規定される。また、フラググループCが「EP役」である場合には、データ「.LOW.wDDMF LG_C」として、2バイト先に格納されているデータ「cPRB_4」（抽籤値「4」の格納領域を指定するラベルデータ）を指定するためのデータが規定される。さらに、フラググループCが「プレミア系」である場合には、データ「.LOW.wDDMF LG_C」として、3バイト先に格納されているデータ「cPRB_HIT」（当籤確定を指定するラベルデータ）を指定するためのデータが規定される。

【1921】

なお、フラググループCが「ハズレ」である場合には、「wDDMF LG_C」は「0」となり、フラググループCが「EP役」である場合には、「wDDMF LG_C」は「1」となり、フラググループCが「プレミア系」である場合には、「wDDMF LG_C」は「2」となる。

【1922】

（3）抽籤値のラベルデータの構成

BB中ART抽籤テーブル（通常BB又はART中BB中用）の先頭アドレスから2バイト先のアドレスに格納されているデータ「cPRB_0」は、フラググループCが「ハズ

10

20

30

40

50

レ」である場合に参照される非当籤確定データ（後述の当否要求有りで且つ非当籤決定を示すデータ：抽籤値「0」とは異なる値が割り当てられる）が格納されているアドレスを指定するためのラベルデータである。BB中ART抽籤テーブル（通常BB又はART中BB中用）の先頭アドレスから3バイト先のアドレスに格納されているデータ「cPRB__4」は、フラググループCが「EP役」である場合に用いられる抽籤値「4」が格納されているアドレスを指定するためのラベルデータである。また、BB中ART抽籤テーブル（通常BB又はART中BB中用）の先頭アドレスから4バイト先のアドレスに格納されているデータ「cPRB__HIT」は、フラググループCが「プレミア系」である場合に参照される当籤確定データ（後述の当否要求有りで且つ当籤決定を示すデータ：抽籤値「256」とは異なる値が割り当てられる）が格納されているアドレスを指定するためのラベルデータである。

10

【1923】

ここで、図279に、ソースプログラム上で参照される、上述したデータ「cPRB__0」等のラベルデータと、該ラベルデータに対応付けられた抽籤値データとの対応を規定した当否判定用係数テーブルの構成を示す。なお、本実施形態において、当否判定用係数テーブルは、BB中ART抽籤だけでなく、当籤／非当籤を決定する他の抽籤においても共通で用いられる。

【1924】

当否判定用係数テーブルで規定されている抽籤値データの種別を大きく分類すると、当否要求有りの抽籤値データと当否要求無しの抽籤値データとに分けられる。当否要求有りの抽籤値データは、抽籤の当籤又は非当籤が、抽籤を行わなくても確定しているデータのことである。当否要求無しの抽籤値データは、抽籤処理を行ってはじめて当籤／非当籤が確定するデータである。

20

【1925】

当否要求の有無の情報は、抽籤値データを構成する1バイト（8ビット）データ内のビット0に格納され、当否要求有りの場合にはビット0に「1」が格納され、当否要求無しの場合には、ビット0に「0」が格納される。また、当否要求有りの抽籤値データにおいて当籤確定／非当籤確定の情報は、抽籤値データ（1バイト）のビット1に格納され、当籤確定の場合にはビット1に「1」が格納され、非当籤確定の場合には、ビット1に「0」が格納される。なお、実際の処理では、抽籤値データのビット0の値を検出し、ビット0に「1」が格納されている場合（当否要求有りの場合）には、抽籤処理が行われず、ビット0に「0」が格納されている場合（当否要求無しの場合）には、抽籤処理が行われる。

30

【1926】

図279の当否判定用係数テーブル中において、ラベル「cPRB__HIT」で指定される値（ $1 \times 2 + (1)$ ）及びラベル「cPRB__0」で指定される値（ $0 \times 2 + (1)$ ）が当否要求有りの抽籤値データである。なお、前者（当籤確定）の抽籤値データには「256」とは異なる値（「3（00000011B）」）が割り当てられ、後者（非当籤確定）の抽籤値データには「0」とは異なる値（「1（00000001B）」）が割り当てられる。

【1927】

一方、図279の当否判定用係数テーブル中において、ラベル「cPRB__1」～「cPRB__192」で指定されるアドレスに格納された抽籤値データが当否要求無しの抽籤値データである。なお、本実施形態では、ラベル「cPRB__###」で指定されるアドレスには、抽籤値「###」が格納される。

40

【1928】

それゆえ、図278に示すBB中ART抽籤テーブル（通常BB又はART中BB中用）を用いたART抽籤において、抽籤値データとして、BB中ART抽籤テーブルの先頭アドレスから2バイト先のアドレスに格納されているデータ「cPRB__0」が選択された場合には、抽籤は行われず、非当籤が確定する。図278に示すBB中ART抽籤テーブル（通常BB又はART中BB中用）を用いたART抽籤において、抽籤値データとして

50

、BB中ART抽籤テーブルの先頭アドレスから3バイト先のアドレスに格納されているデータ「cPRB_4」が選択された場合には、抽籤値「4」が取得され、該抽籤値と出玉抽籤用乱数値とに基づいてART抽籤が行われる。また、図278に示すBB中ART抽籤テーブル（通常BB又はART中BB中用）を用いたART抽籤において、抽籤値データとして、BB中ART抽籤テーブルの先頭アドレスから4バイト先のアドレスに格納されているデータ「cPRB_HIT」が選択された場合には、抽籤は行われず、当籤が確定する。

【1929】

〔BB当籤時状態別CZ抽籤テーブル（A）～（C）〕

図280は、ソースプログラム上で実際に参照される、図222（A）～222（C）のBB当籤時状態別CZ抽籤テーブルの構成例を示す図である。なお、ソースプログラム上では、図222（A）のBB当籤時状態別CZ抽籤テーブル（低確時用）、図222（B）のBB当籤時状態別CZ抽籤テーブル（高確時用）、及び、図222（C）のBB当籤時状態別CZ抽籤テーブル（超高確時用）が一つのテーブルで構成される。

【1930】

ソースプログラム上で実際に参照されるBB当籤時状態別CZ抽籤テーブルは、図280に示すように、ラベル「dBHSVCTB」が付された先頭アドレスから16バイト先のアドレスまでの格納領域（17バイトの格納領域）に配置される。

【1931】

BB当籤時状態別CZ抽籤テーブルの先頭アドレスには、抽籤回数と、CZ抽籤で使用する出玉抽籤用乱数値が格納された乱数格納領域のアドレスを指定するためのオフセット値とが格納された1バイトの複合データ「.LOW.(wRAND1BS+cRNDAT09)+20H*(0)」が格納される。なお、この複合データでは、抽籤回数は0回であり（抽籤指定無し）、オフセット値は「0FH」（「F00FH」の下位側1バイトの値）となる。

【1932】

複合データの格納アドレス（BB当籤時状態別CZ抽籤テーブルの先頭アドレス）の次のアドレスに格納されているデータ「.LOW.wDDM_STS」は、現在の抽籤状態（低確、高確又は超高確）に関するデータ（出玉ステータスデータ）である。

【1933】

出玉ステータスデータの格納アドレスの次のアドレスに格納されているデータ「.LOW.wDDMFLG_A」は、現ゲームで取得された抽籤用フラグの種別に対応するフラググループ（フラググループA）の種別に関するデータ（フラググループAの指定データ）である。具体的には、データ「.LOW.wDDMFLG_A」は、フラググループAの種別に対応する抽籤値群が格納された領域を指定するためのデータである。

【1934】

なお、本実施形態では、図222（A）～222（C）に示すように、BB当籤時状態別CZ抽籤テーブルで用いられる抽籤用フラグの種別は、「通常リプ、ベル」、「弱スイカ」、「弱チェリー」、「弱チャンスリプ」、「強スイカ」、「強チェリー」、「強チャンスリプ」、「強ベル」、「リーチ目リプ」及び「確定チェリー」の10種類が設けられているが、ソースプログラム上では、これらの抽籤用フラグが4種類のフラググループ（フラググループA）に振り分けられる。そして、BB当籤時状態別CZ抽籤テーブルを使用したCZ抽籤では、ソースプログラム上において、10種類の抽籤用フラグの代わりに4種類のフラググループAが抽籤パラメータとして用いられる。

【1935】

具体的には、抽籤用フラグ「通常リプ、ベル」及び「確定チェリー」が「ハズレ&プレミア系」と称するフラググループAでまとめられ、抽籤用フラグ「弱スイカ」、「弱チェリー」及び「弱チャンスリプ」が「弱レア&弱スイカ」と称するフラググループAでまとめられ、抽籤用フラグ「強スイカ」が「強スイカ」と称するフラググループAでまとめられ、抽籤用フラグ「強チェリー」、「強チャンスリプ」、「強ベル」及び「リーチ目リプ」

10

20

30

40

50

が「強レア&EP役」と称するフラググループAでまとめられる。

【1936】

フラググループAの指定データの格納アドレスの次のアドレスから2バイト先のアドレスまでの領域（3バイトの格納領域）には、1バイトの領域毎に、各抽籤状態（低確、高確又は超高確）における抽籤値（抽籤係数）群の格納領域の先頭アドレスを指定するためのデータが格納される。例えば、データ「LOW.WDDMFLG_A」の格納アドレスの次のアドレスに格納されているデータ「(80H)+dBBH_LOW-\$」は、抽籤状態が「低確」である場合に使用される抽籤値（抽籤係数）群の格納領域の先頭アドレス（ラベル「dBBH_LOW」が付されたアドレス）を指定するためのデータである。

【1937】

なお、上述したBB当籤時状態別CZ抽籤テーブルの先頭アドレスから5バイト先のアドレスまでの格納領域（6バイトの格納領域）に格納された各種データは、本発明に係る抽籤選択情報の一具体例を示すものである。また、これらのデータが格納されている領域（6バイトの格納領域）は、本発明に係る抽籤選択格納領域の一具体例を示すものである。

【1938】

そして、BB当籤時状態別CZ抽籤テーブル内において、ラベル「dBBH_LOW」が付されたアドレスから10バイト先のアドレスまでの格納領域（11バイトの格納領域）に、抽籤状態毎の抽籤値（抽籤係数）群の格納領域が配置される。この抽籤値群の格納領域内では、ラベル「dBBH_LOW」が付されたアドレスから2バイト先のアドレスまでの格納領域（3バイトの格納領域）に低確時用の抽籤値群が格納され、ラベル「dBBH_HIG」が付されたアドレスから3バイト先のアドレスまでの格納領域（4バイトの格納領域）に高確時用の抽籤値群が格納され、ラベル「dBBH_SPH」が付されたアドレスから3バイト先のアドレスまでの格納領域（4バイトの格納領域）に超高確時用の抽籤値群が格納される。なお、上述したBB当籤時状態別CZ抽籤テーブルで規定されている抽籤状態毎の抽籤値（抽籤係数）群の格納領域は、本発明に係る抽籤値格納領域の一具体例を示すものである。

【1939】

本実施形態では、図280に示すように、抽籤値群の格納領域内において、ラベル「dBBH_HIG」が付されたアドレス（高確時用の抽籤値格納領域の先頭アドレス）には、抽籤状態が高確であり且つフラググループAが「ハズレ&プレミア系」である場合に使用される抽籤値データ（非当籤確定データ）を指定するためのラベルデータ「cPRB_0」が格納される。また、本実施形態では、BB当籤時状態別CZ抽籤テーブルを使用したCZ抽籤において、抽籤状態が低確であり且つフラググループAが「強レア&EP役」（抽籤用フラグが「強チェリー」、「強チャンスリブ」、「強ベル」又は「リーチ目リブ」）である場合にも、非当籤が確定する（図222（A）～222（C）参照）。

【1940】

それゆえ、抽籤状態が低確であり且つフラググループAが「強レア&EP役」である場合に使用される抽籤値データ（非当籤確定データ）は、抽籤状態が高確であり且つフラググループAが「ハズレ&プレミア系」である場合に使用される抽籤値データ（非当籤確定データ）と同じになる。そして、図280に示すBB当籤時状態別CZ抽籤テーブル内において、前者の抽籤値データと後者の抽籤値データとを別個に規定すると、両者は隣り合うアドレスに格納されることになる。

【1941】

そこで、本実施形態では、テーブルデータの容量を節約するため、図280に示すように、高確時用の抽籤値格納領域の先頭アドレス（ラベル「dBBH_HIG」が付されたアドレス）に格納されるデータ（ラベルデータ「cPRB_0」）を、抽籤状態が低確であり且つフラググループAが「強レア&EP役」である場合のデータとしても用いる（兼用する）。すなわち、ソースプログラム上で参照されるBB当籤時状態別CZ抽籤テーブル内では、配置位置が互いに隣り合う2つの抽籤値群の格納領域間（ラベル「dBBH_LOW」が付されたアドレスから始まる低確時用の抽籤値群の格納領域とラベル「dBBH

10

20

30

40

50

「H I G」が付されたアドレスから始まる高確時用の抽籤値群の格納領域との間)において、抽籤値に関するデータ(ラベルデータ)の一部が兼用されている。

【1942】

〔C Z中ART抽籤テーブル(A)～(H)〕

図281は、ソースプログラム上で実際に参照される、図226(A)～226(D)及び図227(E)～227(H)のC Z中ART抽籤テーブルの構成例を示す図である。なお、ソースプログラム上では、遊技状況毎に分けられた図226(A)～226(D)及び図227(E)～227(H)の8種類のC Z中ART抽籤テーブルが一つのテーブルで構成される。

【1943】

ソースプログラム上で実際に参照されるC Z中ART抽籤テーブルは、図281に示すように、ラベル「d V C S V O T B」が付された先頭アドレスから30バイト先のアドレスまでの格納領域(31バイトの格納領域)に配置される。

【1944】

C Z中ART抽籤テーブルの先頭アドレスには、抽籤回数と、ART抽籤で使用する出玉抽籤用乱数値が格納された乱数格納領域のアドレスを指定するためのオフセット値とが格納された1バイトの複合データ「.LOW.(w R A N D 1 B S + c R N D A T 1 2) + 2 0 H * (0)」が格納される。なお、この複合データでは、抽籤回数は0回であり(抽籤指定無し)、オフセット値は「10H」(「F010H」の下位側1バイトの値)となる。

【1945】

複合データの格納アドレス(C Z中ART抽籤テーブルの先頭アドレス)の次のアドレスに格納されているデータ「.LOW.w D D M F L G _ D」は、現ゲームで取得された抽籤用フラグの種別に対応するフラググループ(フラググループD)の種別に関するデータ(フラググループDの指定データ)である。具体的には、データ「.LOW.w D D M F L G _ D」は、フラググループDの種別に対応する抽籤値群が格納された領域を指定するためのデータである。

【1946】

なお、本実施形態では、図226(A)～226(D)及び図227(E)～227(H)に示すように、C Z中ART抽籤テーブルで用いられる抽籤用フラグの種別は、「はずれ」、「通常リブ、ベル」、「弱スイカ」、「弱チェリー」、「弱チャンスリブ」、「強スイカ」、「強チェリー」、「強チャンスリブ」、「強ベル」、「フェイク7」、「7揃い」及び「BAR揃い」の12種類が設けられているが、ソースプログラム上では、これらの抽籤用フラグが4種類のフラググループ(フラググループD)に振り分けられる。そして、C Z中ART抽籤テーブルを使用したART抽籤では、ソースプログラム上において、12種類の抽籤用フラグの代わりに4種類のフラググループDが抽籤パラメータとして用いられる。

【1947】

具体的には、抽籤用フラグ「はずれ」及び「通常リブ、ベル」が「ハズレ」と称するフラググループDでまとめられ、「弱チェリー」及び「フェイク7」が「弱レアB」と称するフラググループDでまとめられ、「弱スイカ」、「弱チャンスリブ」、「強スイカ」及び「強チェリー」、「強チャンスリブ」、「強ベル」が「その他レア役系」と称するフラググループDでまとめられ、「7揃い」及び「BAR揃い」が「プレミア系」と称するフラググループDでまとめられる。ただし、本実施形態では、詳細な説明は省略するが、抽籤用フラグ「はずれ」は、遊技状況が「初当たりC Z」である場合には、フラググループD「ハズレ」に属するが、その他の遊技状況(「C Z (ART後)」、「特殊C Z」)では、その遊技状況に応じて属するフラググループDが適宜変更される。

【1948】

フラググループDの指定データの格納アドレスの次のアドレスに格納されているデータ「.LOW.w D D M P A R _ X」は、現在の遊技状況を示すデータ(0(N V C 終了次遊技)～8(V C R):遊技状況指定データ)である。なお、図281に示すC Z中ART

10

20

30

40

50

抽籤テーブルに格納されているラベル「wDDMPAR__X」で指定される領域は、図216（モード移行抽籤テーブル（D），（E））で説明したラベル「wDDMPAR__X」で指定される領域と同じ領域である。しかしながら、この領域に格納されるデータは抽籤の種類により異なり、この領域は汎用的に使用される。

【1949】

遊技状況指定データの格納アドレスの次のアドレスから3バイト先のアドレスまでの格納領域（4バイトの格納領域）には、1バイトの領域毎に、フラググループDの種別に対応付けられた抽籤値（抽籤係数）群の格納領域の先頭アドレスを指定するためのデータが格納される。例えば、データ「.LOW.wDDMPAR__X」の格納アドレスの次のアドレスに格納されているデータ「(80H)+dVCVO__0-\$」は、フラググループDの種別が「ハズレ」である場合に使用される抽籤値（抽籤係数）群の格納領域の先頭アドレス（ラベル「dVCVO__0」が付されたアドレス）を指定するためのデータである。

【1950】

なお、上述したCZ中ART抽籤テーブルの先頭アドレスから6バイト先のアドレスまでの格納領域（7バイトの格納領域）に格納された各種データは、本発明に係る抽籤選択情報の一具体例を示すものである。また、これらのデータが格納されている格納領域（7バイトの格納領域）は、本発明に係る抽籤選択格納領域の一具体例を示すものである。

【1951】

そして、CZ中ART抽籤テーブル内において、ラベル「dVCVO__0」が付されたアドレスから23バイト先のアドレスまでの格納領域に、フラググループDの種別毎の抽籤値（抽籤係数）群の格納領域が配置される。なお、CZ中ART抽籤テーブルで規定されているフラググループDの種別毎の抽籤値（抽籤係数）群の格納領域は、本発明に係る抽籤値格納領域の一具体例を示すものである。

【1952】

この抽籤値群の格納領域内では、ラベル「dVCVO__0」が付されたアドレスから8バイト先のアドレスまでの格納領域（9バイトの格納領域）にフラググループDの種別が「ハズレ」である場合の抽籤値群が格納される。

【1953】

フラググループDの種別が「ハズレ」である場合の抽籤値群の格納領域の次のアドレス、すなわち、ラベル「dVCVO__2」及び「dVCVO__3」が付されたアドレスから5バイト先のアドレスまでの格納領域（6バイトの格納領域）には、フラググループDの種別が「その他レア役系」及び「プレミアム系」である場合の抽籤値群が格納される。本実施形態では、図226（A）～226（D）及び図227（E）～227（H）に示すように、フラググループDの種別が「その他レア役系」である場合の遊技状況毎の抽籤値（「256」）は、フラググループDの種別が「プレミアム系」である場合のそれと同じであるので、ソースプログラム上では、同じ抽籤値群の格納領域が使用される（兼用される）。

【1954】

また、フラググループDの種別が「その他レア役系」及び「プレミアム系」である場合の抽籤値群の格納領域の次のアドレス、すなわち、ラベル「dVCVO__1」が付されたアドレスから8バイト先のアドレスまでの格納領域（9バイトの格納領域）には、フラググループDの種別が「弱レアB」である場合の抽籤値群が格納される。

【1955】

なお、本実施形態では、抽籤値群の格納領域内において、ラベル「dVCVO__1」が付されたアドレスから2バイト先のアドレスまでの領域には、フラググループDの種別が「弱レアB」であり且つ遊技状況の値が「0（NVC 終了次遊技）」～「2（NVC 遊技数1以上）」である場合に使用される抽籤値データ（当籤確定データ）を指定するためのラベルデータ「cPRB__HIT」が格納される。また、本実施形態では、CZ中ART抽籤テーブルを使用したART抽籤において、フラググループDの種別が「その他レア役系」又は「プレミアム系」であり且つ遊技状況の値が「6（AVC2 遊技数 0）」～「8（VCR）」である場合にも、当籤が確定する。それゆえ、フラググループDの種別が「

10

20

30

40

50

その他レア役系」又は「プレミア系」であり且つ遊技状況の値が「6」～「8」である場合に使用される抽籤値データ（当籤確定データ）は、フラググループDの種別が「弱レアB」であり且つ遊技状況の値が「0」～「2」である場合に使用される抽籤値データと同じになる。

【1956】

そこで、本実施形態では、テーブルデータの容量を節約するため、図281に示すように、CZ中ART抽籤テーブルの抽籤値群の格納領域内において、先頭アドレス側から、ラベル「dVCVO__0」、「dVCVO__1」、「dVCVO__2」（「dVCVO__3」）の順（フラググループDの種別を示す数値の小さい順）で各格納領域を配置せずに、ラベル「dVCVO__0」、「dVCVO__2」（「dVCVO__3」）、「dVCVO__1」の順で各格納領域を配置する。そして、ラベル「dVCVO__1」が付されたアドレスから2バイト先のアドレスまでの領域に格納されるデータ（ラベルデータ「cPRB__HIT」）を、フラググループの種別が「その他レア役系」又は「プレミア系」であり且つ遊技状況の値が「6」～「8」である場合のデータとしても用いる（兼用する）。すなわち、ソースプログラム上で参照されるCZ中ART抽籤テーブル内では、フラググループ毎の抽籤値群の格納領域の配置順序を適宜調整して、配置位置が隣り合う2つの抽籤値群の格納領域間（ラベル「dVCVO__2」及び「dVCVO__3」が付されたアドレスから始まる抽籤値群の格納領域とラベル「dVCVO__1」が付されたアドレスから始まる高確時用の抽籤値群の格納領域との間）において、抽籤値に関するデータ（ラベルデータ）の一部が兼用されている。

【1957】

〔各種効果〕

上述のように、本実施形態では、ソースプログラム上で使用される抽籤テーブルにおいて、抽籤回数と、抽籤で使用する出玉抽籤用乱数値が格納された乱数格納領域のアドレスを指定するためのオフセット値とが格納された1バイトの複合データが規定される。この場合、2つのデータを別個に規定する必要がなくなるので、その分、テーブルデータの容量（メインROM102の使用容量）を低減することができる。その結果、メインROM102内のプログラム領域の空き容量を増やし、ROM容量が圧迫されること抑制することができるとともに、処理の高速化も図ることができる。さらに、本実施形態では、メインROM102の空き容量を増大させることができるので、遊技性能や出玉性能に関するデータやプログラムを格納するための容量を確保することができる。すなわち、メインROM102の増えた空き容量を活用して、遊技性を高めることが可能になる。

【1958】

また、本実施形態では、抽籤テーブル内で上述した構成の複合データを規定することにより、異なる種別の乱数を用いた抽籤処理が多くなっても、効率よく、使用する乱数を指定することができる。それゆえ、本実施形態では、処理のさらなる高速化が可能になる。

【1959】

さらに、本実施形態では、上述のように、ソースプログラム上で使用される抽籤テーブル内において、配置位置が隣り合う2つの抽籤値群の格納領域間で抽籤値群の格納領域の一部又は全部を兼用することができるので、抽籤テーブルで規定するデータ量を減らすことができる。それゆえ、本実施形態では、メインROM102内のプログラム領域の空き容量をより一層増やすことが可能になる。

【1960】

＜主制御回路の動作説明＞

次に、図282～図303を参照して、主制御回路90のメインCPU101が、プログラムを用いて実行する各種処理の内容について説明する。なお、上記第1実施形態のパチスロ1と同様の処理については、その説明を省略又は簡略して説明する。

【1961】

〔設定変更確認処理〕

まず、図282及び図283を参照して、電源投入（リセット割込）時処理（図64参照

10

20

30

40

50

）中のS15で行う、本実施形態の設定変更確認処理について説明する。なお、図282は、設定変更確認処理の手順を示すフローチャートであり、図283は、該フローチャート中のS1513の処理を実行するためのソースプログラムの一例を示す図である。

【1962】

まず、メインCPU101は、メインRAM103内の規定外RAM領域の初期化処理を行う（S1501）。次いで、メインCPU101は、1割り込み待ち処理を行う（S1502）。この処理は、上記第1実施形態（図68中のS42の処理）と同様にして行われる。

【1963】

次いで、メインCPU101は、RAM初期化処理を行う（S1503）。この処理では、メインCPU101は、図272に示す本実施形態のメインRAM103の遊技用RAM領域内の「RAM異常時」又は「設定変更開始時」のアドレスを、初期化開始の先頭アドレスとして設定し、該先頭アドレスから遊技用RAM領域の最終アドレスまでの情報を消去（クリア）する。

【1964】

次いで、メインCPU101は、設定用鍵型スイッチ54がオン状態であるか否かを判別する（S1504）。なお、設定用鍵型スイッチ54に差し込まれる設定キー（不図示）は、本実施形態のパチスロ1の設定（設定値「1」、「2」、「5」、「6」（設定値の内部値は「0」～「3」））を操作するための操作キーであり、設定キーがオンされていると、設定用鍵型スイッチ54がオン状態となる。

【1965】

S1504において、メインCPU101が、設定用鍵型スイッチ54がオン状態でないと判別したとき（S1504がNO判定の場合）、メインCPU101は、設定変更確認処理を終了し、処理を電源投入（リセット割込）時処理（図64参照）のS16の処理に移す。一方、S1504において、メインCPU101が、設定用鍵型スイッチ54がオン状態であると判別したとき（S1504がYES判定の場合）、メインCPU101は、メダル受付禁止の処理を行う（S1505）。

【1966】

次いで、メインCPU101は、Lレジスタに設定変更開始又は設定確認開始の情報（005H）をセットし、図70で説明した設定変更コマンド（設定変更／設定確認開始）の生成格納処理を行う（S1506）。次いで、メインCPU101は、エラーカウントリレーをオン状態にセットする（S1507）。

【1967】

次いで、メインCPU101は、設定変更及び設定確認のいずれが行われたかを判別する（S1508）。

【1968】

S1508において、メインCPU101が、設定変更が行われていない（設定確認が行われた）と判別したとき（S1508がNO判定の場合）、メインCPU101は、後述のS1515の処理を行う。一方、S1508において、メインCPU101が、設定変更が行われた（設定確認が行われていない）と判別したとき（S1508がYES判定の場合）、メインCPU101は、設定値の更新処理を行う（S1509）。なお、この処理では、更新後の設定値（内部値）は、Aレジスタに一旦格納される。

【1969】

次いで、メインCPU101は、設定値7セグ表示設定処理を行う（S1510）。この処理により、更新後の設定値が情報表示器6内の7セグLEDで表示可能になる。設定値7セグ表示設定処理の詳細については、後述の図284を参照しながら後で説明する。なお、図示しないが、設定値7セグ表示設定処理の前には、Aレジスタに一旦格納された設定値（内部値）をスタックポインタSPに退避させる処理が行われ、設定値7セグ表示設定処理の後には、スタックポインタSPに退避させていた設定値（内部値）をAレジスタに戻す処理が行われる。

10

20

30

40

50

【1970】

次いで、メインCPU101は、リセットスイッチ76がオン状態であるか否かを判別する(S1511)。

【1971】

S1511において、メインCPU101が、リセットスイッチ76がオン状態であると判別したとき(S1511がYES判定の場合)、メインCPU101は、処理をS1509の処理に戻し、S1509以降の処理を繰り返す。一方、S1511において、メインCPU101が、リセットスイッチ76がオン状態でないと判別したとき(S1511がNO判定の場合)、メインCPU101は、スタートスイッチ79がオン状態であるか否かを判別する(S1512)。

10

【1972】

S1512において、メインCPU101が、スタートスイッチ79がオン状態でないと判別したとき(S1512がNO判定の場合)、メインCPU101は、処理をS1511の処理に戻し、S1511以降の処理を繰り返す。

【1973】

一方、S1512において、メインCPU101が、スタートスイッチ79がオン状態であると判別したとき(S1512がYES判定の場合)、メインCPU101は、設定値(内部値)の格納処理、並びに、設定値の高低情報及び偶奇情報の算出格納処理を行う(S1513)。

【1974】

このS1513の処理では、メインCPU101は、更新後の設定値(内部値)をメインRAM103に設けられた設定値格納領域(不図示)に格納するとともに、更新後の設定値(内部値)に対して所定の演算を行い、該設定値の高低情報及び偶奇情報を生成(算出)する。なお、本実施形態では、所定の演算として設定値の内部値を「2」で除算する演算を行い、該除算結果として得られる、商の値を設定値の高低情報とし、余りの値を設定値の偶奇情報とする。そして、メインCPU101は、生成(算出)された設定値の高低情報及び偶奇情報を、それぞれメインRAM103に設けられた図示しない設定値高價格納領域(高低情報格納領域)及び設定値偶奇格納領域(偶奇情報格納領域)に格納する。

20

【1975】

次いで、メインCPU101は、設定用鍵型スイッチ54がオフ状態であるか否かを判別する(S1514)。

30

【1976】

S1514において、メインCPU101が、設定用鍵型スイッチ54がオフ状態でないと判別したとき(S1514がNO判定の場合)、メインCPU101は、S1514の処理を繰り返す。一方、S1514において、メインCPU101が、設定用鍵型スイッチ54がオフ状態であると判別したとき(S1514がYES判定の場合)、メインCPU101は、後述のS1515の処理を行う。

【1977】

S1508がNO判定の場合又はS1514がYES判定の場合、メインCPU101は、設定変更及び設定確認のいずれが行われたか否かを判別する(S1515)。

40

【1978】

S1515において、メインCPU101が、設定変更が行われていない(設定確認が行われた)と判別したとき(S1515がNO判定の場合)、メインCPU101は、後述のS1517の処理を行う。一方、S1515において、メインCPU101が、設定変更が行われた(設定確認が行われていない)と判別したとき(S1515がYES判定の場合)、メインCPU101は、RAM初期化処理を行う(S1516)。この処理では、メインCPU101は、図272に示すメインRAM103の遊技用RAM領域内の図示しない「設定変更終了時」のアドレス(設定値格納領域の次のアドレス)を、初期化開始の先頭アドレスとして設定し、該先頭アドレスから遊技用RAM領域の最終アドレスまでの情報を消去(クリア)する。

50

【1979】

S1516の処理後又はS1515がNO判定の場合、メインCPU101は、Lレジスタに設定変更終了又は設定確認終了の情報(004H)をセットし、図70で説明した設定変更コマンド(設定変更／設定確認終了)の生成格納処理を行う(S1517)。そして、S1517の処理後、メインCPU101は、設定変更確認処理を終了し、処理を電源投入(リセット割込)時処理(図64参照)のS16の処理に移す。

【1980】

本実施形態では、上述のようにして、設定変更確認処理が行われる。なお、本実施形態のパチスロ1では、上述のように、主制御基板71(メインCPU101)は、上記S1513の処理において、設定値に対して所定の演算を施し、設定値の高低情報及び偶奇情報を算出(生成)する。それゆえ、主制御基板71(メインCPU101)は、設定値種別算出手段としても機能する。

【1981】

上述した設定変更確認処理中のS1513の処理(設定値の内部値の格納処理、並びに、設定値の高低情報及び偶奇情報の算出格納処理)は、メインCPU101が、図283のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。具体的には、次のような処理が行われる。

【1982】

まず、メインCPU101がソースコード「LDQ (wWAVENUM), A」を実行すると、Aレジスタに格納された更新後の設定値の内部値(「0」～「3」)が、Qレジスタの格納データ(上位側アドレス値)と整数値「wWAVENUM」(下位側アドレス値)とで指定されるメインRAM103内のアドレス(先頭アドレス)に配置された設定値格納領域に格納される。次いで、メインCPU101がソースコード「DIV E, A, 2」を実行すると、Aレジスタにセットされているデータ値(設定値の内部値「0」～「3」)が「2」で除算され、該除算結果で得られる、商の値が設定値の高低情報としてAレジスタに格納され、余りの値が設定値の偶奇情報としてEレジスタに格納される。

【1983】

この演算処理(除算処理)において、設定値が「1」(設定値の内部値が「0」)である場合には、商の値が「0」となり、余りの値が「0」となる。設定値が「2」(設定値の内部値が「1」)である場合には、商の値が「0」となり、余りの値が「1」となる。設定値が「5」(設定値の内部値が「2」)である場合には、商の値が「1」となり、余りの値が「0」となる。また、設定値が「6」(設定値の内部値が「3」)である場合には、商の値が「1」となり、余りの値が「1」となる。すなわち、設定値が低値(「1」又は「2」)である場合に設定値の内部値に対して上記演算(除算)が行われると、Aレジスタに「0」がセットされ、設定値が高値(「5」又は「6」)である場合に設定値の内部値に対して上記演算(除算)が行われると、Aレジスタに「1」がセットされる。また、設定値が奇数(「1」又は「5」)である場合に設定値の内部値に対して上記演算(除算)が行われると、Eレジスタに「0」がセットされ、設定値が偶数(「2」又は「6」)である場合に設定値の内部値に対して上記演算(除算)が行われると、Eレジスタに「1」がセットされる。

【1984】

次いで、メインCPU101がソースコード「LDQ (wLOW_HIH), A」を実行すると、Aレジスタに格納された設定値の高低情報(1又は0)が、Qレジスタの格納データ(上位側アドレス値)と整数値「wLOW_HIH」(下位側アドレス値)とで指定されるメインRAM103内のアドレス(先頭アドレス)に配置された設定値高低格納領域(不図示)に格納される。

【1985】

次いで、メインCPU101がソースコード「LDQ (wEVN_ODD), E」が実行されると、Eレジスタに格納された設定値の偶奇情報(1又は0)が、Qレジスタの格納データ(上位側アドレス値)と整数値「wEVN_ODD」(下位側アドレス値)とで

10

20

30

40

50

指定されるメインRAM103内のアドレス（先頭アドレス）に配置された設定値偶奇格納領域（不図示）に格納される。なお、本実施形態では、設定値の高低情報の格納処理を設定値の偶奇情報の格納処理より先に行う例を説明したが、本発明はこれに限定されず、設定値の高低情報の格納処理が設定値の偶奇情報の格納処理の後に行われてもよい。また、設定値が第1実施形態のように、「1」～「6」（内部値が「0」～「5」）であれば、設定値の高中低情報（「2」、「1」又は「0」）が、設定値高低格納領域に格納されてもよい。

【1986】

上述した設定変更確認処理中のS1513の処理において、設定値の高低情報及び偶奇情報を所定の演算処理により予め生成（算出）し、それぞれを対応する格納領域に格納した場合、次のような効果が得られる。

10

【1987】

本実施形態では、上述のように、設定値の高低情報又は偶奇情報に応じて抽籤値（抽籤係数）が変化する抽籤テーブル（例えば、図215のモード移行抽籤テーブル、図220のモード別CZ抽籤テーブル等）を用いて抽籤処理を行う場合がある。このような場合、本実施形態では、設定値高低格納領域及び設定値偶奇格納領域にそれぞれ予め格納されている設定値の高低情報及び偶奇情報を参照するだけで所望の抽籤値（抽籤テーブル）を選択して抽籤処理を実行することができる。

【1988】

それゆえ、本実施形態では、抽籤処理毎に設定値の高低及び／又は偶奇を判定する処理が必要無くなり、その処理分の処理命令（ソースコード）を省略することができ、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。この結果、本実施形態では、メインROM102の空き容量を増やし、ROM容量が圧迫されることを抑制することができるとともに、処理の高速化も図ることができる。さらに、本実施形態では、メインROM102の空き容量を増やすことができるので、遊技性能や出玉性能に関するデータやプログラムを格納するための容量を確保することができる。それゆえ、本実施形態では、メインROM102の増えた空き容量を活用して、遊技性を高めることも可能になる。

20

【1989】

〔設定値7セグ表示処理〕

30

次に、図284及び図285を参照して、設定変更確認処理（図282参照）中のS1510で行う、設定値7セグ表示処理について説明する。なお、図284は、設定値7セグ表示処理の手順を示すフローチャートであり、図285は、該フローチャート中のS1521及びS1522の処理を実行するためのソースプログラムの一例を示す図である。

【1990】

まず、メインCPU101は、設定値（内部値）の奇数化処理を行う（S1521）。この処理では、メインCPU101は、設定値の内部値（「0」～「3」）のデータ（8ビットデータ）のビット0に「1」をセット（所定のビットセット処理）して、設定値の内部値を奇数にする。なお、この奇数化処理前の設定値の内部値が奇数（「1」又は「3」）である場合には、設定値（内部値）のビット0の値がすでに「1」であるので、この奇数化処理において、設定値のビット0の値（「1」）が維持される。

40

【1991】

次いで、メインCPU101は、7セグ表示用設定値の生成処理（S1522）。この処理では、メインCPU101は、設定値の内部値「0」、「1」、「2」及び「3」を、それぞれ、情報表示器6内の7セグLEDで表示される設定値の数値（7セグ表示用設定値）「1」、「2」、「5」及び「6」を変換する。具体的には、メインCPU101は、奇数化処理前の設定値の内部値に奇数化処理後の設定値の内部値を加算して、奇数化処理前の設定値の内部値に対応する7セグ表示用設定値を生成する。

【1992】

次いで、メインCPU101は、7セグ表示用設定値を表示データ格納領域（不図示）に

50

格納する（S1523）。次いで、メインCPU101は、図161で説明した7セグ表示データ生成処理を行う（S1524）。この処理では、メインCPU101は、7セグ表示用設定値に基づいて、設定値表示データ（7セグ表示データ）を生成する。そして、S1524の処理後、メインCPU101は、設定値7セグ表示処理を終了し、処理を設定変更確認処理（図282参照）のS1511の処理に移す。

【1993】

本実施形態では、上述のようにして、設定値7セグ表示処理が行われる。なお、設定値を表示する7セグLEDは、本発明に係る表示器の一具体例を示すものである。また、本実施形態のパチスロ1では、上述のように、主制御基板71（メインCPU101）は、設定値7セグ表示処理において、設定値表示データ（7セグ表示データ）を生成する。それゆえ、主制御基板71（メインCPU101）は、設定値表示手段としても機能する。さらに、主制御基板71（メインCPU101）は、設定値7セグ表示処理において、設定値の内部値を、対応する7セグ表示用設定値に変換する。それゆえ、主制御基板71（メインCPU101）は、設定値変換手段としても機能する。

【1994】

上述した設定値7セグ表示処理中のS1521及びS1522の処理（設定値の内部値の奇数化処理及び7セグ表示用設定値の生成処理）は、メインCPU101が、図285のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。具体的には、次のような処理が行われる。

【1995】

まず、メインCPU101がソースコード「LD E, A」を実行すると、Aレジスタにセットされているデータ値（設定値の内部値「0」～「3」）がEレジスタに格納される。この処理により、奇数化処理前の設定値の内部値をEレジスタに退避させる。

【1996】

次いで、メインCPU101がソースコード「SET 0, A」（SET命令：ビットセット処理）を実行すると、Aレジスタのビット0に「1」がセットされる。すなわち、このSET命令により、Aレジスタに格納されている設定値の内部値に対して奇数化処理が実行される。そして、このSET命令では、奇数化処理後の設定値の内部値は、Aレジスタに格納される。

【1997】

例えば、設定値の内部値が「0」（偶数）である場合において上記SET命令が実行されると、Aレジスタに格納されている奇数化処理前の設定値の内部値「0000000B」が奇数化処理より、「00000001B」（設定値の内部値＝「1」）に変換され、奇数化処理後の設定値の内部値「00000001B」がAレジスタに格納される。設定値の内部値が「1」（奇数）である場合において上記SET命令が実行されると、Aレジスタに格納されている奇数化処理前の設定値の内部値「0000001B」は、奇数化処理より変化せず、奇数化処理後の設定値の内部値「00000001B」がAレジスタに格納される。設定値の内部値が「2」（偶数）である場合において上記SET命令が実行されると、Aレジスタに格納されている奇数化処理前の設定値の内部値「0000010B」が奇数化処理より、「00000011B」（設定値の内部値＝「3」）に変換され、奇数化処理後の設定値の内部値「00000011B」がAレジスタに格納される。また、設定値の内部値が「3」（奇数）である場合において上記SET命令が実行されると、Aレジスタに格納されている奇数化処理前の設定値の内部値「0000011B」は、奇数化処理より変化せず、奇数化処理後の設定値の内部値「00000011B」がAレジスタに格納される。

【1998】

その後、メインCPU101がソースコード「ADD A, E」（ADD命令）を実行すると、Aレジスタに格納されている奇数化処理後の設定値の内部値と、Eレジスタに格納されている奇数化処理前の設定値の内部値とが加算され、該加算結果がAレジスタに格納される。このADD命令により、奇数化処理前の設定値の内部値に対応する7セグ表示用

10

20

30

40

50

設定値（「1」、「2」、「5」又は「6」：加算結果）が生成される。

【1999】

例えば、奇数化処理前の設定値の内部値が「0」である場合において上記ADD命令が実行されると、Aレジスタに格納されている奇数化処理後の設定値の内部値「1（0000001B）」と、Eレジスタに格納されている奇数化処理前の設定値の内部値「0（0000000B）」とが加算され、奇数化処理前の設定値の内部値「0」に対応する7セグ表示用設定値「1（0000001B）」が生成される。奇数化処理前の設定値の内部値が「1」である場合において上記ADD命令が実行されると、Aレジスタに格納されている奇数化処理後の設定値の内部値「1（0000001B）」と、Eレジスタに格納されている奇数化処理前の設定値の内部値「1（0000001B）」とが加算され、奇数化処理前の設定値の内部値「1」に対応する7セグ表示用設定値「2（0000010B）」が生成される。奇数化処理前の設定値の内部値が「2」である場合において上記ADD命令が実行されると、Aレジスタに格納されている奇数化処理後の設定値の内部値「3（0000011B）」と、Eレジスタに格納されている奇数化処理前の設定値の内部値「2（0000010B）」とが加算され、奇数化処理前の設定値の内部値「2」に対応する7セグ表示用設定値「5（0000101B）」が生成される。また、奇数化処理前の設定値の内部値が「3」である場合において上記ADD命令が実行されると、Aレジスタに格納されている奇数化処理後の設定値の内部値「3（0000011B）」と、Eレジスタに格納されている奇数化処理前の設定値の内部値「3（0000011B）」とが加算され、奇数化処理前の設定値の内部値「3」に対応する7セグ表示用設定値「6（0000111B）」が生成される。

【2000】

上述のように、設定値の内部値に対応する7セグ表示用設定値を特定の演算処理（本実施形態では、ビットセット処理と加算処理との組み合わせ）により生成することにより、次のような効果が得られる。

【2001】

一般的に、設定値の内部値に対応する7セグ表示用設定値を変換する手法としては、例えば、設定値の内部値を7セグ表示用設定値に変換するための表示変換用テーブルを用意する手法が用いられる。しかしながら、この手法では、メインROM102に、表示変換用テーブルを格納するとともに、変換処理において、該表示変換用テーブルにアクセスするための処理をソースプログラム上で規定する必要がある。

【2002】

それに対して、上述した本実施形態の手法では、設定値の内部値に対応する7セグ表示用設定値を特定の演算処理により生成するので、表示変換用テーブル、及び、該表示変換用テーブルにアクセスするための処理が不要となり、その分、データテーブル及びソースプログラムの容量（メインROM102の使用容量）を低減することができる。それゆえ、本実施形態では、メインROM102内のプログラム領域及びデータテーブル領域の空き容量を増やし、ROM容量が圧迫されること抑制することができるとともに、処理の高速化も図ることができる。さらに、本実施形態では、メインROM102の空き容量を増やすことができるので、遊技性能や出玉性能に関するデータやプログラムを格納するための容量を確保することができる。それゆえ、本実施形態では、メインROM102の増えた空き容量を活用して、遊技性を高めることが可能になる。

【2003】

〔メインCPUの制御によるパチスロのメイン処理〕

次に、図286を参照して、メインCPU101の制御により実行される本実施形態のパチスロ1のメイン処理（主要動作処理）について説明する。なお、図286は、メイン処理の手順を示すフローチャート（メインフロー）である。

【2004】

まず、メインCPU101は、RAM初期化処理（一遊技終了時アドレス）を行う（S2001）。この処理は、上記第1実施形態（図82中のS201の処理）と同様にして行

10

20

30

40

50

われる。次いで、メインCPU101は、図83で説明したメダル受付・スタートチェック処理を行う(S2002)。

【2005】

次いで、メインCPU101は、図91で説明した乱数取得処理を行う(S2003)。この処理では、メインCPU101は、内部当籤役抽籤用の乱数値(ハードラッチ乱数:0~65535)及びART関連の各種出玉抽籤で用いられる各種出玉抽籤用乱数値(ソフトラッチ乱数:0~65535、0~255)を抽出し、該抽出した各乱数値を、メインRAM103内の対応する乱数値格納領域(図272及び図273参照)に格納する。

【2006】

次いで、メインCPU101は、S2003で抽出した内部当籤役抽籤用の乱数値を用いて、図92で説明した内部抽籤処理を行う(S2004)。なお、この処理で決定される内部当籤役の種類、ボーナスに係る役が内部当籤役として決定された場合に移行するRT状態の種類、処理の制御パラメータ(ループ回数)などは、本実施形態のパチスロ1の仕様に対応するものに適宜変更される。

【2007】

次いで、メインCPU101は、図柄設定処理を行う(S2005)。この処理では、メインCPU101は、例えば、当り要求フラグステータス(フラグステータス情報)から内部当籤役を生成する処理、当り要求フラグデータの展開処理、当り要求フラグデータを当り要求フラグ格納領域を格納する処理等を行う。なお、図柄設定処理の詳細については、後述の図287及び図288を参照しながら後で説明する。

【2008】

次いで、メインCPU101は、ゲーム数更新処理を行う(S2006)。この処理では、メインCPU101は、今回の遊技のスタートに伴い各種遊技状態のゲーム数(残りゲーム数)を更新する。例えば、メインCPU101は、RT1状態やRT3状態中のゲーム数、CZ前兆やART前兆中の前兆ゲーム数、CZ中のゲーム数、又は、通常ART中のゲーム数を更新(1減算)する。

【2009】

次いで、メインCPU101は、スタートコマンド生成格納処理を行う(S2007)。この処理は、上記第1実施形態(図82中のS2006の処理)と同様にして行われる。次いで、メインCPU101は、図102で説明した第2インターフェースボード制御処理を行う(S2008)。なお、第2インターフェースボード制御処理は、メインRAM103の規定外用作業領域で実行される。

【2010】

次いで、メインCPU101は、スタート時出玉関連抽籤処理を行う(S2009)。この処理では、メインCPU101は、主に、S2003で抽出した各種出玉抽籤用乱数値を用い、図250~図263で説明した遊技開始時に遊技状態に応じて行われるART関連の各種出玉抽籤を行う。また、この処理では、メインCPU101は、遊技ロックの抽籤処理も行う。なお、本実施形態では、例えば、遊技者にとって有利な遊技状態への移行が確定した場合に遊技ロックが発生し、これにより、有利な遊技状態への移行確定が遊技者に報知される。

【2011】

次いで、メインCPU101は、遊技ロック設定処理を行う(S2010)。この処理では、メインCPU101は、S2009で行われた遊技ロック抽籤の結果に応じて、遊技開始時の遊技ロックを実行する。なお、遊技ロック設定処理の詳細については、後述の図292を参照しながら後で説明する。

【2012】

次いで、メインCPU101は、リール停止初期設定処理を行う(S2011)。次いで、メインCPU101は、リール回転開始処理を行う(S2012)。次いで、メインCPU101は、リール回転開始コマンド生成格納処理を行う(S2013)。このS2011~S2013の処理は、上記第1実施形態(図82中のS2009~S2011の処理)

10

20

30

40

50

と同様にして行われる。

【2013】

次いで、メインCPU101は、図134及び図135で説明した引込優先順位格納処理を行う（S2014）。次いで、メインCPU101は、図138で説明したリール停止制御処理を行う（S2015）。次いで、メインCPU101は、図145で説明した入賞検索処理を行う（S2016）。なお、これらの各処理において、設定される処理の各種制御パラメータ（ループ回数、チェック回数等）は、本実施形態のパチスロ1の仕様に対応するものに適宜変更される。

【2014】

次いで、メインCPU101は、図148で説明したイリーガルヒットチェック処理を行う（S2017）。次いで、メインCPU101は、図150で入賞チェック・メダル払出処理を行う（S2018）。なお、これらの各処理において、設定される処理の各種制御パラメータ（チェック回数等）は、本実施形態のパチスロ1の仕様に対応するものに適宜変更される。

【2015】

次いで、メインCPU101は、遊技ロック設定処理を行う（S2019）。この処理では、メインCPU101は、S2009で行われた遊技ロック抽籤の結果に応じて、遊技終了時の遊技ロックを実行する。なお、遊技ロック設定処理の詳細については、後述の図292を参照しながら後で説明する。

【2016】

次いで、メインCPU101は、停止時出玉関連抽籤処理を行う（S2020）。この処理では、メインCPU101は、主に、遊技状態に応じて、図250～図263で説明したリール停止時の各種抽籤を行う。

【2017】

次いで、メインCPU101は、BBチェック処理を行う（S2021）。この処理では、メインCPU101は、ボーナス状態の作動及び終了を制御する。具体的には、メインCPU101は、略称「クラウンBB」に係る図柄組合せが有効ラインに沿って表示されるとCBBを作動させ、また、略称「BB」に係る図柄組合せが有効ラインに沿って表示されるとNBBを作動させる。また、メインCPU101は、BB（CBB又はNBB）の作動中に払い出されたメダルの枚数が規定枚数に達すると、BBの作動を終了する。

【2018】

次いで、メインCPU101は、RTチェック処理を行う（S2022）。この処理では、メインCPU101は、図178に示す移行条件に応じて、RT状態の移行制御を行う。なお、メインCPU101は、通常状態中にRT状態をRT3状態に移行させた場合には、ART状態のセット数を付与する（図209（B-2）参照）。

【2019】

次いで、メインCPU101は、報知状態移行処理を行う（S2023）。この処理では、メインCPU101は、図206及び図207に示す移行条件に応じて、報知（ART）機能の作動の有無を考慮した遊技状態の移行制御を行う。そして、S2023の処理後（一遊技終了後）、メインCPU101は、処理をS2001の処理に戻し、S2001以降の処理を繰り返す。

【2020】

〔図柄設定処理〕

次に、図287～図291を参照して、メインフロー（図286参照）中のS2005で行う本実施形態の図柄設定処理について説明する。なお、図287及び図288は、図柄設定処理の手順を示すフローチャートであり、図289は、図柄設定処理のソースプログラム上で、実際に参照されるボーナス作動中小役当籤番号変換テーブルの構成を示す図であり、図290は、図柄設定処理で補正対象となる小役当籤番号の補正後（変換後）の小役当籤番号を示す図であり、図291は、該フローチャート中のS2033～S2035の処理を実行するためのソースプログラムの一例を示す図である。

【2021】

まず、メインCPU101は、内部抽籤処理で取得された当り要求フラグステータスに基づいて、特賞当籤番号及び小役当籤番号を抽出し、該抽出された特賞当籤番号及び小役当籤番号をメインRAM103内の当籤番号格納領域（不図示）に保存する（S2031）。この処理は、上記第1実施形態（図97中のS321の処理）と同様に行われ、除算処理により生成された、商の値が特賞（ボーナス）当籤番号となり、余りの値が小役当籤番号となる。また、この際、生成された小役当籤番号はAレジスタ（アキュムレータ）に格納される。なお、図示しないが、本実施形態における当籤番号格納領域の構成は、本実施形態のパチスロ1の仕様に対応した構成となる。

【2022】

次いで、メインCPU101は、ボーナス（有利な特別遊技）作動中であるか否かを判別する（S2032）。この判定処理は、遊技状態フラグ格納領域を参照して行われる。なお、図示しないが、本実施形態における遊技状態フラグ格納領域の構成は、本実施形態のパチスロ1の仕様に対応した構成となる。

【2023】

S2032において、メインCPU101が、ボーナス作動中でないと判別したとき（S2032がNO判定の場合）、メインCPU101は、後述のS2038の処理を行う。一方、S2032において、メインCPU101が、ボーナス作動中であると判別したとき（S2032がYES判定の場合）、メインCPU101は、ボーナス作動中小役当籤番号変換テーブルをセットする（S2033）。この処理では、メインCPU101は、ボーナス作動中小役当籤番号変換テーブルの格納領域の先頭アドレスをHLレジスタ（第1ペアレジスタ）にセットする。

【2024】

なお、本実施形態の図柄設定処理において、小役当籤番号の変換対象となる小役当籤番号は、「46」（強ベル）、「45」（共通ベル）、「3」（弱チェリブ）及び「0」（ハズレ）の4種類である。そして、ボーナス作動中小役当籤番号変換テーブルは、図289に示すように、先頭アドレス「dBBFRTSLTB」～アドレス「dBBFRTSLTB+3」までの領域（4バイト）に配置され、変換前の小役当籤番号「46」（強ベル）は、アドレス「dBBFRTSLTB」の領域に格納され、変換前の小役当籤番号「45」（共通ベル）は、アドレス「dBBFRTSLTB+1」の領域に格納され、変換前の小役当籤番号「3」（弱チェリブ）は、アドレス「dBBFRTSLTB+2」の領域に格納され、変換前の小役当籤番号「0」（ハズレ）は、アドレス「dBBFRTSLTB+3」の領域に格納される。なお、これらの変換対象となる小役当籤番号では、例えば、リールの停止制御等が、ボーナス作動時と非作動時とで異なる。

【2025】

次いで、メインCPU101は、検索カウンタ（検索値）に「4」をセットする（S2034）。なお、この処理で検索カウンタにセットされる値は、変換対象の小役の種別数、すなわち、ボーナス作動中小役当籤番号変換テーブルのバイト数である。また、この処理では、検索カウンタはBCレジスタ（第2ペアレジスタ）で構成される。

【2026】

次いで、メインCPU101は、変換対象検索処理を行う（S2035）。この処理では、メインCPU101は、S2031で取得された小役当籤番号と、ボーナス作動中小役当籤番号変換テーブルに格納された変換対象となる小役当籤番号とが一致するか否かを順次、チェックする。また、この処理では、このチェック処理が一回実施される度に、検索カウンタの値を1減算するとともに、チェック対象（変換対象）の小役当籤番号を変更する（参照するボーナス作動中小役当籤番号変換テーブル内のアドレスを+1更新する）。そして、このチェック処理は、検索カウンタの値が「0」になるまで繰り返される。すなわち、本実施形態の変換対象検索処理では、このチェック処理が4回繰り返される。なお、後述するように、この変換対象検索処理は、一つの命令コード（CPIR命令）により一括して実行される。

【2027】

次いで、メインCPU101は、S2035の変換対象検索処理において、S2031で取得された小役当籤番号と一致する小役当籤番号がボーナス作動中小役当籤番号変換テーブル内にあったか否かを判別する（S2036）。

【2028】

S2036において、メインCPU101が、S2031で取得された小役当籤番号と一致する小役当籤番号がボーナス作動中小役当籤番号変換テーブル内になかったと判別したとき（S2036がNO判定の場合）、メインCPU101は、後述のS2038の処理を行う。

【2029】

一方、S2036において、メインCPU101が、S2031で取得された小役当籤番号と一致する小役当籤番号がボーナス作動中小役当籤番号変換テーブル内にあったと判別したとき（S2036がYES判定の場合）、メインCPU101は、小役当籤番号の変換（補正）処理を行う（S2037）。この処理では、メインCPU101は、S2031で取得された小役当籤番号とボーナス作動中小役当籤番号変換テーブル内の変換対象の小役当籤番号とが一致したときの検索カウンタの値（検索値0～3）に「52」（補正值）を加算した値を変換後の小役当籤番号とする。

【2030】

具体的には、図290に示すように、S2031で取得された小役当籤番号が「46」（強ベル）である時には、検索値が「3」となるので、変換後の小役当籤番号は、「55（＝3＋52）」となり、小役当籤番号が「46」から「55」（BB中強ベル）に変換される。S2031で取得された小役当籤番号が「45」（共通ベル）である時には、検索値が「2」となるので、変換後の小役当籤番号は、「54（＝2＋52）」となり、小役当籤番号が「45」から「54」（BB中共通ベル）に変換される。S2031で取得された小役当籤番号が「3」（弱チェリプ）である時には、検索値が「1」となるので、変換後の小役当籤番号は、「53（＝1＋52）」となり、小役当籤番号が「3」から「53」（BB中弱チェリプ）に変換される。また、S2031で取得された小役当籤番号が「0」（ハズレ）である時には、検索値が「0」となるので、変換後の小役当籤番号は、「52（＝0＋52）」となり、小役当籤番号が「0」から「52」（BB中ハズレ）に変換される。

【2031】

S2037の処理後、又は、S2032或いはS2036がNO判定の場合、メインCPU101は、抽出された小役当籤番号に基づいて、小役が当籤したか否かを判別する（S2038）。この処理は、上記第1実施形態（図97中のS322の処理）と同様にして行われる。

【2032】

S2038において、メインCPU101が、小役が当籤していないと判別したとき（S2038がNO判定の場合）、メインCPU101は、後述のS2047の処理を行う。一方、S2038において、メインCPU101が、小役が当籤したと判別したとき（S2038がYES判定の場合）、メインCPU101は、小役当籤番号を減算結果の初期値としてセットする（S2039）。

【2033】

次いで、メインCPU101は、当り要求フラグテーブル（不図示）をセットする（S2040）。なお、この処理では、本実施形態のパチスロ1の仕様に対応する当り要求フラグテーブルがセットされる。次いで、メインCPU101は、減算結果を1減算し、該減算結果を更新する（S2041）。次いで、メインCPU101は、減算結果が「0」未満であるか否かを判別する（S2042）。上述したS2039～S2042の処理は、上記第1実施形態（図97中のS323～S326の処理）と同様にして行われる。

【2034】

S2042において、メインCPU101が、減算結果が「0」未満でないと判別したと

10

20

30

40

50

き（S2042がNO判定の場合）、メインCPU101は、ビット数算出処理を行う（S2043）。次いで、メインCPU101は、ビット数算出処理を行う（S2044）。なお、S2043及びS2044の処理は、上記第1実施形態（図97中のS327及びS328の処理）と同様にして行われる。そして、上述したS2041～S2043の処理は、小役当籤番号の回数だけ繰り返される。

【2035】

一方、S2042において、メインCPU101が、減算結果が「0」未満であると判別したとき（S2042がYES判定の場合）、メインCPU101は、当り要求フラグ格納領域（内部当籤役格納領域）のセット処理を行う（S2045）。なお、この処理は、上記第1実施形態（図97中のS329の処理）と同様にして行われる。また、図示しないが、本実施形態における当り要求フラグ格納領域の構成は、本実施形態のパチスロ1の仕様に対応した構成となる。次いで、メインCPU101は、図101で説明した圧縮データ格納処理を行う（S2046）。

10

【2036】

S2046の処理後又はS2038がNO判定の場合、メインCPU101は、持越役格納領域を参照して、持越役があるか否かを判別する（S2047）。なお、図示しないが、本実施形態における持越格納領域の構成は、本実施形態のパチスロ1の仕様に対応した構成となる。

【2037】

S2047において、メインCPU101が、持越役があると判別したとき（S2047がYES判定の場合）、メインCPU101は、後述のS2050の処理を行う。一方、S2047において、メインCPU101が、持越役がないと判別したとき（S2047がNO判定の場合）、メインCPU101は、S2031の処理で抽出された特賞当籤番号に基づいて、BB（CBB又はNBB）が当籤したか否かを判別する（S2048）。

20

【2038】

S2048において、メインCPU101が、BBが当籤していないと判別したとき（S2048がNO判定の場合）、メインCPU101は、図柄判定処理を終了し、処理をメインフロー（図286参照）のS2006に移す。一方、S2048において、メインCPU101が、BBが当籤したと判別したとき（S2048がYES判定の場合）、メインCPU101は、当籤した特賞当籤番号を持越役格納領域に格納する（S2049）。

30

【2039】

S2049の処理後又はS2047がYES判定の場合、メインCPU101は、特賞当籤番号を当籤番号格納領域（不図示）にセットし、当り要求フラグ格納領域（不図示）に当り要求フラグデータをセットし、RT状態をRT6状態又はRT7状態にセットする（S2050）。そして、S2050の処理後、メインCPU101は、図柄設定処理を終了し、処理をメインフロー（図286参照）のS2006に移す。

【2040】

本実施形態では、上述のようにして、図柄設定処理が行われる。なお、本実施形態のパチスロ1では、上述のように、主制御基板71（メインCPU101）は、図柄設定処理において、ボーナス作動中小役当籤番号変換テーブルに基づいて、小役当籤番号を変換（補正）する。それゆえ、主制御基板71（メインCPU101）は、内部当籤役変換手段としても機能する。また、図柄設定処理において参照されるボーナス作動中小役当籤番号変換テーブルは、本発明に係る特定情報格納領域及び特定当籤番号格納領域の一具体例を示すものである。

40

【2041】

上述した図柄設定処理中のS2033～S2036の処理（小役当籤番号の変換（補正）対象検索処理）は、メインCPU101が、図291のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。具体的には、次のような処理が行われる。

【2042】

50

まず、メインCPU101がソースコード「LD HL, dBBFRTSLTB」を実行すると、HLレジスタに整数値「dBBFRTSLTB」で表されるアドレス（ボーナス作動中小役当籤番号変換テーブルの格納領域の先頭アドレス）が格納される（S2033の処理が実行される）。次いで、メインCPU101がソースコード「LD BC, 4」を実行すると、BCレジスタに、「4」（検索カウンタ（検索値）の初期値）が格納される（S2034の処理が実行される）。

【2043】

次いで、メインCPU101がCPIR命令を実行する（S2035及びS2036の処理が実行される）。CPIR命令は、メインCPU101専用のブロックサーチ（検索）命令であり、この命令が実行されると、次の処理が行われる。

【2044】

まず、HLレジスタで指定されるメモリ（メインROM102／メインRAM103）に格納されているデータ（本実施形態では、ボーナス作動中小役当籤番号変換テーブルにセットされている変換対象の小役当籤番号）と、Aレジスタに格納されているデータ（本実施形態では、S2031で取得された小役当籤番号）とが比較される。次いで、HLレジスタの値（ボーナス作動中小役当籤番号変換テーブル内のアドレス）が1加算される（チェックする変換対象の小役当籤番号が更新される）。次いで、BCレジスタの値（検索カウンタの値）が1減算される。

【2045】

そして、上述した比較処理の結果として、HLレジスタで指定されるメモリに格納されているデータ（現在チェックされている変換対象の小役当籤番号）と、Aレジスタに格納されているデータ（S2031で取得された小役当籤番号）とが一致するという結果が得られた場合、又は、BCレジスタの値（本実施形態では、検索カウンタの値）が「0」である場合（CPIR命令の終了条件が成立した場合）、上述の一連の処理が終了する（CPIR命令による処理が終了する）。一方、CPIR命令の終了条件が成立していない場合には、上述した、HLレジスタで指定されるメモリに格納されているデータとAレジスタに格納されているデータとの比較処理、HLレジスタの値の更新（1加算）処理、BCレジスタの値の更新（1減算）処理、及び、比較結果の判定処理が、CPIR命令の終了条件が成立するまで繰り返し実行される。

【2046】

上述した小役当籤番号の変換（補正）対象検索処理において、CPIR命令を使用した場合、次のような効果が得られる。CPIR命令では、上述のように、HLレジスタで指定されるメモリに格納されているデータとAレジスタに格納されているデータとの比較処理、HLレジスタの値の更新（1加算）処理、BCレジスタの値の更新（1減算）処理、及び、比較結果の判定処理の4つの処理を1つの命令コードで一括して実行することができる。すなわち、CPIR命令では、従来、4つの命令コードにより実行されていたこの4つの処理を1つの命令コードで実行することができる。

【2047】

それゆえ、メインCPU101専用のCPIR命令（ブロックサーチ命令）を用いることにより、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。その結果、メインROM102内のプログラム領域の空き容量を増やし、ROM容量が圧迫されること抑制することができるとともに、処理の高速化も図ることができる。さらに、本実施形態では、メインROM102の空き容量を増やすことができるので、遊技性能や出玉性能に関するデータやプログラムを格納するための容量を確保することができる。それゆえ、本実施形態では、メインROM102の増えた空き容量を活用して、遊技性を高めることが可能になる。

【2048】

〔遊技ロック設定処理〕

次に、図292及び図293を参照して、メインフロー（図286参照）中のS2010及びS2019で行う遊技ロック設定処理について説明する。なお、図292は、遊技口

10

20

30

40

50

ック設定処理の手順を示すフローチャートであり、図293は、該フローチャート中のS2061～S2064の処理を実行するためのソースプログラムの一例を示す図である。

【2049】

まず、メインCPU101は、遊技ロック情報の取得処理を行う（S2061）。本実施形態では、S2009の処理内で実行される遊技ロック抽籤に当籤すると、メインRAM103に設けられた遊技ロック判定フラグ（判定フラグ）又は遊技ロック予約フラグ（予約フラグ）の格納領域（不図示：フラグ格納領域）に遊技ロック情報（遊技ロックの実行の有無やロック内容など）が格納される。この際、遊技開始時の遊技ロックに当籤した場合には、遊技ロック情報が遊技ロック判定フラグに格納され、遊技終了時（リールの全停止後）の遊技ロックに当籤した場合には、遊技ロック情報が遊技ロック予約フラグに格納される。一方、遊技ロック抽籤に当籤しなかった場合（遊技ロックが発生しない場合）には、遊技ロック判定フラグ及び遊技ロック予約フラグのいずれにも遊技ロック情報は格納されない。そして、S2061の処理では、メインCPU101は、遊技ロック判定フラグから遊技ロック情報を取得する（読み出す）。

【2050】

次いで、メインCPU101は、遊技ロック予約フラグに格納されているデータを遊技ロック判定フラグに格納する（S2062）。遊技終了時（リールの全停止後）の遊技ロック抽籤に当籤した場合には、この処理により、遊技ロック判定フラグに遊技ロック情報が格納され、遊技終了時（リールの全停止後）の遊技ロック設定処理（S2019）において、遊技ロックが実行される。次いで、メインCPU101は、遊技ロック予約フラグに格納されているデータをクリアする（S2063）。

【2051】

次いで、メインCPU101は、S2061の処理において遊技ロック情報が取得されたか否かを判別する（S2064）。なお、遊技開始時の遊技ロック設定処理（S2010）では、遊技ロック抽籤に当籤しなかった場合だけでなく、遊技終了時（リールの全停止後）の遊技ロックに当籤して遊技ロック予約フラグに遊技ロック情報が格納されている場合にも、S2064の判定結果はNO判定となる。ただし、遊技終了時（リールの全停止後）の遊技ロックに当籤した場合、遊技開始時の遊技ロック設定処理（S2010）中のS2062の処理において、遊技ロック予約フラグに格納されているデータが遊技ロック判定フラグにコピーされるので、遊技終了時（リールの全停止後）の遊技ロック設定処理（S2019）では、S2064の判定結果はYES判定となる。

【2052】

S2064において、メインCPU101が、遊技ロック情報が取得されなかったと判別したとき（S2064がNO判定の場合）、メインCPU101は、遊技ロック設定処理を終了する。一方、S2064において、メインCPU101が、遊技ロック情報が取得されたと判別したとき（S2064がYES判定の場合）、メインCPU101は、取得された遊技ロック情報に基づいて遊技停止（遊技ロック）実行処理を行う（S2065）。そして、S2065の処理後、メインCPU101は、遊技ロック設定処理を終了する。

【2053】

なお、実行中の遊技ロック設定処理が遊技開始時の遊技ロック設定処理（S2010）である場合には、メインCPU101は、遊技ロック設定処理の終了後、処理をメインフロー（図286参照）のS2011に移す。一方、実行中の遊技ロック設定処理が、遊技終了時（リールの全停止後）の遊技ロック設定処理（S2019）である場合には、メインCPU101は、遊技ロック設定処理の終了後、処理をメインフロー（図286参照）のS2020に移す。

【2054】

本実施形態では、上述のようにして、遊技ロック設定処理が行われる。なお、本実施形態のパチスロ1では、上述のように、主制御基板71（メインCPU101）は、遊技ロック設定処理を行う。それゆえ、主制御基板71（メインCPU101）は、ロック実行手段としても機能する。また、主制御基板71（メインCPU101）は、スタート時出玉

10

20

30

40

50

関連抽籤処理内において、遊技ロック抽籤を行う。それゆえ、主制御基板 7 1（メイン CPU 1 0 1）は、遊技ロック抽籤手段としても機能する。

【2055】

上述した遊技ロック設定処理中の S 2 0 6 1～S 2 0 6 4 の処理は、メイン CPU 1 0 1 が、図 2 9 3 のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。具体的には、次のような処理が行われる。

【2056】

まず、メイン CPU 1 0 1 がソースコード「LDQ HL, wLOCK__RSV-1」（所定の読み出し命令）を実行すると、Qレジスタ（拡張レジスタ）の内容（遊技ロック予約フラグのアドレスの上位側のアドレス値）と、整数値「wLOCK__RSV-1」（遊技ロック予約フラグのアドレスの下位側のアドレス値から 1 減算した値）とで指定されるアドレスが、HLレジスタ（ペアレジスタ、第 3 のレジスタ）に格納される。なお、本実施形態では、次の INLD 命令実行時に、HLレジスタの値に「1」及び「2」が加算されるので、ソースコード「LDQ HL, wLOCK__RSV-1」では、遊技ロック予約フラグのアドレス「wLOCK__RSV」から予め「1」が減算されている。また、本実施形態では、遊技ロック予約フラグのアドレス「wLOCK__RSV」の次のアドレスが遊技ロック判定フラグのアドレス「wLOCK__FLG」となる。

【2057】

次いで、メイン CPU 1 0 1 がソースコード「INLD AE, (HL)」を実行する。INLD 命令は、メイン CPU 1 0 1 専用の指定レジスタロード命令であり、この命令が実行されると、次の処理が行われる。

【2058】

まず、HLレジスタに格納されている値（アドレス「wLOCK__RSV-1」）に「1」を加算した値（遊技ロック予約フラグのアドレス「wLOCK__RSV」）で指定されるアドレスに格納されているデータと、HLレジスタに格納されている値に「2」を加算した値（遊技ロック判定フラグのアドレス「wLOCK__FLG」）で指定されるアドレスに格納されているデータとが、AEレジスタに読み出される。この際、Aレジスタ（第 1 のレジスタ）に遊技ロック判定フラグのアドレス「wLOCK__FLG」に格納されているデータが格納され、Eレジスタ（第 2 のレジスタ）に遊技ロック予約フラグのアドレス「wLOCK__RSV」に格納されているデータが格納される。その後、HLレジスタの値に「2」が加算され、HLレジスタには、遊技ロック予約フラグのアドレス「wLOCK__FLG」が格納される。本実施形態では、この INLD 命令を実行することにより、S 2 0 6 1 の処理が実行され、遊技ロック判定フラグに遊技ロック情報が格納されていれば、その遊技ロック情報が A レジスタに読み出される。

【2059】

次いで、メイン CPU 1 0 1 がソースコード「LD (HL), E」を実行すると、E レジスタに格納されたデータ（遊技ロック予約フラグに格納されているデータ）が HL レジスタで指定されるアドレス（遊技ロック判定フラグのアドレス「wLOCK__FLG」）に格納される。本実施形態では、この LD 命令を実行することにより、S 2 0 6 2 の処理が実行される。

【2060】

次いで、メイン CPU 1 0 1 がソースコード「CRLQ (wLOCK__RSV)」を実行すると、Qレジスタの内容（遊技ロック予約フラグの上位側のアドレス値）と、整数値「wLOCK__RSV」（遊技ロック予約フラグの下位側のアドレス値）とで指定されるアドレスに格納されたデータ（遊技ロック予約フラグに格納されているデータ）がクリアされる。本実施形態では、この CRLQ 命令を実行することにより、S 2 0 6 3 の処理が実行される。

【2061】

そして、メイン CPU 1 0 1 がソースコード「RT Z, A」が実行されると、A レジスタに格納されているデータ（遊技ロック判定フラグのデータ；遊技ロック情報）の有無の

10

20

30

40

50

判定処理が行われ、Aレジスタに格納されているデータ（遊技ロック情報）が無ければ（ゼロフラグ（Z）が「1」（ON状態）であれば）、処理が終了し（リターン）、Aレジスタに格納されているデータ（遊技ロック情報）があれば（ゼロフラグ（Z）が「0」（OFF状態）であれば）、図示しない次のソースコードが実行される。本実施形態では、このRT命令を実行することにより、S2064の判定処理が実行される。

【2062】

ここで、本実施形態における遊技ロックの具体的な動作例を説明する。例えば、遊技ロック抽籤で、遊技開始時の遊技ロック（例えば10秒ロック）に当籤し、且つ、遊技終了時の遊技ロック（例えば20秒ロック）にも当籤した場合には、遊技ロック判定フラグに遊技開始時の遊技ロック情報（10秒ロック）が格納され、遊技ロック予約フラグに遊技終了時の遊技ロック情報（20秒ロック）が格納される。次いで、S2010の処理において遊技ロック設定処理が行われると、S2062の処理で遊技ロック予約フラグに格納された遊技ロック情報が遊技ロック判定フラグに格納されるが、S2065の遊技停止実行処理では10秒ロック（10秒間の遊技停止、又は、10秒間の操作無効化）が実行される。そして、10秒ロックの終了後、処理がメインフロー（図286参照）のS2011の処理に移る。その後、S2019の処理において遊技ロック設定処理が行われると、この時点では、遊技ロック判定フラグに遊技終了時の遊技ロック情報（20秒ロック）が格納されているので、S2065の遊技停止実行処理では20秒ロック（20秒間の遊技停止、又は、20秒間の操作無効化）が実行される。そして、20秒ロックの終了後、処理がメインフロー（図286参照）のS2020の処理に移る。

【2063】

上述した遊技ロック情報の取得処理において、INLD命令を使用した場合、次のような効果が得られる。INLD命令では、上述のように、遊技ロック判定フラグのアドレス指定処理、遊技ロック判定フラグに格納されているデータ（1バイト）の読み出し処理、及び、遊技ロック予約フラグに格納されているデータ（1バイト）の読み出し処理の3つの処理を1つの命令コードで実行することができる。すなわち、本実施形態では、従来、2バイトのデータを読み出す際に、3つの命令コードにより実行されていたこの3つの処理を1つの命令コードで一括して実行することができる。

【2064】

それゆえ、メインCPU101専用のINLD命令（指定レジスタロード命令）を用いて遊技ロック情報を読み出すことにより、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。その結果、メインROM102内のプログラム領域の空き容量を増やし、ROM容量が圧迫されること抑制するとともに、処理の高速化も図ることができる。さらに、本実施形態では、メインROM102の空き容量を増やすことができるので、遊技性能や出玉性能に関するデータやプログラムを格納するための容量を確保することができる。それゆえ、本実施形態では、メインROM102の増えた空き容量を活用して、遊技性を高めることが可能になる。

【2065】

〔試射試験信号制御処理（規定外）〕

次に、図294～図296を参照して、割込処理（図158参照）中のS911で行う、本実施形態の試射試験信号制御処理（規定外）について説明する。なお、図294は、試射試験信号制御処理（規定外）のソースプログラム上で、実際に参照される規定外出力ポート格納領域の構成を示す図であり、図295は、規定外出力ポート格納領域に格納される各規定外出力ポートの出力データの各ビットと、各ビットに割り付けられる情報内容との関係を示した図であり、図296は、試射試験信号制御処理（規定外）の手順を示すフローチャートである。

【2066】

試射試験信号制御処理（規定外）の具体的な処理を説明する前に、図294及び図295を参照して、規定外出力ポートの種別と、各規定外出力ポートの出力データの各ビットに割り付けられる情報との関係について説明する。本実施形態では、試験信号の出力ポート

としては、条件装置 1～8 の ON / OFF 信号（条件装置信号）が出力される出力ポート、ボーナス（特賞）の ON / OFF 信号（特賞信号）が出力される出力ポート、及び、各リールの回胴制動信号が出力される出力ポートの 3 種類が設けられる。そして、生成された各試験信号は、8 ビットで構成され、規定外 RAM 領域に設けられた規定外出力ポート格納領域にそれぞれ別個に格納される。

【2067】

具体的には、図 294 に示すように、規定外 RAM 領域内において、アドレス「F216」（ラベル「WOXOUT0」）に規定外出力ポート格納領域が配置される。なお、図 294 中の「DS 3」は、規定外出力ポート格納領域として 3 バイトの領域が確保されていることを示す（「DS」はデータストレージを表す疑似命令）。また、図 294 中の「条件装置 1～8（条件装置信号）」の前に記載の「OF4H」は、条件装置信号の出力ポートのポート番号を示し、「特賞信号」の前に記載の「OF5H」は、特賞信号の出力ポートのポート番号を示し、「回胴制動信号」の前に記載の「OF6H」は、各リールの回胴制動信号の出力ポートのポート番号を示す。また、これらの 3 つの出力ポートは、マイクロプロセッサ 91 内の外部バスインターフェース 104 の端子に接続された出力 IC のポートに対応する。すなわち、これらの 3 つの出力ポートは、外部バスインターフェース 104 を介してメイン CPU 101 に接続される。そして、これらの 3 つの出力ポートにセットされた試験信号は、試験機用第 1 インターフェースボード 301 を介して試験機にポート（I/O）マップド I/O 方式で出力される。

【2068】

また、図 295 に示すように、条件装置信号（条件装置 1～8 の ON / OFF 信号）のビット 0（B0）～ビット 7（B7）には、それぞれ条件装置 1～条件装置 8 の ON / OFF 情報がセットされる。そして、条件装置信号では、ON 状態となる条件装置のビットには「1」がセットされ、OFF 状態となる条件装置のビットには「0」がセットされる。

【2069】

特賞信号のビット 3（B3）には、図 295 に示すように、BB の作動 / 非作動情報がセットされ、ビット 4（B4）には RB の作動 / 非作動情報がセットされ、ビット 5（B5）には MB の作動 / 非作動情報がセットされ、ビット 6（B6）には CB の作動 / 非作動情報がセットされ、ビット 7（B7）には SB の作動 / 非作動情報がセットされる。そして、特賞信号では、作動中のボーナス種別のビットに「1」がセットされ、その他のボーナス種別のビットには「0」がセットされる。なお、特賞信号のビット 0（B0）～ビット 2（B2）は未使用となる。

【2070】

また、各リールの回胴制動信号のビット 0（B0）には、図 295 に示すように、左リール 3L（R1）の回胴制御情報がセットされ、ビット 1（B1）には中リール 3C（R2）の回胴制御情報がセットされ、ビット 2（B2）には右リール 3R（R3）の回胴制御情報がセットされる。なお、回胴制動信号のビット 3（B3）～ビット 7（B7）は未使用となる。

【2071】

次に、図 296 を参照して、本実施形態の試射試験信号制御処理（規定外）について説明する。

まず、メイン CPU 101 は、メイン RAM 103 のスタックエリアのアドレスを退避させる（S2071）。次いで、メイン CPU 101 は、スタックポインタ（SP）に規定外スタックエリアのアドレスをセットする（S2072）。次いで、メイン CPU 101 は、全レジスタのデータを退避させる（S2073）。

【2072】

次いで、メイン CPU 101 は、回胴制動信号生成処理を行う（S2074）。この処理では、メイン CPU 101 は、試験機用第 1 インターフェースボード 301 等を介して試験機に出力される、各リールの回胴制動信号（試験信号）の生成、及び、該回胴制動信号（試験信号）の規定外出力ポート格納領域へのセット処理を行う。なお、回胴制動信号生

10

20

30

40

50

成処理の詳細については、後述の図 2 9 7 を参照しながら後で説明する。

【2073】

次いで、メインCPU101は、特賞信号制御処理を行う（S2075）。この処理では、メインCPU101は、試験機に出力される特賞信号（試験信号）の規定外出力ポート格納領域へのセット処理を行う。なお、特賞信号制御処理の詳細については、後述の図 2 9 8 及び図 2 9 9 を参照しながら後で説明する。

【2074】

次いで、メインCPU101は、条件装置信号制御処理を行う（S2076）。この処理では、メインCPU101は、条件装置信号制御フラグの状態に対応する条件装置信号（条件装置1～8のON/OFF信号：試験信号）の規定外出力ポート格納領域へのセット処理を行う。なお、条件装置信号制御処理の詳細については、後述の図 3 0 0 及び図 3 0 1 を参照しながら後で説明する。

【2075】

次いで、メインCPU101は、試験信号出力処理を行う（S2077）。この処理では、メインCPU101は、規定外出力ポート格納領域に格納された各試験信号の出力信号の一括出力処理を行う。なお、試験信号出力処理の詳細については、後述の図 3 0 2 を参照しながら後で説明する。

【2076】

次いで、メインCPU101は、S2073の処理で退避させた全レジスタのデータの復帰処理を行う（S2078）。次いで、メインCPU101は、S2071の処理で退避させたスタックエリアのアドレスをスタックポインタ（SP）にセットする（S2079）。そして、S2079の処理後、メインCPU101は、試射試験信号制御処理を終了し、処理を割込処理（図 1 5 8 参照）中のS912の処理に移す。

【2077】

〔回胴制動信号生成処理〕

次に、図 2 9 7 を参照して、試射試験信号制御処理（図 2 9 6 参照）中のS2074で行う本実施形態の回胴制動信号生成処理について説明する。なお、図 2 9 7 は、回胴制動信号生成処理の手順を示すフローチャートである。

【2078】

まず、メインCPU101は、規定外作業領域に回胴制御データ格納領域（不図示）をセットする（S2081）。次いで、メインCPU101は、リール数に「3」をセットし、回胴制動信号及びその生成状態（1バイトデータ）をクリアする（S2082）。

【2079】

次いで、メインCPU101は、回胴制御データが「停止中未満」のデータであるか否かを判別する（S2083）。

【2080】

S2083において、メインCPU101が、回胴制御データが「停止中未満」のデータであると判別したとき（S2083がYES判定の場合）、メインCPU101は、後述のS2085の処理を行う。一方、S2083において、メインCPU101が、回胴制御データが「停止中未満」のデータでないと判別したとき（S2083がNO判定の場合）、メインCPU101は、回胴制御データが「静定ホールド制御終了」のデータであるか否かを判別する（S2084）。

【2081】

S2084において、メインCPU101が、回胴制御データが「静定ホールド制御終了」のデータであると判別したとき（S2084がYES判定の場合）、メインCPU101は、後述のS2086の処理を行う。一方、S2084において、メインCPU101が、回胴制御データが「静定ホールド制御終了」のデータでないと判別したとき（S2084がNO判定の場合）、又は、S2083がYES判定の場合、メインCPU101は、回胴制動信号の生成状態（1バイトデータ）のビット3をオン状態（「1」）にする（S2085）。

10

20

30

40

50

【2082】

S2085の処理後又はS2084がYES判定の場合、メインCPU101は、生成状態の各ビットのデータを1ビット分、右（ビット7からビット0に向かう方向）にシフトする（S2086）。次いで、メインCPU101は、回胴制御データ格納領域のアドレスを次の制御対象のリールのアドレスに更新する（S2087）。

【2083】

次いで、メインCPU101は、リール数を1減算する（S2088）。次いで、メインCPU101は、リール数が「0」であるか否かを判別する（S2089）。

【2084】

S2089において、メインCPU101が、リール数が「0」でないと判別したとき（S2089がNO判定の場合）、メインCPU101は、処理をS2083の処理に戻し、S2083以降の処理を繰り返す。

10

【2085】

一方、S2089において、メインCPU101が、リール数が「0」とであると判別したとき（S2089がYES判定の場合）、メインCPU101は、生成状態のデータを規定外出力ポート格納領域内の回胴制動信号の格納領域にセットする（S2090）。そして、S2090の処理後、メインCPU101は、回胴制動信号生成処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2075の処理に移す。

【2086】

〔特賞信号制御処理〕

20

次に、図298及び図299を参照して、試射試験信号制御処理（図296参照）中のS2075で行う本実施形態の特賞信号制御処理について説明する。なお、図298及び図299は、特賞信号制御処理の手順を示すフローチャートである。

【2087】

まず、メインCPU101は、遊技状態フラグ格納領域（不図示）を参照して、遊技状態フラグを取得する（S2101）。次いで、メインCPU101は、遊技状態がRB遊技状態であるか否かを判別する（S2102）。

【2088】

S2102において、メインCPU101が、遊技状態がRB遊技状態であると判別したとき（S2102がYES判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のRB情報格納ビット（ビット4）に「1」（オン状態）をセットする（S2103）。一方、S2102において、メインCPU101が、遊技状態がRB遊技状態でないと判別したとき（S2102がNO判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のRB情報格納ビット（ビット4）に「0」（オフ状態）をセットする（S2104）。

30

【2089】

S2103又はS2104の処理後、メインCPU101は、遊技状態がBB遊技状態であるか否かを判別する（S2105）。

【2090】

S2105において、メインCPU101が、遊技状態がBB遊技状態であると判別したとき（S2105がYES判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のBB情報格納ビット（ビット3）に「1」（オン状態）をセットする（S2106）。一方、S2105において、メインCPU101が、遊技状態がBB遊技状態でないと判別したとき（S2105がNO判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のBB情報格納ビット（ビット3）に「0」（オフ状態）をセットする（S2107）。

40

【2091】

S2106又はS2107の処理後、メインCPU101は、遊技状態がCB遊技状態であるか否かを判別する（S2108）。

【2092】

50

S 2 1 0 8において、メインCPU101が、遊技状態がCB遊技状態であると判別したとき（S 2 1 0 8がYES判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のCB情報格納ビット（ビット6）に「1」（オン状態）をセットする（S 2 1 0 9）。一方、S 2 1 0 8において、メインCPU101が、遊技状態がCB遊技状態でないと判別したとき（S 2 1 0 8がNO判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のCB情報格納ビット（ビット6）に「0」（オフ状態）をセットする（S 2 1 1 0）。

【2093】

S 2 1 0 9又はS 2 1 1 0の処理後、メインCPU101は、遊技状態がMB遊技状態であるか否かを判別する（S 2 1 1 1）。

【2094】

S 2 1 1 1において、メインCPU101が、遊技状態がMB遊技状態であると判別したとき（S 2 1 1 1がYES判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のMB情報格納ビット（ビット5）に「1」（オン状態）をセットする（S 2 1 1 2）。一方、S 2 1 1 1において、メインCPU101が、遊技状態がMB遊技状態でないと判別したとき（S 2 1 1 1がNO判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のMB情報格納ビット（ビット5）に「0」（オフ状態）をセットする（S 2 1 1 3）。

【2095】

S 2 1 1 2又はS 2 1 1 3の処理後、メインCPU101は、遊技状態がSB遊技状態であるか否かを判別する（S 2 1 1 4）。

【2096】

S 2 1 1 4において、メインCPU101が、遊技状態がSB遊技状態であると判別したとき（S 2 1 1 4がYES判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のSB情報格納ビット（ビット7）に「1」（オン状態）をセットする（S 2 1 1 5）。一方、S 2 1 1 4において、メインCPU101が、遊技状態がSB遊技状態でないと判別したとき（S 2 1 1 4がNO判定の場合）、メインCPU101は、規定外出力ポート格納領域内の特賞信号のSB情報格納ビット（ビット7）に「0」（オフ状態）をセットする（S 2 1 1 6）。

【2097】

そして、S 2 1 1 5又はS 2 1 1 6の処理後、メインCPU101は、特賞信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS 2 0 7 6の処理に移す。なお、本実施形態のパチスロ1では、ボーナス種別としてBBのみが設けられているので、S 2 1 0 8～S 2 1 1 6の処理を省略してもよい。

【2098】

〔条件装置信号制御処理〕

次に、図300及び図301を参照して、試射試験信号制御処理（図296参照）中のS 2 0 7 6で行う本実施形態の条件装置信号制御処理について説明する。なお、図300及び図301は、条件装置信号制御処理の手順を示すフローチャートである。

【2099】

まず、メインCPU101は、条件装置信号制御フラグが初期状態であるか否かを判別する（S 2 1 2 1）。S 2 1 2 1において、メインCPU101が、条件装置信号制御フラグが初期状態であると判別したとき（S 2 1 2 1がYES判定の場合）、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS 2 0 7 7の処理に移す。

【2100】

一方、S 2 1 2 1において、メインCPU101が、条件装置信号制御フラグが初期状態でないと判別したとき（S 2 1 2 1がNO判定の場合）、メインCPU101は、条件装置信号制御フラグが再遊技状態識別信号のオン状態を示すものであるか否かを判別する（S 2 1 2 2）。

10

20

30

40

50

【2101】

S2122において、メインCPU101が、条件装置信号制御フラグが再遊技状態識別信号のオン状態を示すものであると判別したとき（S2122がYES判定の場合）、メインCPU101は、条件装置信号制御状態に役物条件装置信号のオン状態をセットする（S2123）。次いで、メインCPU101は、規定外出力ポート格納領域内の条件装置1～6の情報格納ビット（条件装置信号のビット0～ビット5）にRT遊技状態の情報をセットする（S2124）。そして、S2124の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2077の処理に移す。

【2102】

一方、S2122において、メインCPU101が、条件装置信号制御フラグが再遊技状態識別信号のオン状態を示すものでないと判別したとき（S2122がNO判定の場合）、メインCPU101は、条件装置信号制御フラグが再遊技状態識別信号のオフ状態を示すものであるか否かを判別する（S2125）。

【2103】

S2125において、メインCPU101が、条件装置信号制御フラグが再遊技状態識別信号のオフ状態を示すものであると判別したとき（S2125がYES判定の場合）、メインCPU101は、規定外出力ポート格納領域内の条件装置1～8の情報格納ビット（条件装置信号のビット0～ビット7）に「0」（オフ状態）をセットする（S2126）。そして、S2126の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2077の処理に移す。

【2104】

一方、S2125において、メインCPU101が、条件装置信号制御フラグが再遊技状態識別信号のオフ状態を示すものでないと判別したとき（S2125がNO判定の場合）、メインCPU101は、条件装置信号制御状態が役物条件装置信号のオン状態であるか否かを判別する（S2127）。

【2105】

S2127において、メインCPU101が、条件装置信号制御状態が役物条件装置信号のオン状態であると判別したとき（S2127がYES判定の場合）、メインCPU101は、規定外出力ポート格納領域内の条件装置1～6の情報格納ビット（条件装置信号のビット0～ビット5）に特賞当籤番号の情報をセットし、条件装置8の情報格納ビット（条件装置信号のビット7）に「1」（オン状態）をセットする（S2128）。次いで、メインCPU101は、条件装置信号出力待ちタイマーに所定の待ち時間（本実施形態では、24.58ms）をセットする（S2129）。次いで、メインCPU101は、条件装置信号制御状態に条件装置信号出力待ちの状態をセットする（S2130）。そして、S2130の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2077の処理に移す。

【2106】

一方、S2127において、メインCPU101が、条件装置信号制御状態が役物条件装置信号のオン状態でないと判別したとき（S2127がNO判定の場合）、メインCPU101は、条件装置信号制御状態が条件装置信号出力待ちの状態であるか否かを判別する（S2131）。

【2107】

S2131において、メインCPU101が、条件装置信号制御状態が条件装置信号出力待ちの状態であると判別したとき（S2131がYES判定の場合）、メインCPU101は、条件装置信号出力待ちタイマーの値が「0」であるか否かを判別する（S2132）。

【2108】

S2132において、メインCPU101が、条件装置信号出力待ちタイマーの値が「0」でないと判別したとき（S2132がNO判定の場合）、メインCPU101は、条件

10

20

30

40

50

装置信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2077の処理に移す。一方、S2132において、メインCPU101が、条件装置信号出力待ちタイマーの値が「0」とであると判別したとき（S2132がYES判定の場合）、メインCPU101は、条件装置信号制御状態に小役条件装置信号のオン状態又は条件装置信号のオフ状態をセットする（S2133）。そして、S2133の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2077の処理に移す。

【2109】

ここで再度、S2131の処理に戻って、S2131において、メインCPU101が、条件装置信号制御状態が条件装置信号出力待ちの状態でない」と判別したとき（S2131がNO判定の場合）、メインCPU101は、条件装置信号制御状態が小役条件装置信号のオン状態であるか否かを判別する（S2134）。

10

【2110】

S2134において、メインCPU101が、条件装置信号制御状態が小役条件装置信号のオン状態であると判別したとき（S2134がYES判定の場合）、メインCPU101は、規定外出力ポート格納領域内の条件装置1～6の情報格納ビット（条件装置信号のビット0～ビット5）に小役当籤番号の情報をセットし、条件装置7の情報格納ビット（条件装置信号のビット6）に「1」（オン状態）をセットする（S2135）。次いで、条件装置信号出力待ちタイマーに所定の待ち時間（本実施形態では、24.58ms）をセットする（S2136）。次いで、メインCPU101は、条件装置信号制御状態に条件装置信号出力待ちの状態をセットする（S2137）。そして、S2137の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2077の処理に移す。

20

【2111】

一方、S2134において、メインCPU101が、条件装置信号制御状態が小役条件装置信号のオン状態でない」と判別したとき（S2134がNO判定の場合）、メインCPU101は、規定外出力ポート格納領域内の条件装置1～8の情報格納ビット（条件装置信号のビット0～ビット7）に「0」（オフ状態）をセットする（S2138）。そして、S2138の処理後、メインCPU101は、条件装置信号制御処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2077の処理に移す。

30

【2112】

〔試験信号出力処理〕

次に、図302及び図303を参照して、試射試験信号制御処理（図296参照）中のS2077で行う試験信号出力処理について説明する。なお、図302は、試験信号出力処理の手順を示すフローチャートであり、図303は、試験信号出力処理を実行するためのソースプログラムの一例を示す図である。

【2113】

まず、メインCPU101は、規定外出力ポート格納領域のアドレス（「WOXOUT0」：図294参照）をセットする（S2141）。次いで、メインCPU101は、出力ポートの初期出力ポート番号（本実施形態では、「0F4H」：図294参照）をセットする（S2142）。次いで、メインCPU101は、出力ポートのポート数（本実施形態では、「3」）をセットする（S2143）。なお、S2141～S2143の処理の順序は、この例に限定されず、任意に設定することができる。

40

【2114】

次いで、メインCPU101は、一括ポート出力命令（後述のOTICR命令）を実行する（S2144）。この処理により、ポート番号「0F4H」～「0F6H」の3つの出力ポートのそれぞれに、対応する試験信号（条件装置信号、特賞信号、回胴制動信号）が連続して出力される。そして、S2144の処理後、メインCPU101は、試験信号出力処理を終了し、処理を試射試験信号制御処理（図296参照）中のS2078の処理に移す。

50

【2115】

本実施形態では、上述のようにして、試験信号出力処理が行われる。本実施形態のパチスロ1では、上述のように、主制御基板71（メインCPU101）は、試験信号出力処理を行う。それゆえ、主制御基板71（メインCPU101）は、信号出力手段としても機能する。また、本実施形態における規定外出力ポート格納領域は、本発明に係る出力データ格納領域の一具体例を示すものである。

【2116】

上述した試験信号出力処理は、メインCPU101が、図303のソースプログラムで規定されている各ソースコードを順次実行することにより行われる。具体的には、次のような処理が行われる。

【2117】

まず、メインCPU101がソースコード「LD HL, woxout0」を実行すると、ラベル「woxout0」で指定されるアドレス（規定外出力ポート格納領域の先頭アドレス「F216」）がHLレジスタ（ペアレジスタ）に読み出される。本実施形態では、このLD命令を実行することにより、S2141の処理が実行される。

【2118】

次いで、メインCPU101がソースコード「LD BC, 100H*3+0F4H」を実行すると、出力ポートの初期ポート番号「0F4H」及びポート数「3」がBCレジスタに読み出される。この際、Bレジスタ（第2のレジスタ）にポート数「3」が格納され、Cレジスタ（第1のレジスタ）に初期ポート番号「0F4H」が格納される。本実施形態では、このLD命令により、S2142及びS2143の処理が一括して実行される。

【2119】

次いで、メインCPU101がソースコード「OTICR」を実行する。OTICR命令は、メインCPU101専用の一括ポート出力命令（連続出力命令）であり、この命令が実行されると、次の処理が行われる。

【2120】

まず、HLレジスタで指定されるメモリのアドレスに格納されている値（試験信号）が、Cレジスタで指定されているポート番号の出力ポートにセット（出力）される（第1の処理）。次いで、Bレジスタの値（ポート数）が1減算され、Cレジスタの値（ポート番号）が1加算され、HLレジスタの値（試験信号の格納領域のアドレス）が1加算される（第2の処理）。そして、更新後のBレジスタの値（ポート数）が「0」である場合には、OTICR命令の処理を終了し、更新後のBレジスタの値（ポート数）が「0」でない場合には、HLレジスタで指定されるメモリに格納されている更新後の値（試験信号）の更新後の出力ポートへのセット処理、Bレジスタの更新（1減算）処理、Cレジスタの更新（1加算）処理、及び、HLレジスタの更新（1加算）処理が繰り返される（第3の処理）。

【2121】

それゆえ、試験信号出力処理において、上記OTICR命令が実行されると、条件装置信号（条件装置1～8のON/OFF信号）のポート番号「0F4H」の出力ポートへの出力処理、特賞信号のポート番号「0F5H」の出力ポートへの出力処理、及び、回胴制動信号のポート番号「0F6H」の出力ポートへの出力処理がこの順で連続して実行される。本実施形態では、このOTICR命令を実行することにより、S2144の処理が実行される。

【2122】

上述した試験信号出力処理において、OTICR命令を使用した場合、次のような効果が得られる。OTICR命令（一括ポート出力命令）では、上述のように、規定外出力ポート格納領域に格納された試験信号の読み出し処理、読み出された試験信号の対応する出力ポートへの出力処理、出力ポート（ポート番号）の更新処理、出力対象の試験信号の更新処理（規定外出力ポート格納領域内における試験信号の格納領域のアドレス更新処理）、及び、これらの一連の処理を終了するか否かの判定処理（ループ処理）の5つの処理を1

10

20

30

40

50

つの命令コードで一括して実行することができる。すなわち、本実施形態では、従来、複数種の試験信号を複数種の出力ポートに出力（セット）する処理において、5つの命令コードにより実行されていたこの5つの処理を1つの命令コードで実行することができる。

【2123】

それゆえ、メインCPU101専用のOTICR命令を用いて複数種の試験信号の複数種の出力ポートへの出力処理を行うことにより、ソースプログラムの容量（メインROM102の使用容量）を低減することができる。その結果、メインROM102内のプログラム領域の空き容量を増やし、ROM容量が圧迫されること抑制することができるとともに、処理の高速化も図ることができる。さらに、本実施形態では、メインROM102の空き容量を増やすことができるので、遊技性能や出玉性能に関するデータやプログラムを格納するための容量を確保することができる。それゆえ、本実施形態では、メインROM102の増えた空き容量を活用して、遊技性を高めることが可能になる。なお、OTICR命令は、上記第1実施形態で説明したポートI/OマップドI/O形式で、信号をメインCPU101の外部に出力するための「OUT」命令の拡張命令である。

【2124】

〔第2実施形態における各種変形例〕

上記第2実施形態では、ART状態に移行すると、ART状態の1ゲーム目にランク決めARTを行い、ART当籤時のランクを昇格させることとしているが、これに限られるものではなく、ランク決めART（昇格抽籤）は行わないこととしてもよい。この場合には、ART状態中は、ART当籤時のランクから定まる抽籤状態に基づき、各種の抽籤を行うことになる。

【2125】

また、上記第2実施形態では、各種の抽籤を抽籤用フラグに応じて行うこととしているが、抽籤用フラグは、内部当籤役から決定されるため、各種の抽籤は内部当籤役に応じて行っているといえることができる。すなわち、主制御回路90は、内部当籤役に応じて各種の抽籤を行うことができる。

【2126】

また、上記第2実施形態では、BB終了時にART状態のストックがある場合には（すなわち、ART中BB終了時には）、遊技状態がART準備中に移行し、その後、ART準備中においてRT状態がRT4状態に移行すると、遊技状態がART状態に移行する。ここで、BB終了時はRT状態がRT0状態に移行するため、ART準備中は、RT0状態からRT4状態までRT状態を移行させることになる。このとき、RT0状態からRT2状態への移行は、押し順ベル当籤時の、押し順ミスに伴い略称「RT2移行図柄」に係る図柄組合せが停止表示されることで行われるため、主制御回路90は、ART準備中に押し順ベルに当籤すると、RT0状態に滞在中は、正解の押し順を報知しない（なお、他のRT状態に滞在中は、正解の押し順を報知することとしてもよく、また、報知しないこととしてもよい）。

【2127】

また、上記第2実施形態では、ART準備中にRT状態がRT1状態やRT3状態に移行することがあり、主制御回路90は、ART準備中にRT状態がRT1状態やRT3状態に移行した場合に、後述する特典を付与することとしてもよい。なお、ART準備中（すなわち、RT4状態に移行する前）にRT状態がRT1状態に移行する場合としては、RT0状態からRT1状態に移行する場合と、RT2状態からRT1状態に移行する場合とがあるが（図178参照）、RT2状態からRT1状態に移行する場合とは、報知された押し順を無視して押し順ミスした場合であるため、主制御回路90は、ART準備中にRT2状態からRT1状態に移行した場合には特典を付与することなく、ART準備中にRT0状態からRT1状態に移行した場合には特典を付与することとしてもよい（なお、図180に示す内部抽籤テーブルでは、RT0状態では、「F__通常リプ」が単独で内部当籤役として決定されることがないが、「F__通常リプ」に対して所定の抽籤値を割り振ることで、RT0状態からRT1状態に移行可能にすることができる）。

また、特典の付与は、ART準備中にRT状態がRT1状態又はRT3状態に移行したことを契機として行うだけでなく、当該契機に加えて（又は代えて）、RT1状態又はRT3状態中の内部当籤役（抽籤用フラグ）に応じて特典を付与することとしてもよい。例えば、RT1状態又はRT3状態中は、「F__弱チェリプ」が内部当籤役として決定される確率が高く、また、RT1状態やRT3状態は、固定ゲーム数が消化されるまで継続するため、ART準備中が長引くほど特典付与の期待度を上げることができる。

【2128】

また、上記第2実施形態では、CZからART状態に移行する場合に、CZの残りゲーム数を保持したままとし、ART状態の終了後に保持した残りゲーム数に対して、ART状態への移行に伴い上乗せされたゲーム数を加算したゲーム数分だけCZを行うこととしているが、これに限られるものではない。主制御回路90は、例えば、CZからART状態に移行する場合にCZの残りゲーム数を破棄するとともに、ART状態の終了後には、所定ゲーム数分のCZを行うこととしてもよい。この所定ゲーム数は任意であるが、例えば、終了したART状態に用いたランク（当籤時のランクであってもよく、昇格抽籤後のランクであってもよい）に応じて決定することとしてもよい。また、所定ゲーム数を決定するタイミングは、CZからART状態へ移行したタイミング（CZの中断時）であってもよく、また、ランク決めARTから通常ARTへ移行したタイミング（ランク決めARTの終了時）であってもよく、また、ART状態からCZへ移行したタイミング（通常ARTの終了時）であってもよい。

【2129】

また、上記第2実施形態及び各種変形例では、遊技機としてパチスロを例に挙げて説明したが、本発明はこれに限定されない。上記第2実施形態及び各種変形例で採用したリール制御に係る特徴や設定変更及び確認に係る特徴などのパチスロ1に特有の特徴以外の特徴は、「パチンコ」と呼ばれる遊技機にも適用可能であり、同様の効果が得られる。例えば、ソースプログラム上における各種プログラム及び抽籤テーブルで適用されているデータ容量の各種節約技術や、メインCPU101専用命令コードを使用した各種処理（Qレジスタを用いたアドレスの指定命令、CPIR命令、OTICR命令、INLD命令を用いた各種処理）、規定外ROM領域及び規定外RAM領域を使用した各種処理などの特徴は、「パチンコ」にも適用可能である。

【2130】

<第1及び第2実施形態で使用されるソースコード上のラベルの説明>

上記第1及び第2実施形態のパチスロ1のソースプログラムで使用されているラベルは、所定の法則に従い定義されている。例えば、図65Cに記載のラベル「cPA__SEGC OM」等のように先頭文字が小文字「c」であれば、それは定数（const、又はconstantの「c」）を表し、図67に記載のラベル「w1__CTRL」等のように、先頭文字が小文字「w」であれば、それはメインRAM103（workRAMの「w」）を表し、図95Aのラベル「dSDTR__NR」のように先頭文字が小文字「d」であれば、それはメインROM102（dataROMの「d」）を表す。なお、小文字「c」の意味は定数であるため、小文字「c」の使用法が幅広い範囲で適用可能となっている。例えば、図65Cに示すソースコードで説明した例ではラベル「cPA__SEGC OM」をアドレス値として使用するが、例えば、図88に示すソースコードで説明した例のように、ラベル「cERR__CR」を単に数値（この例ではエラーコード値）として使用することもできる。

【2131】

<第1及び第2実施形態におけるその他の各種変形例>

上記実施形態のパチスロ1では、様々な特典を付与することとしているが、遊技者に付与する特典は任意である。例えば、遊技者に付与される出玉に関する特典（例えば、ART状態への移行、ART状態の継続期間の上乗せ、CZへの移行、CZの継続期間の上乗せ）だけでなく、出玉以外の特典（例えば、特典映像の開放や、パチスロ1のカスタム開放など）であってもよい。

【2132】

また、ART状態中（第1実施形態ではCT状態中也含む）に行われる報知の内容は上述した例に限定されず、任意である。例えば、遊技者にとって有利な状態となる特別な図柄組合せが表示される停止操作の順序（押し順）を報知してもよいし、当該図柄組合せが表示されるために必要な停止操作のタイミング（狙うべき図柄）を報知するようにしてもよい。

【2133】

遊技者にとって有利な状態としては、再遊技に係る内部当籤役の当籤確率は変化しない（又は遊技性に影響を与えない程度の範囲で変化する）が、遊技者にとって有利な停止操作の態様を報知する機能、すなわち、AT機能が作動する遊技状態であってもよい。また、遊技者にとって有利な状態としては、再遊技に係る内部当籤役の当籤確率が高くなる再遊技高確率状態（リプレイタイム）が作動するとともに、遊技者にとって有利な停止操作の態様を報知する機能が作動する、すなわち、ART機能が作動する遊技状態であってもよい。

【2134】

また、ART状態の継続期間の管理方法も任意である。例えば、ゲーム数により継続期間を管理することとしてもよく、また、セット数により継続期間を管理することとしてもよく、また、ART状態中に払い出されるメダルの枚数や差枚数により継続期間を管理することとしてもよく、また、ART状態中にメダルの払い出しに影響を与える報知を行った回数（ナビ回数）により継続期間を管理することとしてもよく、また、ART状態中の任意のタイミングで行う継続判定により継続期間を管理することとしてもよく、また、ART状態中に特定の図柄組合せが表示されるとART状態を終了させることとしてもよい。この場合、上乘せの対象は、ゲーム数、セット数、ナビ回数、差枚数などのようにART状態の継続期間の管理方法によって適宜調整される。

【2135】

また、CZの継続期間の管理も任意である。ゲーム数により継続期間を管理することとしてもよく、また、セット数により継続期間を管理することとしてもよく、また、CZ中の任意のタイミングで行う継続判定により継続期間を管理することとしてもよく、また、CZ中に特定の図柄組合せが表示されるとCZを終了させることとしてもよく、また、CZ中に行ったART抽籤の回数により継続期間を管理することとしてもよく、また、CZ中にART状態に移行した回数により継続期間を管理することとしてもよい。この場合、上乘せの対象は、ゲーム数、セット数、抽籤回数などのようにCZの継続期間の管理方法によって適宜調整される。

【2136】

また、上記実施形態のパチスロ1では、遊技者側から見て、リール表示窓4の左側に設けたサブ表示装置18において各種表示画面を表示する例を説明したが、本発明はこれに限定されない。例えば、遊技者側から見て、リール表示窓4の右側にも別のサブ表示装置を設け、このサブ表示装置においても各種表示画面を表示する構成にしてもよい。この場合、リール表示窓4の右側に設けられたサブ表示装置の表示面上にタッチセンサを設け、このタッチセンサから出力されるタッチ入力情報に基づいて、当該サブ表示装置の表示画面を切り替える構成にしてもよい。

【2137】

また、上記実施形態のパチスロ1では、報知（ART）機能の作動を、メイン（主制御基板71）側の制御の下に行うこととしているが、これに限られるものではなく、サブ（副制御基板72）側の制御により報知（ART）機能の作動を行うこととしてもよい。

【2138】

また、上記実施形態のパチスロ1では、CZやART状態へは、通常状態中に行う移行抽籤に当籤することで移行する。ここで、上記実施形態のパチスロ1では、この移行抽籤に対して設定差を持たせることもあり得るが、この移行抽籤を「設定値に差のない情報」に基づいて行うこととしてもよい。なお、「設定値に差のない情報」とは、内部抽籤処理に

10

20

30

40

50

において全ての設定値において同一の確率で内部当籤役として決定される役（設定不問役）や、全リールの停止時に全ての設定値において同一の確率で表示されることになる図柄組合せ（設定差のない図柄組合せ）を少なくとも含むものである。

【2139】

すなわち、上記変形例のパチスロ1では、設定不問役が内部当籤役として決定された場合に、C ZやA R T状態への移行抽籤を行い、設定差のある役（設定差役）が内部当籤役として決定された場合には、C ZやA R T状態への移行抽籤を行わない。また、パチスロ1では、全リールの停止時に設定差のない図柄組合せが表示された場合に、C ZやA R T状態への移行抽籤を行い、設定差のある図柄組合せが表示された場合には、C ZやA R T状態への移行抽籤を行わない。

10

なお、本明細書において、抽籤を行わないとは、抽籤自体を行わないことだけでなく、抽籤はするものの必ず非当籤となることを含むものである。

【2140】

また、C ZやA R T状態への移行抽籤は、予め定められており一切変動しない完全確率に基づき行われることとしてもよい。「予め定められており一切変動しない完全確率」とは、少なくとも設定値に基づいて変動することのない確率（すなわち、設定値に関わらず当籤する確率が同一の確率）を意味する。

なお、「予め定められており一切変動しない完全確率」として、更に、「設定値に差のない情報」に対して一義的に定められた確率であることを含めることとしてもよい。近年の遊技機では、移行抽籤に当籤し易い状態と当籤し難い状態とを設けることがある。このような場合に、「設定値に差のない情報に対して一義的に定められた確率」で移行抽籤を行う場合には、例えば、設定不問役の当籤時には、いかなる遊技状態であっても同一の確率で当籤する移行抽籤を行うことになる。反対に、「設定値に差のない情報に対して一義的に定められた確率ではない確率（すなわち、変動する確率）」で移行抽籤を行う場合には、例えば、設定不問役の当籤時に、現在の遊技状態に応じた確率で当籤する移行抽籤、すなわち、現在の遊技状態が高確率状態である場合には、高い確率で当籤する移行抽籤を行い、現在の遊技状態が低確率状態である場合には、低い確率で当籤する移行抽籤、を行うことになる。

20

【2141】

このように「設定値に差のない情報」に基づいて行う移行抽籤を、設定値に関わらず当籤する確率が同一の確率で行うことで、変形例のパチスロ1では、C ZやA R T状態への移行に関する期待度を各設定値に対して同一にすることができ、出玉性能の差を、設定差のある設定差役の当籤確率に抑えることができる。

30

【2142】

また、C ZやA R T状態へ移行するための抽籤としては、（1）内部当籤役を決定するための内部抽籤、（2）移行を行うか否かの移行抽籤、（3）移行する場合の移行先を決定するための有利区間種別抽籤の3つの抽籤が考えられる。パチスロ1では、これら3つの抽籤を、それぞれ個別の乱数値を用いて行うこととしてもよく、また、全ての抽籤を共通の乱数値を用いて行うこととしてもよく、また、2つの抽籤を共通の乱数値を用いて行い、残り1つの抽籤を個別の乱数値を用いて行うこととしてもよい。なお、共通の乱数値を用いるとは、例えば、（1）内部抽籤において「0～128」の乱数範囲において所定の設定不問役が決定される場合に、（2）「0～64」を移行抽籤に非当籤の乱数範囲、「65～128」を移行抽籤に当籤の乱数範囲とし、また、（3）「65～83」を移行先がC Z1である乱数範囲、「83～99」を移行先がC Z2である乱数範囲、「99～115」を移行先がC Z3である乱数範囲、「115～128」を移行先がA R T状態である乱数範囲とすることである。言い換えると、一つの抽籤において当籤する乱数範囲の一部に、他の抽籤の抽籤結果を割り当てることである。

40

【2143】

【その他、本発明に係る遊技機の拡張性】

上記第1及び第2実施形態のパチスロ1では、遊技者のメダルの投入操作（すなわち、手

50

持ちのメダルをメダル投入口14に対して投入する操作、又は、クレジットされたメダルをMAXベットボタン15a或いは1ベットボタン15bを操作して投入する操作)により遊技が開始され、遊技が終了したときにメダルの払い出しがある場合には、ホッパー装置51を駆動してメダル払出口24からメダルが払い出され、又は、クレジットされる形態について説明したが、本発明はこれに限定されるものではない。

【2144】

例えば、遊技者によって遊技に必要な遊技媒体が投入され、それに基づいて遊技が行われ、その遊技の結果に基づいて特典が付与される(例えば、メダルが払い出される)形態の全てに対して、本発明を適用することができる。すなわち、物理的な遊技者の動作によって遊技媒体が投入され(掛けられ)、遊技媒体が払い出される形態のみならず、主制御回路90(主制御基板71)自体が、遊技者が保有する遊技媒体を電磁的に管理し、メダルレスで遊技を可能にする形態であってもよい。なお、この場合、遊技者が保有する遊技媒体を電磁的に管理するのは、主制御回路90(主制御基板71)に装着され(接続され)且つ遊技媒体を管理する遊技媒体管理装置であってもよい。

【2145】

この場合、遊技媒体管理装置は、ROM及びRWM(あるいは、RAM)を有し、遊技機に設けられる装置であって、図示しない外部の遊技媒体取扱装置と所定のインターフェースを介して双方向通信可能に接続されるものであり、遊技媒体の貸出動作(すなわち、遊技者が遊技媒体の投入操作を行う上で、必要な遊技媒体を提供する動作)或いは遊技媒体の払い出しに係る役に入賞(当該役が成立)した場合における遊技媒体の払出動作(すなわち、遊技者に対して遊技媒体の払い出しを行う上で、必要な遊技媒体を獲得させる動作)、又は、遊技の用に供する遊技媒体を電磁的に記録する動作を行い得るものとすればよい。また、遊技媒体管理装置は、実際の遊技媒体数の管理のみならず、例えば、その遊技媒体数の管理結果に基づいて、保有する遊技媒体数を表示する保有遊技媒体数表示装置(不図示)をパチスロ1の前面に設け、この保有遊技媒体数表示装置に表示される遊技媒体数を管理するものであってもよい。すなわち、遊技媒体管理装置は、遊技者が遊技の用に供することができる遊技媒体の総数を電磁的方法により記録し、表示することができるものとすればよい。

【2146】

また、この場合、遊技媒体管理装置は、遊技者が、記録された遊技媒体数を示す信号を、外部の遊技媒体取扱装置に対して自由に送信させることができる性能(機能)を有することが望ましい。また、遊技媒体管理装置は、遊技者が直接操作する場合以外の場合には、記録された遊技媒体数を減ずることができない性能を有することが望ましい。また、遊技媒体管理装置と外部の遊技媒体取扱装置との間に外部接続端子板(不図示)が設けられる場合には、遊技媒体管理装置は、その外部接続端子板を介してでなければ、遊技者が、記録された遊技媒体数を示す信号を送信できない性能を有することが望ましい。

【2147】

遊技機には、上記の他、遊技者が操作可能な貸出操作手段、返却(精算)操作手段、外部接続端子板が設けられ、遊技媒体取扱装置には、紙幣等の有価価値の投入口、記録媒体(例えばICカード)の挿入口、携帯端末から電子マネー等の入金を行うための非接触通信アンテナ等、その他貸出操作手段、返却操作手段等の各種操作手段、遊技媒体取扱装置側外部接続端子板が設けられるようにしてもよい(いずれも不図示)。

【2148】

その際の遊技の流れとしては、例えば、遊技者が遊技媒体取扱装置に対し、上記いずれかの方法で有価価値を入金し、上記いずれかの貸出操作手段の操作に基づいて所定数の有価価値を減算し、遊技媒体取扱装置から遊技媒体管理装置に対し、減算した有価価値に対応する遊技媒体を増加させる。そして、遊技者は遊技を行い、さらに遊技媒体が必要な場合には上記操作を繰り返し行う。その後、遊技の結果、所定数の遊技媒体を獲得し、遊技を終了する際には、上記いずれかの返却操作手段を操作することにより遊技媒体管理装置から遊技媒体取扱装置に対し、遊技媒体数を送信し、遊技媒体取扱装置はその遊技媒体数を

記録した記録媒体を排出する。また、遊技媒体管理装置は遊技媒体数を送信したときに、自身が記憶する遊技媒体数をクリアする。遊技者は排出された記録媒体を景品交換するために景品カウンター等に持って行くか、又は、記録された遊技媒体に基づいて他の遊技台で遊技を行うために遊技台を移動する。

【2149】

なお、上記例では、遊技媒体管理装置から全遊技媒体数を遊技媒体取扱装置に対して送信したが、遊技機又は遊技媒体取扱装置側で遊技者が所望する遊技媒体数のみを送信し、遊技者が所持する遊技媒体を分割して処理することとしてもよい。また、上記例では、遊技媒体取扱装置が記録媒体を排出することとしたが、現金又は現金等価物を排出するようにしてもよいし、携帯端末等に記憶させるようにしてもよい。また、遊技媒体取扱装置は遊技場の会員記録媒体を挿入可能とし、遊技媒体を会員記録媒体に貯留して、後日、該貯留された遊技媒体を用いて再遊技可能とするようにしてもよい。

10

【2150】

また、遊技機又は遊技媒体取扱装置において、図示しない所定の操作手段を操作することにより遊技媒体取扱装置又は遊技媒体管理装置に対し、遊技媒体又は有価価値のデータ通信をロックするロック操作を実行可能としてもよい。その際には、ワンタイムパスワード等の遊技者にしか知り得ない情報を設定することや遊技機又は遊技媒体取扱装置に設けられた撮像手段により遊技者を記憶するようにしてもよい。

【2151】

なお、遊技媒体管理装置は、上述のように、メダルレスでのみ遊技を可能とするものであってもよいし、物理的な遊技者の動作によって遊技媒体が投入され（掛けられ）、遊技媒体が払い出される形態、及び、メダルレスで遊技を可能とする形態の両方の形態で遊技を可能とするものであってもよい。後者の場合には、遊技媒体管理装置が、上述のセレクトタ66やホッパー装置51を直接的に制御する方式を採用することもできるし、これらが主制御回路90（主制御基板71）によって制御され、その制御結果が送信されることに基づいて、遊技者が遊技の用に供することができる遊技媒体の総数を電磁的方法により記録し且つ表示する制御を行い得る方式を採用することもできる。

20

【2152】

また、上記例では、遊技媒体管理装置を、パチスロ1に適用する場合について説明しているが、例えば、遊技球を用いるスロットマシンや封入式遊技機においても同様に遊技媒体管理装置を設け、遊技者の遊技媒体が管理されるようにすることもできる。

30

【2153】

上述した遊技媒体管理装置を設けた場合には、遊技媒体が物理的に遊技に供される場合に比べて、遊技機内部のセレクトタ66やホッパー装置51などの装置を減らすことができ、遊技機の内価及び製造コストを削減できるのみならず、遊技者が直接遊技媒体に接触しないようにすることもでき、遊技環境が改善され、騒音も減らすことができるとともに、装置を減らしたことにより遊技機の消費電力を減らすことも可能になる。また、上述した遊技媒体管理装置を設けた場合には、遊技媒体や遊技媒体の投入口や払出口を介した不正行為を防止することができる。すなわち、上述した遊技媒体管理装置を設けた場合には、遊技機をとりまく種々の環境を改善可能な遊技機を提供することが可能になる。

40

【2154】

3. 第3実施形態

本発明の第3実施形態では、上記第1及び第2実施形態で説明した各種ソースプログラムで使した各種命令コードだけでなく、さらに各種機能を改良及び拡張可能とする新たな各種命令コードが設けられた遊技機を説明する。なお、本実施形態の遊技機において、新たな各種命令コードを備えること以外の構成は、上記第1及び第2実施形態の遊技機の構成と同様にすることができる。それゆえ、以下では、上記第1及び第2実施形態と同様の構成の説明は省略し、本実施形態で新たに設けられた各種命令コード、及び、それを使用した各種ソースプログラム例について説明する。

【2155】

50

本実施形態において、新たな各種命令コード及び各種データで構成（規定）された各種プログラムは、上記第1及び第2実施形態と同様に、マイクロプロセッサ91内のメインROM102（第1記憶部）に記憶される（図9参照）。また、それらの各種プログラムの実行時には、上記第1及び第2実施形態と同様に、マイクロプロセッサ91内のメインCPU101（演算制御部）が、各種プログラムを実行する。

【2156】

また、本実施形態では、メインROM102及びメインRAM103のメモリマップも上記実施形態の構成（図12参照）と同様とし、メインROM102及びメインRAM103内のアドレス値（アドレス空間）は、16ビットデータで表される。さらに、本実施形態では、メインCPU101が有する各種レジスタの構成も、上記第1及び第2実施形態の構成（図11参照）と同様である。

【2157】

<各種命令コード>

本実施形態の遊技機で使用され得る各種命令コードの一例を、図304～図357に示す。なお、図304～図357のそれぞれは、本実施形態の遊技機で使用可能な各種命令コードの一覧表であり、命令コード毎に、「命令番号」（図中の「No」欄）、「新規命令であるか否かの情報」（図中の「新」欄）、「ニーモニック（命令コードの表記形態）」、「バイト数」、「オペレーションの内容（命令実行時の処理内容）」（図中の「オペレーション」欄）、「命令実行時のフラグの状態変化態様」（図中の「フラグ」欄）及び「動作（オペレーション）の説明」（図中の「説明」欄）をまとめた表である。

【2158】

「新規命令であるか否かの情報」の欄では、本実施形態の遊技機で新規に設けられた命令コードに対しては「○」印を記載し、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用されている命令コードに対しては無印とする。

【2159】

また、「命令実行時のフラグの状態変化態様」の欄には、命令実行時における、1バイトのフラグ・レジスタF（図11参照）に格納される、Sフラグ（サインフラグ）、Zフラグ（ゼロフラグ）、TZフラグ（第2ゼロフラグ）、Hフラグ（ハーフキャリーフラグ）、P/Vフラグ（パリティ／オーバーフローフラグ）、Nフラグ（加算／減算フラグ）及びC（CY）フラグ（キャリーフラグ）のそれぞれの状態の変化態様を示す。

【2160】

なお、フラグ・レジスタFに格納される各フラグは、演算命令の実行結果が反映されるフラグである。Sフラグでは、例えば、8ビット及び16ビットの算術演算の実行結果の値が、負の値であればSフラグに「1」が格納され、正の値であればSフラグに「0」が格納される。Zフラグ（第1のゼロフラグ）では、例えば、8ビット及び16ビットの算術命令の実行結果の値において、全てのビットが「0」であればZフラグに「1」が格納され、一つのビットでも「1」があればZフラグに「0」が格納される。TZフラグ（第2のゼロフラグ、所定のフラグ）では、例えば、8ビット及び16ビットのロード命令、8ビット及び16ビットの算術命令等の実行結果の値において、全てのビットが「0」であればTZフラグに「1」が格納され、一つのビットでも「1」があればTZフラグに「0」が格納される。Hフラグでは、例えば、8ビットの算術命令の実行結果の値において、ビット3からビット4に桁上がりがある場合にはHフラグに「1」が格納され、ビット3からビット4に桁上がりがない場合にはHフラグに「0」が格納される。P/Vフラグでは、例えば、算術演算の結果、オーバーフローが生じた場合にはP/Vフラグに「1」が格納され、オーバーフローが生じなかった場合にはP/Vフラグに「0」が格納される。Nフラグでは、例えば、加算命令が実行されればNフラグに「0」が格納され、減算命令が実行されればNフラグに「1」が格納される。また、Cフラグ（CYフラグ）では、例えば、算術命令の結果、桁上がりがある場合にはCフラグに「1」が格納され、桁上がりがない場合にはCフラグに「0」が格納される。

【2161】

フラグの状態変化態様の欄中の所定フラグに「1」が記載されている命令コード（例えば、後述の図329参照）では、当該命令コードの実行により、所定フラグに「1」が格納される（セットされる）ことを示す。フラグの状態変化態様の欄中の所定フラグに「0」が記載されている命令コード（例えば、後述の図339参照）では、当該命令コードの実行により、所定フラグに「0」が格納される（セットされる）ことを示す。また、フラグの状態変化態様の欄中の所定フラグに「・」が記載されている命令コード（例えば、後述の図304参照）では、当該命令コードの実行により、所定フラグの状態が変化しない（命令実行前の状態が維持される）ことを示す。さらに、フラグの状態変化態様の欄中の所定フラグに上下の矢印が記載されている命令コード（例えば、後述の図309参照）では、当該命令コードの実行結果に応じて、所定フラグの状態が適宜変化する（適宜設定される）ことを示す。

10

【2162】

〔エクステンジ命令コード〕

図304は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能なエクステンジ命令（交換命令）コードの一覧表である。

【2163】

図304中の命令番号A1（「EX AF, AF'」）～命令番号A9（「EX (SP, IY)」）のエクステンジ命令コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図304中の命令番号A10（「EX r, (ss)」）～命令番号A19（「EXQ r, (k)」）のエクステンジ命令コードは、本実施形態で新たに設けられた命令コードである。なお、エクステンジ命令コードの実行時には、その命令コードの種別（命令番号）に関係なく、各フラグの状態は変化しない。

20

【2164】

エクステンジ命令コードは、図304中の「ニーモニック」欄に示すように、オペランド(OP)として、命令(第1OP)、第1格納先(第2OP)及び第2格納先(第3OP)がこの順で配置(規定)されて構成される。例えば、命令番号No. A10のエクステンジ命令コード「EX r, (ss)」では、第1OP(命令)は「EX」であり、第2OP(第1格納先を示す情報)は「r」であり、第3OP(第2格納先を示す情報)は「(ss)」である。そして、エクステンジ命令コードが実行されると、第1格納先に格納されているデータ(値)と第2格納先に格納されているデータ(値)とが交換される。以下、主な、エクステンジ命令コードについて詳述する

30

【2165】

(1) エクステンジ命令コード「EX ss, qq」(No. A6)

エクステンジ命令コード「EX ss, qq」は、2バイトの命令コードである。なお、このエクステンジ命令コードのニーモニックで第2OPに規定されている「ss」は、16ビットの汎用ペアレジスタ(図11で説明したBC、DE、HLレジスタ：以下、「ペアレジスタss」ともいい、また、「レジスタペアss」と称されていることもある)を示し、第3OPに規定されている「qq」は、16ビットのインデックス・レジスタ(図11で説明したインデックス・レジスタIX、IY：以下、「インデックス・レジスタqq」又は「IXレジスタ」、「IYレジスタ」ともいう)を示す。

40

【2166】

エクステンジ命令コード「EX ss, qq」が実行されると、16ビットの汎用ペアレジスタss(BC、DE又はHLレジスタ)に格納されている値と、インデックス・レジスタqq(IX又はIYレジスタ)に格納されている値とが交換される。すなわち、この命令が実行されると、命令実行前のペアレジスタssに格納されていた値がインデックス・レジスタqqに格納され、命令実行前のインデックス・レジスタqqに格納されていた値がペアレジスタssに格納される。

【2167】

図305に、ソースプログラム上で実際に使用され得るエクステンジ命令コード「EX

50

「ss, qq」の種類の一覧を示す。エクスチェンジ命令コード「EX ss, qq」としては、3種類の第2OP（BC、DE及びHLレジスタ）のそれぞれに対して、2種類の第3OP（IX及びIYレジスタ）が規定可能であり、合計で6種類の命令コードが設けられている。

【2168】

例えば、エクスチェンジ命令コード「EX BC, IX」（No. A6a）が実行されると、BCレジスタに格納されている値と、IXレジスタに格納されている値とが交換される。すなわち、エクスチェンジ命令コード「EX BC, IX」が実行されると、命令実行前のBCレジスタに格納されていた値がIXレジスタに格納され、命令実行前のIXレジスタに格納されていた値がBCレジスタに格納される。

【2169】

（2）エクスチェンジ命令コード「EX r, (ss)」（No. A10）

エクスチェンジ命令コード「EX r, (ss)」は、2バイトの命令コードである。なお、このエクスチェンジ命令コードのニーモニックで第2OPに規定されている「r」は、8ビットの汎用レジスタ（A、B、C、D、E、H、Lレジスタ：以下、「レジスタr」ともいう）を示し、第3OPに規定されている「(ss)」は、ペアレジスタssに格納されている値で指定されるアドレス領域を示す。なお、ペアレジスタssに格納されている値としては、他の命令の実行によりメインRAM103内のアドレス領域又は内蔵レジスタエリア内のアドレス領域が指定されている必要がある。

【2170】

エクスチェンジ命令コード「EX r, (ss)」が実行されると、レジスタr（A、B、C、D、E、H又はLレジスタ）に格納されている値と、ペアレジスタss（BC、DE、又はHLレジスタ）に格納された値で指定されるアドレス領域（指定メモリ）に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX r, (ss)」が実行されると、命令実行前のレジスタrに格納されていた値がペアレジスタssで指定されるアドレス領域に格納され、命令実行前のペアレジスタssで指定されるアドレス領域に格納されていた値がレジスタrに格納される。なお、指定メモリとは、メインROM102（0000H～2FFFFH）、メインRAM103（F000H～F3FFFH）、内蔵レジスタ領域（FE00H～FEBFH）、及び、XCSデコードエリア（FED0H～FEFDH）内において指定されるアドレス領域のことである（図12参照）。

【2171】

図306に、ソースプログラム上で実際に使用され得るエクスチェンジ命令コード「EX r, (ss)」の種類の一覧を示す。エクスチェンジ命令コード「EX r, (ss)」としては、5種類の第2OPのそれぞれに対して、3種類の第3OPが規定可能であり、合計で15種類の命令コードが設けられている。

【2172】

具体的には、第3OPとして「(BC)」(BCレジスタで指定されるアドレス領域)が規定される場合には、第2OPには、A、D、E、H及びLレジスタのいずれかが規定される（No. A10a～A10e）。すなわち、この場合には、第2OPにはB及びCレジスタは規定されない。第3OPとして「(DE)」(DEレジスタで指定されるアドレス領域)が規定される場合には、第2OPには、A、B、C、H及びLレジスタのいずれかが規定される（No. A10f～A10j）。すなわち、この場合には、第2OPにはD及びEレジスタは規定されない。また、第3OPとして「(HL)」(HLレジスタで指定されるアドレス領域)が規定される場合には、第2OPには、A、B、C、D及びEレジスタのいずれかが規定される（No. A10k～A10o）。すなわち、この場合には、第2OPにはH及びLレジスタは規定されない。

【2173】

例えば、エクスチェンジ命令コード「EX A, (BC)」(No. A10a)が実行されると、Aレジスタに格納されている値と、BCレジスタに格納された値で指定されるア

10

20

30

40

50

ドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX A, (BC)」が実行されると、命令実行前のAレジスタに格納されていた値がBCレジスタで指定されるアドレス領域に格納され、命令実行前のBCレジスタで指定されるアドレス領域に格納されていた1バイトの値がAレジスタに格納される。

【2174】

例えば、エクスチェンジ命令コード「EX A, (DE)」(No. A10f)が実行されると、Aレジスタに格納されている値と、DEレジスタに格納された値で指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX A, (DE)」が実行されると、命令実行前のAレジスタに格納されていた値がDEレジスタで指定されるアドレス領域に格納され、命令実行前のDEレジスタで指定されるアドレス領域に格納されていた1バイトの値がAレジスタに格納される。

10

【2175】

また、例えば、エクスチェンジ命令コード「EX A, (HL)」(No. A10k)が実行されると、Aレジスタに格納されている値と、HLレジスタに格納された値で指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX A, (HL)」が実行されると、命令実行前のAレジスタに格納されていた値がHLレジスタで指定されるアドレス領域に格納され、命令実行前のHLレジスタで指定されるアドレス領域に格納されていた1バイトの値がAレジスタに格納される。

【2176】

20

(3) エクスチェンジ命令コード「EX A, (mn)」(No. A11)～「EX L, (mn)」(No. A17)

エクスチェンジ命令コード「EX A, (mn)」～「EX L, (mn)」はそれぞれ、4バイトの命令コードである。なお、このエクスチェンジ命令コードのニーモニックで第2OPに規定されている「A」、「B」、「C」、「D」、「E」、「H」及び「L」は、それぞれ、A、B、C、D、E、H及びLレジスタ(8ビットの汎用レジスタ)を示し、第3OPに規定されている「(mn)」は、16ビットの整数値mn(16ビットデータmn: 0～65535)で指定されるアドレス領域を示す。なお、16ビットの整数値mnとしては、メインRAM103内のアドレス領域又は内蔵レジスタエリア内のアドレス領域が予め指定されている必要がある。

30

【2177】

エクスチェンジ命令コード「EX A, (mn)」～「EX L, (mn)」のいずれか(エクスチェンジ命令コード「EX r, (mn)」)が実行されると、第2OPに規定されたレジスタr(A、B、C、D、E、H又はLレジスタ)に格納されている値と、16ビットの整数値mnで指定されるアドレス領域(指定メモリ)に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX r, (mn)」が実行されると、命令実行前の1バイトのレジスタrに格納されていた値が16ビットの整数値mnで指定されるアドレス領域に格納され、命令実行前の16ビットの整数値mnで指定されるアドレス領域に格納されていた1バイトの値が1バイトのレジスタrに格納される。

40

【2178】

例えば、エクスチェンジ命令コード「EX A, (mn)」(No. A11)が実行されると、Aレジスタに格納されている値と、16ビットの整数値mnで指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX A, (mn)」が実行されると、命令実行前のAレジスタに格納されていた値が16ビットの整数値mnで指定されるアドレス領域に格納され、命令実行前の16ビットの整数値mnで指定されるアドレス領域に格納されていた1バイトの値がAレジスタに格納される。

【2179】

また、例えば、エクスチェンジ命令コード「EX H, (mn)」(No. A16)が実

50

行されると、Hレジスタに格納されている値と、16ビットの整数値mnで指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX H, (mn)」が実行されると、命令実行前のHレジスタに格納されていた値が16ビットの整数値mnで指定されるアドレス領域に格納され、命令実行前の16ビットの整数値mnで指定されるアドレス領域に格納されていた1バイトの値がHレジスタに格納される。

【2180】

(4) エクスチェンジ命令コード「EX r, (qq)」(No. A18)

エクスチェンジ命令コード「EX r, (qq)」は、2バイトの命令コードである。なお、このエクスチェンジ命令コードのニーモニックで第3OPに規定されている「(qq)」は、インデックス・レジスタqqに格納された値で指定されるアドレス領域を示す。なお、インデックス・レジスタqqに格納されている値としては、他の命令の実行によりメインRAM103内のアドレス領域又は内蔵レジスタエリア内のアドレス領域が指定されている必要がある。

【2181】

エクスチェンジ命令コード「EX r, (qq)」が実行されると、レジスタr(A、B、C、D、E、H又はLレジスタ)に格納されている値と、インデックス・レジスタqq(IX又はIYレジスタ)に格納された値で指定されるアドレス領域(指定メモリ)に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX r, (qq)」が実行されると、命令実行前のレジスタrに格納されていた値がインデックス・レジスタqqで指定されるアドレス領域に格納され、命令実行前のインデックス・レジスタqqで指定されるアドレス領域に格納されていた1バイトの値がレジスタrに格納される。

【2182】

図307に、ソースプログラム上で実際に使用され得るエクスチェンジ命令コード「EX r, (qq)」の種類の一覧を示す。エクスチェンジ命令コード「EX r, (qq)」としては、7種類の第2OPのそれぞれに対して、2種類の第3OPが規定可能であり、合計で14種類の命令コードが設けられている。

【2183】

具体的には、第3OPとして「(IX)」(IXレジスタで指定されるアドレス領域)が規定される場合には、第2OPには、A、B、C、D、E、H及びLレジスタのいずれかが規定される(No. A18a~A18g)。第3OPとして「(IY)」(IYレジスタで指定されるアドレス領域)が規定される場合には、第2OPには、A、B、C、D、E、H及びLレジスタのいずれかが規定される(No. A18h~A18n)。

【2184】

例えば、エクスチェンジ命令コード「EX A, (IX)」(No. A18a)が実行されると、Aレジスタに格納されている値と、IXレジスタに格納された値で指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX A, (IX)」が実行されると、命令実行前のAレジスタに格納されていた値がIXレジスタで指定されるアドレス領域に格納され、命令実行前のIXレジスタで指定されるアドレス領域に格納されていた1バイトの値がAレジスタに格納される。

【2185】

また、例えば、エクスチェンジ命令コード「EX A, (IY)」(No. A18h)が実行されると、Aレジスタに格納されている値と、IYレジスタに格納された値で指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EX A, (IY)」が実行されると、命令実行前のAレジスタに格納されていた値がIYレジスタで指定されるアドレス領域に格納され、命令実行前のIYレジスタで指定されるアドレス領域に格納されていた1バイトの値がAレジスタに格納される。

【2186】

10

20

30

40

50

(5) エクスチェンジ命令コード「EXQ r, (k)」(No. A19)

エクスチェンジ命令コード「EXQ r, (k)」は、Qレジスタ(図11で説明した拡張レジスタ)を用いてアドレス指定を行うエクスチェンジ命令コードである。エクスチェンジ命令コード「EXQ r, (k)」は、3バイトの命令コードである。なお、このエクスチェンジ命令コードのニーモニックで第3OPに規定されている「(k)」は、Qレジスタに格納された値と8ビットの整数値k(0~255)とで指定されるアドレス領域(指定メモリ)の下位側1バイトのアドレスを示す。なお、Qレジスタに格納されている値としては、メインRAM103内のアドレス領域又は内蔵レジスタエリア内のアドレス領域の上位側アドレスが指定されている必要がある。すなわち、メインRAM103内のアドレス領域を指定する場合には、QレジスタにはメインRAM103内のアドレス領域の上位側アドレスを示す「F0H」~「F3H」のいずれかの値が格納されている必要があり、内蔵レジスタエリア内のアドレス領域を指定する場合には、内蔵レジスタエリア内のアドレス領域の上位側アドレスを示す「FEH」が格納されている必要がある。

【2187】

エクスチェンジ命令コード「EXQ r, (k)」が実行されると、レジスタr(A、B、C、D、E、H又はLレジスタ)に格納されている値と、Qレジスタ(拡張レジスタ)に格納された値(指定メモリの上位側1バイトのアドレス)及び8ビットの整数値k(指定メモリの下位側1バイトのアドレス)で指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EXQ r, (k)」が実行されると、命令実行前のレジスタrに格納されていた値がQレジスタに格納された値及び8ビットの整数値kで指定されるアドレス領域に格納され、命令実行前のQレジスタに格納された値及び8ビットの整数値kで指定されるアドレス領域に格納されていた1バイトの値がレジスタrに格納される。

【2188】

図308に、ソースプログラム上で実際に使用され得るエクスチェンジ命令コード「EXQ r, (k)」の種類の一覧を示す。エクスチェンジ命令コード「EXQ r, (k)」としては、「EXQ A, (k)」(No. A19a)~「EXQ L, (k)」(No. A19g)の7種類の命令コードが設けられている。

【2189】

例えば、エクスチェンジ命令コード「EXQ A, (k)」(No. A19a)が実行されると、Aレジスタに格納されている値と、Qレジスタに格納された値及び8ビットの整数値kで指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EXQ A, (k)」が実行されると、命令実行前のAレジスタに格納されていた値がQレジスタに格納された値及び8ビットの整数値kで指定されるアドレス領域に格納され、命令実行前のQレジスタに格納された値及び8ビットの整数値kで指定されるアドレス領域に格納されていた1バイトの値がAレジスタに格納される。

【2190】

また、例えば、エクスチェンジ命令コード「EXQ C, (k)」(No. A19c)が実行されると、Cレジスタに格納されている値と、Qレジスタに格納された値及び8ビットの整数値kで指定されるアドレス領域に格納されている1バイトの値とが交換される。すなわち、エクスチェンジ命令コード「EXQ C, (k)」が実行されると、命令実行前のCレジスタに格納されていた値がQレジスタに格納された値及び8ビットの整数値kで指定されるアドレス領域に格納され、命令実行前のQレジスタに格納された値及び8ビットの整数値kで指定されるアドレス領域に格納されていた1バイトの値がCレジスタに格納される。

【2191】

(6) その他

図304~図308では、エクスチェンジ命令コードのニーモニックにおいて、第2OPに第1格納先を配置(規定)し、第3OPに第2格納先を配置(規定)する例を示し説明

10

20

30

40

50

したが、本実施形態の遊技機では、同じ処理動作を行う命令コードとして、第2OPに第2格納先を配置(規定)し、第3OPに第1格納先を配置(規定)したエクスチェンジ命令コードも設けられている。

【2192】

例えば、エクスチェンジ命令コード「EX ss, qq」(No. A6)に対しては、「EX qq, ss」というニーモニクのエクスチェンジ命令コードが設けられ、エクスチェンジ命令コード「EX r, (ss)」(No. A10)に対しては、「EX (ss), r」というニーモニクのエクスチェンジ命令コードが設けられる。また、例えば、エクスチェンジ命令コード「EX A, (mn)」(No. A11)～「EX L, (mn)」(No. A17)に対してはそれぞれ、「EX (mn), A」～「EX (mn), L」というニーモニクのエクスチェンジ命令コードが設けられる。さらに、例えば、エクスチェンジ命令コード「EX r, (qq)」(No. A18)に対しては、「EX (qq), r」というニーモニクのエクスチェンジ命令コードが設けられ、エクスチェンジ命令コード「EXQ r, (k)」(No. A19)に対しては、「EXQ (k), r」というニーモニクのエクスチェンジ命令コードが設けられる。

【2193】

図304～図308に示す各種エクスチェンジ命令コードのうち、交換対象となる2つ値の格納先の一方を1バイトのレジスタrとする、エクスチェンジ命令コード「EX r, (ss)」(No. A10)～「EXQ r, (k)」(No. A19)の説明では、当該1バイトのレジスタrが汎用レジスタ(A、B、C、D、E、H、Lレジスタ)である例を説明した。しかし、これらのエクスチェンジ命令コードでは、プログラム上、1バイトのレジスタrとして汎用レジスタ以外の1バイトのレジスタを規定することもできる。

【2194】

例えば、マイクロプロセッサ91には、各種回路(例えば、乱数回路110、タイマー回路113、シリアル通信回路114, 115等)が内蔵されるとともに(図9参照)、これらの各種回路を制御するための各種制御レジスタが内蔵され、当該各種制御レジスタは、FE00H～FEBFHの内蔵レジスタエリア(図12A参照)のアドレス領域に割り当てられている。

【2195】

乱数回路110には、例えば、16ビット乱数のハードラッチを行う際の各種設定を行うためのレジスタ(RL0ハードラッチ選択レジスタ(FE5BH、FE5CH))、8ビット乱数のハードラッチを行う際の各種設定を行うためのレジスタ(RSハードラッチ選択レジスタ(FE60H))、RL0ハードラッチ乱数値レジスタに乱数値が取り込まれたことを要因として外部割込みを発生させるか否かの設定を行うレジスタ(RL割込み制御レジスタ)、16ビット乱数RLの最大値を設定するレジスタ(RL0最大値設定レジスタ(下位側: FE67H、上位側: FE68H))、8ビット乱数RSの最大値を設定するレジスタ(RS0最大値設定レジスタ(FE6FH))、ソフトウェアで乱数の取り込みを行った際に乱数値が取り込まれたことを示すレジスタ(乱数ソフトラッチフラグレジスタ(FE75H))、乱数ソフトラッチフラグレジスタで取り込まれた乱数値が格納されるレジスタ(RL0ソフトラッチ乱数値レジスタ(下位側: FE76H、上位側: FE77H))、ハードラッチされた16ビットの乱数値が格納されるレジスタ(RL0ハードラッチ乱数値レジスタ(下位側: FE85H、上位側: FE86H))、ハードラッチされた8ビットの乱数値が格納されるレジスタ(RS0ハードラッチ乱数値レジスタ(FE99H))、等が設けられている。

【2196】

また、タイマー回路113(PTC)には、例えば、タイマー回路113内のカウンタを動作させるためのクロックを選択するレジスタ(PTC0プリスケアラレジスタ(FE01H))、タイマー回路113の割込み及びカウンタの動作条件を設定するレジスタ(PTC0制御レジスタ(FE02H))、タイマー回路113内のカウンタ値を設定するレジスタ(PTC0カウンタ設定レジスタ(FE03H))、タイマー回路113内のカウ

10

20

30

40

50

ンタ値を読み出すためのレジスタ（PTC0カウンタレジスタ（FE29H））、等が設けられている。

【2197】

さらに、シリアル通信回路114（SCU0）、115（SCU1）には、例えば、各SCU0のボーレートを設定するレジスタ（SCU0ボーレートレジスタ（下位側：FE12H、上位側：FE13H）、SCU1ボーレートレジスタ（下位側：FE17H、上位側：FE18H））、SCU0コマンドレジスタ（FE54H）、送受信の状態を確認するためのレジスタ（SCU0ステータスレジスタ（FE52H））、各SCUの送信状態を確認するためのレジスタ（SCU1コマンドステータスレジスタ（FE55H））、各SCUの送受信データ格納用レジスタ（SCU0データレジスタ（FE53H）、SCU1データレジスタ（FE56H））、等が設けられている。

10

【2198】

上記各種制御レジスタのうち、データの読み書きが可能な制御レジスタ（読書き用制御レジスタ）は、RL0ハードラッチ選択レジスタ（FE5BH、FE5CH）、RSハードラッチ選択レジスタ、RL0最大値設定レジスタ、RS0最大値設定レジスタ、PTC0プリスケアラレジスタ、PTC0カウンタ設定レジスタ、SCU0、1ボーレートレジスタ、及び、SCU0、1データレジスタである。また、データの読み出しのみが可能な制御レジスタ（読み出し専用制御レジスタ）は、乱数ソフトラッチフラグレジスタ、RL0ソフトラッチ乱数値レジスタ、RL0ハードラッチ乱数値レジスタ、RS0ハードラッチ乱数値レジスタ、PTC0カウンタレジスタ、及び、SCU0ステータスレジスタである。

20

【2199】

なお、マイクロプロセッサ91には、上述した各種制御レジスタだけでなく、その他の各種制御レジスタが内蔵される。具体的には、RL0ハードラッチ選択レジスタ以外にはRL1～RL3ハードラッチ選択レジスタが存在し、RL0最大値設定レジスタ以外にはRL1～RL3最大値設定レジスタが存在し、RS0最大値設定レジスタ以外にはRS1～RS3最大値設定レジスタが存在し、PTC0プリスケアラレジスタ以外にはPTC1、2プリスケアラレジスタが存在し、PTC0カウンタ設定レジスタ以外にはPTC1、2カウンタ設定レジスタが存在し、SCU0、1ボーレートレジスタ以外にはSCU2ボーレートレジスタが存在し、SCU0、1データレジスタ以外にはSCU2データレジスタが存在し、RL0ソフトラッチ乱数値レジスタ以外にはRL1～RL3ソフトラッチ乱数値レジスタが存在し、RL0ハードラッチ乱数値レジスタ以外にはRL1～RL3ハードラッチ乱数値レジスタが存在し、RS0ハードラッチ乱数値レジスタ以外にはRS1～RS3ハードラッチ乱数値レジスタが存在し、また、PTC0カウンタレジスタ以外にはPTC1、2カウンタレジスタが存在する。これらのその他の各種制御レジスタでは、各制御レジスタのアドレスが異なるだけで機能自体は上述した制御レジスタのそれと同じであるので、ここでは、これらのその他の各種制御レジスタの説明を省略する。

30

【2200】

なお、RL割込み制御レジスタ、SCU0コマンドレジスタ及びSCU1コマンドステータスレジスタでは、読書き可能なビットと、読み出し専用のビットとが混在して設けられている。このような制御レジスタにおいて、読書き可能なビット及び読み出し専用のビットのビット位置は、制御レジスタの種別に応じて異なる。また、PTC0制御レジスタでは、読書き可能なビットと、読み出し専用のビットと、書込み専用のビットとが混在して設けられている。

40

【2201】

そして、上述したエクステンジ命令コード「EX r, (ss)」(No. A10)～「EXQ r, (k)」(No. A19)において、第2格納先（「ss」又は「r」：第3OP）に読書き用制御レジスタのアドレスが設定又は規定されて命令が実行されると、読書き用制御レジスタに格納されている値と、第1格納先（第2OP）のレジスタに格納されている値とが交換される。

【2202】

50

なお、例えば、エクスチェンジ命令コード「EX r, (ss)」～「EXQ r, (k)」において、第2格納先（「ss」又は「r」：第3OP）に読み出し専用制御レジスタが規定されて命令が実行されると、読み出し専用制御レジスタに格納されている値は、第1格納先（第2OP）の格納領域に格納されるが、第1格納先に格納されている値は、読み出し専用制御レジスタに格納されない。また、例えば、エクスチェンジ命令コード「EX r, (ss)」～「EXQ r, (k)」において、第2格納先（「ss」又は「r」：第3OP）にRL割込み制御レジスタが規定されて命令が実行されると、第2格納先（第3OP）のRL割込み制御レジスタに格納されている値は、第1格納先（第2OP）に格納されるが、第1格納先（第2OP）に格納されている値のうち、RL割込み制御レジスタの読み書き可能なビットに対応するデータは、当該読み書き可能なビットに格納されるが、RL割込み制御レジスタの読み出し専用のビットに対応するデータは、当該読み出し専用のビットに格納されない。

【2203】

〔8ビットロード命令コード〕

図309～図312は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能な8ビットデータの読み出し命令コード（8ビットロード命令コード）の一覧表である。

【2204】

図309に記載の命令番号B1（「LD r, s」）～命令番号B15（「LDQ (k), n」）の8ビットロード命令コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図310～図312に記載の命令番号B16（「LDZ (mn)」）～命令番号B59（「LDF C, (mn), d」）の8ビットロード命令コードは、本実施形態で新たに設けられた命令コードである。なお、8ビットロード命令コードの実行時には、その命令コードの種別（命令番号）に関係なく、TZフラグの状態が実行結果に応じて適宜変化し（設定され）、その他の各フラグの状態は変化しない。

以下、主な、8ビットロード命令コードについて詳述する。

【2205】

（1）8ビットロード命令コード「LD r, s」（No. B1）

8ビットロード命令コード「LD r, s」は、1バイトの命令コードであり、オペランド（OP）として、命令「LD」（第1OP）、読み出し（ロード）先「r」（第2OP）及び読み出し元「s」（第3OP）がこの順で配置（規定）されて構成される。なお、この8ビットロード命令コードのニーモニックで第3OP（読み出し元）に規定されている「s」は、1バイトの汎用レジスタ（A、B、C、D、E、H、Lレジスタ：以下、「レジスタs」ともいう）を示す。

【2206】

8ビットロード命令コード「LD r, s」が実行されると、1バイトのレジスタsに格納されている値（8ビットデータ）が、1バイトのレジスタrにロード（セット）される。

【2207】

図313～図315に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LD r, s」の種類の一覧を示す。8ビットロード命令コード「LD r, s」としては、7種類の第2OP（A、B、C、D、E、H及びLレジスタ）のそれぞれに対して、6種類の第3OP（第2OP以外のレジスタ）が規定可能であり、合計で42種類の命令コードが設けられている。

【2208】

例えば、8ビットロード命令コード「LD A, B」（No. B1a）が実行されると、Bレジスタに格納されている値が、Aレジスタにロード（セット）される。例えば、8ビットロード命令コード「LD A, C」（No. B1b）が実行されると、Cレジスタに格納されている値が、Aレジスタにロード（セット）される。例えば、8ビットロード命令コード「LD A, D」（No. B1c）が実行されると、Dレジスタに格納されている値が、Aレジスタにロード（セット）される。例えば、8ビットロード命令コード「L

10

20

30

40

50

D A, E」(No. B1d)が実行されると、Eレジスタに格納されている値が、Aレジスタにロード(セット)される。例えば、8ビットロード命令コード「LD A, H」(No. B1e)が実行されると、Hレジスタに格納されている値が、Aレジスタにロード(セット)される。また、例えば、8ビットロード命令コード「LD A, L」(No. B1f)が実行されると、Lレジスタに格納されている値が、Aレジスタにロード(セット)される。

【2209】

(2) 8ビットロード命令コード「LD r, n」(No. B2)

8ビットロード命令コード「LD r, n」は、2バイトの命令コードであり、オペランド(OP)として、命令「LD」(第1OP)、読み出し(ロード)先「r」(第2OP)及び8ビットの整数値「n(0~255)」(第3OP)がこの順で配置(規定)されて構成される。

10

【2210】

8ビットロード命令コード「LD r, n」が実行されると、8ビットの整数値n(8ビットデータ)が、1バイトのレジスタr(A、B、C、D、E、H又はLレジスタ)にロード(セット)される。

【2211】

図316に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LD r, n」の種類の一覧を示す。8ビットロード命令コード「LD r, n」としては、「LD A, n」(No. B2a)~「LD L, n」(No. B2g)の7種類の命令コードが設けられている。

20

【2212】

例えば、8ビットロード命令コード「LD A, n」(No. B2a)が実行されると、8ビットの整数値nが、Aレジスタにロード(セット)される。また、例えば、8ビットロード命令コード「LD C, n」(No. B2c)が実行されると、8ビットの整数値nが、Cレジスタにロード(セット)される。

【2213】

(3) 8ビットロード命令コード「LD r, (qq)」(No. B5)

8ビットロード命令コード「LD r, (qq)」は、2バイトの命令コードであり、オペランド(OP)として、命令「LD」(第1OP)、読み出し(ロード)先「r」(第2OP)及び読み出し元「(qq)」(第3OP)がこの順で配置(規定)されて構成される。

30

【2214】

8ビットロード命令コード「LD r, (qq)」が実行されると、インデックス・レジスタqq(IX、IYレジスタ)に格納された値で指定されるアドレス領域(指定メモリ)に格納されている値(8ビットデータ)が、1バイトのレジスタr(A、B、C、D、E、H又はLレジスタ)にロード(セット)される。

【2215】

図317に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LD r, (qq)」の種類の一覧を示す。8ビットロード命令コード「LD r, (qq)」としては、7種類の第2OPのそれぞれに対して、2種類の第3OPが規定可能であり、合計で14種類の命令コードが設けられている。

40

【2216】

具体的には、第3OPとして「(IX)」(IXレジスタで指定されるアドレス領域)が規定される場合、第2OPには、A、B、C、D、E、H及びLレジスタのいずれかが規定される(No. B5a~B5g)。また、第3OPとして「(IY)」(IYレジスタで指定されるアドレス領域)が規定される場合、第2OPには、A、B、C、D、E、H及びLレジスタのいずれかが規定される(No. B5h~B5n)。

【2217】

例えば、8ビットロード命令コード「LD A, (IX)」(No. B5a)が実行され

50

ると、IXレジスタで指定されるアドレス領域（指定メモリ）に格納されている値が、Aレジスタにロード（セット）される。また、例えば、8ビットロード命令コード「LD A, (IY)」(No. B5h)が実行されると、IYレジスタで指定されるアドレス領域に格納されている値が、Aレジスタにロード（セット）される。

【2218】

(4) 8ビットロード命令コード「LD A, (DE)」(No. B6)、「LD r, (DE)」(No. B7)

8ビットロード命令コード「LD A, (DE)」は、1バイトの命令コードであり、オペランド(OP)として、命令「LD」(第1OP)、読み出し(ロード)先「A」(第2OP)及び読み出し元「(DE)」(第3OP)がこの順で配置(規定)されて構成される。また、8ビットロード命令コード「LD r, (DE)」は、2バイトの命令コードであり、オペランド(OP)として、命令「LD」(第1OP)、読み出し(ロード)先「r (Aレジスタ以外の汎用レジスタ)」(第2OP)及び読み出し元「(DE)」(第3OP)がこの順で配置(規定)されて構成される。すなわち、ロード元がDEレジスタで指定されるアドレス領域である場合、ロード先がAレジスタであるときには、8ビットロード命令コード「LD A, (DE)」は、1バイトの命令コードとなるが、ロード先がAレジスタ以外の汎用レジスタであるときには、その8ビットロード命令コード「LD r, (DE)」は、2バイトの命令コードとなる。

【2219】

8ビットロード命令コード「LD A, (DE)」が実行されると、DEレジスタに格納された値で指定されるアドレス領域に格納されている値(8ビットデータ)が、Aレジスタにロード（セット）される。また、8ビットロード命令コード「LD r, (DE)」が実行されると、DEレジスタに格納された値で指定されるアドレス領域に格納されている値(8ビットデータ)が、レジスタr (Aレジスタ以外の汎用レジスタ)にロード（セット）される。

【2220】

図318に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LD r, (DE)」の種類の一覧を示す。8ビットロード命令コード「LD r, (DE)」としては、「LD B, (DE)」(No. B7a)～「LD L, (DE)」(No. B7f)の6種類の命令コードが設けられている。

【2221】

例えば、8ビットロード命令コード「LD B, (DE)」(No. B7a)が実行されると、DEレジスタで指定されるアドレス領域に格納されている値が、Aレジスタにロード（セット）される。また、例えば、8ビットロード命令コード「LD H, (DE)」(No. B7e)が実行されると、DEレジスタで指定されるアドレス領域に格納されている値が、Hレジスタにロード（セット）される。

【2222】

(5) 8ビットロード命令コード「LD r, (HL)」(No. B8)

8ビットロード命令コード「LD r, (HL)」は、1バイトの命令コードであり、オペランド(OP)として、命令「LD」(第1OP)、読み出し(ロード)先「r」(第2OP)及び読み出し元「(HL)」(第3OP)がこの順で配置(規定)されて構成される。すなわち、ロード元がHLレジスタで指定されるアドレス領域である場合、ロード先がいずれの汎用レジスタであっても、その8ビットロード命令コード「LD r, (HL)」は、1バイトの命令コードとなる。

【2223】

8ビットロード命令コード「LD r, (HL)」が実行されると、HLレジスタに格納された値で指定されるアドレス領域に格納されている値(8ビットデータ)が、1バイトのレジスタrにロード（セット）される。

【2224】

図319に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LD

10

20

30

40

50

r, (HL)」の種類の一覧を示す。8ビットロード命令コード「LD r, (HL)」としては、「LD A, (HL)」(No. B8a)～「LD L, (HL)」(No. B8g)の7種類の命令コードが設けられている。

【2225】

例えば、8ビットロード命令コード「LD A, (HL)」(No. B8a)が実行されると、HLレジスタで指定されるアドレス領域に格納されている値が、Aレジスタにロード(セット)される。例えば、8ビットロード命令コード「LD D, (HL)」(No. B8d)が実行されると、HLレジスタで指定されるアドレス領域に格納されている値が、Dレジスタにロード(セット)される。また、例えば、8ビットロード命令コード「LD H, (HL)」(No. B8f)が実行されると、HLレジスタで指定されるアドレス領域に格納されている値が、Hレジスタにロード(セット)される。

【2226】

(6) 8ビットロード命令コード「LDQ A, (k)」(No. B9)、「LDQ r, (k)」(No. B10)

8ビットロード命令コード「LDQ A, (k)」及び「LDQ r, (k)」は、Qレジスタ(拡張レジスタ)を用いてアドレス指定を行う8ビットロード命令コードである。8ビットロード命令コード「LDQ A, (k)」は、2バイトの命令コードであり、オペランド(OP)として、命令「LDQ」(第1OP)、読み出し(ロード)先「A」(第2OP)及び読み出し元「(k)」(第3OP)がこの順で配置(規定)されて構成される。また、8ビットロード命令コード「LDQ r, (k)」は、3バイトの命令コードであり、オペランド(OP)として、命令「LDQ」(第1OP)、読み出し(ロード)先「r(Aレジスタ以外の汎用レジスタ)」(第2OP)及び読み出し元「(k)」(第3OP)がこの順で配置(規定)されて構成される。

【2227】

すなわち、ロード元がQレジスタに格納された値(指定メモリの上位側1バイトのアドレス)と8ビットの整数値k(指定メモリの下位側1バイトのアドレス)とで指定されるアドレス領域(指定メモリ)である場合、ロード先がAレジスタであるときには、その8ビットロード命令コード「LDQ A, (k)」は、2バイトの命令コードとなるが、ロード先がAレジスタ以外の汎用レジスタであるときには、その8ビットロード命令コード「LDQ r, (k)」は、3バイトの命令コードとなる。

【2228】

8ビットロード命令コード「LDQ A, (k)」が実行されると、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域に格納されている値(8ビットデータ)が、Aレジスタにロード(セット)される。また、8ビットロード命令コード「LDQ r, (k)」が実行されると、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域に格納されている値(8ビットデータ)が、レジスタr(Aレジスタ以外の汎用レジスタ)にロード(セット)される。

【2229】

図320に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LDQ r, (k)」の種類の一覧を示す。8ビットロード命令コード「LDQ r, (k)」としては、「LDQ B, (k)」(No. B10a)～「LDQ L, (k)」(No. B10f)の6種類の命令コードが設けられている。

【2230】

例えば、8ビットロード命令コード「LDQ B, (k)」(No. B10a)が実行されると、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域に格納されている値が、Bレジスタにロード(セット)される。また、例えば、8ビットロード命令コード「LDQ H, (k)」(No. B10e)が実行されると、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域に格納されている値が、Hレジスタにロード(セット)される。

【2231】

10

20

30

40

50

(7) 8ビットロード命令コード「LD (HL), r」(No. B11)

8ビットロード命令コード「LD (HL), r」は、1バイトの命令コードであり、オペランド(OP)として、命令「LD」(第1OP)、読み出し(ロード)先「(HL)」(第2OP)及び読み出し元「r」(第3OP)がこの順で配置(規定)されて構成される。なお、この命令コードの第3OPでは、読み出し(ロード)元となる1バイトのレジスタrに、H及びLレジスタ以外の汎用レジスタ(A、B、C、D又はEレジスタ)が規定される。

【2232】

8ビットロード命令コード「LD (HL), r」が実行されると、レジスタrに格納されている値(8ビットデータ)が、HLレジスタに格納された値で指定されるアドレス領域(指定メモリ)にロード(セット)される。

10

【2233】

図321に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LD (HL), r」の種類の一覧を示す。8ビットロード命令コード「LD (HL), r」としては、「LD (HL), A」(No. B11a)～「LD (HL), E」(No. B11e)の5種類の命令コードが設けられている。

【2234】

例えば、8ビットロード命令コード「LD (HL), A」(No. B11a)が実行されると、Aレジスタに格納されている値が、HLレジスタで指定されるアドレス領域にロード(セット)される。また、例えば、8ビットロード命令コード「LD (HL), C」(No. B11c)が実行されると、Cレジスタに格納されている値が、HLレジスタで指定されるアドレス領域にロード(セット)される。

20

【2235】

(8) 8ビットロード命令コード「LD (qq), r」(No. B12)

8ビットロード命令コード「LD (qq), r」は、2バイトの命令コードであり、オペランド(OP)として、命令「LD」(第1OP)、読み出し(ロード)先「(qq)」(第2OP)及び読み出し元「r」(第3OP)がこの順で配置(規定)されて構成される。

【2236】

8ビットロード命令コード「LD (qq), r」が実行されると、1バイトのレジスタr(A、B、C、D、E、H及びLレジスタ)に格納されている値(8ビットデータ)が、インデックス・レジスタqqに格納された値で指定されるアドレス領域(指定メモリ)にロード(セット)される。

30

【2237】

図322に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LD (qq), r」の種類の一覧を示す。8ビットロード命令コード「LD (qq), r」としては、2種類の第2OPのそれぞれに対して、7種類の第3OPが規定可能であり、合計で14種類の命令コードが設けられている。

【2238】

具体的には、第2OPとして「(IX)」(IXレジスタで指定されるアドレス領域)が規定される場合、第3OPには、A、B、C、D、E、H及びLレジスタのいずれかが規定される(No. B12a～B12g)。また、第2OPとして「(IY)」(IYレジスタで指定されるアドレス領域)が規定される場合、第3OPには、A、B、C、D、E、H及びLレジスタのいずれかが規定される(No. B12h～B12n)。

40

【2239】

例えば、8ビットロード命令コード「LD (IX), A」(No. B12a)が実行されると、Aレジスタに格納されている値が、IXレジスタで指定されるアドレス領域にロード(セット)される。また、例えば、8ビットロード命令コード「LD (IY), A」(No. B12h)が実行されると、Aレジスタに格納されている値が、IYレジスタで指定されるアドレス領域にロード(セット)される。

50

【2240】

(9) 8ビットロード命令コード「LDQ (k), A」(No. B13)、「LDQ (k), r」(No. B14)

8ビットロード命令コード「LDQ (k), A」及び「LDQ (k), r」はそれぞれ、Qレジスタ(拡張レジスタ)を用いてアドレス指定を行う8ビットロード命令コードである。8ビットロード命令コード「LDQ (k), A」は、2バイトの命令コードであり、オペランド(OP)として、命令「LDQ」(第1OP)、読み出し(ロード)先「(k)」(第2OP)及び読み出し元「A」(第3OP)がこの順で配置(規定)されて構成される。また、8ビットロード命令コード「LDQ (k), r」は、3バイトの命令コードであり、オペランド(OP)として、命令「LDQ」(第1OP)、読み出し(ロード)先「(k)」(第2OP)及び読み出し元「r (Aレジスタ以外の汎用レジスタ)」(第3OP)がこの順で配置(規定)されて構成される。

10

【2241】

すなわち、ロード先がQレジスタに格納された値(指定メモリの上位側1バイトのアドレス)と8ビットの整数値k(指定メモリの下位側1バイトのアドレス)とで指定されるアドレス領域(指定メモリ)である場合、ロード元がAレジスタであるときには、その8ビットロード命令コード「LDQ A, (k)」は、2バイトの命令コードとなるが、ロード元がAレジスタ以外の汎用レジスタであるときには、その8ビットロード命令コード「LDQ r, (k)」は、3バイトの命令コードとなる。

【2242】

8ビットロード命令コード「LDQ (k), A」が実行されると、Aレジスタに格納された値(8ビットデータ)が、Qレジスタに格納されている値と8ビットの整数値kとで指定されるアドレス領域(指定メモリ)にロード(セット)される。また、8ビットロード命令コード「LDQ (k), r」が実行されると、レジスタr (Aレジスタ以外の汎用レジスタ)に格納されている値(8ビットデータ)が、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域(指定メモリ)にロード(セット)される。

20

【2243】

図323に、ソースプログラム上で実際に使用され得る8ビットロード命令コード「LDQ (k), r」の種類の一覧を示す。8ビットロード命令コード「LDQ (k), r」としては、「LDQ (k), B」(No. B14a)～「LDQ (k), L」(No. B14f)の6種類の命令コードが設けられている。

30

【2244】

例えば、8ビットロード命令コード「LDQ (k), B」(No. B14a)が実行されると、Bレジスタに格納されている値が、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域にロード(セット)される。また、例えば、8ビットロード命令コード「LDQ (k), H」(No. B14e)が実行されると、Hレジスタに格納されている値が、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域にロード(セット)される。

【2245】

(10) 8ビットロード命令コード「LDZ (mn)」(No. B16)

8ビットロード命令コード「LDZ (mn)」は、4バイトの命令コードであり、オペランド(OP)として、命令「LDZ」(第1OP)及び読み出し(ロード)先「(mn)」(第2OP: 格納先を示す情報)がこの順で配置(規定)されて構成される。

40

【2246】

8ビットロード命令コード「LDZ (mn)」が実行されると、TZフラグ(所定のフラグ、第2のゼロフラグ)がセットされている場合(TZフラグの値が「1」である場合)、16ビットの整数値mnで指定されるアドレス領域(指定メモリ)に「0」(8ビットデータ)がセット(ロード)され、TZフラグがリセットされている場合(TZフラグの値が「0」である場合)、16ビットの整数値mnで指定されるアドレス領域に「1」(8ビットデータ)がセット(ロード)される。

50

【2247】

(11) 8ビットロード命令コード「LDZ (DE)」(No. B17)、「LDZ (HL)」(No. B18)

8ビットロード命令コード「LDZ (DE)」及び「LDZ (HL)」はそれぞれ、2バイトの命令コードである。8ビットロード命令コード「LDZ (DE)」は、オペランド(OP)として、命令「LDZ」(第1OP)及び読み出し(ロード)先「(DE)」(第2OP)がこの順で配置(規定)されて構成され、8ビットロード命令コード「LDZ (HL)」は、オペランド(OP)として、命令「LDZ」(第1OP)及び読み出し(ロード)先「(HL)」(第2OP)がこの順で配置(規定)されて構成される。

【2248】

8ビットロード命令コード「LDZ (DE)」が実行されると、TZフラグがセットされている場合(TZフラグの値が「1」である場合)、DEレジスタに格納された値で指定されるアドレス領域(指定メモリ)に「0」(8ビットデータ)がセット(ロード)され、TZフラグがリセットされている場合(TZフラグの値が「0」である場合)、DEレジスタに格納された値で指定されるアドレス領域に「1」(8ビットデータ)がセット(ロード)される。また、8ビットロード命令コード「LDZ (HL)」が実行されると、TZフラグがセットされている場合(TZフラグの値が「1」である場合)、HLレジスタに格納された値で指定されるアドレス領域(指定メモリ)に「0」(8ビットデータ)がセット(ロード)され、TZフラグがリセットされている場合(TZフラグの値が「0」である場合)、HLレジスタに格納された値で指定されるアドレス領域に「1」(8ビットデータ)がセット(ロード)される。

【2249】

(12) 8ビットロード命令コード「LDZ (IX)」(No. B19)、「LDZ (IY)」(No. B20)

8ビットロード命令コード「LDZ (IX)」及び「LDZ (IY)」はそれぞれ、2バイトの命令コードである。8ビットロード命令コード「LDZ (IX)」は、オペランド(OP)として、命令「LDZ」(第1OP)及び読み出し(ロード)先「(IX)」(第2OP)がこの順で配置(規定)されて構成され、8ビットロード命令コード「LDZ (IY)」は、オペランド(OP)として、命令「LDZ」(第1OP)及び読み出し(ロード)先「(IY)」(第2OP)がこの順で配置(規定)されて構成される。

【2250】

8ビットロード命令コード「LDZ (IX)」が実行されると、TZフラグがセットされている場合(TZフラグの値が「1」である場合)、IXレジスタに格納された値で指定されるアドレス領域(指定メモリ)に「0」(8ビットデータ)がセット(ロード)され、TZフラグがリセットされている場合(TZフラグの値が「0」である場合)、IXレジスタに格納された値で指定されるアドレス領域に「1」(8ビットデータ)がセット(ロード)される。また、8ビットロード命令コード「LDZ (IY)」が実行されると、TZフラグがセットされている場合(TZフラグの値が「1」である場合)、IYレジスタに格納された値で指定されるアドレス領域(指定メモリ)に「0」(8ビットデータ)がセット(ロード)され、TZフラグがリセットされている場合(TZフラグの値が「0」である場合)、IYレジスタに格納された値で指定されるアドレス領域に「1」(8ビットデータ)がセット(ロード)される。

【2251】

(13) 8ビットロード命令コード「LDQZ (k)」(No. B21)

8ビットロード命令コード「LDQZ (k)」は、Qレジスタ(拡張レジスタ)を用いてアドレス指定を行う8ビットロード命令コードである。8ビットロード命令コード「LDQZ (k)」は、3バイトの命令コードであり、オペランド(OP)として、命令「LDQZ」(第1OP)及び読み出し(ロード)先「(k)」(第2OP)がこの順で配置(規定)されて構成される。

【2252】

8ビットロード命令コード「LDQZ (k)」が実行されると、TZフラグがセットされている場合（TZフラグの値が「1」である場合）、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と8ビットの整数値k（指定メモリの下位側1バイトのアドレス）とで指定されるアドレス領域（指定メモリ）に「0」（8ビットデータ）がセット（ロード）され、TZフラグがリセットされている場合（TZフラグの値が「0」である場合）、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域に「1」（8ビットデータ）がセット（ロード）される。

【2253】

(14) 8ビットロード命令コード「LDC (mn)」(No. B22)

8ビットロード命令コード「LDC (mn)」は、4バイトの命令コードであり、オペランド(OP)として、命令「LDC」(第1OP)及び読み出し(ロード)先「(mn)」(第2OP)がこの順で配置(規定)されて構成される。

10

【2254】

8ビットロード命令コード「LDC (mn)」が実行されると、Cフラグ(CYフラグ)がセットされている場合（Cフラグの値が「1」である場合）、16ビットの整数値mnで指定されるアドレス領域（指定メモリ）に「1」（8ビットデータ）がセット（ロード）され、Cフラグがリセットされている場合（Cフラグの値が「0」である場合）、16ビットの整数値mnで指定されるアドレス領域に「0」（8ビットデータ）がセット（ロード）される。

【2255】

20

(15) 8ビットロード命令コード「LDC (DE)」(No. B23)、「LDC (HL)」(No. B24)

8ビットロード命令コード「LDC (DE)」及び「LDC (HL)」はそれぞれ、2バイトの命令コードである。8ビットロード命令コード「LDC (DE)」は、オペランド(OP)として、命令「LDC」(第1OP)及び読み出し(ロード)先「(DE)」(第2OP)がこの順で配置(規定)されて構成され、8ビットロード命令コード「LDC (HL)」は、オペランド(OP)として、命令「LDC」(第1OP)及び読み出し(ロード)先「(HL)」(第2OP)がこの順で配置(規定)されて構成される。

【2256】

8ビットロード命令コード「LDC (DE)」が実行されると、Cフラグ(CYフラグ)がセットされている場合（Cフラグの値が「1」である場合）、DEレジスタに格納された値で指定されるアドレス領域（指定メモリ）に「1」（8ビットデータ）がセット（ロード）され、Cフラグがリセットされている場合（Cフラグの値が「0」である場合）、DEレジスタに格納された値で指定されるアドレス領域に「0」（8ビットデータ）がセット（ロード）される。また、8ビットロード命令コード「LDC (HL)」が実行されると、Cフラグがセットされている場合（Cフラグの値が「1」である場合）、HLレジスタに格納された値で指定されるアドレス領域（指定メモリ）に「1」（8ビットデータ）がセット（ロード）され、Cフラグがリセットされている場合（Cフラグの値が「0」である場合）、HLレジスタに格納された値で指定されるアドレス領域に「0」（8ビットデータ）がセット（ロード）される。

30

【2257】

(16) 8ビットロード命令コード「LDC (IX)」(No. B25)、「LDC (IY)」(No. B26)

8ビットロード命令コード「LDC (IX)」及び「LDC (IY)」はそれぞれ、2バイトの命令コードである。8ビットロード命令コード「LDC (IX)」は、オペランド(OP)として、命令「LDC」(第1OP)及び読み出し(ロード)先「(IX)」(第2OP)がこの順で配置(規定)されて構成され、8ビットロード命令コード「LDC (IY)」は、オペランド(OP)として、命令「LDC」(第1OP)及び読み出し(ロード)先「(IY)」(第2OP)がこの順で配置(規定)されて構成される。

40

【2258】

50

8ビットロード命令コード「LD C (IX)」が実行されると、Cフラグ(CYフラグ)がセットされている場合(Cフラグの値が「1」である場合)、IXレジスタに格納された値で指定されるアドレス領域(指定メモリ)に「1」(8ビットデータ)がセット(ロード)され、Cフラグがリセットされている場合(Cフラグの値が「0」である場合)、IXレジスタに格納された値で指定されるアドレス領域に「0」(8ビットデータ)がセット(ロード)される。また、8ビットロード命令コード「LD C (IY)」が実行されると、Cフラグ(CYフラグ)がセットされている場合(Cフラグの値が「1」である場合)、IYレジスタに格納された値で指定されるアドレス領域(指定メモリ)に「1」(8ビットデータ)がセット(ロード)され、Cフラグがリセットされている場合(Cフラグの値が「0」である場合)、IYレジスタに格納された値で指定されるアドレス領域に「0」(8ビットデータ)がセット(ロード)される。

10

【2259】

(17) 8ビットロード命令コード「LDQC (k)」(No. B27)

8ビットロード命令コード「LDQC (k)」は、Qレジスタ(拡張レジスタ)を用いてアドレス指定を行う8ビットロード命令コードである。8ビットロード命令コード「LDQC (k)」は、3バイトの命令コードであり、オペランド(OP)として、命令「LDQC」(第1OP)及び読み出し(ロード)先「(k)」(第2OP)がこの順で配置(規定)されて構成される。

【2260】

8ビットロード命令コード「LDQC (k)」が実行されると、Cフラグ(CYフラグ)がセットされている場合(Cフラグの値が「1」である場合)、Qレジスタに格納された値(指定メモリの上位側1バイトのアドレス)と8ビットの整数値k(指定メモリの下位側1バイトのアドレス)とで指定されるアドレス領域(指定メモリ)に「1」(8ビットデータ)がセット(ロード)され、Cフラグがリセットされている場合(Cフラグの値が「0」である場合)、Qレジスタに格納された値と8ビットの整数値kとで指定されるアドレス領域に「0」(8ビットデータ)がセット(ロード)される。

20

【2261】

(18) 8ビットロード命令コード「LDF NZ, A, n」(No. B28)、「LDF Z, A, n」(No. B29)、「LDF NC, A, n」(No. B30)、「LDF C, A, n」(No. B31)

30

8ビットロード命令コード「LDF NZ, A, n」、「LDF Z, A, n」、「LDF NC, A, n」及び「LDF C, A, n」はそれぞれ、3バイトの命令コードであり、オペランド(OP)として、命令「LDF」(第1OP)、フラグ条件(NZ、Z、NC及びC:第2OP)、読み出し(ロード)先「A」(第3OP)及び8ビットの整数値「n」(第4OP)がこの順で配置(規定)されて構成される。

【2262】

なお、フラグ条件は、命令実行時に読み出し動作が行われる所定フラグ(Zフラグ、Cフラグ)の状態を示すパラメータ(オペランド)である。具体的には、フラグ条件「NZ」はZフラグがリセットされている場合(Zフラグの値が「0」である場合)であり、フラグ条件「Z」はZフラグがセットされている場合(Zフラグの値が「1」である場合)である。また、フラグ条件「NC」はCフラグ(CYフラグ)がリセットされている場合(Cフラグの値が「0」である場合)であり、フラグ条件「C」はCフラグ(CYフラグ)がセットされている場合(Cフラグの値が「1」である場合)である。

40

【2263】

8ビットロード命令コード「LDF フラグ条件, A, n」が実行されると、フラグ条件が成立している場合には、8ビットの整数値n(8ビットデータ)が、Aレジスタにロード(セット)される。

【2264】

例えば、「LDF NZ, A, n」が実行されると、Zフラグがリセットされている場合(Zフラグの値が「0」である場合)に、8ビットの整数値n(イミディエイトデータ)

50

がAレジスタにロード（セット）される。例えば、「LDF Z, A, n」が実行されると、Zフラグがセットされている場合（Zフラグの値が「1」である場合）に、8ビットの整数値nがAレジスタにロード（セット）される。例えば、「LDF NC, A, n」が実行されると、Cフラグ（CYフラグ）がリセットされている場合（Cフラグの値が「0」である場合）に、8ビットの整数値nがAレジスタにロード（セット）される。また、例えば、「LDF C, A, n」が実行されると、Cフラグがセットされている場合（Cフラグの値が「1」である場合）に、8ビットの整数値nがAレジスタにロード（セット）される。

【2265】

(19) 8ビットロード命令コード「LDF NZ, B, n」(No. B32)、「LDF Z, B, n」(No. B33)、「LDF NC, B, n」(No. B34)、「LDF C, B, n」(No. B35)

10

8ビットロード命令コード「LDF NZ, B, n」、「LDF Z, B, n」、「LDF NC, B, n」及び「LDF C, B, n」はそれぞれ、3バイトの命令コードであり、オペランド（OP）として、命令「LDF」（第1OP）、フラグ条件（第2OP）、読み出し（ロード）先「B」（第3OP）及び8ビットの整数値「n」（第4OP）がこの順で配置（規定）されて構成される。

【2266】

なお、8ビットロード命令コード「LDF フラグ条件, B, n」の処理動作は、8ビットの整数値nの読み出し（ロード）先がBレジスタになること以外は、上記（18）で説明した8ビットロード命令コード「LDF フラグ条件, A, n」の処理動作と同様である。

20

【2267】

(20) 8ビットロード命令コード「LDF NZ, C, n」(No. B36)、「LDF Z, C, n」(No. B37)、「LDF NC, C, n」(No. B38)、「LDF C, C, n」(No. B39)

8ビットロード命令コード「LDF NZ, C, n」、「LDF Z, C, n」、「LDF NC, C, n」及び「LDF C, C, n」はそれぞれ、3バイトの命令コードであり、オペランド（OP）として、命令「LDF」（第1OP）、フラグ条件（第2OP）、読み出し（ロード）先「C」（第3OP）及び8ビットの整数値「n」（第4OP）がこの順で配置（規定）されて構成される。

30

【2268】

なお、8ビットロード命令コード「LDF フラグ条件, C, n」の処理動作は、8ビットデータnの読み出し（ロード）先がCレジスタになること以外は、上記（18）で説明した8ビットロード命令コード「LDF フラグ条件, A, n」の処理動作と同様である。

【2269】

(21) 8ビットロード命令コード「LDF NZ, D, n」(No. B40)、「LDF Z, D, n」(No. B41)、「LDF NC, D, n」(No. B42)、「LDF C, D, n」(No. B43)

8ビットロード命令コード「LDF NZ, D, n」、「LDF Z, D, n」、「LDF NC, D, n」及び「LDF C, D, n」はそれぞれ、3バイトの命令コードであり、オペランド（OP）として、命令「LDF」（第1OP）、フラグ条件（第2OP）、読み出し（ロード）先「D」（第3OP）及び8ビットの整数値「n」（第4OP）がこの順で配置（規定）されて構成される。

40

【2270】

なお、8ビットロード命令コード「LDF フラグ条件, D, n」の処理動作は、8ビットの整数値nの読み出し（ロード）先がDレジスタになること以外は、上記（18）で説明した8ビットロード命令コード「LDF フラグ条件, A, n」の処理動作と同様である。

【2271】

50

(22) 8ビットロード命令コード「LDF NZ, E, n」(No. B44)、「LDF Z, E, n」(No. B45)、「LDF NC, E, n」(No. B46)、「LDF C, E, n」(No. B47)

8ビットロード命令コード「LDF NZ, E, n」、「LDF Z, E, n」、「LDF NC, E, n」及び「LDF C, E, n」はそれぞれ、3バイトの命令コードであり、オペランド(OP)として、命令「LDF」(第1OP)、フラグ条件(第2OP)、読み出し(ロード)先「E」(第3OP)及び8ビットの整数値「n」(第4OP)がこの順で配置(規定)されて構成される。

【2272】

なお、8ビットロード命令コード「LDF フラグ条件, E, n」の処理動作は、8ビットの整数値nの読み出し(ロード)先がEレジスタになること以外は、上記(18)で説明した8ビットロード命令コード「LDF フラグ条件, A, n」の処理動作と同様である。

10

【2273】

(23) 8ビットロード命令コード「LDF NZ, H, n」(No. B48)、「LDF Z, H, n」(No. B49)、「LDF NC, H, n」(No. B50)、「LDF C, H, n」(No. B51)

8ビットロード命令コード「LDF NZ, H, n」、「LDF Z, H, n」、「LDF NC, H, n」及び「LDF C, H, n」はそれぞれ、3バイトの命令コードであり、オペランド(OP)として、命令「LDF」(第1OP)、フラグ条件(第2OP)、読み出し(ロード)先「H」(第3OP)及び8ビットの整数値「n」(第4OP)がこの順で配置(規定)されて構成される。

20

【2274】

なお、8ビットロード命令コード「LDF フラグ条件, H, n」の処理動作は、8ビットの整数値nの読み出し(ロード)先がHレジスタになること以外は、上記(18)で説明した8ビットロード命令コード「LDF フラグ条件, A, n」の処理動作と同様である。

【2275】

(24) 8ビットロード命令コード「LDF NZ, L, n」(No. B52)、「LDF Z, L, n」(No. B53)、「LDF NC, L, n」(No. B54)、「LDF C, L, n」(No. B55)

8ビットロード命令コード「LDF NZ, L, n」、「LDF Z, L, n」、「LDF NC, L, n」及び「LDF C, L, n」はそれぞれ、3バイトの命令コードであり、オペランド(OP)として、命令「LDF」(第1OP)、フラグ条件(第2OP)、読み出し(ロード)先「L」(第3OP)及び8ビットの整数値「n」(第4OP)がこの順で配置(規定)されて構成される。

30

【2276】

なお、8ビットロード命令コード「LDF フラグ条件, L, n」の処理動作は、8ビットの整数値nの読み出し(ロード)先がLレジスタになること以外は、上記(18)で説明した8ビットロード命令コード「LDF フラグ条件, A, n」の処理動作と同様である。

40

【2277】

(25) 8ビットロード命令コード「LDF NZ, (mn), d」(No. B56)、「LDF Z, (mn), d」(No. B57)、「LDF NC, (mn), d」(No. B58)、「LDF C, (mn), d」(No. B59)

8ビットロード命令コード「LDF NZ, (mn), d」、「LDF Z, (mn), d」、「LDF NC, (mn), d」及び「LDF C, (mn), d」はそれぞれ、5バイトの命令コードであり、オペランド(OP)として、命令「LDF」(第1OP)、フラグ条件(第2OP)、読み出し(ロード)先「(mn)」及び8ビットの整数値「d」(第4OP)がこの順で配置(規定)されて構成される。

50

【2278】

8ビットロード命令コード「LDF フラグ条件, (mn), d」が実行されると、フラグ条件が成立している場合には、8ビットの整数値d（8ビットデータ）が、16ビットの整数値mnで指定されるアドレス領域（指定メモリ）にロード（セット）される。

【2279】

例えば、「LDF NZ, (mn), d」が実行されると、Zフラグがリセットされている場合（Zフラグの値が「0」である場合）、8ビットの整数値dが、16ビットの整数値mnで指定されるアドレス領域にロード（セット）される。例えば、「LDF Z, (mn), d」が実行されると、Zフラグがセットされている場合（Zフラグの値が「1」である場合）、8ビットの整数値dが、16ビットの整数値mnで指定されるアドレス領域にロード（セット）される。例えば、「LDF NC, (mn), d」が実行されると、Cフラグ（CYフラグ）がリセットされている場合（Cフラグの値が「0」である場合）、8ビットの整数値dが、16ビットの整数値mnで指定されるアドレス領域にロード（セット）される。また、例えば、「LDF C, (mn), d」が実行されると、Cフラグがセットされている場合（Cフラグの値が「1」である場合）、8ビットの整数値dが、16ビットの整数値mnで指定されるアドレス領域にロード（セット）される。

【2280】

【16ビットロード命令コード】

図324及び図325は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能な16ビットデータの読み出し命令コード（16ビットロード命令コード）の一覧表である。

【2281】

図324に記載の命令番号C1（「LD BC, mn」）～命令番号C12（「LDQ (k), HL」）の16ビットロード命令コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図325に記載の命令番号C13（「LDF NZ, BC, mn」）～命令番号C24（「LDF C, HL, mn」）の16ビットロード命令コードは、本実施形態で新たに設けられた命令コードである。なお、16ビットロード命令コードの実行時には、その命令コードの種別（命令番号）に関係なく、TZフラグの状態が命令の実行結果に応じて適宜変化し（設定され）、その他の各フラグの状態は変化しない。

以下、本実施形態で新たに設けられた16ビットロード命令コードについて詳述する。

【2282】

(1) 16ビットロード命令コード「LDF NZ, BC, mn」（No. C13）、「LDF Z, BC, mn」（No. C14）、「LDF NC, BC, mn」（No. C15）、「LDF C, BC, mn」（No. C16）

16ビットロード命令コード「LDF NZ, BC, mn」、「LDF Z, BC, mn」、「LDF NC, BC, mn」及び「LDF C, BC, mn」はそれぞれ、4バイトの命令コードであり、オペランド（OP）として、命令「LDF」（第1OP）、フラグ条件（第2OP）、読み出し（ロード）先「BC」（第3OP）及び16ビットの整数値「mn」（第4OP）がこの順で配置（規定）されて構成される。

【2283】

16ビットロード命令コード「LDF フラグ条件, BC, mn」が実行されると、フラグ条件が成立している場合には、16ビットの整数値mn（16ビットデータmn）が、BCレジスタにロード（セット）される。この際、16ビットの整数値mn中の上位側8ビットの整数値m（8ビットデータm）がBレジスタにロード（セット）され、下位側8ビットの整数値n（8ビットデータn）が、Cレジスタにロード（セット）される。

【2284】

例えば、「LDF NZ, BC, mn」が実行されると、Zフラグがリセットされている場合（Zフラグの値が「0」である場合）に、16ビットの整数値mnがBCレジスタにロード（セット）される（上位側8ビットの整数値mがBレジスタにセットされ、下位側

10

20

30

40

50

8ビットの整数値 n が C レジスタにセットされる)。例えば、「LDF Z, BC, mn」が実行されると、Zフラグがセットされている場合（Zフラグの値が「1」である場合）に、16ビットの整数値 mn が BC レジスタにロード（セット）される。例えば、「LDF NC, BC, mn」が実行されると、Cフラグ（CYフラグ）がリセットされている場合（Cフラグの値が「0」である場合）に、16ビットの整数値 mn が BC レジスタにロード（セット）される。また、例えば、「LDF C, BC, mn」が実行されると、Cフラグがセットされている場合（Cフラグの値が「1」である場合）に、16ビットの整数値 mn が BC レジスタにロード（セット）される。

【2285】

(2) 16ビットロード命令コード「LDF NZ, DE, mn」(No. C17)、「LDF Z, DE, mn」(No. C18)、「LDF NC, DE, mn」(No. C19)、「LDF C, DE, mn」(No. C20)

10

16ビットロード命令コード「LDF NZ, DE, mn」、「LDF Z, DE, mn」、「LDF NC, DE, mn」及び「LDF C, DE, mn」はそれぞれ、4バイトの命令コードであり、オペランド(OP)として、命令「LDF」(第1OP)、フラグ条件(第2OP)、読み出し(ロード)先「DE」(第3OP)及び16ビットの整数値「mn」(第4OP)がこの順で配置(規定)されて構成される。

【2286】

16ビットロード命令コード「LDF フラグ条件, DE, mn」が実行されると、フラグ条件が成立している場合には、16ビットの整数値 mn (16ビットデータ mn) が、DEレジスタにロード（セット）される。この際、16ビットの整数値 mn 中の上位側8ビットの整数値 m (8ビットデータ m) が D レジスタにロード（セット）され、下位側8ビットの整数値 n (8ビットデータ n) が、Eレジスタにロード（セット）される。

20

【2287】

例えば、「LDF NZ, DE, mn」が実行されると、Zフラグがリセットされている場合（Zフラグの値が「0」である場合）に、16ビットの整数値 mn が DE レジスタにロード（セット）される（上位側8ビットの整数値 m が D レジスタにセットされ、下位側8ビットの整数値 n が E レジスタにセットされる）。例えば、「LDF Z, DE, mn」が実行されると、Zフラグがセットされている場合（Zフラグの値が「1」である場合）に、16ビットの整数値 mn が DE レジスタにロード（セット）される。例えば、「LDF NC, DE, mn」が実行されると、Cフラグ（CYフラグ）がリセットされている場合（Cフラグの値が「0」である場合）に、16ビットの整数値 mn が DE レジスタにロード（セット）される。また、例えば、「LDF C, DE, mn」が実行されると、Cフラグがセットされている場合（Cフラグの値が「1」である場合）に、16ビットの整数値 mn が DE レジスタにロード（セット）される。

30

【2288】

(3) 16ビットロード命令コード「LDF NZ, HL, mn」(No. C21)、「LDF Z, HL, mn」(No. C22)、「LDF NC, HL, mn」(No. C23)、「LDF C, HL, mn」(No. C24)

16ビットロード命令コード「LDF NZ, HL, mn」、「LDF Z, HL, mn」、「LDF NC, HL, mn」及び「LDF C, HL, mn」はそれぞれ、4バイトの命令コードであり、オペランド(OP)として、命令「LDF」(第1OP)、フラグ条件(第2OP)、読み出し(ロード)先「HL」(第3OP)及び16ビットの整数値「mn」(第4OP)がこの順で配置(規定)されて構成される。

40

【2289】

16ビットロード命令コード「LDF フラグ条件, HL, mn」が実行されると、フラグ条件が成立している場合には、16ビットの整数値 mn (16ビットデータ mn) が、HLレジスタにロード（セット）される。この際、16ビットの整数値 mn 中の上位側8ビットの整数値 m (8ビットデータ m) が、Hレジスタにロード（セット）され、下位側8ビットの整数値 n (8ビットデータ n) が、Lレジスタにロード（セット）される。

50

【2290】

例えば、「LDF NZ, HL, mn」が実行されると、Zフラグがリセットされている場合（Zフラグの値が「0」である場合）に、16ビットの整数値mnがHLレジスタにロード（セット）される（上位側8ビットの整数値mがHレジスタにセットされ、下位側8ビットの整数値nがLレジスタにセットされる）。例えば、「LDF Z, HL, mn」が実行されると、Zフラグがセットされている場合（Zフラグの値が「1」である場合）に、16ビットの整数値mnがHLレジスタにロード（セット）される。例えば、「LDF NC, HL, mn」が実行されると、Cフラグ（CYフラグ）がリセットされている場合（Cフラグの値が「0」である場合）に、16ビットの整数値mnがHLレジスタにロード（セット）される。また、例えば、「LDF C, HL, mn」が実行されると、Cフラグがセットされている場合（Cフラグの値が「1」である場合）に、16ビットの整数値mnがHLレジスタにロード（セット）される。

10

【2291】

【ジャンプ命令コード】

図326及び図327は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能なジャンプ命令コードの一覧表である。なお、ジャンプ命令コードは、プログラム上において、処理の実行番地（アドレス）を、命令コードで指定する番地にジャンプ（移動）させるための命令コードである。

【2292】

図326に記載の命令番号D1（「JP (HL)」）～命令番号D5（「JR cc, e」）のジャンプ命令コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図326及び図327に記載の命令番号D6（「TJP」）～命令番号D19（「TJPWL」）のジャンプ命令コードは、本実施形態で新たに設けられた命令コードである。なお、ジャンプ命令コードの実行時には、その命令コードの種別（命令番号）に関係なく、各フラグの状態は変化しない。以下、主な、ジャンプ命令コードについて詳述する。

20

【2293】

(1) ジャンプ命令コード「JR cc, e」（No. D5）

ジャンプ命令コード「JR cc, e」は、2バイトの命令コードであり、オペランド（OP）として、命令「JR」（第1OP）、フラグ条件「cc」（第2OP）及びジャンプ値「e」（第3OP）がこの順で配置（規定）されて構成される。

30

【2294】

なお、第2OPで規定されるフラグ条件「cc」には、Zフラグ、Cフラグ及びTZフラグのいずれか（ccフラグ）の状態が規定される。また、第3OPで規定される、ジャンプ値「e」は、当該ジャンプ命令コード（現オペレーションコード）が規定されている番地（PCレジスタ（プログラム・カウンタPC：図11参照）に格納された値（アドレス値））からジャンプ先番地までの値であり、このジャンプ値eは、8ビットの符号付整数値である。なお、符号付整数値は、8ビットであれば「-127」～「127」の範囲の値となり、16ビットでは「-32767」～「32767」の範囲の値となる。この符号付整数値では、データの最上位ビット（8ビットの値ではビット7、16ビットの値ではビット15）が正又は負の符号を意味し、最上位ビットの値が「1」であれば負（-）を示し、最上位ビットの値が「0」であれば正（+）を示す。

40

【2295】

ジャンプ命令コード「JR cc, e」が実行されると、フラグ条件ccが成立している場合に、当該ジャンプ命令コードが規定されている番地（PC）にジャンプ値eを加算した値（PC+e）がPCレジスタに格納される。その結果、オペレーションコードの番地が、当該ジャンプ命令コードが規定されている番地（PC）からジャンプ値eだけ離れた番地（PC+e）にジャンプ（移動）する。

【2296】

図328に、ソースプログラム上で実際に使用され得るジャンプ命令コード「JR cc

50

、e」の種類の一覧を示す。ジャンプ命令コード「JR cc, e」としては、「JR NZ, e」(No. D5a)～「JR TZ, e」(No. D5f)の6種類の命令コードが設けられている。

【2297】

例えば、ジャンプ命令コード「JR NZ, e」(No. D5a)が実行されると、Zフラグがリセットされている場合(Zフラグの値が「0」である場合)に、当該ジャンプ命令コードが規定されている番地(PC)にジャンプ値eを加算した値(PC+e)がPCレジスタに格納される。これにより、オペレーションコードの番地(処理の実行番地)が、当該ジャンプ命令コードが規定されている番地(PC)からジャンプ値eだけ離れた番地(PC+e)にジャンプする。例えば、ジャンプ命令コード「JR Z, e」(No. D5b)が実行されると、Zフラグがセットされている場合(Zフラグの値が「1」である場合)に、当該ジャンプ命令コードが規定されている番地(PC)にジャンプ値eを加算した値(PC+e)がPCレジスタに格納される。これにより、オペレーションコードの番地が、当該ジャンプ命令コードが規定されている番地(PC)からジャンプ値eだけ離れた番地(PC+e)にジャンプする。

【2298】

例えば、ジャンプ命令コード「JR NC, e」(No. D5c)が実行されると、Cフラグ(CYフラグ)がリセットされている場合(Cフラグの値が「0」である場合)に、当該ジャンプ命令コードが規定されている番地(PC)にジャンプ値eを加算した値(PC+e)がPCレジスタに格納される。これにより、オペレーションコードの番地が、当該ジャンプ命令コードが規定されている番地(PC)からジャンプ値eだけ離れた番地(PC+e)にジャンプする。例えば、ジャンプ命令コード「JR C, e」(No. D5d)が実行されると、Cフラグがセットされている場合(Cフラグの値が「1」である場合)に、当該ジャンプ命令コードが規定されている番地(PC)にジャンプ値eを加算した値(PC+e)がPCレジスタに格納される。これにより、オペレーションコードの番地が、当該ジャンプ命令コードが規定されている番地(PC)からジャンプ値eだけ離れた番地(PC+e)にジャンプする。

【2299】

例えば、ジャンプ命令コード「JR NTZ, e」(No. D5e)が実行されると、TZフラグがリセットされている場合(TZフラグの値が「0」である場合)に、当該ジャンプ命令コードが規定されている番地(PC)にジャンプ値eを加算した値(PC+e)がPCレジスタに格納される。これにより、オペレーションコードの番地が、当該ジャンプ命令コードが規定されている番地(PC)からジャンプ値eだけ離れた番地(PC+e)にジャンプする。また、例えば、ジャンプ命令コード「JR TZ, e」(No. D5f)が実行されると、TZフラグがセットされている場合(TZフラグの値が「1」である場合)に、当該ジャンプ命令コードが規定されている番地(PC)にジャンプ値eを加算した値(PC+e)がPCレジスタに格納される。これにより、オペレーションコードの番地が、当該ジャンプ命令コードが規定されている番地(PC)からジャンプ値eだけ離れた番地(PC+e)にジャンプする。

【2300】

(2) ジャンプ命令コード「TJP」(No. D6)

ジャンプ命令コード「TJP」は、1バイトの命令コードであり、オペランド(OP)として、命令「TJP」(第1OP)のみが配置(規定)されて構成される。

【2301】

ジャンプ命令コード「TJP」が実行されると、PCレジスタ(プログラム・カウンタPC)に格納された値(当該ジャンプ命令コードが規定されている番地)に「1」とAレジスタに格納された値(アドレス相対値)とを加算した値(PC+1+A)で指定されるアドレス領域(指定メモリ)に格納されている8ビット(1バイト)の値が、PCレジスタの下位側(PCl)にロードされる。この結果、オペレーションコードの番地が、指定メモリに格納されている値で指定される番地にジャンプする。

【2302】

(3) ジャンプ命令コード「TJP B」(No. D7)～「TJP L」(No. D12) ジャンプ命令コード「TJP r (B、C、D、E、H又はL)」は、2バイトの命令コードであり、オペランド(OP)として、命令「TJP」(第1OP)及び1バイトのレジスタr (Aレジスタ以外の汎用レジスタ：第2OP)がこの順で配置(規定)されて構成される。

【2303】

ジャンプ命令コード「TJP r (B、C、D、E、H又はL)」(2バイトの命令コード)が実行されると、PCレジスタ(プログラム・カウンタPC)に格納された値(当該ジャンプ命令コードの上位側1バイトの番地)に「2」とレジスタr (Aレジスタ以外の汎用レジスタ)に格納された値(アドレス相対値)とを加算した値($PC + 2 + r$)で指定されるアドレス領域(指定メモリ)に格納されている8ビット(1バイト)の値が、PCレジスタの下位側(PC L)にロードされる。この結果、オペレーションコードの番地が、指定メモリに格納されている値で指定される番地にジャンプする。

10

【2304】

(4) ジャンプ命令コード「TJPW」(No. D13)

ジャンプ命令コード「TJPW」は、1バイトの命令コードであり、オペランド(OP)として、命令「TJPW」(第1OP)のみが配置(規定)されて構成される。

【2305】

ジャンプ命令コード「TJPW」が実行されると、PCレジスタ(プログラム・カウンタPC)に格納された値(当該ジャンプ命令コードが規定されている番地)に「1」とAレジスタに格納された値(アドレス相対値)を2倍した値とを加算した値($PC + 1 + A \times 2$)で指定されるアドレス領域(指定メモリ)に格納されている16ビット(2バイト)の値が、PCレジスタにロードされる。この結果、オペレーションコードの番地が、指定メモリに格納されている値で指定される番地にジャンプする。

20

【2306】

(5) ジャンプ命令コード「TJPW B」(No. D14)～「TJPW L」(No. D19)

ジャンプ命令コード「TJPW B」～「TJPW L」はそれぞれ、2バイトの命令コードであり、オペランド(OP)として、命令「TJPW」(第1OP)及び1バイトのレジスタr (Aレジスタ以外の汎用レジスタ：第2OP)がこの順で配置(規定)されて構成される。

30

【2307】

ジャンプ命令コード「TJPW r (B、C、D、E、H又はL)」(2バイトの命令コード)が実行されると、PCレジスタ(プログラム・カウンタPC)に格納された値(当該ジャンプ命令コードの上位側1バイトの番地)に「2」とレジスタr (Aレジスタ以外の汎用レジスタ)に格納された値(アドレス相対値)を2倍した値とを加算した値($PC + 2 + r \times 2$)で指定されるアドレス領域(指定メモリ)に格納されている16ビット(2バイト)の値が、PCレジスタにロードされる。この結果、オペレーションコードの番地が、指定メモリに格納されている値で指定される番地にジャンプする。

40

【2308】

[コンペアロード命令(加算)コード]

図329～図331は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能なコンペアロード(比較読み出し)命令(加算)コードの一覧表である。なお、コンペアロード命令(加算)コードは、1つの命令コードで、比較処理、比較結果に応じた特定の演算処理(後述の1加算処理又は特定値維持処理)、及び、特定の演算処理の結果のロード(セット)処理を実行可能な命令コードである。

【2309】

図329に記載の命令番号E1(「ICPLD A, r」)～命令番号E6(「ICPWLD (HL), mn」)のコンペアロード命令(加算)コードは、本実施形態だけでな

50

く、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図329～図331に記載の命令番号E7（「ICPLL A, B」）～命令番号E22（「ICPWL HL, mn」）のコンペアロード命令（加算）コードは、本実施形態で新たに設けられた命令コードである。なお、コンペアロード命令（加算）コードの実行時には、その命令コードの種別（命令番号）に関係なく、Nフラグに「1」がセットされ、Nフラグ以外の各フラグの状態は命令の実行結果に応じて適宜変化する（設定される）。以下では、主な、コンペアロード命令（加算）コードについて詳述する。

【2310】

（1）コンペアロード命令（加算）コード「ICPLD A, r」（No. E1）

コンペアロード命令（加算）コード「ICPLD A, r」は、2バイトの命令コードであり、オペランド（OP）として、命令「ICPLD」（第1OP）、第1比較値格納先「A」（第2OP）及び第2比較値格納先「r（Aレジスタ以外の汎用レジスタ）」（第3OP）がこの順で配置（規定）されて構成される。

【2311】

コンペアロード命令（加算）コード「ICPLD A, r」が実行されると、Aレジスタに格納されている値（第1比較値）とAレジスタ以外のレジスタrに格納されている値（第2比較値）とが比較され、比較結果においてレジスタrに格納されている値の方が大きい場合には、Aレジスタに格納されている値に1が加算され、該加算後の値がAレジスタにロード（セット）され、一方、比較結果においてレジスタrに格納されている値の方が大きくない場合には、「0」がAレジスタにロード（セット）される。

【2312】

なお、コンペアロード命令（加算）コード「ICPLD A, r」の実行時に行われる、Aレジスタに格納されている値とAレジスタ以外のレジスタrに格納されている値との比較処理では、Aレジスタに格納されている値からレジスタrに格納されている値を減算する処理（ $A - r$ ）が行われる。また、比較処理の結果はCフラグ（CYフラグ）に反映され、Aレジスタに格納されている値よりレジスタrに格納されている値の方が大きい場合には、Cフラグに「1」がセットされ、それ以外の場合には、Cフラグに「0」がセットされる。それゆえ、コンペアロード命令（加算）コード「ICPLD A, r」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

【2313】

図332に、ソースプログラム上で実際に使用され得るコンペアロード命令（加算）コード「ICPLD A, r」の種類の一覧を示す。コンペアロード命令（加算）コード「ICPLD A, r」としては、「ICPLD A, B」（No. E1a）～「ICPLD A, L」（No. E1f）の6種類の命令コードが設けられている。

【2314】

例えば、コンペアロード命令（加算）コード「ICPLD A, B」（No. E1a）が実行されると、Aレジスタに格納されている値とBレジスタに格納されている値とが比較され、比較結果においてBレジスタに格納されている値の方が大きい場合、Aレジスタに格納されている値に1が加算され、該加算後の値がAレジスタにロード（セット）され、一方、比較結果においてBレジスタに格納されている値の方が大きくない場合には、「0」がAレジスタにロード（セット）される。また、例えば、コンペアロード命令（加算）コード「ICPLD A, H」（No. E1e）が実行されると、Aレジスタに格納されている値とHレジスタに格納されている値とが比較され、比較結果においてHレジスタに格納されている値の方が大きい場合、Aレジスタに格納されている値に1が加算され、該加算後の値がAレジスタにロード（セット）され、一方、比較結果においてHレジスタに格納されている値の方が大きくない場合には、「0」がAレジスタにロード（セット）される。

【2315】

（2）コンペアロード命令（加算）コード「ICPLD (HL), r」（No. E3）

コンペアロード命令（加算）コード「ICPLD (HL), r」は、2バイトの命令コードであり、オペランド(OP)として、命令「ICPLD」(第1OP)、第1比較値格納先「(HL)」(第2OP)及び第2比較値格納先「r (H及びLレジスタ以外の汎用レジスタ)」(第3OP)がこの順で配置(規定)されて構成される。

【2316】

コンペアロード命令（加算）コード「ICPLD (HL), r」が実行されると、HLレジスタに格納された値で指定されるアドレス領域に格納されている値(第1比較値)とレジスタrに格納されている値(第2比較値)とが比較され、比較結果においてレジスタrに格納されている値の方が大きい場合には、HLレジスタで指定されるアドレス領域に格納されている値に1が加算され、該加算後の値がHLレジスタで指定されるアドレス領域にロード(セット)され、一方、比較結果においてレジスタrに格納されている値の方が大きくない場合には、「0」がHLレジスタで指定されるアドレス領域にロード(セット)される。

【2317】

なお、コンペアロード命令（加算）コード「ICPLD (HL), r」の実行時に行われる、HLレジスタで指定されるアドレス領域に格納されている値とレジスタrに格納されている値との比較処理では、HLレジスタで指定されるアドレス領域に格納されている値からレジスタrに格納されている値を減算する処理 $((HL) - r)$ が行われる。また、比較処理の結果はCフラグ(CYフラグ)に反映され、HLレジスタで指定されるアドレス領域に格納されている値よりレジスタrに格納されている値の方が大きい場合には、Cフラグに「1」がセットされ、それ以外の場合には、Cフラグに「0」がセットされる。それゆえ、コンペアロード命令（加算）コード「ICPLD (HL), r」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

【2318】

図333に、ソースプログラム上で実際に使用され得るコンペアロード命令（加算）コード「ICPLD (HL), r」の種類の一覧を示す。コンペアロード命令（加算）コード「ICPLD (HL), r」としては、「ICPLD (HL), A」(No. E3a)～「ICPLD (HL), E」(No. E3e)の5種類の命令コードが設けられている。

【2319】

例えば、コンペアロード命令（加算）コード「ICPLD (HL), A」(No. E3a)が実行されると、HLレジスタで指定されるアドレス領域に格納されている値とAレジスタに格納されている値とが比較され、比較結果においてAレジスタに格納されている値の方が大きい場合には、HLレジスタで指定されるアドレス領域に格納されている値に1が加算され、該加算後の値がHLレジスタで指定されるアドレス領域にロード(セット)され、一方、比較結果においてAレジスタに格納されている値の方が大きくない場合には、「0」がHLレジスタで指定されるアドレス領域にロード(セット)される。また、例えば、コンペアロード命令（加算）コード「ICPLD (HL), D」(No. E3d)が実行されると、HLレジスタで指定されるアドレス領域に格納されている値とDレジスタに格納されている値とが比較され、比較結果においてDレジスタに格納されている値の方が大きい場合には、HLレジスタで指定されるアドレス領域に格納されている値に1が加算され、該加算後の値がHLレジスタで指定されるアドレス領域にロード(セット)され、一方、比較結果においてDレジスタに格納されている値の方が大きくない場合には、「0」がHLレジスタで指定されるアドレス領域にロード(セット)される。

【2320】

(3) コンペアロード命令（加算）コード「ICPLL A, B」(No. E7)～「ICPLL A, L」(No. E12)

コンペアロード命令（加算）コード「ICPLL A, B」～「ICPLL A, L」はそれぞれ、2バイトの命令コードであり、オペランド(OP)として、命令「ICPLL」

（第1OP）、第1比較値格納先「A」（第2OP）及び第2比較値格納先「r（Aレジスタ以外の汎用レジスタ）」がこの順で配置（規定）されて構成される。

【2321】

コンペアロード命令（加算）コード「ICPLL A, r（B、C、D、E、H又はL）」が実行されると、Aレジスタに格納されている値（第1比較値）とレジスタrに格納されている値（第2比較値）とが比較され、比較結果においてレジスタrに格納されている値の方が大きい場合には、Aレジスタに格納されている値に1が加算され、該加算後の値がAレジスタにロード（セット）され、一方、比較結果においてレジスタrに格納されている値の方が大きくない場合には、レジスタrに格納されている値がAレジスタにロード（セット）される。すなわち、レジスタrに格納されている値（第2比較値）は、Aレジスタに格納される値（第1比較値）の上限値である。

10

【2322】

例えば、コンペアロード命令（加算）コード「ICPLL A, B」（No. E7）が実行されると、Aレジスタに格納されている値とBレジスタに格納されている値（上限値）とが比較され、比較結果においてBレジスタに格納されている値の方が大きい場合には、Aレジスタに格納されている値に1が加算され、該加算後の値がAレジスタにロード（セット）され、一方、比較結果においてBレジスタに格納されている値の方が大きくない場合には、Bレジスタに格納されている値がAレジスタにロード（セット）される。また、例えば、コンペアロード命令（加算）コード「ICPLL A, H」（No. E11）が実行されると、Aレジスタに格納されている値とHレジスタに格納されている値（上限値）とが比較され、比較結果においてHレジスタに格納されている値の方が大きい場合には、Aレジスタに格納されている値に1が加算され、該加算後の値がAレジスタにロード（セット）され、一方、比較結果においてHレジスタに格納されている値の方が大きくない場合には、Hレジスタに格納されている値がAレジスタにロード（セット）される。

20

【2323】

なお、コンペアロード命令（加算）コード「ICPLL A, r」の実行時に行われる、Aレジスタに格納されている値とレジスタr（Aレジスタ以外の汎用レジスタ）に格納されている値との比較処理では、Aレジスタに格納されている値からレジスタrに格納されている値を減算する処理（A-r）が行われる。また、比較処理の結果はCフラグ（CYフラグ）に反映され、Aレジスタに格納されている値よりレジスタrに格納されている値の方が大きい場合には、Cフラグに「1」がセットされ、それ以外の場合には、Cフラグに「0」がセットされる。それゆえ、コンペアロード命令（加算）コード「ICPLL A, r」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

30

【2324】

（4）コンペアロード命令（加算）コード「ICPLL A, n」（No. E13）～「ICPLL L, n」（No. E19）

コンペアロード命令（加算）コード「ICPLL A, n」～「ICPLL L, n」はそれぞれ、3バイトの命令コードであり、オペランド（OP）として、命令「ICPLL」（第1OP）、比較値格納先「r（A、B、C、D、E、H、Lレジスタ）」（第2OP）及び8ビットの整数値「n」（第3OP）がこの順で配置（規定）されて構成される。なお、このコンペアロード命令（加算）コードにおいて、第3OPとして規定される8ビットの整数値n（直値）は、第2OPとして規定された比較値格納先に格納される値の上限値である。

40

【2325】

コンペアロード命令（加算）コード「ICPLL r, n」が実行されると、レジスタrに格納されている値と8ビットの整数値n（上限値）とが比較され、比較結果において8ビットの整数値nの方が大きい場合には、レジスタrに格納されている値に1が加算され、該加算後の値がレジスタrにロード（セット）され、一方、比較結果において8ビットの整数値nの方が大きくない場合には、8ビットの整数値nがレジスタrにロード（セッ

50

ト) される。

【2326】

例えば、コンペアロード命令（加算）コード「ICPLL A, n」（No. E13）が実行されると、Aレジスタに格納されている値と8ビットの整数値n（上限値）とが比較され、比較結果において8ビットの整数値nの方が大きい場合には、Aレジスタに格納されている値に1が加算され、該加算後の値がAレジスタにロード（セット）され、一方、比較結果において8ビットの整数値nの方が大きくない場合には、8ビットの整数値nがAレジスタにロード（セット）される。また、例えば、コンペアロード命令（加算）コード「ICPLL E, n」（No. E17）が実行されると、Eレジスタに格納されている値と8ビットの整数値n（上限値）とが比較され、比較結果において8ビットの整数値nの方が大きい場合には、Eレジスタに格納されている値に1が加算され、該加算後の値がEレジスタにロード（セット）され、一方、比較結果において8ビットの整数値nの方が大きくない場合には、8ビットの整数値nがEレジスタにロード（セット）される。

【2327】

なお、コンペアロード命令（加算）コード「ICPLL r, n」の実行時に行われる、レジスタrに格納されている値と8ビットの整数値nとの比較処理では、レジスタrに格納されている値から8ビットの整数値nを減算する処理（ $r - n$ ）が行われる。また、比較処理の結果はCフラグ（CYフラグ）に反映され、レジスタrに格納されている値より8ビットの整数値nの方が大きい場合には、Cフラグに「1」がセットされ、それ以外の場合には、Cフラグに「0」がセットされる。それゆえ、コンペアロード命令（加算）コード「ICPLL r, n」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

【2328】

(5) コンペアロード命令（加算）コード「ICPWLL BC, mn」（No. E20）、「ICPWLL DE, mn」（No. E21）、「ICPWLL HL, mn」（No. E22）

コンペアロード命令（加算）コード「ICPWLL BC, mn」、「ICPWLL DE, mn」及び「ICPWLL HL, mn」はそれぞれ、4バイトの命令コードであり、オペランド（OP）として、命令「ICPWLL」（第1OP）、比較値格納先「ss（BC、DE又はHLレジスタ）」（第2OP）及び16ビットの整数値「mn」（第3OP）がこの順で配置（規定）されて構成される。なお、このコンペアロード命令（加算）コードにおいて、第3OPとして規定される16ビットの整数値mnは、第2OPとして規定された比較値格納先（ペアレジスタss）に格納される値の上限値である。

【2329】

コンペアロード命令（加算）コード「ICPWLL ss（BC、DE又はHL）, mn」が実行されると、ペアレジスタssに格納されている値と16ビットの整数値mn（上限値）とが比較され、比較結果において16ビットの整数値mnの方が大きい場合には、ペアレジスタssに格納されている値に1が加算され、該加算後の値がペアレジスタssにロード（セット）され、一方、比較結果において16ビットの整数値mnの方が大きくない場合には、16ビットの整数値mnがペアレジスタssにロード（セット）される。なお、16ビットの整数値mnがペアレジスタssにロードされる際には、16ビットの整数値mn中の上位側8ビットの整数値mがペアレジスタssの上位側レジスタにロードされ、16ビットの整数値mn中の下位側8ビットの整数値nがペアレジスタssの下位側レジスタにロードされる。

【2330】

例えば、コンペアロード命令（加算）コード「ICPWLL BC, mn」（No. E20）が実行されると、BCレジスタに格納されている値と16ビットの整数値mn（上限値）とが比較され、比較結果において16ビットの整数値mnの方が大きい場合には、BCレジスタに格納されている値に1が加算され、該加算後の値がBCレジスタにロード（セット）され、一方、比較結果において16ビットの整数値mnの方が大きくない場合に

は、16ビットの整数値 mn がBCレジスタにロード（セット）される。なお、16ビットの整数値 mn がBCレジスタにロードされる際には、16ビットの整数値 mn 中の上位側8ビットの整数値 m がBレジスタにロードされ、16ビットの整数値 mn 中の下位側8ビットの整数値 n がCレジスタにロードされる。

【2331】

また、例えば、コンペアロード命令（加算）コード「ICPWL L HL, mn 」（No. E22）が実行されると、HLレジスタに格納されている値と16ビットの整数値 mn （上限値）とが比較され、比較結果において16ビットの整数値 mn の方が大きい場合には、HLレジスタに格納されている値に1が加算され、該加算後の値がHLレジスタにロード（セット）され、一方、比較結果において16ビットの整数値 mn の方が大きくない場合には、16ビットの整数値 mn がHLレジスタにロード（セット）される。なお、16ビットの整数値 mn がHLレジスタにロードされる際には、16ビットの整数値 mn 中の上位側8ビットの整数値 m がHレジスタにロードされ、16ビットの整数値 mn 中の下位側8ビットの整数値 n がLレジスタにロードされる。

【2332】

なお、コンペアロード命令（加算）コード「ICPWL L ss, mn 」の実行時に行われる、ペアレジスタ ss （BC、DE又はHLレジスタ）に格納されている値と16ビットの整数値 mn との比較処理では、ペアレジスタ ss に格納されている値から16ビットの整数値 mn を減算する処理（ $ss - mn$ ）が行われる。また、比較処理の結果はCフラグ（CYフラグ）に反映され、ペアレジスタ ss に格納されている値より16ビットの整数値 mn の方が大きい場合には、Cフラグに「1」がセットされ、それ以外の場合には、Cフラグに「0」がセットされる。それゆえ、コンペアロード命令（加算）コード「ICPWL L ss, mn 」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

【2333】

【コンペアロード命令（減算）コード】

図334～図336は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能なコンペアロード（比較読み出し）命令（減算）コードの一覧表である。なお、コンペアロード命令（減算）コードは、1つの命令コードで、比較処理、比較結果に応じた特定の演算処理（後述の1減算処理又は特定値維持処理）、及び、特定の演算処理の結果のロード（セット）処理を実行可能な命令コードである。

【2334】

図334に記載の命令番号F1（「DCPLD A, r 」）～命令番号F6（「DCPWL L (HL), mn 」）のコンペアロード命令（減算）コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図334～図336に記載の命令番号F7（「DCPLL A, B」）～命令番号F22（「DCPWL L HL, mn 」）のコンペアロード命令（減算）コードは、本実施形態で新たに設けられた命令コードである。なお、コンペアロード命令（減算）コードの実行時には、その命令コードの種別（命令番号）に関係なく、Nフラグに「1」がセットされ、Nフラグ以外の各フラグの状態は命令の実行結果に応じて適宜変化する（設定される）。

以下、主な、コンペアロード命令（減算）コードについて詳述する。

【2335】

（1）コンペアロード命令（減算）コード「DCPLD A, r 」（No. F1）

コンペアロード命令（減算）コード「DCPLD A, r 」は、2バイトの命令コードであり、オペランド（OP）として、命令「DCPLD」（第1OP）、比較値格納先「A」（第2OP）及び特定値格納先「 r （Aレジスタ以外の汎用レジスタ）」（第3OP）がこの順で配置（規定）されて構成される。

【2336】

コンペアロード命令（減算）コード「DCPLD A, r 」が実行されると、Aレジスタに格納されている値（比較値）と「1」とが比較され、比較結果においてAレジスタに格

10

20

30

40

50

納されている値が「1」より小さい場合には、レジスタ r に格納されている値（特定値）が A レジスタにロード（セット）され、一方、比較結果において A レジスタに格納されている値が「1」より小さくない場合には、A レジスタに格納されている値が 1 減算され、該減算後の値が A レジスタにロード（セット）される。

【2337】

なお、コンペアロード命令（減算）コード「DCPLD A, r 」の実行時に行われる、A レジスタに格納されている値と「1」との比較処理では、A レジスタに格納されている値から「1」を減算する処理（ $A - 1$ ）が行われる。また、比較処理の結果は C フラグ（CY フラグ）に反映され、A レジスタに格納されている値が「1」より小さい場合には、C フラグに「1」がセットされ、それ以外の場合には、C フラグに「0」がセットされる。それゆえ、コンペアロード命令（減算）コード「DCPLD A, r 」の実行時に行われる比較結果に応じた処理の上記分岐処理は、C フラグにセットされている値に基づいて行われる。

【2338】

図 337 に、ソースプログラム上で実際に規定され得るコンペアロード命令（減算）コード「DCPLD A, r （A レジスタ以外の汎用レジスタ）」の種類の一覧を示す。コンペアロード命令（減算）コード「DCPLD A, r 」としては、「DCPLD A, B」（No. F1a）～「DCPLD A, L」（No. F1f）の 6 種類の命令コードが設けられている。

【2339】

例えば、コンペアロード命令（減算）コード「DCPLD A, B」（No. F1a）が実行されると、A レジスタに格納されている値と「1」とが比較され、比較結果において A レジスタに格納されている値が「1」より小さい場合には、B レジスタに格納されている値が A レジスタにロード（セット）され、一方、比較結果において A レジスタに格納されている値が「1」より小さくない場合には、A レジスタに格納されている値が 1 減算され、該減算後の値が A レジスタにロード（セット）される。また、例えば、コンペアロード命令（減算）コード「DCPLD A, H」（No. F1e）が実行されると、A レジスタに格納されている値と「1」とが比較され、比較結果において A レジスタに格納されている値が「1」より小さい場合には、H レジスタに格納されている値が A レジスタにロード（セット）され、一方、比較結果において A レジスタに格納されている値が「1」より小さくない場合には、A レジスタに格納されている値が 1 減算され、該減算後の値が A レジスタにロード（セット）される。

【2340】

（2）コンペアロード命令（減算）コード「DCPLD (HL), r 」（No. F3）
コンペアロード命令（減算）コード「DCPLD (HL), r 」は、2 バイトの命令コードであり、オペランド（OP）として、命令「DCPLD」（第 1 OP）、比較値格納先「(HL)」（第 2 OP）及び特定値格納先「 r （H 及び L レジスタ以外の汎用レジスタ）」（第 3 OP）がこの順で配置（規定）されて構成される。

【2341】

コンペアロード命令（減算）コード「DCPLD (HL), r 」が実行されると、HL レジスタに格納された値で指定されるアドレス領域に格納されている値（比較値）と「1」とが比較され、比較結果において HL レジスタで指定されるアドレス領域に格納されている値が「1」より小さい場合には、レジスタ r に格納されている値（特定値）が HL レジスタで指定されるアドレス領域にロード（セット）され、一方、比較結果において HL レジスタで指定されるアドレス領域に格納されている値が「1」より小さくない場合には、HL レジスタで指定されるアドレス領域に格納されている値が 1 減算され、該減算後の値が HL レジスタで指定されるアドレス領域にロード（セット）される。

【2342】

なお、コンペアロード命令（減算）コード「DCPLD (HL), r 」の実行時に行われる、HL レジスタで指定されるアドレス領域に格納されている値と「1」との比較処理

10

20

30

40

50

では、HLレジスタで指定されるアドレス領域に格納されている値から「1」を減算する処理（ $(HL) - 1$ ）が行われる。また、比較処理の結果はCフラグ（CYフラグ）に反映され、HLレジスタで指定されるアドレス領域に格納されている値が「1」より小さい場合には、Cフラグに「1」がセットされ、それ以外の場合には、Cフラグに「0」がセットされる。それゆえ、コンペアロード命令（減算）コード「DCPLD (HL), r」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

【2343】

図338に、ソースプログラム上で実際に規定され得るコンペアロード命令（減算）コード「DCPLD (HL), r」の種類の一覧を示す。コンペアロード命令（減算）コード「DCPLD (HL), r」としては、「DCPLD (HL), A」(No. F3a) ~ 「DCPLD (HL), E」(No. F3e)の5種類の命令コードが設けられている。

【2344】

例えば、コンペアロード命令（減算）コード「DCPLD (HL), A」(No. F3a)が実行されると、HLレジスタで指定されるアドレス領域に格納されている値と「1」とが比較され、比較結果においてHLレジスタで指定されるアドレス領域に格納されている値が「1」より小さい場合には、Aレジスタに格納されている値が、HLレジスタで指定されるアドレス領域にロード（セット）され、一方、比較結果においてHLレジスタで指定されるアドレス領域に格納されている値が「1」より小さくない場合には、HLレジスタで指定されるアドレス領域に格納されている値が1減算され、該減算後の値がHLレジスタで指定されるアドレス領域にロード（セット）される。

【2345】

また、例えば、コンペアロード命令（減算）コード「DCPLD (HL), D」(No. F3d)が実行されると、HLレジスタで指定されるアドレス領域に格納されている値と「1」とが比較され、比較結果においてHLレジスタで指定されるアドレス領域に格納されている値が「1」より小さい場合には、Dレジスタに格納されている値が、HLレジスタで指定されるアドレス領域にロード（セット）され、一方、比較結果においてHLレジスタで指定されるアドレス領域に格納されている値が「1」より小さくない場合には、HLレジスタで指定されるアドレス領域に格納されている値が1減算され、該減算後の値がHLレジスタで指定されるアドレス領域にロード（セット）される。

【2346】

(3) コンペアロード命令（減算）コード「DCPLL A, B」(No. F7) ~ 「DCPLL A, L」(No. F12)

コンペアロード命令（減算）コード「DCPLL A, B」~ 「DCPLL A, L」はそれぞれ、2バイトの命令コードであり、オペランド(OP)として、命令「DCPLL」(第1OP)、第1比較値格納先「A」(第2OP)及び第2比較値格納先「r (Aレジスタ以外の汎用レジスタ)」(第3OP)がこの順で配置（規定）されて構成される。

【2347】

コンペアロード命令（減算）コード「DCPLL A, r (B、C、D、E、H又はL)」が実行されると、Aレジスタに格納されている値（第1比較値）とレジスタr (Aレジスタ以外の汎用レジスタ)に格納されている値（第2比較値）とが比較され、比較結果においてレジスタrに格納されている値の方が小さい場合には、Aレジスタに格納されている値が1減算され、該減算後の値がAレジスタにロード（セット）され、一方、比較結果においてレジスタrに格納されている値の方が小さくない場合には、レジスタrに格納されている値がAレジスタにロード（セット）される。すなわち、レジスタrに格納されている値（第2比較値）は、Aレジスタに格納される値（第1比較値）の下限値である。

【2348】

例えば、コンペアロード命令（減算）コード「DCPLL A, B」(No. F7)が実行されると、Aレジスタに格納されている値とBレジスタに格納されている値（下限値）

とが比較され、比較結果においてBレジスタに格納されている値の方が小さい場合には、Aレジスタに格納されている値が1減算され、該減算後の値がAレジスタにロード（セット）され、一方、比較結果においてBレジスタに格納されている値の方が小さくない場合には、Bレジスタに格納されている値がAレジスタにロード（セット）される。また、例えば、コンペアロード命令（減算）コード「DCPLL A, H」（No. F11）が実行されると、Aレジスタに格納されている値とHレジスタに格納されている値（下限値）とが比較され、比較結果においてHレジスタに格納されている値の方が小さい場合には、Aレジスタに格納されている値が1減算され、該減算後の値がAレジスタにロード（セット）され、一方、比較結果においてHレジスタに格納されている値の方が小さくない場合には、Hレジスタに格納されている値がAレジスタにロード（セット）される。

10

【2349】

なお、コンペアロード命令（減算）コード「DCPLL A, r」の実行時に行われる、Aレジスタに格納されている値とレジスタrに格納されている値（下限値）との比較処理では、Aレジスタに格納されている値からレジスタrに格納されている値を減算する処理（ $A - r$ ）が行われる。また、比較処理の結果はCフラグ（CYフラグ）に反映され、Aレジスタに格納されている値よりレジスタrに格納されている値の方が小さい場合には、Cフラグに「0」がセットされ、それ以外の場合には、Cフラグに「1」がセットされる。それゆえ、コンペアロード命令（減算）コード「DCPLL A, r」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

20

【2350】

（4）コンペアロード命令（減算）コード「DCPLL A, n」（No. F13）～「DCPLL L, n」（No. F19）

コンペアロード命令（減算）コード「DCPLL A, n」～「DCPLL L, n」はそれぞれ、3バイトの命令コードであり、オペランド（OP）として、命令「DCPLL」（第1OP）、比較値格納先「r（A、B、C、D、E、H、Lレジスタ）」（第2OP）及び8ビットの整数値「n」（第3OP）がこの順で配置（規定）されて構成される。なお、このコンペアロード命令（減算）コードにおいて、第3OPとして規定される8ビットの整数値nは、第2OPとして規定された比較値格納先に格納される値の下限値である。

30

【2351】

コンペアロード命令（減算）コード「DCPLL r, n」が実行されると、レジスタrに格納されている値と8ビットの整数値n（下限値）とが比較され、比較結果において8ビットの整数値nの方が小さい場合には、レジスタrに格納されている値が1減算され、該減算後の値がレジスタrにロード（セット）され、一方、比較結果において8ビットの整数値nの方が小さくない場合には、8ビットの整数値nがレジスタrにロード（セット）される。

【2352】

例えば、コンペアロード命令（減算）コード「DCPLL A, n」（No. F13）が実行されると、Aレジスタに格納されている値と8ビットの整数値n（下限値）とが比較され、比較結果において8ビットの整数値nの方が小さい場合には、Aレジスタに格納されている値が1減算され、該減算後の値がAレジスタにロード（セット）され、一方、比較結果において8ビットの整数値nの方が小さくない場合には、8ビットの整数値nがAレジスタにロード（セット）される。また、例えば、コンペアロード命令（減算）コード「DCPLL E, n」（No. F17）が実行されると、Eレジスタに格納されている値と8ビットの整数値n（下限値）とが比較され、比較結果において8ビットの整数値nの方が小さい場合には、Eレジスタに格納されている値が1減算され、該減算後の値がEレジスタにロード（セット）され、一方、比較結果において8ビットの整数値nの方が小さくない場合には、8ビットの整数値nがEレジスタにロード（セット）される。

40

【2353】

50

なお、コンペアロード命令（減算）コード「DCPLL r, n」の実行時に行われる、レジスタrに格納されている値と8ビットの整数値n（下限値）との比較処理では、レジスタrに格納されている値から8ビットの整数値nを減算する処理（r - n）が行われる。また、比較処理の結果はCフラグ（CYフラグ）に反映され、レジスタrに格納されている値より8ビットの整数値nの方が小さい場合には、Cフラグに「0」がセットされ、それ以外の場合には、Cフラグに「1」がセットされる。それゆえ、コンペアロード命令（減算）コード「DCPLL r, n」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

【2354】

(5) コンペアロード命令（減算）コード「DCPWLL BC, mn」（No. F20）、「DCPWLL DE, mn」（No. F21）、「DCPWLL HL, mn」（No. F22）

コンペアロード命令（減算）コード「DCPWLL BC, mn」、「DCPWLL DE, mn」及び「DCPWLL HL, mn」はそれぞれ、4バイトの命令コードであり、オペランド（OP）として、命令「DCPWLL」（第1OP）、比較値格納先「ss（BC、DE又はHLレジスタ）」（第2OP）及び16ビットの整数値「mn」（第3OP）がこの順で配置（規定）されて構成される。なお、このコンペアロード命令（減算）コードにおいて、第3OPとして規定される16ビットの整数値mnは、第2OPとして規定された比較値格納先（ペアレジスタss）に格納される値の下限値である。

【2355】

コンペアロード命令（減算）コード「DCPWLL ss（BC、DE又はHL）, mn」が実行されると、ペアレジスタssに格納されている値と16ビットの整数値mn（下限値）とが比較され、比較結果において16ビットの整数値mnの方が小さい場合には、ペアレジスタssに格納されている値が1減算され、該減算後の値がペアレジスタssにロード（セット）され、一方、比較結果において16ビットの整数値mnの方が小さくない場合には、16ビットの整数値mnがペアレジスタssにロード（セット）される。なお、16ビットの整数値mnがペアレジスタssにロードされる際には、16ビットの整数値mn中の上位側8ビットの整数値mがペアレジスタssの上位側レジスタにロードされ、16ビットの整数値mn中の下位側8ビットの整数値nがペアレジスタssの下位側レジスタにロードされる。

【2356】

例えば、コンペアロード命令（減算）コード「DCPWLL BC, mn」（No. F20）が実行されると、BCレジスタに格納されている値と16ビットの整数値mn（下限値）とが比較され、比較結果において16ビットの整数値mnの方が小さい場合には、BCレジスタに格納されている値が1減算され、該減算後の値がBCレジスタにロード（セット）され、一方、比較結果において16ビットの整数値mnの方が小さくない場合には、16ビットの整数値mnがBCレジスタにロード（セット）される。なお、16ビットの整数値mnがBCレジスタにロードされる際、16ビットの整数値mn中の上位側8ビットの整数値mがBレジスタにロードされ、16ビットの整数値mn中の下位側8ビットの整数値nがCレジスタにロードされる。

【2357】

また、例えば、コンペアロード命令（減算）コード「DCPWLL HL, mn」（No. F22）が実行されると、HLレジスタに格納されている値と16ビットの整数値mn（下限値）とが比較され、比較結果において16ビットの整数値mnの方が小さい場合には、HLレジスタに格納されている値が1減算され、該減算後の値がHLレジスタにロード（セット）され、一方、比較結果において16ビットの整数値mnの方が小さくない場合には、16ビットの整数値mnがHLレジスタにロード（セット）される。なお、16ビットの整数値mnがHLレジスタにロードされる際、16ビットの整数値mn中の上位側8ビットの整数値mがHレジスタにロードされ、16ビットの整数値mn中の下位側8ビットの整数値nがLレジスタにロードされる。

【2358】

なお、コンペアロード命令（減算）コード「DCPWL L s s , m n」の実行時に行われる、ペアレジスタ s s（BC、DE又はHLレジスタ）に格納されている値と16ビットの整数値m n（下限値）との比較処理では、ペアレジスタ s sに格納されている値から16ビットの整数値m nを減算する処理（s s - m n）が行われる。また、比較処理の結果はCフラグ（CYフラグ）に反映され、ペアレジスタ s sに格納されている値より16ビットの整数値m nの方が小さい場合には、Cフラグに「0」がセットされ、それ以外の場合には、Cフラグに「1」がセットされる。それゆえ、コンペアロード命令（減算）コード「DCPWL L s s , m n」の実行時に行われる比較結果に応じた処理の上記分岐処理は、Cフラグにセットされている値に基づいて行われる。

10

【2359】

[シフト命令（左）コード]

図339は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能なシフト命令（左）コードの一覧表である。なお、シフト命令（左）コードは、8ビットデータ（上位側からビット7～ビット0）の全ビット又は1部のビットの値を上位側（左方向：ビット0からビット7に向かう方向）に1ビット分、シフト（移動）させるための命令コードである。

【2360】

図339に記載の命令番号G1（「SLA r」）及び命令番号G2（「SLA（HL）」）のシフト命令（左）コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図339に記載の命令番号G3（「SLBIT A, n」）～命令番号G10（「SLBITQ（k）, n」）のシフト命令（左）コードは、本実施形態で新たに設けられた命令コードである。なお、シフト命令（左）コードの実行時には、その命令コードの種別（命令番号）に関係なく、Hフラグ及びNフラグに「0」がセットされ、その他の各フラグの状態は命令の実行結果に応じて適宜変化する（設定される）。

20

以下、主な、シフト命令（左）コードについて詳述する。

【2361】

(1) シフト命令（左）コード「SLA r」（No. G1）

シフト命令（左）コード「SLA r」は、2バイトの命令コードであり、オペランド（OP）として、命令「SLA」（第1OP）及び1バイトのレジスタr（第2OP）がこの順で配置（規定）されて構成される。

30

【2362】

シフト命令（左）コード「SLA r」が実行されると、レジスタrの各ビット（ビット7～ビット0）の値が、上位側に1ビット分シフト（移動）するとともに、シフト前のビット7の値がCフラグ（CYフラグ）にセットされ、シフト後のビット0がリセットされる（「0」がセットされる）。

【2363】

図340に、ソースプログラム上で実際に使用され得るシフト命令（左）コード「SLA r」の種類の一覧を示す。シフト命令（左）コード「SLA r」としては、「SLA A」（No. G1a）～「SLA L」（No. G1g）の7種類の命令コードが設けられている。

40

【2364】

例えば、シフト命令（左）コード「SLA A」（No. G1a）が実行されると、Aレジスタの各ビットの値が、上位側に1ビット分シフト（移動）するとともに、Aレジスタのシフト前のビット7の値がCフラグ（CYフラグ）にセットされ、シフト後のビット0がリセットされる（「0」がセットされる）。また、例えば、シフト命令（左）コード「SLA D」（No. G1d）が実行されると、Dレジスタの各ビットの値が、上位側に1ビット分シフト（移動）するとともに、Dレジスタのシフト前のビット7の値がCフラグ（CYフラグ）にセットされ、シフト後のビット0がリセットされる（「0」がセット

50

される)。

【2365】

(2) シフト命令(左)コード「SLBIT A, n」(No. G3)～「SLBIT L, n」(No. G9)

シフト命令(左)コード「SLBIT A, n」～「SLBIT L, n」はそれぞれ、3バイトの命令コードであり、オペランド(OP)として、命令「SLBIT」(第1OP)、1バイトのレジスタ「r (A、B、C、D、E、H及びLレジスタ)」(第2OP)及び8ビットの整数値「n」(第3OP)がこの順で配置(規定)されて構成される。なお、このシフト命令(左)コードで第3OPとして規定される8ビットの整数値nは、レジスタr内においてシフト対象となるビットのうちの最下位ビットの位置を示す値(指定ビット位置: 0～7のいずれかの整数値)である。

【2366】

シフト命令(左)コード「SLBIT r, n」が実行されると、レジスタr内の指定ビット位置n(所定のビット位置)のビット(ビットn)から上位側の全ビット(ビット7～ビットn)の値が、上位側(左)に1ビット分シフト(移動)するとともに、レジスタrのシフト前のビット7の値がCフラグ(CYフラグ)にセットされ、シフト後のビットnがリセットされる(「0」がセットされる)。なお、シフト命令(左)コード「SLBIT r, n」の実行時には、レジスタr内の指定ビット位置nのビット(ビットn)より下位側に位置する全てのビット(ビットn-1～ビット0)の値は、シフトされず、シフト前の値が維持される。

【2367】

例えば、シフト命令(左)コード「SLBIT A, n」(No. G3)が実行されると、Aレジスタ内の指定ビット位置nのビット(ビットn)から上位側の全ビット(ビット7～ビットn)の値が、上位側に1ビット分シフトするとともに、Aレジスタのシフト前のビット7の値がCフラグにセットされ、シフト後のビットnがリセットされる。また、シフト命令(左)コード「SLBIT A, n」の実行時には、Aレジスタ内のビットnより下位側に位置する全てのビット(ビットn-1～ビット0)の値は、シフトせずに維持される。

【2368】

また、例えばシフト命令(左)コード「SLBIT D, n」(No. G6)が実行されると、Dレジスタ内の指定ビット位置nのビット(ビットn)から上位側の全ビット(ビット7～ビットn)の値が、上位側に1ビット分シフトするとともに、Dレジスタのシフト前のビット7の値がCフラグにセットされ、シフト後のビットnがリセットされる。また、シフト命令(左)コード「SLBIT D, n」の実行時には、Dレジスタ内のビットnより下位側に位置する全てのビット(ビットn-1～ビット0)の値は、シフトせずに維持される。

【2369】

より具体的には、例えば、レジスタrに格納されている8ビットデータが「10001111B」である場合に、シフト命令(左)コード「SLBIT r, 3」が実行されると、命令実行後のレジスタrに格納される8ビットデータは「00010111B」となり、Cフラグ(CYフラグ)には「1」がセットされる。

【2370】

(3) シフト命令(左)コード「SLBITQ (k), n」(No. G10)

シフト命令(左)コード「SLBITQ (k), n」は、Qレジスタ(拡張レジスタ)を用いてアドレス指定を行うシフト命令(左)コードである。シフト命令(左)コード「SLBITQ (k), n」は、4バイトの命令コードであり、オペランド(OP)として、命令「SLBITQ」(第1OP)、シフト対象データ格納先「(k)」及び8ビットの整数値「n(指定ビット位置)」(第3OP)がこの順で配置(規定)されて構成される。

【2371】

シフト命令（左）コード「SLBITQ (k), n」が実行されると、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と8ビットの整数値k（指定メモリの下位側1バイトのアドレス）とで指定されるアドレス領域（指定メモリ）に格納されている8ビットデータ内の指定ビット位置nのビット（ビットn）から上位側の全ビット（ビット7～ビットn）の値が、上位側に1ビット分シフト（移動）するとともに、当該8ビットデータのシフト前のビット7の値がCフラグにセットされ、シフト後のビットnがリセットされる。なお、シフト命令（左）コード「SLBITQ (k), n」の実行時には、Qレジスタに格納された値と8ビットデータkとで指定されるアドレス領域に格納されている8ビットデータ内のビットnより下位側に位置する全てのビット（ビットn-1～ビット0）の値は、シフトされずに維持される。

10

【2372】

〔シフト命令（右）コード〕

図341は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能なシフト命令（右）コードの一覧表である。なお、シフト命令（右）コードは、8ビットデータ（上位側からビット7～ビット0）の全ビット又は1部のビットの値を下位側（右方向：ビット7からビット0に向かう方向）に1ビット分、シフト（移動）させるための命令コードである。

【2373】

図341に記載の命令番号H1（「SRA r」）及び命令番号H2（「SRA (HL)」）のシフト命令（右）コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図341に記載の命令番号H3（「SRBIT A, n」）～命令番号H10（「SRBITQ (k), n」）のシフト命令（右）コードは、本実施形態で新たに設けられた命令コードである。なお、シフト命令（右）コードの実行時には、その命令コードの種別（命令番号）に関係なく、Hフラグ及びNフラグに「0」がセットされ、その他の各フラグの状態は命令の実行結果に応じて適宜変化する（設定される）。

20

以下、主な、シフト命令（右）コードについて詳述する。

【2374】

（1）シフト命令（右）コード「SRA r」（No. H1）

シフト命令（右）コード「SRA r」は、2バイトの命令コードであり、オペランド（OP）として、命令「SRA」（第1OP）及び1バイトのレジスタr（第2OP）がこの順で配置（規定）されて構成される。

30

【2375】

シフト命令（右）コード「SRA r」が実行されると、レジスタrの各ビット（ビット7～ビット0）の値が、下位側に1ビット分シフト（移動）するとともに、シフト前のビット0の値がCフラグ（CYフラグ）にセットされ、シフト後のビット7にはシフト前のビット7の値がセットされる（状態が維持される）。

【2376】

図342に、ソースプログラム上で実際に使用され得るシフト命令（右）コード「SRA r」の種類の一覧を示す。シフト命令（右）コード「SRA r」としては、「SRA A」（No. H1a）～「SRA L」（No. H1g）の7種類の命令コードが設けられている。

40

【2377】

例えば、シフト命令（右）コード「SRA A」（No. H1a）が実行されると、Aレジスタの各ビットの値が、下位側に1ビット分シフトするとともに、Aレジスタのシフト前のビット0の値がCフラグ（CYフラグ）にセットされ、シフト後のビット7にはシフト前のビット7の値がセットされる。また、例えばシフト命令（右）コード「SRA D」（No. H1d）が実行されると、Dレジスタの各ビットの値が、下位側に1ビット分シフトするとともに、Dレジスタのシフト前のビット0の値がCフラグ（CYフラグ）にセットされ、シフト後のビット7にはシフト前のビット7の値がセットされる。

50

【2378】

(2) シフト命令(右) コード「SRBIT A, n」(No. H3)～「SRBIT L, n」(No. H9)

シフト命令(右) コード「SRBIT A, n」～「SRBIT L, n」はそれぞれ、3バイトの命令コードであり、オペランド(OP)として、命令「SRBIT」(第1OP)、1バイトのレジスタ「r (A、B、C、D、E、H及びLレジスタ)」(第2OP)及び8ビットの整数値「n」(第3OP)がこの順で配置(規定)されて構成される。なお、このシフト命令(右)コードで第3OPとして規定される8ビットの整数値nは、レジスタr内においてシフト対象となるビットのうちの最上位ビットの位置を示す値(指定ビット位置: 0～7のいずれかの整数値)である。

10

【2379】

シフト命令(右) コード「SRBIT r, n」が実行されると、レジスタr内の指定ビット位置n(所定のビット位置)のビット(ビットn)から下位側の全ビット(ビットn～ビット0)の値が、下位側(右)に1ビット分シフト(移動)するとともに、レジスタrのシフト前のビット0の値がCフラグ(CYフラグ)にセットされ、シフト後のビットnがリセットされる(「0」がセットされる)。なお、シフト命令(右)コード「SRBIT r, n」の実行時には、レジスタr内のビットnより上位側に位置する全てのビット(ビット7～ビットn+1)の値は、シフトされず、シフト前の値が維持される。

【2380】

例えば、シフト命令(右)コード「SRBIT A, n」(No. H3)が実行されると、Aレジスタ内の指定ビット位置nのビット(ビットn)から下位側の全ビット(ビットn～ビット0)の値が、下位側に1ビット分シフトするとともに、Aレジスタのシフト前のビット0の値がCフラグにセットされ、シフト後のビットnはリセットされる。また、シフト命令(右)コード「SRBIT A, n」の実行時には、Aレジスタ内のビットnより上位側に位置する全てのビット(ビットn7～ビットn+1)の値は、シフトせずに維持される。

20

【2381】

また、例えばシフト命令(右)コード「SRBIT D, n」(No. H6)が実行されると、Dレジスタ内の指定ビット位置nのビット(ビットn)から下位側の全ビット(ビットn～ビット0)の値が、下位側に1ビット分シフトするとともに、Dレジスタのシフト前のビット0の値がCフラグにセットされ、シフト後のビットnがリセットされる。また、シフト命令(右)コード「SRBIT D, n」の実行時には、Dレジスタ内のビットnより上位側に位置する全てのビット(ビットn7～ビットn+1)の値は、シフトせずに維持される。

30

【2382】

より具体的には、例えば、レジスタrに格納されている8ビットデータが「11111000B」である場合に、シフト命令(右)コード「SRBIT r, 3」が実行されると、命令実行後のレジスタrに格納される8ビットデータは「11110100B」となり、Cフラグ(CYフラグ)には「0」がセットされる。

【2383】

(3) シフト命令(右) コード「SRBITQ (k), n」(No. H10)
シフト命令(右)コード「SRBITQ (k), n」は、Qレジスタ(拡張レジスタ)を用いてアドレス指定を行うシフト命令(右)コードである。シフト命令(右)コード「SRBITQ (k), n」は、4バイトの命令コードであり、オペランド(OP)として、命令「SRBITQ」(第1OP)、シフト対象データ格納先「(k)」及び8ビットの整数値「n(指定ビット位置)」(第3OP)がこの順で配置(規定)されて構成される。

40

【2384】

シフト命令(右)コード「SRBITQ (k), n」が実行されると、Qレジスタに格納された値(指定メモリの上位側1バイトのアドレス)と8ビットの整数値k(指定メモ

50

リの下位側1バイトのアドレス)とで指定されるアドレス領域(指定メモリ)に格納されている8ビットデータ内の指定ビット位置 n のビット(ビット n)から下位側の全ビット(ビット $n \sim$ ビット0)の値が、下位側に1ビット分シフト(移動)するとともに、当該8ビットデータのシフト前のビット0の値がCフラグにセットされ、シフト後のビット n がリセットされる。なお、シフト命令(右)コード「SRBITQ(k), n 」の実行時には、Qレジスタに格納された値と8ビットの整数値 k とで指定されるアドレス領域に格納されている8ビットデータ内のビット n より上位側に位置する全てのビット(ビット7 $\sim$ ビット $n+1$)の値は、シフトせずに維持される。

【2385】

【コール命令コード及びリターン命令コード】

図343及び図344は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能なコール命令コード及びリターン命令コードの一覧表である。なお、コール命令(呼び出し命令)コードは、例えば、プログラム上において所定の処理プログラム(サブルーチンプログラム)を呼び出すため(処理をサブルーチンプログラムに移動(ジャンプ)させるため)に使用される命令コードであり、リターン命令コードは、例えば、プログラム上において、処理を特定の処理プログラム(メインプログラム)に戻すために使用される命令コードである。

【2386】

図343及び図344に記載の命令番号I1(「CALL mn」)及び命令番号I4(「CALLF mn」)のコール命令コード、並びに、命令番号I7(「RET」)のリターン命令コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。そして、図343及び図344に記載の命令番号I2(「CALLD mn」)、命令番号I3(「CALLE mn」)、命令番号I5(「CALLFD mn」)及び命令番号I6(「CALLFE mn」)のコール命令コード、並びに、命令番号I8(「RETD」)及び命令番号I9(「RETE」)のリターン命令コードは、本実施形態で新たに設けられた命令コードである。なお、コール命令コード及びリターン命令コードの実行時には、その命令コードの種別(命令番号)に関係なく、各フラグの状態は変化しない。

以下、各コール命令コード及び各リターン命令コードについて詳述する。

【2387】

(1) コール命令コード「CALL mn」(No. I1)

コール命令コード「CALL mn」は、3バイトの命令コードであり、オペランド(OP)として、命令「CALL」(第1OP)及び16ビットの整数値「mn」(第2OP)がこの順で配置(規定)されて構成される。なお、このコール命令コードで第2OPとして規定される16ビットの整数値 mn は、ジャンプ先(呼び出し先)のアドレス値(番地)である。

【2388】

コール命令コード「CALL mn」が実行されると、当該命令コードの次の命令コードの番地(戻り先のアドレス値)に対応するPCレジスタ(プログラム・カウンタPC)の値が、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域(スタック領域)にプッシュ(格納)され、ジャンプ先のアドレス値 mn (mn 番地)がPCレジスタに格納され、これにより、処理の実行番地が mn 番地にジャンプする。また、この際、スタックポインタSPの値から2減算した値($SP-2$)がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

【2389】

なお、コール命令コード「CALL mn」の実行時において、PCレジスタ(プログラム・カウンタPC)の値を、現在のスタックポインタSPに格納されている値により指定されるアドレス値から始まる2バイトのメモリ領域にプッシュする際には、PCレジスタの上位側の値(PC h)がスタックポインタSPの値から1減算した値で指定されるメモリ領域($SP-1$)に格納され、PCレジスタの下位側の値(PC l)がスタックポイン

10

20

30

40

50

タSPの値から2減算した値で指定されるメモリ領域(SP-2)に格納される。

【2390】

(2) コール命令コード「CALLD mn」(No. I2)

コール命令コード「CALLD mn」は、3バイトの命令コードであり、オペランド(OP)として、命令「CALLD」(第1OP)及び16ビットの整数値「mn」(第2OP)がこの順で配置(規定)されて構成される。なお、このコール命令コードで第2OPとして規定される16ビットの整数値mnは、ジャンプ先(呼び出し先)のアドレス値(番地)である。

【2391】

コール命令コード「CALLD mn」が実行されると、まず、割込みイネーブルフリップフロップがリセット(「IFF←0」)されてINT割込みが無効(割込み禁止)になり、次いで、当該命令コードの次の命令コードの番地(戻り先のアドレス値)に対応するPCレジスタ(プログラム・カウンタPC)が、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域(スタック領域)にプッシュ(格納)され、ジャンプ先のアドレス値mn(mn番地)がPCレジスタに格納され、これにより、処理の実行番地がmn番地にジャンプする。また、この際、スタックポインタSPの値から2減算した値(SP-2)がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

【2392】

すなわち、コール命令コード「CALLD mn」が実行されると、まず、後述のCPUコントロール命令「DI」(後述の図345参照)と同様の割込み禁止処理が行われ、その後、上述したコール命令コード「CALL mn」(No. I1)と同様の処理が行われる。なお、割込みイネーブルフリップフロップ(割込みフリップフロップ)は、メインCPU101内に設けられたフリップフロップ回路であり、メインCPU101への割込み信号の受付を許可/禁止する回路である。

【2393】

(3) コール命令コード「CALLE mn」(No. I3)

コール命令コード「CALLE mn」は、3バイトの命令コードであり、オペランド(OP)として、命令「CALLE」(第1OP)及び16ビットの整数値「mn」(第2OP)がこの順で配置(規定)されて構成される。なお、このコール命令コードで第2OPとして規定される16ビットの整数値mnは、ジャンプ先(呼び出し先)のアドレス値(番地)である。

【2394】

コール命令コード「CALLE mn」が実行されると、まず、割込みイネーブルフリップフロップがセット(「IFF←1」)されてINT割込みが有効(割込み許可)になり、次いで、当該命令コードの次の命令コードの番地(戻り先のアドレス値)に対応するPCレジスタ(プログラム・カウンタPC)が、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域(スタック領域)にプッシュ(格納)され、ジャンプ先のアドレス値mn(mn番地)がPCレジスタに格納され、これにより、処理の実行番地がmn番地にジャンプする。また、この際、スタックポインタSPの値から2減算した値(SP-2)がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

【2395】

すなわち、コール命令コード「CALLE mn」が実行されると、まず、後述のCPUコントロール命令「EI」(後述の図345参照)と同様の割込み許可処理が行われ、その後、上述したコール命令コード「CALL mn」(No. I1)と同様の処理が行われる。

【2396】

(4) コール命令コード「CALLF mn」(No. I4)

コール命令コード「CALLF mn」は、2バイトの命令コードであり、オペランド(

10

20

30

40

50

OP)として、命令「CALLF」(第1OP)及び16ビットの整数値「mn」(第2OP)がこの順で配置(規定)されて構成される。なお、このコール命令コードで第2OPとして規定される16ビットの整数値mnは、ジャンプ先(呼び出し先)のアドレス値(番地)である。

【2397】

コール命令コード「CALLF mn」が実行されると、当該命令コードの次の命令コードの番地(戻り先のアドレス値)に対応するPCレジスタ(プログラム・カウンタPC)が、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域(スタック領域)にプッシュ(格納)され、ジャンプ先のアドレス値mn(mn番地)がPCレジスタに格納され、これにより、処理の実行番地がmn番地にジャンプする。また、この際、スタックポインタSPの値から2減算した値(SP-2)がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

10

【2398】

なお、コール命令コード「CALLF mn」の実行時において、PCレジスタの値を、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域にプッシュする際には、PCレジスタの上位側の値(PC_h)がスタックポインタSPの値から1減算した値で指定されるアドレス領域(SP-1)に格納され、PCレジスタの下位側の値(PC_l)がスタックポインタSPの値から2減算した値で指定されるアドレス領域(SP-2)に格納される。また、コール命令コード「CALLF mn」の実行時において、ジャンプ先のアドレス値mn(mn番地)がPCレジスタに格納される際には、「00H」～「11H」の値(予め決められた8ビットの整数値)がPCレジスタの上位側(PC_h)に格納され、ジャンプ先のアドレス値mn中の下位側8ビットの整数値nがPCレジスタの下位側(PC_l)に格納される。

20

【2399】

上述のように、コール命令コード「CALLF mn」の処理動作(オペレーション)は、コール命令コード「CALL mn」のそれと同様になるが、コール命令コード「CALL mn」が3バイトの命令コードであるのに対して、コール命令コード「CALLF mn」は2バイトの命令コードとなる。

【2400】

(5) コール命令コード「CALLFD mn」(No. I5)

30

コール命令コード「CALLFD mn」は、2バイトの命令コードであり、オペランド(OP)として、命令「CALLFD」(第1OP)及び16ビットの整数値「mn」(第2OP)がこの順で配置(規定)されて構成される。なお、このコール命令コードで第2OPとして規定される16ビットの整数値mnは、ジャンプ先(呼び出し先)のアドレス値(番地)である。

【2401】

コール命令コード「CALLFD mn」が実行されると、まず、割込みイネーブルフラップフロップがリセット(「IFF←0」)されてINT割込みが無効(割込み禁止)になり、次いで、当該命令コードの次の命令コードの番地(戻り先のアドレス値)に対応するPCレジスタ(プログラム・カウンタPC)が、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域(スタック領域)にプッシュ(格納)され、ジャンプ先のアドレス値mn(mn番地)がPCレジスタに格納され、これにより、処理の実行番地がmn番地にジャンプする。また、この際、スタックポインタSPの値から2減算した値(SP-2)がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

40

【2402】

なお、コール命令コード「CALLFD mn」の実行時において、PCレジスタの値を、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域にプッシュする際には、PCレジスタの上位側の値(PC_h)がスタックポインタSPの値から1減算した値で指定されるアドレス領域(SP-1)に格納され、PCレジスタ

50

の下位側の値（PC L）がスタックポインタSPの値から2減算した値で指定されるアドレス領域（SP-2）に格納される。また、コール命令コード「CALLFD mn」の実行時において、ジャンプ先のアドレス値mn（mn番地）がPCレジスタに格納される際には、「00H」～「11H」の値（予め決められた8ビットの整数値）がPCレジスタの上位側（PC h）に格納され、ジャンプ先のアドレス値mn中の下位側8ビットの整数値nがPCレジスタの下位側（PC L）に格納される。

【2403】

すなわち、コール命令コード「CALLFD mn」が実行されると、まず、後述のCPUコントロール命令「DI」（後述の図345参照）と同様の割込み禁止処理が行われ、その後、上述したコール命令コード「CALLF mn」（No. I4）と同様の処理が行われる。

10

【2404】

（6）コール命令コード「CALLFE mn」（No. I6）

コール命令コード「CALLFE mn」は、2バイトの命令コードであり、オペランド（OP）として、命令「CALLFE」（第1OP）及び16ビットの整数値「mn」（第2OP）がこの順で配置（規定）されて構成される。なお、このコール命令コードで第2OPとして規定される16ビットの整数値mnは、ジャンプ先（呼び出し先）のアドレス値（番地）である。

【2405】

コール命令コード「CALLFE mn」が実行されると、まず、割込みイネーブルフリップフロップがセット（「IFF←1」）されてINT割込みが有効（割込み許可）になり、次いで、当該命令コードの次の命令コードの番地（戻り先のアドレス値）に対応するPCレジスタ（プログラム・カウンタPC）が、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域（スタック領域）にプッシュ（格納）され、ジャンプ先のアドレス値mn（mn番地）がPCレジスタに格納され、これにより、処理の実行番地がmn番地にジャンプする。また、この際、スタックポインタSPの値から2減算した値（SP-2）がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

20

【2406】

なお、コール命令コード「CALLFE mn」の実行時において、PCレジスタの値を、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域にプッシュする際には、PCレジスタの上位側の値（PC h）がスタックポインタSPの値から1減算した値で指定されるアドレス領域（SP-1）に格納され、PCレジスタの下位側の値（PC L）がスタックポインタSPの値から2減算した値で指定されるアドレス領域（SP-2）に格納される。また、コール命令コード「CALLFE mn」の実行時において、ジャンプ先のアドレス値mn（mn番地）がPCレジスタに格納される際には、「00H」～「11H」の値（予め決められた8ビットの整数値）がPCレジスタの上位側（PC h）に格納され、ジャンプ先のアドレス値mn中の下位側8ビットの整数値nがPCレジスタの下位側（PC L）に格納される。

30

【2407】

すなわち、コール命令コード「CALLFE mn」が実行されると、まず、後述のCPUコントロール命令「EI」（後述の図345参照）と同様の割込み許可処理が行われ、その後、上述したコール命令コード「CALLF mn」（No. I4）と同様の処理が行われる。

40

【2408】

（7）リターン命令コード「RET」（No. I7）

リターン命令コード「RET」は、1バイトの命令コードであり、オペランド（OP）として、命令「RET」（第1OP）のみが配置（規定）されて構成される。

【2409】

リターン命令コード「RET」が実行されると、現在のスタックポインタSPに格納され

50

ている値により指示される2バイトのメモリ領域に格納されている値がPCレジスタ（プログラム・カウンタPC）にポップ（セット）され、これにより、処理の実行番地が元（戻り先）のプログラムの実行番地に戻る。また、この際、スタックポインタSPの値から2加算した値（ $SP+2$ ）がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

【2410】

なお、リターン命令コード「RET」の実行時において、スタックポインタSPで指示される2バイトのメモリ領域の値をPCレジスタにポップする際には、スタックポインタSPに格納された値（SP）に対応するアドレスに格納されている値をPCレジスタの下位側（PCL）にポップし、スタックポインタSPに格納された値に1加算した値（ $SP+1$ ）に対応するアドレスに格納されている値をPCレジスタの上位側（PCH）にポップする。

【2411】

（8）リターン命令コード「RETD」（No. I8）

リターン命令コード「RETD」は、1バイトの命令コードであり、オペランド（OP）として、命令「RETD」（第1OP）のみが配置（規定）されて構成される。

【2412】

リターン命令コード「RETD」が実行されると、まず、割込みイネーブルフリップフロップがリセット（「IFF←0」）されてINT割込みが無効（割込み禁止）になり、次いで、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域に格納されている値がPCレジスタ（プログラム・カウンタPC）にポップ（セット）され、これにより、処理の実行番地が元（戻り先）のプログラムの実行番地に戻る。また、この際、スタックポインタSPの値から2加算した値（ $SP+2$ ）がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

【2413】

なお、リターン命令コード「RETD」の実行時において、スタックポインタSPで指示される2バイトのメモリ領域の値をPCレジスタにポップする際には、スタックポインタSPに格納された値（SP）に対応するアドレスに格納されている値をPCレジスタの下位側（PCL）にポップし、スタックポインタSPに格納された値に1加算した値（ $SP+1$ ）に対応するアドレスに格納されている値をPCレジスタの上位側（PCH）にポップする。

【2414】

すなわち、リターン命令コード「RETD」が実行されると、まず、後述のCPUコントロール命令「DI」（後述の図345参照）と同様の割込み禁止処理が行われ、その後、上述したリターン命令コード「RET」（No. I7）と同様の処理が行われる。

【2415】

（9）リターン命令コード「RETE」（No. I9）

リターン命令コード「RETE」は、1バイトの命令コードであり、オペランド（OP）として、命令「RETE」（第1OP）のみが配置（規定）されて構成される。

【2416】

リターン命令コード「RETE」が実行されると、まず、割込みイネーブルフリップフロップがセット（「IFF←1」）されてINT割込みが有効（割込み許可）になり、次いで、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域に格納されている値がPCレジスタ（プログラム・カウンタPC）にポップ（セット）され、これにより、処理の実行番地が元（戻り先）のプログラムの実行番地に戻る。また、この際、スタックポインタSPの値から2加算した値（ $SP+2$ ）がスタックポインタSPに格納され、スタックポインタSPの値が更新される。

【2417】

なお、リターン命令コード「RETE」の実行時において、スタックポインタSPで指示される2バイトのメモリ領域の値をPCレジスタにポップする際には、スタックポインタ

10

20

30

40

50

SPに格納された値（SP）に対応するアドレスに格納されている値をPCレジスタの下位側（PCL）にポップし、スタックポインタSPに格納された値に1加算した値（SP+1）に対応するアドレスに格納されている値をPCレジスタの上位側（PCH）にポップする。

【2418】

すなわち、リターン命令コード「RETE」が実行されると、まず、後述のCPUコントロール命令「EI」（後述の図345参照）と同様の割込み許可処理が行われ、その後、上述したリターン命令コード「RET」（No. I7）と同様の処理が行われる。

【2419】

〔CPUコントロール命令コード〕

図345は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能なCPUコントロール命令コードの一覧表である。

【2420】

図345に記載の命令番号J1（「DI」）及び命令番号J2（「EI」）のCPUコントロール命令コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。なお、CPUコントロール命令コードの実行時には、その命令コードの種別（命令番号）に関係なく、各フラグの状態は変化しない。

以下、各CPUコントロール命令コードについて詳述する。

【2421】

（1）CPUコントロール命令コード「DI」（No. J1）

CPUコントロール命令コード「DI」は、1バイトの命令コードであり、オペランド（OP）として、命令「DI」（第1OP）のみが配置（規定）されて構成される。

【2422】

CPUコントロール命令コード「DI」が実行されると、割込みイネーブルフリップフロップがリセット（「IFF←0」）されてINT割込みが無効（割込み禁止）になる。なお、上記第1及び第2の実施形態の遊技機では、電断時（外部）処理（図76参照）及びメインCPUの制御による割込処理（図158）の実行が一時的に禁止される場合がある。しかしながら、この割込み禁止状態で受け付けた割込み信号は、割込み許可状態になれば、メインCPU101は割込み信号に応じた割込み処理を実行する。すなわち、割込みイネーブルフリップフロップをリセットすること（DI命令）は、一時的に、割込み処理を抑止するだけであり、割込みコントローラ112が出力した割込み信号自体が無効になるわけではない。

【2423】

（2）CPUコントロール命令コード「EI」（No. J2）

CPUコントロール命令コード「EI」は、1バイトの命令コードであり、オペランド（OP）として、命令「EI」（第1OP）のみが配置（規定）されて構成される。

【2424】

CPUコントロール命令コード「EI」が実行されると、割込みイネーブルフリップフロップがセット（「IFF←1」）されてINT割込みが有効（割込み許可）になる。

【2425】

〔論理演算命令コード〕

図346は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能な論理演算命令コードの一覧表である。

【2426】

図346に記載の命令番号K1（「AND A」）～命令番号K5（「OR n」）の論理演算命令コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。

【2427】

なお、論理演算命令コードのうち、論理積（AND）に係る命令コードが実行されたときには、その命令コードの種別（命令番号）に関係なく、Hフラグが「1」セットされ、N

10

20

30

40

50

フラグ及びCフラグ（CYフラグ）が「0」にセットされ、その他の各フラグの状態は演算結果に応じて適宜変化する（設定される）。また、論理演算命令コードのうち、論理和（OR）に係る命令コードが実行されたときには、その命令コードの種別（命令番号）に関係なく、Hフラグ、Nフラグ及びCフラグ（CYフラグ）が「0」にセットされ、その他の各フラグの状態は演算結果に応じて適宜変化する（設定される）。

以下、各論理演算命令コードについて詳述する。

【2428】

（1）論理演算命令コード「AND A」（No. K1）、「AND r」（No. K2）
論理演算命令コード「AND A」は、2バイトの命令コードであり、オペランド（OP）として、命令「AND」（第1OP）及びレジスタ「A」（第2OP）がこの順で配置（規定）されて構成される。また、論理演算命令コード「AND r」は、1バイトの命令コードであり、オペランド（OP）として、命令「AND」（第1OP）及び1バイトのレジスタ「r（Aレジスタ以外の汎用レジスタ）」（第2OP）がこの順で配置（規定）されて構成される。なお、これらの論理演算命令コードにおいて、第2OPとして規定されるレジスタ「A」及び「r」は、Aレジスタに格納されている値と論理積を行う値の格納先である。

10

【2429】

論理演算命令コード「AND A」が実行されると、Aレジスタに格納されている値と、Aレジスタに格納されている値との論理積が算出され、演算結果がAレジスタに格納される。また、論理演算命令コード「AND r」が実行されると、Aレジスタに格納されている値と、レジスタr（Aレジスタ以外の汎用レジスタ）に格納されている値との論理積が算出され、演算結果がAレジスタに格納される。

20

【2430】

図347に、ソースプログラム上で実際に使用され得る論理演算命令コード「AND r」の種類の一覧を示す。論理演算命令コード「AND r」としては、「AND B」（No. K2a）～「AND L」（No. K2f）の6種類の命令コードが設けられている。

【2431】

例えば、論理演算命令コード「AND B」（No. K2a）が実行されると、Aレジスタに格納されている値と、Bレジスタに格納されている値との論理積が算出され、演算結果がAレジスタに格納される。例えば、論理演算命令コード「AND D」（No. K2c）が実行されると、Aレジスタに格納されている値と、Dレジスタに格納されている値との論理積が算出され、演算結果がAレジスタに格納される。また、例えば、論理演算命令コード「AND H」（No. K2e）が実行されると、Aレジスタに格納されている値と、Hレジスタに格納されている値との論理積が算出され、演算結果がAレジスタに格納される。

30

【2432】

（2）論理演算命令コード「AND n」（No. K3）

論理演算命令コード「AND n」は、2バイトの命令コードであり、オペランド（OP）として、命令「AND」（第1OP）及び8ビットの整数値「n」（第2OP）がこの順で配置（規定）されて構成される。なお、この論理演算命令コードにおいて、第2OPとして規定される8ビットの整数値「n」は、Aレジスタに格納されている値と論理積を行う値（8ビットデータ）である。

40

【2433】

論理演算命令コード「AND n」が実行されると、Aレジスタに格納されている値と、8ビットの整数値nとの論理積が算出され、演算結果がAレジスタに格納される。

【2434】

（3）論理演算命令コード「OR r」（No. K4）

論理演算命令コード「OR r」は、1バイトの命令コードであり、オペランド（OP）として、命令「OR」（第1OP）及び1バイトのレジスタ「r（A、B、C、D、E、H又はLレジスタ）」（第2OP）がこの順で配置（規定）されて構成される。なお、こ

50

の論理演算命令コードで第2OPとして規定されるレジスタ「r」は、Aレジスタに格納されている値と論理和を行う値の格納先である。

【2435】

論理演算命令コード「OR r」が実行されると、Aレジスタに格納されている値と、レジスタrに格納されている値との論理和が算出され、演算結果がAレジスタに格納される。

【2436】

図348に、ソースプログラム上で実際に使用され得る論理演算命令コード「OR r」の種類の一覧を示す。論理演算命令コード「OR r」としては、「OR A」(No. K4a)～「OR L」(No. K4g)の7種類の命令コードが設けられている。

【2437】

例えば、論理演算命令コード「OR A」(No. K4a)が実行されると、Aレジスタに格納されている値と、Aレジスタに格納されている値との論理和が算出され、演算結果がAレジスタに格納される。例えば、論理演算命令コード「OR C」(No. K4c)が実行されると、Aレジスタに格納されている値と、Cレジスタに格納されている値との論理和が算出され、演算結果がAレジスタに格納される。また、例えば、論理演算命令コード「OR H」(No. K4f)が実行されると、Aレジスタに格納されている値と、Hレジスタに格納されている値との論理和が算出され、演算結果がAレジスタに格納される。

【2438】

(4) 論理演算命令コード「OR n」(No. K5)

論理演算命令コード「OR n」は、2バイトの命令コードであり、オペランド(OP)として、命令「OR」(第1OP)及び8ビットの整数値「n」(第2OP)がこの順で配置(規定)されて構成される。なお、この論理演算命令コードにおいて、第2OPとして規定される8ビットの整数値「n」は、Aレジスタに格納されている値と論理和を行う値(8ビットデータ)である。

【2439】

論理演算命令コード「OR n」が実行されると、Aレジスタに格納されている値と、8ビットの整数値nとの論理和が算出され、演算結果がAレジスタに格納される。

【2440】

[8ビット算術命令(比較)コード]

図349は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能な8ビット算術命令(比較)コードの一覧表である。

【2441】

図349に記載の命令番号L1(「CP r」)～命令番号L7(「CPQ r, (k)」)の8ビット算術命令(比較)コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。なお、8ビット算術命令(比較)コードの実行時には、その命令コードの種別(命令番号)に関係なく、全てのフラグの状態は演算結果(比較結果)に応じて適宜変化する(設定される)。

以下、各論理演算命令コードについて詳述する。

【2442】

(1) 8ビット算術命令(比較)コード「CP r」(No. L1)

8ビット算術命令(比較)コード「CP r」は、1バイトの命令コードであり、オペランド(OP)として、命令「CP」(第1OP)及び1バイトのレジスタ「r」(Aレジスタ以外の汎用レジスタ)」(第2OP)がこの順で配置(規定)されて構成される。なお、この論理演算命令コードで第2OPとして規定されるレジスタ「r」は、Aレジスタに格納されている値と比較を行う値(8ビットデータ)の格納先である。

【2443】

8ビット算術命令(比較)コード「CP r」が実行されるとAレジスタに格納されている値と、レジスタrに格納されている値(8ビットデータ)とが比較され、その比較結果に応じて、各フラグの状態(セット(1)／リセット(0))が設定される。なお、8ビ

10

20

30

40

50

ット算術命令（比較）コード「CP r」の実行時におけるAレジスタに格納されている値とレジスタrに格納されている値との比較処理では、Aレジスタに格納されている値からレジスタrに格納されている値を減算する処理（A-r）が行われる。そして、この比較（減算）処理の結果に応じて、各フラグの状態が設定される。

【2444】

図350に、ソースプログラム上で実際に使用され得る8ビット算術命令（比較）コード「CP r」の種類の一覧を示す。8ビット算術命令（比較）コード「CP r」としては、「CP B」（No. L1a）～「CP L」（No. L1f）の6種類の命令コードが設けられている。

【2445】

例えば、8ビット算術命令（比較）コード「CP B」（No. L1a）が実行されると、Aレジスタに格納されている値と、Bレジスタに格納されている値とが比較され、その比較結果に応じて、各フラグの状態が設定される。また、例えば、8ビット算術命令（比較）コード「CP H」（No. L1e）が実行されると、Aレジスタに格納されている値と、Hレジスタに格納されている値とが比較され、その比較結果に応じて、各フラグの状態が設定される。

【2446】

（2）8ビット算術命令（比較）コード「CP n」（No. L2）

8ビット算術命令（比較）コード「CP n」は、2バイトの命令コードであり、オペランド（OP）として、命令「CP」（第1OP）及び8ビットの整数値「n」（第2OP）がこの順で配置（規定）されて構成される。なお、この論理演算命令コードで第2OPとして規定される8ビットの整数値「n」は、Aレジスタに格納されている値と比較する値（8ビットデータ）である。

【2447】

8ビット算術命令（比較）コード「CP n」が実行されると、Aレジスタに格納されている値と、8ビットの整数値n（8ビットデータ）とが比較され、その比較結果に応じて、各フラグの状態（セット（1）／リセット（0））が設定される。なお、8ビット算術命令（比較）コード「CP n」の実行時におけるAレジスタに格納されている値と8ビットの整数値nとの比較処理では、Aレジスタに格納されている値から8ビットの整数値nを減算する処理（A-n）が行われる。そして、この比較（減算）処理の結果に応じて、各フラグの状態が設定される。

【2448】

（3）8ビット算術命令（比較）コード「CP（DE）」（No. L3）

8ビット算術命令（比較）コード「CP（DE）」は、1バイトの命令コードであり、オペランド（OP）として、命令「CP」（第1OP）及び格納先「（DE）」（第2OP）がこの順で配置（規定）されて構成される。なお、この論理演算命令コードで第2OPとして規定される格納先「（DE）」は、Aレジスタに格納されている値と比較を行う値（8ビットデータ）の格納先（DEレジスタに格納されている値で指定されるアドレス領域）である。

【2449】

8ビット算術命令（比較）コード「CP（DE）」が実行されると、Aレジスタに格納されている値と、DEレジスタで指定されるアドレス領域（指定メモリ）に格納されている値（8ビットデータ）とが比較され、その比較結果に応じて、各フラグの状態（セット（1）／リセット（0））が設定される。なお、8ビット算術命令（比較）コード「CP（DE）」の実行時におけるAレジスタに格納されている値とDEレジスタで指定されるアドレス領域に格納されている値との比較処理では、Aレジスタに格納されている値からDEレジスタで指定されるアドレス領域に格納されている値を減算する処理（A-（DE））が行われる。そして、この比較（減算）処理の結果に応じて、各フラグの状態が設定される。

【2450】

10

20

30

40

50

(4) 8ビット算術命令(比較)コード「CP (HL)」(No. L4)

8ビット算術命令(比較)コード「CP (HL)」は、2バイトの命令コードであり、オペランド(OP)として、命令「CP」(第1OP)及び格納先「(HL)」(第2OP)がこの順で配置(規定)されて構成される。なお、この論理演算命令コードで第2OPとして規定される格納先「(HL)」は、Aレジスタに格納されている値と比較する値(8ビットデータ)の格納先(HLレジスタに格納されている値で指定されるアドレス領域)である。

【2451】

8ビット算術命令(比較)コード「CP (HL)」が実行されると、Aレジスタに格納されている値と、HLレジスタで指定されるアドレス領域(指定メモリ)に格納されている値とが比較され、その比較結果に応じて、各フラグの状態(セット(1)/リセット(0))が設定される。なお、8ビット算術命令(比較)コード「CP (HL)」の実行時におけるAレジスタに格納されている値とHLレジスタで指定されるアドレス領域に格納されている値との比較処理では、Aレジスタに格納されている値からHLレジスタで指定されるアドレス領域に格納されている値を減算する処理($A - (HL)$)が行われる。そして、この比較(減算)処理の結果に応じて、各フラグの状態が設定される。

【2452】

(5) 8ビット算術命令(比較)コード「CP r, (HL)」(No. L5)

8ビット算術命令(比較)コード「CP r, (HL)」は、2バイトの命令コードであり、オペランド(OP)として、命令「CP」(第1OP)、第1格納先「r」(第2OP)及び第2格納先「(HL)」(第3OP)がこの順で配置(規定)されて構成される。なお、この論理演算命令コードにおいて、第2OPとして規定される第1格納先「r」は、Aレジスタ以外の汎用レジスタであり、第3OPとして規定される第2格納先「(HL)」は、レジスタrに格納されている値と比較する値(8ビットデータ)の格納先(HLレジスタに格納されている値で指定されるアドレス領域)である。

【2453】

8ビット算術命令(比較)コード「CP r, (HL)」が実行されると、レジスタrに格納されている値と、HLレジスタで指定されるアドレス領域(指定メモリ)に格納されている値(8ビットデータ)とが比較され、その比較結果に応じて、各フラグの状態(セット(1)/リセット(0))が設定される。なお、8ビット算術命令(比較)コード「CP r, (HL)」の実行時におけるレジスタrに格納されている値とHLレジスタで指定されるアドレス領域に格納されている値との比較処理では、レジスタrに格納されている値からHLレジスタで指定されるアドレス領域に格納されている値を減算する処理($r - (HL)$)が行われる。そして、この比較(減算)処理の結果に応じて、各フラグの状態が設定される。

【2454】

図351に、ソースプログラム上で実際に使用され得る8ビット算術命令(比較)コード「CP r, (HL)」の種類の一覧を示す。8ビット算術命令(比較)コード「CP r, (HL)」としては、「CP B, (HL)」(No. L5a)～「CP L, (HL)」(No. L5f)の6種類の命令コードが設けられている。

【2455】

例えば、8ビット算術命令(比較)コード「CP B, (HL)」(No. L5a)が実行されると、Bレジスタに格納されている値と、HLレジスタで指定されるアドレス領域に格納されている値とが比較され、その比較結果に応じて、各フラグの状態が設定される。また、例えば、8ビット算術命令(比較)コード「CP H, (HL)」(No. L5e)が実行されると、Hレジスタに格納されている値と、HLレジスタで指定されるアドレス領域に格納されている値とが比較され、その比較結果に応じて、各フラグの状態が設定される。

【2456】

(6) 8ビット算術命令(比較)コード「CPQ (k)」(No. L6)

8ビット算術命令（比較）コード「CPQ (k)」は、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う8ビット算術命令（比較）コードである。8ビット算術命令（比較）コード「CPQ (k)」は、2バイトの命令コードであり、オペランド（OP）として、命令「CPQ」（第1OP）及び比較対象値格納先「(k)」（第2OP）がこの順で配置（規定）されて構成される。なお、この論理演算命令コードにおいて、第2OPとして規定される比較対象値格納先「(k)」は、Aレジスタに格納された値と比較を行う値（比較対象値）の格納先（指定メモリ）の下位側アドレス（kは8ビットの整数値）である。

【2457】

8ビット算術命令（比較）コード「CPQ (k)」が実行されると、Aレジスタに格納された値と、Qレジスタに格納された値（指定メモリの上位側アドレス）及び8ビットの整数値k（指定メモリの下位側アドレス）により指定されるアドレス領域（指定メモリ）に格納されている値とが比較され、その比較結果に応じて、各フラグの状態（セット（1）／リセット（0））が設定される。なお、8ビット算術命令（比較）コード「CPQ (k)」の実行時におけるAレジスタに格納された値と、Qレジスタに格納された値及び8ビットの整数値kにより指定されるアドレス領域に格納されている値（比較対象値）との比較処理では、Aレジスタに格納された値から比較対象値を減算する処理（ $A - (Qk)$ ）が行われる。そして、この比較（減算）処理の結果に応じて、各フラグの状態が設定される。

【2458】

(7) 8ビット算術命令（比較）コード「CPQ r, (k)」(No. L7)

8ビット算術命令（比較）コード「CPQ r, (k)」は、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う8ビット算術命令（比較）コードである。8ビット算術命令（比較）コード「CPQ r, (k)」は、3バイトの命令コードであり、オペランド（OP）として、命令「CPQ」（第1OP）、1バイトのレジスタ「r（Aレジスタ以外の汎用レジスタ）」（第2OP）及び比較対象値格納先「(k)」（第3OP）がこの順で配置（規定）されて構成される。なお、この論理演算命令コードにおいて、第3OPとして規定される比較対象値格納先「(k)」は、レジスタrに格納されている値と比較する値（比較対象値）の格納先（指定メモリ）の下位側1バイトのアドレス（kは8ビットの整数値）である。

【2459】

8ビット算術命令（比較）コード「CPQ r, (k)」が実行されると、レジスタrに格納されている値と、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）及び8ビット整数値k（指定メモリの下位側1バイトのアドレス）により指定されるアドレス領域（指定メモリ）に格納されている値とが比較され、その比較結果に応じて、各フラグの状態（セット（1）／リセット（0））が設定される。なお、8ビット算術命令（比較）コード「CPQ r, (k)」の実行時におけるレジスタrに格納されている値と、Qレジスタに格納された値及び8ビットの整数値kにより指定されるアドレス領域に格納されている値（比較対象値）との比較処理では、レジスタrに格納されている値から比較対象値を減算する処理（ $r - (Qk)$ ）が行われる。そして、この比較（減算）処理の結果に応じて、各フラグの状態が設定される。

【2460】

図352に、ソースプログラム上で実際に使用され得る8ビット算術命令（比較）コード「CPQ r, (k)」の種類の一覧を示す。8ビット算術命令（比較）コード「CPQ r, (k)」としては、「CPQ B, (k)」(No. L7a)～「CPQ L, (k)」(No. L7f)の6種類の命令コードが設けられている。

【2461】

例えば、8ビット算術命令（比較）コード「CPQ B, (k)」(No. L7a)が実行されると、Bレジスタに格納されている値と、Qレジスタに格納された値及び8ビットの整数値kにより指定されるアドレス領域（指定メモリ）に格納されている値とが比較さ

10

20

30

40

50

れ、その比較結果に応じて、各フラグの状態（セット（１）／リセット（０））が設定される。また、例えば、８ビット算術命令（比較）コード「CPQH, (k)」(No. L7e) が実行されると、Hレジスタに格納されている値と、Qレジスタに格納された値及び８ビットの整数値kにより指定されるアドレス領域（指定メモリ）に格納されている値とが比較され、その比較結果に応じて、各フラグの状態（セット（１）／リセット（０））が設定される。

【2462】

【１６ビット算術命令（比較）コード】

図３５３は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能な１６ビット算術命令（比較）コードの一覧表である。

【2463】

図３５３に記載の命令番号M1（「CPBC, mn」）～命令番号M3（「CPHL, mn」）の１６ビット算術命令（比較）コードは、本実施形態だけでなく、上記第１及び第２実施形態の遊技機でも使用可能な命令コードである。なお、１６ビット算術命令（比較）コードの実行時には、その命令コードの種別（命令番号）に関係なく、全てのフラグの状態は演算結果（比較結果）に応じて適宜変化する（設定される）。

以下、各論理演算命令コードについて詳述する。

【2464】

（１）１６ビット算術命令（比較）コード「CPBC, mn」(No. M1)

１６ビット算術命令（比較）コード「CPBC, mn」は、４バイトの命令コードであり、オペランド（OP）として、命令「CP」（第１OP）、BCレジスタ（第２OP）及び１６ビットの整数値「mn」（第３OP）がこの順で配置（規定）されて構成される。

【2465】

１６ビット算術命令（比較）コード「CPBC, mn」が実行されると、BCレジスタに格納されている値と、１６ビットの整数値mn（１６ビットデータ）とが比較され、その比較結果に応じて、各フラグの状態（セット（１）／リセット（０））が設定される。なお、１６ビット算術命令（比較）コード「CPBC, mn」の実行時におけるBCレジスタに格納されている値と１６ビットの整数値mnとの比較処理では、BCレジスタに格納されている値から１６ビットの整数値mnを減算する処理（BC-mn）が行われる。そして、この比較（減算）処理の結果に応じて、各フラグの状態が設定される。

【2466】

（２）１６ビット算術命令（比較）コード「CPDE, mn」(No. M2)

１６ビット算術命令（比較）コード「CPDE, mn」は、４バイトの命令コードであり、オペランド（OP）として、命令「CP」（第１OP）、DEレジスタ（第２OP）及び１６ビットの整数値「mn」（第３OP）がこの順で配置（規定）されて構成される。

【2467】

１６ビット算術命令（比較）コード「CPDE, mn」が実行されると、DEレジスタに格納されている値と、１６ビットの整数値mn（１６ビットデータ）とが比較され、その比較結果に応じて、各フラグの状態（セット（１）／リセット（０））が設定される。なお、１６ビット算術命令（比較）コード「CPDE, mn」の実行時におけるDEレジスタに格納されている値と１６ビットの整数値mnとの比較処理では、DEレジスタに格納されている値から１６ビットの整数値mnを減算する処理（DE-mn）が行われる。そして、この比較（減算）処理の結果に応じて、各フラグの状態が設定される。

【2468】

（３）１６ビット算術命令（比較）コード「CPHL, mn」(No. M3)

１６ビット算術命令（比較）コード「CPHL, mn」は、４バイトの命令コードであり、オペランド（OP）として、命令「CP」（第１OP）、HLレジスタ（第２OP）及び１６ビットの整数値「mn」（第３OP）がこの順で配置（規定）されて構成される。

【2469】

１６ビット算術命令（比較）コード「CPHL, mn」が実行されると、HLレジスタ

に格納されている値と、16ビットの整数値 mn （16ビットデータ）とが比較され、その比較結果に応じて、各フラグの状態（セット（1）／リセット（0））が設定される。なお、16ビット算術命令（比較）コード「 $CPHL, mn$ 」の実行時におけるHLレジスタに格納されている値と16ビットの整数値 mn との比較処理では、HLレジスタに格納されている値から16ビットの整数値 mn を減算する処理（ $HL - mn$ ）が行われる。そして、この比較（減算）処理の結果に応じて、各フラグの状態が設定される。

【2470】

〔8ビット算術命令（加算）コード及び8ビット算術命令（減算）コード〕

図354は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能な8ビット算術命令（加算）コード及び8ビット算術命令（減算）コードの一覧表である。

10

【2471】

図354に記載の命令番号N1（「 $INC\ r$ 」）～命令番号N4（「 $DECQ\ (k)$ 」）の8ビット算術命令（加算）コード及び8ビット算術命令（減算）コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。なお、8ビット算術命令（加算）コードの実行時には、その命令コードの種別（命令番号）に関係なく、Nフラグは「0」にセットされ、Cフラグの状態は維持され（変化なし）、Nフラグ及びCフラグ以外の各フラグの状態は演算結果（加算／減算結果）に応じて適宜変化する（設定される）。また、8ビット算術命令（減算）コードの実行時には、その命令コードの種別（命令番号）に関係なく、Nフラグは「1」にセットされ、Cフラグの状態は維持され（変化なし）、Nフラグ及びCフラグ以外の各フラグの状態は演算結果（加算／減算結果）に応じて適宜変化する（設定される）。

20

以下、各8ビット算術命令（加算）コード及び各8ビット算術命令（減算）コードについて詳述する。

【2472】

（1）8ビット算術命令（加算）コード「 $INC\ r$ 」（No. N1）

8ビット算術命令（加算）コード「 $INC\ r$ 」は、1バイトの命令コードであり、オペランド（OP）として、命令「 INC 」（第1OP）及び演算対象値格納先「 r 」（第2OP）がこの順で配置（規定）されて構成される。なお、この8ビット算術命令（加算）コードにおいて、第2OPとして規定される演算対象値格納先「 r 」は、加算演算の対象となる値（8ビットデータ）の格納先となる1バイトのレジスタ r である。

30

【2473】

8ビット算術命令（加算）コード「 $INC\ r$ 」が実行されると、レジスタ r に格納されている値（8ビットデータ）に1が加算され、該加算後の値が、レジスタ r に格納される。

【2474】

図355に、ソースプログラム上で実際に使用され得る8ビット算術命令（加算）コード「 $INC\ r$ 」の種類の一覧を示す。8ビット算術命令（加算）コード「 $INC\ r$ 」としては、「 $INC\ A$ 」（No. N1a）～「 $INC\ L$ 」（No. N1g）の7種類の命令コードが設けられている。

【2475】

例えば、8ビット算術命令（加算）コード「 $INC\ A$ 」（No. N1a）が実行されると、Aレジスタに格納されている値に1が加算され、該加算後の値が、Aレジスタに格納される。また、例えば、8ビット算術命令（加算）コード「 $INC\ H$ 」（No. N1f）が実行されると、Hレジスタに格納されている値に1が加算され、該加算後の値が、Hレジスタに格納される。

40

【2476】

（2）8ビット算術命令（加算）コード「 $INCQ\ (k)$ 」（No. N2）

8ビット算術命令（加算）コード「 $INCQ\ (k)$ 」は、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う8ビット算術命令（加算）コードである。8ビット算術命令（加算）コード「 $INCQ\ (k)$ 」は、3バイトの命令コードであり、オペランド（OP）として、命令「 $INCQ$ 」（第1OP）及び演算対象値格納先「 (k) 」（第2OP

50

）がこの順で配置（規定）されて構成される。なお、この8ビット算術命令（加算）コードにおいて、第2OPとして規定される演算対象値格納先「(k)」は、加算演算の対象となる値（8ビットデータ）の格納先（指定メモリ）の下位側1バイトのアドレス（kは8ビットの整数値）である。

【2477】

8ビット算術命令（加算）コード「INCQ (k)」が実行されると、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と8ビットの整数値k（指定メモリの下位側1バイトのアドレス）とで指定されるアドレス領域（指定メモリ）に格納されている値（8ビットデータ）に1が加算され、該加算後の値が、Qレジスタに格納された値と8ビット整数値kとで指定されるアドレス領域に格納される。

10

【2478】

（3）8ビット算術命令（減算）コード「DEC r」（No. N3）

8ビット算術命令（減算）コード「DEC r」は、1バイトの命令コードであり、オペランド（OP）として、命令「DEC」（第1OP）及び演算対象値格納先「r」（第2OP）がこの順で配置（規定）されて構成される。なお、この8ビット算術命令（減算）コードにおいて、第2OPとして規定される演算対象値格納先「r」は、減算演算の対象となる値（8ビットデータ）の格納先となる1バイトのレジスタrである。

【2479】

8ビット算術命令（減算）コード「DEC r」が実行されると、レジスタrに格納されている値（8ビットデータ）から1が減算され、該減算後の値が、レジスタrに格納される。

20

【2480】

図355に、ソースプログラム上で実際に使用され得る8ビット算術命令（減算）コード「DEC r」の種類の一覧を示す。8ビット算術命令（減算）コード「DEC r」としては、「DEC A」（No. N3a）～「DEC L」（No. N3g）の7種類の命令コードが設けられている。

【2481】

例えば、8ビット算術命令（減算）コード「DEC A」（No. N3a）が実行されると、Aレジスタに格納されている値から1が減算され、該減算後の値が、Aレジスタに格納される。また、例えば、8ビット算術命令（減算）コード「DEC H」（No. N3f）が実行されると、Hレジスタに格納されている値から1が減算され、該減算後の値が、Hレジスタに格納される。

30

【2482】

（4）8ビット算術命令（減算）コード「DECQ (k)」(No. N4)

8ビット算術命令（減算）コード「DECQ (k)」は、Qレジスタ（拡張レジスタ）を用いてアドレス指定を行う8ビット算術命令（減算）コードである。8ビット算術命令（減算）コード「DECQ (k)」は、3バイトの命令コードであり、オペランド（OP）として、命令「DECQ」（第1OP）及び演算対象値格納先「(k)」(第2OP)がこの順で配置（規定）されて構成される。なお、この8ビット算術命令（減算）コードにおいて、第2OPとして規定される演算対象値格納先「(k)」は、減算演算の対象となる値（8ビットデータ）の格納先（指定メモリ）の下位側1バイトのアドレス（kは8ビットの整数値）である。

40

【2483】

8ビット算術命令（減算）コード「DECQ (k)」が実行されると、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と8ビットの整数値k（指定メモリの下位側1バイトのアドレス）とで指定されるアドレス領域（指定メモリ）に格納されている値（8ビットデータ）から1が減算され、該減算後の値が、Qレジスタに格納された値と8ビット整数値kとで指定されるアドレス領域に格納される。

【2484】

【16ビット算術命令（加算）コード及び16ビット算術命令（減算）コード】

50

図356は、本実施形態の遊技機で実行される各種ソースプログラムで使用可能な16ビット算術命令（加算）コード及び16ビット算術命令（減算）コードの一覧表である。

【2485】

図356に記載の命令番号01（「INC BC」）～命令番号010（「ADDWB DE, A」）の16ビット算術命令（加算）コード及び16ビット算術命令（減算）コードは、本実施形態だけでなく、上記第1及び第2実施形態の遊技機でも使用可能な命令コードである。なお、16ビット算術命令（加算）コード「INC BC」～「INC HL」、及び、16ビット算術命令（減算）コード「DEC BC」～「DEC HL」の実行時には、その命令コードの種別（命令番号）に関係なく、TZフラグの状態は演算結果（加算／減算結果）に応じて適宜変化し（設定され）、その他の各フラグの状態は維持される（変化しない）。また、16ビット算術命令（加算）コード「ADDWB HL, A」～「ADDWB DE, A」の実行時には、その命令コードの種別（命令番号）に関係なく、Nフラグに「0」がセットされ、その他の各フラグの状態は演算結果（加算／減算結果）に応じて適宜変化する（設定される）。

【2486】

（1）16ビット算術命令（加算）コード「INC BC」（No. 01）、「INC DE」（No. 02）、「INC HL」（No. 03）

16ビット算術命令（加算）コード「INC BC」、「INC DE」及び「INC HL」はそれぞれ、1バイトの命令コードであり、オペランド（OP）として、命令「INC」（第1OP）及び演算対象値格納先「ss（BC、DE及びHLレジスタ）」（第2OP）がこの順で配置（規定）されて構成される。なお、この16ビット算術命令（加算）コードにおいて、第2OPとして規定される演算対象値格納先「ss」は、加算演算の対象となる値（16ビットデータ）が格納されているペアレジスタssである。

【2487】

16ビット算術命令（加算）コード「INC BC」が実行されると、BCレジスタに格納されている値（16ビットデータ）に1が加算され、該加算後の値が、BCレジスタに格納される。16ビット算術命令（加算）コード「INC DE」が実行されると、DEレジスタに格納されている値に1が加算され、該加算後の値が、DEレジスタに格納される。また、16ビット算術命令（加算）コード「INC HL」が実行されると、HLレジスタに格納されている値に1が加算され、該加算後の値が、HLレジスタに格納される。

【2488】

（2）16ビット算術命令（減算）コード「DEC BC」（No. 04）、「DEC DE」（No. 05）、「DEC HL」（No. 06）

16ビット算術命令（減算）コード「DEC BC」、「DEC DE」及び「DEC HL」はそれぞれ、1バイトの命令コードであり、オペランド（OP）として、命令「DEC」（第1OP）及び演算対象値格納先「ss（BC、DE及びHLレジスタ）」（第2OP）がこの順で配置（規定）されて構成される。なお、この16ビット算術命令（減算）コードにおいて、第2OPとして規定される演算対象値格納先「ss」は、減算演算の対象となる値（16ビットデータ）が格納されているペアレジスタssである。

【2489】

16ビット算術命令（減算）コード「DEC BC」が実行されると、BCレジスタに格納されている値（16ビットデータ）から1が減算され、該減算後の値が、BCレジスタに格納される。16ビット算術命令（減算）コード「DEC DE」が実行されると、DEレジスタに格納された値から1が減算され、該減算後の値が、DEレジスタに格納される。また、16ビット算術命令（減算）コード「DEC HL」が実行されると、HLレジスタに格納された値から1が減算され、該減算後の値が、HLレジスタに格納される。

【2490】

（3）16ビット算術命令（加算）コード「ADDWB HL, A」（No. 07）、「ADDWB HL, r」（No. 08）

16ビット算術命令（加算）コード「ADDWB HL, A」は、1バイトの命令コード

であり、オペランド（OP）として、命令「ADDWB」（第1OP）、第1演算対象値格納先「HL」（第2OP）及び第2演算対象値格納先「A」（第3OP）がこの順で配置（規定）されて構成される。また、16ビット算術命令（加算）コード「ADDWB HL, r」は、2バイトの命令コードであり、オペランド（OP）として、命令「ADDWB」（第1OP）、第1演算対象値格納先「HL」（第2OP）及び第2演算対象値格納先「r（A、H及びLレジスタ以外の汎用レジスタ）」（第3OP）がこの順で配置（規定）されて構成される。なお、これらの16ビット算術命令（加算）コードにおいて、第2OPとして規定される第1演算対象値格納先「HL」は、加算演算の対象となる一方の値（16ビットデータ）が格納されているHLレジスタであり、第3OPとして規定される第2演算対象値格納先「A」及び「r」は、それぞれ、加算演算の対象となる他方の値が格納されているAレジスタ、並びに、A、H及びLレジスタ以外の汎用レジスタである。

10

【2491】

16ビット算術命令（加算）コード「ADDWB HL, A」が実行されると、HLレジスタに格納されている値にAレジスタに格納されている値が加算され、該加算後の値が、HLレジスタに格納される。また、16ビット算術命令（加算）コード「ADDWB HL, r」が実行されると、HLレジスタに格納されている値に、レジスタr（B、C、D又はEレジスタ）に格納されている値が加算され、該加算後の値が、HLレジスタに格納される。

【2492】

20

図357に、ソースプログラム上で実際に使用され得る16ビット算術命令（加算）コード「ADDWB HL, r」の種類の一覧を示す。16ビット算術命令（加算）コード「ADDWB HL, r」としては、「ADDWB HL, B」（No. 08a）～「ADDWB HL, E」（No. 08d）の4種類の命令コードが設けられている。

【2493】

例えば、論理演算命令コード「ADDWB HL, B」（No. 08a）が実行されると、HLレジスタに格納されたている値にBレジスタに格納されている値が加算され、該加算後の値が、HLレジスタに格納される。また、例えば、論理演算命令コード「ADDWB HL, D」（No. 08c）が実行されると、HLレジスタに格納されている値にDレジスタに格納されている値が加算され、該加算後の値が、HLレジスタに格納される。

30

【2494】

（4）16ビット算術命令（加算）コード「ADDWB BC, A」（No. 09）、「ADDWB DE, A」（No. 010）

16ビット算術命令（加算）コード「ADDWB BC, A」は、2バイトの命令コードであり、オペランド（OP）として、命令「ADDWB」（第1OP）、第1演算対象値格納先「BC」（第2OP）及び第2演算対象値格納先「A」（第3OP）がこの順で配置（規定）されて構成される。また、16ビット算術命令（加算）コード「ADDWB DE, A」は、2バイトの命令コードであり、オペランド（OP）として、命令「ADDWB」（第1OP）、第1演算対象値格納先「DE」（第2OP）及び第2演算対象値格納先「A」（第3OP）がこの順で配置（規定）されて構成される。なお、これらの16ビット算術命令（加算）コードにおいて、第2OPとして規定される第1演算対象値格納先「BC」及び「DE」は、加算演算の対象となる一方の値（16ビットデータ）が格納されているBC及びDEレジスタであり、第3OPとして規定される第2演算対象値格納先「A」は、加算演算の対象となる他方の値が格納されているAレジスタである。

40

【2495】

16ビット算術命令（加算）コード「ADDWB BC, A」が実行されると、BCレジスタに格納されている値（16ビットデータ）にAレジスタに格納されている値が加算され、該加算後の値が、BCレジスタに格納される。また、16ビット算術命令（加算）コード「ADDWB DE, A」が実行されると、DEレジスタに格納されている値にAレジスタに格納されている値が加算され、該加算後の値が、DEレジスタに格納される。

50

【2496】

<各種ソースプログラム例>

上記図304～図357で説明した各種命令コードの一部は、上記第1及び第2実施形態において説明したように、遊技の進行を制御するための各種ソースプログラムにおいて適宜使用されている。そして、具体的な説明は省略するが、本実施形態で新たに設けられた命令コード（図304～図357中の「新」欄において○印が記載された命令コード）もまた、遊技の進行を制御するための各種ソースプログラムにおいて適宜採用することができる。

【2497】

ここで、上記図304～図357で説明した本実施形態で新たに設けられた命令コードを使用した各種ソースプログラム例（プログラム例1～29）を説明する。なお、各プログラム例の説明では、本実施形態で新たに設けられた命令コード（新命令コード）の機能の優位性を明確にするため、新命令コードを使用しないで同様の機能（処理動作）を作用させる場合のプログラム（上記第1及び第2の実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム）も比較例として示す。また、下記各種プログラムの実行処理は、メインCPU101により行われる。

【2498】

[エクステンジ命令コードを使用したプログラム例]

まず、本実施形態で新たに設けられたエクステンジ命令コードを使用した各種プログラム例を説明する。

【2499】

(1) プログラム例1

プログラム例1では、HLレジスタを使用して、Aレジスタに格納されている値と、指定されたメインRAM103内のアドレス領域（wARE A0）に格納されている値とを交換する機能（処理動作）を実現するためのプログラム例を説明する。図358A及び図358Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例1-A、及び、本実施形態で新たに設けられたエクステンジ命令コードを用いて生成されたプログラム例1-Bを示す。

【2500】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図358Aに示すように、まず、ソースコード「LD HL, wARE A0」を実行して、交換対象となる一方の値が格納されているメインRAM103内の領域のアドレス（wARE A0）をHLレジスタにセット（ロード）する。次いで、ソースコード「LD B, (HL)」を実行して、HLレジスタにセットされたアドレス（wARE A0）に格納されている値（交換対象となる一方の値）を、Bレジスタにセット（ロード）する。次いで、ソースコード「LD (HL), A」を実行して、Aレジスタに格納されている値（交換対象となる他方の値）を、HLレジスタにセットされたアドレス（wARE A0）の領域（指定メモリ）にセット（ロード）する。そして、ソースコード「LD A, B」を実行して、Bレジスタに格納されている値（交換対象となる一方の値）を、Aレジスタにセット（ロード）する。

【2501】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した4つのソースコードを順次実行することにより、Aレジスタに格納されている値と、指定アドレス領域（wARE A0）に格納されている値とを交換することができる。なお、このソースプログラムでは、ソースコード「LD HL, wARE A0」、「LD B, (HL)」、「LD (HL), A」及び「LD A, B」のバイト数（容量）は、それぞれ、3バイト、1バイト、1バイト及び1バイトであり、プログラム例1-Aのプログラムの合計バイト数は、6バイトとなる。

【2502】

一方、本実施形態で新たに設けられたエクステンジ命令コードを用いた場合、図358

10

20

30

40

50

Bに示すように、まず、ソースコード「LD HL, wAREA0」を実行して、交換対象となる一方の値が格納されている領域のアドレス(wAREA0)をHLレジスタにセット(ロード)する。次いで、ソースコード「EX A, (HL)」(新命令コード:図306中のNo. A10k)を実行して、Aレジスタに格納されている値と、HLレジスタにセットされたアドレス(wAREA0)に格納されている値とを交換する。

【2503】

本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、Aレジスタに格納されている値と、指定アドレス領域(wAREA0)に格納されている値とを交換することができる。なお、このソースプログラムでは、ソースコード「LD HL, wAREA0」及び「EX A, (HL)」のバイト数(容量)は、それぞれ、3バイト及び2バイトであり、プログラム例1-Bのプログラムの合計バイト数は、5バイトとなる。

【2504】

上述のように、プログラム例1では、本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合、それを用いない場合に比べて、同じ機能(処理動作)を実現するためのプログラムの容量(バイト数)を減らすことができる。

【2505】

(2) プログラム例2

プログラム例2では、インデックス・レジスタIXを使用して、Bレジスタに格納されている値と、指定されたメインRAM103内のアドレス領域(wAREA0)に格納されている値とを交換する機能(処理動作)を実現するためのプログラム例を説明する。図359A及び図359Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例2-A、及び、本実施形態で新たに設けられたエクスチェンジ命令コードを用いて生成されたプログラム例2-Bを示す。

【2506】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図359Aに示すように、まず、ソースコード「LD IX, wAREA0」を実行して、交換対象となる一方の値が格納されているメインRAM103内の領域のアドレス(wAREA0)をインデックス・レジスタIXにセット(ロード)する。次いで、ソースコード「LD A, (IX)」を実行して、インデックス・レジスタIXにセットされたアドレス(wAREA0)に格納されている値(交換対象となる一方の値)を、Aレジスタにセット(ロード)する。次いで、ソースコード「LD (IX), B」を実行して、Bレジスタに格納されている値(交換対象となる他方の値)を、インデックス・レジスタIXにセットされたアドレス(wAREA0)の領域にセット(ロード)する。そして、ソースコード「LD B, A」を実行して、Aレジスタに格納されている値(交換対象となる一方の値)を、Bレジスタにセット(ロード)する。

【2507】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した4つのソースコードを順次実行することにより、Bレジスタに格納されている値と、指定アドレス領域(wAREA0)に格納されている値とを交換することができる。なお、このソースプログラムでは、ソースコード「LD IX, wAREA0」、「LD A, (IX)」、「LD (IX), B」及び「LD B, A」のバイト数(容量)は、それぞれ、4バイト、2バイト、2バイト及び1バイトであり、プログラム例2-Aのプログラムの合計バイト数は、9バイトとなる。

【2508】

一方、本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合、図359Bに示すように、まず、ソースコード「LD IX, wAREA0」を実行して、交換対象となる一方の値が格納されている領域のアドレス(wAREA0)をインデックス・レジスタIXにセット(ロード)する。次いで、ソースコード「EX B, (IX)」(新命令コード:図307中のNo. A18b)を実行して、Bレジスタに格納されている値

10

20

30

40

50

と、インデックス・レジスタ IX にセットされたアドレス ($wAREAO$) に格納されている値とを交換する。

【2509】

本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、 B レジスタに格納されている値と、指定アドレス領域 ($wAREAO$) に格納されている値とを交換することができる。なお、このソースプログラムでは、ソースコード「 $LD\ IX, wAREAO$ 」及び「 $EX\ B, (IX)$ 」のバイト数 (容量) は、それぞれ、4バイト及び2バイトであり、プログラム例2-Bのプログラムの合計バイト数は、6バイトとなる。

【2510】

上述のように、プログラム例2においても、本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合、それを用いない場合に比べて、同じ機能 (処理動作) を実現するためのプログラムの容量 (バイト数) を減らすことができる。

【2511】

(3) プログラム例3

プログラム例3では、 Q レジスタ及び C レジスタを使用して、 A レジスタに格納されている値と、指定されたメインRAM103内のアドレス領域に格納されている値とを交換する機能 (処理動作) を実現するためのプログラム例を説明する。図360A及び図360Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例3-A、及び、本実施形態で新たに設けられたエクスチェンジ命令コードを用いて生成されたプログラム例3-Bを示す。

【2512】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図360Aに示すように、まず、ソースコード「 $LDQ\ C, (wAREAO)$ 」を実行して、 Q レジスタに格納された値 (指定メモリの上位側1バイトのアドレス) と8ビットデータ「 $wAREAO$ 」 (指定メモリの下位側1バイトのアドレス) とで指定されるメインRAM103内のアドレス領域 (指定メモリ) に格納されている値 (交換対象となる一方の値) を、 C レジスタにセット (ロード) する。次いで、ソースコード「 $LDQ\ (wAREAO), A$ 」を実行して、 A レジスタに格納されている値 (交換対象となる他方の値) を、 Q レジスタに格納された値と8ビットデータ「 $wAREAO$ 」とで指定されるアドレス領域 (指定メモリ) にセット (ロード) する。そして、ソースコード「 $LD\ A, C$ 」を実行して、 C レジスタに格納されている値 (交換対象となる一方の値) を、 A レジスタにセット (ロード) する。

【2513】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した3つのソースコードを順次実行することにより、 A レジスタに格納されている値と、指定アドレス領域 (Q レジスタに格納された値と8ビットデータ「 $wAREAO$ 」とで指定されるアドレス領域) に格納されている値とを交換することができる。なお、このソースプログラムでは、ソースコード「 $LDQ\ C, (wAREAO)$ 」、「 $LDQ\ (wAREAO), A$ 」及び「 $LD\ A, C$ 」のバイト数 (容量) は、それぞれ、3バイト、3バイト及び1バイトであり、プログラム例3-Aのプログラムの合計バイト数は、7バイトとなる。

【2514】

一方、本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合、図360Bに示すように、ソースコード「 $EXQ\ A, (wAREAO)$ 」 (新命令コード: 図308中のNo. A19a) を実行して、 A レジスタに格納されている値と、 Q レジスタに格納された値 (指定メモリの上位側1バイトのアドレス) 及び8ビットデータ「 $wAREAO$ 」 (指定メモリの下位側1バイトのアドレス) により指定されるメインRAM103内のアドレス領域 (指定メモリ) に格納されている値とを交換する。

【2515】

10

20

30

40

50

本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、Aレジスタに格納されている値と、指定アドレス領域（Qレジスタに格納された値と8ビットデータ「wARE A 0」とで指定されるアドレス領域）に格納されている値とを交換することができる。なお、このソースプログラムでは、ソースコード「EXQ A, (wARE A 0)」のバイト数（容量）は、3バイトであり、プログラム例3-Bのプログラムの合計バイト数も、3バイトとなる。

【2516】

上述のように、プログラム例3においても、本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2517】

【ロード命令コードを使用したプログラム例】

次に、本実施形態で新たに設けられた8ビットロード命令コード及び16ビットロード命令コードを使用した各種プログラム例を説明する。

【2518】

（1）プログラム例4

プログラム例4では、TZフラグの状態（1（セット）／0（リセット））に応じて、指定されたメインRAM103内のアドレス領域（wARE A 1）に「0」又は「1」をロード（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図361A及び図361Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例4-A、及び、本実施形態で新たに設けられた8ビットロード命令コードを用いて生成されたプログラム例4-Bを示す。

【2519】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図361Aに示すように、まず、ソースコード「JR TZ, LABEL E 1」を実行する。この処理において、TZフラグ=1であれば、ラベル「LABEL E 1」の処理にジャンプし、ソースコード「LD A, 0」を実行して、Aレジスタに「0」をセット（ロード）する。一方、TZフラグ=0であれば、ソースコード「LD A, 1」を実行して、Aレジスタに「1」をセット（ロード）した後、ソースコード「JR LABEL E 2」を実行して、ラベル「LABEL E 2」の処理にジャンプする。そして、ソースコード「LD A, 0」又はソースコード「JR LABEL E 2」の実行後、ラベル「LABEL E 2」の処理、すなわち、ソースコード「LD (wARE A 1), A」を実行して、Aレジスタに格納されている値（「1」又は「0」）を、「wARE A 1」で指定されるメインRAM103内のアドレス領域に格納（ロード）する。

【2520】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した5つのソースコードを順次実行することにより、TZフラグ=1であれば、指定アドレス領域（wARE A 1）に「0」が格納され、TZフラグ=0であれば、指定アドレス領域（wARE A 1）に「1」が格納される。なお、このソースプログラムでは、ソースコード「JR TZ, LABEL E 1」、「LD A, 1」、「JR LABEL E 2」、「LD A, 0」及び「LD (wARE A 1), A」のバイト数（容量）は、それぞれ、2バイト、2バイト、2バイト、2バイト及び3バイトとなり、プログラム例4-Aのプログラムの合計バイト数は、11バイトとなる。

【2521】

一方、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、図361Bに示すように、ソースコード「LDZ (wARE A 1)」（新命令コード：図310中のNo. B16）を実行して、TZフラグ=1であれば、「wARE A 1」（16ビットデータ）で指定されるアドレス領域に「0」を格納し、TZフラグ=0であれば、「wARE A 1」で指定されるアドレス領域に「1」を格納する。

【2522】

10

20

30

40

50

本実施形態で新たに設けられたエクスチェンジ命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、TZフラグの状態（1（セット）／0（リセット））に応じて、指定アドレス領域（指定メモリ）に「0」又は「1」をロード（セット）することができる。なお、このソースプログラムでは、ソースコード「LDZ（wAREA1）」のバイト数（容量）は4バイトであり、プログラム例4-Bのプログラムの合計バイト数も、4バイトとなる。

【2523】

上述のように、プログラム例4においても、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2524】

（2）プログラム例5

プログラム例5では、TZフラグの状態（1（セット）／0（リセット））に応じて、HLレジスタで指定されるメインRAM103内のアドレス領域（wAREA1）に「0」又は「1」をロード（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図362A及び図362Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例5-A、及び、本実施形態で新たに設けられた8ビットロード命令コードを用いて生成されたプログラム例5-Bを示す。

【2525】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図362Aに示すように、まず、ソースコード「LD HL, wAREA1」を実行して、HLレジスタに、メインRAM103内の指定アドレス「wAREA1」（16ビットデータ）をロード（セット）する。次いで、ソースコード「LD A, (wAREA0)」を実行して、「wAREA0」で指定されるアドレス領域に格納された値をAレジスタにロード（セット）する。なお、この処理は、TZフラグの状態を変化させるために行っている。次いで、ソースコード「JR TZ, LABEL1」を実行する。この処理において、TZフラグ=1であれば、ラベル「LABEL1」の処理にジャンプし、ソースコード「LD A, 0」を実行して、Aレジスタに「0」をセット（ロード）する。一方、TZフラグ=0であれば、ソースコード「LD A, 1」を実行して、Aレジスタに「1」をセット（ロード）した後、ソースコード「JR LABEL2」を実行して、ラベル「LABEL2」の処理にジャンプする。そして、ソースコード「LD A, 0」又はソースコード「JR LABEL2」の実行後、ラベル「LABEL2」の処理、すなわち、ソースコード「LD (wAREA1), A」を実行して、Aレジスタに格納されている値（「1」又は「0」）を、「wAREA1」で指定されるアドレス領域に格納（ロード）する。

【2526】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した7つのソースコードを順次実行することにより、TZフラグ=1であれば、指定アドレス領域（wAREA1）に「0」が格納され、TZフラグ=0であれば、指定アドレス領域（wAREA1）に「1」が格納される。なお、このソースプログラムでは、ソースコード「LD HL, wAREA1」、「LD A, (wAREA0)」、「JR TZ, LABEL1」、「LD A, 1」、「JR LABEL2」、「LD A, 0」及び「LD (wAREA1), A」のバイト数（容量）は、それぞれ、3バイト、3バイト、2バイト、2バイト、2バイト、2バイト及び3バイトとなり、プログラム例5-Aのプログラムの合計バイト数は、17バイトとなる。

【2527】

一方、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、図362Bに示すように、まず、ソースコード「LD HL, wAREA1」を実行して、HLレジスタに、指定アドレス「wAREA1」（16ビットデータ）をロード（セット）する。次いで、ソースコード「LD A, (wAREA0)」を実行して、「wAREA0」

10

20

30

40

50

で指定されるアドレス領域に格納された値をAレジスタにロード（セット）する。そして、ソースコード「LDZ (HL)」(新命令コード：図310中のNo. B18)を実行して、TZフラグ=1であれば、HLレジスタに格納された値(wAREA1)で指定されるアドレス領域に「0」を格納し、TZフラグ=0であれば、HLレジスタに格納された値(wAREA1)で指定されるアドレス領域に「1」を格納する。

【2528】

本実施形態で新たに設けられた8ビットロード命令コードを用いた場合には、上述した3つのソースコードを順次実行することにより、TZフラグの状態(1(セット)/0(リセット))に応じて、指定アドレス領域(指定メモリ)に「0」又は「1」をロード（セット）することができる。なお、このソースプログラムでは、ソースコード「LD HL, wAREA1」、「LD A, (wAREA0)」及び「LDZ (HL)」のバイト数（容量）は、それぞれ、3バイト、3バイト及び2バイトであり、プログラム例5-Bのプログラムの合計バイト数は、8バイトとなる。

【2529】

上述のように、プログラム例5においても、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2530】

(3) プログラム例6

プログラム例6では、TZフラグの状態(1(セット)/0(リセット))に応じて、Qレジスタを用いて指定されたメインRAM103内のアドレス領域に「0」又は「1」をロード（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図363A及び図363Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例6-A、及び、本実施形態で新たに設けられた8ビットロード命令コードを用いて生成されたプログラム例6-Bを示す。

【2531】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図363Aに示すように、まず、ソースコード「JR TZ, LABEL1」を実行する。この処理において、TZフラグ=1であれば、ラベル「LABEL1」の処理にジャンプし、ソースコード「LD A, 0」を実行して、Aレジスタに「0」をセット（ロード）する。一方、TZフラグ=0であれば、ソースコード「LD A, 1」を実行して、Aレジスタに「1」をセット（ロード）した後、ソースコード「JR LABEL2」を実行して、ラベル「LABEL2」の処理にジャンプする。そして、ソースコード「LD A, 0」又はソースコード「JR LABEL2」の実行後、ラベル「LABEL2」の処理、すなわち、ソースコード「LDQ (wAREA1), A」を実行して、Aレジスタに格納されている値（「1」又は「0」）を、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wAREA1」（8ビットデータ：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納（ロード）する。

【2532】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した5つのソースコードを順次実行することにより、TZフラグ=1であれば、Qレジスタに格納された値と「wAREA1」とで指定されたアドレス領域に「0」が格納され、TZフラグ=0であれば、Qレジスタに格納された値と「wAREA1」とで指定されたアドレス領域に「1」が格納される。なお、このソースプログラムでは、ソースコード「JR TZ, LABEL1」、「LD A, 1」、「JR LABEL2」、「LD A, 0」及び「LDQ (wAREA1), A」のバイト数（容量）は、それぞれ、2バイト、2バイト、2バイト、2バイト及び2バイトであり、プログラム例6-Aのプログラムの合計バイト数は、10バイトとなる。

【2533】

10

20

30

40

50

一方、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、図363Bに示すように、ソースコード「LDQZ (wAREA1)」(新命令コード:図310中のNo. B21)を実行して、TZフラグ=1であれば、Qレジスタに格納された値と「wAREA1」(8ビットデータ)とで指定されるアドレス領域に「0」を格納し、TZフラグ=0であれば、Qレジスタに格納された値と「wAREA1」とで指定されるアドレス領域に「1」を格納する。

【2534】

本実施形態で新たに設けられた8ビットロード命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、TZフラグの状態(1(セット)/0(リセット))に応じて、指定アドレス領域(指定メモリ領域)に「0」又は「1」をロード(セット)することができる。なお、このソースプログラムでは、ソースコード「LDQZ (wAREA1)」のバイト数(容量)は、3バイトであり、プログラム例6-Bのプログラムの合計バイト数も、3バイトとなる。

【2535】

上述のように、プログラム例6においても、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、それを用いない場合に比べて、同じ機能(処理動作)を実現するためのプログラムの容量(バイト数)を減らすことができる。

【2536】

(4) プログラム例7

プログラム例7では、Cフラグ(CYフラグ)の状態(1(セット)/0(リセット)):この例では比較処理の結果)に応じて、指定されたメインRAM103内のアドレス領域(wAREA2)に「1」又は「0」をロード(セット)する機能(処理動作)を実現するためのプログラム例を説明する。図364A及び図364Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例7-A、及び、本実施形態で新たに設けられた8ビットロード命令コードを用いて生成されたプログラム例7-Bを示す。

【2537】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図364Aに示すように、まず、ソースコード「LDA, (wAREA2)」を実行して、「wAREA2」(16ビットデータ)で指定されるメインRAM103内のアドレス領域に格納されている値をAレジスタにロード(セット)する。次いで、ソースコード「CP 10H」を実行して、Aレジスタに格納されている値と、「10H」とを比較し、比較結果に基づいてCフラグ(CYフラグ)の状態を設定する。次いで、ソースコード「JRC, LABELE3」を実行する。この処理において、Cフラグ=1であれば、ラベル「LABLEE3」の処理にジャンプし、ソースコード「LDA, 1」を実行して、Aレジスタに「1」をセット(ロード)する。一方、Cフラグ=0であれば、ソースコード「LDA, 0」を実行して、Aレジスタに「0」をセット(ロード)した後、ソースコード「JRLABLEE4」を実行して、ラベル「LABLEE4」の処理にジャンプする。そして、ソースコード「LDA, 1」又はソースコード「JRLABLEE4」の実行後、ラベル「LABLEE4」の処理、すなわち、ソースコード「LDA (wAREA2), A」を実行して、Aレジスタに格納されている値(「1」又は「0」)を、「wAREA2」で指定されるアドレス領域(指定メモリ)に格納(ロード)する。

【2538】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した7つのソースコードを順次実行することにより、Cフラグ=1であれば、「wAREA2」で指定されたアドレス領域に「1」が格納され、Cフラグ=0であれば、「wAREA2」で指定されたアドレス領域に「0」が格納される。なお、このソースプログラムでは、ソースコード「LDA, (wAREA2)」、「CP 10H」、「JRC, LABELE3」、「LDA, 0」、「JRLABLEE4」、「LDA, 1」及び「LDA (wAREA2), A」のバイト数(容量)は、それぞれ、2バイト、2バイト、

10

20

30

40

50

2 バイト、2 バイト、2 バイト、2 バイト及び 1 バイトであり、プログラム例 7 - A のプログラムの合計バイト数は、13 バイトとなる。

【2539】

一方、本実施形態で新たに設けられた 8 ビットロード命令コードを用いた場合、図 364 B に示すように、まず、ソースコード「LD A, (wAREA2)」を実行して、「wAREA2」で指定されるアドレス領域に格納されている値を A レジスタにロード（セット）する。次いで、ソースコード「CP 10H」を実行して、A レジスタに格納されている値と、「10H」とを比較し、比較結果に基づいて C フラグ（CY フラグ）の状態を設定する。そして、ソースコード「LDC (wAREA2)」（新命令コード：図 310 中の No. B22）を実行して、C フラグ = 1 であれば、「wAREA2」（16 ビットデータ）で指定されるアドレス領域に「1」を格納し、C フラグ = 0 であれば、「wAREA2」で指定されるアドレス領域に「0」を格納する。

10

【2540】

本実施形態で新たに設けられた 8 ビットロード命令コードを用いた場合には、上述した 3 つのソースコードを順次実行することにより、C フラグの状態（1（セット）／0（リセット））に応じて、指定アドレス領域（指定メモリ領域）に「1」又は「0」をロード（セット）することができる。なお、このソースプログラムでは、ソースコード「LD A, (wAREA2)」、「CP 10H」及び「LDC (wAREA2)」のバイト数（容量）は、それぞれ、2 バイト、2 バイト及び 2 バイトであり、プログラム例 7 - B のプログラムの合計バイト数は、6 バイトとなる。

20

【2541】

上述のように、プログラム例 7 においても、本実施形態で新たに設けられた 8 ビットロード命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2542】

（5）プログラム例 8

プログラム例 8 では、C フラグ（CY フラグ）の状態（1（セット）／0（リセット）：この例では比較処理の結果）に応じて、HL レジスタで指定されたメイン RAM 103 内のアドレス領域（wAREA2）に「1」又は「0」をロード（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図 365 A 及び図 365 B に、それぞれ、上記第 1、第 2 実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例 8 - A、及び、本実施形態で新たに設けられた 8 ビットロード命令コードを用いて生成されたプログラム例 8 - B を示す。

30

【2543】

上記第 1 及び第 2 実施形態の遊技機で使用されている命令コードを用いた場合、図 365 A に示すように、まず、ソースコード「LD HL, wAREA2」を実行して、「wAREA2」（16 ビットデータ）で指定されるメイン RAM 103 内のアドレスを HL レジスタにロード（セット）する。次いで、ソースコード「CP (HL)」を実行して、A レジスタに格納されている値と、HL レジスタに格納された値で指定されるアドレス領域（wAREA2）に格納されている値とを比較し、比較結果に基づいて C フラグ（CY フラグ）の状態を設定する。次いで、ソースコード「JR C, LABEL3」を実行する。この処理において、C フラグ = 1 であれば、ラベル「LABEL3」の処理にジャンプし、ソースコード「LD A, 1」を実行して、A レジスタに「1」をセット（ロード）する。一方、C フラグ = 0 であれば、ソースコード「LD A, 0」を実行して、A レジスタに「0」をセット（ロード）した後、ソースコード「JR LABEL4」を実行して、ラベル「LABEL4」の処理にジャンプする。そして、ソースコード「LD A, 1」又はソースコード「JR LABEL4」の実行後、ラベル「LABEL4」の処理、すなわち、ソースコード「LD (HL), A」を実行して、A レジスタに格納されている値（「1」又は「0」）を、HL レジスタに格納された値で指定されるアドレス領域（wAREA2）に格納（ロード）する。

40

50

【2544】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した7つのソースコードを順次実行することにより、Cフラグ=1であれば、HLレジスタで指定されるアドレス領域(wAREA2)に「1」が格納され、Cフラグ=0であれば、HLレジスタで指定されるアドレス領域(wAREA2)に「0」が格納される。なお、このソースプログラムでは、ソースコード「LD HL, wAREA2」、「CP (HL)」、「JR C, LABEL3」、「LD A, 0」、「JR LABEL4」、「LD A, 1」及び「LD (HL), A」のバイト数(容量)は、それぞれ、3バイト、1バイト、2バイト、2バイト、2バイト、2バイト及び1バイトであり、プログラム例8-Aのプログラムの合計バイト数は、13バイトとなる。

10

【2545】

一方、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、図365Bに示すように、まず、ソースコード「LD HL, wAREA2」を実行して、「wAREA2」(16ビットデータ)で指定されるメインRAM103内のアドレスをHLレジスタにロード(セット)する。次いで、ソースコード「CP (HL)」を実行して、Aレジスタに格納されている値と、HLレジスタに格納された値で指定されるアドレス領域(wAREA2)に格納されている値とを比較し、比較結果に基づいてCフラグ(CYフラグ)の状態を設定する。そして、ソースコード「LDC (HL)」(新命令コード：図310中のNo. B24)を実行して、Cフラグ=1であれば、HLレジスタに格納された値(wAREA2)で指定されるアドレス領域に「1」を格納し、Cフラグ=0であれば、HLレジスタに格納された値で指定されるアドレス領域に「0」を格納する。

20

【2546】

本実施形態で新たに設けられた8ビットロード命令コードを用いた場合には、上述した3つのソースコードを順次実行することにより、Cフラグの状態(1(セット)/0(リセット))に応じて、HLレジスタで指定されたメモリ領域に「1」又は「0」をロード(セット)することができる。なお、このソースプログラムでは、ソースコード「LD HL, wAREA2」、「CP (HL)」及び「LDC (HL)」のバイト数(容量)は、それぞれ、3バイト、1バイト及び2バイトであり、プログラム例8-Bのプログラムの合計バイト数は、6バイトとなる。

【2547】

上述のように、プログラム例8においても、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、それを用いない場合に比べて、同じ機能(処理動作)を実現するためのプログラムの容量(バイト数)を減らすことができる。

30

【2548】

(6) プログラム例9

プログラム例9では、Cフラグ(CYフラグ)の状態(1(セット)/0(リセット))：この例では比較処理の結果)に応じて、Qレジスタを用いて指定されたメインRAM103内のアドレス領域に「1」又は「0」をロード(セット)する機能(処理動作)を実現するためのプログラム例を説明する。図366A及び図366Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例9-A、及び、本実施形態で新たに設けられた8ビットロード命令コードを用いて生成されたプログラム例9-Bを示す。

40

【2549】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図366Aに示すように、まず、ソースコード「CPQ (wAREA2)」を実行して、Aレジスタに格納されている値と、Qレジスタに格納された値(指定メモリの上位側1バイトのアドレス)及び「wAREA2」(8ビットデータ：指定メモリの下位側1バイトのアドレス)により指定されるメインRAM103内のアドレス領域(指定メモリ)に格納されている値とを比較し、比較結果に基づいてCフラグ(CYフラグ)の状態を設定する。次いで、ソースコード「JR C, LABEL3」を実行する。この処理において、Cフ

50

ラグ＝1であれば、ラベル「L A B E L E 3」の処理にジャンプし、ソースコード「L D A , 1」を実行して、Aレジスタに「1」をセット（ロード）する。一方、Cフラグ＝0であれば、ソースコード「L D A , 0」を実行して、Aレジスタに「0」をセット（ロード）した後、ソースコード「J R L A B E L E 4」を実行して、ラベル「L A B E L E 4」の処理にジャンプする。そして、ソースコード「L D A , 1」又はソースコード「J R L A B E L E 4」の実行後、ラベル「L A B E L E 4」の処理、すなわち、ソースコード「L D Q (w A R E A 2) , A」を実行して、Aレジスタに格納されている値（「1」又は「0」）を、Qレジスタに格納された値と「w A R E A 2」（8ビットデータ）とで指定されるアドレス領域に格納（ロード）する。

【2550】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した6つのソースコードを順次実行することにより、Cフラグ＝1であれば、Qレジスタに格納された値と「w A R E A 2」（8ビットデータ）とで指定されるアドレス領域に「1」が格納され、Cフラグ＝0であれば、Qレジスタに格納された値と「w A R E A 2」とで指定されるアドレス領域に「0」が格納される。なお、このソースプログラムでは、ソースコード「C P Q (w A R E A 2)」、「J R C , L A B E L E 3」、「L D A , 0」、「J R L A B E L E 4」、「L D A , 1」及び「L D Q (w A R E A 2) , A」のバイト数（容量）は、それぞれ、3バイト、2バイト、2バイト、2バイト、2バイト及び2バイトであり、プログラム例9－Aのプログラムの合計バイト数は、13バイトとなる。

【2551】

一方、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、図366Bに示すように、まず、ソースコード「C P Q (w A R E A 2)」を実行して、Aレジスタに格納されている値と、Qレジスタに格納された値及び「w A R E A 2」（8ビットデータ）により指定されるアドレス領域（指定メモリ）に格納されている値とを比較し、比較結果に基づいてCフラグ（C Y フラグ）の状態を設定する。そして、ソースコード「L D Q C (w A R E A 2)」（新命令コード：図311中のNo. B27）を実行して、Cフラグ＝1であれば、Qレジスタに格納された値と「w A R E A 2」（8ビットデータ）とで指定されるアドレス領域に「1」を格納し、Cフラグ＝0であれば、Qレジスタに格納された値と「w A R E A 2」（8ビットデータ）とで指定されるアドレス領域に「0」を格納する。

【2552】

本実施形態で新たに設けられた8ビットロード命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、Cフラグの状態（1（セット）／0（リセット））に応じて、Qレジスタに格納された値と8ビットデータ（w A R E A 2）とで指定されるメモリ領域に「1」又は「0」をロード（セット）することができる。なお、このソースプログラムでは、ソースコード「C P Q (w A R E A 2)」及び「L D Q C (w A R E A 2)」のバイト数（容量）は、それぞれ、3バイト及び3バイトであり、プログラム例9－Bのプログラムの合計バイト数は、6バイトとなる。

【2553】

上述のように、プログラム例9においても、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2554】

（7）プログラム例10

プログラム例10では、フラグ・レジスタF内の指定フラグ（この例では、Cフラグ）の状態（1（セット）／0（リセット））に応じて、設定された特定値（8ビットデータ）をAレジスタに格納（ロード）する機能（処理動作）を実現するためのプログラム例を説明する。図367A及び図367Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例10－A、及び、本実施形態で

10

20

30

40

50

新たに設けられた8ビットロード命令コードを用いて生成されたプログラム例10-Bを示す。

【2555】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図367Aに示すように、まず、ソースコード「JR NC, LABEL1」を実行する。この処理において、Cフラグ(CYフラグ)=0であれば、ラベル「LABEL1」にジャンプし、処理を終了する。一方、この処理において、Cフラグ(CYフラグ)=1であれば、ソースコード「LD A, 10」を実行して、Aレジスタに「10」(特定値)をセット(ロード)する。

【2556】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、Cフラグの状態に応じて、Aレジスタに「10」を格納することができる。なお、このソースプログラムでは、ソースコード「JR NC, LABEL1」及び「LD A, 10」のバイト数(容量)は、それぞれ、2バイト及び2バイトであり、プログラム例10-Aのプログラムの合計バイト数は、4バイトとなる。

【2557】

一方、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、図367Bに示すように、ソースコード「LDF C, A, 10」(新命令コード:図311中のNo. B31)を実行して、Cフラグ=1であれば、Aレジスタに「10」(特定値)をセット(ロード)する。

【2558】

本実施形態で新たに設けられた8ビットロード命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、Cフラグの状態に応じて、Aレジスタに「10」を格納することができる。なお、このソースプログラムでは、ソースコード「LDF C, A, 10」のバイト数(容量)は、3バイトであり、プログラム例10-Bのプログラムの合計バイト数も、3バイトとなる。

【2559】

上述のように、プログラム例10においても、本実施形態で新たに設けられた8ビットロード命令コードを用いた場合、それを用いない場合に比べて、同じ機能(処理動作)を実現するためのプログラムの容量(バイト数)を減らすことができる。

【2560】

(8) プログラム例11

プログラム例11では、フラグ・レジスタF内の指定フラグ(この例では、Zフラグ)の状態(1(セット)/0(リセット))に応じて、設定された特定値(16ビットデータ)をBCレジスタに格納(ロード)する機能(処理動作)を実現するためのプログラム例を説明する。図368A及び図368Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例11-A、及び、本実施形態で新たに設けられた16ビットロード命令コードを用いて生成されたプログラム例11-Bを示す。

【2561】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図368Aに示すように、まず、ソースコード「JR NZ, LABEL1」を実行する。この処理において、Zフラグ=0であれば、ラベル「LABEL1」にジャンプし、処理を終了する。一方、この処理において、Zフラグ=1であれば、ソースコード「LD BC, 1000」を実行して、BCレジスタに「1000」(特定値)をセット(ロード)する。

【2562】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、Zフラグの状態に応じて、BCレジ

10

20

30

40

50

スタに「1000」を格納することができる。なお、このソースプログラムでは、ソースコード「JR NZ, LABEL1」及び「LD BC, 1000」のバイト数（容量）は、それぞれ、2バイト及び3バイトであり、プログラム例11-Aのプログラムの合計バイト数は、5バイトとなる。

【2563】

一方、本実施形態で新たに設けられた16ビットロード命令コードを用いた場合、図368Bに示すように、ソースコード「LDF Z, BC, 1000」（新命令コード：図325中のNo. C14）を実行して、Zフラグ=1であれば、BCレジスタに「1000」（特定値）をセット（ロード）する。

【2564】

本実施形態で新たに設けられた16ビットロード命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、Zフラグの状態に応じて、BCレジスタに「1000」を格納することができる。なお、このソースプログラムでは、ソースコード「LDF Z, BC, 1000」のバイト数（容量）は、4バイトであり、プログラム例11-Bのプログラムの合計バイト数も、4バイトとなる。

【2565】

上述のように、プログラム例11においても、本実施形態で新たに設けられた16ビットロード命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2566】

〔ジャンプ命令コードを使用したプログラム例〕

次に、本実施形態で新たに設けられたジャンプ命令コードを使用した各種プログラム例を説明する。

【2567】

（1）プログラム例12

プログラム例12では、ジャンプ先のサブルーチンプログラム（以下、単に「サブルーチン」と称す）のアドレスに関するデータ（サブルーチンまでの相対バイト値、サブルーチンの下位側1バイトのアドレス）が格納されたテーブル（ジャンプ先テーブル）を使用して、処理を所望のサブルーチンの処理にジャンプさせる機能（処理動作）を実現するためのプログラム例を説明する。図369A及び図369Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例12-A、及び、本実施形態で新たに設けられたジャンプ命令コードを用いて生成されたプログラム例12-Bを示す。

【2568】

なお、上記第1及び第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例12-Aでは、図369Aに示すように、ジャンプ先テーブル（ラベル「TBL__SUB」のアドレスから始まる4バイト（1バイトデータ（DB）×4個）のテーブル）には、第1～第4サブルーチンまでの相対バイト値（「P__SUB__10-\$」～「P__SUB__40-\$」）が格納されているものとする。また、プログラム例12-Aでは、プログラム例12-Aの一連の処理コードの直下のアドレス（最後のソースコード「JP (HL)」のアドレスの次のアドレス）に、ジャンプ先テーブルの先頭アドレス（ラベル「TBL__SUB」のアドレス）が配置されているものとする。なお、プログラム例12-Aのソースコード及びジャンプ先テーブルで構成されているプログラム上のアドレス空間では、1バイトの領域毎にアドレスが割り当てられている。

【2569】

また、本実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例12-Bでは、図369Bに示すように、ジャンプ先テーブル（ラベル「TBL__SUB」から始まる4バイト（1バイトデータ（DB）×4個）のテーブル）には、第1～第4サブルーチンのアドレスの下位側1バイトのアドレス（「.LOW.P__SUB__10」～「.LOW.P__SUB__40」）が格納されているものとする。また、プログラム例

10

20

30

40

50

12-Bでは、プログラム例12-Bの一連の処理コードの直下のアドレス（最後のソースコード「TJP」のアドレスの次のアドレス）に、ジャンプ先テーブルの先頭アドレス（ラベル「TBL__SUB」のアドレス）が配置されているものとする。なお、プログラム例12-Bのソースコード及びジャンプ先テーブルで構成されているプログラム上のアドレス空間では、1バイトの領域毎にアドレスが割り当てられている。なお、擬似命令「.LOW.」については、上記図66及び図67に示す遊技復帰処理で説明したので、ここではその説明を省略する。

【2570】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図369Aに示すように、まず、ソースコード「LDQ A, (wAREA4)」を実行して、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wAREA4」（8ビットデータ：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Aレジスタにロード（セット）する。なお、この指定メモリには、ジャンプ先テーブルの先頭アドレスからジャンプ先テーブル内の各データ（サブルーチンまでの相対バイト値）の格納アドレスまでの相対値が格納される。具体的には、ジャンプ先となるサブルーチンが第1サブルーチンである場合には、指定メモリ（Qレジスタに格納された値と「wAREA4」（8ビットデータ）とで指定されるアドレス）に「0」が格納され、ジャンプ先となるサブルーチンが第2サブルーチンである場合には、指定メモリに「1」が格納され、ジャンプ先となるサブルーチンが第3サブルーチンである場合には、指定メモリに「2」が格納され、ジャンプ先となるサブルーチンが第4サブルーチンである場合には、指定メモリに「3」が格納されている。

【2571】

次いで、ソースコード「LD HL, TBL__SUB」を実行して、HLレジスタにラベル「TBL__SUB」で示されるジャンプ先テーブルの先頭アドレスをロード（セット）する。次いで、ソースコード「ADDWB HL, A」を実行して、Aレジスタに格納されている値をHLレジスタに格納されている値に加算して、該加算された値をHLレジスタに格納する。この処理により、HLレジスタには、ジャンプ先となるサブルーチンまでの相対バイト値が格納されたジャンプ先テーブル内のアドレスが格納される。次いで、ソースコード「LD A, (HL)」を実行して、HLレジスタにセットされている値で指定されるアドレス領域に格納されている値（ジャンプ先となるサブルーチンまでの相対バイト値）を、Aレジスタにロード（セット）する。

【2572】

次いで、ソースコード「OR A」を実行して、Aレジスタにセットされている値とAレジスタにセットされている値とを論理和し、演算結果をAレジスタに格納する。この処理はCフラグの状態をリセット（クリア）するために行われる。次いで、ソースコード「ADDWB HL, A」を実行して、Aレジスタに格納されている値をHLレジスタに格納されている値に加算して、該加算された値をHLレジスタに格納する。この処理により、ジャンプ先となるサブルーチンの絶対アドレス値が算出され、その絶対アドレス値がHLレジスタに格納される。そして、ソースコード「JPC (HL)」を実行して、HLレジスタにセットされている値をPCレジスタ（プログラム・カウンタPC）にロード（セット）する。この処理により、処理が所望のサブルーチンにジャンプし、サブルーチンの処理が開始される。

【2573】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した7つのソースコードを順次実行することにより、ジャンプ先のサブルーチンのアドレスに関するデータ（サブルーチンまでの相対バイト値）が格納されたジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせることができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wAREA4)」、「LD HL, TBL__SUB」、「ADDWB HL, A」、「LD A, (HL)」、「OR A」

10

20

30

40

50

、「ADDWB HL, A」及び「J P (HL)」のバイト数(容量)は、それぞれ、2 バイト、3 バイト、1 バイト、1 バイト、1 バイト、1 バイト及び1 バイトであり、ジャンプ先テーブルのバイト数(容量)は、4 バイトであるので、プログラム例 12-A のプログラムの合計バイト数は、14 バイトとなる。

【2574】

一方、本実施形態で新たに設けられたジャンプ命令コードを用いた場合、図 369B に示すように、まず、ソースコード「LDQ A, (wARE A4)」を実行して、Qレジスタに格納された値(指定メモリの上位側1バイトのアドレス)と「wARE A4」(8ビットデータ: 指定メモリの下位側1バイトのアドレス)とで指定されるメインRAM 103 内のアドレス領域(指定メモリ)に格納されている値を、Aレジスタにロード(セット)する。なお、この指定メモリには、ジャンプ先テーブルの先頭アドレスからジャンプ先テーブル内の各データ(サブルーチンの下位側1バイトのアドレス)の格納アドレスまでの相対値が格納される。具体的には、ジャンプ先となるサブルーチンが第1サブルーチンである場合には、指定メモリ(Qレジスタに格納された値と「wARE A4」(8ビットデータ)とで指定されるアドレス)に「0」が格納され、ジャンプ先となるサブルーチンが第2サブルーチンである場合には、指定メモリに「1」が格納され、ジャンプ先となるサブルーチンが第3サブルーチンである場合には、指定メモリに「2」が格納され、ジャンプ先となるサブルーチンが第4サブルーチンである場合には、指定メモリに「3」が格納されている。

【2575】

そして、ソースコード「LDQ A, (wARE A4)」の実行後、ソースコード「T J P」(新命令コード: 図 326 中の No. D6)を実行して、現在、PCレジスタにセットされている値(本ソースコードのアドレス)に、「1」とAレジスタに格納されている値とを加算($PC + 1 + A$)し、該加算した値で指定されるアドレスに格納されている値(ジャンプ先となるサブルーチンの下位側1バイトのアドレス)をPCレジスタの下位側バイトにロード(セット)する。この処理により、処理が所望のサブルーチンにジャンプし、サブルーチンの処理が開始される。

【2576】

なお、ソースコード「T J P」(1バイト命令)内における処理では、まず、現在のPCレジスタにセットされている値(本ソースコードのアドレス)に「1」を加算して、ジャンプ先テーブルの先頭アドレスが算出される。次いで、その算出値にAレジスタに格納されている値(Qレジスタに格納された値と「wARE A4」とで指定されるアドレスに格納されている値: 「0」~「3」のいずれか)を加算して、所望のジャンプ先となるサブルーチンの下位側1バイトのアドレスが格納されたジャンプ先テーブル内のアドレスが算出される。それゆえ、ジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせるプログラムにおいて、ジャンプ命令コード「T J P」(1バイト命令)を用いる場合には、ソースコード「T J P」の直下のアドレス(ソースコード「T J P」のアドレスの次のアドレス)に、ジャンプ先テーブルの先頭アドレスを配置する必要がある。

【2577】

本実施形態で新たに設けられたジャンプ命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、ジャンプ先のサブルーチンのアドレスに関するデータ(サブルーチンの下位側1バイトのアドレス)が格納されたジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせることができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wARE A4)」及び「T J P」のバイト数(容量)は、それぞれ、2 バイト及び1 バイトであり、ジャンプ先テーブルのバイト数(容量)は、4 バイトであるので、プログラム例 12-B のプログラムの合計バイト数は、7 バイトとなる。なお、図 369B では、4 つのサブルーチンの処理にジャンプさせる処理において「T J P」命令を使用するプログラム例(「T J P」命令の直下に4 個のジャンプ先サブルーチンの下位側アドレスを記憶する例)を説明したが、本発明はこ

れに限定されない。プログラム 12-B が、例えば、メイン ROM 102 のアドレス「1400H」から記憶されている場合、ジャンプ先サブルーチンが、例えばアドレス「1200H」～「12FFH」の範囲に格納されていれば、Aレジスタに格納できる値の数（0～255）の分までジャンプ先サブルーチンの下位側アドレスを記憶させることができる。

【2578】

上述のように、プログラム例 12 においても、本実施形態で新たに設けられたジャンプ命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2579】

（2）プログラム例 13

プログラム例 13 では、ジャンプ先のサブルーチンのアドレスデータが格納されたテーブル（ジャンプ先テーブル）を使用して、処理を所望のサブルーチンの処理にジャンプさせる機能（処理動作）を実現するためのプログラム例を説明する。図 370A 及び図 370B に、それぞれ、上記第 1、第 2 実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例 13-A、及び、本実施形態で新たに設けられたジャンプ命令コードを用いて生成されたプログラム例 13-B を示す。

【2580】

なお、上記第 1 及び第 2 実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例 13-A では、図 370A に示すように、ジャンプ先テーブル（ラベル「TBL__SUB」のアドレスから始まる 8 バイト（2 バイトデータ（DW）× 4 個）のテーブル）には、第 1～第 4 サブルーチンのアドレス（絶対アドレス：「P__SUB__10」～「P__SUB__40」）が格納されているものとする。また、プログラム例 13-A では、プログラム例 13-A の一連の処理コードの直下のアドレス（最後のソースコード「JP（HL）」のアドレスの次のアドレス）に、ジャンプ先テーブルの先頭アドレス（ラベル「TBL__SUB」のアドレス）が配置されているものとする。なお、プログラム例 13-A のソースコード及びジャンプ先テーブルで構成されているプログラム上のアドレス空間では、1 バイトの領域毎にアドレスが割り当てられている。

【2581】

また、本実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例 13-B では、図 370B に示すように、ジャンプ先テーブル（ラベル「TBL__SUB」から始まる 8 バイト（2 バイトデータ（DW）× 4 個）のテーブル）には、第 1～第 4 サブルーチンのアドレス（絶対アドレス：「P__SUB__10」～「P__SUB__40」）が格納されているものとする。また、プログラム例 13-B では、プログラム例 13-B の一連の処理コードの直下のアドレス（最後のソースコード「TJPW」のアドレスの次のアドレス）に、ジャンプ先テーブルの先頭アドレス（ラベル「TBL__SUB」のアドレス）が配置されているものとする。なお、プログラム例 13-B のソースコード及びジャンプ先テーブルで構成されているプログラム上のアドレス空間では、1 バイトの領域毎にアドレスが割り当てられている。

【2582】

上記第 1 及び第 2 実施形態の遊技機で使用されている命令コードを用いた場合、図 370A に示すように、まず、ソースコード「LDQ A, (wAREA4)」を実行して、Q レジスタに格納された値（指定メモリの上位側アドレス）と「wAREA4」（8 ビットデータ：指定メモリの下位側アドレス）とで指定されるメイン RAM 103 内のアドレス領域（指定メモリ）に格納されている値を、A レジスタにロード（セット）する。なお、この指定メモリ（Q レジスタに格納された値と「wAREA4」（8 ビットデータ）とで指定されるアドレス）に格納されている値は、上記プログラム例 12 と同様に、ジャンプ先テーブルの先頭アドレスからジャンプ先テーブル内の各データ（第 1～第 4 サブルーチンのアドレス）の格納アドレスまでの相対値（「0」～「3」のいずれか）である。

【2583】

10

20

30

40

50

次いで、ソースコード「LD HL, TBL__SUB」を実行して、HLレジスタにラベル「TBL__SUB」で示されるジャンプ先テーブルの先頭アドレスをロード（セット）する。次いで、ソースコード「ADDWB HL, A」を実行して、Aレジスタに格納されている値をHLレジスタに格納されている値に加算して、該加算された値をHLレジスタに格納する。この処理により、HLレジスタには、ジャンプ先となる所望のサブルーチンのアドレスがセットされる。そして、ソースコード「JP (HL)」を実行して、HLレジスタにセットされている値をPCレジスタ（プログラム・カウンタPC）にロード（セット）する。この処理により、処理が所望のサブルーチンにジャンプし、サブルーチンの処理が開始される。

【2584】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した4つのソースコードを順次実行することにより、ジャンプ先のサブルーチンのアドレスが格納されたジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせることができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wAREA4)」、「LD HL, TBL__SUB」、「ADDWB HL, A」及び「JP (HL)」のバイト数（容量）は、それぞれ、2バイト、3バイト、1バイト及び1バイトであり、ジャンプ先テーブルのバイト数（容量）は、8バイトであるので、プログラム例13-Aのプログラムの合計バイト数は、15バイトとなる。

【2585】

一方、本実施形態で新たに設けられたジャンプ命令コードを用いた場合、図370Bに示すように、まず、ソースコード「LDQ A, (wAREA4)」を実行して、Qレジスタに格納された値（指定メモリの上位側アドレス）と「wAREA4」（8ビットデータ：指定メモリの下位側アドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Aレジスタにロード（セット）する。なお、この指定メモリに格納されている値は、上記プログラム例12と同様に、ジャンプ先テーブルの先頭アドレスからジャンプ先テーブル内の各データ（第1～第4サブルーチンのアドレス）の格納アドレスまでの相対値（「0」～「3」）である。

【2586】

そして、ソースコード「TJPW」（新命令コード：図327中のNo. D13）を実行して、現在、PCレジスタにセットされている値（本ソースコードのアドレス）に、「1」とAレジスタに格納されている値を2倍した値とを加算（ $PC + 1 + A \times 2$ ）し、該加算した値で指定されるアドレスに格納されている値（ジャンプ先となるサブルーチンのアドレス（2バイト））をPCレジスタにロード（セット）する。この処理により、処理が所望のサブルーチンにジャンプし、サブルーチンの処理が開始される。

【2587】

なお、ソースコード「TJPW」の処理では、まず、現在のPCレジスタにセットされている値（本ソースコードのアドレス）に「1」を加算して、ジャンプ先テーブルの先頭アドレスが算出される。次いで、その算出値にAレジスタに格納されている値（「0」～「3」のいずれか）を2倍した値（「0」、「2」、「4」又は「6」）を加算して、ジャンプ先のサブルーチンのアドレス（2バイト）が格納されたジャンプ先テーブル内のアドレス（サブルーチンのアドレス中の下位側1バイトの格納アドレス）が算出される。それゆえ、ジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせるプログラムにおいて、ジャンプ命令コード「TJPW」を用いる場合には、ソースコード「TJPW」の直下のアドレス（ソースコード「TJPW」のアドレスの次のアドレス）に、ジャンプ先テーブルの先頭アドレスを配置する必要がある。

【2588】

本実施形態で新たに設けられたジャンプ命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、ジャンプ先のサブルーチンのアドレスが格納されたジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせることができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wARE

10

20

30

40

50

A 4)」及び「TJPW」のバイト数（容量）は、それぞれ、2 バイト及び 1 バイトであり、ジャンプ先テーブルのバイト数（容量）は、8 バイトであるので、プログラム例 1 3－B のプログラムの合計バイト数は、1 1 バイトとなる。

【2 5 8 9】

上述のように、プログラム例 1 3 においても、本実施形態で新たに設けられたジャンプ命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2 5 9 0】

また、プログラム例 1 3－B で説明した「TJPW」命令とプログラム例 1 2－B で説明した「TJP」命令との最大の違いは、「TJPW」命令に続くジャンプ先テーブルのバイト数が、「TJP」命令に続くジャンプ先テーブルのその 2 倍になることである。しかしながら、プログラム例 1 3－B のジャンプ先テーブルにはサブルーチンの絶対アドレスが格納されているため、プログラム例 1 2－B のようにジャンプ先アドレスの制約（下位側アドレスのみを規定）がなく、「TJPW」命令は、ソフトウェア開発者にとって使い易い（使い勝手の良い）命令となる。すなわち、「TJP」命令と「TJPW」命令とを比較した場合、「TJP」命令は、メインROM 1 0 1 の容量低減を優先した命令であり、「TJPW」命令は、メインROM 1 0 1 の容量低減より、ソフトウェア開発者の使い易さを優先した命令となる。

【2 5 9 1】

（3）プログラム例 1 4

プログラム例 1 4 では、ジャンプ先のサブルーチンのアドレスに関するデータ（サブルーチンまでの相対バイト値、サブルーチンの下位側 1 バイトのアドレス）が格納されたテーブル（ジャンプ先テーブル）を使用して、処理を所望のサブルーチンの処理にジャンプさせる機能（処理動作）を実現するためのプログラム例を説明する。図 3 7 1 A 及び図 3 7 1 B に、それぞれ、上記第 1、第 2 実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例 1 4－A、及び、本実施形態で新たに設けられたジャンプ命令コードを用いて生成されたプログラム例 1 4－B を示す。

【2 5 9 2】

なお、この例では、上記プログラム例 1 2 とは異なり、Q レジスタに格納された値（指定メモリの上位側 1 バイトのアドレス）と「wARE A 4」（8 ビットデータ：指定メモリの下位側 1 バイトのアドレス）とで指定されるメインRAM 1 0 3 内のアドレス領域（指定メモリ）に格納されている値を、B レジスタにロード（セット）する例を説明する。また、この例で用いるジャンプ先テーブルの構成は、上記プログラム例 1 2 のそれ（図 3 6 9 A 及び図 3 6 9 B 参照）と同様であるので、その説明は省略する。

【2 5 9 3】

上記第 1 及び第 2 実施形態の遊技機で使用されている命令コードを用いた場合、図 3 7 1 A に示すように、まず、ソースコード「LDQ B, (wARE A 4)」を実行して、Q レジスタに格納された値（指定メモリの上位側 1 バイトのアドレス）と「wARE A 4」（8 ビットデータ：指定メモリの下位側 1 バイトのアドレス）とで指定されるメインRAM 1 0 3 内のアドレス領域（指定メモリ）に格納されている値を、B レジスタにロード（セット）する。なお、この指定メモリには、ジャンプ先テーブルの先頭アドレスからジャンプ先テーブル内の各データ（サブルーチンまでの相対バイト値）の格納アドレスまでの相対値（「0」～「3」のいずれか）が格納される。

【2 5 9 4】

次いで、ソースコード「LD HL, TBL__SUB」を実行して、HL レジスタにラベル「TBL__SUB」で示されるジャンプ先テーブルの先頭アドレスをロード（セット）する。次いで、ソースコード「ADDWB HL, B」を実行して、B レジスタに格納されている値を HL レジスタに格納されている値に加算して、該加算された値を HL レジスタに格納する。この処理により、HL レジスタには、ジャンプ先となるサブルーチンまでの相対バイト値が格納されたジャンプ先テーブル内のアドレスが格納される。次いで、ソ

10

20

30

40

50

ースコード「LD A, (HL)」を実行して、HLレジスタにセットされている値で指定されるアドレス領域に格納された値（ジャンプ先となるサブルーチンまでの相対バイト値）を、Aレジスタにロード（セット）する。

【2595】

次いで、ソースコード「OR A」を実行して、Aレジスタにセットされている値とAレジスタにセットされている値とを論理和し、演算結果をAレジスタに格納する。この処理はCフラグの状態をリセット（クリア）するために行われる。次いで、ソースコード「ADDWB HL, A」を実行して、Aレジスタに格納されている値をHLレジスタに格納されている値に加算して、該加算された値をHLレジスタに格納する。この処理により、ジャンプ先となるサブルーチンの絶対アドレスが算出され、その絶対アドレスがHLレジスタに格納される。そして、ソースコード「JP (HL)」を実行して、HLレジスタにセットされている値をPCレジスタ（プログラム・カウンタPC）にロード（セット）する。この処理により、処理が所望のサブルーチンにジャンプし、サブルーチンの処理が開始される。

【2596】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した7つのソースコードを順次実行することにより、ジャンプ先のサブルーチンのアドレスに関するデータ（サブルーチンまでの相対バイト値）が格納されたジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせることができる。なお、このソースプログラムでは、ソースコード「LDQ B, (wAREA4)」、「LD HL, TBL__SUB」、「ADDWB HL, B」、「LD A, (HL)」、「OR A」、「ADDWB HL, A」及び「JP (HL)」のバイト数（容量）は、それぞれ、3バイト、3バイト、1バイト、1バイト、1バイト、1バイト及び1バイトであり、ジャンプ先テーブルのバイト数（容量）は、4バイトであるので、プログラム例14-Aのプログラムの合計バイト数は、15バイトとなる。

【2597】

一方、本実施形態で新たに設けられたジャンプ命令コードを用いた場合、図371Bに示すように、まず、ソースコード「LDQ B, (wAREA4)」を実行して、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wAREA4」（8ビットデータ：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Bレジスタにロード（セット）する。そして、ソースコード「TJP B」（新命令コード：図326中のNo. D7）を実行して、現在、PCレジスタにセットされている値（ソースコード「TJP B」（2バイト命令）の先頭アドレス）に、「2」とBレジスタに格納されている値とを加算（ $PC + 2 + B$ ）し、該加算した値で指定されるアドレスに格納されている値（ジャンプ先となるサブルーチンの下位側1バイトのアドレス）をPCレジスタの下位側バイトにロード（セット）する。この処理により、処理が所望のサブルーチンにジャンプし、サブルーチンの処理が開始される。

【2598】

なお、ソースコード「TJP B」内における処理では、まず、現在のPCレジスタにセットされている値（ソースコード「TJP B」（2バイト命令）の上位側1バイトのアドレス）に「2」を加算して、ジャンプ先テーブルの先頭アドレスを算出する。次いで、その算出値にBレジスタに格納されている値（Qレジスタに格納された値と「wAREA4」とで指定されるアドレスに格納されている値：「0」～「3」のいずれか）を加算して、ジャンプ先テーブル内における、所望のジャンプ先となるサブルーチンの下位側1バイトのアドレスが格納されたアドレスを算出する。それゆえ、ジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせるプログラムにおいて、ジャンプ命令コード「TJP B」を用いる場合には、ソースコード「TJP B」の直下のアドレス（ソースコード「TJP B」（2バイト命令）の先頭アドレスの2バイト先のアドレス）に、ジャンプ先テーブルの先頭アドレスを配置する必要がある。

【2599】

本実施形態で新たに設けられたジャンプ命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、ジャンプ先のサブルーチンのアドレスに関するデータ（サブルーチンの下位側アドレス）が格納されたジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせることができる。なお、このソースプログラムでは、ソースコード「LDQ B, (wAREA4)」及び「TJP B」のバイト数（容量）は、それぞれ、3バイト及び2バイトであり、ジャンプ先テーブルのバイト数（容量）は、4バイトであるので、プログラム例14-Bのプログラムの合計バイト数は、9バイトとなる。

【2600】

上述のように、プログラム例14においても、本実施形態で新たに設けられたジャンプ命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2601】

なお、プログラム例14-Bは、プログラム例12-Bで説明した処理を、「TJP」命令より1バイト容量の大きい「TJP B」命令を使用して実現した場合の処理である。プログラム例14-Bでは、最初にソースコード「LDQ B, (wAREA4)」を実行することにより、ジャンプ先テーブルに格納されているジャンプ先サブルーチンの下位側アドレスを取得している。しかしながら、実際のプログラムでは、この処理は、ある処理が実行され、その処理結果がBレジスタに格納される処理である。それゆえ、この処理によってBレジスタに格納された値を、「TJP」命令で使用する場合にはBレジスタに格納された値をAレジスタに格納する必要（処理）が発生する。それに対して、「TJP B」命令では、Bレジスタに格納された値をそのまま使用することができる。なお、「TJP B」命令以外の「TJP C」～「TJP L」命令においても同様のことが言える。すなわち、「TJP」命令だけでは、プログラム容量を減らす効果が不十分となる場合を考慮して、本実施形態では、「TJP B」～「TJP L」命令が設けられている。

【2602】

(4) プログラム例15

プログラム例15では、ジャンプ先のサブルーチンのアドレスデータが格納されたテーブル（ジャンプ先テーブル）を使用して、処理を所望のサブルーチンの処理にジャンプさせる機能（処理動作）を実現するためのプログラム例を説明する。図372A及び図372Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例15-A、及び、本実施形態で新たに設けられたジャンプ命令コードを用いて生成されたプログラム例15-Bを示す。

【2603】

なお、この例では、上記プログラム例13とは異なり、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wAREA4」（8ビットデータ：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Dレジスタにロード（セット）する例を説明する。また、この例で用いるジャンプ先テーブルの構成は、上記プログラム例13のそれ（図370A及び図370B参照）と同様であるので、その説明は省略する。

【2604】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図372Aに示すように、まず、ソースコード「LDQ D, (wAREA4)」を実行して、Qレジスタに格納された値（指定メモリの上位側アドレス）と「wAREA4」（8ビットデータ：指定メモリの下位側アドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Dレジスタにロード（セット）する。なお、この指定メモリに格納されている値は、上記プログラム例13と同様に、ジャンプ先テーブルの先頭アドレスからジャンプ先テーブル内の各データ（第1～第4サブルーチンのアドレス）の格納アドレスまでの相対値（「0」～「3」）である。

【2605】

次いで、ソースコード「LD HL, TBL__SUB」を実行して、HLレジスタにラベル「TBL__SUB」で示されるジャンプ先テーブルの先頭アドレスをロード（セット）する。次いで、ソースコード「ADDWB HL, D」を実行して、Dレジスタに格納されている値をHLレジスタに格納されている値に加算して、該加算された値をHLレジスタに格納する。この処理により、HLレジスタには、ジャンプ先となるサブルーチンのアドレスがセットされる。そして、ソースコード「JP (HL)」を実行して、HLレジスタにセットされている値をPCレジスタ（プログラム・カウンタPC）にロード（セット）する。この処理により、処理が所望のサブルーチンにジャンプし、サブルーチンの処理が開始される。

10

【2606】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した4つのソースコードを順次実行することにより、ジャンプ先のサブルーチンのアドレスが格納されたジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせることができる。なお、このソースプログラムでは、ソースコード「LDQ D, (wAREA4)」、「LD HL, TBL__SUB」、「ADDWB HL, D」及び「JP (HL)」のバイト数（容量）は、それぞれ、3バイト、3バイト、1バイト及び1バイトであり、ジャンプ先テーブルのバイト数（容量）は、8バイトであるので、プログラム例15-Aのプログラムの合計バイト数は、16バイトとなる。

【2607】

一方、本実施形態で新たに設けられたジャンプ命令コードを用いた場合、図372Bに示すように、まず、ソースコード「LDQ D, (wAREA4)」を実行して、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wAREA4」（8ビットデータ：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Dレジスタにロード（セット）する。なお、この指定メモリに格納されている値は、上記プログラム例13と同様に、ジャンプ先テーブルの先頭アドレスからジャンプ先テーブル内の各データ（第1～第4サブルーチンのアドレス）の格納アドレスまでの相対値（「0」～「3」）である。そして、ソースコード「TJPW D」（新命令コード：図327中のNo. D16）を実行して、現在、PCレジスタにセットされている値（ソースコード「TJPW D」（2バイト）の上位側1バイトのアドレス）に、「2」とDレジスタに格納されている値を2倍した値とを加算（ $PC + 2 + D \times 2$ ）し、該加算した値で指定されるアドレスに格納されている値（ジャンプ先となるサブルーチンのアドレス（2バイト））をPCレジスタにロード（セット）する。この処理により、処理が所望のサブルーチンにジャンプし、サブルーチンの処理が開始される。

20

30

【2608】

なお、ソースコード「TJPW D」内における処理では、まず、現在のPCレジスタにセットされている値（ソースコード「TJPW D」（2バイト命令）の上位側1バイトのアドレス）に「2」を加算して、ジャンプ先テーブルの先頭アドレスを算出する。次いで、その算出値にDレジスタに格納されている値（Qレジスタに格納された値と「wAREA4」とで指定されるアドレスに格納されている値：「0」～「3」のいずれか）を加算して、ジャンプ先テーブル内における、所望のジャンプ先となるサブルーチンのアドレスが格納されたアドレスが算出される。それゆえ、ジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせるプログラムにおいて、ジャンプ命令コード「TJPW D」を用いる場合には、ソースコード「TJPW D」の直下のアドレス（ソースコード「TJPW D」（2バイト命令）の下位側1バイトのアドレスの次のアドレス）に、ジャンプ先テーブルの先頭アドレスを配置する必要がある。

40

【2609】

本実施形態で新たに設けられたジャンプ命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、ジャンプ先のサブルーチンのアドレスが格納され

50

たジャンプ先テーブルを使用して、処理を所望のサブルーチンの処理にジャンプさせることができる。なお、このソースプログラムでは、ソースコード「LDQ D, (wARE A 4)」及び「TJPW D」のバイト数（容量）は、それぞれ、3バイト及び2バイトであり、ジャンプ先テーブルのバイト数（容量）は、8バイトであるので、プログラム例15-Bのプログラムの合計バイト数は、13バイトとなる。

【2610】

上述のように、プログラム例15においても、本実施形態で新たに設けられたジャンプ命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2611】

なお、プログラム例15-Bは、プログラム例13-Bで説明した処理を、「TJPW」命令より1バイト容量の大きい「TJPW D」命令を使用して実現した場合の処理である。プログラム例15-Bでは、最初にソースコード「LDQ D, (wARE A 4)」を実行することにより、ジャンプ先テーブルに格納されているジャンプ先サブルーチンの下位側アドレスを取得している。しかしながら、実際のプログラムでは、この処理は、ある処理が実行され、その処理結果がDレジスタに格納される処理である。それゆえ、この処理によってDレジスタに格納された値を、「TJPW」命令で使用する場合にはDレジスタに格納された値をAレジスタに格納する必要（処理）が発生する。それに対して、「TJPW D」命令では、Dレジスタに格納された値をそのまま使用することができる。なお、「TJPW D」命令以外の「TJPW B」、「TJPW C」、「TJPW E」～「TJPW L」命令においても同様のことが言える。すなわち、「TJPW」命令だけでは、プログラム容量を減らす効果が不十分となる場合を考慮して、本実施形態では、「TJPW B」～「TJPW L」命令が設けられている。

【2612】

〔コンペアロード命令（加算）コードを使用したプログラム例〕

次に、本実施形態で新たに設けられたコンペアロード命令（加算）コードを使用した各種プログラム例を説明する。

【2613】

（1）プログラム例16

プログラム例16では、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wARE A 5」（8ビットの整数値：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている所定値（8ビットデータ）と、その上限値とを比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図373A及び図373Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例16-A、及び、本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いて生成されたプログラム例16-Bを示す。

【2614】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図373Aに示すように、まず、ソースコード「LDQ A, (wARE A 5)」を実行して、Qレジスタに格納された値と「wARE A 5」（8ビットの整数値）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Aレジスタにロード（セット）する。なお、この指定メモリに格納された値は、上限値との比較対象となるパラメータ値（所定値）である。

【2615】

次いで、ソースコード「INC A」を実行して、Aレジスタに格納されている値に1を加算し、該加算された値をAレジスタに格納（セット）する。次いで、ソースコード「CP 10」を実行して、Aレジスタに格納されている値と、「10」（上限値）とを比較し、比較結果に基づいてCフラグ（CYフラグ）の状態を設定する。次いで、ソースコー

10

20

30

40

50

ド「J R C, L A B E L E 1」を実行する。この処理において、Cフラグ=1（Aレジスタに格納されている値が「10」以下である場合）であれば、ラベル「L A B E L E 1」の処理にジャンプする。一方、Cフラグ=0（Aレジスタに格納されている値が「10」より大きい場合）であれば、ソースコード「L D A, 10」を実行して、Aレジスタに「10」をセット（ロード）する。そして、ソースコード「L D A, 10」又はソースコード「J R C, L A B E L E 1」の実行後、ラベル「L A B E L E 1」の処理、すなわち、ソースコード「L D Q (w A R E A 5), A」を実行して、Aレジスタに格納されている値を、Qレジスタに格納された値と「w A R E A 5」（8ビットの整数値）とで指定されるアドレス領域（指定メモリ）に格納（セット）する。

【2616】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した6つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、その上限値とを比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「L D Q A, (w A R E A 5)」、「I N C A」、「C P 10」、「J R C, L A B E L E 1」、「L D A, 10」及び「L D Q (w A R E A 5), A」のバイト数（容量）は、それぞれ、2バイト、1バイト、2バイト、2バイト、2バイト及び2バイトとなり、プログラム例16-Aのプログラムの合計バイト数は、11バイトとなる。

【2617】

一方、本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合、図373Bに示すように、まず、ソースコード「L D Q A, (w A R E A 5)」を実行して、Qレジスタに格納された値と「w A R E A 5」（8ビットの整数値）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Aレジスタにロード（セット）する。次いで、ソースコード「I C P L L A, 10」（新命令コード：図330中のNo. E13）を実行して、Aレジスタに格納されている値と「10」（上限値）とを比較し、「10」（上限値）の方がAレジスタに格納されている値より大きい場合には、Aレジスタに格納されている値に1を加算し、該1加算された値をAレジスタにセット（ロード）し、一方、「10」（上限値）の方がAレジスタに格納されている値より大きくない場合（Aレジスタに格納されている値 $\geq$ 10）には、Aレジスタに「10」をセット（ロード）する。次いで、「L D Q (w A R E A 5), A」を実行して、Aレジスタに格納されている値を、Qレジスタに格納された値と「w A R E A 5」（8ビットデータ）とで指定されるアドレス領域（指定メモリ）に格納（セット）する。

【2618】

本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合には、上述した3つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、その上限値とを比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「L D Q A, (w A R E A 5)」、「I C P L L A, 10」及び「L D Q (w A R E A 5), A」のバイト数（容量）は、それぞれ、2バイト、3バイト及び2バイトであり、プログラム例16-Bのプログラムの合計バイト数は、7バイトとなる。

【2619】

上述のように、プログラム例16においても、本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2620】

（2）プログラム例17

プログラム例17では、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「w A R E A 5」（8ビットの整数値：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内の2バイト分のアドレス領域（指定メモリ：w A R E A 5 + 0, + 1）に格納されている所定値（16ビットデータ）と、その上限値とを

10

20

30

40

50

比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図374A及び図374Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例17-A、及び、本実施形態で新たに設けられたコンペロード命令（加算）コードを用いて生成されたプログラム例17-Bを示す。

【2621】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図374Aに示すように、まず、ソースコード「LDQ BC, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるメインRAM103内の2バイト分のアドレス領域（指定メモリ）に格納されている値を、BCレジスタにロード（セット）する。なお、この指定メモリに格納されている値は、上限値との比較対象となるパラメータ値（所定値）である。

【2622】

次いで、ソースコード「INC BC」を実行してBCレジスタに格納されている値に1を加算し、該加算された値をBCレジスタに格納（セット）する。次いで、ソースコード「CP BC, 1000」を実行して、BCレジスタに格納されている値と、「1000」（上限値）とを比較し、比較結果に基づいてCフラグ（CYフラグ）の状態を設定する。次いで、ソースコード「JR C, LABEL1」を実行する。この処理において、Cフラグ=1（BCレジスタに格納されている値が「1000」以下である場合）であれば、ラベル「LABEL1」の処理にジャンプする。一方、Cフラグ=0（BCレジスタに格納されている値が「1000」より大きい場合）であれば、ソースコード「LD BC, 1000」を実行して、BCレジスタに「1000」をセット（ロード）する。そして、ソースコード「LD BC, 1000」又はソースコード「JR C, LABEL1」の実行後、ラベル「LABEL1」の処理、すなわち、ソースコード「LDQ (wAREA5), BC」を実行して、BCレジスタに格納されている値を、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定される2バイト分のアドレス領域（指定メモリ）に格納（セット）する。

【2623】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した6つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、その上限値とを比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「LDQ BC, (wAREA5)」、「INC BC」、「CP BC, 1000」、「JR C, LABEL1」、「LD BC, 1000」及び「LDQ (wAREA5), BC」のバイト数（容量）は、それぞれ、3バイト、1バイト、4バイト、2バイト、3バイト及び3バイトとなり、プログラム例17-Aのプログラムの合計バイト数は、16バイトとなる。

【2624】

一方、本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合、図374Bに示すように、まず、ソースコード「LDQ BC, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるメインRAM103内の2バイト分のアドレス領域（指定メモリ）に格納されている値を、BCレジスタにロード（セット）する。次いで、ソースコード「ICPWL B C, 1000」（新命令コード：図331中のNo. E20）を実行して、BCレジスタに格納されている値と「1000」（上限値）とを比較し、「1000」（上限値）の方がBCレジスタに格納されている値より大きい場合には、BCレジスタに格納されている値に1を加算し、該1加算された値をBCレジスタにセット（ロード）し、一方、「1000」（上限値）の方がBCレジスタに格納されている値より大きくない場合（BCレジスタに格納されている値 $\geq$ 1000）には、BCレジスタに「1000」をセット（ロード）する。次いで、「LDQ (wAREA5), BC」を実行して、BCレジスタに格

納されている値を、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定される2バイト分のアドレス領域（指定メモリ）に格納（セット）する。

【2625】

本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合には、上述した3つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、その上限値とを比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「LDQ BC, (wAREA5)」、「ICPWL L BC, 1000」及び「LDQ (wAREA5), BC」のバイト数（容量）は、それぞれ、3バイト、4バイト及び3バイトであり、プログラム例17-Bのプログラムの合計バイト数は、10バイトとなる。

10

【2626】

上述のように、プログラム例17においても、本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2627】

（3）プログラム例18

プログラム例18では、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wAREA5」（8ビットの整数値：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている所定値（8ビットデータ）と、Cレジスタに格納された値（上限値）とを比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図375A及び図375Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例18-A、及び、本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いて生成されたプログラム例18-Bを示す。

20

【2628】

上記第1及び第2の実施形態の遊技機で使用されている命令コードを用いた場合、図375Aに示すように、まず、ソースコード「LDQ A, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Aレジスタにロード（セット）する。なお、この指定メモリに格納された値は、Cレジスタに格納された値（上限値）との比較対象となるパラメータ値（所定値）である。次いで、ソースコード「INC A」を実行して、Aレジスタに格納されている値に1を加算し、該加算された値をAレジスタに格納（セット）する。

30

【2629】

次いで、ソースコード「CP C」を実行して、Aレジスタに格納されている値と、Cレジスタに格納されている値（上限値）とを比較し、比較結果に基づいてCフラグ（CYフラグ）の状態を設定する。次いで、ソースコード「JR C, LABEL1」を実行する。この処理において、Cフラグ=1（Aレジスタに格納されている値がCレジスタに格納された値以下である場合）であれば、ラベル「LABEL1」の処理にジャンプする。一方、Cフラグ=0（Aレジスタに格納されている値がCレジスタに格納された値より大きい場合）であれば、ソースコード「LD A, C」を実行して、Cレジスタに格納されている値（上限値）をAレジスタにセット（ロード）する。そして、ソースコード「LD A, C」又はソースコード「JR C, LABEL1」の実行後、ラベル「LABEL1」の処理、すなわち、ソースコード「LDQ (wAREA5), A」を実行して、Aレジスタに格納されている値を、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるアドレス領域（指定メモリ）に格納（セット）する。

40

【2630】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述

50

した6つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、Cレジスタに格納されている値（上限値）とを比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wAREA5)」、「INC A」、「CP C」、「JR C, LABELE1」、「LD A, C」及び「LDQ (wAREA5), A」のバイト数（容量）は、それぞれ、2バイト、1バイト、1バイト、2バイト、1バイト及び2バイトとなり、プログラム例18-Aのプログラムの合計バイト数は、9バイトとなる。

【2631】

一方、本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合、
図375Bに示すように、まず、ソースコード「LDQ A, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Aレジスタにロード（セット）する。次いで、ソースコード「ICPLL A, C」（新命令コード：図329中のNo. E8）を実行して、Aレジスタに格納されている値とCレジスタに格納されている値（上限値）とを比較し、Cレジスタに格納されている値の方がAレジスタに格納されている値より大きい場合には、Aレジスタに格納されている値に1を加算し、該1加算された値をAレジスタにセット（ロード）し、一方、Cレジスタに格納されている値の方がAレジスタに格納されている値より大きくない場合（Aレジスタに格納されている値 $\geq$ Cレジスタに格納されている値）には、Cレジスタに格納されている値（上限値）をAレジスタにセット（ロード）する。次いで、「LDQ (wAREA5), A」を実行して、Aレジスタに格納されている値を、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるアドレス領域（指定メモリ）に格納（セット）する。

【2632】

本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合には、上述した3つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、Cレジスタに格納されている値（上限値）とを比較し、その比較結果に応じて、1加算された所定値又はその上限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wAREA5)」、「ICPLL A, C」及び「LDQ (wAREA5), A」のバイト数（容量）は、それぞれ、2バイト、3バイト及び2バイトであり、プログラム例18-Bのプログラムの合計バイト数は、7バイトとなる。

【2633】

上述のように、プログラム例18においても、本実施形態で新たに設けられたコンペアロード命令（加算）コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2634】

【コンペアロード命令（減算）コードを使用したプログラム例】

次に、本実施形態で新たに設けられたコンペアロード命令（減算）コードを使用した各種プログラム例を説明する。

【2635】

（1）プログラム例19

プログラム例19では、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wAREA5」（8ビットの整数値：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている所定値（8ビットデータ）と、その下限値とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図376A及び図376Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例19

ーA、及び、本実施形態で新たに設けられたコンペロード命令（減算）コードを用いて生成されたプログラム例19-Bを示す。

【2636】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図376Aに示すように、まず、ソースコード「LDQ A, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Aレジスタにロード（セット）する。なお、この指定メモリに格納された値は、下限値との比較対象となるパラメータ値（所定値）である。次いで、ソースコード「DEC A」を実行して、Aレジスタに格納されている値を1減算し、該減算された値をAレジスタに格納（セット）する。

10

【2637】

次いで、ソースコード「CP 10」を実行して、Aレジスタに格納されている値と、「10」（下限値）とを比較し、比較結果に基づいてCフラグ（CYフラグ）の状態を設定する。次いで、ソースコード「JR NC, LABEL1」を実行する。この処理において、Cフラグ=0（Aレジスタに格納されている値が「10」より大きい場合）であれば、ラベル「LABEL1」の処理にジャンプする。一方、Cフラグ=0（Aレジスタに格納されている値が「10」以下である場合）であれば、ソースコード「LD A, 10」を実行して、Aレジスタに「10」をセット（ロード）する。そして、ソースコード「LD A, 10」又はソースコード「JR NC, LABEL1」の実行後、ラベル「LABEL1」の処理、すなわち、ソースコード「LDQ (wAREA5), A」を実行して、Aレジスタに格納されている値を、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるアドレス領域（指定メモリ）に格納（セット）する。

20

【2638】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した6つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、その下限値とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wAREA5)」、「DEC A」、「CP 10」、「JR NC, LABEL1」、「LD A, 10」及び「LDQ (wAREA5), A」のバイト数（容量）は、それぞれ、2バイト、1バイト、2バイト、2バイト、2バイト及び2バイトとなり、プログラム例19-Aのプログラムの合計バイト数は、11バイトとなる。

30

【2639】

一方、本実施形態で新たに設けられたコンペアロード命令（減算）コードを用いた場合、図376Bに示すように、まず、ソースコード「LDQ A, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるメインRAM103内のアドレス領域（指定メモリ）に格納されている値を、Aレジスタにロード（セット）する。次いで、ソースコード「DCPLL A, 10」（新命令コード：図335中のNo. F13）を実行して、Aレジスタに格納されている値と「10」（下限値）とを比較し、「10」（下限値）の方がAレジスタに格納されている値より小さい場合、Aレジスタに格納されている値を1減算し、該1減算された値をAレジスタにセット（ロード）し、一方、「10」（下限値）の方がAレジスタに格納されている値より小さくない場合（Aレジスタに格納されている値 $\leq$ 10）、Aレジスタに「10」をセット（ロード）する。次いで、「LDQ (wAREA5), A」を実行して、Aレジスタに格納されている値を、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるアドレス領域（指定メモリ）に格納（セット）する。

40

【2640】

本実施形態で新たに設けられたコンペアロード命令（減算）コードを用いた場合には、上

50

述した3つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、その下限値とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wAREA5)」、「DCPLL A, 10」及び「LDQ (wAREA5), A」のバイト数（容量）は、それぞれ、2バイト、3バイト及び2バイトであり、プログラム例19-Bのプログラムの合計バイト数は、7バイトとなる。

【2641】

上述のように、プログラム例19においても、本実施形態で新たに設けられたコンペアロード命令（減算）コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2642】

（2）プログラム例20

プログラム例20では、Qレジスタに格納された値（指定メモリの上位側1バイトのアドレス）と「wAREA5」（8ビットの整数値：指定メモリの下位側1バイトのアドレス）とで指定されるメインRAM103内のアドレス領域（指定メモリ：wAREA5+0, +1）に格納されている所定値（16ビットデータ）と、その下限値とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納（セット）する機能（処理動作）を実現するためのプログラム例を説明する。図377A及び図377Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例20-A、及び、本実施形態で新たに設けられたコンペアロード命令（減算）コードを用いて生成されたプログラム例20-Bを示す。

【2643】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図377Aに示すように、まず、ソースコード「LDQ BC, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定されるメインRAM103内の2バイト分のアドレス領域（指定メモリ）に格納されている値を、BCレジスタにロード（セット）する。なお、この指定メモリに格納されている値は、下限値との比較対象となるパラメータ値（所定値）である。次いで、ソースコード「DEC BC」を実行してBCレジスタに格納されている値を1減算し、該減算された値をBCレジスタに格納（セット）する。

【2644】

次いで、ソースコード「CP BC, 1000」を実行して、BCレジスタに格納されている値と、「1000」（下限値）とを比較し、比較結果に基づいてCフラグ（CYフラグ）の状態を設定する。次いで、ソースコード「JR NC, LABEL1」を実行する。この処理において、Cフラグ=0（BCレジスタに格納されている値が「1000」以上である場合）であれば、ラベル「LABEL1」の処理にジャンプする。一方、Cフラグ=1（BCレジスタに格納されている値が「1000」より小さい場合）であれば、ソースコード「LD BC, 1000」を実行して、BCレジスタに「1000」をセット（ロード）する。そして、ソースコード「LD BC, 1000」又はソースコード「JR NC, LABEL1」の実行後、ラベル「LABEL1」の処理、すなわち、ソースコード「LDQ (wAREA5), BC」を実行して、BCレジスタに格納されている値を、Qレジスタに格納された値と「wAREA5」（8ビットの整数値）とで指定される2バイト分のアドレス領域（指定メモリ）に格納（セット）する。

【2645】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した6つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、その下限値とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納（セット）することができる。なお、このソースプログラムでは、ソースコード「LDQ BC, (wAREA5)」、「DEC BC」、「CP BC, 1000」、「JR NC, LABEL1」、「LD BC, 1000」及び「LDQ (w

10

20

30

40

50

AREA5), BC」のバイト数(容量)は、それぞれ、3バイト、1バイト、4バイト、2バイト、3バイト及び3バイトとなり、プログラム例20-Aのプログラムの合計バイト数は、16バイトとなる。

【2646】

一方、本実施形態で新たに設けられたコンペアロード命令(減算)コードを用いた場合、図377Bに示すように、まず、ソースコード「LDQ BC, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」(8ビットの整数値)とで指定されるメインRAM103内の2バイト分のアドレス領域(指定メモリ)に格納されている値を、BCレジスタにロード(セット)する。次いで、ソースコード「DCPWL BC, 1000」(新命令コード:図336中のNo. F20)を実行して、BCレジスタに格納されている値と「1000」(下限値)とを比較し、「1000」(下限値)の方がBCレジスタに格納されている値より小さい場合、BCレジスタに格納されている値を1減算し、該1減算された値をBCレジスタにセット(ロード)し、一方、「1000」(下限値)の方がBCレジスタに格納されている値より小さくない場合(BCレジスタに格納されている値 $\leq$ 1000)、BCレジスタに「1000」をセット(ロード)する。次いで、「LDQ (wAREA5), BC」を実行して、BCレジスタに格納されている値を、Qレジスタに格納された値と「wAREA5」(8ビットの整数値)とで指定される2バイト分のアドレス領域(指定メモリ)に格納(セット)する。

【2647】

本実施形態で新たに設けられたコンペアロード命令(減算)コードを用いた場合には、上述した3つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、その下限値とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納(セット)することができる。なお、このソースプログラムでは、ソースコード「LDQ BC, (wAREA5)」、「DCPWL BC, 1000」及び「LDQ (wAREA5), BC」のバイト数(容量)は、それぞれ、3バイト、4バイト及び3バイトであり、プログラム例20-Bのプログラムの合計バイト数は、10バイトとなる。

【2648】

上述のように、プログラム例20においても、本実施形態で新たに設けられたコンペアロード命令(減算)コードを用いた場合、それを用いない場合に比べて、同じ機能(処理動作)を実現するためのプログラムの容量(バイト数)を減らすことができる。

【2649】

(3) プログラム例21

プログラム例21では、Qレジスタに格納された値(指定メモリの上位側1バイトのアドレス)と「wAREA5」(8ビットの整数値:指定メモリの下位側1バイトのアドレス)とで指定されるメインRAM103内のアドレス領域(指定メモリ)に格納されている所定値(8ビットデータ)と、Cレジスタに格納されている値(下限値)とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納(セット)する機能(処理動作)を実現するためのプログラム例を説明する。図378A及び図378Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例21-A、及び、本実施形態で新たに設けられたコンペアロード命令(減算)コードを用いて生成されたプログラム例21-Bを示す。

【2650】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図378Aに示すように、まず、ソースコード「DECQ (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」(8ビットの整数値)とで指定されるメインRAM103内のアドレス領域(指定メモリ)に格納されている値を1減算する。なお、この指定メモリに格納されている値は、Cレジスタに格納されている値(下限値)と比較対象となるパラメータ値(所定値)である。次いで、ソースコード「LDQ A, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」(8ビットの整数

10

20

30

40

50

値)とで指定されるアドレス領域(指定メモリ)に格納されている値を、Aレジスタにロード(セット)する。

【2651】

次いで、ソースコード「CP C」を実行して、Aレジスタに格納されている値と、Cレジスタに格納されている値(下限値)とを比較し、比較結果に基づいてCフラグ(CYフラグ)の状態を設定する。次いで、ソースコード「JR NC, LABEL1」を実行する。この処理において、Cフラグ=0(Aレジスタに格納されている値がCレジスタに格納されている値以上である場合)であれば、ラベル「LABEL1」の処理にジャンプし、処理を終了する。一方、Cフラグ=1(Aレジスタに格納されている値がCレジスタに格納されている値より小さい場合)であれば、ソースコード「LDQ (wAREA5), C」を実行して、Qレジスタに格納された値と「wAREA5」(8ビットの整数値)とで指定されるアドレス領域(指定メモリ)にCレジスタに格納されている値(下限値)をセット(ロード)する。

【2652】

上記第1及び第2の実施形態の遊技機で使用されている命令コードを用いた場合には、上述した5つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、Cレジスタに格納されている値(下限値)とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納(セット)することができる。なお、このソースプログラムでは、ソースコード「DECQ (wAREA5)」、「LDQ A, (wAREA5)」、「CP C」、「JR NC, LABEL1」及び「LDQ (wAREA5), C」のバイト数(容量)は、それぞれ、3バイト、2バイト、1バイト、2バイト及び2バイトとなり、プログラム例21-Aのプログラムの合計バイト数は、10バイトとなる。

【2653】

一方、本実施形態で新たに設けられたコンペアロード命令(減算)コードを用いた場合、図378Bに示すように、まず、ソースコード「LDQ A, (wAREA5)」を実行して、Qレジスタに格納された値と「wAREA5」(8ビットの整数値)とで指定されるメインRAM103内のアドレス領域(指定メモリ)に格納されている値を、Aレジスタにロード(セット)する。次いで、ソースコード「DCPLL A, C」(新命令コード:図334中のNo. F8)を実行して、Aレジスタに格納されている値とCレジスタに格納されている値(下限値)とを比較し、Cレジスタに格納されている値の方がAレジスタに格納されている値より小さい場合、Aレジスタに格納されている値を1減算し、該1減算された値をAレジスタにセット(ロード)し、一方、Cレジスタに格納されている値の方がAレジスタに格納されている値より小さくない場合(Aレジスタに格納されている値 $\leq$ Cレジスタに格納されている値)、Cレジスタに格納されている値をAレジスタにセット(ロード)する。次いで、「LDQ (wAREA5), A」を実行して、Aレジスタに格納されている値を、Qレジスタに格納された値と「wAREA5」(8ビットの整数値)とで指定されるアドレス領域(指定メモリ)に格納(セット)する。

【2654】

本実施形態で新たに設けられたコンペアロード命令(減算)コードを用いた場合には、上述した3つのソースコードを順次実行することにより、指定メモリに格納されている所定値と、Cレジスタに格納されている値(下限値)とを比較し、その比較結果に応じて、1減算された所定値又はその下限値を指定メモリに格納(セット)することができる。なお、このソースプログラムでは、ソースコード「LDQ A, (wAREA5)」、「DCPLL A, C」及び「LDQ (wAREA5), A」のバイト数(容量)は、それぞれ、2バイト、2バイト及び2バイトであり、プログラム例21-Bのプログラムの合計バイト数は、6バイトとなる。

【2655】

上述のように、プログラム例21においても、本実施形態で新たに設けられたコンペアロード命令(減算)コードを用いた場合、それを用いない場合に比べて、同じ機能(処理動

10

20

30

40

50

作)を実現するためのプログラムの容量(バイト数)を減らすことができる。

【2656】

[シフト命令(左)コード及びシフト命令(右)コードを使用したプログラム例]

次に、本実施形態で新たに設けられたシフト命令(左)コード及びシフト命令(右)コードを使用した各種プログラム例を説明する。

【2657】

(1)プログラム例22

プログラム例22では、Aレジスタ内の特定ビット(この例では、ビット3)及びそれより上位側に位置する全てのビット(ビット4~7)の値を上位側(左)に1ビット分シフトさせるとともに、特定ビットの位置より下位側に位置する全てのビット(ビット2~ビット0)の値をシフトさせずに維持する機能(処理動作)を実現するためのプログラム例を説明する。図379A及び図379Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例22-A、及び、本実施形態で新たに設けられたシフト命令(左)コードを用いて生成されたプログラム例22-Bを示す。

【2658】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図379Aに示すように、まず、ソースコード「LD B, A」を実行して、Aレジスタに格納されている値をBレジスタに格納(セット)する。この処理により、Aレジスタに格納されていた元の値をBレジスタに退避させる。次いで、ソースコード「AND 11111000 B」を実行して、レジスタAに格納されている値と、8ビットデータ「11111000 B」とを論理積し、この演算結果をAレジスタに格納(セット)する。この処理により、レジスタAに格納されている値(8ビットデータ)のうち、ビット2~ビット0の値がクリアされ(「0」がセットされ)、論理積演算後の8ビットデータがAレジスタに格納(セット)される。

【2659】

次いで、ソースコード「SLA A」を実行して、Aレジスタに格納されている各ビットの値を上位側(左)に1ビット分シフトするとともに、シフト前のビット7の値をCフラグ(CYフラグ)にセットし、シフト後のビット0には「0」をセットする。次いで、ソースコード「LD C, A」を実行して、Aレジスタに格納されている値(シフト処理後の8ビットデータ)をCレジスタに格納(セット)する。この処理により、Aレジスタに格納されているシフト処理後の値をCレジスタに退避させる。次いで、ソースコード「LD A, B」を実行して、Bレジスタに格納されている値(Aレジスタに格納されていた元の値)をAレジスタに格納(セット)する。すなわち、Aレジスタに格納されていた元の値をAレジスタに復帰させる。

【2660】

次いで、ソースコード「AND 00000111 B」を実行して、レジスタAに格納されている値と、8ビットデータ「00000111 B」とを論理積し、この演算結果をAレジスタに格納(セット)する。この処理では、Aレジスタに格納されていた元の値(8ビットデータ)において、シフト対象となるビット7~ビット3の値がクリアされ、且つ、Aレジスタに格納されていた元の値(8ビットデータ)中のビット3より下位側に位置する全てのビット(ビット2~ビット0)の値が元の値に維持された8ビットデータが生成され、その値がAレジスタに格納される。

【2661】

そして、ソースコード「OR C」を実行して、Aレジスタに格納されている値(ビット7~ビット3がクリアされ、且つ、ビット2~ビット0において元の値が維持されている8ビットデータ)と、Cレジスタに格納されている値(ビット2~ビット0がクリアされ、且つ、ビット7~ビット3に対してシフト処理が行われている8ビットデータ)とを論理和し、この演算結果をAレジスタに格納(セット)する。この処理により、ビット2~ビット0では元の値が維持され、且つ、ビット7~ビット3ではシフト処理後の値がセッ

トされた8ビットデータがAレジスタに格納（セット）される。

【2662】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した7つのソースコードを順次実行することにより、Aレジスタ内の特定ビットから上位側に位置する全てのビット（ビット3～7）を上位側（左）に1ビット分シフトさせ、且つ、特定ビットの位置より下位側に位置する全てのビット（ビット2～ビット0）の値（状態）を維持することができる。なお、このソースプログラムでは、ソースコード「LDB, A」、「AND 11111000B」、「SLA A」、「LD C, A」、「LDA, B」、「AND 00000111B」及び「OR C」のバイト数（容量）は、それぞれ、1バイト、2バイト、2バイト、1バイト、1バイト、2バイト及び1バイトとなり、プログラム例22-Aのプログラムの合計バイト数は、10バイトとなる。

10

【2663】

一方、本実施形態で新たに設けられたシフト命令（左）コードを用いた場合、図379Bに示すように、ソースコード「SLBIT A, 3」（新命令コード：図339中のNo. G3）を実行して、Aレジスタ内の指定ビット位置「3」のビット3から上位側の各ビット（ビット7～ビット3）の値を、上位側（左）に1ビット分シフト（移動）させるとともに、Aレジスタのシフト前のビット7の値をCフラグにセットし、シフト後のビット3に「0」をセットし、ビット3より下位側の各ビット（ビット2～ビット0）の値を維持する。

【2664】

20

本実施形態で新たに設けられたシフト命令（左）コードを用いた場合には、上述した1つのソースコードのみを実行することにより、Aレジスタ内の特定ビットから上位側に位置する全てのビット（ビット3～7）を上位側（左）に1ビット分シフトさせ、且つ、特定ビットの位置より下位側に位置する全てのビット（ビット2～ビット0）の値（状態）を維持することができる。なお、このソースプログラムでは、ソースコード「SLBIT A, 3」のバイト数（容量）は、3バイトであり、プログラム例22-Bのプログラムの合計バイト数も、3バイトとなる。

【2665】

上述のように、プログラム例22においても、本実施形態で新たに設けられたシフト命令（左）コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

30

【2666】

（2）プログラム例23

プログラム例23では、Aレジスタ内の特定ビット（この例では、ビット3）及びそれより下位側に位置する全てのビット（ビット2～0）の値を下位側（右）に1ビット分シフトさせるとともに、特定ビットの位置より上位側に位置する全てのビット（ビット7～ビット4）の値をシフトさせずに維持する機能（処理動作）を実現するためのプログラム例を説明する。図380A及び図380Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例23-A、及び、本実施形態で新たに設けられたシフト命令（右）コードを用いて生成されたプログラム例23-Bを示す。

40

【2667】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図380Aに示すように、まず、ソースコード「LDB, A」を実行して、Aレジスタに格納されている値をBレジスタに格納（セット）する。この処理により、Aレジスタに格納されていた元の値をBレジスタに退避させる。次いで、ソースコード「AND 00001111B」を実行して、レジスタAに格納されている値と、8ビットデータ「00001111B」とを論理積し、この演算結果をAレジスタに格納（セット）する。この処理により、レジスタAに格納されている値（8ビットデータ）のうち、ビット7～ビット4の値がクリアされ（「0」がセットされ）、論理積演算後の8ビットデータがAレジスタに格

50

納（セット）される。

【2668】

次いで、ソースコード「SRA A」を実行して、Aレジスタに格納されている各ビットの値を下位側（右）に1ビット分シフトするとともに、シフト前のビット0の値をCフラグ（CYフラグ）にセットし、シフト後のビット7にはシフト前の値をセットする。次いで、ソースコード「LDC, A」を実行して、Aレジスタに格納されている値（シフト処理後の8ビットデータ）をCレジスタに格納（セット）する。この処理により、Aレジスタに格納されているシフト処理後の値をCレジスタに退避させる。次いで、ソースコード「LDA, B」を実行して、Bレジスタに格納されている値（Aレジスタに格納されていた元の値）をAレジスタに格納（セット）する。すなわち、Aレジスタに格納されていた元の値をAレジスタに復帰させる。

10

【2669】

次いで、ソースコード「AND 11110000B」を実行して、レジスタAに格納されている値と、8ビットデータ「11110000B」とを論理積し、この演算結果をAレジスタに格納（セット）する。この処理では、Aレジスタに格納されていた元の値（8ビットデータ）において、シフト対象となるビット3～ビット0の値がクリアされ、且つ、Aレジスタに格納されていた元の値（8ビットデータ）中のビット3より上位側に位置する全てのビット（ビット7～ビット4）の値が元の値に維持された8ビットデータが生成され、その値がAレジスタに格納される。

20

【2670】

そして、ソースコード「OR C」を実行して、Aレジスタに格納されている値（ビット3～ビット0がクリアされ、且つ、ビット7～ビット4において元の値が維持されている8ビットデータ）と、Cレジスタに格納されている値（ビット7～ビット4がクリアされ、且つ、ビット3～ビット0に対してシフト処理が行われている8ビットデータ）とを論理和し、この演算結果をAレジスタに格納（セット）する。この処理により、ビット7～ビット4では元の値が維持され、且つ、ビット3～ビット0ではシフト処理後の値がセットされた8ビットデータがAレジスタに格納（セット）される。

【2671】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した7つのソースコードを順次実行することにより、Aレジスタ内の特定ビットから下位側に位置する全てのビット（ビット3～0）を下位側（右）に1ビット分シフトさせ、且つ、特定ビットの位置より上位側に位置する全てのビット（ビット7～ビット4）の値（状態）を維持することができる。なお、このソースプログラムでは、ソースコード「LDB, A」、「AND 00001111B」、「SRA A」、「LDC, A」、「LDA, B」、「AND 11110000B」及び「OR C」のバイト数（容量）は、それぞれ、1バイト、2バイト、2バイト、1バイト、1バイト、2バイト及び1バイトとなり、プログラム例23-Aのプログラムの合計バイト数は、10バイトとなる。

30

【2672】

一方、本実施形態で新たに設けられたシフト命令（右）コードを用いた場合、図380Bに示すように、ソースコード「SRBIT A, 3」（新命令コード：図341中のNo. H3）を実行して、Aレジスタ内の指定ビット位置「3」のビット3から下位側の各ビット（ビット3～ビット0）の値を、下位側（右）に1ビット分シフト（移動）させるとともに、Aレジスタのシフト前のビット0の値をCフラグにセットし、シフト後のビット3に「0」をセットし、ビット3より上位側の各ビット（ビット7～ビット4）の値を維持する。

40

【2673】

本実施形態で新たに設けられたシフト命令（右）コードを用いた場合には、上述した1つのソースコードのみを実行することにより、Aレジスタ内の特定ビットから下位側に位置する全てのビット（ビット3～0）を下位側（右）に1ビット分シフトさせ、且つ、特定ビットの位置より上位側に位置する全てのビット（ビット7～ビット4）の値（状態）を

50

維持することができる。なお、このソースプログラムでは、ソースコード「SRBIT A, 3」のバイト数（容量）は、3バイトであり、プログラム例23-Bのプログラムの合計バイト数も、3バイトとなる。

【2674】

上述のように、プログラム例23においても、本実施形態で新たに設けられたシフト命令（右）コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2675】

〔コール命令コードを使用したプログラム例〕

次に、本実施形態で新たに設けられたコール命令コードを使用した各種プログラム例を説明する。

【2676】

（1）プログラム例24

プログラム例24では、割込み禁止処理を行った後に、サブルーチンを呼び出す（コールする）機能（処理動作）を実現するためのプログラム例を説明する。図381A及び図381Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例24-A、及び、本実施形態で新たに設けられたコール命令コードを用いて生成されたプログラム例24-Bを示す。

【2677】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図381Aに示すように、まず、ソースコード「DI」を実行して、割込みイネーブルフリップフロップをリセットし、INT割込みを無効（割込み禁止）にする。次いで、ソースコード「CALL SUB__PROC__1」を実行して、当該命令コードの次の命令コードの番地（戻り先のアドレス値）に対応するPCレジスタ（プログラム・カウンタPC）の値を現在のスタックポインタSPに格納されている値により指示される2バイト分のメモリ領域（スタック領域）に格納するとともに、ジャンプ先のサブルーチンのアドレス値「SUB__PROC__1」（番地）をPCレジスタに格納する。この処理により、処理の実行番地が「SUB__PROC__1」番地にジャンプし、サブルーチンの処理が開始される。なお、ソースコード「CALL SUB__PROC__1」の処理では、上記処理動作とともに、スタックポインタSPの値が更新される（現在のスタックポインタSPの値から2減算した値（ $SP-2$ ）がスタックポインタSPに格納される）。

【2678】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、割込み禁止処理を行った後に、サブルーチンを呼び出すことができる。なお、このソースプログラムでは、ソースコード「DI」及び「CALL SUB__PROC__1」のバイト数（容量）は、それぞれ、1バイト及び3バイトとなり、プログラム例24-Aのプログラムの合計バイト数は、4バイトとなる。

【2679】

一方、本実施形態で新たに設けられたコール命令コードを用いた場合、図381Bに示すように、ソースコード「CALLD SUB__PROC__1」（新命令コード：図343中のNo. I2）を実行して、まず、割込みイネーブルフリップフロップをリセットしてINT割込みを無効（割込み禁止）にし、次いで、当該命令コードの次の命令コードの番地（戻り先のアドレス値）に対応するPCレジスタ（プログラム・カウンタPC）の値を現在のスタックポインタSPに格納されている値により指示される2バイト分のメモリ領域（スタック領域）に格納するとともに、ジャンプ先のサブルーチンのアドレス値「SUB__PROC__1」（番地）をPCレジスタに格納する。この処理により、処理の実行番地が「SUB__PROC__1」番地にジャンプし、サブルーチンの処理が開始される。なお、ソースコード「CALLD SUB__PROC__1」の処理においても、上記処理動作とともに、スタックポインタSPの値が更新される（現在のスタックポインタSPの値

10

20

30

40

50

から2減算した値（ $SP-2$ ）がスタックポインタ SP に格納される）。

【2680】

本実施形態で新たに設けられたコール命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、割込み禁止処理を行った後に、サブルーチン呼び出すことができる。なお、このソースプログラムでは、ソースコード「CALLD SUB__PROC__1」のバイト数（容量）は、3バイトであり、プログラム例24-Bのプログラムの合計バイト数も、3バイトとなる。

【2681】

上述のように、プログラム例24においても、本実施形態で新たに設けられたコール命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2682】

（2）プログラム例25

プログラム例25では、割込み禁止処理を行った後に、サブルーチン呼び出す（コールする）機能（処理動作）を実現するための別のプログラム例を説明する。図382A及び図382Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例25-A、及び、本実施形態で新たに設けられたコール命令コードを用いて生成されたプログラム例25-Bを示す。

【2683】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図382Aに示すように、まず、ソースコード「DI」を実行して、割込みイネーブルフリップフロップをリセットし、INT割込みを無効（割込み禁止）にする。次いで、ソースコード「CALLF SUB__PROC__1」を実行して、当該命令コードの次の命令コードの番地（戻り先のアドレス値）に対応するPCレジスタ（プログラム・カウンタPC）の値を現在のスタックポインタ SP に格納されている値により指示される2バイト分のメモリ領域（スタック領域）に格納するとともに、ジャンプ先のサブルーチンのアドレス値「SUB__PROC__1」（番地）をPCレジスタに格納する。この際、「00H」～「11H」の値（予め決められた8ビットの整数値）がPCレジスタの上位側（PCh）に格納され、ジャンプ先のアドレス値「SUB__PROC__1」の下位側の8ビットデータがPCレジスタの下位側（PCL）に格納される。この処理により、処理の実行番地が「SUB__PROC__1」番地にジャンプし、サブルーチンの処理が開始される。なお、ソースコード「CALLF SUB__PROC__1」の処理では、上記処理動作とともに、スタックポインタ SP の値が更新される（現在のスタックポインタ SP の値から2減算した値（ $SP-2$ ）がスタックポインタ SP に格納される）。

【2684】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、割込み禁止処理を行った後に、サブルーチン呼び出すことができる。なお、このソースプログラムでは、ソースコード「DI」及び「CALLF SUB__PROC__1」のバイト数（容量）は、それぞれ、1バイト及び2バイトとなり、プログラム例25-Aのプログラムの合計バイト数は、3バイトとなる。

【2685】

一方、本実施形態で新たに設けられたコール命令コードを用いた場合、図382Bに示すように、ソースコード「CALLFD SUB__PROC__1」（新命令コード：図343中のNo. I5）を実行して、まず、割込みイネーブルフリップフロップをリセットしてINT割込みを無効（割込み禁止）にし、次いで、当該命令コードの次の命令コードの番地（戻り先のアドレス値）に対応するPCレジスタ（プログラム・カウンタPC）の値を現在のスタックポインタ SP に格納されている値により指示される2バイト分のメモリ領域（スタック領域）に格納するとともに、ジャンプ先のサブルーチンのアドレス値「SUB__PROC__1」（番地）をPCレジスタに格納する。この際、「00H」～「11

10

20

30

40

50

H」の値（予め決められた8ビットの整数値）がPCレジスタの上位側（PC_h）に格納され、ジャンプ先のアドレス値「SUB__PROC__1」の下位側の8ビットデータがPCレジスタの下位側（PC_l）に格納される。この処理により、処理の実行番地が「SUB__PROC__1」番地にジャンプし、サブルーチンの処理が開始される。なお、ソースコード「CALLFD SUB__PROC__1」の処理においても、スタックポインタSPの値が更新される（現在のスタックポインタSPの値から2減算した値（SP-2）がスタックポインタSPに格納される）。

【2686】

本実施形態で新たに設けられたコール命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、割込み禁止処理を行った後に、サブルーチンを呼び出すことができる。なお、このソースプログラムでは、ソースコード「CALLFD SUB__PROC__1」のバイト数（容量）は、2バイトであり、プログラム例25-Bのプログラムの合計バイト数も、2バイトとなる。

【2687】

上述のように、プログラム例25においても、本実施形態で新たに設けられたコール命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2688】

（3）プログラム例26

プログラム例26では、割込み許可処理を行った後に、サブルーチンを呼び出す（コールする）機能（処理動作）を実現するためのプログラム例を説明する。図383A及び図383Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例26-A、及び、本実施形態で新たに設けられたコール命令コードを用いて生成されたプログラム例26-Bを示す。

【2689】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図383Aに示すように、まず、ソースコード「EI」を実行して、割込みイネーブルフリップフロップをセットし、INT割込みを有効（割込み許可）にする。次いで、ソースコード「CALL SUB__PROC__1」を実行して、当該命令コードの次の命令コードの番地（戻り先のアドレス値）に対応するPCレジスタ（プログラム・カウンタPC）の値を現在のスタックポインタSPに格納されている値により指定されるアドレス値から始まる2バイト分のメモリ領域（スタック領域）に格納するとともに、ジャンプ先のサブルーチンのアドレス値「SUB__PROC__1」（番地）をPCレジスタに格納する。この処理により、処理の実行番地が「SUB__PROC__1」番地にジャンプし、サブルーチンの処理が開始される。なお、ソースコード「CALL SUB__PROC__1」の処理では、上記処理動作とともに、スタックポインタSPの値が更新される（現在のスタックポインタSPの値から2減算した値（SP-2）がスタックポインタSPに格納される）。

【2690】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、割込み許可処理を行った後に、サブルーチンを呼び出すことができる。なお、このソースプログラムでは、ソースコード「EI」及び「CALL SUB__PROC__1」のバイト数（容量）は、それぞれ、1バイト及び3バイトとなり、プログラム例26-Aのプログラムの合計バイト数は、4バイトとなる。

【2691】

一方、本実施形態で新たに設けられたコール命令コードを用いた場合、図383Bに示すように、ソースコード「CALLE SUB__PROC__1」（新命令コード：図343中のNo. I3）を実行して、まず、割込みイネーブルフリップフロップをセットしてINT割込みを有効（割込み許可）にし、次いで、当該命令コードの次の命令コードの番地（戻り先のアドレス値）に対応するPCレジスタ（プログラム・カウンタPC）の値を現

10

20

30

40

50

在のスタックポインタSPに格納されている値により指示される2バイト分のメモリ領域（スタック領域）に格納するとともに、ジャンプ先のサブルーチンのアドレス値「SUB__PROC__1」（番地）をPCレジスタに格納する。この処理により、処理の実行番地が「SUB__PROC__1」番地にジャンプし、サブルーチンの処理が開始される。なお、ソースコード「CALLLE SUB__PROC__1」の処理においても、上記処理動作とともに、スタックポインタSPの値が更新される（現在のスタックポインタSPの値から2減算した値（SP-2）がスタックポインタSPに格納される）。

【2692】

本実施形態で新たに設けられたコール命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、割込み許可処理を行った後に、サブルーチンを呼び出すことができる。なお、このソースプログラムでは、ソースコード「CALLLE SUB__PROC__1」のバイト数（容量）は、3バイトであり、プログラム例26-Bのプログラムの合計バイト数も、3バイトとなる。

10

【2693】

上述のように、プログラム例26においても、本実施形態で新たに設けられたコール命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2694】

（4）プログラム例27

プログラム例27では、割込み許可処理を行った後に、サブルーチンを呼び出す（コールする）機能（処理動作）を実現するための別のプログラム例を説明する。図384A及び図384Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例27-A、及び、本実施形態で新たに設けられたコール命令コードを用いて生成されたプログラム例27-Bを示す。

20

【2695】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図384Aに示すように、まず、ソースコード「EI」を実行して、割込みイネーブルフリップフロップをセットし、INT割込みを有効（割込み許可）にする。次いで、ソースコード「CALLLF SUB__PROC__1」を実行して、当該命令コードの次の命令コードの番地（戻り先のアドレス値）に対応するPCレジスタ（プログラム・カウンタPC）の値を現在のスタックポインタSPに格納されている値により指示される2バイト分のメモリ領域（スタック領域）に格納するとともに、ジャンプ先のサブルーチンのアドレス値「SUB__PROC__1」（番地）をPCレジスタに格納する。この際、「00H」～「11H」の値（予め決められた8ビットの整数値）がPCレジスタの上位側（PCh）に格納され、ジャンプ先のアドレス値「SUB__PROC__1」の下位側の8ビットデータがPCレジスタの下位側（PCL）に格納される。この処理により、処理の実行番地が「SUB__PROC__1」番地にジャンプし、サブルーチンの処理が開始される。なお、ソースコード「CALLLF SUB__PROC__1」の処理では、上記処理動作とともに、スタックポインタSPの値が更新される（現在のスタックポインタSPの値から2減算した値（SP-2）がスタックポインタSPに格納される）。

30

40

【2696】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、割込み許可処理を行った後に、サブルーチンを呼び出すことができる。なお、このソースプログラムでは、ソースコード「EI」及び「CALLLF SUB__PROC__1」のバイト数（容量）は、それぞれ、1バイト及び2バイトとなり、プログラム例27-Aのプログラムの合計バイト数は、3バイトとなる。

【2697】

一方、本実施形態で新たに設けられたコール命令コードを用いた場合、図384Bに示すように、ソースコード「CALLFE SUB__PROC__1」（新命令コード：図34

50

3中のNo. I 6) を実行して、まず、割込みイネーブルフリップフロップをセットしてINT割込みを有効(割込み許可)にし、次いで、当該命令コードの次の命令コードの番地(戻り先のアドレス値)に対応するPCレジスタ(プログラム・カウンタPC)の値を現在のスタックポインタSPに格納されている値により指示される2バイト分のメモリ領域(スタック領域)に格納するとともに、ジャンプ先のサブルーチンのアドレス値「SUB__PROC__1」(番地)をPCレジスタに格納する。この際、「00H」～「11H」の値(予め決められた8ビットの整数値)がPCレジスタの上位側(PC_h)に格納され、ジャンプ先のアドレス値「SUB__PROC__1」の下位側の8ビットデータがPCレジスタの下位側(PC_l)に格納される。この処理により、処理の実行番地が「SUB__PROC__1」番地にジャンプし、サブルーチンの処理が開始される。なお、ソースコード「CALLFD SUB__PROC__1」の処理においても、上記処理動作とともに、スタックポインタSPの値が更新される(現在のスタックポインタSPの値から2減算した値(SP-2)がスタックポインタSPに格納される)。

【2698】

本実施形態で新たに設けられたコール命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、割込み許可処理を行った後に、サブルーチンと呼び出すことができる。なお、このソースプログラムでは、ソースコード「CALLFE SUB__PROC__1」のバイト数(容量)は、2バイトであり、プログラム例27-Bのプログラムの合計バイト数も、2バイトとなる。

【2699】

上述のように、プログラム例27においても、本実施形態で新たに設けられたコール命令コードを用いた場合、それを用いない場合に比べて、同じ機能(処理動作)を実現するためのプログラムの容量(バイト数)を減らすことができる。

【2700】

【リターン命令コードを使用したプログラム例】

次に、本実施形態で新たに設けられたリターン命令コードを使用した各種プログラム例を説明する。

【2701】

(1) プログラム例28

プログラム例28では、割込み禁止処理を行った後に、処理の番地を元のプログラム内の所定の戻り先番地に戻す機能(処理動作)を実現するためのプログラム例を説明する。図385A及び図385Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例28-A、及び、本実施形態で新たに設けられたリターン命令コードを用いて生成されたプログラム例28-Bを示す。

【2702】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図385Aに示すように、まず、ソースコード「DI」を実行して、割込みイネーブルフリップフロップをリセットし、INT割込みを無効(割込み禁止)にする。次いで、ソースコード「RET」を実行して、現在のスタックポインタSPに格納されている値により指定されるアドレス値から始まる2バイトのメモリ領域に格納されている値をPCレジスタ(プログラム・カウンタPC)にセットする。この処理により、処理の実行番地が元のプログラム内の所定の戻り先番地に戻り、元のプログラム内の所定の戻り先番地から処理が開始される。なお、ソースコード「RET」の処理では、上記処理動作とともに、スタックポインタSPの値が更新される(現在のスタックポインタSPの値から2加算した値(SP+2)がスタックポインタSPに格納される)。

【2703】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、割込み禁止処理を行った後に、処理の番地を元のプログラム内の所定の戻り先番地に戻すことができる。なお、このソースプログラムでは、ソースコード「DI」及び「RET」のバイト数(容量)は、それぞれ、

1バイト及び1バイトとなり、プログラム例28-Aのプログラムの合計バイト数は、2バイトとなる。

【2704】

一方、本実施形態で新たに設けられたリターン命令コードを用いた場合、図385Bに示すように、ソースコード「RETD」（新命令コード：図344中のNo. I8）を実行して、まず、割込みイネーブルフリップフロップをリセットしてINT割込みを無効（割込み禁止）にし、次いで、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域に格納されている値をPCレジスタ（プログラム・カウンタPC）にセットする。この処理により、処理の実行番地が元のプログラム内の所定の戻り先番地に戻り、元のプログラム内の所定の戻り先番地から処理が開始される。なお、ソースコード「RETD」の処理では、上記処理動作とともに、スタックポインタSPの値が更新される（現在のスタックポインタSPの値から2加算した値（SP+2）がスタックポインタSPに格納される）。

10

【2705】

本実施形態で新たに設けられたリターン命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、割込み禁止処理を行った後に、処理の番地を元のプログラム内の所定の戻り先番地に戻すことができる。なお、このソースプログラムでは、ソースコード「RETD」のバイト数（容量）は、1バイトであり、プログラム例28-Bのプログラムの合計バイト数も、1バイトとなる。

【2706】

20

上述のように、プログラム例28においても、本実施形態で新たに設けられたリターン命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2707】

（2）プログラム例29

プログラム例29では、割込み許可処理を行った後に、処理の番地を元のプログラム内の所定の戻り先番地に戻す機能（処理動作）を実現するためのプログラム例を説明する。図386A及び図386Bに、それぞれ、上記第1、第2実施形態の遊技機で使用されている命令コードを用いて生成されたプログラム例29-A、及び、本実施形態で新たに設けられたリターン命令コードを用いて生成されたプログラム例29-Bを示す。

30

【2708】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合、図386Aに示すように、まず、ソースコード「EI」を実行して、割込みイネーブルフリップフロップをセットし、INT割込みを有効（割込み許可）にする。次いで、ソースコード「RET」を実行して、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域に格納されている値をPCレジスタ（プログラム・カウンタPC）にセットする。この処理により、処理の実行番地が元のプログラム内の所定の戻り先番地に戻り、元のプログラム内の所定の戻り先番地から処理が開始される。なお、ソースコード「RET」の処理では、上記処理動作とともに、スタックポインタSPの値が更新される（現在のスタックポインタSPの値から2加算した値（SP+2）がスタックポインタSPに格納される）。

40

【2709】

上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合には、上述した2つのソースコードを順次実行することにより、割込み許可処理を行った後に、処理の番地を元のプログラム内の所定の戻り先番地に戻すことができる。なお、このソースプログラムでは、ソースコード「EI」及び「RET」のバイト数（容量）は、それぞれ、1バイト及び1バイトとなり、プログラム例29-Aのプログラムの合計バイト数は、2バイトとなる。

【2710】

一方、本実施形態で新たに設けられたリターン命令コードを用いた場合、図386Bに示

50

すように、ソースコード「R E T E」（新命令コード：図344中のNo. I 9）を実行して、まず、割り込みイネーブルフリップフロップをセットしてINT割り込みを有効（割り込み許可）にし、次いで、現在のスタックポインタSPに格納されている値により指示される2バイトのメモリ領域に格納されている値をPCレジスタ（プログラム・カウンタPC）にセットする。この処理により、処理の実行番地が元のプログラム内の所定の戻り先番地に戻り、元のプログラム内の所定の戻り先番地から処理が開始される。なお、ソースコード「R E T E」の処理では、上記処理動作とともに、スタックポインタSPの値が更新される（現在のスタックポインタSPの値から2加算した値（SP+2）がスタックポインタSPに格納される）。

【2711】

本実施形態で新たに設けられたリターン命令コードを用いた場合には、上述した1つのソースコードのみを実行することにより、割り込み許可処理を行った後に、処理の番地を元のプログラム内の所定の戻り先番地に戻すことができる。なお、このソースプログラムでは、ソースコード「R E T E」のバイト数（容量）は、1バイトであり、プログラム例29-Bのプログラムの合計バイト数も、1バイトとなる。

【2712】

上述のように、プログラム例29においても、本実施形態で新たに設けられたリターン命令コードを用いた場合、それを用いない場合に比べて、同じ機能（処理動作）を実現するためのプログラムの容量（バイト数）を減らすことができる。

【2713】

さらに、上述したプログラム例1～プログラム例29の説明（図358～図386）から明らかなように、本実施形態で新たに設けられた各種命令コードを使用した場合、それを用いない場合に比べて、プログラムソースの行数を減らすことができるので、プログラムソースが見易くなる。この結果、プログラムバグの発見が容易になり、プログラム作成に掛かる時間を短縮することができる。

【2714】

〔本実施形態の新命令コードの適用例〕

図72及び図73で説明した上記第1実施形態の通信データ格納処理（サブルーチン「SB__OUTPUT__00」）では、その処理の開始時にソースコード「D I」を実行して割り込み禁止処理を行う（図73参照）。それゆえ、通信データ格納処理（サブルーチン「SB__OUTPUT__00」）の呼び出し先となるプログラム内において、サブルーチン「SB__OUTPUT__00」の呼び出すためのソースコードを、本実施形態で新たに設けたコール命令コード「CALLD」や「CALLFD」を使用して規定すれば、通信データ格納処理（サブルーチン「SB__OUTPUT__00」）の開始時に実行するソースコード「D I」（1バイト）を省略することができる。この場合には、プログラム上において、1バイト分の容量を削減することができる。

【2715】

さらに、図73では明記していないが、上記第1実施形態の通信データ格納処理（サブルーチン「SB__OUTPUT__00」）を終了して呼び出し先（図68の設定変更確認処理又は図70の設定変更コマンド生成格納処理）に処理を戻すための処理を行う前（ソースコード「R E T」（1バイト）を実行する前）には、ソースコード「E I」（1バイト）を行って割り込み許可処理を行う。それゆえ、通信データ格納処理（サブルーチン「SB__OUTPUT__00」）から呼び出し先のプログラムに戻る際のソースコードを、本実施形態で新たに設けたリターン命令コード「R E T E」（1バイト）を使用して規定した場合には、ソースコード「E I」（1バイト）を省略することができる。この場合には、プログラム上において、1バイト分の容量を削減することができる。

【2716】

すなわち、上記第1実施形態の通信データ格納処理（サブルーチン「SB__OUTPUT__00」）の呼び出しに関する処理、及び、通信データ格納処理（サブルーチン「SB__OUTPUT__00」）から呼び出し先への戻りに関する処理に、本実施形態で新たに設けたコー

10

20

30

40

50

ル命令コード「CALLD」や「CALLFD」、及び、リターン命令コード「RETE」を使用した場合には、プログラム上では、合計2バイト分の容量を削減することができる。

【2717】

上述のように、処理の実行中に割込み処理が実行されると実行中の処理に不具合が発生するようなサブルーチンを呼び出す場合、本実施形態で新たに設けたコール命令コード「CALLFD」、「CALLD」及び「RETE」命令は、コール命令コード「CALLF」、「CALL」及び「RET」命令に比べて、容量の面で有効な命令となる。特に、第1実施形態で説明した、規定外RAM初期化処理（図68中のS41参照）、チェックサム生成処理（図77参照）、チェックサム処理（図79及び図80参照）、第2インターフェースボード制御処理（図102参照）、リール停止可能信号OFF処理（図141参照）、リール停止可能信号ON処理（図142参照）、規定外ポート出力処理（図143参照）、及び、試射試験信号制御処理（図166）等の処理、すなわち、規定外ROM領域（図12B参照）に記憶されているプログラムを、遊技用ROM領域に記憶されているプログラムから呼び出す場合に、呼び出し元のプログラム（例えば、メイン処理（図82））が、割込み禁止（「DI」命令）を実行した後、規定外ROM領域に記憶されたプログラム（例えば、メイン処理のS207で呼び出される「第2インターフェースボード制御処理（規定外）」）を呼び出し（「CALL」）、呼び出したプログラムから戻ってきた場合には、呼び出し元のプログラムが、割込み許可（「EI」命令）を実行する必要があるような処理には、本実施形態で新たに設けたコール命令コード「CALLFD」、「CALLD」及び「RETE」命令は、容量低減の観点で有効である。

【2718】

但し、例外として、割込処理（図158参照）から呼び出される試射試験信号制御処理（規定外）（図158のS911参照）等において呼び出される規定外ROM領域に記憶されたプログラムのように、割込み初期設定（KISS：2FDH）が多重割込み禁止に設定されている場合には、割込みコントローラ112から割込要求信号が出力されることがないので、割込み禁止（「DI」命令）を実行する必要はない。それゆえ、この処理では、コール命令コード「CALLFD」及び「CALLD」を使用されない。

【2719】

また、図88に示した上記第1実施形態のメダル投入チェック処理のソースプログラムでは、「状態変化時のチェック」中の処理において、ソースコード「LD B, (HL)」（1バイト）、「LD (HL), A」（1バイト）及び「LD A, B」（1バイト）がこの順で行われ、HLレジスタに格納された値で指定されるアドレス領域に格納されている値と、Aレジスタに格納されている値との交換処理が行われている。それゆえ、本実施形態では、この一連の8ビットデータの交換処理のソースコード（合計3バイト）を、本実施形態で新たに設けた交換命令コード「EX」を使用して、一つのソースコード「EX A, (HL)」（2バイト）で置き換えることができる。そして、この場合には、プログラム上において、1バイト分の容量を削減することができる。

【2720】

また、本実施形態で新たに設けた8ビットロード命令及び16ビットロード命令は、汎用性の高い命令である。それゆえ、上記第1実施形態及び第2実施形態で説明した図88のメダル投入チェック処理以外の各種フローチャートの各種制御プログラムにおいても、適宜、新たに設けた8ビットロード命令及び16ビットロード命令に置き換えることができる。

【2721】

さらに、図278に代表されるような、複合データが記憶されているテーブルデータからビットデータを取得して処理を行う場合などにおいて、本実施形態で新たに設けたシフト命令（「SLBIT A, n」～「SRBIT Q (k), n」）を使用した場合には、上記第1及び第2実施形態の遊技機でのみ使用されている命令コードを使用するよりも複合データ内からの必要なビットデータの抽出が容易になる。

【2722】

〔本実施形態で新たに設けられた各種命令コードによる各種効果〕

本実施形態の遊技機では、上述のように、新たに各種命令コードを追加したので、機能面において、更なる改良が施されたマイクロプロセッサを搭載した遊技機を提供することができる。また、本実施形態では、新たな各種命令コードが設けられているので、各種機能をさらに拡張することも可能なる。

【2723】

また、近年、遊技機業界に特有の理由から、AT遊技の抽籤に係るテーブル及びプログラムも主制御基板に設けられたROMに記憶する必要がある。このため、規則により小容量に制限されている主制御基板のROM容量が圧迫されることになる。それゆえ、従来、主制御基板側（メイン側）に設けられたROMの空き容量を増やし、ROM容量が圧迫されることを抑制することが求められている。

【2724】

それに対して、本実施形態の遊技機では、上記プログラム例1～29で説明したように、新たに設けられた各種命令コードを用いた場合、上記第1及び第2実施形態の遊技機で使用されている命令コードを用いた場合に比べて、プログラムの容量（バイト数）を減らすことができる。それゆえ、本実施形態では、上記要望に応じて、主制御基板側（メイン側）に設けられたROMの空き容量を増やし、ROM容量が圧迫されることを抑制することができる。また、この場合、増えた容量分のROMの空き領域を利用して遊技性を高める（拡張する）ことも可能なる。

【2725】

〔その他の第3実施形態に係る遊技機の拡張性〕

本実施形態のパチスロ1では、スタートレバー16の押下からリール3L、3C、3Rの変動が定速回転に至るまでの間（加速中：遊技ロック（図286及び図292参照）中）に遊技媒体の払出しを伴わない、リール3L、3C、3Rを遊技者の停止操作によって一時的に停止させたように見せる演出（以降「疑似遊技」という）を行うことができる。疑似遊技を開始すると、リール3L、3C、3Rを停止させるためのストップボタン17L、17C、17Rが押下可能になる。遊技者が、疑似遊技中に、ストップボタン17L、17C、17Rを押下すると、ストップボタンの押下に応じて、リール3L、3C、3Rいずれかを疑似停止させ、リール3L、3C、3Rの全てが疑似停止した後に、リール3L、3C、3Rは、ランダム遅延を行い、通常のリール制御に戻る。なお、疑似停止とは、リール3L、3C、3Rが完全に停止した状態ではなく、リール3L、3C、3Rが微振動を行う状態であり、微振動とは、リール3L、3C、3Rが順回転と逆回転を100msec～500msecの間隔で交互に行われている状態を示す。なお、メインCPU101がリール3L、3C、3Rを振動するように制御を行っていても遊技者が認識できない振動であれば疑似停止とは認められない。ランダム遅延とは、リール3L、3C、3Rにそれぞれ遅延時間を設け、遅延後リールの回転を再開させる。また、遅延時間は、約100msec～約3秒程であり、各遅延時間は乱数に基づいて決定される。また、疑似遊技中に遊技者に疑似遊技中が認識可能になるように疑似遊技中ランプを設けてもよい。さらに、疑似遊技自体を複数回の渡って行ってもよい。

【2726】

<付記>

〔第1～第9の遊技機〕

従来、リセット入力からプログラムの起動までの時間をランダム化させる遊技機が開示されている（例えば、特開2009-034162号公報参照）。

【2727】

上述した特開2009-034162号公報で開示されている遊技機では、電源投入時の不正行為を有効に阻止することができるが、マイクロプロセッサに関しては、他の機能についても更なる改良が求められる。

【2728】

本発明は上記課題を鑑みなされたものであり、本発明の目的は、各種機能に対して更なる改良が施されたマイクロプロセッサを搭載した遊技機を提供することである。

【2729】

上記課題を解決するために、本発明では、以下のような構成の第1の遊技機を提供する。

【2730】

演算制御部（例えば、メインCPU101）、第1記憶部（例えば、メインROM102）及び第2記憶部（例えば、メインRAM103）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記第1記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、交換命令（例えば、「EX r、(ss)」等）が含まれ、

前記交換命令では、オペランドとして、命令（例えば、「EX」）、第1交換先を示す情報（例えば、「r」）、及び、第2交換先を示す情報（例えば、「(ss)」）がこの順で配置され、

前記演算制御部により前記交換命令が実行されると、前記第1交換先に格納されているデータと、前記第2交換先に格納されているデータとが交換される

ことを特徴とする遊技機。

【2731】

また、前記本発明の第1の遊技機では、前記演算制御部は、1バイトで構成されたレジスタ（例えば、汎用レジスタ）を有し、

前記第1記憶部及び前記第2記憶部のそれぞれのアドレス空間は、16ビットの値で表され、

前記第1交換先を示す情報に前記レジスタの情報がセットされ、前記第2交換先を示す情報に前記第2記憶部内の所定のアドレスを指定可能な情報がセットされた状態で、前記演算制御部により前記交換命令が実行された場合には、前記レジスタに格納されているデータと、前記第2記憶部内の前記所定のアドレスの領域に格納されているデータとが交換されるようにしてもよい。

【2732】

また、前記本発明の第1の遊技機では、前記マイクロプロセッサは、さらに、複数の回路、及び、該複数の回路のそれぞれを制御するための複数種の制御レジスタを内蔵し、

前記複数種の制御レジスタのそれぞれには、16ビットの値で表されたアドレスが割り当てられ、

前記複数種の制御レジスタには、読み出し専用制御レジスタ及び読書き用制御レジスタが含まれ、

前記第1交換先及び前記第2交換先の一方向の交換先を示す情報に前記読み出し専用制御レジスタのアドレスがセットされている状態で前記演算制御部より前記交換命令が実行された場合には、前記読み出し専用制御レジスタのデータは、前記第1交換先及び前記第2交換先の方の交換先に格納されるが、前記他方の交換先のデータは前記読み出し専用制御レジスタに格納されず、

前記一方の交換先に示す情報に前記読書き用制御レジスタのアドレスがセットされている状態で前記演算制御部より前記交換命令が実行された場合には、前記読書き用制御レジスタのデータは前記他方の交換先に格納され、前記他方の交換先のデータは、前記読書き用制御レジスタに格納されるようにしてもよい。

【2733】

また、上記課題を解決するために、本発明では、以下のような構成の第2の遊技機を提供する。

【2734】

演算制御部（例えば、メインCPU101）、第1記憶部（例えば、メインROM102）及び第2記憶部（例えば、メインRAM103）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、命令の実行結果が反映されるフラグレジスタ（例えば、フラグ・レジスタF）を有し、

前記第1記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定の読み出し命令（例えば、「LDZ (mn)」等）が含まれ、

前記フラグレジスタには、命令の実行結果に応じて状態が変化する所定のフラグ（例えば、TZフラグ）が含まれ、

前記特定の読み出し命令では、オペランドとして、命令（例えば、「LDZ」）、及び、格納先を示す情報（例えば、「(mn)」）がこの順で配置され、

前記演算制御部により前記特定の読み出し命令が実行されると、前記所定のフラグの状態に応じた値（例えば、「0」又は「1」）が前記格納先にセットされる

ことを特徴とする遊技機。

【2735】

また、前記本発明の第2の遊技機では、前記フラグレジスタには、前記所定のフラグ以外にも、命令の実行結果に応じて状態が変化する複数種のフラグが含まれ、

前記演算制御部による前記特定の読み出し命令の実行時には、前記フラグレジスタ内の前記所定のフラグ以外のフラグの状態は変化しないようにしてもよい。

【2736】

また、上記課題を解決するために、本発明では、以下のような構成の第3の遊技機を提供する。

【2737】

演算制御部（例えば、メインCPU101）、第1記憶部（例えば、メインROM102）及び第2記憶部（例えば、メインRAM103）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、

命令の実行結果が反映されるフラグレジスタ（例えば、フラグ・レジスタF）と、

前記第2記憶部内のアドレスの一部を指定可能な拡張レジスタ（例えば、Qレジスタ）と、を有し、

前記フラグレジスタには、命令の実行による桁上りの発生の有無に応じて状態が変化するキャリーフラグ（例えば、C(CY)フラグ）が含まれ、

前記第1記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定の読み出し命令（例えば、「LDQC (k)」等）が含まれ、

前記特定の読み出し命令では、オペランドとして、命令（例えば、「LDQC」）、及び、格納先の一部を示す情報（例えば、「(k)」）がこの順で配置され、

前記演算制御部により前記キャリーフラグの状態を変化させることが可能な命令が実行された後に、前記特定の読み出し命令が実行されると、前記キャリーフラグの状態に応じた値（例えば、「0」又は「1」）が、前記拡張レジスタに格納された値と前記格納先の一部を示す情報とで指定される前記格納先にセットされる

ことを特徴とする遊技機。

【2738】

また、前記本発明の第3の遊技機では、前記フラグレジスタには、前記キャリーフラグと、命令の実行結果が0であるか否かに応じて状態が変化する第1のゼロフラグ（例えば、Zフラグ）及び第2のゼロフラグ（例えば、TZフラグ）と、その他の複数種のフラグとが含まれ、

前記演算制御部による前記特定の読み出し命令の実行時には、前記フラグレジスタ内の前記第2のゼロフラグ以外のフラグの状態は変化しないようにしてもよい。

【2739】

10

20

30

40

50

また、上記課題を解決するために、本発明では、以下のような構成の第4の遊技機を提供する。

【2740】

演算制御部（例えば、メインCPU101）、第1記憶部（例えば、メインROM102）及び第2記憶部（例えば、メインRAM103）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、

1バイトで構成されたレジスタ（例えば、Aレジスタ（アキュムレータ））と、

2バイトで構成され、実行すべき処理コードのアドレスが保存されるプログラムカウンタ（例えば、PCレジスタ）と、を有し、

前記第1記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定のジャンプ命令（例えば、「TJP」、「TJPW」等）が含まれ、

前記特定のジャンプ命令では、オペランドとして、命令（例えば、「TJP」、「TJPW」）が配置され、

前記第1記憶部には、

前記特定のジャンプ命令、及び、ジャンプ先アドレスを指定可能なデータが規定された第1プログラムと、

前記第1プログラムからジャンプして処理を行うための複数の第2プログラムと、が記憶され、

前記演算制御部により前記特定のジャンプ命令が実行されると、前記プログラムカウンタに格納されている値と前記レジスタに格納されている値とに基づいて前記ジャンプ先アドレスを指定可能なデータが取得され、当該取得された前記ジャンプ先アドレスを指定可能なデータが前記プログラムカウンタにセットされる

ことを特徴とする遊技機。

【2741】

また、前記本発明の第4の遊技機では、前記ジャンプ先アドレスを指定可能なデータは、1バイトで構成され、前記第2プログラムが記憶された2バイトで表されるアドレス値の下位側1バイトのアドレス値であり、

前記特定のジャンプ命令が実行された場合には、取得された1バイトの前記ジャンプ先アドレスを指定可能なデータを前記プログラムカウンタの下位側1バイトの領域にセットするようにしてもよい。

【2742】

また、前記本発明の第4の遊技機では、前記ジャンプ先アドレスを指定可能なデータは、2バイトで構成され、前記第2プログラムが記憶された2バイトのアドレス値であり、

前記特定のジャンプ命令が実行された場合には、取得された2バイトの前記ジャンプ先アドレスを指定可能なデータを前記プログラムカウンタにセットするようにしてもよい。

【2743】

また、前記本発明の第4の遊技機では、前記特定のジャンプ命令が実行された場合に使用される前記レジスタは、アキュムレータであるようにしてもよい。

【2744】

また、前記本発明の第4の遊技機では、前記特定のジャンプ命令が実行された場合に使用される前記レジスタは、アキュムレータ以外の1バイトの汎用レジスタであるようにしてもよい。

【2745】

また、上記課題を解決するために、本発明では、以下のような構成の第5の遊技機を提供する。

【2746】

演算制御部（例えば、メインCPU101）、第1記憶部（例えば、メインROM102

10

20

30

40

50

）及び第２記憶部（例えば、メインＲＡＭ１０３）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、１バイトで構成されたレジスタ（例えば、汎用レジスタ）を有し、

前記第１記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定の比較読み出し命令（例えば、「ＩＣＰＬＬ Ａ，ｒ」、「ＤＣＰＬＬ Ａ，ｒ」等）が含まれ、

前記特定の比較読み出し命令では、オペランドとして、命令（例えば、「ＩＣＰＬＬ」、「ＤＣＰＬＬ」）、前記レジスタを示す情報、及び、閾値（例えば、上限値、下限値）に関する情報（例えば、直値ｎ又はレジスタｒ）がこの順で配置され、

前記演算制御部により前記特定の比較読み出し命令が実行されると、前記レジスタに格納されている値が前記閾値と比較され、当該比較結果に応じて、前記レジスタに格納されている値に対して特定の演算（例えば、１加算、１減算）が行われ、当該演算の結果が前記レジスタに格納される、又は、前記閾値が前記レジスタに格納される

ことを特徴とする遊技機。

【２７４７】

また、前記本発明の第５の遊技機では、前記特定の比較読み出し命令の実行により行われる前記特定の演算は、前記レジスタに格納されている値に１を加算する演算であり、

前記特定の比較読み出し命令に実行により行われる前記レジスタに格納されている値と前記閾値との比較において、前記閾値が前記レジスタに格納されている値より大きいという結果が得られた場合には、前記特定の演算の結果が前記レジスタに格納され、前記閾値が前記レジスタに格納された値より大きくないという結果が得られた場合には、前記閾値が前記レジスタに格納されるようにしてもよい。

【２７４８】

また、前記本発明の第５の遊技機では、前記特定の比較読み出し命令の実行により行われる前記特定の演算は、前記レジスタに格納されている値を１減算する演算であり、

前記特定の比較読み出し命令に実行により行われる前記レジスタに格納されている値と前記閾値との比較において、前記閾値が前記レジスタに格納されている値より小さいという結果が得られた場合には、前記特定の演算の結果が前記レジスタに格納され、前記閾値が前記レジスタに格納された値より小さくないという結果が得られた場合には、前記閾値が前記レジスタに格納されるようにしてもよい。

【２７４９】

また、上記課題を解決するために、本発明では、以下のような構成の第６の遊技機を提供する。

【２７５０】

演算制御部（例えば、メインＣＰＵ１０１）、第１記憶部（例えば、メインＲＯＭ１０２）及び第２記憶部（例えば、メインＲＡＭ１０３）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、１バイトで構成されたレジスタ（例えば、汎用レジスタ）を有し、

前記第１記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定のシフト命令（例えば、「ＳＬＢＩＴ Ａ，ｎ」、「ＳＲＢＩＴ Ａ，ｎ」等）が含まれ、

前記特定のシフト命令では、オペランドとして、命令（例えば、「ＳＬＢＩＴ」、「ＳＲＢＩＴ」）、前記レジスタを示す情報、及び、前記レジスタ内の所定のビット位置（例えば、指定ビット位置ｎ）がこの順で配置され、

前記演算制御部により前記特定のシフト命令が実行されると、前記レジスタ内の前記所定のビット位置から前記命令が示す方向の最端ビット位置までの領域に格納された値が、前記命令が示す方向に１ビット分シフトする

ことを特徴とする遊技機。

【2751】

また、前記本発明の第6の遊技機では、前記特定のシフト命令が示す方向が上位方向である場合には、前記特定のシフト命令の実行により、前記レジスタ内の前記所定のビット位置から最上位のビット位置までの領域に格納された値が上位方向に1ビット分シフトするが、前記所定のビット位置より下位側に位置するビット位置の領域に格納された値はシフトしないようにしてもよい。

【2752】

また、前記本発明の第6の遊技機では、前記特定のシフト命令が示す方向が下位方向である場合には、前記特定のシフト命令の実行により、前記レジスタ内の前記所定のビット位置から最下位のビット位置までの領域に格納された値が下位方向に1ビット分シフトするが、前記所定のビット位置より上位側に位置するビット位置の領域に格納された値はシフトしないようにしてもよい。

【2753】

また、上記課題を解決するために、本発明では、以下のような構成の第7の遊技機を提供する。

【2754】

演算制御部（例えば、メインCPU101）、第1記憶部（例えば、メインROM102）及び第2記憶部（例えば、メインRAM103）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、

前記第2記憶部に配置されたスタック領域のアドレスが保存されるスタックポインタ（例えば、スタックポインタSP）と、

実行すべき処理コードのアドレスが保存されるプログラムカウンタ（例えば、PCレジスタ）と、を有し、

前記第1記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定の呼び出し命令（例えば、「CALLD mn」、「CALLLE mn」等）が含まれ、

前記特定の呼び出し命令では、オペランドとして、命令（例えば、「CALLD」、「CALLLE」等）、及び、呼び出し先に関する情報（例えば、「mn」）がこの順で配置され、

前記演算制御部により前記特定の呼び出し命令が実行されると、割込み制御が行われ、処理コードの戻り先のアドレスが前記スタックポインタを使用して前記スタック領域に格納され、且つ、前記呼び出し先のアドレスが前記プログラムカウンタにセットされる

ことを特徴とする遊技機。

【2755】

また、前記本発明の第7の遊技機では、前記特定の呼び出し命令の実行により行われる前記割込み制御は、割込み処理を禁止することであるようにしてもよい。

【2756】

また、前記本発明の第7の遊技機では、前記特定の呼び出し命令の実行により行われる前記割込み制御は、割込み処理を許可することであるようにしてもよい。

【2757】

また、上記課題を解決するために、本発明では、以下のような構成の第8の遊技機を提供する。

【2758】

演算制御部（例えば、メインCPU101）、第1記憶部（例えば、メインROM102）及び第2記憶部（例えば、メインRAM103）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、

前記第2記憶部に配置されたスタック領域のアドレスが保存されるスタックポインタと、

実行すべき処理コードのアドレスが保存されるプログラムカウンタと、を有し、
前記第 1 記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、
前記複数種の命令には、特定のリターン命令（例えば、「RET D」、「RET E」）が含まれ、
前記特定のリターン命令では、オペランドとして、命令（例えば、「RET D」、「RET E」）が配置され、
前記演算制御部により前記特定のリターン命令が実行されると、割込み制御が行われ、処理コードの戻り先のアドレスが前記スタックポインタを使用して前記スタック領域から取得され、当該取得された前記戻り先のアドレスが前記プログラムカウンタにセットされることを特徴とする遊技機。

10

【2759】

また、前記本発明の第 8 の遊技機では、前記特定のリターン命令の実行により行われる前記割込み制御は、割込み処理を禁止することであるようにしてもよい。

【2760】

また、前記本発明の第 8 の遊技機では、前記特定のリターン命令の実行により行われる前記割込み制御は、割込み処理を許可することであるようにしてもよい。

【2761】

また、上記課題を解決するために、本発明では、以下のような構成の第 9 の遊技機を提供する。

20

【2762】

演算制御部（例えば、メイン CPU 101）、第 1 記憶部（例えば、メイン ROM 102）及び第 2 記憶部（例えば、メイン RAM 103）を内蔵したマイクロプロセッサを備えた遊技機であって、

前記演算制御部は、命令の実行結果が反映されるフラグレジスタ（例えば、フラグ・レジスタ F）を有し、

前記第 1 記憶部は、前記演算制御部による制御において使用可能な、複数種の命令及び複数のデータで構成されたプログラムを記憶可能であり、

前記複数種の命令には、特定の読み出し命令（例えば、「LDF Z, A, n」、「LDF Z, BC, mn」等）が含まれ、

30

前記特定の読み出し命令では、オペランドとして、命令（例えば、「LDF」）、前記フラグレジスタ内の指定フラグ（例えば、Z フラグ、C フラグ）の情報、格納先を示す情報（例えば、「A」、「BC」等）、及び、格納データ（例えば、「n」、「mn」）がこの順で配置され、

前記演算制御部により、前記特定の読み出し命令が実行されると、前記指定フラグの状態に応じて、前記格納データが前記格納先に格納されることを特徴とする遊技機。

【2763】

また、前記本発明の第 9 の遊技機では、前記演算制御部は、1 バイトで構成されるレジスタ（例えば、汎用レジスタ）を有し、

40

前記格納先は、前記レジスタであり、

前記格納データは、1 バイトデータであるようにしてもよい。

【2764】

また、前記本発明の第 9 の遊技機では、前記演算制御部は、1 バイト単位の 2 つのレジスタで構成され、2 バイトのデータを扱うことが可能なペアレジスタ（例えば、BC, DE, HL レジスタ）を有し、

前記格納先は、前記ペアレジスタであり、

前記格納データは、2 バイトデータであるようにしてもよい。

【2765】

また、前記本発明の第 9 の遊技機では、前記フラグレジスタには、命令の実行結果が 0 で

50

あるか否かに応じて状態が変化するゼロフラグ、及び、命令の実行による桁上がりの発生の有無に応じて状態が変化するキャリーフラグが含まれ、

前記指定フラグは、前記ゼロフラグ又は前記キャリーフラグであるようにしてもよい。

【2766】

上記構成の本発明の第1～第9の遊技機によれば、各種機能に対して更なる改良が施されたマイクロプロセッサを搭載した遊技機を提供することができる。

【符号の説明】

【2767】

1…パチスロ、3L, 3C, 3R…リール、4…リール表示窓、6…情報表示器、11…表示装置、17L, 17C, 17R…ストップボタン、18…サブ表示装置、71…主制御基板、72…副制御基板、90…主制御回路、91…マイクロプロセッサ、101…メインCPU、102…メインROM、103…メインRAM、107…演算回路、114…第1シリアル通信回路、115…第2シリアル通信回路、200…副制御回路、201…サブCPU 201、301…第1インターフェースボード、302…第2インターフェースボード

10

20

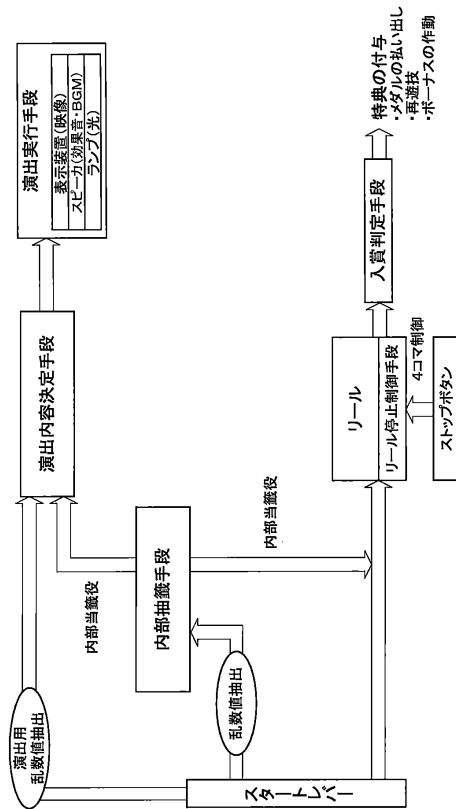
30

40

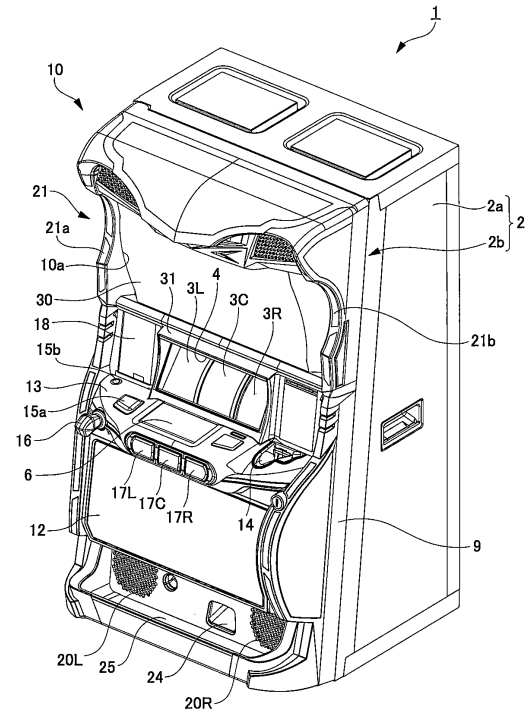
50

【図面】

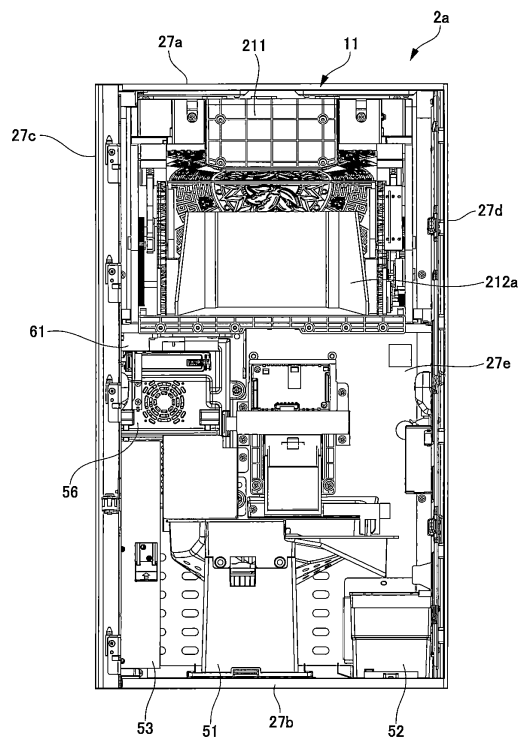
【図 1】



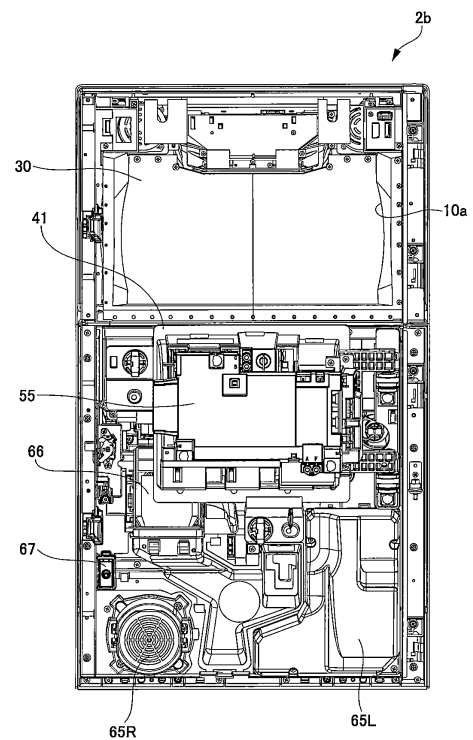
【図 2】



【図 3】



【図 4】



10

20

30

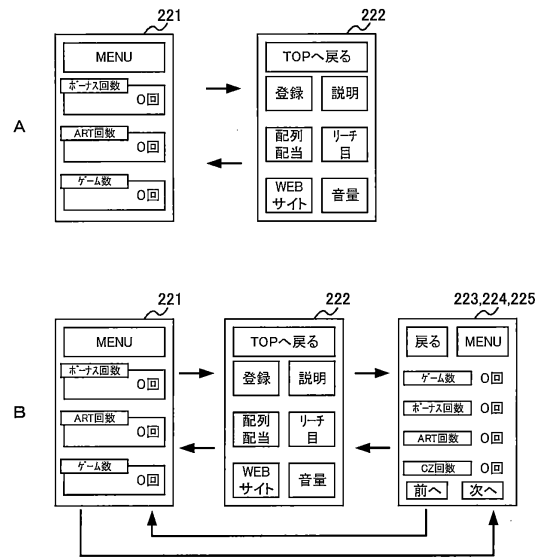
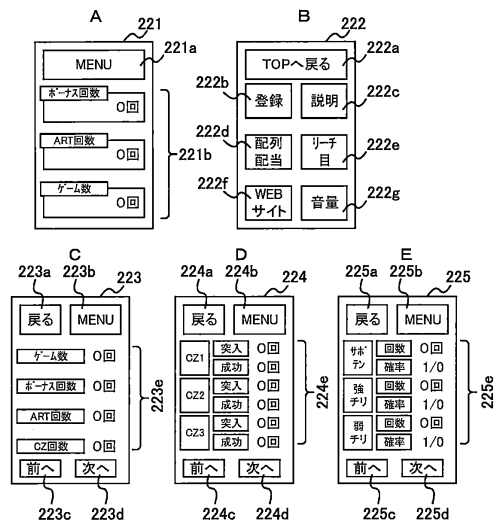
40

50

【図 5】

【図 6】

サブ表示装置の表示画面

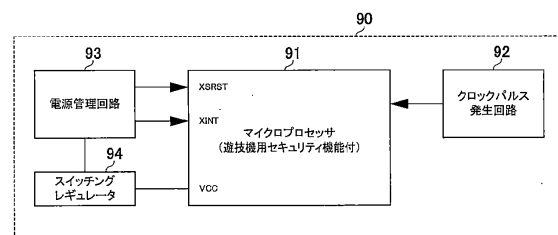
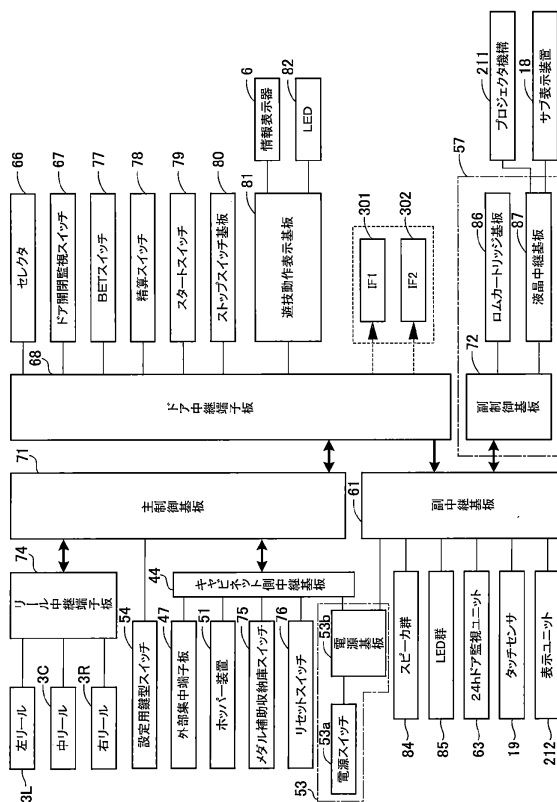


10

20

【図 7】

【圖 8】

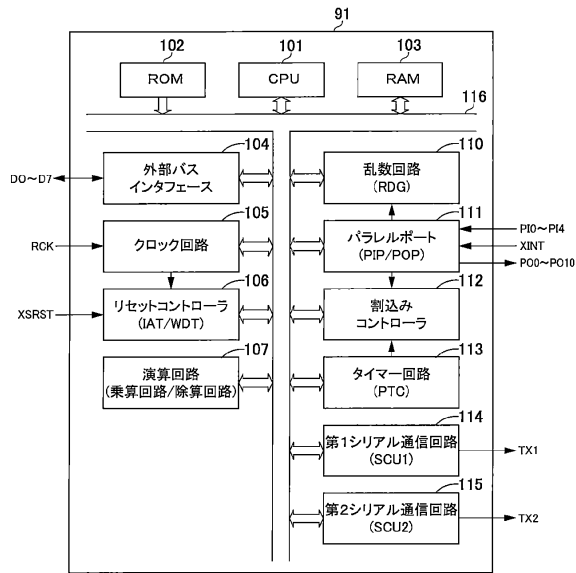


30

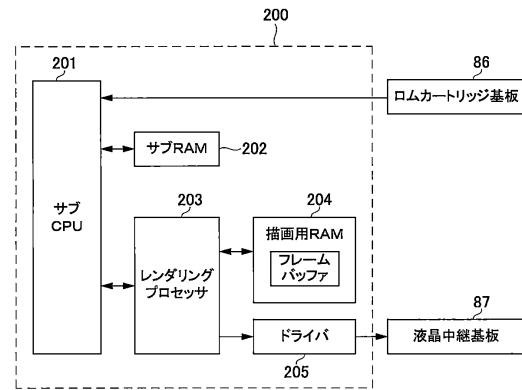
40

50

【圖 9】



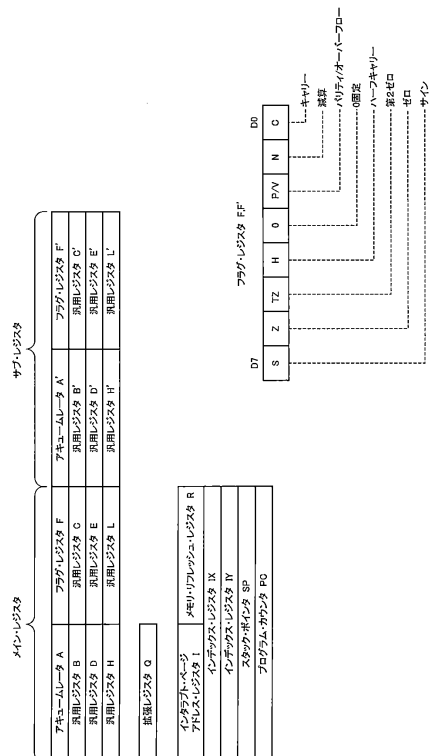
【図 10】



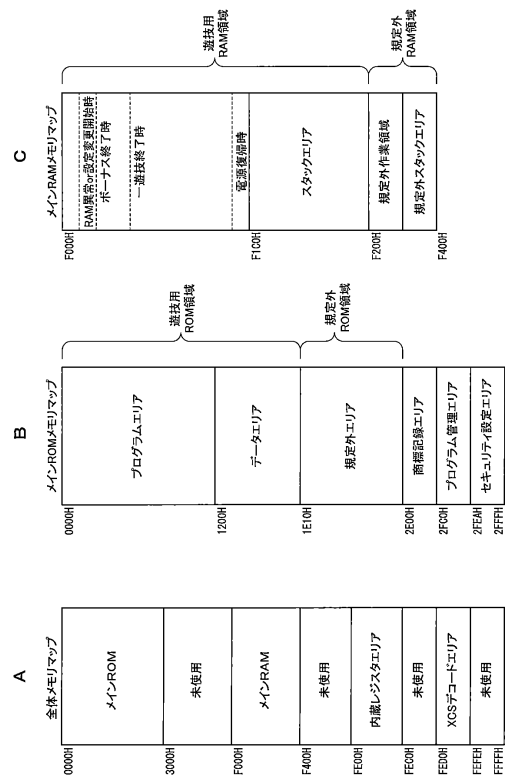
10

20

【図 11】



【図 12】

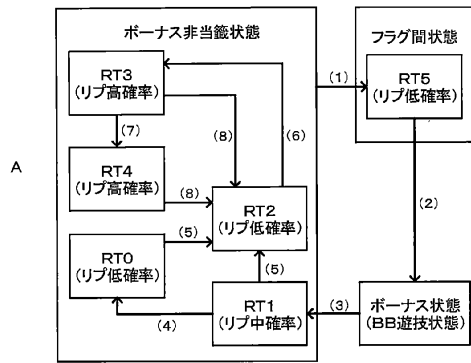


30

40

50

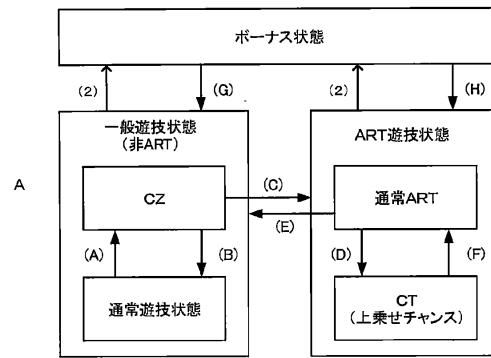
【図 13】



B

移行条件	内容
(1)	F.BB1.2に内部当籤
(2)	F.BB1.2入賞(C.BB1.2が表示)
(3)	ボーナス終了(216枚を超える払出)
(4)	20ゲーム経過
(5)	ベルにぼし目表示
(6)	RT3移行リプレイ表示
(7)	RT4移行リプレイ表示
(8)	ベルにぼし目又はRT12移行リプレイ表示

【図 14】



B

移行条件	内容
(A)	通常遊技状態中のCZ抽籤に当籤
(B)	CZ中のART抽籤に非当籤
(C)	CZ中のART抽籤に当籤
(D)	通常ART中のCT抽籤に当籤
(E)	通常ART終了
(F)	CT終了
(G)	ボーナス状態中のART抽籤に非当籤
(H)	ボーナス状態中のART抽籤に当籤

【図 15】

図柄配置テーブル

左リール		中リール		右リール	
図柄位置	図柄	図柄位置	図柄	図柄位置	図柄
19	サボテン3	19	サボテン3	19	サボテン1
18	リプレイ	18	リプレイ	18	チリ上2
17	サボテン3	17	帽子	17	青7
16	帽子	16	チリ上1	16	帽子
15	リプレイ	15	チリ下	15	リプレイ
14	チリ上1	14	サボテン3	14	サボテン3
13	チリ下	13	リプレイ	13	チリ上1
12	青7	12	帽子	12	チリ下
11	帽子	11	白7	11	帽子
10	リプレイ	10	リプレイ	10	リプレイ
9	サボテン1	9	チリ上2	9	サボテン3
8	帽子	8	帽子	8	チリ上1
7	サボテン2	7	チリ上1	7	青7
6	帽子	6	リプレイ	6	帽子
5	リプレイ	5	帽子	5	リプレイ
4	サボテン3	4	青7	4	サボテン1
3	白7	3	チリ上1	3	チリ上2
2	白7	2	白7	2	白7
1	帽子	1	リプレイ	1	帽子
0	リプレイ	0	帽子	0	リプレイ

図柄コード表

図柄コード	内容	
	図柄	データ
1	白7	00000001
2	青7	00000010
3	チリ上1	00000011
4	チリ上2	00000100
5	チリ下	00000101
6	リプレイ	00000110
7	帽子	00000111
8	サボテン1	00001000
9	サボテン2	00001001
10	サボテン3	00001010

【図 16】

内部抽籤テーブル(設定1)

No.	略称	RT0	RT1	RT2	RT3	RT4
1	はずれ	37779	8549	37849	1789	1299
2	F.BB1	20	20	20	20	20
3	F.BB1+F.リーチ目リブA	5	5	5	5	5
4	F.BB1+F.リーチ目リブB	1	1	1	1	1
5	F.BB1+F.リーチ目リブC	5	5	5	5	5
6	F.BB1+F.リーチ目リブD	1	1	1	1	1
7	F.BB1+F.サボ1	15	15	15	15	15
8	F.BB1+F.サボ2	10	10	10	10	10
9	F.BB1+F.強サリ	5	5	5	5	5
10	F.BB1+F.強サリ1	15	15	15	15	15
11	F.BB1+F.強サリ2	15	15	15	15	15
12	F.BB1+F.特殊1	5	5	5	5	5
13	F.BB1+F.特殊3	5	5	5	5	5
14	F.BB2	20	20	20	20	20
15	F.BB2+F.リーチ目リブA	5	5	5	5	5
16	F.BB2+F.リーチ目リブB	1	1	1	1	1
17	F.BB2+F.リーチ目リブC	5	5	5	5	5
18	F.BB2+F.リーチ目リブD	1	1	1	1	1
19	F.BB2+F.サボ1	15	15	15	15	15
20	F.BB2+F.サボ2	10	10	10	10	10
21	F.BB2+F.強サリ	5	5	5	5	5
22	F.BB2+F.強サリ1	15	15	15	15	15
23	F.BB2+F.強サリ2	15	15	15	15	15
24	F.BB2+F.特殊2	5	5	5	5	5
25	F.リブ	0	0	0	0	5500
26	F.維持リブ	0	0	0	0	13400
27	F.1維持リブ	0	0	0	0	3300
28	F.リーチ目リブA	0	0	0	0	1200
29	F.リーチ目リブB	0	0	0	0	500
30	F.リーチ目リブC	0	0	0	0	600
31	F.リーチ目リブD	0	0	0	0	500
32	F.維持リブA	8900	37800	1300	0	11800
33	F.維持リブB	0	330	330	0	330
34	F.維持リブ1st	0	0	0	11230	2750
35	F.維持リブ2nd	0	0	0	11230	2750
36	F.維持リブ3rd	0	0	0	11230	2750
37	F.RT3リブ1st	0	0	1800	0	0
38	F.RT3リブ213	0	0	1800	0	0
39	F.RT3リブ231	0	0	1800	0	0
40	F.RT3リブ3rd	0	0	1800	0	0
41	F.RT4リブ123	0	0	0	2800	0
42	F.RT4リブ132	0	0	0	2800	0
43	F.RT4リブ2nd	0	0	0	2800	0
44	F.RT4リブ3rd	0	0	0	2800	0
45	F.3採ヘル1st	5250	5250	5250	5250	5250
46	F.3採ヘル2nd	5250	5250	5250	5250	5250
47	F.3採ヘル3rd	5250	5250	5250	5250	5250
48	F.共通ヘル	1000	1000	1000	1000	1000
49	F.サボ1	610	610	610	610	610
50	F.サボ2	128	128	128	128	128
51	F.弱リ	870	870	870	870	870
52	F.強リ1	150	150	150	150	150
53	F.強リ2	150	150	150	150	150

【図 17】

内部拾音テーブル(設定1)				内部拾音テーブル(設定2)			
No.	略称	RT5 (BB1)		No.	略称	RT5 (BB2)	
1	はずり(F BB1)	29923		1	はずり(F BB2)	29923	
2	F BB1+F リア7p	0		2	F BB2+F リア7p	0	
3	F BB1+F 隣リ7p	0		3	F BB2+F 隣リ7p	0	
4	F BB1+F 1隣リ7p	0		4	F BB2+F 1隣リ7p	0	
5	F BB1+F リア3p7p/A	4100		5	F BB2+F リア3p7p/A	4100	
6	F BB1+F リア3p7p/B	1650		6	F BB2+F リア3p7p/B	1650	
7	F BB1+F リア7p7p/D	2050		7	F BB2+F リア7p7p/D	2050	
8	F BB1+F リア7p7p/D	500		8	F BB2+F リア7p7p/D	500	
9	F BB1+F 維持リ7p/A	5080		9	F BB2+F 維持リ7p/A	5080	
10	F BB1+F 維持リ7p/B	0		10	F BB2+F 維持リ7p/B	0	
11	F BB1+F 維持リ7p 1st	1640		11	F BB2+F 維持リ7p 1st	1640	
12	F BB1+F 維持リ7p 2nd	1640		12	F BB2+F 維持リ7p 2nd	1640	
13	F BB1+F 維持リ7p 3rd	1640		13	F BB2+F 維持リ7p 3rd	1640	
14	F BB1+F RT3p7p 1st	410		14	F BB2+F RT3p7p 1st	410	
15	F BB1+F RT3p7p 213	410		15	F BB2+F RT3p7p 213	410	
16	F BB1+F RT3p7p 231	410		16	F BB2+F RT3p7p 231	410	
17	F BB1+F RT4p7p 3rd	410		17	F BB2+F RT4p7p 3rd	410	
18	F BB1+F RT4p7p 123	0		18	F BB2+F RT4p7p 123	0	
19	F BB1+F RT4p7p 132	0		19	F BB2+F RT4p7p 132	0	
20	F BB1+F RT4p7p 2nd	0		20	F BB2+F RT4p7p 2nd	0	
21	F BB1+F RT4p7p 3rd	0		21	F BB2+F RT4p7p 3rd	0	
22	F BB1+F 3折ベル 1st	5250		22	F BB2+F 3折ベル 1st	5250	
23	F BB1+F 3折ベル 2nd	5250		23	F BB2+F 3折ベル 2nd	5250	
24	F BB1+F 3折ベル 3rd	5250		24	F BB2+F 3折ベル 3rd	5250	
25	F BB1+F 共通ベル	1000		25	F BB2+F 共通ベル	1000	
26	F BB1+F 音#1	610		26	F BB2+F 音#1	610	
27	F BB1+F 音#2	128		27	F BB2+F 音#2	128	
28	F BB1+F 弱リ	870		28	F BB2+F 弱リ	870	
29	F BB1+F 強リ1	150		29	F BB2+F 強リ1	150	
30	F BB1+F 強リ2	150		30	F BB2+F 強リ2	150	
31	F BB1+F 特殊1	5		31	F BB2+F 特殊1	5	
32	F BB1+F 特殊2	5		32	F BB2+F 特殊2	5	
33	F BB1+F 特殊3	5		33	F BB2+F 特殊3	5	

【圖 18】

[illegible]

【圖 19】

[illegible]

【圖 20】

[illegible]

【図 2 1】

[illegible]

【圖 22】

[illegible]

10

20

【圖 23】

当番役	図柄組合せ							
	C	C	C	C	C	C	C	C
	強 2枚C 0-03	強 2枚C 0-04	強 2枚C 0-05	強 2枚C 0-06	強 2枚C 0-07	強 2枚C 0-08	強 2枚C 0-09	強 2枚C 0-10
F_BB1								
F_BB2								
F_チリウ'								
F_嬢リリウ'								
F_1嬢リリウ'								
F_リ-チ目リウ'A								
F_リ-チ目リウ'B								
F_リ-チ目リウ'C								
F_リ-チ目リウ'D								
F_嬢持リウ'A								
F_嬢持リウ'B								
F_嬢持リウ'1st								
F_嬢持リウ'2nd								
F_嬢持リウ'3rd								
F_RT31ウ'1st								
F_RT31ウ'213								
F_RT31ウ'231								
F_RT31ウ'3rd								
F_RT41ウ'123								
F_RT41ウ'132								
F_RT41ウ'2nd								
F_RT41ウ'3rd								
F_3枚△△1st								
F_3枚△△2nd								
F_3枚△△3rd								
F_共通△△								
F_サキ1								
F_サキ2								
F_弱リ								
F_強リ1	○	○	○	○	○	○	○	○
F_強リ2	○	○	○	○	○	○	○	○
F_特殊1								
F_特殊2								
F_特殊3								
F_RB役1								
F_RB役2								
F_RB役3								
F_RB役4	○	○	○	○	○	○	○	○

【圖 24】

内部当籤役と停止図柄組合せ(略称)との対応関係		
当籤役	停止操作	
	押し順 正解	押し順 不正解
F.チリワ	チリリプ(否3連)	リプリイ
F.確リツリ	チリリプ(3連)	リプリイ
F.1確リツリ	チリリプ(3連)	リプリイ
F.リーチ目リ	リーチ目リ	リプリイ
F.リーチ目リ	リーチ目リ	リプリイ
F.リーチ目リ	リーチ目リ	リプリイ
F.リーチ目リ	リーチ目リ	リプリイ
F.維持リツA		リプレイ
F.維持リツB		リプリイ
F.維持リツ1st	リプレイ	RT2移行リ
F.維持リツ2nd	リプレイ	RT2移行リ
F.維持リツ3rd	リプレイ	RT2移行リ
F.RT3リツ1st	RT3移行リ	リプリイ
F.RT3リツ213	RT3移行リ	リプリイ
F.RT3リツ231	RT3移行リ	リプリイ
F.RT3リツ3rd	RT3移行リ	リプリイ
F.RT4リツ123	RT4移行リ	リプリイ
F.RT4リツ132	RT4移行リ	リプリイ
F.RT4リツ2nd	RT4移行リ	リプリイ
F.RT4リツ3rd	RT4移行リ	リプリイ
F.3択ベル1st	ベル	ベルにほし目 or 1枚出目
F.3択ベル2nd	ベル	ベルにほし目 or 1枚出目
F.3択ベル3rd	ベル	ベルにほし目 or 1枚出目
F.共通ベル		ベル
F.サホ1		サボテン
F.サホ2		サボテン
F.弱リツ		弱チエリー
F.強リツ1		強チエリー
F.強リツ2		強チエリー

30

40

50

【図 25】

内部当籤役と停止図柄組合せとの対応関係

当籤役	BB(C,BB1) の成立可否	当籤役	BB(C,BB2) の成立可否
はずれ(F BB1)	○	はずれ(F BB2)	○
F BB1+F 確判1/2	×	F BB2+F 確判1/2	×
F BB1+F 確判1/3	×	F BB2+F 確判1/3	×
F BB1+F 1確判1/2	×	F BB2+F 確判1/2	×
F BB1+F リーチ目1/2A	×	F BB2+F リーチ目1/2A	×
F BB1+F リーチ目1/2B	×	F BB2+F リーチ目1/2B	×
F BB1+F リーチ目1/2C	×	F BB2+F リーチ目1/2C	×
F BB1+F リーチ目1/2D	×	F BB2+F リーチ目1/2D	×
F BB1+F 維持1/2A	×	F BB2+F 維持1/2A	×
F BB1+F 維持1/2B	×	F BB2+F 維持1/2B	×
F BB1+F 維持1/2 1st	×	F BB2+F 維持1/2 1st	×
F BB1+F 維持1/2 2nd	×	F BB2+F 維持1/2 2nd	×
F BB1+F 維持1/2 3rd	×	F BB2+F 維持1/2 3rd	×
F BB1+F RT3/1 1st	×	F BB2+F RT3/1 1st	×
F BB1+F RT3/1 213	×	F BB2+F RT3/1 213	×
F BB1+F RT3/1 231	×	F BB2+F RT3/1 231	×
F BB1+F RT3/1 3rd	×	F BB2+F RT3/1 3rd	×
F BB1+F RT4/1 123	×	F BB2+F RT4/1 123	×
F BB1+F RT4/1 132	×	F BB2+F RT4/1 132	×
F BB1+F RT4/1 2nd	×	F BB2+F RT4/1 2nd	×
F BB1+F RT4/1 3rd	×	F BB2+F RT4/1 3rd	×
F BB1+F 3枚ベル 1st	×	F BB2+F 3枚ベル 1st	×
F BB1+F 3枚ベル 2nd	×	F BB2+F 3枚ベル 2nd	×
F BB1+F 3枚ベル 3rd	×	F BB2+F 3枚ベル 3rd	×
F BB1+F 共通ベル	×	F BB2+F 共通ベル	×
F BB1+F サホ1	×	F BB2+F サホ1	×
F BB1+F サホ2	×	F BB2+F サホ2	×
F BB1+F 弱判1	×	F BB2+F 弱判1	×
F BB1+F 強判1/1	×	F BB2+F 強判1/1	×
F BB1+F 強判1/2	×	F BB2+F 強判1/2	×
F BB1+F 特殊1	○	F BB2+F 特殊1	○
F BB1+F 特殊2	○	F BB2+F 特殊2	○
F BB1+F 特殊3	○	F BB2+F 特殊3	○

【圖 26】

リール停止初期設定テーブル

内部当籤役 (小役 当籤番号)	引込優先順位 テーブル 選択テーブル	引込優先順位 テーブル番号	停止テーブル 番号	遊技状態	備考
0	—	00	0	全遊技状態	はずれ
1	0	—	1	全遊技状態	F.リリフ
2	1	—	2	全遊技状態	F.確リリフ
3	1	—	3	全遊技状態	F.1確リリフ
4	2	—	4	一般遊技中	F.リチ目リフA
	3	—		フラグ関状態	
5	4	—	5	一般遊技中	F.リチ目リフB
	3	—		フラグ関状態	
6	4	—	6	一般遊技中	F.リチ目リフC
	3	—		フラグ関状態	
7	4	—	7	一般遊技中	F.リチ目リフD
	3	—		フラグ関状態	
8	0	—	8	全遊技状態	F.維持リフA
9	5	—	9	全遊技状態	F.維持リフB
10	6	—	10	全遊技状態	F.維持リフ1st
11	7	—	11	全遊技状態	F.維持リフ2nd
12	8	—	12	全遊技状態	F.維持リフ3rd
13	9	—	13	全遊技状態	F.RT3リフ1st
14	10	—	14	全遊技状態	F.RT3リフ213
15	11	—	15	全遊技状態	F.RT3リフ231
16	12	—	16	全遊技状態	F.RT3リフ3rd
17	13	—	17	全遊技状態	F.RT4リフ123
18	14	—	18	全遊技状態	F.RT4リフ132
19	15	—	19	全遊技状態	F.RT4リフ2nd
20	16	—	20	全遊技状態	F.RT4リフ3rd
21	17	—	21	全遊技状態	F.3折ヘル1st
22	18	—	22	全遊技状態	F.3折ヘル2nd
23	19	—	23	一般遊技中	F.3折ヘル3rd
	20	—		フラグ関状態	
24	21	—	24	全遊技状態	F.共通ヘル
25	21	—	25	全遊技状態	F.サキ1
26	21	—	26	全遊技状態	F.サキ2
27	21	—	27	全遊技状態	F.弱リ
28	21	—	28	全遊技状態	F.強リ1
29	21	—	29	全遊技状態	F.強リ2
30	—	00	30	全遊技状態	F.特殊1
31	—	00	31	全遊技状態	F.特殊2
32	0	—	32	全遊技状態	F.特殊3
33	22	—	33	全遊技状態	F.RB役1
34	22	—	34	全遊技状態	F.RB役2
35	23	—	35	全遊技状態	F.RB役3
36	—	00	36	全遊技状態	F.RB役4

【図 27】

[illegible]

【圖 28】

当り要求フラグ格納領域、入賞作動フラグ格納領域(その1)

[illegible]

【圖 29】

リ要求フラグ格納領域、入費作動フラグ格納領域(その2)		データ		コンピュータ(左→右)		名称	出力 3桁16	内容(略称)
当り要求 格納領域0、 入費作動 格納領域5	B7	0/1	番1	白	白	C-0-1107.01	0	リ-白リフ
			番1	白	白	C-0-1107.02	0	リ-白リフ
			番1	白	白	C-0-1107.03	0	リ-白リフ
			番1	白	白	C-0-1107.04	0	リ-白リフ
			番1	白	白	C-0-1107.05	0	リ-白リフ
			番1	白	白	C-0-1107.06	0	リ-白リフ
			番1	白	白	C-0-1107.07	0	リ-白リフ
			番1	白	白	C-0-1107.08	0	リ-白リフ
			番1	白	白	C-0-1107.09	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.21	0	リ-白リフ
当り要求 格納領域4、 入費作動 格納領域4	B5	0/1	番1	白	白	C-0-1107.01	0	リ-白リフ
			番1	白	白	C-0-1107.02	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.23	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.24	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.25	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.26	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.27	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.28	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.29	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.30	0	リ-白リフ
	B3	0/1	サテナツ	リフ	リフ	C-0-1107.01	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.02	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.03	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.04	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.05	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.06	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.07	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.08	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.09	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.10	0	リ-白リフ
	B1	0/1	リフ	リフ	リフ	C-0-1107.01	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.02	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.03	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.04	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.05	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.06	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.07	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.08	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.09	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.10	0	リ-白リフ
当り要求 格納領域4、 入費作動 格納領域4	B7	0/1	白	白	白	C-0-1107.01	0	リ-白リフ
			白	白	白	C-0-1107.02	0	リ-白リフ
			白	白	白	C-0-1107.03	0	リ-白リフ
			白	白	白	C-0-1107.04	0	リ-白リフ
			白	白	白	C-0-1107.05	0	リ-白リフ
			白	白	白	C-0-1107.06	0	リ-白リ
当り要求 格納領域0、 入費作動 格納領域3	B6	0/1	リフ	リフ	リフ	C-0-1107.01	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.02	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.03	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.04	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.05	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.06	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.07	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.08	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.09	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.10	0	リ-白リフ
	B5	0/1	サテナツ	リフ	リフ	C-0-1107.01	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.02	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.03	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.04	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.05	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.06	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.07	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.08	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.09	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.10	0	リ-白リフ
	B4	0/1	サテナツ	リフ	リフ	C-0-1107.01	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.02	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.03	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.04	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.05	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.06	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.07	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.08	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.09	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.10	0	リ-白リフ
	B3	0/1	白	白	白	C-0-1107.01	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.01	0	リ-白リフ
	B2	0/1	リフ	リフ	リフ	C-0-1107.01	0	リ-白リフ
			リフ	リフ	リフ	C-0-1107.01	0	リ-白リフ
	B1	0/1	サテナツ	リフ	リフ	C-0-1107.01	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.01	0	リ-白リフ
	B0	0/1	サテナツ	リフ	リフ	C-0-1107.01	0	リ-白リフ
			サテナツ	リフ	リフ	C-0-1107.01	0	リ-白リフ

【図 30】

当り要求フラグ格納領域、入食作動フラグ格納領域(その3)								
格納領域		データ	コンビネーション(左=中=右)		名称	抽出し 3枚掛け	内容(略称)	
当り要求 格納領域2、 入食作動 格納領域2	B7	0/1	帽子	着7	サボテン1	C.1st A 01	1	1枚抽出
			帽子	着7	サボテン2	C.1st A 02	1	1枚抽出
			帽子	チリ上2	サボテン1	C.1st A 03	1	1枚抽出
			帽子	チリ上2	サボテン2	C.1st A 04	1	1枚抽出
	B8	0/1	帽子	チリ下	サボテン1	C.1st B 01	1	1枚抽出
			帽子	チリ下	サボテン2	C.1st B 02	1	1枚抽出
	B5	0/1	帽子	サボテン2	サボテン1	C.1st C 01	1	1枚抽出
			帽子	サボテン2	サボテン2	C.1st C 02	1	1枚抽出
	B4	0/1	白7	リブレイ	サボテン1	C.2nd A 01	1	1枚抽出
			白7	リブレイ	サボテン2	C.2nd A 02	1	1枚抽出
			サボテン2	リブレイ	サボテン1	C.2nd A 03	1	1枚抽出
			サボテン2	リブレイ	サボテン2	C.2nd A 04	1	1枚抽出
着7			リブレイ	サボテン1	C.2nd B 01	1	1枚抽出	
着7			リブレイ	サボテン2	C.2nd B 02	1	1枚抽出	
B3	0/1	サボテン1	リブレイ	サボテン1	C.2nd B 03	1	1枚抽出	
		サボテン1	リブレイ	サボテン2	C.2nd B 04	1	1枚抽出	
B2	0/1	白7	チリ下	チリ下	C.SP 01	1	確定抽出	
B1	0	着7	チリ下	チリ下	C.SP 02	1	確定抽出	
B0	0				—			
当り要求 格納領域1、 入食作動 格納領域1	B7	0/1	サボテン1	サボテン2	サボテン1	C.3枚A 01	3	サボテン
			サボテン1	サボテン2	サボテン2	C.3枚A 02	3	サボテン
	B6	0/1	サボテン2	サボテン2	サボテン1	C.3枚B 01	3	サボテン
			サボテン2	サボテン2	サボテン2	C.3枚B 02	3	サボテン
	B5	0/1	サボテン2	サボテン2	サボテン1	C.3枚C 01	3	サボテン
			サボテン2	サボテン2	サボテン2	C.3枚C 02	3	サボテン
	B4	0/1	リブレイ	チリ上2	サボテン2	C.3枚D 01	3	サボテン
			リブレイ	サボテン2	チリ上2	C.3枚E 01	3	サボテン
	B3	0/1	リブレイ	サボテン2	チリ上2	C.3枚E 02	3	サボテン
			帽子	サボテン2	リブレイ	C.3枚F 01	3	サボテン
	B1	0				—		
	B0	0				—		
当り要求 格納領域0、 入食作動 格納領域0	B7	0/1	着7	リブレイ	リブレイ	C.強2枚A 01	2	強チチェリー
			リブレイ	リブレイ	帽子	C.強2枚B 01	2	強チチェリー
			リブレイ	サボテン2	帽子	C.強2枚B 02	2	強チチェリー
			リブレイ	チリ上2	帽子	C.強2枚B 03	2	強チチェリー
	B6	0/1	着7	白7	リブレイ	C.強2枚A 01	2	強チチェリー
			着7	チリ上1	リブレイ	C.強2枚A 02	2	強チチェリー
			着7	サボテン2	リブレイ	C.強2枚A 03	2	強チチェリー
			着7	リブレイ	チリ上1	C.強2枚B 01	2	強チチェリー
	B5	0/1	着7	リブレイ	チリ上2	C.強2枚B 02	2	強チチェリー
			リブレイ	着7	白7	C.強2枚C 01	2	強チチェリー
			リブレイ	着7	着7	C.強2枚C 02	2	強チチェリー
			リブレイ	サボテン2	白7	C.強2枚C 03	2	強チチェリー
	B4	0/1	リブレイ	サボテン2	白7	C.強2枚C 04	2	強チチェリー
			リブレイ	サボテン2	着7	C.強2枚C 05	2	強チチェリー
			リブレイ	サボテン2	チリ下	C.強2枚C 06	2	強チチェリー
			リブレイ	チリ上2	白7	C.強2枚C 07	2	強チチェリー
	B3	0/1	リブレイ	チリ上2	着7	C.強2枚C 08	2	強チチェリー
			リブレイ	チリ上2	チリ下	C.強2枚C 09	2	強チチェリー
						—		
						—		
	B2	0				—		
	B1	0				—		
	B0	0				—		

※B0～B7:ビット0～ビット7, 「0/1」:「0」又は「1」

【図 3 1】

持越役格納領域		データ		内容
格納領域種別				
持越役 格納領域	ビット7	0		未使用
	ビット6	0		未使用
	ビット5	0		未使用
	ビット4	0		未使用
	ビット3	0		未使用
	ビット2	0		未使用
	ビット1	0 or 1		BB2
	ビット0	0 or 1		BB1

【図 3 2】

遊技状態フラグ格納領域		データ		内容
格納領域種別				
遊技状態 フラグ 格納領域	ビット7		0	未使用
	ビット6	0 or 1		RT5状態
	ビット5	0 or 1		RT4状態
	ビット4	0 or 1		RT3状態
	ビット3	0 or 1		RT2状態
	ビット2	0 or 1		RT1状態
	ビット1	0 or 1		RB遊技状態
	ビット0	0 or 1		BB遊技状態

【図 3 3】

作動ストップボタン格納領域

データ		内容
ビット7	0	未使用
ビット6	0 or 1	右ストップボタン操作有効
ビット5	0 or 1	中ストップボタン操作有効
ビット4	0 or 1	左ストップボタン操作有効
ビット3	0	未使用
ビット2	0 or 1	右ストップボタン操作
ビット1	0 or 1	中ストップボタン操作
ビット0	0 or 1	左ストップボタン操作

※ビット0～2は、「0」で操作なし「1」で操作あり

※ビット4～6は、「0」で無効「1」で有効

【図 3 4】

押下順序格納領域

データ		内容
ビット7	0	未使用
ビット6	0	未使用
ビット5	0 or 1	右→中→左
ビット4	0 or 1	右→左→中
ビット3	0 or 1	中→右→左
ビット2	0 or 1	中→左→右
ビット1	0 or 1	左→右→中
ビット0	0 or 1	左→中→右

※ビット0～6は、「0」で無効「1」で有効

【図 3 5】

図柄コード格納領域

格納領域種別	データ		内容
図柄コード 格納領域 11	ビット7	0 or 1	R RT移行目A 01
			R RT移行目A 02
			R RT移行目A 03
			R RT移行目A 04
	ビット6	0 or 1	R RT移行目B 01
			R RT移行目B 02
			未使用
			未使用
図柄コード 格納領域 0	ビット5	0	未使用
			未使用
			未使用
			未使用
	ビット4	0	未使用
			未使用
			未使用
			未使用
図柄コード 格納領域 0	ビット3	0 or 1	C 弱2枚A 01
			C 弱2枚B 01
			C 弱2枚B 02
			C 弱2枚B 03
	ビット2	0 or 1	C 強2枚A 01
			C 強2枚A 02
			C 強2枚A 03
			C 強2枚B 01
	ビット1	0 or 1	C 強2枚B 02
			C 強2枚C 01
			C 強2枚C 02
			C 強2枚C 03
	ビット0	0 or 1	C 強2枚C 04
			C 強2枚C 05
			C 強2枚C 06
			C 強2枚C 07
図柄コード 格納領域 0	ビット0	0	未使用
			未使用
			未使用
			未使用

【図 3 6】

内部当籤役とサブフラグとの対応表(その1)

内部当籤役 (小役当籤番号)	サブフラグ (フラグデータ)	サブフラグEX (フラグデータ)	サブフラグD (フラグデータ)
はずれ(00)	ハズレ(00)	ハズレ(00)	ハズレ(00)
F.チリリブ(01)	2連チリリブ(01)	リブレイ(01)	
F.確チリリブ(02)	3連チリリブA(02)	リブレイ(01), 確定役(06), 3連チリリブ(07)	ハズレ(00), 確定役(04), 3連チリリブ(05)
F.1確チリリブ(03)	3連チリリブB(03)		
F.リーチ目リブA(04)	リーチ目リブ1(04)		
F.リーチ目リブB(05)	リーチ目リブ2(05)	リブレイ(01), 確定役(06), リーチ目リブ(08)	ハズレ(00), 確定役(04), リーチ目リブ(06)
F.リーチ目リブC(06)	リーチ目リブ3(06)		
F.リーチ目リブD(07)	リーチ目リブ4(07)		
F.維持リブA(08)			
F.維持リブB(09)	リブレイ(08)		
F.維持リブ_1st(10)			
F.維持リブ_2nd(11)	押し順リブ1(09)		
F.維持リブ_3rd(12)			
F.RT3リブ_1st(13)			
F.RT3リブ_213(14)	押し順リブ2(10)		
F.RT3リブ_231(15)			
F.RT3リブ_3rd(16)			
F.RT4リブ_123(17)	押し順リブ3(11)		
F.RT4リブ_132(18)			
F.RT4リブ_2nd(19)			
F.RT4リブ_3rd(20)			
F.3択ベル_1st(21)			
F.3択ベル_2nd(22)	押し順ベル(12)		
F.3択ベル_3rd(23)			
F.共通ベル(24)	共通ベル(13)		
F.サボ1(25)			
F.サボ2(26)	サボテン(14)	サボテン(03)	サボテン(01)
F.弱チリ(27)	弱チェリー(15)	弱チェリー(04)	弱チェリー(02)
F.強チリ1(28)	強チェリー(16)	強チェリー(05)	強チェリー(03)
F.強チリ2(29)			
F.特殊1(30)	リーチ目1(17)		
F.特殊2(31)	リーチ目2(18)	ハズレ(00)	ハズレ(00)
F.特殊3(32)			

10

20

30

40

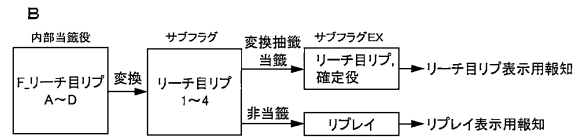
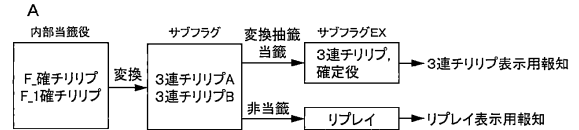
50

【図 37】

内部当籤役とサブフラグとの対応表(その2)

内部当籤役 (特賞当籤番号)	サブフラグ
F_BB1(01)	BB
F_BB2(02)	

【図 38】

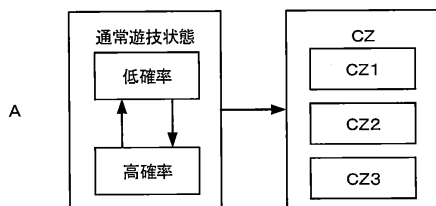


※ 変換抽籤はRT4中のみ行う

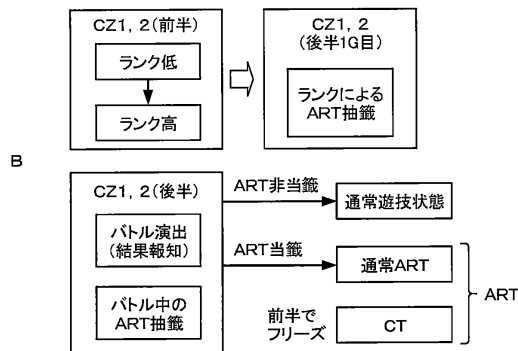
10

【図 39】

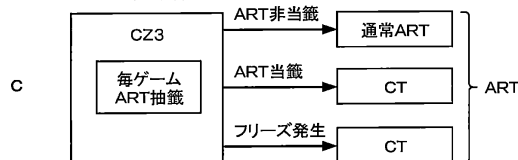
一般遊技状態中の遊技の流れ



CZ1, 2中の遊技の流れ



CZ3中の遊技の流れ



【図 40】

通常中高確率抽籤テーブル(毎ゲーム参照)

現在 の状態	移行後 の状態	共通ベル	サポテン	当籤役 弱チエリー	強チエリー	それ以外
低確率中	低確率	極高	極高	高	極高	確定
	高確率	極低	低	中	低	0
高確率中	低確率	0	0	0	0	低
	高確率	確定	確定	確定	確定	極高

通常中高確率抽籤テーブル(状況によって参照)

移行後 の状態	設定変更時	ボーナス 終了時	CZ, ART 終了時
低確率	極高	0	極高
高確率	低	確定	低

確率分母256

※ 0<極々低<極低<低<中<高<極高<極々高<確定(=100%)

20

30

40

50

【図 4 1】

CZ抽籤テーブル(初当たり用)					
現在の状態	内容	当籤役			
		サポテン	弱チェリー	強チェリー	BB
A 低確率中	非当籤	極高	確定	高	高
	CZ1当籤	極低	0	低	低
	CZ2当籤	極低	0	極低	極低
	CZ3当籤	0	0	0	0
高確率中	非当籤	高	極高	中	中
	CZ1当籤	中	0	中	中
	CZ2当籤	低	0	低	低
	CZ3当籤	極低	極低	極低	極低

CZ抽籤テーブル(引き戻し用)		
状況	内容	抽籤値
B CZ失敗時 ART終了時	非当籤	極高
	CZ1当籤	低
	CZ3当籤	0

※ 確率分母256

【図 4 2】

CZ1中モードアップ抽籤テーブル					
現在のモード	内容	当籤役			
		リプレイ	共通ベル	弱チェリー サポテン	強チェリー
モード1 モード2	非当選	極高	0	0	0
	モード1UP	低	確定	極高	0
	モード2UP	0	0	極低	0
	モード3UP	0	0	極低	極高
	モード4UP	0	0	0	0
	モード5UP	0	0	0	0
	モード6UP	0	0	極低	低
	モード6UP フリーズ発生	0	0	0	0
モード3	非当選	極高	0	0	0
	モード1UP	低	確定	極高	0
	モード2UP	0	0	極低	0
	モード3UP	0	0	極低	極高
	モード4UP	0	0	0	0
	モード5UP	0	0	0	0
	モード6UP	0	0	極低	低
	モード6UP フリーズ発生	0	0	0	0
モード4	非当選	極高	0	0	0
	モード1UP	低	確定	極高	0
	モード2UP	0	0	極低	0
	モード3UP	0	0	極低	極高
	モード4UP	0	0	0	0
	モード5UP	0	0	0	0
	モード6UP	0	0	極低	低
	モード6UP フリーズ発生	0	0	0	0
モード5 モード6	非当選	極高	0	0	0
	モード1UP	低	確定	極高	0
	モード2UP	0	0	極低	0
	モード3UP	0	0	極低	極高
	モード4UP	0	0	0	0
	モード5UP	0	0	0	0
	モード6UP	0	0	極低	低
	モード6UP フリーズ発生	0	0	0	0
CZ2開始 2G以内	非当選	極高	0	0	0
	モード1UP	低	確定	極高	0
	モード2UP	0	0	低	0
	モード3UP	0	0	0	0
	モード4UP	0	0	0	0
	モード5UP	0	0	0	0
	モード6UP	0	0	0	0
	モード6UP フリーズ発生	0	0	0	確定

※ 確率分母256

【図 4 3】

CZ2中ポイント抽籤テーブル					
現在のポイント	内容	当籤役			
		リプレイ	共通ベル	弱チェリー サポテン	強チェリー
ポイント 0~9	非当選	0	0	0	0
	ポイント1UP	極高	中	0	0
	ポイント2UP	0	中	高	0
	ポイント3UP	極低	極低	低	0
	ポイント4UP	極低	極低	極低	0
	ポイント5UP	極低	極低	極低	高
	ポイント10UP	0	0	極低	中
	ポイント10UP フリーズ発生	0	0	0	0
CZ2開始 2G以内	非当選	0	0	0	0
	ポイント1UP	極高	中	0	0
	ポイント2UP	0	中	高	0
	ポイント3UP	極低	極低	低	0
	ポイント4UP	極低	極低	極低	0
	ポイント5UP	0	0	0	0
	ポイント10UP	0	0	0	0
	ポイント10UP フリーズ発生	0	0	0	確定

※ 確率分母256

【図 4 4】

CZ中ART抽籤テーブル(CZ1用)												
内容	現在のモード											
	モード1	モード2	モード3	モード4	モード5	モード6	モード7	モード8	モード9	モード10	モード11	モード12
非当籤	極高	極高	極高	極高	極高	極高	極高	極高	極高	極高	極高	極高
ART当籤	0	0	0	0	0	0	0	0	0	0	0	0

CZ中ART抽籤テーブル(CZ2用)												
内容	現在のモード											
	モード1	モード2	モード3	モード4	モード5	モード6	モード7	モード8	モード9	モード10	モード11	モード12
非当籤	極高	極高	極高	極高	極高	極高	極高	極高	極高	極高	極高	極高
ART当籤	0	0	0	0	0	0	0	0	0	0	0	0

CZ中ART抽籤テーブル(CZ1, 2共通 後半ハットル中用)				
内容	当籤役			
	弱チェリー サポテン	強チェリー	中	中
非当籤	極高	極高	中	中
ART当籤	低	低	中	中

※ 確率分母256

10

20

30

40

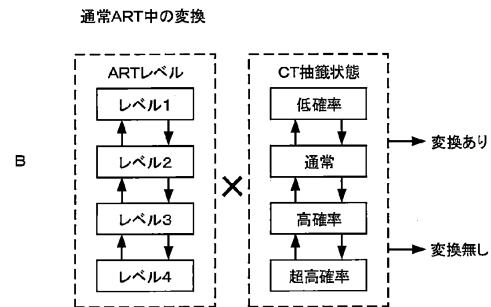
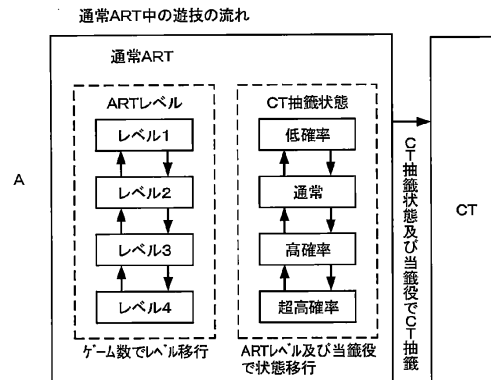
50

【図 4 5】

CZ中ART抽籤テーブル(CZ3用)						
状態	内容	リプレイ	共通ベル	当籤役 弱チェリー サボテン	強チェリー	それ以外
CZ3開始 2G以内	非当籤	極高	確定	中	0	確定
	ART当籤+通常CT	0	0	0	0	0
	ART当籤+高確CT	0	0	0	0	0
	フリーズ+通常CT	低	0	中	確定	0
CZ3開始 3G～17 G	非当籤	極高	極高	中	0	極高
	ART当籤+通常CT	極低	低	中	中	極低
	ART当籤+高確CT	0	0	0	0	0
	フリーズ+通常CT	極低	0	極低	中	0
	フリーズ+高確CT	0	0	0	0	0

※ 確率分母256

【図 4 6】



【図 4 7】

ART中フラグ変換抽籤テーブル(第1段階用)		
内容	当籤役	
	F.確チリP or F.1確チリP	
変換無し	極々高	
変換あり(仮)	極々低	

※ 確率分母256

ART中フラグ変換抽籤テーブル(第2段階用(第1段階で変換あり(仮)に当籤))				
ARTレベル	CT抽籤状態	内容	当籤役	
			F.確チリP or F.1確チリP	F.リチ目リP A～D
初回 (一度CTに当籤 するまで)	低確率	変換無し	0	極高
		変換有り	確定	低
	通常	変換無し	0	極高
		変換有り	確定	低
	高確率	変換無し	0	0
		変換有り	確定	確定
	超高確率	変換無し	0	0
		変換有り	確定	確定
ARTレベル1	低確率	変換無し	極高	極々高
		変換有り	極低	極々低
	通常	変換無し	極高	極々高
		変換有り	極低	極々低
	高確率	変換無し	極高	高
		変換有り	極低	中
	超高確率	変換無し	極高	0
		変換有り	極低	確定
ARTレベル 2～4	低確率	変換無し	極高	極々高
		変換有り	低	極々低
	通常	変換無し	極高	極々高
		変換有り	低	極々低
	高確率	変換無し	0	高
		変換有り	確定	中
	超高確率	変換無し	0	0
		変換有り	確定	確定

※ 確率分母256

【図 4 8】

ARTレベル決定テーブル(ART当籤時)		
ARTレベル	抽籤値	
ARTレベル1	低	
ARTレベル2	高	
ARTレベル3	極低	
ARTレベル4	極高	

※ ART当籤時にフリーズ発生している場合は、ARTレベル2が決定
※ 確率分母256

ARTレベル決定テーブル(通常ART中)				
経過ゲーム数	移行先のARTレベル	現在のARTレベル		
	ARTレベル	ARTレベル1	ARTレベル2	ARTレベル3
50ゲーム経過	ARTレベル1	確定	0	0
	ARTレベル2	0	確定	0
	ARTレベル3	0	0	確定
100ゲーム経過	ARTレベル1	確定	低	0
	ARTレベル2	0	高	0
	ARTレベル3	0	0	確定
200ゲーム経過	ARTレベル1	確定	極高	0
	ARTレベル2	0	低	0
	ARTレベル3	0	0	確定
300ゲーム経過	ARTレベル1	確定	極高	0
	ARTレベル2	0	極低	中
	ARTレベル3	0	0	中
500ゲーム経過	ARTレベル1	確定	極高	0
	ARTレベル2	0	極低	極高
	ARTレベル3	0	0	低
1000ゲーム経過	ARTレベル1	確定	極高	0
	ARTレベル2	0	極低	極高
	ARTレベル3	0	0	極低
CT突入時 (0～100ゲーム)	ARTレベル1	中	低	0
	ARTレベル2	中	高	0
	ARTレベル3	0	0	確定
CT突入時 (101～500ゲーム)	ARTレベル1	極高	中	0
	ARTレベル2	極低	中	0
	ARTレベル3	0	0	確定
CT突入時 (501ゲーム以上)	ARTレベル1	極高	極高	0
	ARTレベル2	極低	極低	中
	ARTレベル3	0	0	中

※ 確率分母256

【図 49】

【圖 50】

通常ART中高確率抽籤テーブル

現在の状態	移行後の 状態	当番役					
		押し順へル 共通へル	サボテン	弱チヤリ	強チヤリ	3連リブ リチ目リブ	それ以外
低確率 (ARTへル1)	低確率	極高	中	中	極高	極高	確定
	通常	極低	中	中	0	0	0
	高確率	0	極低	低	0	0	0
	超高確率	0	0	0	0	極低	0
低確率 (ARTへル 2~4)	低確率	極高	中	中	極高	極高	確定
	通常	極低	中	中	0	0	0
	高確率	0	極低	極低	低	0	0
	超高確率	0	0	0	0	極低	0
通常 (ARTへル1)	低確率	0	0	0	0	極高	中
	通常	極高	中	中	高	0	中
	高確率	極低	中	中	中	0	0
	超高確率	0	0	0	極低	極低	0
通常 (ARTへル 2~4)	低確率	0	0	0	0	極高	低
	通常	極高	中	中	高	0	高
	高確率	低	中	中	中	0	0
	超高確率	0	0	0	極低	極低	0
高確率	低確率	0	0	0	0	極高	中
	通常	0	0	0	0	0	0
	高確率	確定	極高	極高	極高	0	中
	超高確率	0	極低	極低	極低	極低	0
超高確率	低確率	0	0	0	0	0	0
	通常	0	0	0	0	0	0
	高確率	0	0	0	0	低	0
	超高確率	確定	確定	確定	確定	高	確定

※ 確率分母256

ART中CT抽籤テーブル

現在の状態	内容	当役位					
		サボテン	弱フェリ	強フェリ	3連チリ ^ア リーチ目リ ^ア	BB	それ以外
低確率	非常籤	極高	極高	極高	0	確定	極度高
	通常CT	極低	極低	極低	確定	0	0
	高確率CT	0	0	0	0	0	極々低
通常	非常籤	極高	極高	極高	0	確定	極度高
	通常CT	極低	極低	低	確定	0	0
	高確率CT	0	0	0	0	0	極々低
高確率	非常籤	極高	極高	中	0	確定	極度高
	通常CT	低	極低	中	確定	0	極々低
	高確率CT	0	0	0	0	0	極々低
超高確率	非常籤	極高	極高	0	0	確定	極高
	通常CT	低	極低	確定	確定	0	極低
	高確率CT	0	0	0	0	0	極々低

※ サボテン、弱チェリー、強チェリー、3連チリリップ、リーチ目リップ、BBは、確率分母256
それ以外は、確率分母65536

※ BBは内部当籤時にはCTの抽籤を行わず(必ず非当籤)、ボーナス状態終了時に抽籤する。

※ 確率分母256

10

20

【図 5 1】

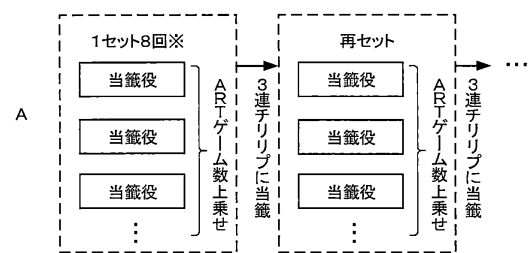
通常ART中上乘せ抽籤テーブル

内容	当籤役		
	サテン	弱チエリ	強チエリ
非当籤	極高	極々高	0
上乗せ 10G	0	0	中
上乗せ 20G	0	0	中
上乗せ 30G	低	0	低
上乗せ 50G	極低	0	極低
上乗せ 100G	極低	極々低	極低
上乗せ 300G	極々低	極々低	極々低

※ 確率分母65536

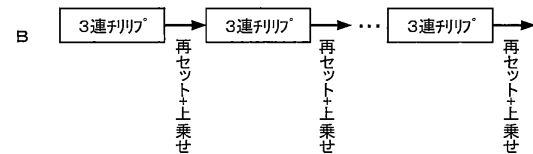
【圖 5 2】

CT中の遊技の流れ1

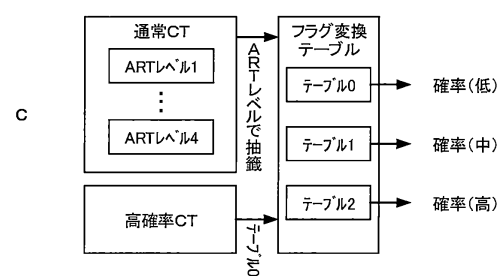


※ 上乗せできない遊技に限り、回数を減算

CT中の遊技の流れ2



CT中の変換



30

40

50

【図 5 3】

CT中テーブル抽籤テーブル

内容	状態				
	ARTレベル1 (通常CT)	ARTレベル2 (通常CT)	ARTレベル3,4 (通常CT)	初回CT	高確率CT
テーブル0	極高	高	高	高	確定
テーブル1	極低	低	低	低	0
テーブル2	極々低	極々低	極々低	極々低	0

※ 確率分母256

【図 5 4】

CT中フラグ変換抽籤テーブル

フラグ 変換テーブル	内容	当籤役	
		F_確率リブ or F_1確率リブ	F_リーチ目リブA～D
テーブル0	変換無し	0	高
	変換有り	確定	低
テーブル1	変換無し	0	低
	変換有り	確定	高
テーブル2	変換無し	0	0
	変換有り	確定	確定

※ 確率分母256

10

【図 5 5】

CT中上乗せ抽籤テーブル

状態	内容	当籤役			
		サボテン	弱チェリー	強チェリー	3連チリリブ
通常CT	非当籤	0	0	0	0
	上乗せ 10G	極高	極高	0	極高 ※1
	上乗せ 20G	極低	極低	0	極低 ※2
	上乗せ 30G	0	0	極高	0
	上乗せ 50G	極々低	極々低	極々低	極々低
	上乗せ 100G	極々低	極々低	極々低	極々低
	上乗せ 300G	極々低	極々低	極々低	極々低
	上乗せ 300G	極々低	極々低	極々低	極々低
高確率CT	非当籤	0	0	0	0
	上乗せ 10G	0	0	0	0
	上乗せ 20G	0	0	0	0
	上乗せ 30G	0	0	0	0
	上乗せ 50G	極高	極高	高	極高
	上乗せ 100G	極低	極低	低	極低
	上乗せ 300G	極々低	極々低	極々低	極々低
	上乗せ 300G	極々低	極々低	極々低	極々低

※ 同一のCT中に3連チリリブが当籤した回数に応じて、上乗せゲーム数が変わる。

1～8回: ※1は10ゲーム、※2は20ゲーム

9～16回: ※1、2ともに20ゲーム

17～24回: ※1、2ともに30ゲーム

25回以上: ※1、2ともに50ゲーム

※ 確率分母65536

※ その他の役については非当籤

【図 5 6】

CT中セット数上乗せ抽籤テーブル

状態	内容	当籤役
		リーチ目リブ
通常CT	非当籤	0
	通常CT当籤	極々高
	高確率CT当籤	極々低
高確率CT	非当籤	0
	通常CT当籤	0
	高確率CT当籤	確定

※ 確率分母256

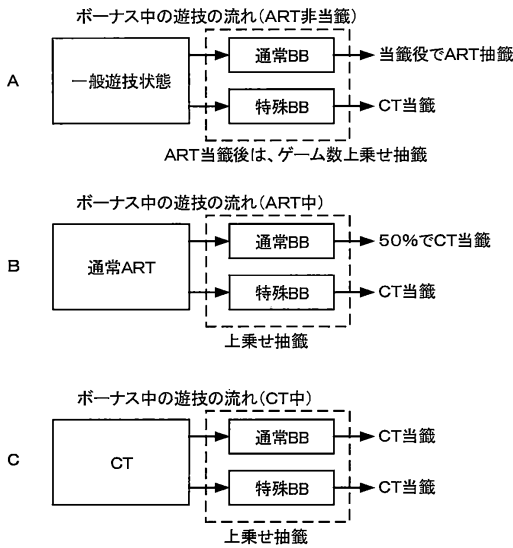
20

30

40

50

【図 5 7】



【図 5 8】

ボーナス種別抽籤テーブル

状態	内容	抽籤値
CT中	通常BB	極々高
	特殊BB	極々低
それ以外	通常BB	極々高
	特殊BB	極々低

※ 確率分母256

10

20

【図 5 9】

ボーナス中ARTゲーム数上乘せ抽籤テーブル

ボーナス種別	内容	当籤役		
		F.RB役1,2	F.RB役3	F.RB役4
通常BB	非当籤	0	確定	確定
	5G	0	0	0
	10G	0	0	0
	20G	0	0	0
	30G	0	0	0
	50G	極高	極高	0
	100G	極低	極低	0
	300G	極々低	極々低	0
特殊BB	非当籤	0	0	0
	5G	0	0	高
	10G	0	極高	低
	20G	0	極低	極低
	30G	0	極低	極低
	50G	0	0	極々低
	100G	極高	0	極々低
	300G	低	0	極々低

※ 確率分母256

【図 6 0】

ボーナス終了時CT抽籤テーブル

内容	ボーナス種別など				
	通常BB (一般遊技状態)	通常BB (通常ART)	特殊BB (一般or通常ART)	通常CT中 (通常or特殊BB)	高確率CT中 (通常or特殊BB)
非当籤	確定	50%	0	0	0
通常CT当籤	0	50%	確定	50%	0
高確率CT当籤	0	0	0	50%	確定

※ 確率分母256

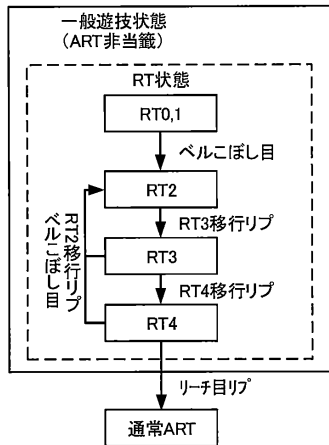
30

40

50

【図 6 1】

一般遊技状態中の遊技の流れ(その他)



【図 6 2】

非ART中フラグ変換抽籤テーブル

内容	当義役
	F-リチ目リザA~D
変換無し	極高
変換あり	極低

※ 確率分母256

【図 6 3】

ナビマークの対応関係

ART 最難中

メインナビ											
1	2	3	4	5	6	7	8	9	10	11	
内容											
A	ペンナビ	中	左-中-右	左-中-右	中-左-右	中-左-中	右-左-中	右-中-左	白7	青7	
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								

ART 中 (B&Bナビ)

メインナビ											
1	2	3	4	5	6	7	8	9	10	11	
内容											
B	ペンナビ	中	左-中-右	左-中-右	中-左-右	中-左-中	右-左-中	右-中-左	白7	青7	
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								

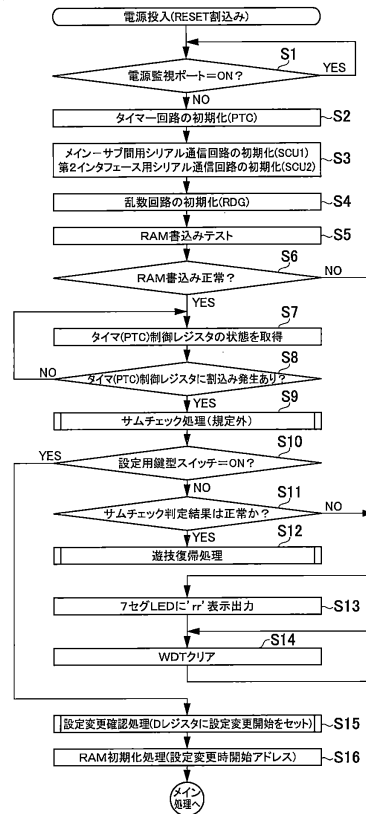
ART 中 (B&Bナビ)

メインナビ											
1	2	3	4	5	6	7	8	9	10	11	
内容											
C	ペンナビ	中	左-中-右	左-中-右	中-左-右	中-左-中	右-左-中	右-中-左	白7	青7	
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								

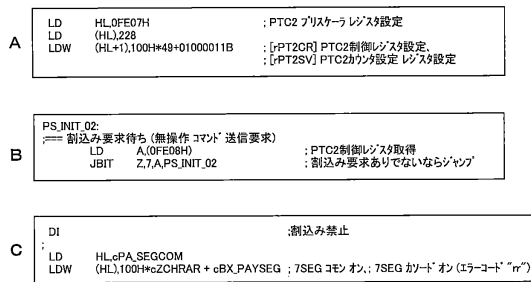
ART 中 (B&Bナビ)

メインナビ											
1	2	3	4	5	6	7	8	9	10	11	
内容											
D	ペンナビ	中	左-中-右	左-中-右	中-左-右	中-左-中	右-左-中	右-中-左	白7	青7	
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								
	F 3Rへ△, 1△	F 3Rへ△, 2nd	F 3Rへ△, 3rd								

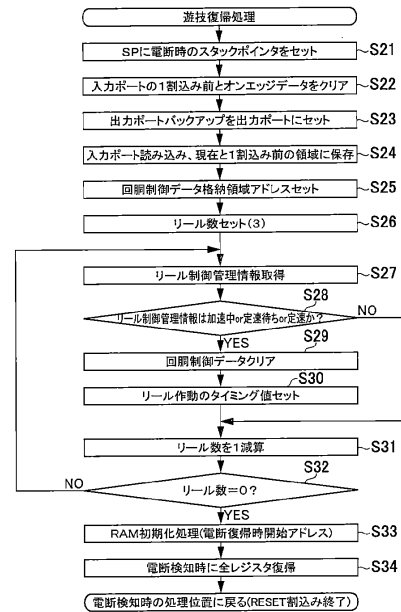
【図 6 4】



【図 65】



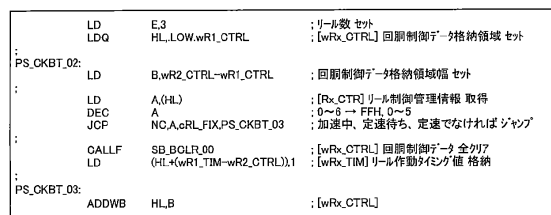
【図 66】



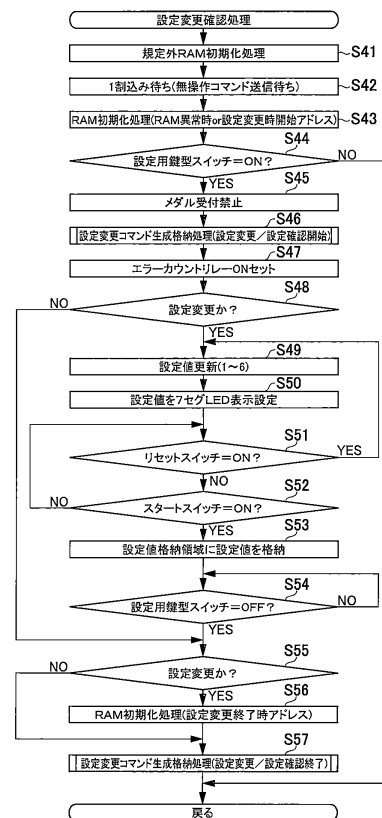
10

20

【図 67】



【図 68】



30

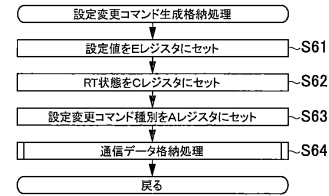
40

50

【図 69】

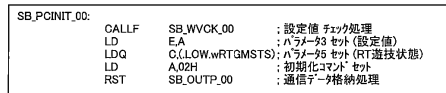


【図 70】

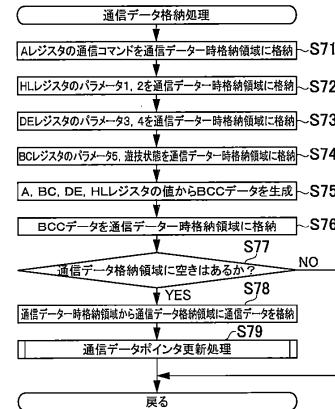


10

【図 71】



【図 72】



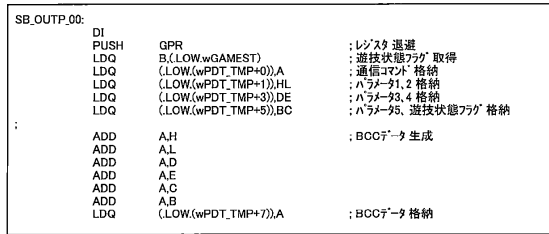
20

30

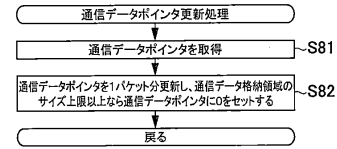
40

50

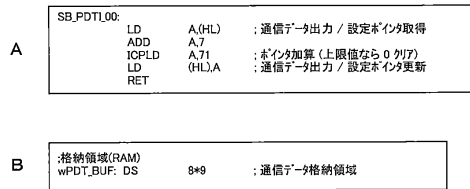
【図 7 3】



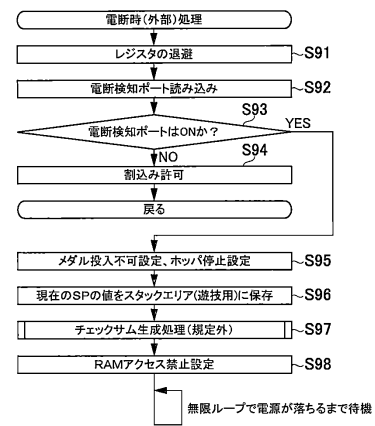
【図 7 4】



【図 7 5】



【図 7 6】



10

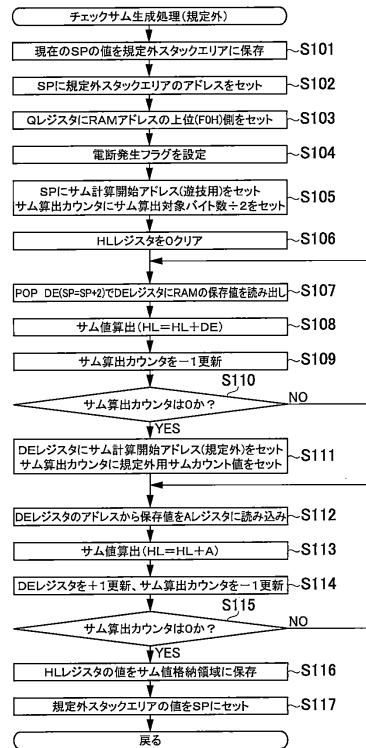
20

30

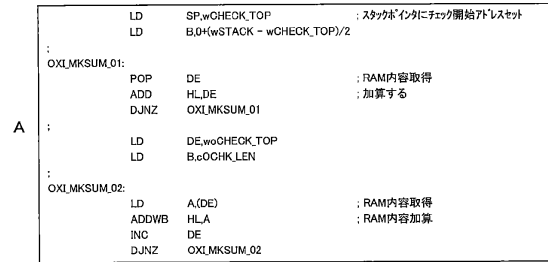
40

50

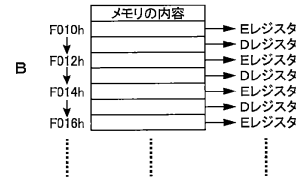
【図 77】



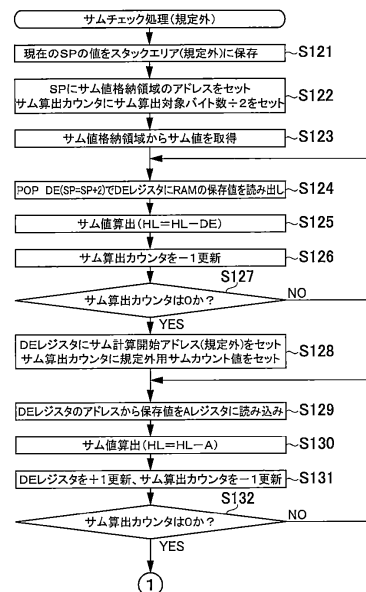
【図 78】



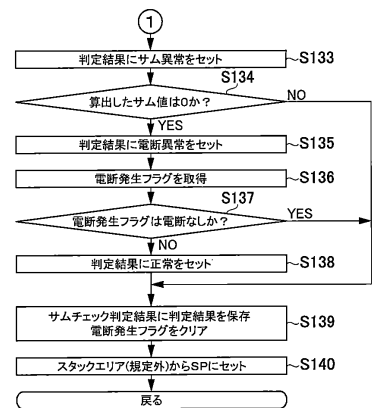
POP DE命令実行時のスタックポインタ及びRAM間のデータの流れ



【図 79】



【図 80】



10

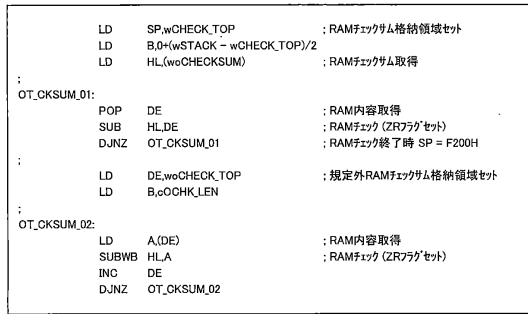
20

30

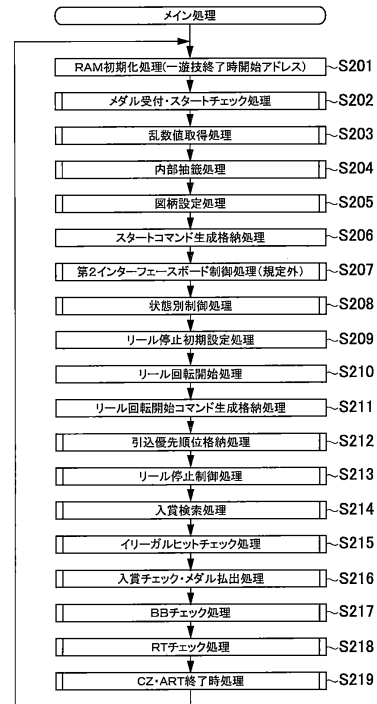
40

50

【図 8 1】



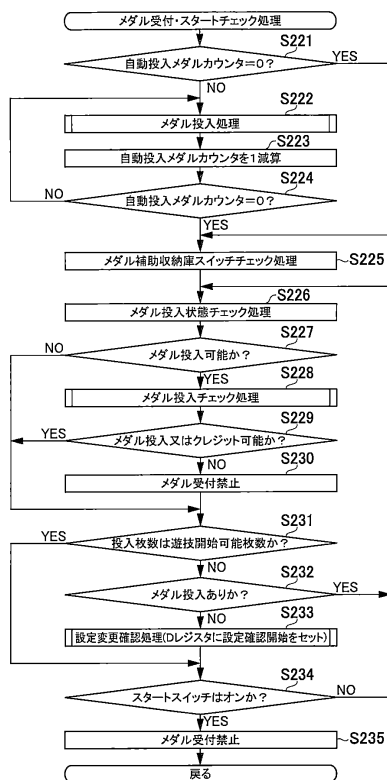
【図 8 2】



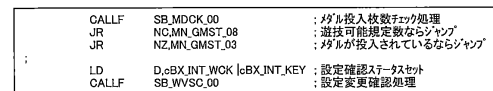
10

20

【図 8 3】



【図 8 4】

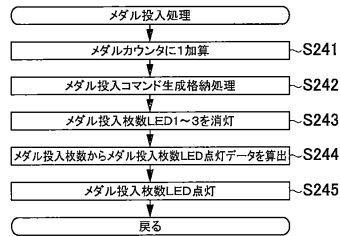


30

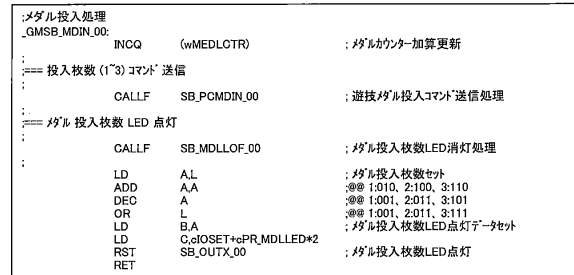
40

50

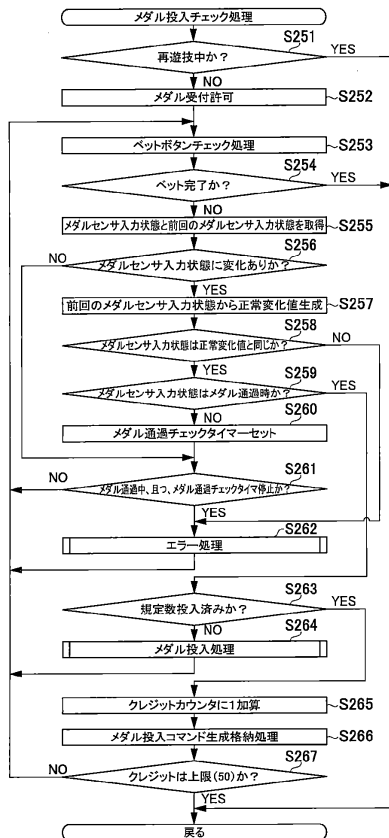
【図 85】



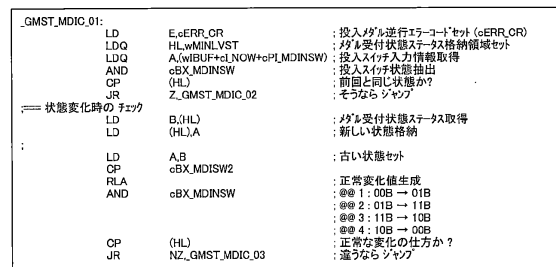
【図 86】



【図 87】



【図 88】



10

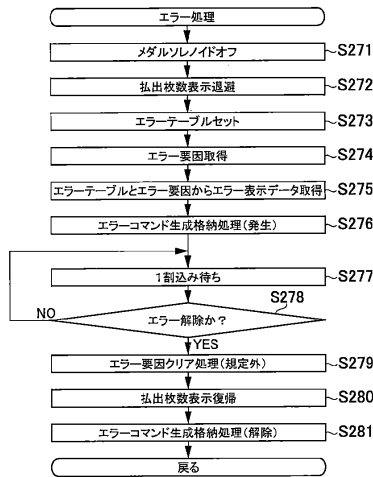
20

30

40

50

【図 89】

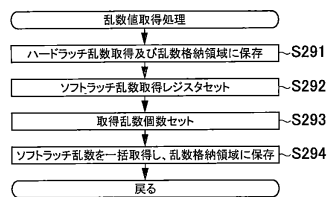


【図 90】

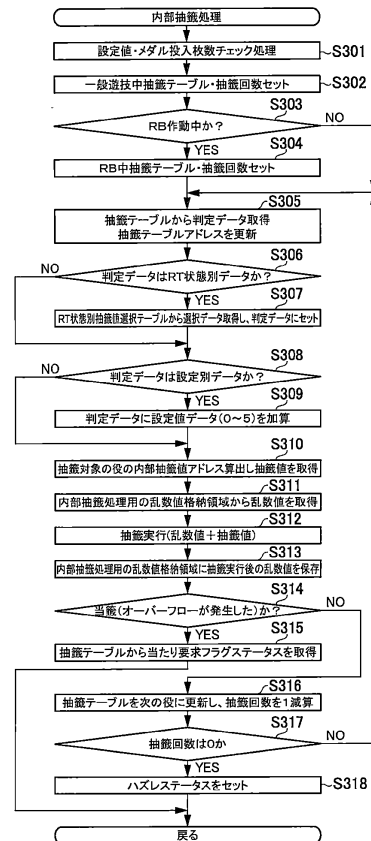
dERRTB	EGU	\$	
dERR_HE	DB	00000000B	; ホットチェックなし
dERR_HJ	DB	00110111B,01001111B	; ホットエラーエンブレムエラー "HE"
dERR_HJ	DB	01000000B	; ホットエラーエンブレムエラー "HJ"
dERR_CC	DB	00100000B	; メダル通過チェックスイチ2
dERR_CJ	DB	01001110B,01001110B	; 投入メダル通過時間エラー "CC"
dERR_CJ	DB	00000100B	; メダル通過チェックスイチ1
dERR_CE	DB	01001110B,00011000B	; 投入メダル通過時間エラー "CJ"
dERR_CE	DB	00000011B	; 投入スイチ1, 2
dERR_CR	DB	01001110B,01001111B	; 投入メダル通過時間エラー "CE"
dERR_CR	DB	00000011B	; 投入スイチ1, 2
dERR_CO	DB	01001110B,00001001B	; 投入メダル運行エラー "Cr"
dERR_CO	DB	10000000B	; 補助収納庫スイチ
dERR_CO	DB	01001110B,00011101B	; 遊技メダル補助収納庫満杯エラー "Co"
dERR_EE	DB	00000000B	; ホットチェックなし
dERR_EE	DB	01001111B,01001111B	; イリガヒエラー "EE"

10

【図 91】



【図 92】



20

30

40

50

【図 9 3】

A	LD	HL,dSDBT_NR	: 一般遊技中抽籤テーブルセット
	LD	B,dLOTCTR_NR	: 一般遊技中抽籤回数セット
	RST	SB GAMEST_00	: 遊技状態フラグ取得処理
	JR	NC,MN_WCAL_01	: RB未作動時ならジャンプ
B	LD	HL,dSDBT_RB	: RB中抽籤テーブルセット
	LD	B,dLOTCTR_RB	: RB中抽籤回数セット
	LDIN	AC,(HL)	: A: 判定データ、抽籤値番号 C: 当たり要求フラグステータス
	MN_WCAL_05:		
	AND	dMSKX_PBAD	
	RLC	C	: 設定別データか?
	JR	NC,MN_WCAL_06	: 違うならジャンプ
	MUL	A,6	: *6
	ADDQ	A,(LOW.wWAVENUM)	: 設定値データ(0~5)加算

【図 9 4】

内部抽籤テーブル		DB 特賞当籤番号+小役当籤番号、抽籤選択テーブルor抽籤係数テーブル	レジスタ、レジスタ
dSDBT_NR	EQU		\$
DB	((0H*25H)+01H, (dRPPTTR01-dRTPRB_SEL)*2+001H		: F.チリリ
DB	((0H*25H)+02H, (dRPPTTR02-dRTPRB_SEL)*2+001H		: F.確りリリ
~			
DB	((0H*25H)+12H, (dRPPTTR17890-dRTPRB_SEL)*2+001H		: F.RT4リリ_132
DB	((0H*25H)+13H, (dRPPTTR17890-dRTPRB_SEL)*2+001H		: F.RT4リリ_2nd
DB	((0H*25H)+14H, (dRPPTTR17890-dRTPRB_SEL)*2+001H		: F.RT4リリ_3rd
DB	((0H*25H)+15H, ((dNMLB00F2123-dPRB_DW_WV)/(06H*2))*2+0C0H		: F.3択へル_1st
DB	((0H*25H)+16H, ((dNMLB00F2123-dPRB_DW_WV)/(06H*2))*2+0C0H		: F.3択へル_2nd
DB	((0H*25H)+17H, ((dNMLB00F2123-dPRB_DW_WV)/(06H*2))*2+0C0H		: F.3択へル_3rd
DB	((0H*25H)+18H, ((dNMLB00F24-dPRB_DW)/2)*2+040H		: F.失速へル
DB	((0H*25H)+19H, ((dNMLB00F25-dPRB_DW)/2)*2+040H		: F.サホ1
DB	((0H*25H)+1AH, (dNMLB00F26-dPRB_DB)*2+000H		: F.サホ2
DB	((0H*25H)+1BH, ((dNMLB00F27-dPRB_DW_WV)/(06H*2))*2+0C0H		: F.弱判
DB	((0H*25H)+1CH, ((dNMLB00F289-dPRB_DB_WV)/(06H)*2+080H		: F.強判1
DB	((0H*25H)+1DH, ((dNMLB00F289-dPRB_DB_WV)/(06H)*2+080H		: F.強判2
DB	((0H*25H)+00H, ((dNMLB12F00-dPRB_DB_WV)/(06H)*2+080H		: F.BB1
DB	((0H*25H)+04H, (dNMLB12F0401-dPRB_DB)*2+000H		: F.BB1+F.リチ目リリA
DB	((0H*25H)+05H, (dNMLB12F057-dPRB_DB)*2+000H		: F.BB1+F.リチ目リリB
DB	((0H*25H)+06H, (dNMLB12F062-dPRB_DB)*2+000H		: F.BB1+F.リチ目リリC
DB	((0H*25H)+07H, (dNMLB12F057-dPRB_DB)*2+000H		: F.BB1+F.リチ目リリD
DB	((0H*25H)+1AH, (dNMLB12F25-dPRB_DB)*2+000H		: F.BB1+F.サホ1
DB	((0H*25H)+1BH, (dNMLB12F26-dPRB_DB)*2+000H		: F.BB1+F.サホ2
~			
DB	((2H*25H)+1DH, (dNMLB12F289-dPRB_DB)*2+000H		: F.BB2+F.強判
DB	((2H*25H)+1EH, (dNMLB12F289-dPRB_DB)*2+000H		: F.BB2+F.強判2
DB	((2H*25H)+1FH, (dNMLB12F0401-dPRB_DB)*2+000H		: F.BB2+F.特殊2
dLOTCTR_NR	EQU	(\$-dSDBT_NR)/2	

10

【図 9 5】

RT状態別抽籤値選択テーブル		dRTPRB_SEL EQU \$
dRPPTTR01:	F.チリリ	
DB	00010000B	
DB	((dRT4B00F01-dPRB_DW)/2)*2+040H	: RT4 作動中
dRPPTTR02:	F.確りリリ	
DB	00010000B	
DB	((dRT4B00F02-dPRB_DW_WV)/(06H*2))*2+0C0H	: RT4 作動中
dRPPTTR03:	F.1確りリリ	
DB	00010000B	
DB	((dRT4B00F03-dPRB_DW)/2)*2+040H	: RT4 作動中
~		
dRPPTTR08:	F.維持リリA	
DB	00110111B	
DB	((dRT0B00F08-dPRB_DW)/2)*2+040H	: RT未作動時
DB	((dRT1B00F08-dPRB_DW)/2)*2+040H	: RT1 作動中
DB	((dRT2B00F08-dPRB_DW)/2)*2+040H	: RT2 作動中
DB	((dRT4B00F08-dPRB_DW_WV)/(06H*2))*2+0C0H	: RT4 作動中
DB	((dRT5B00F08-dPRB_DW)/2)*2+040H	: RT5 作動中
~		
dRPPTTR09:	F.維持リリB	
DB	00010110B	
DB	((dRT124B00F09-dPRB_DW)/2)*2+040H	: RT1 作動中
DB	((dRT124B00F09-dPRB_DW)/2)*2+040H	: RT2 作動中
DB	((dRT124B00F09-dPRB_DW)/2)*2+040H	: RT4 作動中
~		
dRPPTTR1012:	F.維持リリ_1st, F.維持リリ_2nd, F.維持リリ_3rd	
DB	00111000B	
DB	((dRT3B00F1012-dPRB_DW)/2)*2+040H	: RT3 作動中
DB	((dRT4B00F1012-dPRB_DW)/2)*2+040H	: RT4 作動中
DB	((dRT5B00F1012-dPRB_DW)/2)*2+040H	: RT5 作動中
~		
dRPPTTR17890:	F.RT4リリ_123, F.RT4リリ_132, F.RT4リリ_2nd, F.RT4リリ_3rd	
DB	00001000B	
DB	((dRT3B00F17890-dPRB_DW)/2)*2+040H	: RT3 作動中

【図 9 6】

内部抽籤値テーブル選択テーブル		dPRB_SEL:	DB dPRB_DB ~ dPRB_SEL	: [0] 1ハイト内部抽籤値テーブル
		DB dPRB_DW ~ dPRB_SEL		: [1] 2ハイト内部抽籤値テーブル
		DB dPRB_DB_WV ~ dPRB_SEL		: [2] 1ハイト設定別内部抽籤値テーブル
		DB dPRB_DW_WV ~ dPRB_SEL		: [3] 2ハイト設定別内部抽籤値テーブル
内部抽籤値テーブル				
1ハイト内部抽籤値テーブル				
dNMLB12F0401:	DB 5			: F.BB1+F.リチ目リリA, F.BB2+F.リチ目リリB, F.BB1+F.特殊1, F.BB2+F.特殊2
~				
dNMLB00F26:	DB 128			: F.サホ2
dNMLB00F34:	DB 100			: F.弱判2
2ハイト内部抽籤値テーブル				
dPRB_DW:				
dRT0B00F08:	DW 8900			: [RT0] F.維持リリA
dRT1B00F08:	DW 37800			: [RT1] F.維持リリA
dRT124B00F08:	DW 330			: [RT1, RT2, RT4] F.維持リリB
~				
[RT2]				
dRT2B00F08:	DW 1300			: F.維持リリA
dRT12B00F13456:	DW 1800			: F.RT3リリ_1st, F.RT3リリ_213, F.RT3リリ_231, F.RT3リリ_3rd
~				
[RT5]				
dRT5B00F1012:	DW 1640			: F.維持リリ_1st, F.維持リリ_2nd, F.維持リリ_3rd
dRT5B00F13456:	DW 410			: F.RT3リリ_1st, F.RT3リリ_213, F.RT3リリ_231, F.RT3リリ_3rd
dRT5B00F04:	DW 4100			: F.リチ目リリA
dRT5B00F05:	DW 1650			: F.リチ目リリB
~				
dNMLB00F33:	DW 300			: F.弱判1
dNMLB00F35:	DW 7300			: F.弱判3
dNMLB00F36:	DW 57836			: F.弱判4
1ハイト設定別内部抽籤値テーブル				
dPRB_DB_WV:				
dNMLB00F289:	DB 150, 150, 150, 150, 160, 170			: F.強判1, F.強判2
dNMLB12F00:	DB 20, 21, 22, 23, 24, 28			: F.BB1, F.BB2
dNMLB12F27:	DB 5, 7, 8, 11, 13, 15			: F.BB1+F.弱判1, F.BB2+F.弱判2
2ハイト設定別内部抽籤値テーブル				
dPRB_DW_WV:				
dRT4B00F02:	DW 13400, 11000, 13400, 11000, 13400, 11000			: [RT4] F.確りリリ
dRT4B00F08:	DW 11800, 14100, 11680, 13870, 11530, 13720			: [RT4] F.維持リリA
dNMLB00F2123:	DW 5250, 5260, 5230, 5260, 5190, 5210			: F.3択へル_1st, F.3択へル_2nd, F.3択へル_3rd
dNMLB00F27:	DW 870, 900, 930, 960, 990, 1020			: F.弱判

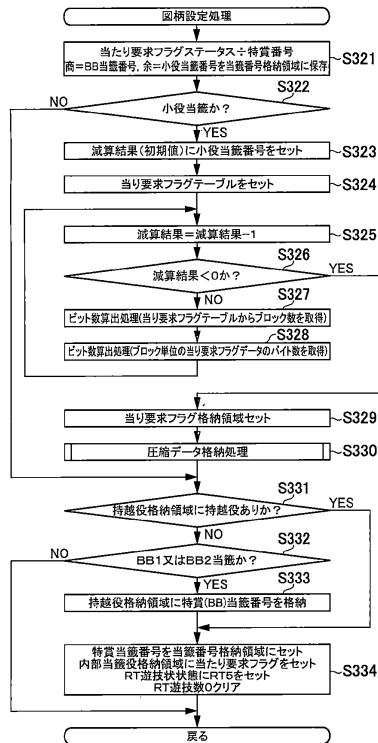
20

30

40

50

【図 97】



【図 98】

当籤番号一覧(定義)			
特賞番号(ボーナス番号)			
cNHT_BB01	EQU	01	: @@ 01:F_BB1
cNHT_BB02	EQU	02	: @@ 02:F_BB2
小役当籤番号			
cNHT_FC01	EQU	01	: @@ 01:F_リリブ
cNHT_FC02	EQU	02	: @@ 02:F_確リリブ
cNHT_FC03	EQU	03	: @@ 03:F_1 確リリブ
cNHT_FC04	EQU	04	: @@ 04:F_リリブ A
cNHT_FC05	EQU	05	: @@ 05:F_リリブ B
cNHT_FC06	EQU	06	: @@ 06:F_リリブ C
cNHT_FC07	EQU	07	: @@ 07:F_リリブ D
cNHT_FC08	EQU	08	: @@ 08:F_維持リブ A
cNHT_FC09	EQU	09	: @@ 09:F_維持リブ B
cNHT_FC10	EQU	10	: @@ 10:F_維持リブ_1st
cNHT_FC11	EQU	11	: @@ 11:F_維持リブ_2nd
cNHT_FC12	EQU	12	: @@ 12:F_維持リブ_3rd
cNHT_FC13	EQU	13	: @@ 13:F_RT3 リブ_1st
cNHT_FC14	EQU	14	: @@ 14:F_RT3 リブ_231
cNHT_FC15	EQU	15	: @@ 15:F_RT3 リブ_231
cNHT_FC16	EQU	16	: @@ 16:F_RT3 リブ_3rd
cNHT_FC17	EQU	17	: @@ 17:F_RT4 リブ_123
cNHT_FC18	EQU	18	: @@ 18:F_RT4 リブ_132
cNHT_FC19	EQU	19	: @@ 19:F_RT4 リブ_2nd
cNHT_FC20	EQU	20	: @@ 20:F_RT4 リブ_3rd
cNHT_FC21	EQU	21	: @@ 21:F_3択へル_1st
cNHT_FC22	EQU	22	: @@ 22:F_3択へル_2nd
cNHT_FC23	EQU	23	: @@ 23:F_3択へル_3rd
cNHT_FC24	EQU	24	: @@ 24:F_共通へル
cNHT_FC25	EQU	25	: @@ 25:F_サホ1
cNHT_FC26	EQU	26	: @@ 26:F_サホ2
cNHT_FC27	EQU	27	: @@ 27:F_弱リ
cNHT_FC28	EQU	28	: @@ 28:F_強リ1
cNHT_FC29	EQU	29	: @@ 29:F_強リ2
cNHT_FC30	EQU	30	: @@ 30:F_特殊1
cNHT_FC31	EQU	31	: @@ 31:F_特殊2
cNHT_FC32	EQU	32	: @@ 32:F_特殊3
cNHT_FC33	EQU	33	: @@ 33:F_RB 役1
cNHT_FC34	EQU	34	: @@ 34:F_RB 役2
cNHT_FC35	EQU	35	: @@ 35:F_RB 役3
cNHT_FC36	EQU	36	: @@ 36:F_RB 役4

10

20

【図 99】

```

== 小役当り要求 フラグ テーブル 検索処理
LD HL,#HTFLGTB ; 小役当り要求フラグテーブル セット
MN_FGST_03:
DEC D
JR Z,MN_FGST_05

CALLF SB_BTNM_00 ; ビット数算出処理
CALLF SB_BTNM_00 ; ビット数算出処理
JR MN_FGST_03

== 小役当り要求 フラグ テーブル 展開処理
MN_FGST_05:
LDQ DE,LOWWAVERIT ; 当り要求フラグ格納領域 セット
CALLF SB_BTNP_00 ; 圧縮データ格納処理

```

【図 100】

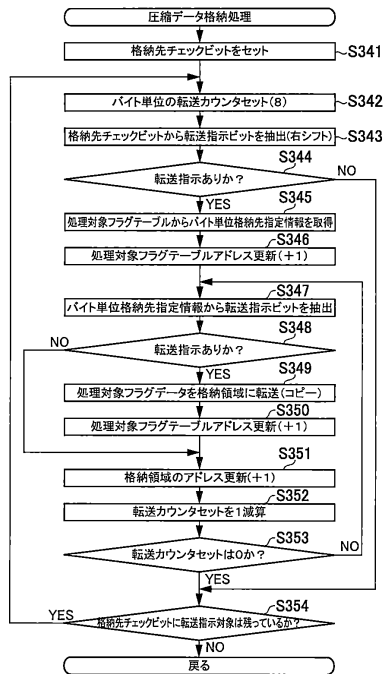
: 当り要求フラグテーブル			
cNHT_FC01	EQU	01	: @@ 01:F_リリブ
cNHT_FC02	EQU	02	: @@ 02:F_確リリブ
cNHT_FC03	EQU	03	: @@ 03:F_1 確リリブ
cNHT_FC04	EQU	04	: @@ 04:F_リリブ A
cNHT_FC05	EQU	05	: @@ 05:F_リリブ B
cNHT_FC06	EQU	06	: @@ 06:F_リリブ C
cNHT_FC07	EQU	07	: @@ 07:F_リリブ D
cNHT_FC08	EQU	08	: @@ 08:F_維持リブ A
cNHT_FC09	EQU	09	: @@ 09:F_維持リブ B
cNHT_FC10	EQU	10	: @@ 10:F_維持リブ_1st
cNHT_FC11	EQU	11	: @@ 11:F_維持リブ_2nd
cNHT_FC12	EQU	12	: @@ 12:F_維持リブ_3rd
cNHT_FC13	EQU	13	: @@ 13:F_RT3 リブ_1st
cNHT_FC14	EQU	14	: @@ 14:F_RT3 リブ_231
cNHT_FC15	EQU	15	: @@ 15:F_RT3 リブ_231
cNHT_FC16	EQU	16	: @@ 16:F_RT3 リブ_3rd
cNHT_FC17	EQU	17	: @@ 17:F_RT4 リブ_123
cNHT_FC18	EQU	18	: @@ 18:F_RT4 リブ_132
cNHT_FC19	EQU	19	: @@ 19:F_RT4 リブ_2nd
cNHT_FC20	EQU	20	: @@ 20:F_RT4 リブ_3rd
cNHT_FC21	EQU	21	: @@ 21:F_3択へル_1st
cNHT_FC22	EQU	22	: @@ 22:F_3択へル_2nd
cNHT_FC23	EQU	23	: @@ 23:F_3択へル_3rd
cNHT_FC24	EQU	24	: @@ 24:F_共通へル
cNHT_FC25	EQU	25	: @@ 25:F_サホ1
cNHT_FC26	EQU	26	: @@ 26:F_サホ2
cNHT_FC27	EQU	27	: @@ 27:F_弱リ
cNHT_FC28	EQU	28	: @@ 28:F_強リ1
cNHT_FC29	EQU	29	: @@ 29:F_強リ2
cNHT_FC30	EQU	30	: @@ 30:F_特殊1
cNHT_FC31	EQU	31	: @@ 31:F_特殊2
cNHT_FC32	EQU	32	: @@ 32:F_特殊3
cNHT_FC33	EQU	33	: @@ 33:F_RB 役1
cNHT_FC34	EQU	34	: @@ 34:F_RB 役2
cNHT_FC35	EQU	35	: @@ 35:F_RB 役3
cNHT_FC36	EQU	36	: @@ 36:F_RB 役4

30

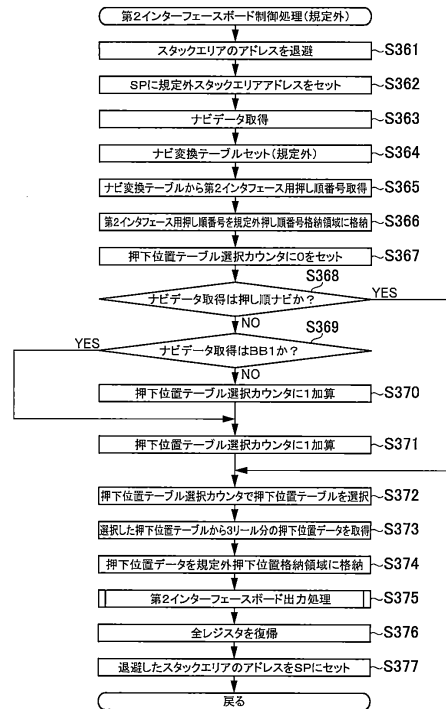
40

50

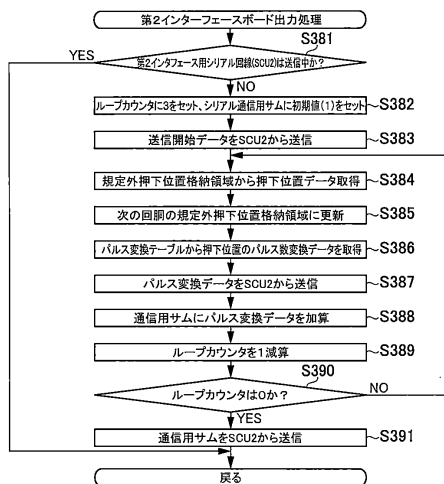
【図101】



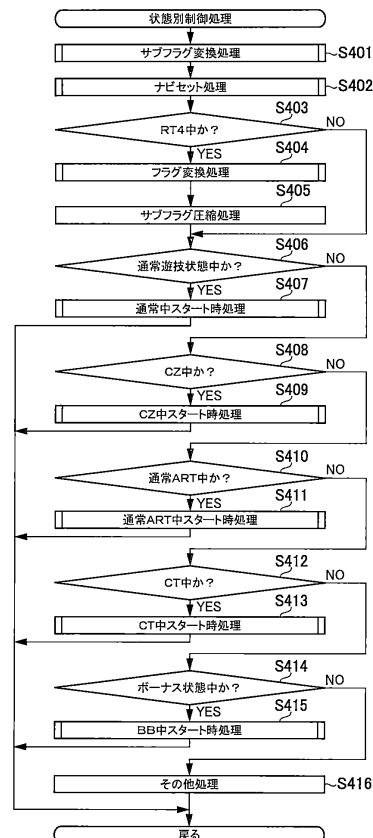
【図102】



【図103】



【図104】



10

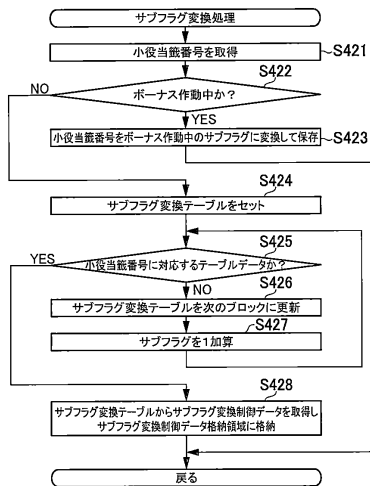
20

30

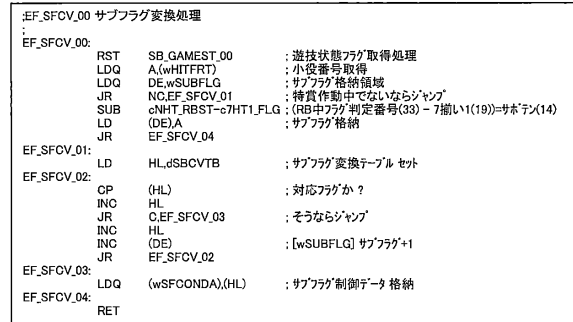
40

50

【図 105】



【図 106】



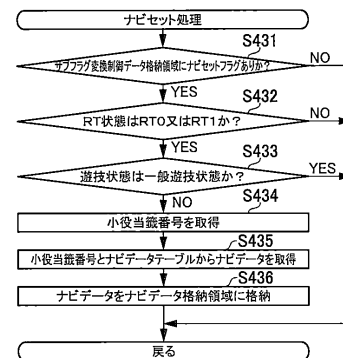
10

20

【図 107】

==	サブフラグ変換テーブル
dSBCVTB	EQU \$
DB	00+1, 00010000B ; 00:ハズレ
DB	01+1, 00000000B ; 01:2連リリブ
DB	02+1, 00000011B ; 02:3連リリブA
DB	03+1, 00000011B ; 03:3連リリブB
DB	04+1, 00000001B ; 04:リーチ目リブ1
DB	05+1, 00000001B ; 05:リーチ目リブ2
DB	06+1, 00000001B ; 06:リーチ目リブ3
DB	07+1, 00000001B ; 07:リーチ目リブ4
DB	09+1, 00000000B ; 08:リプレイ
DB	12+1, 01000000B ; 09:押し順リブ1
DB	16+1, 01000000B ; 10:押し順リブ2
DB	20+1, 01000000B ; 11:押し順リブ3
DB	23+1, 01000100B ; 12:押し順ヘル
DB	24+1, 00000000B ; 13:共通ヘル
DB	26+1, 00000000B ; 14:サボテン
DB	27+1, 00000000B ; 15:弱チゾー
DB	29+1, 00000000B ; 16:強チゾー
DB	31+1, 00011000B ; 17:リーチ目1
DB	32+1, 00011000B ; 18:リーチ目2

【図 108】



30

40

50

【図 1 0 9】

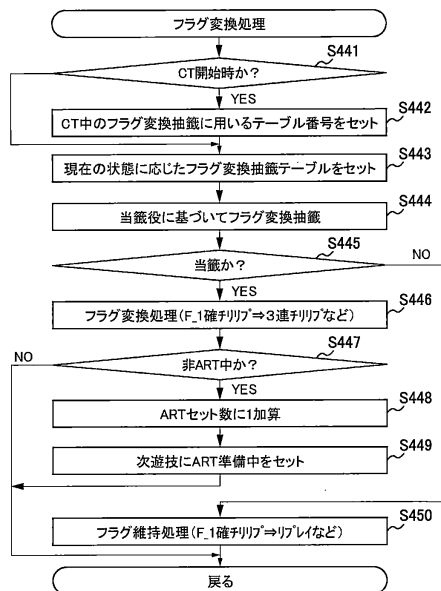
LDQ	A,(wHITFRT)	; 小役番号格納領域(0 ~ 36)
LD	HL,dNAVIPTNTB-10	; ナビテーブル小役番号10 ~ 23)
RST	SB,ADRA,00	
LDQ	(wNAVIPTN),A	; ナビをセット

【図 1 1 0】

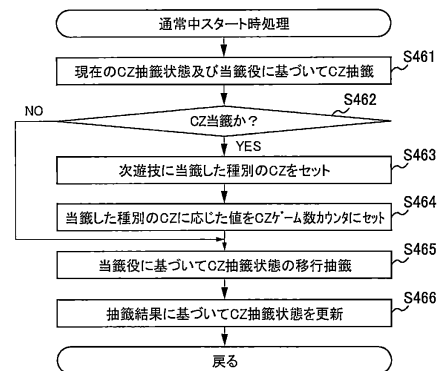
dNAVIPTNTB	EQU	\$
DB	7	; 10.F.維持リブ_1st
DB	8	; 11.F.維持リブ_2nd
DB	9	; 12.F.維持リブ_3rd
DB	1	; 13.F.RT3リブ_1st
DB	3	; 14.F.RT3リブ_213
DB	4	; 15.F.RT3リブ_231
DB	6	; 16.F.RT3リブ_3rd
DB	1	; 17.F.RT4リブ_123
DB	2	; 18.F.RT4リブ_132
DB	3	; 19.F.RT4リブ_2nd
DB	6	; 20.F.RT4リブ_3rd
DB	7	; 21.F.3択ベル_1st
DB	8	; 22.F.3択ベル_2nd
DB	9	; 23.F.3択ベル_3rd

10

【図 1 1 1】



【図 1 1 2】



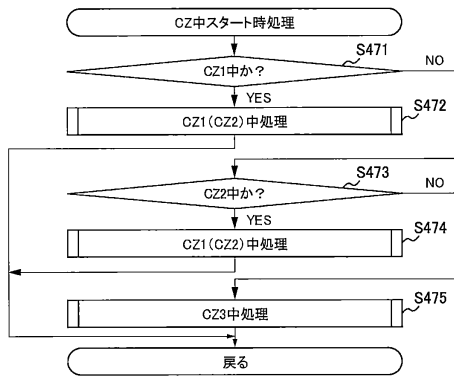
20

30

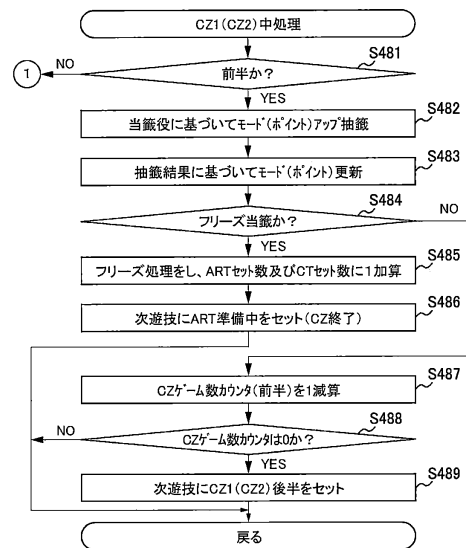
40

50

【図 1 1 3】



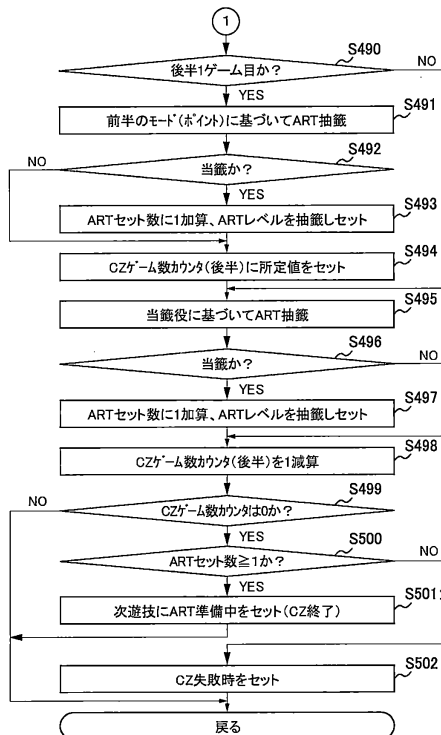
【図 1 1 4】



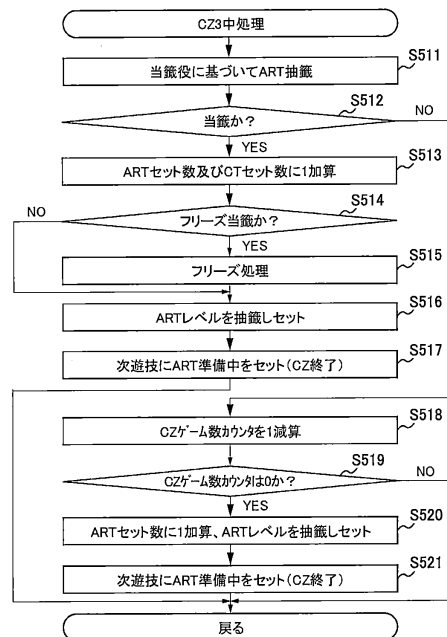
10

20

【図 1 1 5】



【図 1 1 6】

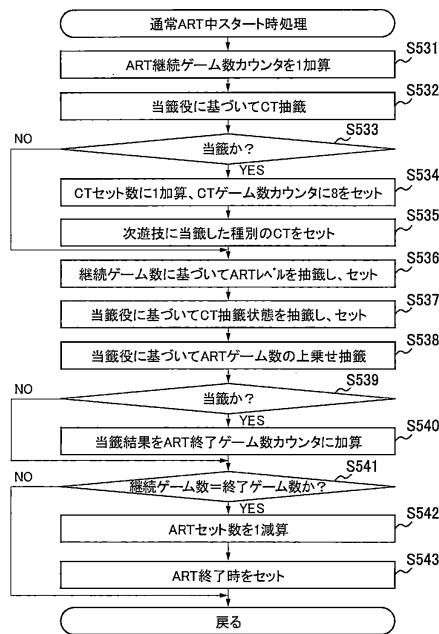


30

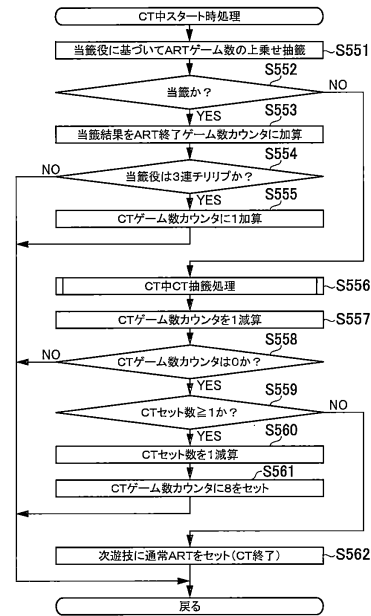
40

50

【図 1 1 7】



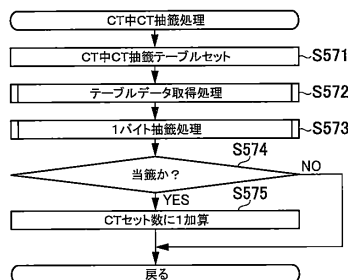
【図 1 1 8】



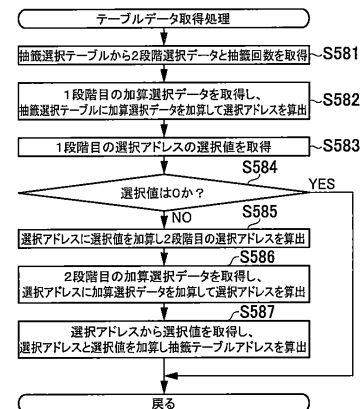
10

20

【図 1 1 9】



【図 1 2 0】



30

40

50

【図 1 2 1】

```

;:= テーブルデータ取得処理
ESB_GTD2_00:
    LDIN    DE,(HL)    ; 加算RAM7アドレス取得 E: 1回目 D: 2回目
    LD      B,(HL)     ; 抽籤回数取得
    INC     HL
    LDQ     A,(E)
    RST     SB_ADRA,00
    RT      Z,A
    ADDWB   HLA
ESB_GTD2_20:
    LDQ     A,(D)
    RST     SB_ADRA,00
    RST     SB_ADRA,00
    RET
  
```

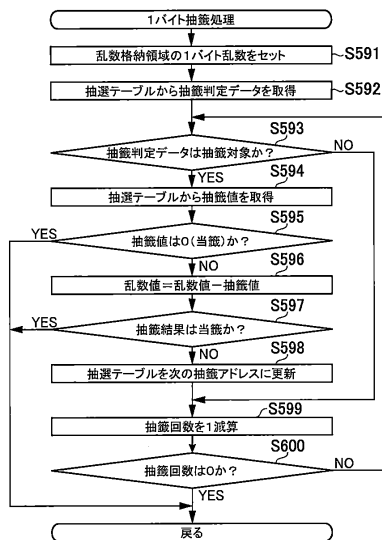
【図 1 2 2】

```

;:= CT中CT当籤抽籤テーブル
cABS_HIT EQU 0
dCTCTSTTB DB $ ; 当籤役別テーブル選択相対テーブル
DB .LOW.wSUBFLGD ; サブフラグD
DB .LOW.wCT_KIND ; CT種別
DB 2 ; 抽籤回数
DB 0,0,0,0,0 ; ハズレ, サブテン, 弱チェリー, 強チェリー, 確定役, 3連チリリ*
DB dCTCTS_RER-$ ; リーチ目リリ*
dCTCTS_RER DB dNMCTCTS_RER-$ ; 通常 リーチ目リリ*
DB dSPCTCTS_RER-$ ; 高確 リーチ目リリ*
dNMCTCTS_RER DB 00000011B ; 通常 リーチ目リリ*
DB 1 ; 判定ヒット
DB 255 ; 高確CT
; @@ ; 通常CT
dSPCTCTS_RER DB 00000001B ; ハズレ
DB cABS_HIT ; 高確 リーチ目リリ*
; @@ ; 判定ヒット
; @@ ; 高確CT
; @@ ; 通常CT
; @@ ; ハズレ
  
```

10

【図 1 2 3】



【図 1 2 4】

```

; ESB_LOTP_00 1バイト抽籤処理
ESB_LOTP_00:
    LDQ     DE,wRANDOMP ; 乱数値P
ESB_LOTP1_00:
    LD      C,(HL)     ; 抽籤判定データ取得
    INC     HL
ESB_LOTP1_01:
    SRL     C
    JR      NC,ESB_LOTP1_02 ; 抽籤対象でなければジャンプ
    JR      Z,(HL),ESB_LOTP1_03 ; 0なら当籤
;
    LD      A,(DE)
    SUB     (HL)
    LD      (DE),A ; 抽籤実行
    JR      C,ESB_LOTP1_03 ; 当籤ならジャンプ
;
    INC     HL ; 次のデータへ
ESB_LOTP1_02:
    DJNZ    ESB_LOTP1_01
ESB_LOTP1_03:
    LD      AB
    RET
  
```

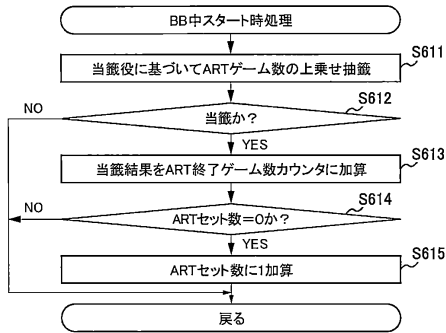
20

30

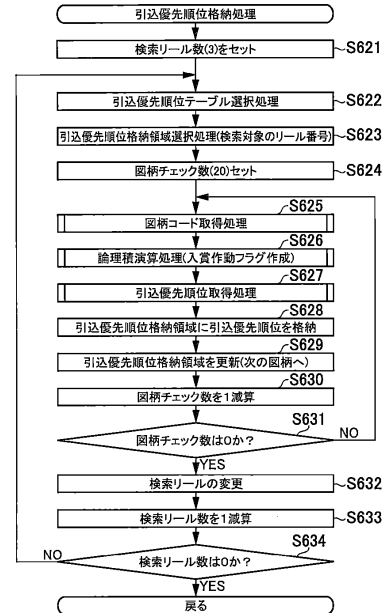
40

50

【図 1 2 5】



【図 1 2 6】



10

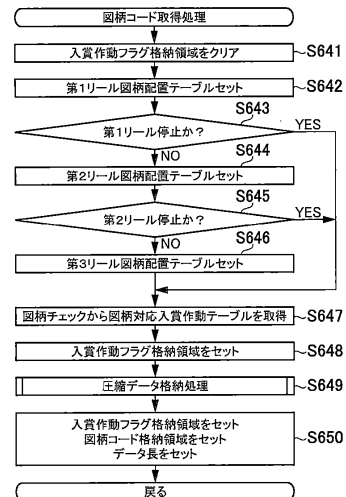
20

【図 1 2 7】

```

LD   A,cREEL_ZCNT
SUB   B           ; 図柄コード取得処理
CALLF _STSB_RLTB_00
      ; @@ DE = [WHITFLAG] 入賞作動フラグ格納領域
      ; @@ HL = [WXARAY] 図柄コード格納領域
      ; @@ B = cARYSIZE
;
;== 1ラインに対する入賞作動フラグ作成
;
EX   DE,HL
CALLF SB_DAND_00  ; 論理積演算処理
  
```

【図 1 2 8】



30

40

50

【図 1 2 9】

_STSB_RLTB_00:			
CALLF	SB_HTFLGCR_00	; 入賞作動フラグクリア	
:			
LD	HL,dREELTB1	; 第一回胴図柄記憶テーブル	
JBIT	NZ,0,C,_STSB_RLTB_01	; 第一回胴ならジャンプ	
LD	HL,dREELTB2	; 第二回胴図柄記憶テーブル	
JBIT	NZ,1,C,_STSB_RLTB_01	; 第二回胴ならジャンプ	
LD	HL,dREELTB3	; 第三回胴図柄記憶テーブル	
_STSB_RLTB_01:			
RST	SB_ADRA_00	; 図柄コード相対アドレス取得	
ADDWB	HL,A	; 図柄コードアドレス取得	
:			
LDQ	DE,wHITFLAG		
CALLF	SB_BTEP_00	; 入賞作動フラグデータ解凍格納処理	
:			
LDQ	DE,wHITFLAG	; 入賞作動フラグ格納領域セット	
LDQ	HL,wXARAY	; 図柄コード格納領域セット	
LD	B,cARYSIZE	; データ長セット	
RET			

【図 1 3 0】

;== 第一回胴 図柄配置テーブル			
dREELTB1	EQU	\$	
DB	dR1_REP_-\$		; 図柄位置 0
DB	dR1_BEL_-\$		; 1
DB	dR1_SVN1_-\$		; 2
~			
DB	dR1_REP_-\$		; 18
DB	dR1_CCT2_-\$		; 19
;== 図柄対応入賞作動テーブル(第一回胴)			
dR1_SVN1:			; 白7 (第一回胴)
DB	01011100B		; 格納領域 +2, +3, +4, +6
DB	00010000B 00000100B		; フラグ
DB	00100000B 00001000B		
DB	10000000B 01000000B 00100000B		
DB	00100000B 00010000B		; 00001000B
DB	00000100B		; 格納領域 +10
;C,BB1			
DB	10000000B		
dR1_SVN2:			
DB	00100101B		; 青7 (第一回胴)
-以下省略-			

10

【図 1 3 1】

;== 第二回胴 図柄配置テーブル			
dREELTB2	EQU	\$	
DB	dR2_BEL_-\$		; 図柄位置 0
DB	dR2_REP_-\$		; 1
DB	dR2_SVN1_-\$		; 2
~			
DB	dR2_REP_-\$		; 18
DB	dR2_CCT2_-\$		; 19
;== 図柄対応入賞作動テーブル(第二回胴)			
dR2_SVN1:			; 白7 (第二回胴)
DB	011111001B		; 格納領域 +0, +3, +4, +5, +6
DB	00010000B		
DB	01000000B 00000001B		
DB	10000000B 00100000B 00000100B		
DB	10000000B		
DB	00010000B 00000100B		
DB	00000110B		; 格納領域 +8, +10
DB	10000000B 00100000B		
DB	10000000B		
dR2_SVN2:			
DB	00011101B		; 青7 (第二回胴)
-以下省略-			

【図 1 3 2】

;== 第三回胴 図柄配置テーブル			
dREELTB3	EQU	\$	
DB	dR3_REP_-\$		; 図柄位置 0
DB	dR3_BEL_-\$		; 1
DB	dR3_SVN1_-\$		; 2
~			
DB	dR3_CHI2_-\$		; 18
DB	dR3_CCT1_-\$		; 19
;== 図柄対応入賞作動テーブル(第三回胴)			
dR3_SVN1:			; 白7 (第三回胴)
DB	011111001B		; 格納領域 +0, +3, +4, +5, +6
DB	00001000B		
DB	00100000B 00010000B 00000100B		
DB	00010000B		
DB	10000000B 00100000B 00010000B		
DB	00100000B 00000001B		
DB	00000101B		; 格納領域 +8, +10
DB	00001000B 00000100B		
DB	10000000B		
dR3_SVN2:			
DB	011111001B		; 青7 (第三回胴)
-以下省略-			

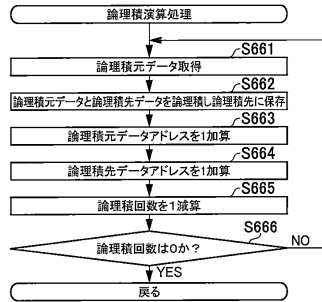
20

30

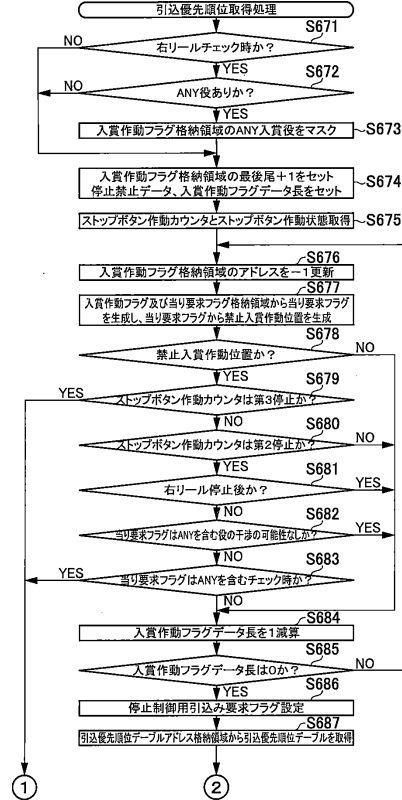
40

50

【図 1 3 3】



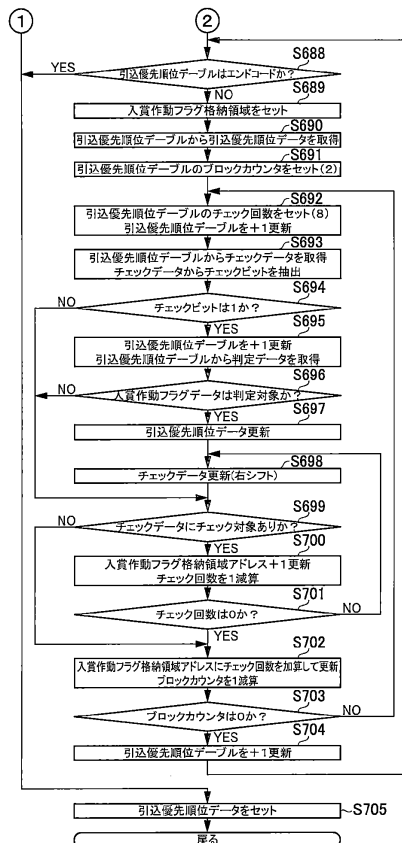
【図 1 3 4】



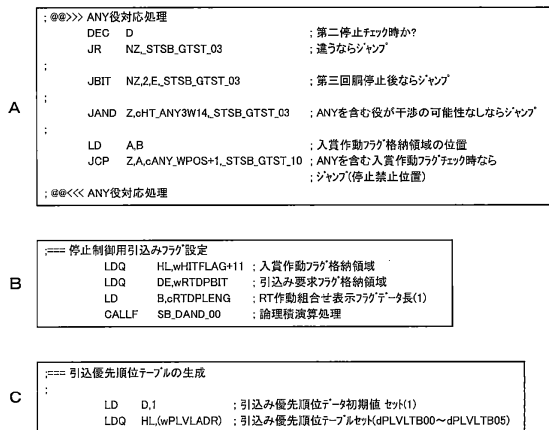
10

20

【図 1 3 5】



【図 1 3 6】



30

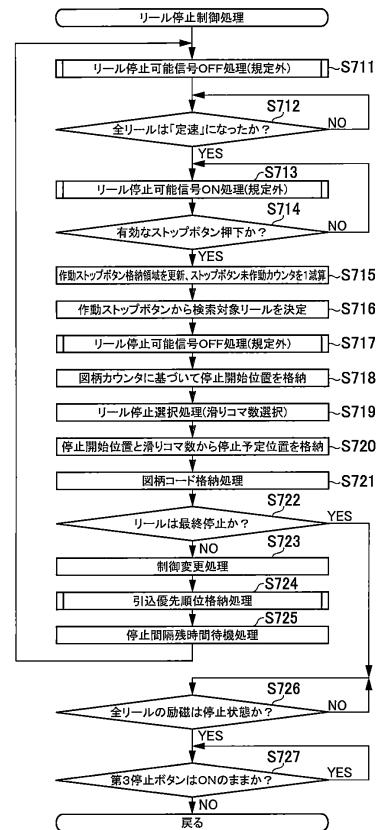
40

50

【図 137】

; 引込優先順位テーブル			
dPLVLTB00	EQU	\$	
DB	03EH		; 引込優先順位データ
DB	10001000B		; チェックデータ
DB	080H, 003H		; フラグ判定データ
DB	00000001B		; チェックデータ
DB	040H		; フラグ判定データ
;			
DB	01EH		; 引込優先順位データ
DB	10001011B		; チェックデータ
DB	048H, 008H, 004H, 034H		; フラグ判定データ
DB	00000000B		; チェックデータ
;			
DB	002H		; 引込優先順位データ
DB	00000000B		; チェックデータ
DB	00000100B		; チェックデータ
DB	000H		; フラグ判定データ
;			
DB	000H		; エントリ
dPLVLTB01	EQU	\$	
DB	01EH		; 引込優先順位データ
;			
dPLVLTB05	EQU	\$	
DB	00EH		; 引込優先順位データ
DB	00010000B		; チェックデータ
DB	004H		; フラグ判定データ
DB	00000000B		; チェックデータ
;			
DB	008H		; 引込優先順位データ
DB	00001000B		; チェックデータ
DB	080H		; フラグ判定データ
DB	00000000B		; チェックデータ
;			
DB	002H		; 引込優先順位データ
DB	00010000B		; チェックデータ
DB	002H		; フラグ判定データ
DB	00000000B		; チェックデータ
;			
DB	000H		; エントリ

【図 138】



10

20

【図 139】

; リールストップ可能信号OFF処理(領域外モジュール)			
LD	HL, OT, SLEDOF_00		
; 領域外呼び出し処理			
CALLF	SB, LOGALL_00		
CALLF	SB, FXCK_00		
JR	CMN_STOP_00		
; 定数以外でないならジャンプ			
LDQ	HL, wSLAMPBIT		
LDQ	A, wBUF+4, LOW+4, PL, STPBTM		
AND	(HL)		
JR	NZ, MN_STOP_02		
; ストップ可能信号ON処理(領域外モジュール)			
LD	HL, OT, SLEDOF_00		
JR	MN_STOP_01		
; ストップ開始時の初期化(同時押下時は左の画面を最優先)			
LDQ	IX, wR1, OTRL - (wR2, OTRL - wR1, OTRL)		
LDQ	C, 080H		
; 第一画面 制御データ格納領域セット			
; ストップ可能信号ON処理(領域外モジュール)			

【図 140】

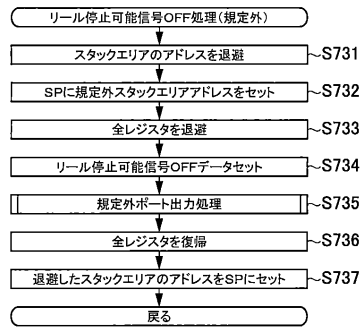
MN_STOP_05:			
; 制御終了のチェック			
LDQ	A, (LOW, wR1, TIM)		
ORQ	(LOW, wR2, TIM)		
ORQ	(LOW, wR3, TIM)		
JR	NZ, MN_STOP_05		
; 全画面の励磁タイマー合成			
; 停止終了でないならジャンプ			

30

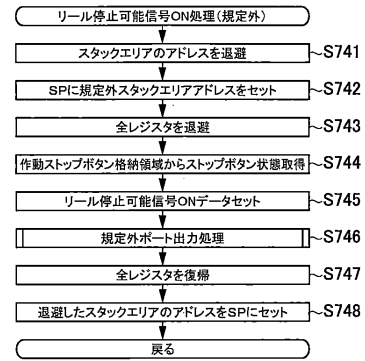
40

50

【図 1 4 1】

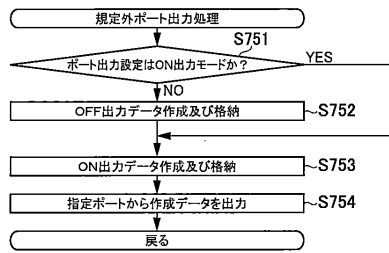


【図 1 4 2】

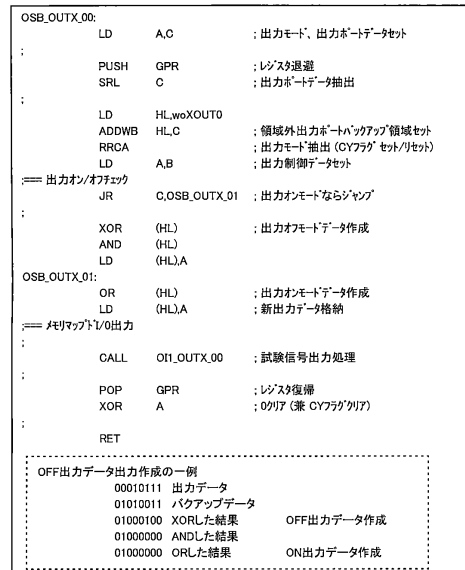


10

【図 1 4 3】



【図 1 4 4】



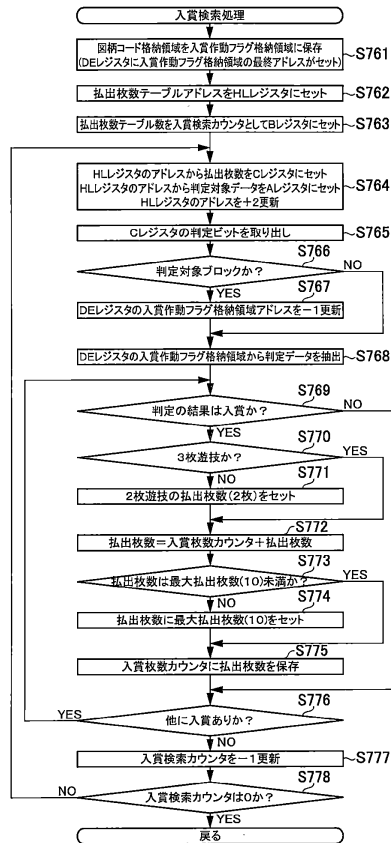
20

30

40

50

【図 145】



【図 146】

MN_CKLN_00:	LDQ	HL,LOW.wXARAY	: 図柄コード格納領域セット
	LDQ	DE,LOW.wHTFLAG	: 入賞作動フラグ格納領域セット
	LD	B,12	
	LDIRS		: 図柄コードを入賞作動フラグに転送
:			
	LD	HL,dPAYNUMTB	: 払出枚数データテーブル
	LD	B,5	: 払出枚数データテーブルサイズ
MN_CKLN_01:	LDIN	AC,(HL)	: C = 払出枚数, A = 判定データ
	SRL	C	: ハ'作要更タイミングか?
	JR	NC,MN_CKLN_02	: 違うならジャンプ
	DEC	DE	
MN_CKLN_02:	AND	(DE)	: 判定ビットを抽出
MN_CKLN_03:	JSLAA	NC,MN_CKLN_06	: 入賞なしならジャンプ
== 入賞 9/16 加算 (重複入賞 対応)	PUSH	AF	
	LDQ	A,(LOW.wMEDLCCTR)	: 9/16カウンタ取得
	JOP	Z,A,3,MN_CKLN_04	: 3 枚遊技時ならジャンプ
	LD	C,2	: 2枚遊技の払い出し枚数セット
MN_CKLN_04:	LDQ	A,(LOW.wHTCCTR)	: 入賞枚数カウンタ取得
	ADD	A,C	: 払い出し枚数加算
:			
	JOP	C,A,9,MN_CKLN_05	: 獲得枚数の最大値未達ならジャンプ
	LD	A,9	: 獲得枚数の最大値セット
MN_CKLN_05:	LDQ	(LOW.wHTCCTR),A	: 入賞枚数カウンタ保存
	POP	AF	
MN_CKLN_06:	JR	NZ,MN_CKLN_03	: 他に入賞あきらならジャンプ
	DJNZ	MN_CKLN_01	
	RET		

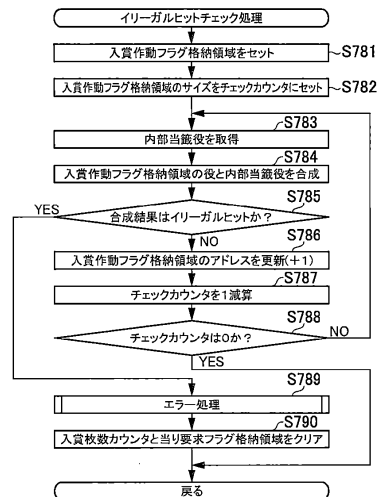
10

20

【図 147】

抽出枚数データテーブル	
dPAYNUMTB	\$
EQU	
DB	9+2 + 0, 111111110B : @@ F,RB役1~4
DB	9+2 + 0, 111111111B : @@ F,3R&A_1st, F,3R&A_2nd, F,3R&A_3rd, F,共通&A
DB	1+2 + 0, 111111110B : @@ F,サナ1, F,サナ2
DB	3+2 + 0, 111111000B : @@ F,サナ1, F,サナ2
DB	2+2 + 0, 111110000B : @@ 弱チリー, 強チリー, F,持珠1~3
cPAYNUMTBNUM	\$(dPAYNUMTB)/2

【図 148】



30

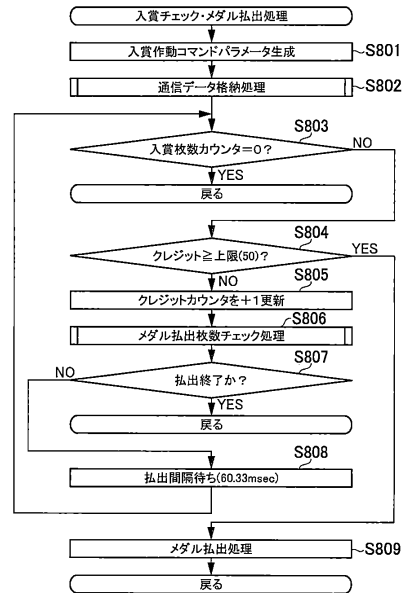
40

50

【図 149】

```
;;イーガルチェック処理
MN_ILCK_00:
    LDQ    HL,LOW.wHITFLAG      ;入賞作動フラグ格納領域セット
    LD      B,wWV LENG         ;入賞作動フラグデータ長セット
;
MN_ILCK_01:
    LD      A,(HL+(wWAVEBIT-wHITFLAG)) ; 当り要求フラグ取得
    XOR     (HL)                ; イーガルヒットはあるか?
    AND     (HL)                ; 入賞作動フラグと合成
    JR      NZ,MN_ILCK_02       ; そうならジャンプ
;
    INC     HL
    DJNZ    MN_ILCK_01
    RET
;
;;イーガルヒットエラーの表示
MN_ILCK_02:
    LD      E,7                 ; イーガルヒットエラーコードセット
    CALLF   SB_ERRS_00         ; エラー処理
```

【図 150】

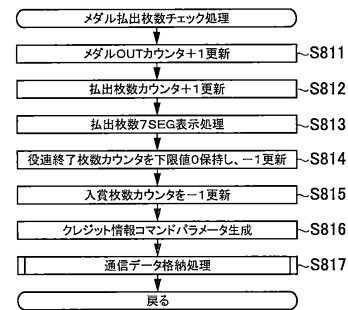


10

【図 151】

```
MN_HOPS_02:
;;払出の制御
    CALLF   SB_CDMX_00          ; クレジットは上限に達しているか?
    LD      A,SBX_PAY_HIT       ; 入賞時入賞フラグセット
    JR      NC,SB_MPAY_00       ; そうならメダル払出処理へ
;
    INC     (HL)                ; [wCREDIT] クレジットカウンタ加算更新
;
    CALLF   SB_MPOK_00          ; メダル払出枚数チェック処理
    RET     Z                   ; 払出終了ならリターン
;
    LD      BC,STM_PAYC         ; 払出間隔待ち
    RST     SB_WIBC_00          ; 払出間隔待ち
;
    JR      MN_HOPS_02         ; 払出処理継続
```

【図 152】



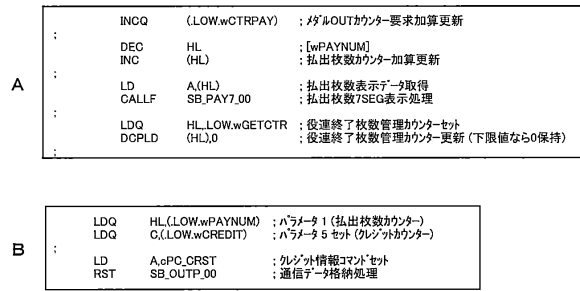
20

30

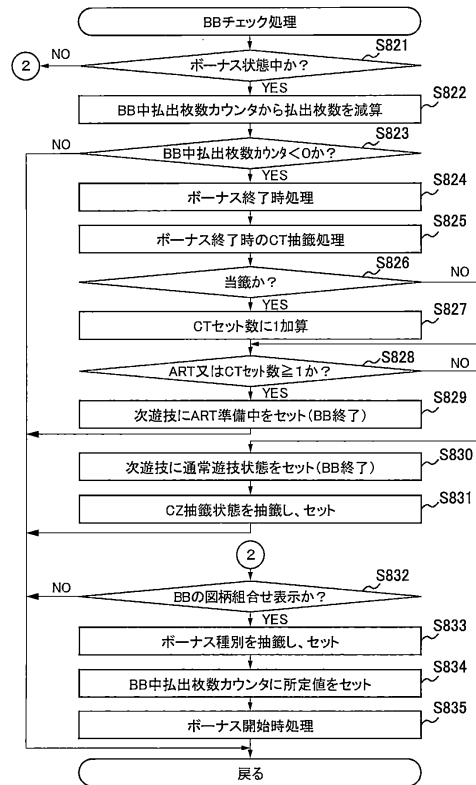
40

50

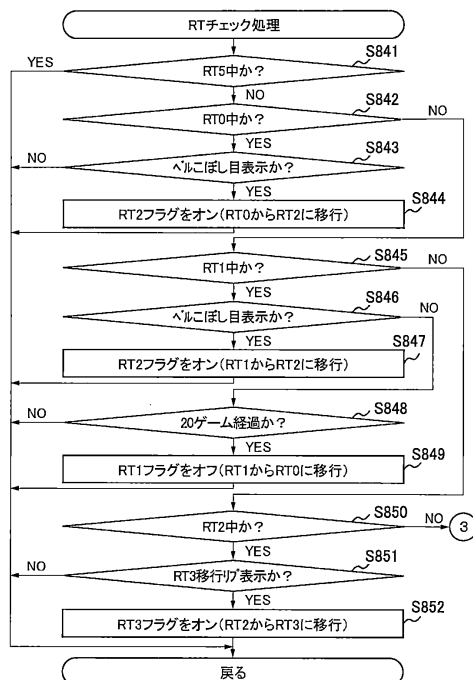
【図 153】



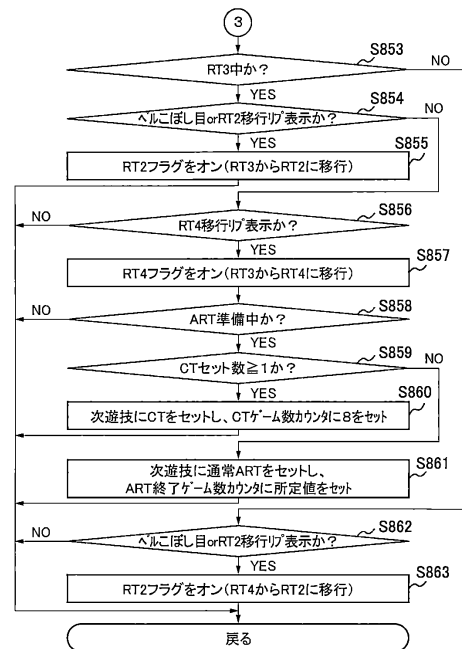
【図 154】



【図 155】



【図 156】



10

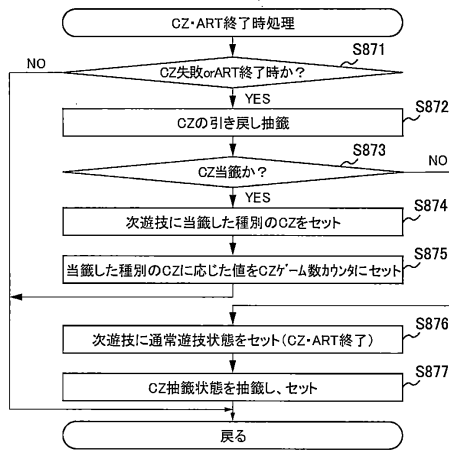
20

30

40

50

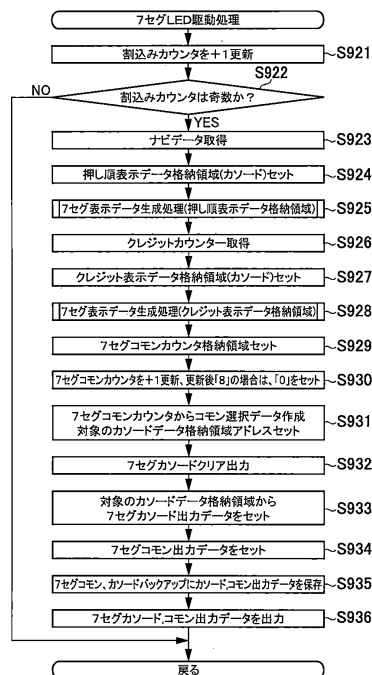
【図 157】



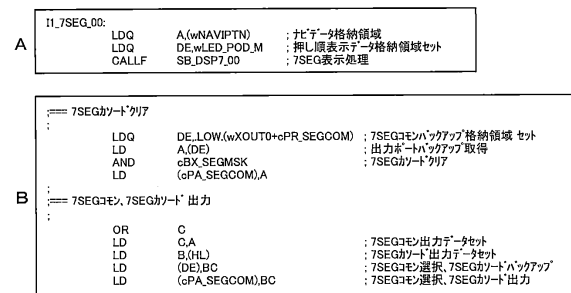
【図 158】



【図 159】



【図 160】



10

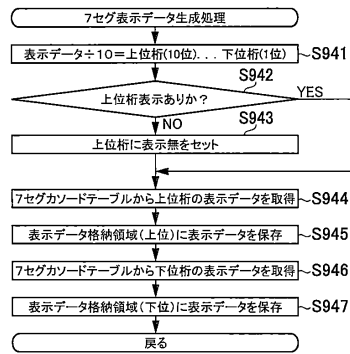
20

30

40

50

【図 1 6 1】



【図 1 6 2】

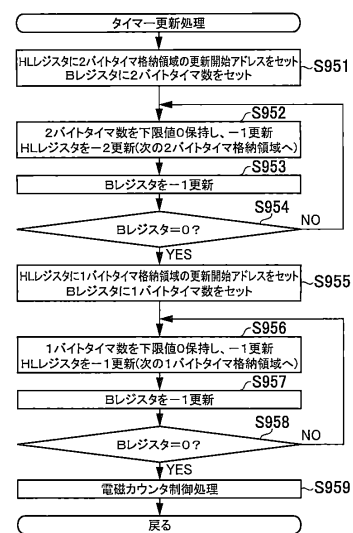
```
SB_DSP7_00:
    DIV    C,A,10      ; A = (商) 上位表示データ
                    ; C = (余) 下位表示データ
    JT     NZ,A,SB_DSP7_01 ; 上位表示ありならジャンプ
:
    LD     A,10        ; 上位に無表示データ[10] セット
SB_DSP7_01:
    CALLF  SB_DCD7_00  ; 7SEGデコード処理(上位)
    LD     A,C          ; 7SEGデコードデータ(下位) セット
SB_DCD7_00:
    LD     HL,dSEGPARM ; 7SEGカソードデータテーブルセット
    RST    SB_ADRA_00
    LD     (DE),A        ; 表示データ格納
    INC    DE            ; 次の格納領域へ
    INC    RET
```

10

【図 1 6 3】

dSEGPARM	EQU	\$
DB	01111110B	: "0" 表示
DB	00110000B	: "1" 表示
DB	01101101B	: "2" 表示
DB	01111001B	: "3" 表示
DB	00110011B	: "4" 表示
DB	01011011B	: "5" 表示
DB	01011111B	: "6" 表示
DB	01110000B	: "7" 表示
DB	01111111B	: "8" 表示
DB	01111011B	: "9" 表示
DB	00000000B	: 表示なしデータ

【図 1 6 4】



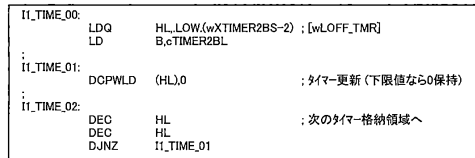
20

30

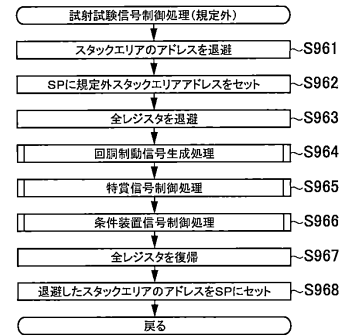
40

50

【図 165】



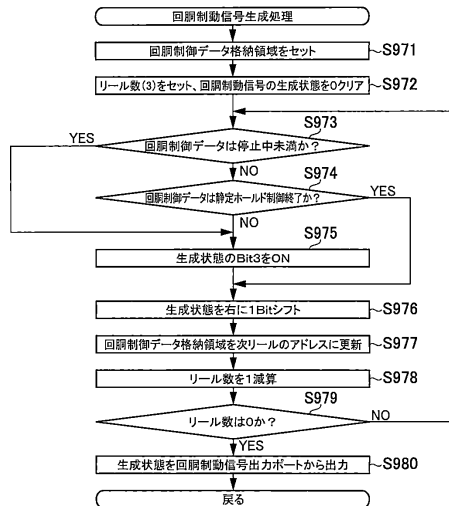
【図 166】



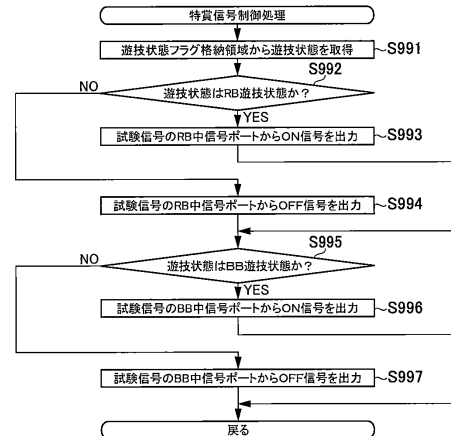
10

20

【図 167】



【図 168】

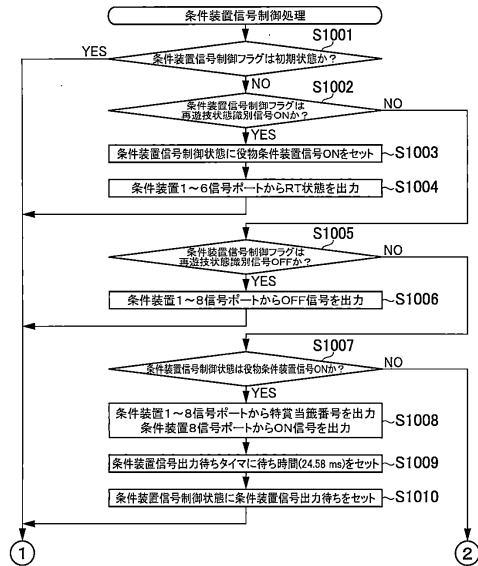


30

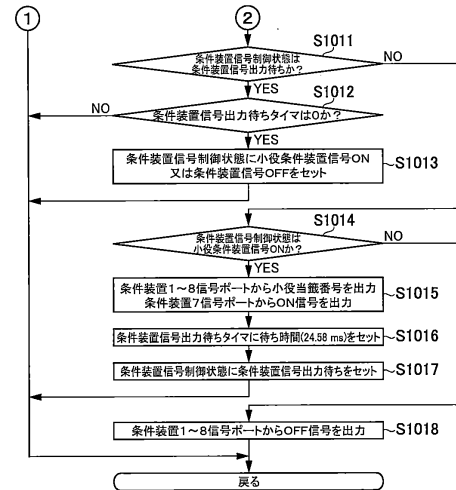
40

50

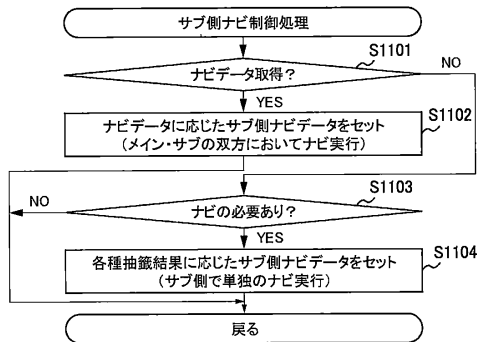
【図 1 6 9】



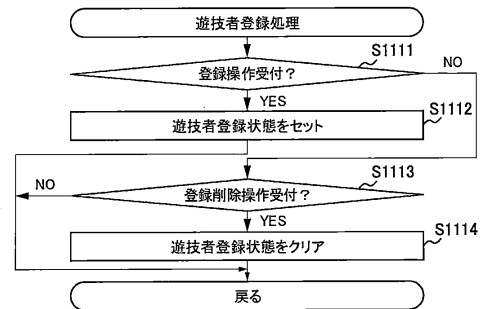
【図 1 7 0】



【図 1 7 1】



【図 1 7 2】



10

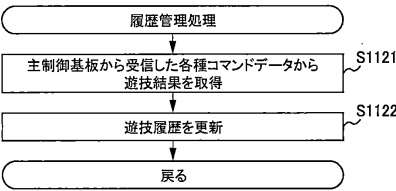
20

30

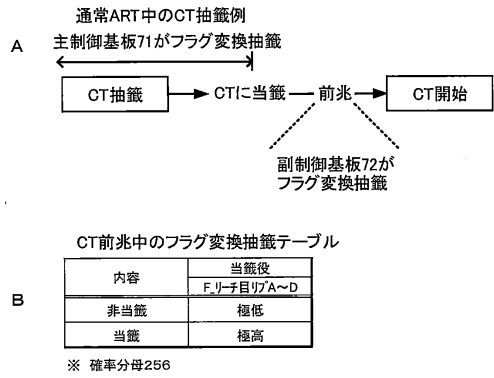
40

50

【図 1 7 3】



【図 1 7 4】



10

20

【図 1 7 5】

A 内部当籤役と停止図柄組合せ(略称)との対応関係

当籤役	第1停止操作		
	左(1st)	中(2nd)	右(3rd)
F.確リプレイ	チリリブ(3連)	リプレイ	チリリブ(2連)
F.1確リプレイ	チリリブ(2連)	リプレイ	チリリブ(3連)

B 通常ART中のフラグ変換抽籤とサブ側のナビとの関係

当籤役	フラグ変換抽籤		
	1段階目非当籤	1段階目当籤 2段階目非当籤	1段階目当籤 2段階目当籤
F.確リプレイ	リプレイナビ (第1停止操作 中)	チリリブ揃りナビ (第1停止操作 右)	チリリブ揃いナビ (第1停止操作 左)
F.1確リプレイ	リプレイナビ (第1停止操作 中)	チリリブ揃りナビ (第1停止操作 左)	チリリブ揃いナビ (第1停止操作 右)

30

【図 1 7 6】

ナビデータの対応関係

RT5中(BB1フラグ間)

内容	メイン側ナビ													
	1	2	3	...	10	11	12	13	14	...				
	左	中	右	...	白7 F.特殊1 F.特殊3	白7 F.特殊1 F.特殊3	白7	左	中	右	...			
白7ナビ														
ベルナビ							F.3取へル.1st	F.3取へル.2nd	F.3取へル.3rd					

A

RT5中(BB2フラグ間)

内容	メイン側ナビ													
	1	2	3	...	10	11	12	13	14	...				
	左	中	右	...	白7 F.特殊1 F.特殊3	白7 F.特殊1 F.特殊3	白7	左	中	右	...			
青7ナビ														
ベルナビ							F.3取へル.1st	F.3取へル.2nd	F.3取へル.3rd					

B

40

50

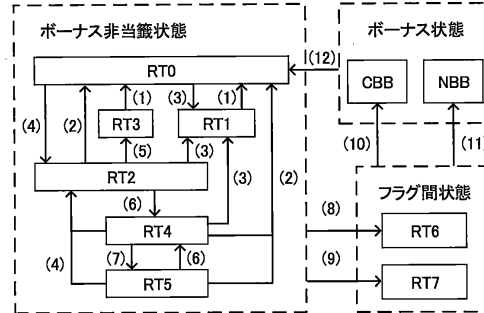
【図 177】

押し順とロック状態との関係							
	当役後	押し順					
A		左-中-右	左-右-中	中-左-右	中-右-左	右-左-中	右-中-左
F.維持リプA	0⇒1	維持		維持			1⇒0
F.維持リプB		維持		0⇒1	維持		維持
	***	***	***	***	***	***	***

※ ロック状態0:ロックし難い状態
ロック状態1:ロックし易い状態

ロック状態を移行するための報知			
ロックによる利益	報知抽籤	報知手段	
		指示モニタ	表示装置
あり	メイン側で抽籤	○ 報知	○ 報知
なし	サブ側で抽籤	× 報知せず	○ 報知

【圖 178】



移行条件	内容	RTの種類	リプレイ確率	ゲーム数
(1)	規定ゲーム数消化	RT0	低確率	∞
(2)	RT0移行図柄が表示	RT1	低確率	10G
(3)	RT1移行図柄が表示	RT2	中確率	∞
(4)	RT2移行図柄が表示	RT3	中確率	20G
(5)	RT3移行図柄が表示	RT4	高確率	∞
(6)	RT4移行図柄が表示	RT5	高確率	∞
(7)	RT5移行図柄が表示	RT6	中確率	∞
(8)	F_クラウンBB、又は、F_青BBに当籤	RT7	中確率	∞
(9)	F_赤BBに当籤			
(10)	クラウンBBの図柄が表示			
(11)	BBの図柄が表示			
(12)	ボーナス終了			

【図 179】

左リール		中リール		右リール	
図柄位置	図柄	図柄位置	図柄	図柄位置	図柄
5	リプレイ	5	ベル2	5	チェリー
4	ベル1	4	スイカ2	4	スイカ1
3	スイカ1	3	青7	3	青7
2	リプレイ	2	リプレイ	2	ベル2
1	ベル1	1	チェリー	1	リプレイ
0	スイカ1	0	ベル2	0	チェリー
20	リプレイ	20	スイカ2	20	スイカ1
19	BAR	19	BAR	19	BAR
18	スイカ2	18	リプレイ	18	ベル3
17	ベル3	17	チェリー	17	リプレイ
16	赤7	16	ベル3	16	赤7
15	リプレイ	15	赤7	15	スイカ2
14	BAR	14	BAR	14	BAR
13	スイカ2	13	リプレイ	13	ベル3
12	ベル2	12	チェリー	12	リプレイ
11	青7	11	ベル1	11	チェリー
10	リプレイ	10	リプレイ	10	ベル1
9	チェリー	9	チェリー	9	リプレイ
8	リプレイ	8	ベル1	8	チェリー
7	ベル1	7	リプレイ	7	ベル1
6	スイカ1	6	チェリー	6	リプレイ

※図柄コード

図柄コード	内容	
	図柄	データ
1	赤7	00000001
2	青7	00000010
3	BAR	00000011
4	リプレイ	00000100
5	ベル1	00000101
6	ベル2	00000110
7	ベル3	00000111
8	スイカ1	00001000
9	スイカ2	00001001
10	チェリー	00001011

【図 180】

[illegible]

【図 181】

内部抽籤テーブル

略称	抽籤値 クワンBB フラグ間用(RT6)	略称	抽籤値 赤BB フラグ間用(RT7)
F.クワンBB	29667	F.赤BB	25783
F.クワンBB+F.通常リゾ	164	F.赤BB+F.通常リゾ	492
F.クワンBB+F.維持リゾ	5050	F.赤BB+F.維持リゾ	6264
F.クワンBB+F.弱チリゾ	874	F.赤BB+F.弱チリゾ	3688
F.クワンBB+F.強チリゾ	66	F.赤BB+F.強チリゾ	66
F.クワンBB+F.チンズリゾ	388	F.赤BB+F.チンズリゾ	388
F.クワンBB+F.リーチ目リゾ	8	F.赤BB+F.リーチ目リゾ	8
F.クワンBB+F.フレイグリゾ	328	F.赤BB+F.フレイグリゾ	164
F.クワンBB+F.リゾA	0	F.赤BB+F.リゾA	0
F.クワンBB+F.リゾB	0	F.赤BB+F.リゾB	0
F.クワンBB+F.バリリゾ	0	F.赤BB+F.バリリゾ	0
F.クワンBB+F.6択維持リゾ123	1480	F.赤BB+F.6択維持リゾ123	1480
F.クワンBB+F.6択維持リゾ132	1480	F.赤BB+F.6択維持リゾ132	1480
F.クワンBB+F.6択維持リゾ213	1480	F.赤BB+F.6択維持リゾ213	1480
F.クワンBB+F.6択維持リゾ231	1480	F.赤BB+F.6択維持リゾ231	1480
F.クワンBB+F.6択維持リゾ312	1480	F.赤BB+F.6択維持リゾ312	1480
F.クワンBB+F.6択維持リゾ321	1480	F.赤BB+F.6択維持リゾ321	1480
F.クワンBB+F.6択突入リゾ123	0	F.赤BB+F.6択突入リゾ123	0
F.クワンBB+F.6択突入リゾ132	0	F.赤BB+F.6択突入リゾ132	0
F.クワンBB+F.6択突入リゾ213	0	F.赤BB+F.6択突入リゾ213	0
F.クワンBB+F.6択突入リゾ231	0	F.赤BB+F.6択突入リゾ231	0
F.クワンBB+F.6択突入リゾ312	0	F.赤BB+F.6択突入リゾ312	0
F.クワンBB+F.6択突入リゾ321	0	F.赤BB+F.6択突入リゾ321	0
F.クワンBB+F.6択転落リゾ123	0	F.赤BB+F.6択転落リゾ1xx	0
F.クワンBB+F.6択転落リゾ132	0	F.赤BB+F.6択転落リゾ2xx	0
F.クワンBB+F.6択転落リゾ213	0	F.赤BB+F.6択転落リゾ3xx	0
F.クワンBB+F.6択転落リゾ231	0	F.赤BB+F.6択転落リゾ123	0
F.クワンBB+F.6択転落リゾ312	0	F.赤BB+F.6択転落リゾ132	0
F.クワンBB+F.6択転落リゾ321	0	F.赤BB+F.6択転落リゾ213	0
F.クワンBB+F.3択昇格リゾ1xx	328	F.赤BB+F.6択転落リゾ321	0
F.クワンBB+F.3択昇格リゾ2xx	0	F.赤BB+F.6択転落リゾ321	0
F.クワンBB+F.3択昇格リゾ3xx	0	F.赤BB+F.6択転落リゾ321	0
F.クワンBB+F.123△△A	1435	F.赤BB+F.123△△A	1435
F.クワンBB+F.123△△B	1435	F.赤BB+F.123△△B	1435
F.クワンBB+F.132△△A	1435	F.赤BB+F.132△△A	1435
F.クワンBB+F.132△△B	1435	F.赤BB+F.132△△B	1435
F.クワンBB+F.213△△A	1435	F.赤BB+F.213△△A	1435
F.クワンBB+F.213△△B	1435	F.赤BB+F.213△△B	1435
F.クワンBB+F.231△△A	1435	F.赤BB+F.231△△A	1435
F.クワンBB+F.231△△B	1435	F.赤BB+F.231△△B	1435
F.クワンBB+F.312△△A	1435	F.赤BB+F.312△△A	1435
F.クワンBB+F.312△△B	1435	F.赤BB+F.312△△B	1435
F.クワンBB+F.321△△A	1435	F.赤BB+F.321△△A	1435
F.クワンBB+F.321△△B	1435	F.赤BB+F.321△△B	1435
F.クワンBB+F.共通△△	1314	F.赤BB+F.共通△△	1314
F.クワンBB+F.強△△	24	F.赤BB+F.強△△	24
F.クワンBB+F.弱△△	730	F.赤BB+F.弱△△	730
F.クワンBB+F.強△△A	224	F.赤BB+F.強△△A	224
F.クワンBB+F.強△△B	104	F.赤BB+F.強△△B	104
F.クワンBB+F.強△△リ	164	F.赤BB+F.強△△リ	164
F.クワンBB+F.確定△△リ	3	F.赤BB+F.確定△△リ	3

【図 182】

内部抽籤テーブル

略称	抽籤値 青BB フラグ間用(RT6)	略称	抽籤値 青BB CBB用
F.青BB	29667	BB中ハズレ	0
F.青BB+F.通常リゾ	164	F.フレイグリゾ	3688
F.青BB+F.維持リゾ	5050	F.リゾA	388
F.青BB+F.弱チリゾ	874	F.リゾB	66
F.青BB+F.強チリゾ	66	F.バリリゾ	66
F.青BB+F.チンズリゾ	388	F.赤BBベル	61328
F.青BB+F.リーチ目リゾ	8		
F.青BB+F.フレイグリゾ	328		
F.青BB+F.リゾA	0		
F.青BB+F.リゾB	0		
F.青BB+F.バリリゾ	0		
F.青BB+F.6択維持リゾ123	1480		
F.青BB+F.6択維持リゾ132	1480		
F.青BB+F.6択維持リゾ213	1480		
F.青BB+F.6択維持リゾ231	1480		
F.青BB+F.6択維持リゾ312	1480		
F.青BB+F.6択維持リゾ321	1480		
F.青BB+F.6択突入リゾ123	0		
F.青BB+F.6択突入リゾ132	0		
F.青BB+F.6択突入リゾ213	0		
F.青BB+F.6択突入リゾ231	0		
F.青BB+F.6択突入リゾ312	0		
F.青BB+F.6択突入リゾ321	0		
F.青BB+F.6択転落リゾ123	0		
F.青BB+F.6択転落リゾ132	0		
F.青BB+F.6択転落リゾ213	0		
F.青BB+F.6択転落リゾ231	0		
F.青BB+F.6択転落リゾ312	0		
F.青BB+F.6択転落リゾ321	0		
F.青BB+F.3択昇格リゾ1xx	328		
F.青BB+F.3択昇格リゾ2xx	0		
F.青BB+F.3択昇格リゾ3xx	0		
F.青BB+F.123△△A	1435		
F.青BB+F.123△△B	1435		
F.青BB+F.132△△A	1435		
F.青BB+F.132△△B	1435		
F.青BB+F.213△△A	1435		
F.青BB+F.213△△B	1435		
F.青BB+F.231△△A	1435		
F.青BB+F.231△△B	1435		
F.青BB+F.312△△A	1435		
F.青BB+F.312△△B	1435		
F.青BB+F.321△△A	1435		
F.青BB+F.321△△B	1435		
F.青BB+F.共通△△	1314		
F.青BB+F.強△△	24		
F.青BB+F.弱△△	730		
F.青BB+F.強△△A	224		
F.青BB+F.強△△B	104		
F.青BB+F.強△△リ	164		
F.青BB+F.確定△△リ	3		

10

20

【図 183】

図柄組合せテーブル

図柄組合せ	当籤倍
R.3rd移行.01	
R.3rd移行.02	
R.3rd移行.03	
R.3rd移行.04	
R.3rd移行.05	
R.3rd移行.06	
R.3rd移行.07	
R.3rd移行.08	
R.3rd移行.09	
R.3rd移行.10	
R.3rd移行.11	
R.3rd移行.12	
R.3rd移行.13	
R.3rd移行.14	
R.3rd移行.15	
R.3rd移行.16	
R.3rd移行.17	
R.3rd移行.18	
R.2nd移行.01	
R.2nd移行.02	
R.2nd移行.03	
R.2nd移行.04	
R.2nd移行.05	
R.2nd移行.06	
R.2nd移行.07	
R.2nd移行.08	
R.2nd移行.09	
R.1st移行.01	
R.1st移行.02	
R.1st移行.03	
R.1st移行.04	
R.1st移行.05	
R.1st移行.06	
R.1st移行.07	
R.1st移行.08	
R.1st移行.09	
R.1st移行.10	
R.1st移行.11	
R.1st移行.12	
R.1st移行.13	
R.1st移行.14	
R.1st移行.15	
R.1st移行.16	
R.1st移行.17	
R.1st移行.18	
T.停止制御A	
T.停止制御B.01	
T.停止制御B.02	
T.停止制御B.03	
T.停止制御B.04	
T.停止制御C	
T.停止制御D.01	
T.停止制御D.02	
T.停止制御D.03	
T.停止制御D.04	
T.停止制御E.01	
T.停止制御E.02	
T.停止制御E.03	
T.停止制御E.04	
T.停止制御F.01	
T.停止制御F.02	
T.停止制御F.03	
T.停止制御F.04	
T.停止制御G	
T.停止制御H	
T.停止制御I	

【図 184】

図柄組合せテーブル

図柄組合せ	当籤倍
T.停止制御.02	
T.停止制御.03	
T.停止制御.04	
T.停止制御.05	
T.停止制御.06	
T.停止制御.07	
T.停止制御.08	
T.停止制御.09	
T.停止制御.10	
T.停止制御.11	
T.停止制御.12	
T.停止制御.13	
T.停止制御.14	
T.停止制御.15	
T.停止制御.16	
T.停止制御.17	
T.停止制御.18	
T.停止制御.19	
T.停止制御.20	
T.停止制御.21	
T.停止制御.22	
T.停止制御.23	
T.停止制御.24	
T.停止制御.25	
T.停止制御.26	
T.停止制御.27	
T.停止制御.28	
T.停止制御.29	
T.停止制御.30	
T.停止制御.31	
T.停止制御.32	
T.停止制御.33	
T.停止制御.34	
T.停止制御.35	
T.停止制御.36	
T.停止制御.37	
T.停止制御.38	
T.停止制御.39	
T.停止制御.40	
T.停止制御.41	
T.停止制御.42	
T.停止制御.43	
T.停止制御.44	
T.停止制御.45	
T.停止制御.46	
T.停止制御.47	
T.停止制御.48	
T.停止制御.49	
T.停止制御.50	
T.停止制御.51	
T.停止制御.52	
T.停止制御.53	
T.停止制御.54	
T.停止制御.55	
T.停止制御.56	
T.停止制御.57	
T.停止制御.58	
T.停止制御.59	
T.停止制御.60	
T.停止制御.61	
T.停止制御.62	
T.停止制御.63	
T.停止制御.64	
T.停止制御.65	
T.停止制御.66	
T.停止制御.67	
T.停止制御.68	
T.停止制御.69	
T.停止制御.70	
T.停止制御.71	
T.停止制御.72	
T.停止制御.73	
T.停止制御.74	
T.停止制御.75	
T.停止制御.76	
T.停止制御.77	
T.停止制御.78	
T.停止制御.79	
T.停止制御.80	
T.停止制御.81	
T.停止制御.82	
T.停止制御.83	
T.停止制御.84	
T.停止制御.85	
T.停止制御.86	
T.停止制御.87	
T.停止制御.88	
T.停止制御.89	
T.停止制御.90	
T.停止制御.91	
T.停止制御.92	
T.停止制御.93	
T.停止制御.94	
T.停止制御.95	
T.停止制御.96	
T.停止制御.97	
T.停止制御.98	
T.停止制御.99	
T.停止制御.100	

30

40

50

【図 185】

図柄組合せテーブル

[illegible]

【図 186】

図柄組合せテーブル

[illegible]

10

【図 187】

図柄組合せテーブル

[illegible]

【図 188】

図柄組合せテーブル

品名	単位	数量
右1:板取B 01	㎡	1.00
右1:板取B 01	㎡	1.00
右1:板取B 02	㎡	1.00
右1:板取B 03	㎡	1.00
右1:板取B 04	㎡	1.00
右1:板取B 05	㎡	1.00
右1:板取B 06	㎡	1.00
右移行段 01	㎡	1.00
右移行段 02	㎡	1.00
右移行段 03	㎡	1.00
中移行段A 01	㎡	1.00
中移行段A 02	㎡	1.00
中移行段A 03	㎡	1.00
中移行段A 04	㎡	1.00
中移行段A 05	㎡	1.00
中移行段A 06	㎡	1.00
中移行段A 07	㎡	1.00
中移行段A 08	㎡	1.00
中移行段B 01	㎡	1.00
中移行段B 02	㎡	1.00
中移行段B 03	㎡	1.00
中移行段B 04	㎡	1.00
中移行段B 05	㎡	1.00
中移行段B 06	㎡	1.00
右移行段 01	㎡	1.00
右移行段 02	㎡	1.00
右移行段 03	㎡	1.00
右移行段 04	㎡	1.00
右移行段 05	㎡	1.00
右移行段 06	㎡	1.00
右移行段 07	㎡	1.00
右移行段 08	㎡	1.00
右移行段 09	㎡	1.00
右移行段 10	㎡	1.00
右移行段 11	㎡	1.00
右移行段 12	㎡	1.00
右移行段 13	㎡	1.00
右移行段 14	㎡	1.00
右移行段 15	㎡	1.00
右移行段 16	㎡	1.00
右移行段 17	㎡	1.00
右移行段 18	㎡	1.00
右移行段 19	㎡	1.00
右移行段 20	㎡	1.00
右移行段 21	㎡	1.00
右移行段 22	㎡	1.00
右移行段 23	㎡	1.00
右移行段 24	㎡	1.00
右移行段 25	㎡	1.00
右移行段 26	㎡	1.00
右移行段 27	㎡	1.00
右移行段 28	㎡	1.00
右移行段 29	㎡	1.00
右移行段 30	㎡	1.00
右移行段 31	㎡	1.00
右移行段 32	㎡	1.00
右移行段 33	㎡	1.00
右移行段 34	㎡	1.00
右移行段 35	㎡	1.00
右移行段 36	㎡	1.00
右移行段 37	㎡	1.00
右移行段 38	㎡	1.00
右移行段 39	㎡	1.00
右移行段 40	㎡	1.00
右移行段 41	㎡	1.00
右移行段 42	㎡	1.00
右移行段 43	㎡	1.00
右移行段 44	㎡	1.00
右移行段 45	㎡	1.00
右移行段 46	㎡	1.00
右移行段 47	㎡	1.00
右移行段 48	㎡	1.00
右移行段 49	㎡	1.00
右移行段 50	㎡	1.00
右移行段 51	㎡	1.00
右移行段 52	㎡	1.00
右移行段 53	㎡	1.00
右移行段 54	㎡	1.00
右移行段 55	㎡	1.00
右移行段 56	㎡	1.00
右移行段 57	㎡	1.00
右移行段 58	㎡	1.00
右移行段 59	㎡	1.00
右移行段 60	㎡	1.00
右移行段 61	㎡	1.00
右移行段 62	㎡	1.00
右移行段 63	㎡	1.00
右移行段 64	㎡	1.00
右移行段 65	㎡	1.00
右移行段 66	㎡	1.00
右移行段 67	㎡	1.00
右移行段 68	㎡	1.00
右移行段 69	㎡	1.00
右移行段 70	㎡	1.00
右移行段 71	㎡	1.00
右移行段 72	㎡	1.00
右移行段 73	㎡	1.00
右移行段 74	㎡	1.00
右移行段 75	㎡	1.00
右移行段 76	㎡	1.00
右移行段 77	㎡	1.00
右移行段 78	㎡	1.00
右移行段 79	㎡	1.00
右移行段 80	㎡	1.00
右移行段 81	㎡	1.00
右移行段 82	㎡	1.00
右移行段 83	㎡	1.00
右移行段 84	㎡	1.00
右移行段 85	㎡	1.00
右移行段 86	㎡	1.00
右移行段 87	㎡	1.00
右移行段 88	㎡	1.00
右移行段 89	㎡	1.00
右移行段 90	㎡	1.00
右移行段 91	㎡	1.00
右移行段 92	㎡	1.00
右移行段 93	㎡	1.00
右移行段 94	㎡	1.00
右移行段 95	㎡	1.00
右移行段 96	㎡	1.00
右移行段 97	㎡	1.00
右移行段 98	㎡	1.00
右移行段 99	㎡	1.00
右移行段 100	㎡	1.00

30

40

【圖 189】

図柄組合せテーブル

[illegible]

【圖 190】

図柄組合せテーブル

[illegible]

10

20

【図 191】

図柄組合せテーブル

[illegible]

【図 192】

図柄組合せテーブル

[illegible]

30

40

50

【図 193】

図柄組合せテーブル

[illegible]

【図 194】

図柄組合せテーブル

[illegible]

10

20

【図 195】

図柄組合せテーブル

[illegible]

【図 196】

図柄組合せテーブル

[illegible]

30

40

50

【図 197】

図柄組合せの内容				
コンビネーション		名称	払出 3BET	略称
ベル1	ベル1	赤7	R 3rd移行_01	0
ベル1	ベル1	チェリー	R 3rd移行_02	0
ベル1	ベル2	赤7	R 3rd移行_03	0
ベル1	ベル2	チェリー	R 3rd移行_04	0
ベル1	ベル3	赤7	R 3rd移行_05	0
ベル1	ベル3	チェリー	R 3rd移行_06	0
ベル2	ベル1	赤7	R 3rd移行_07	0
ベル2	ベル1	チェリー	R 3rd移行_08	0
ベル2	ベル2	赤7	R 3rd移行_09	0
ベル2	ベル2	チェリー	R 3rd移行_10	0
ベル2	ベル3	赤7	R 3rd移行_11	0
ベル2	ベル3	チェリー	R 3rd移行_12	0
ベル3	ベル1	赤7	R 3rd移行_13	0
ベル3	ベル1	チェリー	R 3rd移行_14	0
ベル3	ベル2	赤7	R 3rd移行_15	0
ベル3	ベル2	チェリー	R 3rd移行_16	0
ベル3	ベル3	赤7	R 3rd移行_17	0
ベル3	ベル3	チェリー	R 3rd移行_18	0
ベル1	チェリー	ベル1	R 2nd移行_01	0
ベル1	チェリー	ベル2	R 2nd移行_02	0
ベル1	チェリー	ベル3	R 2nd移行_03	0
ベル2	チェリー	ベル1	R 2nd移行_04	0
ベル2	チェリー	ベル2	R 2nd移行_05	0
ベル2	チェリー	ベル3	R 2nd移行_06	0
ベル3	チェリー	ベル1	R 2nd移行_07	0
ベル3	チェリー	ベル2	R 2nd移行_08	0
ベル3	チェリー	ベル3	R 2nd移行_09	0
スイカ2	ベル1	ベル1	R 1st移行_01	0
スイカ2	ベル1	ベル2	R 1st移行_02	0
スイカ2	ベル1	ベル3	R 1st移行_03	0
スイカ2	ベル2	ベル1	R 1st移行_04	0
スイカ2	ベル2	ベル2	R 1st移行_05	0
スイカ2	ベル2	ベル3	R 1st移行_06	0
スイカ2	ベル3	ベル1	R 1st移行_07	0
スイカ2	ベル3	ベル2	R 1st移行_08	0
スイカ2	ベル3	ベル3	R 1st移行_09	0
リプレイ	ベル1	ベル1	R 1st移行_10	0
リプレイ	ベル1	ベル2	R 1st移行_11	0
リプレイ	ベル1	ベル3	R 1st移行_12	0
リプレイ	ベル2	ベル1	R 1st移行_13	0
リプレイ	ベル2	ベル2	R 1st移行_14	0
リプレイ	ベル2	ベル3	R 1st移行_15	0
リプレイ	ベル3	ベル1	R 1st移行_16	0
リプレイ	ベル3	ベル2	R 1st移行_17	0
リプレイ	ベル3	ベル3	R 1st移行_18	0
スイカ2	スイカ2	リプレイ	T 停止制御A	0
スイカ2	チェリー	スイカ2	T 停止制御B_01	0
スイカ2	チェリー	スイカ1	T 停止制御B_02	0
スイカ1	スイカ2	スイカ2	T 停止制御B_03	0
スイカ1	チェリー	スイカ1	T 停止制御B_04	0
リプレイ	スイカ2	リプレイ	T 停止制御C	0
リプレイ	チェリー	ベル2	T 停止制御D_01	0
リプレイ	チェリー	ベル3	T 停止制御D_02	0

【図 198】

図柄組合せの内容				略称
コンビネーション	名称		払出 3BET	
スイカ2	チェリー	リプレイ	T.停止制御E01	制御用出目
リプレイ	チェリー	リプレイ	T.停止制御E02	
スイカ2	チェリー	ベル2	T.停止制御F01	
スイカ2	チェリー	ベル3	T.停止制御F02	
リプレイ	赤7	スイカ2	T.停止制御G01	
リプレイ	赤7	スイカ1	T.停止制御G02	
リプレイ	ベル1	スイカ2	T.停止制御G03	
リプレイ	ベル1	スイカ1	T.停止制御G04	
BAR	スイカ2	チェリー	T.停止制御H	
BAR	赤7	青7	T.停止制御I01	
BAR	赤7	BAR	T.停止制御I02	
BAR	赤7	チェリー	T.停止制御J03	
BAR	リプレイ	青7	T.停止制御J01	
BAR	リプレイ	BAR	T.停止制御J02	
BAR	リプレイ	チェリー	T.停止制御J03	
BAR	チェリー	青7	T.停止制御K01	
BAR	チェリー	BAR	T.停止制御K02	
スイカ1	チェリー	青7	T.停止制御K03	
スイカ1	チェリー	BAR	T.停止制御K04	
BAR	チェリー	リプレイ	T.停止制御L01	
スイカ1	チェリー	リプレイ	T.停止制御L02	
スイカ1	スイカ1	スイカ1	BB制御用	
赤7	青7		C.7RRR01	クワンBB
ベル3	青7	スイカ2	C.7RRR02	
ベル3	赤7	スイカ2	C.7RRR03	
青7	青7	青7	C.7RRR04	BB
ベル2	青7	ベル2	C.7RRR05	
リプレイ	リプレイ	リプレイ	C.0DJ7	
ベル1	リプレイ	ベル1	C.0UJ701	斜めリプレイ
ベル1	リプレイ	スイカ2	C.0UJ702	
ベル1	リプレイ	スイカ1	C.0UJ703	
ベル2	リプレイ	ベル1	C.0UJ704	
ベル2	リプレイ	スイカ2	C.0UJ705	
ベル2	リプレイ	スイカ1	C.0UJ706	
ベル3	リプレイ	ベル1	C.0UJ707	
ベル3	リプレイ	スイカ2	C.0UJ708	
ベル3	リプレイ	スイカ1	C.0UJ709	
ベル1	リプレイ	リプレイ	C.0UJ710	
ベル2	リプレイ	リプレイ	C.0UJ712	
ベル3	リプレイ	リプレイ	C.0UJ713	
青7	チェリー	ベル1	C.0UJ715	
青7	チェリー	スイカ2	C.0UJ716	
青7	チェリー	スイカ1	C.0UJ717	
チェリー	ベル1	ベル1	C.0UJ718	
チェリー	ベル1	スイカ2	C.0UJ719	
チェリー	ベル1	スイカ1	C.0UJ720	
チェリー	ベル2	ベル1	C.0UJ724	
チェリー	ベル2	スイカ2	C.0UJ725	
チェリー	ベル2	スイカ1	C.0UJ726	
チェリー	ベル3	ベル1	C.0UJ727	
チェリー	ベル3	スイカ2	C.0UJ728	
チェリー	ベル3	スイカ1	C.0UJ729	
青7	チェリー	赤7	C.0UJ730	

10

20

【図 199】

図柄組合せの内容				払出 3BET	略称
コンビネーション			名称		
青7	チェリー	チェリー	C.RT377 02	0	弱チェリブ (RT3移行図柄)
チェリー	チェリー	赤7	C.RT377 03	0	
チェリー	チェリー	チェリー	C.RT377 04	0	
BAR	チェリー	赤7	C.RT377 01	0	
BAR	チェリー	チェリー	C.RT377 02	0	
スイカ1	チェリー	赤7	C.RT377 03	0	
スイカ1	チェリー	チェリー	C.RT377 04	0	
BAR	ベル1	ベル1	C.BB77 01	0	BB中リプレイ
BAR	ベル1	スイカ2	C.BB77 02	0	
BAR	ベル1	スイカ1	C.BB77 03	0	
BAR	ベル2	ベル1	C.BB77 04	0	
BAR	ベル2	スイカ2	C.BB77 05	0	
BAR	ベル2	スイカ1	C.BB77 06	0	
BAR	ベル3	ベル1	C.BB77 07	0	
BAR	ベル3	スイカ2	C.BB77 08	0	
BAR	ベル3	スイカ1	C.BB77 09	0	
スイカ1	ベル1	ベル1	C.BB77 10	0	
スイカ1	ベル1	スイカ2	C.BB77 11	0	
スイカ1	ベル1	スイカ1	C.BB77 12	0	
スイカ1	ベル2	ベル1	C.BB77 13	0	
スイカ1	ベル2	スイカ2	C.BB77 14	0	
スイカ1	ベル2	スイカ1	C.BB77 15	0	
スイカ1	ベル3	ベル1	C.BB77 16	0	
スイカ1	ベル3	スイカ2	C.BB77 17	0	
スイカ1	ベル3	スイカ1	C.BB77 18	0	
ベル1	青7	リプレイ	C.BT77 01	0	平行リプレイ (RT1移行図柄)
ベル1	BAR	リプレイ	C.BT77 02	0	
ベル1	ベル1	リプレイ	C.BT77 03	0	
ベル2	青7	リプレイ	C.BT77 04	0	
ベル2	BAR	リプレイ	C.BT77 05	0	
ベル2	ベル1	リプレイ	C.BT77 06	0	
ベル3	青7	リプレイ	C.BT77 07	0	
ベル3	BAR	リプレイ	C.BT77 08	0	
ベル3	ベル1	リプレイ	C.BT77 09	0	
赤7	リプレイ	赤7	C.CT77 01	0	
赤7	リプレイ	チェリー	C.CT77 02	0	
青7	リプレイ	赤7	C.CT77 03	0	
青7	リプレイ	チェリー	C.CT77 04	0	
スイカ1	リプレイ	赤7	C.CT77 05	0	
スイカ1	リプレイ	チェリー	C.CT77 06	0	
リプレイ	チェリー	ベル1	C.TP77 01	0	
リプレイ	チェリー	スイカ2	C.TP77 02	0	
リプレイ	チェリー	スイカ1	C.TP77 03	0	
赤7	リプレイ	ベル1	C.RT977 01	0	強チャンスリブ (RT3移行図柄)
赤7	リプレイ	スイカ2	C.RT977 02	0	
赤7	リプレイ	スイカ1	C.RT977 03	0	
青7	リプレイ	ベル1	C.RT977 04	0	
青7	リプレイ	スイカ2	C.RT977 05	0	
青7	リプレイ	スイカ1	C.RT977 06	0	
スイカ1	リプレイ	チェリー	C.CT77 07	0	
スイカ1	リプレイ	スイカ2	C.RT977 08	0	
スイカ1	リプレイ	スイカ1	C.RT977 09	0	
スイカ1	リプレイ	スイカ2	C.RT977 10	0	
スイカ1	リプレイ	スイカ1	C.RT977 11	0	
スイカ1	リプレイ	スイカ2	C.RT977 12	0	
スイカ1	リプレイ	スイカ1	C.RT977 13	0	
スイカ1	リプレイ	スイカ2	C.RT977 14	0	
スイカ1	リプレイ	スイカ1	C.RT977 15	0	
スイカ1	リプレイ	スイカ2	C.RT977 16	0	
スイカ1	リプレイ	スイカ1	C.RT977 17	0	

【図 200】

図柄組合せの内容				払出 3BET	略称
コンビネーション		名称			
赤7	赤7	赤7	C.0D7777 A 01	0	7揃いリブ (RT5移行図柄)
赤7	赤7	チェリー	C.0D7777 B 01	0	
赤7	スイカ2	チェリー	C.0D7777 C 02	0	
赤7	チェリー	赤7	C.0D7777 D 01	0	
スイカ2	チェリー	赤7	C.0D7777 C 02	0	
リプレイ	チェリー	赤7	C.0D7777 C 03	0	
青7	赤7	赤7	C.0D7777 D 01	0	
青7	赤7	チェリー	C.0D7777 D 02	0	
スイカ1	赤7	赤7	C.0D7777 D 03	0	
スイカ1	赤7	チェリー	C.0D7777 D 04	0	
赤7	BAR	BAR	C.0TP7777 B 01	0	
赤7	リプレイ	BAR	C.0TP7777 B 01	0	
スイカ1	リプレイ	BAR	C.0TP7777 B 02	0	
スイカ1	リプレイ	BAR	C.0TP7777 B 03	0	
チェリー	リプレイ	BAR	C.0TP7777 B 04	0	
スイカ2	赤7	BAR	C.0U7777 A 0	0	
スイカ2	赤7	リプレイ	C.0U7777 B 01	0	
スイカ2	リプレイ	リプレイ	C.0U7777 B 02	0	
青7	赤7	BAR	C.0U7777 B 03	0	
青7	スイカ2	BAR	C.0U7777 C 02	0	
ベル1	赤7	BAR	C.0U7777 C 03	0	
ベル1	スイカ2	BAR	C.0U7777 C 04	0	
赤7	BAR	リプレイ	C.0UBAR777 A 0	0	
赤7	チェリー	リプレイ	C.0UBAR777 B 0	0	
BAR	BAR	BAR	C.0CDBAR777 A 0	0	
BAR	BAR	リプレイ	C.0CDBAR777 B 01	0	
BAR	リプレイ	リプレイ	C.0CDBAR777 B 02	0	
BAR	BAR	ベル1	C.0CDBAR777 C 02	0	
BAR	BAR	ベル1	C.0CDBAR777 C 03	0	
赤7	赤7	BAR	C.0BTBAR777 A 0	0	
赤7	スイカ2	BAR	C.0BTBAR777 B 0	0	
スイカ1	赤7	BAR	C.0BTBAR777 C 01	0	
スイカ1	スイカ2	BAR	C.0BTBAR777 C 02	0	
赤7	赤7	ベル1	C.0JA79A 01	0	
赤7	赤7	スイカ2	C.0JA79A 02	0	
赤7	赤7	スイカ1	C.0JA79B 01	0	
赤7	チェリー	スイカ2	C.0JA79B 02	0	
赤7	チェリー	スイカ1	C.0JA79B 03	0	
スイカ2	赤7	スイカ2	C.0JA79C 01	0	
スイカ2	赤7	スイカ1	C.0JA79C 02	0	
スイカ2	リプレイ	スイカ2	C.0JA79D 01	0	
スイカ2	リプレイ	スイカ1	C.0JA79D 02	0	
スイカ2	赤7	チェリー	C.0JA79E 0	0	
スイカ2	赤7	赤7	C.0JA79F 01	0	
リプレイ	赤7	赤7	C.0JA79F 02	0	
スイカ2	リプレイ	赤7	C.0JA79G 01	0	
スイカ2	リプレイ	チェリー	C.0JA79G 02	0	
リプレイ	リプレイ	赤7	C.0JA79G 03	0	
リプレイ	リプレイ	チェリー	C.0JA79G 04	0	
スイカ2	スイカ2	青7	C.0特狭777 A 01	0	
スイカ2	スイカ2	BAR	C.0特狭777 B 02	0	
スイカ2	スイカ2	スイカ1	C.0特狭777 C 03	0	
リプレイ	スイカ2	BAR	C.0特狭777 D 04	0	

【図 201】

図柄組合せの内容			払出 3BET	略称
コンビネーション		名称		
スイカ2	スイカ2	チェリー	C.特殊J'B_01	リーチ目リブ
リブレイ	スイカ2	チェリー	C.特殊J'B_02	
スイカ2	チェリー	青7	C.特殊J'C_01	
スイカ2	チェリー	BAR	C.特殊J'C_02	
リブレイ	チェリー	青7	C.特殊J'C_03	
リブレイ	チェリー	BAR	C.特殊J'C_04	
BAR	赤7	リブレイ	C.特殊J'D_01	
BAR	スイカ2	リブレイ	C.特殊J'D_02	
ベル1	ベル1	ベル1	C.CD×J'AAA_01	ベル
ベル1	ベル1	ベル3	C.CD×J'AAA_02	
ベル1	ベル2	ベル1	C.CD×J'AAA_03	
ベル1	ベル2	ベル3	C.CD×J'AAA_04	
ベル2	ベル1	ベル1	C.CD×J'AAA_05	
ベル2	ベル1	ベル3	C.CD×J'AAA_06	
ベル2	ベル2	ベル1	C.CD×J'AAA_07	
ベル2	ベル2	ベル3	C.CD×J'AAA_08	
ベル1	ベル1	ベル2	C.CD×J'AAB_01	
ベル1	ベル2	ベル2	C.CD×J'AAB_02	
ベル2	ベル1	ベル2	C.CD×J'AAB_03	
ベル2	ベル2	ベル2	C.CD×J'AAB_04	
ベル1	ベル3	ベル1	C.CD×J'ABA_01	
ベル1	ベル3	ベル3	C.CD×J'ABA_02	
ベル2	ベル3	ベル1	C.CD×J'ABA_03	
ベル2	ベル3	ベル3	C.CD×J'ABA_04	
ベル1	ベル3	ベル2	C.CD×J'ABB_01	
ベル3	ベル1	ベル1	C.CD×J'BAA_01	
ベル3	ベル1	ベル3	C.CD×J'BAA_02	
ベル3	ベル2	ベル1	C.CD×J'BAA_03	
ベル3	ベル2	ベル3	C.CD×J'BAA_04	
ベル3	ベル1	ベル2	C.CD×J'BAB_01	
ベル3	ベル2	ベル2	C.CD×J'BAB_02	
ベル3	ベル3	ベル1	C.CD×J'BBA_01	
ベル3	ベル3	ベル3	C.CD×J'BBA_02	
ベル3	ベル3	ベル2	C.CD×J'BBB_01	
ベル1	赤7	赤7	C.TPA×J_01	
ベル1	赤7	チェリー	C.TPA×J_02	
ベル1	スイカ2	赤7	C.TPA×J_03	
ベル1	スイカ2	チェリー	C.TPA×J_04	
ベル1	リブレイ	赤7	C.TPA×J_05	
ベル1	リブレイ	チェリー	C.TPA×J_06	
ベル2	赤7	赤7	C.TPA×J_07	
ベル2	赤7	チェリー	C.TPA×J_08	
ベル2	スイカ2	赤7	C.TPA×J_09	
ベル2	スイカ2	チェリー	C.TPA×J_10	
ベル2	リブレイ	赤7	C.TPA×J_11	
ベル2	リブレイ	チェリー	C.TPA×J_12	
ベル3	赤7	赤7	C.TPA×J_13	
ベル3	赤7	チェリー	C.TPA×J_14	
ベル3	スイカ2	赤7	C.TPA×J_15	
ベル3	スイカ2	チェリー	C.TPA×J_16	
ベル3	リブレイ	赤7	C.TPA×J_17	
ベル3	リブレイ	チェリー	C.TPA×J_18	

【図 202】

図柄組合せの内容				払出 3BET	略称
コンビネーション			名称		
スイカ2	ベル1	リブレイ	C.OTA×J_01	8	ベル
スイカ2	ベル2	リブレイ	C.OTA×J_02	8	
スイカ2	ベル3	リブレイ	C.OTA×J_03	8	
リブレイ	ベル1	リブレイ	C.OTA×J_04	8	
リブレイ	ベル2	リブレイ	C.OTA×J_05	8	
リブレイ	ベル3	リブレイ	C.OTA×J_06	8	
BAR	ベル1	赤7	C.QUA×J_01	8	
BAR	ベル1	チェリー	C.QUA×J_02	8	
BAR	ベル2	赤7	C.QUA×J_03	8	
BAR	ベル2	チェリー	C.QUA×J_04	8	
BAR	ベル3	赤7	C.QUA×J_05	8	
BAR	ベル3	チェリー	C.QUA×J_06	8	
スイカ1	ベル1	赤7	C.QUA×J_07	8	
スイカ1	ベル1	チェリー	C.QUA×J_08	8	
スイカ1	ベル2	赤7	C.QUA×J_09	8	
スイカ1	ベル2	チェリー	C.QUA×J_10	8	
スイカ1	ベル3	赤7	C.QUA×J_11	8	
スイカ1	ベル3	チェリー	C.QUA×J_12	8	
チェリー	ベル1	赤7	C.QUA×J_13	8	
チェリー	ベル1	チェリー	C.QUA×J_14	8	
チェリー	ベル2	赤7	C.QUA×J_15	8	
チェリー	ベル2	チェリー	C.QUA×J_16	8	
チェリー	ベル3	赤7	C.QUA×J_17	8	
チェリー	ベル3	チェリー	C.QUA×J_18	8	
BAR	チェリー	ベル1	C.BTA×J_01	8	
BAR	チェリー	ベル2	C.BTA×J_02	8	
BAR	チェリー	ベル3	C.BTA×J_03	8	
スイカ1	チェリー	ベル1	C.BTA×J_04	8	
スイカ1	チェリー	ベル2	C.BTA×J_05	8	
スイカ1	チェリー	ベル3	C.BTA×J_06	8	
チェリー	チェリー	ベル1	C.BTA×J_07	8	
チェリー	チェリー	ベル2	C.BTA×J_08	8	
チェリー	チェリー	ベル3	C.BTA×J_09	8	
ベル1	赤7	リブレイ	C.左1枚役A_01	1	
ベル1	スイカ2	リブレイ	C.左1枚役A_02	1	
ベル2	赤7	リブレイ	C.左1枚役A_03	1	
ベル2	スイカ2	リブレイ	C.左1枚役A_04	1	
ベル3	赤7	リブレイ	C.左1枚役A_05	1	
ベル3	スイカ2	リブレイ	C.左1枚役A_06	1	
ベル1	チェリー	青7	C.左1枚役B_01	1	
ベル1	チェリー	BAR	C.左1枚役B_02	1	
ベル2	チェリー	青7	C.左1枚役B_03	1	
ベル2	チェリー	BAR	C.左1枚役B_04	1	
ベル3	チェリー	青7	C.左1枚役B_05	1	
ベル3	チェリー	BAR	C.左1枚役B_06	1	
BAR	リブレイ	ベル1	C.右1枚役A_01	1	
BAR	リブレイ	ベル2	C.右1枚役A_02	1	
BAR	リブレイ	ベル3	C.右1枚役A_03	1	
チェリー	リブレイ	ベル1	C.右1枚役A_04	1	
チェリー	リブレイ	ベル2	C.右1枚役A_05	1	
チェリー	リブレイ	ベル3	C.右1枚役A_06	1	
スイカ2	青7	ベル1	C.右1枚役B_01	1	
スイカ2	青7	ベル2	C.右1枚役B_02	1	

【図 203】

図柄組合せの内容			払出 3BET	略称
コンビネーション				
スイカ2	青7	ベル3	C.右1枚役B_03	1枚出目
スイカ2	BAR	ベル1	C.右1枚役B_04	
スイカ2	BAR	ベル2	C.右1枚役B_05	
スイカ2	BAR	ベル3	C.右1枚役B_06	
ベル1	チェリー	リブレイ	C.左移行役_01	
ベル2	チェリー	リブレイ	C.左移行役_02	
ベル3	チェリー	リブレイ	C.左移行役_03	
スイカ2	ベル1	赤7	C.中移行役A_01	
スイカ2	ベル1	チェリー	C.中移行役A_02	
スイカ2	ベル2	赤7	C.中移行役A_03	
スイカ2	ベル2	チェリー	C.中移行役A_04	
リブレイ	ベル1	赤7	C.中移行役A_05	
リブレイ	ベル1	チェリー	C.中移行役A_06	
リブレイ	ベル2	赤7	C.中移行役A_07	
リブレイ	ベル2	チェリー	C.中移行役A_08	
スイカ2	ベル3	スイカ2	C.中移行役B_01	
スイカ2	ベル3	スイカ1	C.中移行役B_02	
スイカ2	ベル3	チェリー	C.中移行役B_03	
リブレイ	ベル3	スイカ2	C.中移行役B_04	
リブレイ	ベル3	スイカ1	C.中移行役B_05	
リブレイ	ベル3	チェリー	C.中移行役B_06	
スイカ2	リブレイ	ベル1	C.右移行役_01	
スイカ2	リブレイ	ベル2	C.右移行役_02	
スイカ2	リブレイ	ベル3	C.右移行役_03	
リブレイ	リブレイ	ベル1	C.右移行役_04	
リブレイ	リブレイ	ベル2	C.右移行役_05	
リブレイ	リブレイ	ベル3	C.右移行役_06	
スイカ2	スイカ2	スイカ2	C.ス化A_01	
スイカ2	スイカ2	スイカ1	C.ス化A_02	
リブレイ	スイカ2	ベル2	C.ス化B_01	
リブレイ	スイカ2	ベル3	C.ス化B_02	
スイカ2	スイカ2	ベル2	C.ス化C_01	
スイカ2	スイカ2	ベル3	C.ス化C_02	
リブレイ	スイカ2	スイカ2	C.ス化D_01	
リブレイ	スイカ2	スイカ1	C.ス化D_02	
スイカ1	スイカ2	スイカ2	C.ス化E_01	
スイカ1	スイカ2	スイカ1	C.ス化E_02	
BAR	スイカ2	青7	C.ス化F_01	
BAR	スイカ2	BAR	C.ス化F_02	
青7	チェリー	青7	C.チェリ-A_01	
青7	チェリー	BAR	C.チェリ-A_02	
チェリー	ベル1	青7	C.チェリ-B_01	
チェリー	ベル1	BAR	C.チェリ-B_02	
チェリー	ベル2	青7	C.チェリ-B_03	
チェリー	ベル2	BAR	C.チェリ-B_04	
チェリー	ベル3	青7	C.チェリ-B_05	
チェリー	ベル3	BAR	C.チェリ-B_06	
チェリー	ベル1	リブレイ	C.チェリ-C_01	
チェリー	ベル2	リブレイ	C.チェリ-C_02	
チェリー	ベル3	リブレイ	C.チェリ-C_03	
チェリー	チェリー	青7	C.チェリ-D_01	
チェリー	チェリー	BAR	C.チェリ-D_02	
青7	チェリー	リブレイ	C.チェリ-E_01	
チェリー	チェリー	リブレイ	C.チェリ-E_02	

【図 204】

図柄組合せの内容			払出 3BET	略称
コンビネーション				
リブレイ	青7	スイカ2	C.SP役A_01	確定出目
リブレイ	青7	スイカ1	C.SP役A_02	
リブレイ	BAR	スイカ2	C.SP役A_03	
リブレイ	BAR	スイカ1	C.SP役A_04	
リブレイ	青7	ベル1	C.SP役B_01	
リブレイ	BAR	ベル1	C.SP役B_02	
リブレイ	リブレイ	スイカ2	C.SP役C_01	
リブレイ	リブレイ	スイカ1	C.SP役C_02	
リブレイ	青7	ベル2	C.SP役D_01	
リブレイ	青7	ベル3	C.SP役D_02	
リブレイ	BAR	ベル2	C.SP役E_01	
リブレイ	BAR	ベル3	C.SP役E_02	
青7	リブレイ	青7	C.SP役F_01	
青7	リブレイ	BAR	C.SP役F_02	
青7	リブレイ	リブレイ	C.SP役G_01	
ベル1	ベル1	赤7	C.VA×J_01	
ベル1	ベル1	チェリー	C.VA×J_02	
ベル1	ベル2	赤7	C.VA×J_03	
ベル1	ベル2	チェリー	C.VA×J_04	
ベル1	ベル3	赤7	C.VA×J_05	
ベル1	ベル3	チェリー	C.VA×J_06	
ベル2	ベル1	赤7	C.VA×J_07	
ベル2	ベル1	チェリー	C.VA×J_08	
ベル2	ベル2	赤7	C.VA×J_09	
ベル2	ベル2	チェリー	C.VA×J_10	
ベル2	ベル3	赤7	C.VA×J_11	
ベル2	ベル3	チェリー	C.VA×J_12	
ベル3	ベル1	赤7	C.VA×J_13	
ベル3	ベル1	チェリー	C.VA×J_14	
ベル3	ベル2	赤7	C.VA×J_15	
ベル3	ベル2	チェリー	C.VA×J_16	
ベル3	ベル3	赤7	C.VA×J_17	
ベル3	ベル3	チェリー	C.VA×J_18	

10

20

30

40

50

【図 205】

(A) 内部当籤役と図柄組合せとの対応関係(非押し順役)

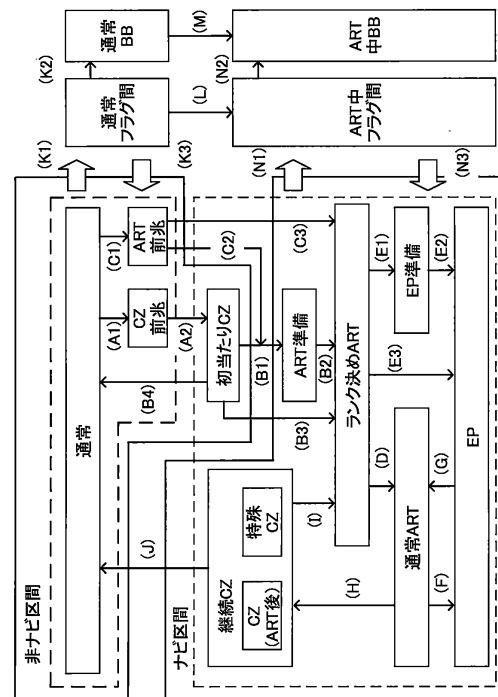
当籤役	図柄組合せ	当籤役	図柄組合せ
F.クラウンBB	クラウンBB	F.BARIリブ	BAR揃いリブ(RT5移行図柄)
F.赤BB	BB	F.共通ベル	ベル
F.青BB		F.強ベル	
F.通常リブ	平行リブ(RT1移行図柄)	F.弱スッパ	スッパ
F.維持リブ	斜めリブ	F.強スッパA	
F.弱チェリ	弱チェリ(RT3移行図柄)	F.強スッパB	チェリ
F.強チェリ	強チェリ(RT5移行図柄)	F.強スッパC	
F.チャンスリブ	小山リブ(RT4移行図柄)	F.確定チェリ	確定スッパ
F.リーチ目リブ	リーチ目リブ	F.CBBベル	CBB中ベル
F.フェイクリブ	フェイクリブ	F.BB中弱チェリ	BB中リブ
F.7リブA	7揃いリブ(RT5移行図柄)	F.BB中共通ベル	ベル
F.7リブB		F.BB中強ベル	

(B) 内部当籤役と図柄組合せとの対応関係(押し順役)

当籤役	図柄組合せ(押し順)					
	123	132	213	231	312	321
F.6枚維持リブ123	C.CUIリブ	C.BTリブ	C.CTリブ	C.CTリブ	C.BTリブ	C.BTリブ
F.6枚維持リブ132	C.BTリブ	C.CUIリブ	C.CTリブ	C.CTリブ	C.BTリブ	C.BTリブ
F.6枚維持リブ213	C.BTリブ	C.BTリブ	C.CUIリブ	C.CTリブ	C.BTリブ	C.BTリブ
F.6枚維持リブ231	C.BTリブ	C.BTリブ	C.CTリブ	C.CUIリブ	C.BTリブ	C.BTリブ
F.6枚維持リブ312	C.BTリブ	C.BTリブ	C.CTリブ	C.CTリブ	C.CUIリブ	C.BTリブ
F.6枚維持リブ321	C.BTリブ	C.BTリブ	C.CTリブ	C.CTリブ	C.BTリブ	C.CUIリブ
F.6枚突入リブ123	小山ワイルド	C.BTリブ	C.CTリブ	C.CTリブ	C.BTリブ	C.BTリブ
F.6枚突入リブ132	C.BTリブ	小山ワイルド	C.CTリブ	C.CTリブ	C.BTリブ	C.BTリブ
F.6枚突入リブ213	C.BTリブ	C.BTリブ	小山ワイルド	C.CTリブ	C.BTリブ	C.BTリブ
F.6枚突入リブ231	C.BTリブ	C.BTリブ	C.CTリブ	小山ワイルド	C.BTリブ	C.BTリブ
F.6枚突入リブ312	C.BTリブ	C.BTリブ	C.CTリブ	C.CTリブ	小山ワイルド	C.BTリブ
F.6枚突入リブ321	C.BTリブ	C.BTリブ	C.CTリブ	C.CTリブ	C.BTリブ	小山ワイルド
F.6枚転落リブ123	C.CUIリブ	小山ワイルド	小山ワイルド	小山ワイルド	小山ワイルド	小山ワイルド
F.6枚転落リブ132	小山ワイルド	C.CUIリブ	小山ワイルド	小山ワイルド	小山ワイルド	小山ワイルド
F.6枚転落リブ213	小山ワイルド	小山ワイルド	C.CUIリブ	小山ワイルド	小山ワイルド	小山ワイルド
F.6枚転落リブ231	小山ワイルド	小山ワイルド	小山ワイルド	C.CUIリブ	小山ワイルド	小山ワイルド
F.6枚転落リブ312	小山ワイルド	小山ワイルド	小山ワイルド	小山ワイルド	C.CUIリブ	小山ワイルド
F.6枚転落リブ321	小山ワイルド	小山ワイルド	小山ワイルド	小山ワイルド	小山ワイルド	C.CUIリブ
F.3枚昇格リブ1xx	強チャンスリブ	強チャンスリブ	C.CTリブ	C.CTリブ	C.CUIリブ	C.CUIリブ
F.3枚昇格リブ2xx	C.CUIリブ	C.CUIリブ	強チャンスリブ	強チャンスリブ	C.BTリブ	C.BTリブ
F.3枚昇格リブ3xx	C.BTリブ	C.BTリブ	C.CUIリブ	C.CUIリブ	強チャンスリブ	強チャンスリブ

※ 押し順ベルについては、省略

【図 206】



【図 207】

移行条件	移行元	移行先	内容
(A1)	通常	CZ前兆	CZ抽籤に当籤、かつ、前兆あり
(A2)	CZ前兆	初当たりCZ	前兆G数を消化
(B1)	初当たりCZ	ART準備	ART抽籤に当籤、かつ、低RT
(B2)	ART準備	ランク決めART	高RTに移行
(B3)	初当たりCZ	ランク決めART	ART抽籤に当籤、かつ、高RT
(B4)	初当たりCZ	通常	ART抽籤に非当籤
(C1)	通常	ART前兆	ART抽籤に当籤、かつ、前兆あり
(C2)	ART前兆	ART準備	前兆G数を消化、かつ、低RT
(C3)	ART前兆	ランク決めART	前兆G数を消化、かつ、高RT
(D)	ランク決めART	通常ART	1G消化、かつ、ランク3以下
(E1)	ランク決めART	EP準備	1G消化、かつ、ランク4、かつ、RT4
(E2)	EP準備	EP	RT5移行
(E3)	ランク決めART	EP	1G消化、かつ、ランク4、かつ、RT5
(F)	通常ART	EP	ナビ高確中に3枚昇格リブ当籤し、RT5移行
(G)	EP	通常ART	保証なし時の転落リブ当籤時に押し順ミスし、RT4移行
(H)	通常ART	継続CZ	通常ARTのG数消化
(I)	継続CZ	ランク決めART	ART抽籤に当籤、又は、ストック放出
(J)	継続CZ	通常	ART抽籤に非当籤、かつ、ストックなし
(K1)	特定状態	通常BBフラグ間	BB当籤
(K2)	通常フラグ間	通常BB	BB作動
(K3)	通常BB	特定状態	BB終了
(L)	通常フラグ間	ART中フラグ間	ART抽籤に当籤
(M)	通常BB	ART中BB	ART抽籤に当籤
(N1)	所定状態	ART中フラグ間	BB当籤
(N2)	ART中フラグ間	ART中BB	BB作動
(N3)	ART中BB	ART準備	BB終了

【図 208】

(A) 内部当籤役と抽籤用フラグとの対応関係(スタート時の抽籤用)

当籤役	抽籤用フラグ	当籤役	抽籤用フラグ
はずれ	はずれ	6枚維持リブ	通常リブ
F.クラウンBB	BB(CBB)	押し順ベル	ベル
F.赤BB	BB(NBB)	F.共通ベル	強ベル
F.青BB	通常リブ	F.弱スッパ	弱スッパ
F.通常リブ	通常リブ	F.強スッパA	強スッパ
F.維持リブ	弱チェリ	F.強スッパB	強スッパ
F.弱チェリ	強チャンスリブ	F.強スッパC	強スッパ
F.強チェリ	弱チャンスリブ	F.確定チェリ	確定スッパ
F.チャンスリブ	リーチ目リブ	F.CBBベル	CBB中ベル
F.リーチ目リブ	フェイクリブ	BB中のはずれ	ベルはずれA
F.フェイクリブ	7揃い	F.BB中弱チェリ	ベルはずれB
F.7リブA	BAR揃い	F.BB中共通ベル	中強ベル
F.7リブB		F.BB中強ベル	斜めベル

(B) 内部当籤役と抽籤用フラグとの対応関係(スタート時又は停止時の抽籤用)

当籤役	抽籤用フラグ			備考
	非ナビ区間(停止時)	ナビ区間(スタート時)	ナビ高確	ナビ高確
6枚突入リブ	弱チャンスリブ	通常リブ	弱チャンスリブ	正解時にRT2⇒RT4
6枚転落リブ	通常リブ	弱チャンスリブ	弱チャンスリブ	不正解時にRT5⇒RT4
3枚昇格リブ	強チャンスリブ	通常リブ	強チャンスリブ	正解時にRT4⇒RT5

10

20

30

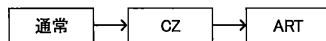
40

50

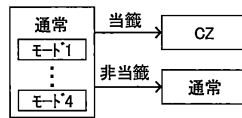
【図 2 0 9】

通常中の遊技の流れ

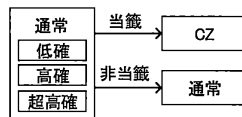
(A)通常⇒CZ⇒ART



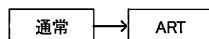
(A-1)通常⇒CZ (ベル当籤時のモードに応じた抽籤)



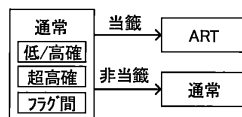
(A-2)通常⇒CZ (抽籤用フラグと状態に応じた抽籤)



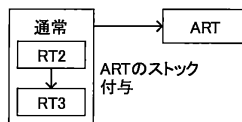
(B)通常⇒ART



(B-1)通常⇒ART (抽籤用フラグと状態に応じた抽籤)



(B-2)通常⇒ART (RT3への移行)



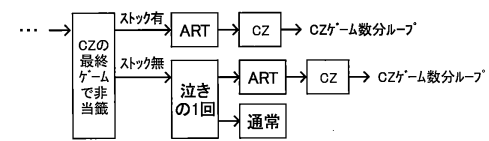
【図 2 1 0】

CZ中の遊技の流れ

(A)CZからARTへの基本的な流れ



※ ループの最初のCZを「初当たりCZ」と呼び、ART後のCZを「継続CZ」と呼ぶ



(B)初当たりCZの概要

残りCZゲーム数	...	2	1	0	終了 次遊技
ART移行契機	フェーズA	フェーズB	フェーズC		

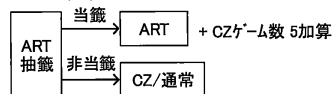
(C)継続CZの概要

残りCZゲーム数	...	2	1	0	0	終了 次遊技
ART移行契機	フェーズA	フェーズD	フェーズC			

【図 2 1 1】

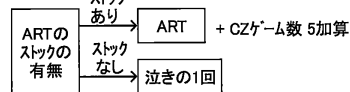
(D)各フェーズの概要

抽籤フェーズA、B、C

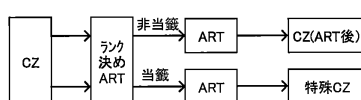


※ 抽籤フェーズ毎に当籤確率が異なる

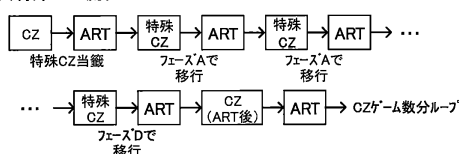
放出フェーズD



(E)継続CZの種別抽籤



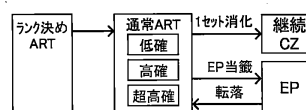
(F)特殊CZの流れ



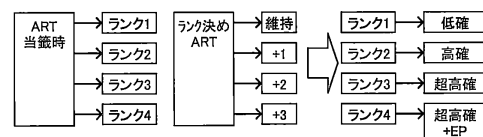
【図 2 1 2】

ART中の遊技の流れ

(A)ARTの概要

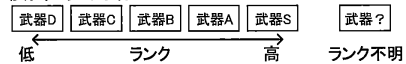


(B)ART中のランク(低確・高確・超高確)の決め方



(C)ランク示唆演出の概要

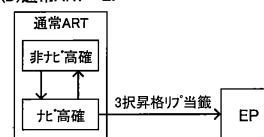
ART移行時のランク示唆



通常ART中の抽籤状態示唆

ARTの抽籤状態	演出ステージ
低確	ステージ1
高確	ステージ2
超高確	ステージ3

(D)通常ART⇒EP



10

20

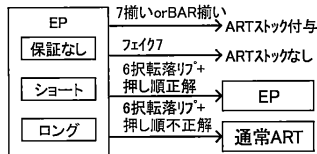
30

40

50

【図 2 1 3】

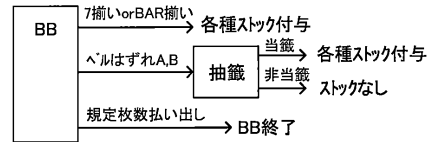
(E)EPの概要



※ 7揃い、BAR揃い又はフェイク7当籤時に転落モード移行(転落)
(ただし、7揃い又はBAR揃いが一度当籤するまでは転落なし)
※ 転落モード「保証なし」では、6枚転落リフの押し順を報知せず、
転落モード「ショート」「ロング」では、6枚転落リフの押し順を報知する。
※ レア役や自力押し順正解時に転落モードの移行(昇格)

【図 2 1 4】

BB中の遊技の流れ



※ NBB中は、ヘルはずれで、各種抽籤
CBB中は、7揃いorBAR揃いで、各種ストック付与
※ 抽籤の種類や、ストックの種類は、状態(通常BBorART中BB)
によって異なる。

【図 2 1 5】

(A) モード移行抽籤テーブル(NBB終了後)

抽籤結果	移行前のモード				偶数設定			
	モード1	モード2	モード3	モード4	モード1	モード2	モード3	モード4
モード1	192	0	0	0	176	0	0	0
モード2	48	192	0	0	48	176	0	0
モード3	12	48	192	0	24	48	176	0
モード4	4	16	64	256	8	32	80	256

(B) モード移行抽籤テーブル(CBB終了時)

抽籤結果	移行前のモード				偶数設定			
	モード1	モード2	モード3	モード4	モード1	モード2	モード3	モード4
モード1	128	0	0	0	96	0	0	0
モード2	96	128	0	0	96	96	0	0
モード3	24	96	128	0	48	96	96	0
モード4	8	32	128	256	16	64	160	256

(C) モード移行抽籤テーブル(初当たりCZ失敗時)

抽籤結果	移行前のモード				偶数設定			
	モード1	モード2	モード3	モード4	モード1	モード2	モード3	モード4
モード1	240	224	192	0	224	192	128	0
モード2	12	24	48	0	24	48	96	0
モード3	3	6	12	0	6	12	24	0
モード4	1	2	4	256	2	4	8	256

(D) モード移行抽籤テーブル(継続CZ終了時)

抽籤結果	奇数設定	偶数設定
	抽籤値	
モード1	192	176
モード2	48	48
モード3	12	24
モード4	4	8

(E) モード移行抽籤テーブル(設定変更時)

抽籤結果	奇数設定	偶数設定
	抽籤値	
モード1	176	128
モード2	64	96
モード3	16	32
モード4	0	0

【図 2 1 6】

モード移行抽籤テーブル(D), (E)

```

dVCMD2TB EQU $
DB .LOW(.wRAND1BS+cRNDAT02)+20H*(3)
DB .LOW.wDDMPAR_X ;0:設定変更時、1:継続CZ終了時
DB .LOW.wEVN_ODD ;設定値 偶奇情報

;
DB (80H)+dVCMD2_A-$ ;0:設定変更時、1:継続CZ終了時
DB (80H)+dVCMD2_B-$

;
dVCMD2_A EQU $ ;設定変更時
DB (80H)+dVCMD1_16-$ ;奇数
DB (80H)+dVCMD1_17-$ ;偶数

;
dVCMD2_B EQU $ ;継続CZ終了時
DB (80H)+dVCMD1_03-$ ;奇数
DB (80H)+dVCMD1_04-$ ;偶数

;
dVCMD1_03 EQU $
DB 0000111B
DB 4 ;B0:モード4
DB 12 ;B1:モード3
DB 48 ;B2:モード2

;
dVCMD1_04 EQU $
DB 0000111B
DB 8 ;B0:モード4
DB 24 ;B1:モード3
DB 48 ;B2:モード2

;
dVCMD1_16 EQU $
DB 0000110B
DB 16 ;B0:モード4
DB 64 ;B1:モード3

;
dVCMD1_17 EQU $
DB 0000110B
DB 32 ;B0:モード4
DB 96 ;B1:モード3

```

10

20

30

40

50

【図 2 1 7】

(A)状態移行抽籤テーブル(低確時用)

抽籤結果	抽籤用フラグ					
	はずれ	通常時、ベル	弱スロ	弱チャリ	強スロ	強チャリ
低確	256	256	128	204	230	0
高確保証なし	0	0	0	0	256	256
高確A	0	0	127	50	0	0
高確B	0	0	0	0	24	0
超高確	0	0	1	2	2	0

(B)状態移行抽籤テーブル(高確保証10未満時用)

抽籤結果	抽籤用フラグ					
	はずれ	通常時、ベル	弱スロ	弱チャリ	強スロ	強チャリ
低確	26	26	0	0	0	0
高確保証なし	230	230	128	204	230	256
高確A	0	0	127	50	0	0
高確B	0	0	0	0	24	0
超高確	0	0	1	2	2	0

※保証1以上ある場合、抽籤の結果として低確に滞在したままとする。

(C)状態移行抽籤テーブル(高確保証10以上時用)

抽籤結果	抽籤用フラグ					
	はずれ	通常時、ベル	弱スロ	弱チャリ	強スロ	強チャリ
低確	0	256	224	240	256	256
高確保証なし	0	0	0	0	0	0
高確A	0	0	0	0	0	0
高確B	0	0	0	0	0	0
超高確	0	0	32	18	16	0

【図 2 1 8】

(D)状態移行抽籤テーブル(超高確時用)

抽籤結果	抽籤用フラグ					
	はずれ	通常時、ベル	弱スロ	弱チャリ	強スロ	強チャリ
低確	0	0	0	0	0	256
高確保証なし	0	0	0	0	0	0
高確A	0	0	224	224	224	0
高確B	0	0	0	0	0	0
超高確	256	256	32	32	32	0

(E)状態移行抽籤テーブル(BB終了時用)

抽籤結果	移行前の抽籤状態	
	低確	高確
低確	128	0
高確保証なし	0	0
高確A	0	0
高確B	124	224
超高確	4	32

(F)状態移行抽籤テーブル(CZ終了時用)

抽籤結果	抽籤値	
	低確	高確
低確	0	0
高確保証なし	0	256
高確A	0	0
高確B	0	0
超高確	0	0

(G)状態移行抽籤テーブル(設定変更時用)

抽籤結果	抽籤値	
	低確	高確
低確	192	0
高確保証なし	0	0
高確A	0	64
高確B	0	0
超高確	0	0

【図 2 1 9】

高確保証ゲーム数抽籤テーブル

抽籤結果	高確A当籤時	高確B当籤時
+10G	4	64
+20G	12	64
+30G	48	128
+50G	192	0

【図 2 2 0】

モード別CZ抽籤テーブル

抽籤結果	低設定 現在のモード				高設定 現在のモード			
	モード1	モード2	モード3	モード4	モード1	モード2	モード3	モード4
非当籤	256	255	250	214	256	254	250	214
当籤	0	1	6	42	0	2	6	42

※ 通常時のベル当籤時にCZ抽籤を行う。
同遊技においてART抽籤に当籤していた場合には、抽籤をしない。

10

20

30

40

50

【図 2 2 1】

```

=====
モード別CZ抽籤テーブル
=====

dNRMVCTB EQU $ ;
DB _LOW.(wRAND1BS+cRANDAT07)+20H*(0) ;LOW.(wRAND1BS+cRANDAT07)+20H*(0)
DB _LOW.wLOW_HIH ;設定値 高低情報
DB _LOW.wVC_MODE ;現在のモード1~4

;

;
DB (80H)+dNRMVCL-$ ;設定値 0:低
DB (80H)+dNRMVCH-$ ;設定値 1:高

;

;
dNRMVCL EQU $ ;低設定
DB cPRB_0 ;[0]VC モード 1
DB cPRB_1 ;[1]VC モード 2
DB cPRB_6 ;[2]VC モード 3
DB cPRB_42 ;[3]VC モード 4

;

;
dNRMVCH EQU $ ;高設定
DB cPRB_0 ;[0]VC モード 1
DB cPRB_2 ;[1]VC モード 2
DB cPRB_6 ;[2]VC モード 3
DB cPRB_42 ;[3]VC モード 4

```

【圖 2 2 2】

抽籤結果	抽籤用フлаг(BB当選時に重なりて当籤した当籤役に依りた抽籤用フлаг)							
	通常型、A	弱カ	弱キカ	弱キスワグ	強カ	強キカ	強キスワグ	強カ
非当籤	256	256	256	256	256	256	256	256
当籤	0	0	0	0	0	0	0	0

(A)BB当籤時狀態別CZ抽籤テーブル(低確時用)

抽籤結果	抽籤用フラグ(8日当選時に重複して当選した当選役に応じた抽籤用フラグ)					
	常務若、人	常務若	常務若	常務若	常務若	常務若
非当選	256	254	254	128	128	128
当選	256	254	254	128	128	128

(B)BB当籤時狀態別CZ抽籤テーブル(高確時用)

	抽籤用ラフ(8月当歳時に重製して当該した当歳役に応じた抽籤用ラフ)						
抽籤結果	運番付、ベル	弱スガリ	強サナリ	強スガリ	強サナリ	強スガリ	強ナリ
非当歳	0	0	0	0	0	0	0
当歳	0	0	0	0	0	0	0

※ 通常時のBB当籤時に重複して当籤した役に応じてCZ抽籤を行う。
同遊技においてART抽籤に当籤していた場合には、抽籤をしない。

【図 2 2 3】

[illegible]

(A) 状態別CZ抽籤テーブル(設定1、2(低設定)、低確時用)

[illegible]

抽籤用フラグ

	抽籤結果								
	はすれ								
	通牌7								
	弱チー								
	強スリ								
	弱チー								
	弱チー								
	強スリ								
	強スリ								
	強ヘル								
	フェイ								
	バー								
当選	1	64	64	64	128	0	0	192	255
非常当選	1	64	64	64	128	0	0	192	255
抽籤結果	255	192	192	192	128	0	0	192	255
はすれ	255	192	192	192	128	0	0	192	255
通過用フラグ	255	192	192	192	128	0	0	192	255

抽籤用フラグ

抽籤結果	強チリ	強スカ	通當ワ	弱スカ	弱チリ	弱チヤスワ	強スカ	強チリ	強チヤスワ	強ヘル	フェケツ	7柄い	BAR柄い
非当籤	256	256	253	253	216	176	176	256	256	256	253	256	256
当籤	0	0	?	?	?	40	80	80	80	80	80	0	0

(D) 状態別の抽籤テーブル(設定5,6(高設定)、低確時用)

	抽籤結果 はすれ	通常賞	頭スチール	頸スチール	強サチール	強サハシノリ	強ヘル	7角い	BAR揃い
当選者	266	256	250	250	178	96	96	250	256
非当選者	0	0	6	6	6	80	160	160	6
当選者	0	0	6	6	6	80	160	160	6

抽籤用フラグ

	押さえてフラグ	抽籤結果 はすれ	通常ワ	強スカ	弱チリ-	弱チスワ	強スカ	強チリ-	強チスワ	強ヘル	フェイ7	7揃い	BAR揃い
当選	252	252	176	176	96	0	0	0	0	176	252	252	
非当選	4	4	80	80	160	256	256	256	256	80	4	4	

抽籤用フラグ

【図 2 2 4】

CZ前兆ゲーム数抽籤テーブル

抽籤結果	BB非常籤時	BB當籤時
0G	0	6
5~8G	0	1
9~12G	16	1
13~20G	48	3
21~28G	176	13
29~30G	16	1
31~34G	0	

※ 当籤した前兆ゲーム数範囲の中から、均等に前兆ゲーム数を決定

【図 2 2 5】

(A)BB当籤時ART抽籤テーブル(NBB当籤、状態：通常or高確時用)												
抽籤用フラグ(BB当籤時に重複して当籤した当籤役に応じた抽籤用フラグ)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	254	254	254	254	254	254	254	254	254	254	254	254
当籤	2	2	2	2	2	2	2	2	2	2	2	2

(B)BB当籤時ART抽籤テーブル(CBB当籤、状態：通常or高確時用)												
抽籤用フラグ(BB当籤時に重複して当籤した当籤役に応じた抽籤用フラグ)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	252	252	252	252	252	252	252	252	252	252	252	252
当籤	4	4	4	4	4	4	4	4	4	4	4	4

(C)BB当籤時ART抽籤テーブル(状態：超高確時orCZ前兆時orART前兆時用)												
抽籤用フラグ(BB当籤時に重複して当籤した当籤役に応じた抽籤用フラグ)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	192	192	192	192	192	192	192	192	192	192	192	192
当籤	64	64	64	64	64	64	64	64	64	64	64	64

※ BB当籤時に重複して当籤した役に応じてART抽籤を行う。

【図 2 2 6】

(A) 通常時ART抽籤テーブル(低設定、状態：低確or高確時用)												
抽籤用フラグ(低設定、通常フラグ時用)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	256	256	256	256	256	256	256	256	256	256	256	256
当籤	0	0	0	0	0	0	0	0	0	0	0	0

(B) 通常時ART抽籤テーブル(低設定、状態：超高確時orCZ前兆時orART前兆時用)												
抽籤用フラグ(低設定、通常フラグ時用)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	256	256	256	256	256	256	256	256	256	256	256	256
当籤	0	0	0	0	0	0	0	0	0	0	0	0

(C) 通常時ART抽籤テーブル(低設定、通常フラグ時用)												
抽籤用フラグ(低設定、通常フラグ時用)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	256	256	256	256	256	256	256	256	256	256	256	256
当籤	0	0	0	0	0	0	0	0	0	0	0	0

(D) 通常時ART抽籤テーブル(高設定、状態：低確or高確時用)												
抽籤用フラグ(高設定、通常フラグ時用)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	256	256	256	256	256	256	256	256	256	256	256	256
当籤	0	0	0	0	0	0	0	0	0	0	0	0

(E) 通常時ART抽籤テーブル(高設定、状態：超高確時orCZ前兆時orART前兆時用)												
抽籤用フラグ(高設定、通常フラグ時用)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	256	256	256	256	256	256	256	256	256	256	256	256
当籤	0	0	0	0	0	0	0	0	0	0	0	0

(F) 通常時ART抽籤テーブル(高設定、通常フラグ時用)												
抽籤用フラグ(高設定、通常フラグ時用)												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	256	256	256	256	256	256	256	256	256	256	256	256
当籤	0	0	0	0	0	0	0	0	0	0	0	0

【図 2 2 7】

抽籤結果	抽籤値
0G	0
5~8G	16
9~12G	0
13~20G	24
21~28G	88
29~30G	32
31~34G	96

※ 通常時にARTに当籤した場合に前兆ゲーム数を抽籤
当籤した前兆ゲーム数範囲の中から、均等に前兆ゲーム数を決定

【図 2 2 8】

(A)CZ中ART抽籤テーブル(初当たりCZ、かつ、残りゲーム以上時用)												
抽籤用フラグ												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	254	0	0	0	0	0	0	0	0	0	0	0
当籤	2	256	256	256	256	256	256	256	256	256	256	256

(B)CZ中ART抽籤テーブル(初当たりCZ、かつ、残りゲーム時用)												
抽籤用フラグ												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	224	0	0	0	0	0	0	0	0	0	0	0
当籤	32	256	256	256	256	256	256	256	256	256	256	256

(C)CZ中ART抽籤テーブル(初当たりCZ終了の次遊技用)												
抽籤用フラグ												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	256	0	0	0	0	0	0	0	0	0	0	0
当籤	0	256	256	256	256	256	256	256	256	256	256	256

(D)CZ中ART抽籤テーブル(CZ(ART後)、かつ、非ナビ高確時用)												
抽籤用フラグ												
抽籤結果	通常ワ、ベル	弱ス缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶	弱チ缶
非当籤	224	255	0	224	0	0	0	0	0	0	224	0
当籤	32	1	256	32	256	256	256	256	256	256	32	256

10

20

30

40

50

【図 2 2 9】

(E)Z中ART抽籤テーブル(GZ(ART後)、かつ、ナビ高確時用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	0	256	0	224	0	0	0	0	0	224	0
当籤	256	1	256	32	256	256	256	256	256	32	256

(F)OZ中ART抽籤テーブル(OZ(ART後)終了の次遊技用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	256	256	0	256	0	0	0	0	0	256	0
当籤	0	0	256	0	256	256	256	256	256	0	256

(G)OZ中ART抽籤テーブル(特殊OZ、かつ、ナビ高確時用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	128	240	0	128	0	0	0	0	0	128	0
当籤	128	16	256	128	256	256	256	256	256	128	256

(H)OZ中ART抽籤テーブル(特殊OZ、かつ、ナビ高確時用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	0	240	0	128	0	0	0	0	0	128	0
当籤	256	16	256	128	256	256	256	256	256	128	256

※ BB当籤時は、本テーブルは参照せず、ART中BB当籤時ストップ抽籤テーブルを参照(必ず当籤)

【図 2 3 0】

(A)ART中ストップ抽籤テーブル(通常ART、かつ、状態:低確or高確、かつ、ナビ高確中用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	256	256	256	256	255	252	252	252	252	256	0
当籤	0	0	0	0	0	1	4	4	4	0	256

(B)ART中ストップ抽籤テーブル(通常ART、かつ、状態:低確or高確、かつ、ナビ高確中用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	256	256	256	256	255	252	252	252	252	256	0
当籤	0	0	0	0	0	1	4	4	4	0	256

(C)ART中ストップ抽籤テーブル(通常ART、かつ、状態:超高確、かつ、ナビ高確中用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	256	256	255	256	256	240	224	224	224	256	0
当籤	0	0	1	0	0	16	32	32	32	0	256

(D)ART中ストップ抽籤テーブル(通常ART、かつ、状態:超高確、かつ、ナビ高確中用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	255	256	255	256	256	240	224	224	224	256	0
当籤	1	0	1	0	0	16	32	32	32	0	256

【図 2 3 1】

(E)ART中ストップ抽籤テーブル(EP中、かつ、転落モード:保証なし時用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	256	256	256	256	255	252	256	252	256	0	0
当籤	0	0	0	0	1	4	0	4	0	256	256

(F)ART中ストップ抽籤テーブル(EP中、かつ、転落モード:ショート時用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	255	256	255	256	256	224	256	224	256	0	0
当籤	1	0	1	0	0	16	32	0	32	0	256

(G)ART中ストップ抽籤テーブル(EP中、かつ、転落モード:ロング時用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	128	224	128	256	256	0	0	256	0	256	0
当籤	128	32	128	0	0	256	256	0	256	0	256

【図 2 3 2】

(H)ART中ストップ抽籤テーブル(ラング決めART、又は、EP準備中)、かつ、ナビ高確中用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	256	256	255	224	192	192	192	256	0	0	0
当籤	0	1	0	1	32	64	64	64	0	256	256

(I)ART中ストップ抽籤テーブル(ラング決めART、又は、EP準備中)、かつ、ナビ高確中用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	255	256	255	224	192	256	192	256	0	0	0
当籤	1	0	1	32	64	0	64	0	256	256	256

(J)ART中ストップ抽籤テーブル(ART準備中用)

抽籤用フラグ											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	256	256	256	255	252	252	252	256	256	256	0
当籤	0	0	0	0	1	4	4	4	0	0	0

(K)ART中ストップ抽籤テーブル(ART中フラグ間用)

抽籤用フラグ(持ち越しているBBと重複している当籤役に応じた抽籤用フラグ)											
抽籤結果	はずれ	通常ワ、ベル	弱スハ	弱チー	弱チン	弱チン	強チン	強チン	強チン	強チン	BAR揃い
非当籤	256	256	255	256	256	240	224	224	224	256	0
当籤	0	1	0	0	0	16	32	32	32	0	256

※ 同遊技がBB当籤(初当たり)時である場合には、抽籤をしない。

【図 2 3 3】

(A)ART 中 BB 当籤時ストック抽籤テーブル (状態: 低確 or 高確、又は、EP 中の転落モード: 保証なし時用)

抽籤結果	抽籤用フラグ (BB 当籤時に重複して当籤した当籤役に応じた抽籤用フラグ)									
	通常ワ、ベル	強スカ	弱チロー	弱チハカワ	強スカ	強チロー	強チハカワ	強ベル	ワチ目ワ	確定チロー
非当籤	252	252	252	252	252	252	252	252	4	0
当籤	4	4	4	4	4	4	4	4	4	256

(B)ART 中 BB 当籤時ストック抽籤テーブル (状態: 超高確、又は、EP 中の転落モード: ショート時用)

抽籤結果	抽籤用フラグ (BB 当籤時に重複して当籤した当籤役に応じた抽籤用フラグ)									
	通常ワ、ベル	強スカ	弱チロー	弱チハカワ	強スカ	強チロー	強チハカワ	強ベル	ワチ目ワ	確定チロー
非当籤	224	224	224	224	224	224	224	224	0	0
当籤	32	32	32	32	32	32	32	32	32	256

(C)ART 中 BB 当籤時ストック抽籤テーブル (ランク決め ART、又は、EP 準備中時用)

抽籤結果	抽籤用フラグ (BB 当籤時に重複して当籤した当籤役に応じた抽籤用フラグ)									
	通常ワ、ベル	強スカ	弱チロー	弱チハカワ	強スカ	強チロー	強チハカワ	強ベル	ワチ目ワ	確定チロー
非当籤	192	192	192	192	192	192	192	192	192	0
当籤	64	64	64	64	64	64	64	64	64	256

(D)ART 中 BB 当籤時ストック抽籤テーブル (CZ 又は CZ 終了の次選択)、又は、EP 中の転落モード: ロング時用)

抽籤結果	抽籤用フラグ (BB 当籤時に重複して当籤した当籤役に応じた抽籤用フラグ)									
	通常ワ、ベル	強スカ	弱チロー	弱チハカワ	強スカ	強チロー	強チハカワ	強ベル	ワチ目ワ	確定チロー
非当籤	0	0	0	0	0	0	0	0	0	0
当籤	256	256	256	256	256	256	256	256	256	256

(E)ART 中 BB 当籤時ストック抽籤テーブル (ART 準備中用)

抽籤結果	抽籤用フラグ (BB 当籤時に重複して当籤した当籤役に応じた抽籤用フラグ)									
	通常ワ、ベル	強スカ	弱チロー	弱チハカワ	強スカ	強チロー	強チハカワ	強ベル	ワチ目ワ	確定チロー
非当籤	252	252	252	252	252	252	252	252	252	0
当籤	4	4	4	4	4	4	4	4	4	256

【図 2 3 4】

ART 当籤時ストック数抽籤テーブル

抽籤結果	低設定 当籤契機				高設定 当籤契機			
	初当り	上乗せ A	上乗せ B	上乗せ C	初当り	上乗せ A	上乗せ B	上乗せ C
ストック+1	0	252	254	224	0	248	252	192
ストック+2	248	4	2	32	224	8	4	64
ストック+3	8	0	0	0	32	0	0	0

当籤契機	内容
初当り	通常中、CZ 前兆中、ART 前兆中、初当たり CZ 中、通常フラグ間、又は、通常 BB 中の ART 当籤時
上乗せ A	EP 中の ART 当籤時
上乗せ B	継続 CZ (CZ (ART 後) 又は特殊 CZ) 中の ART 当籤時
上乗せ C	上記以外の ART 当籤時

10

20

【図 2 3 5】

特殊 CZ 移行抽籤テーブル

抽籤結果	ART ストック	
	なし	あり
非当籤	255	254
当籤	1	2

※ ランク決め ART 開始時に抽籤を行う。
ART ストックなし時に当籤すると、ストック+1

【図 2 3 6】

(A)ART 当籤時ランク抽籤テーブル (当籤タイミング: 通常時、CZ 前兆時、ART 前兆時、又は、EP 以外の ART 中)

抽籤結果	抽籤用フラグ (BB 非当籤時)									
	はずれ	強スカ	弱チロー	弱チハカワ	強スカ	強チロー	強チハカワ	強ベル	ワチ目ワ	確定チロー
ランク1	192	192	192	192	192	192	192	192	192	0
ランク2	58	58	58	58	58	58	58	58	58	0
ランク3	5	5	5	5	5	5	5	5	5	230
ランク4	1	1	1	1	1	1	1	1	1	26

(B)ART 当籤時ランク抽籤テーブル (当籤タイミング: 通常時、CZ 前兆時、ART 前兆時、又は、EP 以外の ART 中)

抽籤結果	抽籤用フラグ (NEB と重複して当籤した当籤役に応じた抽籤用フラグ)									
	通常ワ、ベル	強スカ	弱チロー	弱チハカワ	強スカ	強チロー	強チハカワ	強ベル	ワチ目ワ	確定チロー
ランク1	192	192	192	192	192	192	192	192	192	0
ランク2	58	58	58	58	58	58	58	58	58	230
ランク3	5	5	5	5	5	5	5	5	5	24
ランク4	1	1	1	1	1	1	1	1	1	2

(C)ART 当籤時ランク抽籤テーブル (当籤タイミング: 通常時、CZ 前兆時、ART 前兆時、又は、EP 以外の ART 中)

抽籤結果	抽籤用フラグ (CBB と重複して当籤した当籤役に応じた抽籤用フラグ)									
	はずれ、通常ワ、ベル	強スカ	弱チロー	弱チハカワ	強スカ	強チロー	強チハカワ	強ベル	ワチ目ワ	確定チロー
ランク1	128	128	128	128	128	128	128	128	128	0
ランク2	116	116	116	116	116	116	116	116	116	230
ランク3	10	10	10	10	10	10	10	10	10	24
ランク4	2	2	2	2	2	2	2	2	2	2

30

40

50

【図 2 3 7】

(D)ART 当籤時ラング抽籤テーブル(当籤タイミング:ART準備中)

抽籤結果	抽籤用フラグ (BB非当籤時)			
	強スレ	強スレ	強スレ	強スレ
ラング1	192	192	192	192
ラング2	58	58	58	58
ラング3	5	5	5	5
ラング4	1	1	1	1

(E)ART 当籤時ラング抽籤テーブル(当籤タイミング:ART準備中)

抽籤結果	抽籤用フラグ (BBと重複して当籤した当籤役に応じた抽籤用フラグ)											
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	192	192	192	192	192	192	192	192	192	192	192	192
ラング2	58	58	58	58	58	58	58	58	58	58	58	58
ラング3	5	5	5	5	5	5	5	5	5	5	5	5
ラング4	1	1	1	1	1	1	1	1	1	1	1	1

(F)ART 当籤時ラング抽籤テーブル(当籤タイミング:ART準備中)

抽籤結果	抽籤用フラグ (BBと重複して当籤した当籤役に応じた抽籤用フラグ)											
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	128	128	128	128	128	128	128	128	128	128	128	128
ラング2	116	116	116	116	116	116	116	116	116	116	116	116
ラング3	10	10	10	10	10	10	10	10	10	10	10	10
ラング4	2	2	2	2	2	2	2	2	2	2	2	2

【図 2 3 8】

(G)ART 当籤時ラング抽籤テーブル(当籤タイミング:EP中)

抽籤結果	抽籤用フラグ (BB非当籤時)							
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	192	192	192	192	192	192	192	192
ラング2	58	58	58	58	58	58	58	58
ラング3	5	5	5	5	5	5	5	5
ラング4	1	1	1	1	1	1	1	1

(H)ART 当籤時ラング抽籤テーブル(当籤タイミング:EP中)

抽籤結果	抽籤用フラグ (BBと重複して当籤した当籤役に応じた抽籤用フラグ)											
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	128	128	128	128	128	128	128	128	128	128	128	128
ラング2	116	116	116	116	116	116	116	116	116	116	116	116
ラング3	10	10	10	10	10	10	10	10	10	10	10	10
ラング4	2	2	2	2	2	2	2	2	2	2	2	2

(I)ART 当籤時ラング抽籤テーブル(EP中)

抽籤結果	抽籤用フラグ (BBと重複して当籤した当籤役に応じた抽籤用フラグ)											
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	0	0	0	0	0	0	0	0	0	0	0	0
ラング2	230	230	230	230	230	230	230	230	230	230	230	230
ラング3	24	24	24	24	24	24	24	24	24	24	24	24
ラング4	2	2	2	2	2	2	2	2	2	2	2	2

【図 2 3 9】

(J)ART 当籤時ラング抽籤テーブル(当籤タイミング:CZ中)

抽籤結果	抽籤用フラグ (BB非当籤時)											
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	230	230	230	230	230	230	230	230	230	230	230	230
ラング2	24	24	24	24	24	24	24	24	24	24	24	24
ラング3	1	1	1	1	1	1	1	1	1	1	1	1
ラング4	1	1	1	1	1	1	1	1	1	1	1	1

(K)ART 当籤時ラング抽籤テーブル(当籤タイミング:CZ中)

抽籤結果	抽籤用フラグ (BBと重複して当籤した当籤役に応じた抽籤用フラグ)											
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	128	128	128	128	128	128	128	128	128	128	128	128
ラング2	116	116	116	116	116	116	116	116	116	116	116	116
ラング3	10	10	10	10	10	10	10	10	10	10	10	10
ラング4	2	2	2	2	2	2	2	2	2	2	2	2

(L)ART 当籤時ラング抽籤テーブル(当籤タイミング:CZ中)

抽籤結果	抽籤用フラグ (BBと重複して当籤した当籤役に応じた抽籤用フラグ)											
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	128	128	128	128	128	128	128	128	128	128	128	128
ラング2	116	116	116	116	116	116	116	116	116	116	116	116
ラング3	10	10	10	10	10	10	10	10	10	10	10	10
ラング4	2	2	2	2	2	2	2	2	2	2	2	2

【図 2 4 0】

(M)ART 当籤時ラング抽籤テーブル(当籤タイミング:フラグ開状態)

抽籤結果	抽籤用フラグ (持ち越しているBBと重複している当籤役に応じた抽籤用フラグ)							
	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	192	192	192	192	192	192	192	192
ラング2	58	58	58	58	58	58	58	58
ラング3	5	5	5	5	5	5	5	5
ラング4	1	1	1	1	1	1	1	1

(N)ART 当籤時ラング抽籤テーブル(当籤タイミング:BB中)

抽籤結果	抽籤用フラグ			
	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	192	192	192	192
ラング2	58	58	58	58
ラング3	5	5	5	5
ラング4	1	1	1	1

(O)ART 当籤時ラング抽籤テーブル(当籤タイミング:複数ストックした場合の2つ目以降)

抽籤結果	抽籤用フラグ			
	通常スレ	通常スレ	通常スレ	通常スレ
ラング1	230	230	230	230
ラング2	24	24	24	24
ラング3	1	1	1	1
ラング4	1	1	1	1

【図 2 4 1】

ストック放出順序抽籤テーブル

抽籤結果	抽籤値
4⇒3⇒2⇒1	64
4⇒3⇒1⇒2	192

【図 2 4 2】

(A)ラング決めART中ラング昇格抽籤テーブル(非ナビ高確時用)

抽籤結果	抽籤用フラグ (EB非当確時)											
	はずれ	通常バ	ベル	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
維持	192	254	0	192	0	0	0	0	0	0	192	0
ランク+1	62	0	248	62	248	248	248	248	248	62	192	192
ランク+2	1	1	7	1	7	7	7	7	7	1	56	56
ランク+3	1	1	1	1	1	1	1	1	1	1	1	8

(B)ラング決めART中ラング昇格抽籤テーブル(ナビ高確時用)

抽籤結果	抽籤用フラグ (EB非当確時)											
	はずれ	通常バ	ベル	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
維持	0	254	0	192	0	0	0	0	0	0	192	0
ランク+1	248	0	248	62	248	248	248	248	248	62	192	192
ランク+2	7	7	1	7	1	7	7	7	7	1	56	56
ランク+3	1	1	1	1	1	1	1	1	1	1	1	8

(C)ラング決めART中ラング昇格抽籤テーブル(BB当確時用)

抽籤結果	抽籤用フラグ (BBと重複して当籤したとき抽籤用フラグ)											
	はずれ	通常バ	ベル	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
維持	0	0	0	0	0	0	0	0	0	0	0	0
ランク+1	192	192	192	192	192	192	192	192	192	192	192	0
ランク+2	56	56	56	56	56	56	56	56	56	56	56	128
ランク+3	8	8	8	8	8	8	8	8	8	8	8	128

【図 2 4 3】

(A)ナビ高確ゲーム数獲得抽籤テーブル(通常ART、かつ、状態:低確時用)

抽籤結果	抽籤用フラグ											
	はずれ	通常バ	ベル	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
非当籤	255	224	255	255	255	192	192	192	255	255	255	256
ナビ高確+5G	0	0	28	0	0	0	56	56	0	0	0	0
ナビ高確+10G	1	0	4	1	1	1	8	8	8	1	0	0

(B)ナビ高確ゲーム数獲得抽籤テーブル(通常ART、かつ、状態:高確時用)

抽籤結果	抽籤用フラグ											
	はずれ	通常バ	ベル	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
非当籤	224	255	224	224	252	128	128	128	224	256	256	256
ナビ高確+5G	28	1	28	28	0	0	112	112	112	28	0	0
ナビ高確+10G	4	0	4	4	4	16	16	16	16	4	0	0

(C)ナビ高確ゲーム数獲得抽籤テーブル(通常ART、かつ、状態:超高確時用)

抽籤結果	抽籤用フラグ											
	はずれ	通常バ	ベル	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
非当籤	128	243	128	128	0	0	0	0	128	256	256	256
ナビ高確+5G	13	112	112	224	224	128	128	128	128	112	0	0
ナビ高確+10G	16	0	16	16	32	32	128	128	128	16	0	0

【図 2 4 4】

(A)CZゲーム数獲得抽籤テーブル(ART準備中用)

抽籤結果	抽籤用フラグ											
	はずれ	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
非当籤	256	192	0	128	0	0	0	0	0	0	256	0
CZ+5G	0	56	224	112	224	0	0	0	0	0	0	0
CZ+10G	0	6	28	14	28	224	224	224	224	0	0	0
CZ+20G	0	1	3	1	3	28	28	28	28	0	0	0
CZ+30G	0	1	1	1	1	4	4	4	4	0	0	0

(B)CZゲーム数獲得抽籤テーブル(通常ART、かつ、非ナビ高確時用)

抽籤結果	抽籤用フラグ											
	はずれ	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
非当籤	256	224	256	192	128	0	0	0	0	0	0	0
CZ+5G	0	28	0	56	112	224	224	224	224	224	224	224
CZ+10G	0	2	0	6	14	28	28	28	28	28	28	28
CZ+20G	0	1	0	1	1	3	3	3	3	3	3	3
CZ+30G	0	1	0	1	1	1	1	1	1	1	1	1

(C)CZゲーム数獲得抽籤テーブル(通常ART、かつ、ナビ高確時用)

抽籤結果	抽籤用フラグ											
	はずれ	弱スガ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ	弱チリ
非当籤	224	224	256	192	128	0	256	0	0	0	0	0
CZ+5G	28	28	0	56	112	224	0	224	224	224	224	224
CZ+10G	2	2	0	6	14	28	0	28	28	28	28	28
CZ+20G	1	1	0	1	1	3	0	3	3	3	3	3
CZ+30G	1	1	0	1	1	1	0	1	1	1	1	1

※ CZ中の「A」「B」「C」で当籤又は
CZ中の「A」「B」「C」でストック放出すると
CZゲーム数獲得

10

20

30

40

50

【図 2 4 5】

ARTゲーム数獲得抽籤テーブル(ART中、ART準備中、継続OZ中用)

抽籤結果	抽籤用フラグ	
	NBB	CBB
ART+10G	224	0
ART+20G	28	224
ART+30G	4	32

※ ART中のBB当籤でARTゲーム数を獲得

【図 2 4 6】

転落モード抽籤テーブル

抽籤結果	EPの当籤タイミング	
	ランク決めART中	ランク決めART中以外
保証なし	0	0
ショート	254	192
ロング	2	64

10

【図 2 4 7】

(A)転落モード移行抽籤テーブル(転落モード:保証なし時用)

抽籤結果	抽籤用フラグ									
	はずれ	通常J'ベル	強入付	強入付	強入付	強入付	強入付	強入付	強入付	EP継続時
保証なし	192	256	192	0	0	0	0	256	256	0
ショート	63	0	63	252	240	240	0	0	0	255
ロング	1	0	1	4	16	16	0	0	0	1

※ EP継続時:保証なし時の6枚転落J'P当籤において押し順に正解した時

(B)転落モード移行抽籤テーブル(転落モード:ショート時用)

抽籤結果	抽籤用フラグ									
	はずれ	通常J'ベル	強入付	強入付	強入付	強入付	強入付	強入付	強入付	BAR揃い
保証なし	0	0	0	0	0	0	0	248	248	248
ショート	248	256	248	224	192	192	8	8	8	8
ロング	8	0	8	32	64	64	0	0	0	0

(C)転落モード移行抽籤テーブル(転落モード:ロング時用)

抽籤結果	抽籤用フラグ									
	はずれ	通常J'ベル	強入付	強入付	強入付	強入付	強入付	強入付	強入付	BAR揃い
保証なし	0	0	0	0	0	0	0	52	0	0
ショート	0	0	0	0	0	0	0	0	0	0
ロング	256	256	256	256	256	256	204	256	256	256

※ BB当籤時は抽籤を行わない。

【図 2 4 8】

EP中BB当籤時転落モード移行抽籤テーブル

抽籤結果	NBB当籤時の現在のモード			CBB当籤時の現在のモード		
	保証なし	ショート	ロング	保証なし	ショート	ロング
保証なし	0	0	0	0	0	0
ショート	240	192	0	224	128	0
ロング	16	64	256	32	128	256

20

30

40

50

【図 2 4 9】

(A)BB中ART抽籤テーブル(通常BB、又は、ART中BB中用)

抽籤用フラグ									
抽籤結果	CBBヘル	中段ヘル	斜めヘル	ヘルハスUA	ヘルハスUB	フレイ7	フレイ7	フレイ7	BAR抽い
非当籤	256	256	256	256	252	252	252	0	0
当籤	0	0	0	0	4	4	0	256	256

※ 通常BB中に当籤した場合は、次遊技からART中BB

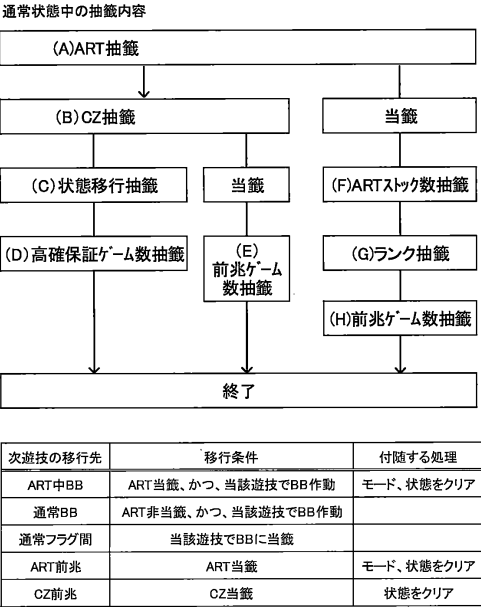
(B)BB中CZ抽籤テーブル(通常BB中用)

抽籤用フラグ									
抽籤結果	CBBヘル	中段ヘル	斜めヘル	ヘルハスUA	ヘルハスUB	フレイ7	フレイ7	フレイ7	BAR抽い
非当籤	256	256	256	256	146	146	256	256	256
当籤	0	0	0	0	110	110	0	0	0

(C)BB中EPストック抽籤テーブル(ART中BB中用)

抽籤用フラグ									
抽籤結果	CBBヘル	中段ヘル	斜めヘル	ヘルハスUA	ヘルハスUB	フレイ7	フレイ7	フレイ7	BAR抽い
非当籤	256	256	256	256	146	146	256	0	0
当籤	0	0	0	0	110	110	0	256	256

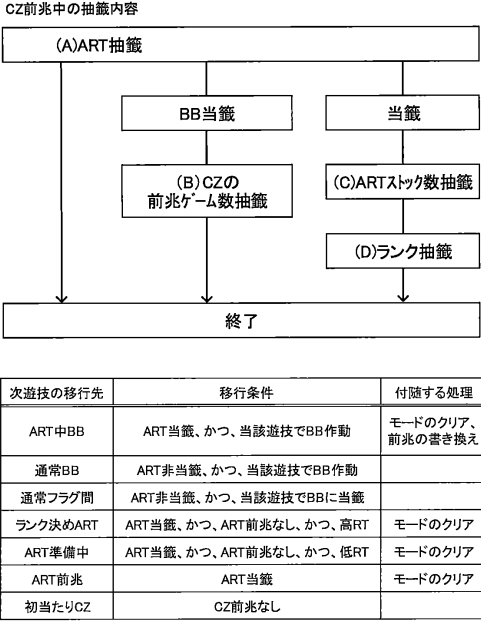
【図 2 5 0】



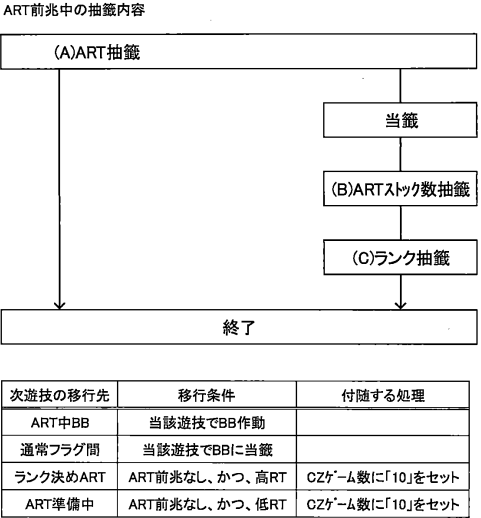
10

20

【図 2 5 1】



【図 2 5 2】

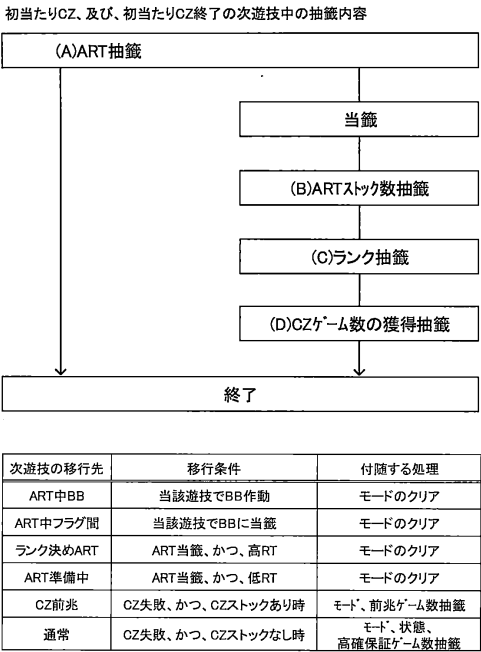


30

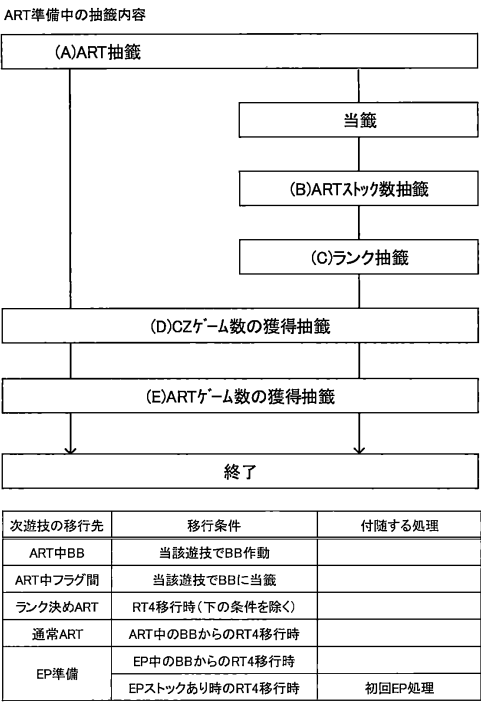
40

50

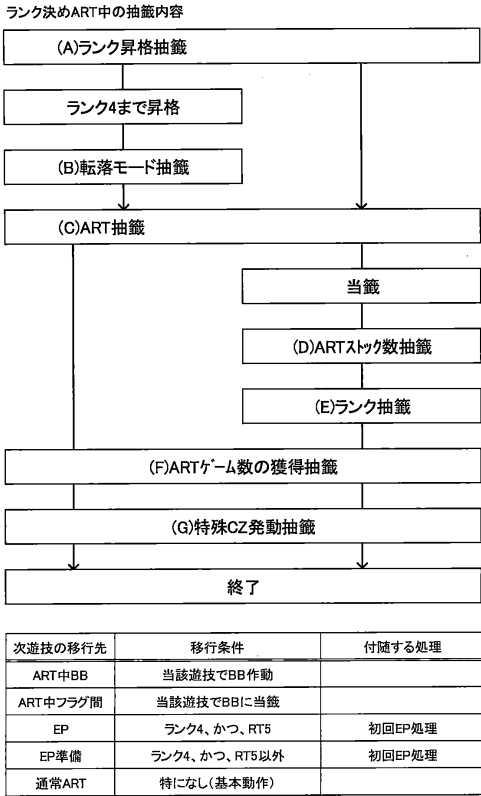
【図 2 5 3】



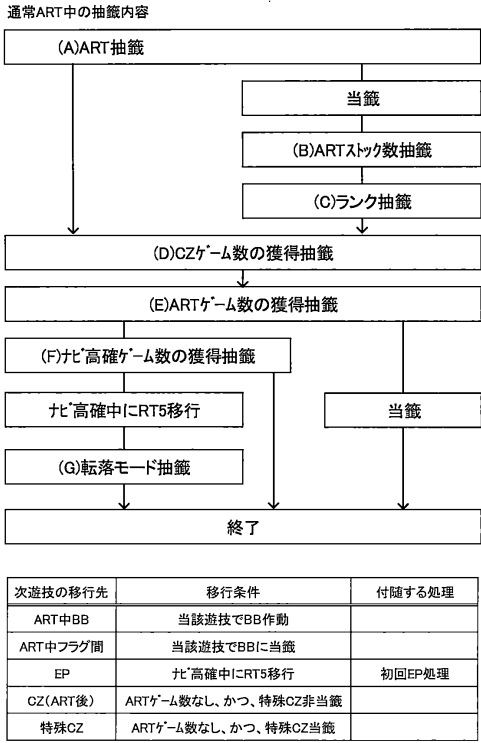
【図 2 5 4】



【図 2 5 5】



【図 2 5 6】



10

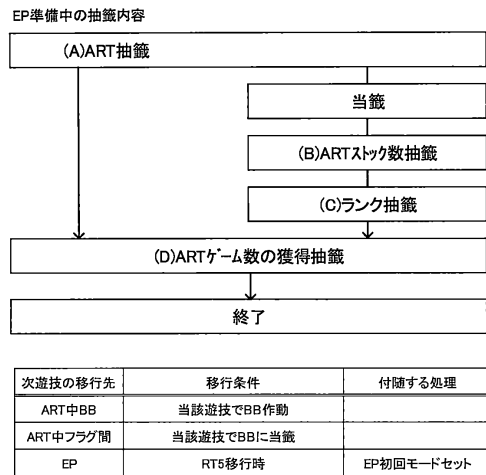
20

30

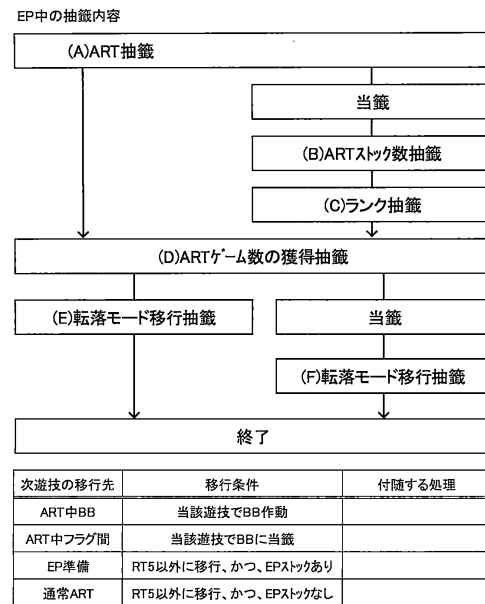
40

50

【図 2 5 7】



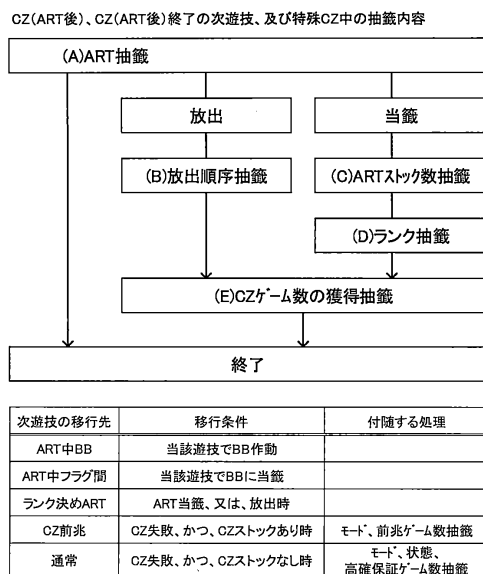
【図 2 5 8】



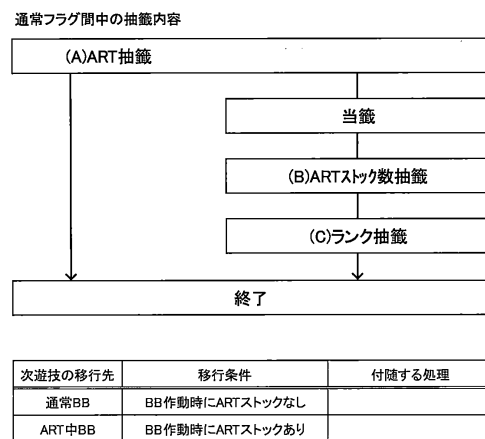
10

20

【図 2 5 9】



【図 2 6 0】

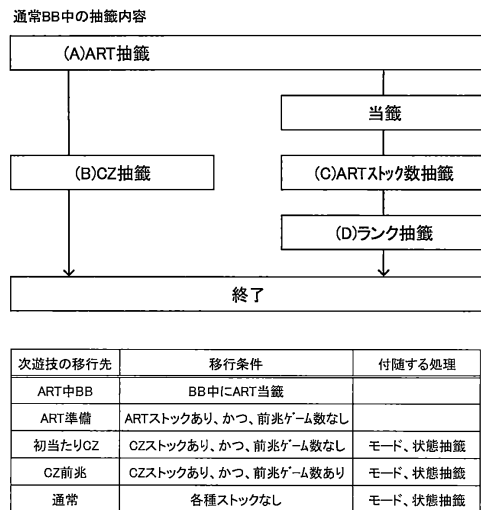


30

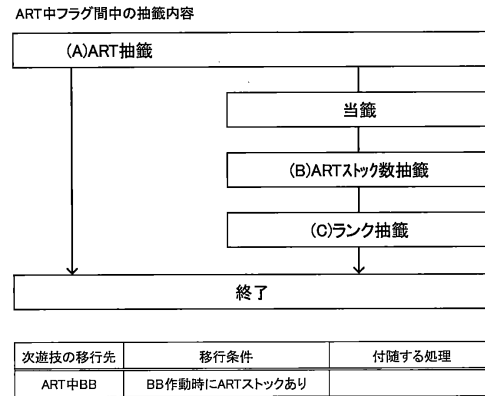
40

50

【図 2 6 1】



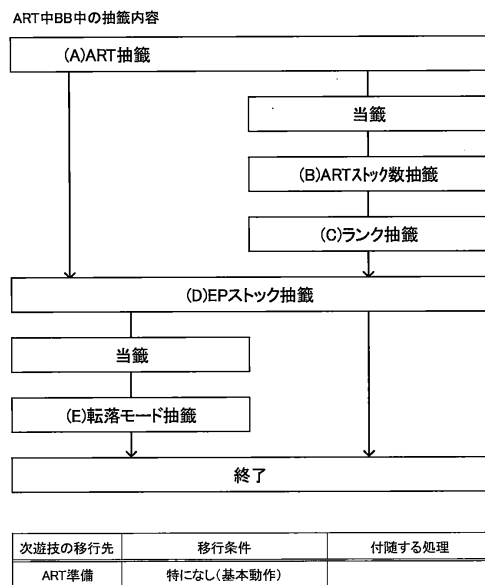
【図 2 6 2】



10

20

【図 2 6 3】



【図 2 6 4】

(A) CZの残りゲーム数と各抽籤との関係例1

残りCZゲーム数	...	8	7	6	5	4	3	2	1	0
ART抽籤	...	テーブルA	テーブルB	テーブルC	テーブルD	テーブルE				
ランク抽籤	...	テーブルV	テーブルW	テーブルX	テーブルY	テーブルZ				

(B) CZの残りゲーム数と各抽籤との関係例2

残りCZゲーム数	...	8	7	6	5	4	3	2	1	0
ART抽籤	...	テーブルA	テーブルB	テーブルC	テーブルD	テーブルE				
ランク抽籤	...	テーブルX	テーブルY	テーブルZ						

30

40

50

【図 2 6 5】

(A) GZ中のストックの有無と各抽籤との関係例1

残りGZゲーム数	...	8	7	6	5	4	3	2	1	0
ストックあり時										
ART抽籤	テーブルA									
ランク抽籤	テーブルZ									
ストックなし時										
ART抽籤	テーブルB									
ランク抽籤	テーブルY									

(B) GZ中のストックの有無と各抽籤との関係例2

残りGZゲーム数	...	8	7	6	5	4	3	2	1	0
ストックあり時										
ART抽籤	...	テーブルA	テーブルB	テーブルC	テーブルD	テーブルE				
ランク抽籤	...	テーブルV	テーブルW	テーブルX	テーブルY	テーブルZ				
ストックなし時										
ART抽籤	...	テーブルF	テーブルG	テーブルH	テーブルI	テーブルJ				
ランク抽籤	...	テーブルQ	テーブルR	テーブルS	テーブルT	テーブルU				

(C) GZ中のストックの有無と各抽籤との関係例3

残りGZゲーム数	...	8	7	6	5	4	3	2	1	0				
ストックあり時														
ART抽籤	...	テーブルA	テーブルB	テーブルC	テーブルD	テーブルE								
ランク抽籤	...	テーブルX	テーブルY	テーブルZ										
ストックなし時														
ART抽籤	...	テーブルF	テーブルG	テーブルH										
ランク抽籤	...	テーブルU	テーブルV	テーブルW										

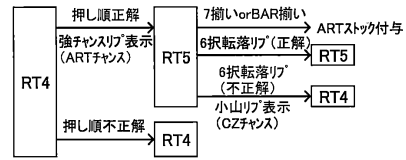
【図 2 6 7】

押し順ベル当籤時の押し順と図柄組合せとの対応関係

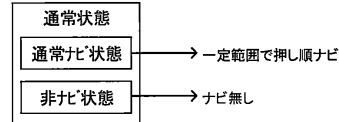
当籤役	123	132	213	231	312	321
F.123ベルA	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.123ベルB	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.132ベルA	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.132ベルB	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.213ベルA	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.213ベルB	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.231ベルA	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.231ベルB	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.312ベルA	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.312ベルB	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.321ベルA	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル
F.321ベルB	C.左移行役	C.右移行役	C.ODベル	C.ODベル	C.ODベル	C.ODベル

【図 2 6 6】

(A) 通常状態中のRT4,RT5におけるART抽籤



(B) 通常状態中のナビ制御



10

20

【図 2 6 8】

(A) 押し順ベル当籤時のRT制御

当籤役	123	132	213	231	312	321
F.123ベルA	8枚	RT0移行	1/4で8枚 3/4でRT2移行			
F.123ベルB	RT0移行	8枚				
F.132ベルA	1/4で8枚 3/4でRT2移行		8枚	RT0移行	1/4で8枚 3/4でRT2移行	
F.132ベルB			RT0移行	8枚		
F.213ベルA	1/4で8枚 3/4でRT2移行				8枚	RT0移行
F.213ベルB					RT0移行	8枚
F.231ベルA					8枚	RT0移行
F.231ベルB					RT0移行	8枚

30

(B) 押し順ベル当籤時のRT制御の別例1

当籤役	123	132	213	231	312	321
F.123ベルA	8枚	1/4で8枚 3/4でRT2移行	RT0移行			
F.123ベルB	1/4で8枚 3/4でRT2移行		8枚	RT0移行		
F.132ベルA			RT0移行			
F.132ベルB			8枚	1/4で8枚 3/4でRT2移行	RT0移行	
F.132ベルC						
F.132ベルD						
F.213ベルA	RT0移行		8枚			
F.213ベルB						
F.213ベルC						
F.213ベルD						
...	...					

40

50

【図 2 6 9】

(C) 押し順ヘル当籤時のRT制御の別例2

当番役	図柄組合せ(押し順毎)					
	123	132	213	231	312	321
F.123△A	8枚	1/4で8枚	1/4で8枚 3/4でRT0移行	1/4で8枚 3/4でRT0移行		
F.123△B		3/4でRT2移行				
F.123△C						
F.123△D						
F.132△A	1/4で8枚 3/4でRT2移行	8枚		1/4で8枚 3/4でRT0移行		
F.132△B						
F.132△C						
F.132△D						
F.213△A	1/4で8枚 3/4でRT0移行		8枚	1/4で8枚 3/4でRT2移行	1/4で8枚 3/4でRT0移行	
F.213△B						
F.213△C						
F.213△D						
...	...					

(D) 押し順ヘル当籤時のRT制御の別例3

当番役	図柄組合せ(押し順毎)					
	123	132	213	231	312	321
F.123 ^a ↗A	8枚	1/2で8枚 1/2でRT2移行	1/4で8枚 3/4でRT0移行		1枚	
F.123 ^a ↘B						
F.123 ^a ↗C						
F.123 ^a ↘D						
F.132 ^a ↗A	1/2で8枚 1/2でRT2移行	8枚	1/2で8枚 1/2でRT2移行	1枚	1/4で8枚 3/4でRT0移行	
F.132 ^a ↘B						
F.132 ^a ↗C						
F.132 ^a ↘D						
F.213 ^a ↗A	1/4で8枚 3/4でRT0移行	8枚	1/2で8枚 1/2でRT2移行	1枚	1/4で8枚 3/4でRT0移行	
F.213 ^a ↘B						
F.213 ^a ↗C						
F.213 ^a ↘D						
...	...					

【図 270】

(A) ナビ高確ゲーム数の更新例1

[illegible]

(B) ナビ高確ゲーム数の更新例2

遊技仕様	通常ART	通常ART	通常ART	通常ART	通常ART	EP	EP	EP	通常ART
ナゾ罐ゲーム数	0	0	5	4	0	0	0	0	...
変動	0	+5	-1	-1	0	0	0	0	...
内部リセ役	3枚昇格	F. 逆パター	6枚維持	3枚昇格	F. 逆パター	F. 777A	...	6枚逆転	押し順ヘル
抽選用フラガ	通常	通常	通常	通常	通常	通常	...	弱チヤヘル	ヘル
抽選用フラ	DTA	DTA	DTA	DTA	DTA	DTA	...	DTA	DTA

(C) ナビ高確ゲーム数の更新例3

遊技状態	通常ART	通常ART	通常ART	通常ART	緑線CZ	ワン決めART	通常ART
ナニが爆弾〜人数	0	0	5	4	3=0	0	...
変動	0	+	-	-	0	0	...
内部の裏役	3択実格1	F.選エー	6枚維持1	3枚実格1	F.選エー	F.選エー	押しヘッ
抽選用フラグ	通常1	通常1	通常1	通常1	通常1	通常1	押しヘッ
RT球機	RT4	RT4	RT4	RT4	RT4	RT5	RT5

【図 2 7 1】

(A) ナビ高確の有無と継続CZ中のART抽籤(3択昇格リブ当籤時)

遊技状態	継続CZ	
ナビ高確ゲーム数	なし	あり
抽籤用フラグ	通常リブ	強チャンスリブ
ART抽籤	低確率	当籤
次遊技のRT状態	RT4	RT5

(B) 七・高確の有無とランク決めART中の昇格抽籤1(3択昇格リブ当籤時)

遊技状態	ランク決めART	
ナビ高確ゲーム数	なし	あり
抽籤用フラグ	通常リブ	強チャンスリブ
昇格抽籤	維持	+1以上

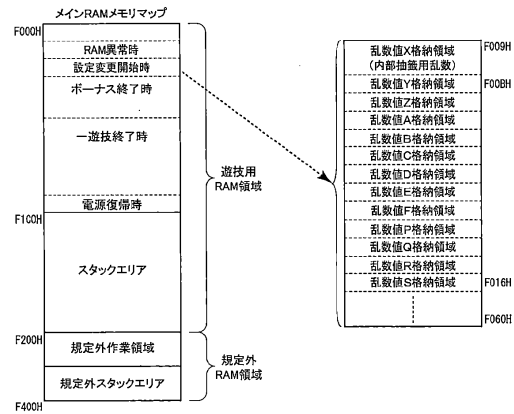
(C) ナビ高確の有無とランク決めART中の昇格抽籤2(はずれ時)

遊技状態	ランク決めART	
ナビ高確ゲーム数	なし	あり
抽籤用フラグ	はずれ	はずれ
昇格抽籤	維持	+1以上

(D) RT状態とランク決めART中の昇格抽籤

遊技状態	ランク決めART	
RT状態	RT4	RT5
抽籤用フラグ	フェイク7揃い、BAR揃い	
昇格抽籤	昇格確率高い	
フラグの当籤確率	低い	高い

【図 2 7 2】



【図 273】

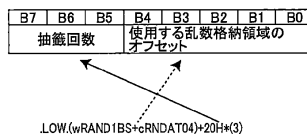
乱数格納領域(RAM)の配置					
行番	アドレス	ラベル	疑似命令	値	コメント
15958	F009	wRANDOMX:	DS	2	; 乱数値 X 格納領域 (内部抽籤用乱数)
15959	F00B	wRAND1BS	EQU	\$	
15961	F00B	wRANDOMY:	DS	1	; 乱数値 Y 格納領域 (出玉抽籤用乱数)
15962	F00C	wRANDOMZ:	DS	1	; 乱数値 Z 格納領域 (出玉抽籤用乱数)
15964	F00D	wRANDOMA:	DS	1	; 乱数値 A 格納領域 (出玉抽籤用乱数)
15965	F00E	wRANDOMB:	DS	1	; 乱数値 B 格納領域 (出玉抽籤用乱数)
15966	F00F	wRANDOMC:	DS	1	; 乱数値 C 格納領域 (出玉抽籤用乱数)
15967	F010	wRANDOMD:	DS	1	; 乱数値 D 格納領域 (出玉抽籤用乱数)
15968	F011	wRANDOMF:	DS	1	; 乱数値 E 格納領域 (出玉抽籤用乱数)
15969	F012	wRANDOMF:	DS	1	; 乱数値 F 格納領域 (出玉抽籤用乱数)
15971	F013	wRANDOMP:	DS	1	; 乱数値 P 格納領域 (出玉抽籤用乱数)
15972	F014	wRANDOMQ:	DS	1	; 乱数値 Q 格納領域 (出玉抽籤用乱数)
15973	F015	wRANDOMR:	DS	1	; 乱数値 R 格納領域 (出玉抽籤用乱数)
15974	F016	wRANDOMS:	DS	1	; 乱数値 S 格納領域 (出玉抽籤用乱数)

【図 274】

状態移行抽籤テーブル(BB終了時用)			
dBDMCGB	EQU	\$	
DB		LOW(wRAND1BS+cRNDAT04)*20H*(3)	; 抽籤乱数+抽籤回数
DB		LOW.wDDM_STS	; 抽籤パラメータ(テーブルオフセット)
DB		(80H)+dBDMC_L-\$	
DB		(80H)+dBDMC_H-\$	
dBDMC_L	EQU	\$	
DB		00000101B	
DB		124	
DB		3	
dBDMC_H	EQU	\$	
DB		00000101B	
DB		224	
DB		cHITABS	

10

【図 275】



【図 276】

乱数指定用オフセット番号					
行番	値(HEX)	ラベル	疑似命令	値(DEC)	コメント
2820	000B	cRNDAT01	EQU	11	; 0F016H
2821	000B	cRNDAT02	EQU	11	; 0F016H
2822	000B	cRNDAT03	EQU	11	; 0F016H
2823	0000	cRNDAT04	EQU	0	; 0F00BH
2824	0000	cRNDAT05	EQU	0	; 0F00BH
2825	0006	cRNDAT06	EQU	6	; 0F011H
2826	0004	cRNDAT07	EQU	4	; 0F00FH
2827	0004	cRNDAT08	EQU	4	; 0F00FH
2828	0004	cRNDAT09	EQU	4	; 0F00FH
2829	0001	cRNDAT10	EQU	1	; 0F00CH
2830	0005	cRNDAT11	EQU	5	; 0F010H
2831	0005	cRNDAT12	EQU	5	; 0F010H
2832	0005	cRNDAT13	EQU	5	; 0F010H
2833	0005	cRNDAT14	EQU	5	; 0F010H
2834	0005	cRNDAT15	EQU	5	; 0F010H
2835	0003	cRNDAT16	EQU	3	; 0F00EH
2836	0004	cRNDAT17	EQU	4	; 0F00FH
2837	0004	cRNDAT18	EQU	4	; 0F00FH
2838	0007	cRNDAT19	EQU	7	; 0F012H
2839	000A	cRNDAT20	EQU	10	; 0F015H
2840	0000	cRNDAT21	EQU	0	; 0F00BH
2841	0001	cRNDAT22	EQU	1	; 0F00CH
2842	0001	cRNDAT23	EQU	1	; 0F00CH
2843	000B	cRNDAT24	EQU	11	; 0F016H
2844	0006	cRNDAT25	EQU	6	; 0F011H
2845	000B	cRNDAT26	EQU	11	; 0F016H
2846	0001	cRNDAT27	EQU	1	; 0F00CH
2847	0001	cRNDAT28	EQU	1	; 0F00CH
2848	0000	cRNDAT29	EQU	0	; 0F00BH
2849	000B	cRNDAT30	EQU	11	; 0F016H
2850	000B	cRNDAT31	EQU	11	; 0F016H
2851	0006	cRNDAT32	EQU	6	; 0F011H
2852	0002	cRND_NAV	EQU	2	; 0F00DH
2853	0002	cRND_ZNG	EQU	2	; 0F00DH

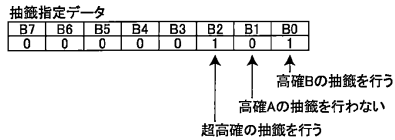
20

30

40

50

【図 277】



【図 278】

BB中ART抽籤テーブル(通常BB又はART中BB中用)			
dBBSVOTB	EQU	\$	
DB		LOW.(wRAND1BS+cRNDAT15)+20H*(0)	
DB		LOW.wDDMFLG_C	
;			
DB	cPRB_0		:0:ハスレ
DB	cPRB_4		:1:EP役
DB	cPRB_HIT		:2:プレミア系

10

20

【図 279】

当否判定用係数テーブル			
;			
B0:当否要求、B1:当籤(1)・非当籤(0)			
cPRB_HIT	EQU	1*2+(1)	; B0:当否要求、B1:当籤
cPRB_0	EQU	0*2+(1)	; B0:当否要求、B1:非当籤
;			
B0:抽籤指示、B4:抽籤回数			
ddDMPRB	EQU	\$	
DB		1*10H+0001B	
;			
cPRB_1	EQU	(\$-ddDMPRB)*2+(0)	; 2
DB		1	
cPRB_2	EQU	(\$-ddDMPRB)*2+(0)	; 4
DB		2	
cPRB_3	EQU	(\$-ddDMPRB)*2+(0)	; 6
DB		3	
cPRB_4	EQU	(\$-ddDMPRB)*2+(0)	; 8
DB		4	
cPRB_5	EQU	(\$-ddDMPRB)*2+(0)	; 10
DB		5	
cPRB_6	EQU	(\$-ddDMPRB)*2+(0)	; 12
DB		6	
cPRB_8	EQU	(\$-ddDMPRB)*2+(0)	; 14
DB		8	
cPRB_10	EQU	(\$-ddDMPRB)*2+(0)	; 16
DB		10	
cPRB_16	EQU	(\$-ddDMPRB)*2+(0)	; 18
DB		16	
cPRB_32	EQU	(\$-ddDMPRB)*2+(0)	; 20
DB		32	
cPRB_40	EQU	(\$-ddDMPRB)*2+(0)	; 22
DB		40	
cPRB_42	EQU	(\$-ddDMPRB)*2+(0)	; 24
DB		42	
cPRB_64	EQU	(\$-ddDMPRB)*2+(0)	; 26
DB		64	
cPRB_80	EQU	(\$-ddDMPRB)*2+(0)	; 28
DB		80	
cPRB_110	EQU	(\$-ddDMPRB)*2+(0)	; 30
DB		110	
cPRB_128	EQU	(\$-ddDMPRB)*2+(0)	; 32
DB		128	
cPRB_160	EQU	(\$-ddDMPRB)*2+(0)	; 34
DB		160	
cPRB_192	EQU	(\$-ddDMPRB)*2+(0)	; 36
DB		192	

【図 280】

BB当籤時状態別CZ抽籤テーブル(A)～(C)			
;			
dBHSVOTB	EQU	\$	
DB		LOW.(wRAND1BS+cRNDAT09)+20H*(0)	
DB		LOW.wDDMSTS	
DB		LOW.wDDMFLG_A	
;			
DB	(80H)+dBBH_LOW-\$		: [0]通常、低確
DB	(80H)+dBBH_HIG-\$		: [1]通常、高確
DB	(80H)+dBBH_SPH-\$		: [2]通常、超高
;			
dBBH_LOW	EQU	\$	
DB	cPRB_0		: [0]ハスレ&プレミア系
DB	cPRB_0		: [1]弱レア&弱スィカ
DB	cPRB_0		: [2]強スィカ
;			
dBBH_HIG	EQU	\$	
DB	cPRB_0		: [0]ハスレ&プレミア系 ; 低確 [3]強レア&EP役
DB	cPRB_2		: [1]弱レア&弱スィカ
DB	cPRB_64		: [2]強スィカ
DB	cPRB_128		: [3]強レア&EP役
;			
dBBH_SPH	EQU	\$	
DB	cPRB_HIT		: [0]ハスレ&プレミア系
DB	cPRB_HIT		: [1]弱レア&弱スィカ
DB	cPRB_HIT		: [2]強スィカ
DB	cPRB_HIT		: [3]強レア&EP役
;			

30

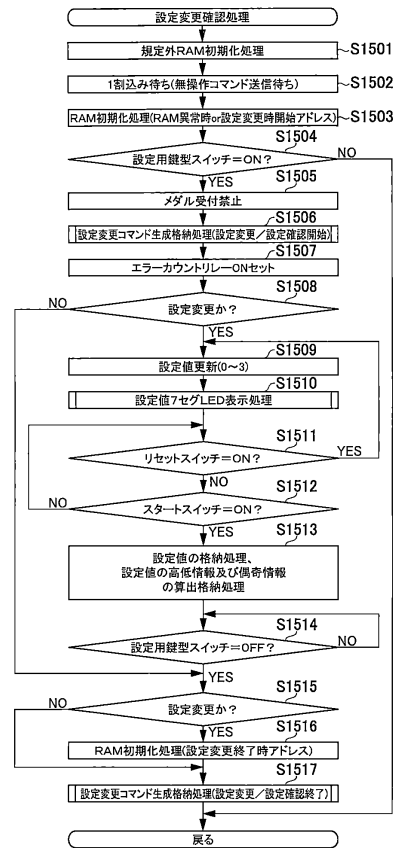
40

50

【図 2 8 1】

CZ中ART抽籤テーブル(A)~(H)				
dVCSVOTB	EQU	\$		
DB	LOW(wRAND1BS+cRNDAT12)+20H*(0)			
DB	LOW(wDDMFLQ.D		フラグ/ループD[0]~[3]	
DB	LOW(wDDMPARX		遊技状況:0~8	
;				
DB	(80H)+dVCVO_0-\$		[0] ハズレ	
DB	(80H)+dVCVO_1-\$		[1] 弱レア	
DB	(80H)+dVCVO_2-\$		[2] その他役系	
DB	(80H)+dVCVO_3-\$		[3] プレミア系	
;				
dVCVO_0	EQU	\$		
DB	cPRB_0		0:NVC 終了次遊技	
DB	cPRB_32		1:NVC 遊技数 0	
DB	cPRB_2		2:NVC 遊技数 1 以上	
DB	cPRB_0		3:AVC1 終了次遊技	
DB	cPRB_1		4:AVC1 遊技数 0	
DB	cPRB_1		5:AVC1 遊技数 1 以上	
DB	cPRB_1		6:AVC2 遊技数 0	
DB	cPRB_1		7:AVC2 遊技数 1 以上	
DB	cPRB_16		8:VCR	
;				
dVCVO_2	EQU	\$		
dVCVO_3	EQU	\$		
DB	cPRB_HIT		0:NVC 終了次遊技	
DB	cPRB_HIT		1:NVC 遊技数 0	
DB	cPRB_HIT		2:NVC 遊技数 1 以上	
DB	cPRB_HIT		3:AVC1 終了次遊技	
DB	cPRB_HIT		4:AVC1 遊技数 0	
DB	cPRB_HIT		5:AVC1 遊技数 1 以上	
;				
dVCVO_1	EQU	\$		
DB	cPRB_HIT		[1] 弱レア	
DB	cPRB_HIT		[2] その他役系	
DB	cPRB_HIT		[3] プレミア系	
DB	cPRB_HIT		0:NVC 終了次遊技	
DB	cPRB_HIT		1:NVC 遊技数 0	
DB	cPRB_HIT		2:NVC 遊技数 1 以上	
DB	cPRB_0		3:AVC1 終了次遊技	
DB	cPRB_32		4:AVC1 遊技数 0	
DB	cPRB_32		5:AVC1 遊技数 1 以上	
DB	cPRB_32		6:AVC2 遊技数 0	
DB	cPRB_32		7:AVC2 遊技数 1 以上	
DB	cPRB_128		8:VCR	

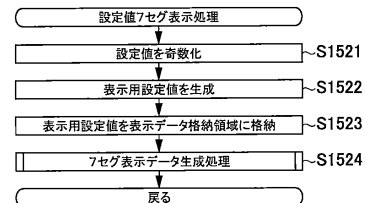
【図 2 8 2】



【図 2 8 3】

LDQ	(wWAVENUM), A		設定値 格納
;			
DIV	E, A, 2		高低/偶奇情報 生成
/;			
/;			
/;			
/;			
LDQ	(wLOW_HIH), A		設定値 高低情報 格納
LDQ	(wEVN_ODD), E		設定値 偶奇情報 格納

【図 2 8 4】



10

20

30

40

50

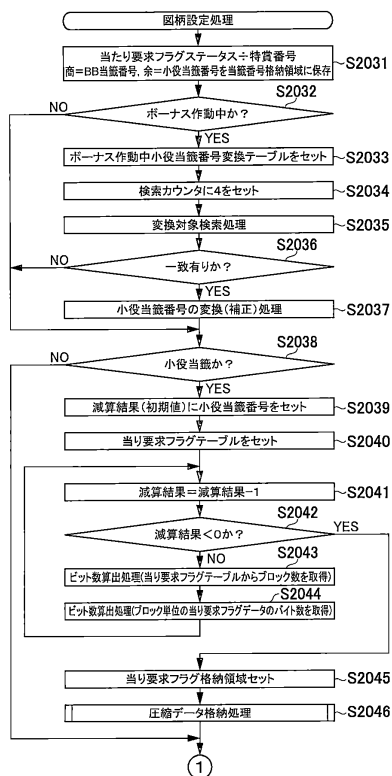
【図 285】

LD	E, A	: 設定値を退避(奇数化前)
SET	0, A	: 奇数化
ADD	A, E	: 表示用設定値生成(1,2,5,6)

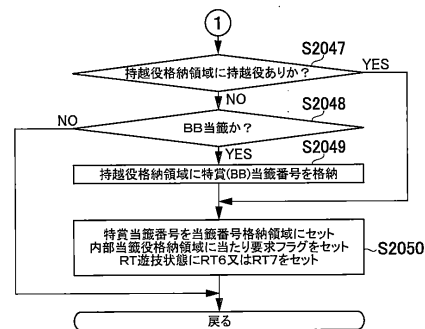
【図 286】



【図 287】



【図 288】



10

20

30

40

50

【図 289】

;== ボーナス作動中小役当籤番号変換テーブル			
dBBFRTSLTB	EQU	\$	
DB	46		; 強ベル
DB	45		; 共通ベル
DB	3		; 弱チェリブ
DB	0		; ハズレ

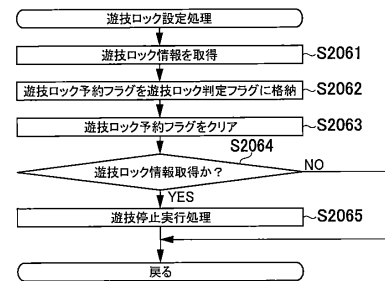
【図 290】

変換後の小役当籤番号			
検索値	補正値	変換値	
3	52	55	BB中強ベル
2	52	54	BB中共通ベル
1	52	53	BB中弱チェリブ
0	52	52	BB中ハズレ

【図 291】

LD	HL, dBBFRTSLTB	; ボーナス作動中小役当籤番号変換テーブルをセット
LD	BC, 4	; 検索カウンタの初期値をセット
CPIR		; 変換対象検索処理

【図 292】



10

20

【図 293】

LDQ	HL, wLOCK_RSV-1	; 遊技ロック予約フラグ
INLD	AE, (HL)	; A : wLOCK_FLG(遊技ロック判定フラグ)
/.....		
		; E : wLOCK_RSV(遊技ロック予約フラグ)
LD	(HL), E	; 遊技ロック判定フラグに遊技ロック予約フラグを格納
CLRQ	(wLOCK_RSV)	; 遊技ロック予約フラグをクリア
RT	Z, A	; 遊技ロック情報無しならリターン

【図 294】

F216	woXOUT0	DS	3	; 規定外出力ポート格納領域
F219	;			; +0 0F4H 条件装置1~8(条件装置信号)
F219	;			; +1 0F5H 特賞信号
F219	;			; +2 0F6H 回胴制動信号

30

40

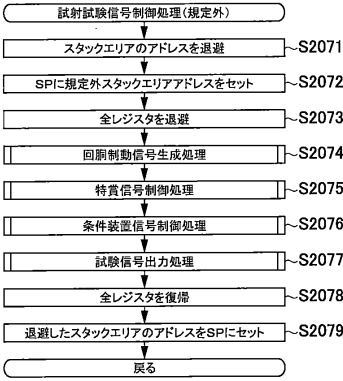
50

【図 2 9 5】

規定外出力ポート格納領域内のBIT割り付け

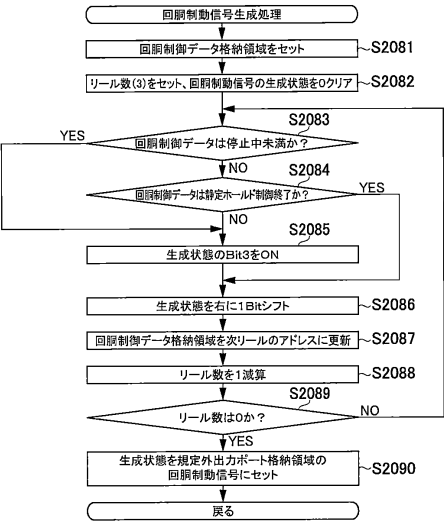
内容	B7	B6	B5	B4	B3	B2	B1	B0
条件装置1~8(条件装置信号)	8	7	6	5	4	3	2	1
特賞信号	SB	CB	MB	RB	BB	-	-	-
回胴制動信号	-	-	-	-	-	R3	R2	R1

【図 2 9 6】

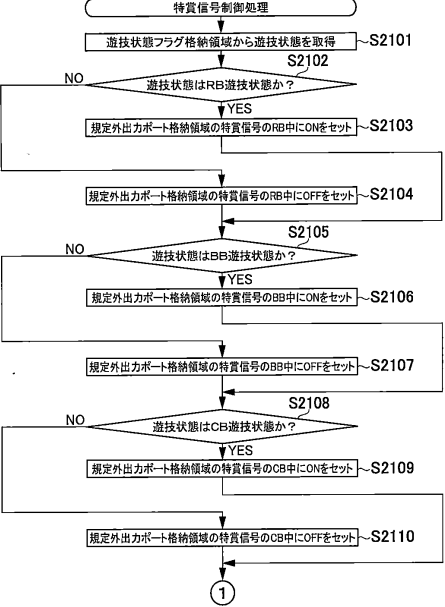


10

【図 2 9 7】



【図 2 9 8】



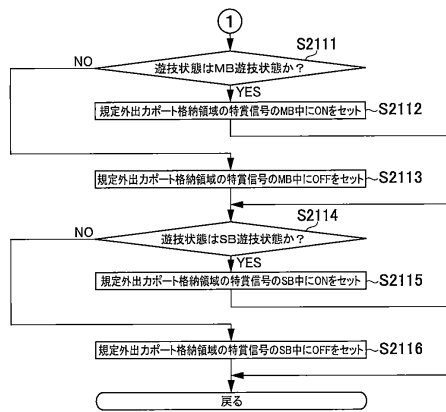
20

30

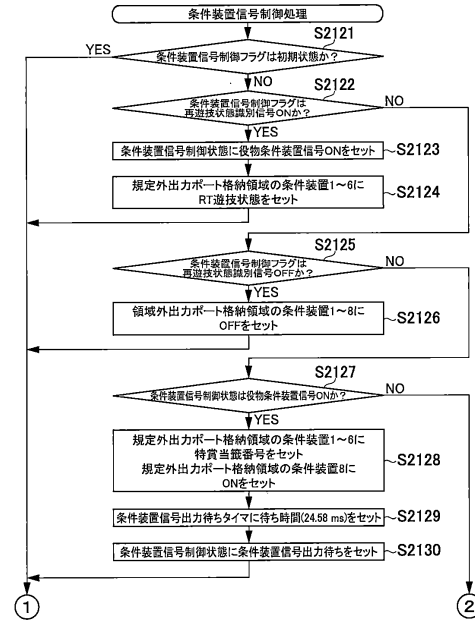
40

50

【図 299】



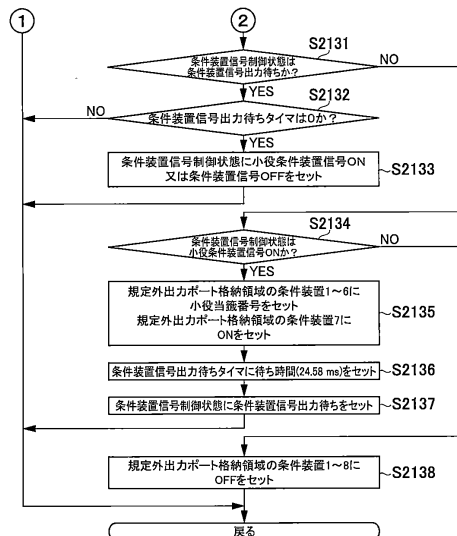
【図 300】



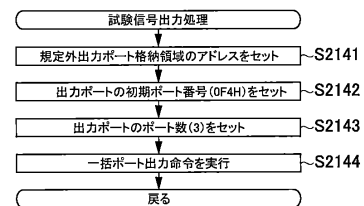
10

20

【図 301】



【図 302】



30

40

50

【図 3 0 3】

LD	HL, woXOUT0	: 規定外出力ポート格納領域のアドレスセット
LD	BC, 100H*3 + 0F4H	: 初期ポート番号: 0F4H, ポート数: 3
OTICR		: [F4H] ~ [F6H] に一括出力

【図 3 0 4】

No	新	二一モニック	小 数	オペレーション	フラグ							説明
					S	Z	TZ	H	P/V	OV		
A1	EX	AF, AF'	1	AF $\leftrightarrow$ AF'	.	.	.	.	.	.	.	ヘアリスAFとヘアリスAF'の値を交換
A2	EXX	EX BC, DE	1	BC $\leftrightarrow$ BC'DE $\leftrightarrow$ DE'HL $\leftrightarrow$ HL'	.	.	.	.	.	.	.	ヘアリスABCD, HLとヘアリスABCD', DE', HL'の値を交換
A3	EX	BC, DE	2	BC $\leftrightarrow$ DE	.	.	.	.	.	.	.	ヘアリスABCDとヘアリスABD', DE'の値を交換
A4	EX	BC, HL	2	BC $\leftrightarrow$ HL	.	.	.	.	.	.	.	ヘアリスABCDとヘアリスABD', HLの値を交換
A5	EX	DE, HL	1	DE $\leftrightarrow$ HL	.	.	.	.	.	.	.	ヘアリスABDとヘアリスABD', DE'の値を交換
A6	EX	ss, qq	2	ss $\leftrightarrow$ qq	.	.	.	.	.	.	.	ヘアリスssとヘアリスqqの値を交換
A7	EX	(SP), HL	1	HL $\leftrightarrow$ (SP)HL $\leftrightarrow$ (SP+1)	.	.	.	.	.	.	.	ヘアリスHLとヘアリスSPの指定するHLの値を交換
A8	EX	(SP), IX	2	IXL $\leftrightarrow$ (SP)IXHL $\leftrightarrow$ (SP+1)	.	.	.	.	.	.	.	ヘアリスIXとヘアリスSPの指定するIXの値を交換
A9	EX	(SP), IY	2	IYL $\leftrightarrow$ (SP)IYHL $\leftrightarrow$ (SP+1)	.	.	.	.	.	.	.	ヘアリスIYとヘアリスSPの指定するIYの値を交換
A10	EX	r, (ss)	2	r $\leftrightarrow$ (ss)	.	.	.	.	.	.	.	ヘアリスrの値とヘアリスssで指定されるrのHLの値を交換
A11	OX	A, (mm)	4	A $\leftrightarrow$ (mm)	.	.	.	.	.	.	.	16bit外数値で指定されるrのHLとヘアリスAの値を交換
A12	OX	B, (mm)	4	B $\leftrightarrow$ (mm)	.	.	.	.	.	.	.	16bit外数値で指定されるrのHLとヘアリスBの値を交換
A13	OX	C, (mm)	4	C $\leftrightarrow$ (mm)	.	.	.	.	.	.	.	16bit外数値で指定されるrのHLとヘアリスCの値を交換
D14	OX	D, (mm)	4	D $\leftrightarrow$ (mm)	.	.	.	.	.	.	.	16bit外数値で指定されるrのHLとヘアリスDの値を交換
A15	OX	E, (mm)	4	E $\leftrightarrow$ (mm)	.	.	.	.	.	.	.	16bit外数値で指定されるrのHLとヘアリスEの値を交換
A16	OX	H, (mm)	4	H $\leftrightarrow$ (mm)	.	.	.	.	.	.	.	16bit外数値で指定されるrのHLとヘアリスHの値を交換
A17	OX	L, (mm)	4	L $\leftrightarrow$ (mm)	.	.	.	.	.	.	.	16bit外数値で指定されるrのHLとヘアリスLの値を交換
A18	OX	r, (qq)	2	r $\leftrightarrow$ (qq)	.	.	.	.	.	.	.	ヘアリスrの値とヘアリスqqの指定するrのHLの値を交換
A19	OXQ	r, (k)	3	r $\leftrightarrow$ (k)	.	.	.	.	.	.	.	ヘアリスrの値とヘアリスk, 8bit外数値の指定するrのHLの値を交換

へんせきしんせき

【図 3 0 5】

[illegible]

＜EX ss, qq(命令No.A6):エクスチェンジ命令＞

【図 3 0 6】

No	ニモニック	ハ件数	オペレーション	フラグ								説明
				S	Z	TZ	H	P	N	CV		
A10a	EX A, (BC)	2	A⇔(BC)	.	.	.	.	.	.	.	へリツ3BQCの指定4つの値とレジスタAの値を交換	
A10b	EX A, (BC)	2	A⇔(BC)	.	.	.	.	.	.	.	へリツ3BQCの指定4つの値とレジスタAの値を交換	
A10c	EX D, (BC)	2	D⇔(BC)	.	.	.	.	.	.	.	へリツ3BQCの指定4つの値とレジスタDの値を交換	
A10d	EX E, (BC)	2	E⇔(BC)	.	.	.	.	.	.	.	へリツ3BQCの指定4つの値とレジスタEの値を交換	
A10e	EX H, (BC)	2	H⇔(BC)	.	.	.	.	.	.	.	へリツ3BQCの指定4つの値とレジスタHの値を交換	
A10f	EX L, (BC)	2	L⇔(BC)	.	.	.	.	.	.	.	へリツ3BQCの指定4つの値とレジスタLの値を交換	
A10g	EX A, (DE)	2	A⇔(DE)	.	.	.	.	.	.	.	へリツ3BDEの指定4つの値とレジスタAの値を交換	
A10h	EX B, (DE)	2	B⇔(DE)	.	.	.	.	.	.	.	へリツ3BDEの指定4つの値とレジスタBの値を交換	
A10i	EX C, (DE)	2	C⇔(DE)	.	.	.	.	.	.	.	へリツ3BDEの指定4つの値とレジスタCの値を交換	
A10j	EX H, (DE)	2	H⇔(DE)	.	.	.	.	.	.	.	へリツ3BDEの指定4つの値とレジスタHの値を交換	
A10k	EX L, (DE)	2	L⇔(DE)	.	.	.	.	.	.	.	へリツ3BDEの指定4つの値とレジスタLの値を交換	
A10l	EX A, (HL)	2	A⇔(HL)	.	.	.	.	.	.	.	へリツ3HLの指定4つの値とレジスタAの値を交換	
A10m	EX B, (HL)	2	B⇔(HL)	.	.	.	.	.	.	.	へリツ3HLの指定4つの値とレジスタBの値を交換	
A10n	EX C, (HL)	2	C⇔(HL)	.	.	.	.	.	.	.	へリツ3HLの指定4つの値とレジスタCの値を交換	
A10o	EX D, (HL)	2	D⇔(HL)	.	.	.	.	.	.	.	へリツ3HLの指定4つの値とレジスタDの値を交換	
A10p	EX E, (HL)	2	E⇔(HL)	.	.	.	.	.	.	.	へリツ3HLの指定4つの値とレジスタEの値を交換	

＜EX r, (ss) (命令No.A10): エクスチェンジ命令＞

【図 3 0 7】

<EX r (np) (命令NoA18): エクスチェンジ命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
A18a	EX A, (X)	2	A⇌(X)	•	•	•	•	•	•	•	レジスタAの値とレジスタXの指定メモリの内容を交換
A18b	EX B, (X)	2	B⇌(X)	•	•	•	•	•	•	•	レジスタBの値とレジスタXの指定メモリの内容を交換
A18c	EX C, (X)	2	C⇌(X)	•	•	•	•	•	•	•	レジスタCの値とレジスタXの指定メモリの内容を交換
A18d	EX D, (X)	2	D⇌(X)	•	•	•	•	•	•	•	レジスタDの値とレジスタXの指定メモリの内容を交換
A18e	EX E, (X)	2	E⇌(X)	•	•	•	•	•	•	•	レジスタEの値とレジスタXの指定メモリの内容を交換
A18f	EX H, (X)	2	H⇌(X)	•	•	•	•	•	•	•	レジスタHの値とレジスタXの指定メモリの内容を交換
A18g	EX L, (X)	2	L⇌(X)	•	•	•	•	•	•	•	レジスタLの値とレジスタXの指定メモリの内容を交換
A18h	EX A, (Y)	2	A⇌(Y)	•	•	•	•	•	•	•	レジスタAの値とレジスタYの指定メモリの内容を交換
A18i	EX B, (Y)	2	B⇌(Y)	•	•	•	•	•	•	•	レジスタBの値とレジスタYの指定メモリの内容を交換
A18j	EX C, (Y)	2	C⇌(Y)	•	•	•	•	•	•	•	レジスタCの値とレジスタYの指定メモリの内容を交換
A18k	EX D, (Y)	2	D⇌(Y)	•	•	•	•	•	•	•	レジスタDの値とレジスタYの指定メモリの内容を交換
A18l	EX E, (Y)	2	E⇌(Y)	•	•	•	•	•	•	•	レジスタEの値とレジスタYの指定メモリの内容を交換
A18m	EX H, (Y)	2	H⇌(Y)	•	•	•	•	•	•	•	レジスタHの値とレジスタYの指定メモリの内容を交換
A18n	EX L, (Y)	2	L⇌(Y)	•	•	•	•	•	•	•	レジスタLの値とレジスタYの指定メモリの内容を交換

【図 3 0 8】

<EXQ r, (k) (命令NoA19): エクスチェンジ命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
A19a	EXQ A, (k)	3	A⇌(Qk)	•	•	•	•	•	•	•	レジスタAの値とレジスタQ, 8C(外部登録)の指定メモリの内容を交換
A19b	EXQ B, (k)	3	B⇌(Qk)	•	•	•	•	•	•	•	レジスタBの値とレジスタQ, 8C(外部登録)の指定メモリの内容を交換
A19c	EXQ C, (k)	3	C⇌(Qk)	•	•	•	•	•	•	•	レジスタCの値とレジスタQ, 8C(外部登録)の指定メモリの内容を交換
A19d	EXQ D, (k)	3	D⇌(Qk)	•	•	•	•	•	•	•	レジスタDの値とレジスタQ, 8C(外部登録)の指定メモリの内容を交換
A19e	EXQ E, (k)	3	E⇌(Qk)	•	•	•	•	•	•	•	レジスタEの値とレジスタQ, 8C(外部登録)の指定メモリの内容を交換
A19f	EXQ H, (k)	3	H⇌(Qk)	•	•	•	•	•	•	•	レジスタHの値とレジスタQ, 8C(外部登録)の指定メモリの内容を交換
A19g	EXQ L, (k)	3	L⇌(Qk)	•	•	•	•	•	•	•	レジスタLの値とレジスタQ, 8C(外部登録)の指定メモリの内容を交換

【図 3 0 9】

<8ビットロード命令(その1)>

No	新	ニーモニック	バイト数	オペレーション	フラグ							説明
					S	Z	TZ	H	P/V	N	OY	
B1		LD r, s	1	r←s	•	•	•	•	•	•	•	レジスタsの内容をレジスタrにセット
B2		LD r, n	2	r←n	•	•	•	•	•	•	•	8C(リテラル)の内容をレジスタrにセット
B3		LD A, (mm)	3	A←(mm)	•	•	•	•	•	•	•	16C(リテラル)の内容をレジスタAにセット
B4		LD A, (BC)	1	A←(BC)	•	•	•	•	•	•	•	レジスタBCで指定されるメモリの内容をレジスタAにセット
B5		LD r, (qq)	2	r←(qq)	•	•	•	•	•	•	•	レジスタqqで指定されるメモリの内容をレジスタrにセット
B6		LD A, (DE)	1	A←(DE)	•	•	•	•	•	•	•	レジスタDEで指定されるメモリの内容をレジスタAにセット
B7		LD r, (DE)	2	r←(DE)	•	•	•	•	•	•	•	レジスタDEで指定されるメモリの内容をレジスタrにセット
B8		LD r, (HL)	1	r←(HL)	•	•	•	•	•	•	•	レジスタHLで指定されるメモリの内容をレジスタrにセット
B9		LDQ A, (k)	2	A←(Qk)	•	•	•	•	•	•	•	レジスタQ, 8C(リテラル)の内容をレジスタAにセット
B10		LDQ r, (k)	3	r←(Qk)	•	•	•	•	•	•	•	レジスタQ, 8C(リテラル)の内容をレジスタrにセット
B11		LD (HL), r	1	(HL)←r	•	•	•	•	•	•	•	レジスタrの内容をレジスタHLにセット
B12		LD (qq), r	2	(qq)←r	•	•	•	•	•	•	•	レジスタrの内容をレジスタqqにセット
B13		LDQ (k), A	2	(Qk)←A	•	•	•	•	•	•	•	レジスタAの内容をレジスタQ, 8C(リテラル)にセット
B14		LDQ (k), r	3	(Qk)←r	•	•	•	•	•	•	•	レジスタrの内容をレジスタQ, 8C(リテラル)にセット
B15		LDQ (k), n	3	(Qk)←n	•	•	•	•	•	•	•	8C(リテラル)の内容をレジスタQ, 8C(リテラル)にセット

【図 3 1 0】

<8ビットロード命令(その2)>

No	新	ニーモニック	バイト数	オペレーション	フラグ							説明
					S	Z	TZ	H	P/V	N	OY	
B16	○	LDZ (m)	4	if TZ=1 (mm)←0 else (mm)←1	•	•	•	•	•	•	•	TZフラグがセットの場合、整数mmの指定メモリに0をセット、TZフラグがリセットの場合、整数mmの指定メモリに1をセット
B17	○	LDZ (DE)	2	if TZ=1 (DE)←0 else (DE)←1	•	•	•	•	•	•	•	TZフラグがセットの場合、レジスタDEの指定メモリに0をセット、TZフラグがリセットの場合、レジスタDEの指定メモリに1をセット
B18	○	LDZ (HL)	2	if TZ=1 (HL)←0 else (HL)←1	•	•	•	•	•	•	•	TZフラグがセットの場合、レジスタHLの指定メモリに0をセット、TZフラグがリセットの場合、レジスタHLの指定メモリに1をセット
B19	○	LDZ (X)	2	if TZ=1 (X)←0 else (X)←1	•	•	•	•	•	•	•	TZフラグがセットの場合、レジスタXの指定メモリに0をセット、TZフラグがリセットの場合、レジスタXの指定メモリに1をセット
B20	○	LDZ (Y)	2	if TZ=1 (Y)←0 else (Y)←1	•	•	•	•	•	•	•	TZフラグがセットの場合、レジスタYの指定メモリに0をセット、TZフラグがリセットの場合、レジスタYの指定メモリに1をセット
B21	○	LDQZ (k)	3	if TZ=1 (k)←0 else (k)←1	•	•	•	•	•	•	•	TZフラグがセットの場合、レジスタQ, 8C(外部登録)の指定メモリに0をセット、TZフラグがリセットの場合、レジスタQ, 8C(外部登録)の指定メモリに1をセット
B22	○	LDC (mm)	4	if OY=1 (mm)←1 else (mm)←0	•	•	•	•	•	•	•	OYフラグがセットの場合、整数mmで指定されるメモリに1をセット、OYフラグがリセットの場合、整数mmで指定されるメモリに0をセット
B23	○	LDC (DE)	2	if OY=1 (DE)←1 else (DE)←0	•	•	•	•	•	•	•	OYフラグがセットの場合、レジスタDEの指定メモリに1をセット、OYフラグがリセットの場合、レジスタDEの指定メモリに0をセット
B24	○	LDC (HL)	2	if OY=1 (HL)←1 else (HL)←0	•	•	•	•	•	•	•	OYフラグがセットの場合、レジスタHLの指定メモリに1をセット、OYフラグがリセットの場合、レジスタHLの指定メモリに0をセット

【図 3 1 1】

<8ビットロード命令(その3)>

No	新 ニーモニック	オペレーション	バイト 数	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
B25	O LDC (X)	if OY=1 (OX)-1 else (OX)-0	2	・	・	Φ	・	・	・	・	OF7がセフトの場合、インデックスXの指定メモリにセフトし、OF7がセフトの場合、インデックスXの指定メモリにセフト
B26	O LDC (Y)		2	・	・	Φ	・	・	・	・	OF7がセフトの場合、インデックスYの指定メモリにセフトし、OF7がセフトの場合、インデックスYの指定メモリにセフト
B27	O LDQC (k)	if OY=1 (k)-1 else (k)-0	3	・	・	Φ	・	・	・	・	OF7がセフトの場合、レジスタk、整数値の指定メモリにセフトし、OF7がセフトの場合、レジスタk、整数値の指定メモリにセフト
B28	O LDF NZ, A, n		3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタAに8ビットデータを書きロード
B29	O LDF Z, A, n	if Z=1 A-n if Z=0 A-n	3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタAに8ビットデータを書きロード
B30	O LDF NC, A, n		3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタAに8ビットデータを書きロード
B31	O LDF C, A, n	if OY=1 A-n if Z=0 B-n	3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタAに8ビットデータを書きロード
B32	O LDF NZ, B, n		3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタBに8ビットデータを書きロード
B33	O LDF Z, B, n	if Z=1 B-n if Z=0 B-n	3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタBに8ビットデータを書きロード
B34	O LDF NC, B, n		3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタBに8ビットデータを書きロード
B35	O LDF C, B, n	if OY=1 B-n if Z=0 C-n	3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタBに8ビットデータを書きロード
B36	O LDF NZ, C, n		3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタCに8ビットデータを書きロード
B37	O LDF Z, C, n	if Z=1 C-n if Z=0 C-n	3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタCに8ビットデータを書きロード
B38	O LDF NC, C, n		3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタCに8ビットデータを書きロード
B39	O LDF C, C, n	if OY=1 C-n if Z=0 D-n	3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタCに8ビットデータを書きロード
B40	O LDF NZ, D, n		3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタDに8ビットデータを書きロード

【図 3 1 2】

<8ビットロード命令(その4)>

No	新 ニーモニック	オペレーション	バイト 数	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
B41	O LDF Z, D, n	if Z=1 D-n if OY=1 D-n	3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタDに8ビットデータを書きロード
B42	O LDF NC, D, n		3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタDに8ビットデータを書きロード
B43	O LDF C, D, n	if OY=1 D-n if Z=0 E-n	3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタDに8ビットデータを書きロード
B44	O LDF NZ, E, n		3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタEに8ビットデータを書きロード
B45	O LDF N, E, n	if Z=1 E-n if OY=1 E-n	3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタEに8ビットデータを書きロード
B46	O LDF NC, E, n		3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタEに8ビットデータを書きロード
B47	O LDF C, E, n	if Z=0 E-n if Z=1 H-n	3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタEに8ビットデータを書きロード
B48	O LDF NZ, H, n		3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタHに8ビットデータを書きロード
B49	O LDF Z, H, n	if Z=1 H-n if OY=1 H-n	3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタHに8ビットデータを書きロード
B50	O LDF NC, H, n		3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタHに8ビットデータを書きロード
B51	O LDF C, H, n	if Z=0 L-n if Z=1 L-n	3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタLに8ビットデータを書きロード
B52	O LDF NZ, L, n		3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタLに8ビットデータを書きロード
B53	O LDF Z, L, n	if Z=1 L-n if OY=1 L-n	3	・	・	Φ	・	・	・	・	Z7がセフトならレジスタLに8ビットデータを書きロード
B54	O LDF NC, L, n		3	・	・	Φ	・	・	・	・	OF7がセフトならレジスタLに8ビットデータを書きロード
B55	O LDF C, L, n	if Z=0 (mn)-d if Z=1 (mn)-d	5	・	・	Φ	・	・	・	・	OF7がセフトならレジスタmnの指定メモリに8ビットデータを書きロード
B56	O LDF Z, (mn), d		5	・	・	Φ	・	・	・	・	Z7がセフトならレジスタmnの指定メモリに8ビットデータを書きロード
B57	O LDF C, (mn), d	if Z=0 (mn)-d if OY=0 (mn)-d	5	・	・	Φ	・	・	・	・	OF7がセフトなら整数mnの指定メモリに8ビットデータを書きロード
B58	O LDF NC, (mn), d		5	・	・	Φ	・	・	・	・	OF7がセフトなら整数mnの指定メモリに8ビットデータを書きロード
B59	O LDF C, (mn), d	if Z=0 (mn)-d if OY=1 (mn)-d	5	・	・	Φ	・	・	・	・	OF7がセフトなら整数mnの指定メモリに8ビットデータを書きロード
B60	O LDF Z, (mn), d		5	・	・	Φ	・	・	・	・	Z7がセフトなら整数mnの指定メモリに8ビットデータを書きロード

【図 3 1 3】

<LD r, s (命令No.B1, その1): 8ビットロード命令>

No	ニーモニック	オペレーション	バイト 数	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
B1a	LD A, B	A←B	1	・	・	Φ	・	・	・	・	レジスタBの内容をレジスタAにセフト
B1b	LD A, C		1	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタAにセフト
B1c	LD A, D	A←D	1	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタAにセフト
B1d	LD A, E		1	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタAにセフト
B1e	LD A, H	A←H	1	・	・	Φ	・	・	・	・	レジスタHの内容をレジスタAにセフト
B1f	LD A, L		1	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタAにセフト
B1g	LD B, A	B←A	1	・	・	Φ	・	・	・	・	レジスタAの内容をレジスタBにセフト
B1h	LD B, C		1	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタBにセフト
B1i	LD B, D	B←D	1	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタBにセフト
B1j	LD B, E		1	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタBにセフト
B1k	LD B, H	B←H	1	・	・	Φ	・	・	・	・	レジスタHの内容をレジスタBにセフト
B1l	LD B, L		1	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタBにセフト
B1m	LD C, A	C←A	1	・	・	Φ	・	・	・	・	レジスタAの内容をレジスタCにセフト
B1n	LD C, B		1	・	・	Φ	・	・	・	・	レジスタBの内容をレジスタCにセフト
B1o	LD C, D	C←D	1	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタCにセフト
B1p	LD C, E		1	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタCにセフト
B1q	LD C, H	C←H	1	・	・	Φ	・	・	・	・	レジスタHの内容をレジスタCにセフト
B1r	LD C, L		1	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタCにセフト

【図 3 1 4】

<LD r, s (命令No.B1, その2): 8ビットロード命令>

No	ニーモニック	オペレーション	バイト 数	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
B1s	LD D, A	D←A	1	・	・	Φ	・	・	・	・	レジスタAの内容をレジスタDにセフト
B1t	LD D, B		1	・	・	Φ	・	・	・	・	レジスタBの内容をレジスタDにセフト
B1u	LD D, C	D←C	1	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタDにセフト
B1v	LD D, E		1	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタDにセフト
B1w	LD D, H	D←H	1	・	・	Φ	・	・	・	・	レジスタHの内容をレジスタDにセフト
B1x	LD D, L		1	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタDにセフト
B1y	LD E, A	E←A	1	・	・	Φ	・	・	・	・	レジスタAの内容をレジスタEにセフト
B1z	LD E, B		1	・	・	Φ	・	・	・	・	レジスタBの内容をレジスタEにセフト
B1A	LD E, C	E←C	1	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタEにセフト
B1B	LD E, D		1	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタEにセフト
B1C	LD E, H	E←H	1	・	・	Φ	・	・	・	・	レジスタHの内容をレジスタEにセフト
B1D	LD E, L		1	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタEにセフト
B1E	LD H, A	H←A	1	・	・	Φ	・	・	・	・	レジスタAの内容をレジスタHにセフト
B1F	LD H, B		1	・	・	Φ	・	・	・	・	レジスタBの内容をレジスタHにセフト
B1G	LD H, C	H←C	1	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタHにセフト
B1H	LD H, D		1	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタHにセフト
B1I	LD H, E	H←E	1	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタHにセフト
B1J	LD H, L		1	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタHにセフト

【図 3 1 5】

<LD r, s (命令No.B1, その3): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ								説明
				S	Z	TZ	H	P/V	N	OY		
B1K	LD L, A	1	L←A	・	・	Φ	・	・	・	・	・	レジスタAの内容をレジスタLにセット
B1L	LD L, B	1	L←B	・	・	Φ	・	・	・	・	・	レジスタBの内容をレジスタLにセット
B1M	LD L, C	1	L←C	・	・	Φ	・	・	・	・	・	レジスタCの内容をレジスタLにセット
B1N	LD L, D	1	L←D	・	・	Φ	・	・	・	・	・	レジスタDの内容をレジスタLにセット
B1O	LD L, E	1	L←E	・	・	Φ	・	・	・	・	・	レジスタEの内容をレジスタLにセット
B1P	LD L, H	1	L←H	・	・	Φ	・	・	・	・	・	レジスタHの内容をレジスタLにセット

【図 3 1 6】

<LD r, n (命令No.B2): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ								説明
				S	Z	TZ	H	P/V	N	OY		
B2a	LD A, n	2	A ← n	・	・	Φ	・	・	・	・	・	8ビットリテラルnをレジスタAにセット
B2b	LD B, n	2	B ← n	・	・	Φ	・	・	・	・	・	8ビットリテラルnをレジスタBにセット
B2c	LD C, n	2	C ← n	・	・	Φ	・	・	・	・	・	8ビットリテラルnをレジスタCにセット
B2d	LD D, n	2	D ← n	・	・	Φ	・	・	・	・	・	8ビットリテラルnをレジスタDにセット
B2e	LD E, n	2	E ← n	・	・	Φ	・	・	・	・	・	8ビットリテラルnをレジスタEにセット
B2f	LD H, n	2	H ← n	・	・	Φ	・	・	・	・	・	8ビットリテラルnをレジスタHにセット
B2g	LD L, n	2	L ← n	・	・	Φ	・	・	・	・	・	8ビットリテラルnをレジスタLにセット

【図 3 1 7】

<LD r, (qr) (命令No.B5): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ								説明
				S	Z	TZ	H	P/V	N	OY		
B5a	LD A, (0x)	2	A←(0x)	・	・	Φ	・	・	・	・	・	インデックス0xで指定されるメモリの内容をレジスタAにセット
B5b	LD B, (0x)	2	B←(0x)	・	・	Φ	・	・	・	・	・	インデックス0xで指定されるメモリの内容をレジスタBにセット
B5c	LD C, (0x)	2	C←(0x)	・	・	Φ	・	・	・	・	・	インデックス0xで指定されるメモリの内容をレジスタCにセット
B5d	LD D, (0x)	2	D←(0x)	・	・	Φ	・	・	・	・	・	インデックス0xで指定されるメモリの内容をレジスタDにセット
B5e	LD E, (0x)	2	E←(0x)	・	・	Φ	・	・	・	・	・	インデックス0xで指定されるメモリの内容をレジスタEにセット
B5f	LD H, (0x)	2	H←(0x)	・	・	Φ	・	・	・	・	・	インデックス0xで指定されるメモリの内容をレジスタHにセット
B5g	LD L, (0x)	2	L←(0x)	・	・	Φ	・	・	・	・	・	インデックス0xで指定されるメモリの内容をレジスタLにセット
B5h	LD A, (0y)	2	A←(0y)	・	・	Φ	・	・	・	・	・	インデックス0yで指定されるメモリの内容をレジスタAにセット
B5i	LD B, (0y)	2	B←(0y)	・	・	Φ	・	・	・	・	・	インデックス0yで指定されるメモリの内容をレジスタBにセット
B5j	LD C, (0y)	2	C←(0y)	・	・	Φ	・	・	・	・	・	インデックス0yで指定されるメモリの内容をレジスタCにセット
B5k	LD D, (0y)	2	D←(0y)	・	・	Φ	・	・	・	・	・	インデックス0yで指定されるメモリの内容をレジスタDにセット
B5l	LD E, (0y)	2	E←(0y)	・	・	Φ	・	・	・	・	・	インデックス0yで指定されるメモリの内容をレジスタEにセット
B5m	LD H, (0y)	2	H←(0y)	・	・	Φ	・	・	・	・	・	インデックス0yで指定されるメモリの内容をレジスタHにセット
B5n	LD L, (0y)	2	L←(0y)	・	・	Φ	・	・	・	・	・	インデックス0yで指定されるメモリの内容をレジスタLにセット

【図 3 1 8】

<LD r, (DE) (命令No.B7): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明	
				S	Z	TZ	H	P/V	N	CV		
B7a	LD B, (DE)	2	B←(DE)	・	・	Φ	・	・	・	・	・	アドレスDEで指定されるメモリの内容をレジスタBにセット
B7b	LD C, (DE)	2	C←(DE)	・	・	Φ	・	・	・	・	・	アドレスDEで指定されるメモリの内容をレジスタCにセット
B7c	LD D, (DE)	2	D←(DE)	・	・	Φ	・	・	・	・	・	アドレスDEで指定されるメモリの内容をレジスタDにセット
B7d	LD E, (DE)	2	E←(DE)	・	・	Φ	・	・	・	・	・	アドレスDEで指定されるメモリの内容をレジスタEにセット
B7e	LD H, (DE)	2	H←(DE)	・	・	Φ	・	・	・	・	・	アドレスDEで指定されるメモリの内容をレジスタHにセット
B7f	LD L, (DE)	2	L←(DE)	・	・	Φ	・	・	・	・	・	アドレスDEで指定されるメモリの内容をレジスタLにセット

10

20

30

40

50

【図 3 1 9】

<LD r, (HL) (命令No.B8): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
B8a	LD A, (HL)	1	A←(HL)	・	・	Φ	・	・	・	・	ペアレジスタHLで指定されるメモリの内容をレジスタAにセト
B8b	LD B, (HL)	1	B←(HL)	・	・	Φ	・	・	・	・	ペアレジスタHLで指定されるメモリの内容をレジスタBにセト
B8c	LD C, (HL)	1	C←(HL)	・	・	Φ	・	・	・	・	ペアレジスタHLで指定されるメモリの内容をレジスタCにセト
B8d	LD D, (HL)	1	D←(HL)	・	・	Φ	・	・	・	・	ペアレジスタHLで指定されるメモリの内容をレジスタDにセト
B8e	LD E, (HL)	1	E←(HL)	・	・	Φ	・	・	・	・	ペアレジスタHLで指定されるメモリの内容をレジスタEにセト
B8f	LD H, (HL)	1	H←(HL)	・	・	Φ	・	・	・	・	ペアレジスタHLで指定されるメモリの内容をレジスタHにセト
B8g	LD L, (HL)	1	L←(HL)	・	・	Φ	・	・	・	・	ペアレジスタHLで指定されるメモリの内容をレジスタLにセト

【図 3 2 0】

<LDQ r, (k) (命令No.B10): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
B10a	LDQ B, (k)	3	B←(Qk)	・	・	Φ	・	・	・	・	レジスタQk 8ビットデータkの指定メモリの内容をレジスタBにセト
B10b	LDQ G, (k)	3	G←(Qk)	・	・	Φ	・	・	・	・	レジスタQk 8ビットデータkの指定メモリの内容をレジスタGにセト
B10c	LDQ D, (k)	3	D←(Qk)	・	・	Φ	・	・	・	・	レジスタQk 8ビットデータkの指定メモリの内容をレジスタDにセト
B10d	LDQ E, (k)	3	E←(Qk)	・	・	Φ	・	・	・	・	レジスタQk 8ビットデータkの指定メモリの内容をレジスタEにセト
B10e	LDQ H, (k)	3	H←(Qk)	・	・	Φ	・	・	・	・	レジスタQk 8ビットデータkの指定メモリの内容をレジスタHにセト
B10f	LDQ L, (k)	3	L←(Qk)	・	・	Φ	・	・	・	・	レジスタQk 8ビットデータkの指定メモリの内容をレジスタLにセト

【図 3 2 1】

<LD (HL), r (命令No.B11): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
B11a	LD (HL), A	1	(HL)←A	・	・	Φ	・	・	・	・	レジスタAの内容をレジスタHLで指定されるメモリにセト
B11b	LD (HL), B	1	(HL)←B	・	・	Φ	・	・	・	・	レジスタBの内容をレジスタHLで指定されるメモリにセト
B11c	LD (HL), C	1	(HL)←C	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタHLで指定されるメモリにセト
B11d	LD (HL), D	1	(HL)←D	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタHLで指定されるメモリにセト
B11e	LD (HL), E	1	(HL)←E	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタHLで指定されるメモリにセト

【図 3 2 2】

<LD (qq), r (命令No.B12): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
B12a	LD (QX), A	2	(QX)←A	・	・	Φ	・	・	・	・	レジスタAの内容をレジスタQXに指定されるメモリにセト
B12b	LD (QX), B	2	(QX)←B	・	・	Φ	・	・	・	・	レジスタBの内容をレジスタQXに指定されるメモリにセト
B12c	LD (QX), C	2	(QX)←C	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタQXに指定されるメモリにセト
B12d	LD (QX), D	2	(QX)←D	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタQXに指定されるメモリにセト
B12e	LD (QX), E	2	(QX)←E	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタQXに指定されるメモリにセト
B12f	LD (QX), H	2	(QX)←H	・	・	Φ	・	・	・	・	レジスタHの内容をレジスタQXに指定されるメモリにセト
B12g	LD (QX), L	2	(QX)←L	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタQXに指定されるメモリにセト
B12h	LD (QY), A	2	(QY)←A	・	・	Φ	・	・	・	・	レジスタAの内容をレジスタQYに指定されるメモリにセト
B12i	LD (QY), B	2	(QY)←B	・	・	Φ	・	・	・	・	レジスタBの内容をレジスタQYに指定されるメモリにセト
B12j	LD (QY), C	2	(QY)←C	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタQYに指定されるメモリにセト
B12k	LD (QY), D	2	(QY)←D	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタQYに指定されるメモリにセト
B12l	LD (QY), E	2	(QY)←E	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタQYに指定されるメモリにセト
B12m	LD (QY), H	2	(QY)←H	・	・	Φ	・	・	・	・	レジスタHの内容をレジスタQYに指定されるメモリにセト
B12n	LD (QY), L	2	(QY)←L	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタQYに指定されるメモリにセト

【図 3 2 3】

<LDQ (k), r (命令NoB14): 8ビットロード命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	CY	
B14a	LDQ (k), B	3	(Qk)←B	・	Φ	・	・	・	・	・	レジスタBの内容をレジスタQ、8ビット-外の指定されたリにセット
B14b	LDQ (k), C	3	(Qk)←C	・	・	Φ	・	・	・	・	レジスタCの内容をレジスタQ、8ビット-外の指定されたリにセット
B14c	LDQ (k), D	3	(Qk)←D	・	・	Φ	・	・	・	・	レジスタDの内容をレジスタQ、8ビット-外の指定されたリにセット
B14d	LDQ (k), E	3	(Qk)←E	・	・	Φ	・	・	・	・	レジスタEの内容をレジスタQ、8ビット-外の指定されたリにセット
B14e	LDQ (k), H	3	(Qk)←H	・	・	Φ	・	・	・	・	レジスタHの内容をレジスタQ、8ビット-外の指定されたリにセット
B14f	LDQ (k), L	3	(Qk)←L	・	・	Φ	・	・	・	・	レジスタLの内容をレジスタQ、8ビット-外の指定されたリにセット

【図 3 2 4】

<16ビットロード命令(その1)>

No	新	ニーモニック	バイト 数	オペレーション	フラグ							説明
					S	Z	TZ	H	P/V	N	CY	
C1		LD B0, mn	3	B0←mn	・	・	Φ	・	・	・	・	16ビット-外の指定されたレジスタにセット
C2		LD DE, mn	3	DE←mn	・	・	Φ	・	・	・	・	16ビット-外の指定されたレジスタにセット
C3		LD HL, mn	3	HL←mn	・	・	Φ	・	・	・	・	16ビット-外の指定されたレジスタにセット
C4		LD SP, mn	3	SP←mn	・	・	Φ	・	・	・	・	16ビット-外の指定されたレジスタにセット
C5		LD IX, mn	4	IX←mn	・	・	Φ	・	・	・	・	16ビット-外の指定されたレジスタにセット
C6		LD IY, mn	4	IY←mn	・	・	Φ	・	・	・	・	16ビット-外の指定されたレジスタにセット
C7		LDQ BC, (k)	3	C←(Qk) B←(Qk+1)	・	・	Φ	・	・	・	・	レジスタQと8ビット-外の指定されたレジスタの内容をレジスタBCにセット
C8		LDQ DE, (k)	3	E←(Qk) D←(Qk+1)	・	・	Φ	・	・	・	・	レジスタQと8ビット-外の指定されたレジスタの内容をレジスタDEにセット
C9		LDQ HL, (k)	2	L←(Qk) H←(Qk+1)	・	・	Φ	・	・	・	・	レジスタQと8ビット-外の指定されたレジスタの内容をレジスタHLにセット
C10		LDQ (k), BC	3	(Qk)←C (Qk+1)←B	・	・	Φ	・	・	・	・	レジスタBCの内容をレジスタQと8ビット-外の指定されたレジスタにセット
C11		LDQ (k), DE	3	(Qk)←E (Qk+1)←D	・	・	Φ	・	・	・	・	レジスタDEの内容をレジスタQと8ビット-外の指定されたレジスタにセット
C12		LDQ (k), HL	2	(Qk)←L (Qk+1)←H	・	・	Φ	・	・	・	・	レジスタHLの内容をレジスタQと8ビット-外の指定されたレジスタにセット

【図 3 2 5】

<16ビットロード命令(その2)>

No	新	ニーモニック	バイト 数	オペレーション	フラグ							説明	
					S	Z	TZ	H	P/V	N	CY		
C13	○	LDF NZ, BC, mn	4	if Z=0 BC←mn(B←m, C←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタBCに16ビット-4mnをロード
C14	○	LDF Z, BC, mn	4	if Z=1 BC←mn(B←m, C←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタBCに16ビット-4mnをロード
C15	○	LDF NG, BC, mn	4	if CY=0 BC←mn(B←m, C←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタBCに16ビット-4mnをロード
C16	○	LDF O, BC, mn	4	if CY=1 BC←mn(B←m, C←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタBCに16ビット-4mnをロード
C17	○	LDF NZ, DE, mn	4	if Z=0 DE←mn(D←m, E←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタDEに16ビット-4mnをロード
C18	○	LDF Z, DE, mn	4	if Z=1 DE←mn(D←m, E←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタDEに16ビット-4mnをロード
C19	○	LDF NG, DE, mn	4	if CY=0 DE←mn(D←m, E←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタDEに16ビット-4mnをロード
C20	○	LDF O, DE, mn	4	if CY=1 DE←mn(D←m, E←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタDEに16ビット-4mnをロード
C21	○	LDF NZ, HL, mn	4	if Z=0 HL←mn(H←m, L←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタHLに16ビット-4mnをロード
C22	○	LDF Z, HL, mn	4	if Z=1 HL←mn(H←m, L←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタHLに16ビット-4mnをロード
C23	○	LDF NG, HL, mn	4	if CY=0 HL←mn(H←m, L←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタHLに16ビット-4mnをロード
C24	○	LDF O, HL, mn	4	if CY=1 HL←mn(H←m, L←n)	・	・	Φ	・	・	・	・	・	フラグがセットならレジスタHLに16ビット-4mnをロード

【図 3 2 6】

<ジャンプ命令(その1)>

No	新	ニーモニック	バイト数	オペレーション	フラグ								説明
					S	Z	TZ	H	P/V	N	CY		
D1		JP (HL)	1	PC←HL	・	・	・	・	・	・	・	・	レジスタHLの内容をプログラムのカウンタにセット
D2		JP (IX)	2	PC←IX	・	・	・	・	・	・	・	・	レジスタIXの内容をプログラムのカウンタにセット
D3		JP (IY)	2	PC←IY	・	・	・	・	・	・	・	・	レジスタIYの内容をプログラムのカウンタにセット
D4		JR e	2	PC←PC+e	・	・	・	・	・	・	・	・	本命令のオペランドがある番地からの値だけ離れた番地にジャンプ(eは8ビットの符号付整数)
D5		JR ca, e	2	if cc=OK PC←PC+e	・	・	・	・	・	・	・	・	ca値なら本命令のオペランドがある番地からの値だけ離れた番地にジャンプ(eは8ビットの符号付整数)
D6	○	TJP	1	PC←(PC+1+A)	・	・	・	・	・	・	・	・	プログラムのカウンタにレジスタAの内容を加算し、その値が指定する16ビットの外の値をプログラムのカウンタの下位にセット
D7	○	TJP B	2	PC←(PC+2+B)	・	・	・	・	・	・	・	・	プログラムのカウンタにレジスタBの内容を加算し、その値が指定する16ビットの外の値の内容をプログラムのカウンタの下位にセット
D8	○	TJP C	2	PC←(PC+2+C)	・	・	・	・	・	・	・	・	プログラムのカウンタにレジスタCの内容を加算し、その値が指定する16ビットの外の値の内容をプログラムのカウンタの下位にセット
D9	○	TJP D	2	PC←(PC+2+D)	・	・	・	・	・	・	・	・	プログラムのカウンタにレジスタDの内容を加算し、その値が指定する16ビットの外の値の内容をプログラムのカウンタの下位にセット
D10	○	TJP E	2	PC←(PC+2+E)	・	・	・	・	・	・	・	・	プログラムのカウンタにレジスタEの内容を加算し、その値が指定する16ビットの外の値の内容をプログラムのカウンタの下位にセット
D11	○	TJP H	2	PC←(PC+2+H)	・	・	・	・	・	・	・	・	プログラムのカウンタにレジスタHの内容を加算し、その値が指定する16ビットの外の値の内容をプログラムのカウンタの下位にセット

【図 3 2 7】

＜ジャンプ命令(その2)＞

No	新	二ーモニック	ハ/件数	オペレーション	フラグ							説明
					S	Z	TZ	H	P/V	N	OY	
D12	○	TJP L	2	PO←(PO+2)L	・	・	・	・	・	・	・	フワリがフワリに2レジスタの内容を加算し、その値が指定するハ/件のメモリの内容をフワリがフワリの下位にロード
D13	○	TJPW	1	PO←(PO+1+A+2)	・	・	・	・	・	・	・	フワリがフワリに2レジスタの内容を加算し、その値が指定するハ/件のメモリの内容をフワリがフワリにロード
D14	○	TJPW B	2	PO←(PO+2+B+2)	・	・	・	・	・	・	・	フワリがフワリに2レジスタの内容を加算し、その値が指定するハ/件のメモリの内容をフワリがフワリにロード
D15	○	TJPW C	2	PO←(PO+2+C+2)	・	・	・	・	・	・	・	フワリがフワリに2レジスタの内容を加算し、その値が指定するハ/件のメモリの内容をフワリがフワリにロード
D16	○	TJPW D	2	PO←(PO+2+D+2)	・	・	・	・	・	・	・	フワリがフワリに2レジスタの内容を加算し、その値が指定するハ/件のメモリの内容をフワリがフワリにロード
D17	○	TJPW E	2	PO←(PO+2+E+2)	・	・	・	・	・	・	・	フワリがフワリに2レジスタの内容を加算し、その値が指定するハ/件のメモリの内容をフワリがフワリにロード
D18	○	TJPW H	2	PO←(PO+2+H+2)	・	・	・	・	・	・	・	フワリがフワリに2レジスタの内容を加算し、その値が指定するハ/件のメモリの内容をフワリがフワリにロード
D19	○	TJPW L	2	PO←(PO+2+L+2)	・	・	・	・	・	・	・	フワリがフワリに2レジスタの内容を加算し、その値が指定するハ/件のメモリの内容をフワリがフワリにロード

【図 3 2 8】

＜JR co, e (命令No.D5)ジャンプ命令＞

No	二ーモニック	ハ/件数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
D5a	JR NZ, e	2	if Z=0 PC←PC+e	・	・	・	・	・	・	・	フワリがフワリに本命令のオペコードからeの値だけ離れた番地にジャンプ(eは8ビットの符号付整数)
D5b	JR Z, e	2	if Z=1 PC←PC+e	・	・	・	・	・	・	・	フワリがフワリに本命令のオペコードからeの値だけ離れた番地にジャンプ(eは8ビットの符号付整数)
D5c	JR NO, e	2	if OY=0 PC←PC+e	・	・	・	・	・	・	・	フワリがフワリに本命令のオペコードからeの値だけ離れた番地にジャンプ(eは8ビットの符号付整数)
D5d	JR O, e	2	if OY=1 PC←PC+e	・	・	・	・	・	・	・	フワリがフワリに本命令のオペコードからeの値だけ離れた番地にジャンプ(eは8ビットの符号付整数)
D5e	JR NTZ, e	2	if TZ=0 PC←PC+e	・	・	・	・	・	・	・	フワリがフワリに本命令のオペコードからeの値だけ離れた番地にジャンプ(eは8ビットの符号付整数)
D5f	JR TZ, e	2	if TZ=1 PC←PC+e	・	・	・	・	・	・	・	フワリがフワリに本命令のオペコードからeの値だけ離れた番地にジャンプ(eは8ビットの符号付整数)

【図 3 2 9】

＜コンペアロード命令_加算(その1)＞

No	新	二一モニック	ハ/件数	オペレーション	フラグ										説明
					S	Z	TZ	H	P/V	N	OY				
E1	○	ICPLD A, r	2	A← if OY=1 A←A+1 else A←0	○	○	○	○	○	○	1	○	○	レジスタAとレジスタrの値を比較し、比較結果でレジスタAの値が大きい場合、レジスタAを1増加し、それ以外はレジスタAに0をセット	
E2	○	ICPLD A, n	3	A←n if OY=1 A←A+1 else A←0	○	○	○	○	○	○	1	○	○	レジスタAとレジスタnの値を比較し、比較結果でレジスタAの値が大きい場合、レジスタAを加算し、それ以外はレジスタAに0をセット	
E3	○	ICPLD (HL), r	2	(HL)← if OY=1 ((HL)+(HL)+1 else (HL)←0	○	○	○	○	○	○	1	○	○	レジスタHLとレジスタrが指定するメモリ上の値1バイトを比較し、比較結果でレジスタHLの値が大きい場合、レジスタHLの値を1増加し、それ以外はレジスタHLが指定するメモリ上の値をセト	
E4	○	ICPLD (HL), n	3	(HL)←n if OY=1 ((HL)+(HL)+1 else (HL)←0	○	○	○	○	○	○	1	○	○	レジスタHLとレジスタnが指定するメモリ上の値1バイトを比較し、比較結果でレジスタHLの値が大きい場合、レジスタHLの値を1増加し、それ以外はレジスタHLが指定するメモリ上の値をセト	
E5	○	ICPLD HL, mn	4	HL←mn if OY=1 HL←HL+1 else HL←0	○	○	○	○	○	○	1	○	○	レジスタHLの値とレジスタmnを比較し、比較結果でレジスタHLの値が大きい場合、レジスタHLの値を1増加し、それ以外はレジスタHLが指定するメモリ上の値をセト	
E6	○	ICPWL D (HL), mn	4	((HL+1)D)←mn if OY=1 ((HL+1)D)←mn else ((HL+1)D)←0	○	○	○	○	○	○	1	○	○	レジスタHLが指定するメモリ上の値1バイトの値とレジスタmnを比較し、比較結果でレジスタHLが指定するメモリ上の値が大きい場合、レジスタHLの値を1増加し、それ以外はレジスタHLが指定するメモリ上の値をセト	
E7	○	ICPUL A, B	2	A←B if OY=1 A←A+1 else A←B	○	○	○	○	○	○	1	○	○	レジスタAとレジスタBの値を比較し、比較結果でレジスタAの値が大きい場合、レジスタAを加算し、それ以外はレジスタAにレジスタBの値をセト	
E8	○	ICPLD A, C	2	A←C if OY=1 A←A+1 else A←C	○	○	○	○	○	○	1	○	○	レジスタAとレジスタCの値を比較し、比較結果でレジスタAの値が大きい場合、レジスタAを加算し、それ以外はレジスタAにレジスタCの値をセト	

【図 3 3 0】

＜コンペアロード命令_加算(その2)＞

No	新	二ーモニック	ハ/件数	オペレーション	フラグ							説明	
					S	Z	TZ	H	P/V	N	OY		
E9	○	ICPLL A, D	2	A←D if OY=1 A←A+1 else A←D	○	○	○	○	○	○	1	○	レジスタAとレジスタDの値を比較し、比較結果でレジスタDの値が大きい場合、レジスタAを1増加し、それ以外はレジスタAとレジスタDの値をセト
E10	○	ICPLL A, E	2	A←E if OY=1 A←A+1 else A←E	○	○	○	○	○	○	1	○	レジスタAとレジスタEの値を比較し、比較結果でレジスタEの値が大きい場合、レジスタAを1増加し、それ以外はレジスタAとレジスタEの値をセト
E11	○	ICPLL A, H	2	A←H if OY=1 A←A+1 else A←H	○	○	○	○	○	○	1	○	レジスタAとレジスタHの値を比較し、比較結果でレジスタHの値が大きい場合、レジスタAを1増加し、それ以外はレジスタAとレジスタHの値をセト
E12	○	ICPLL A, L	2	A←L if OY=1 A←A+1 else A←L	○	○	○	○	○	○	1	○	レジスタAとレジスタLの値を比較し、比較結果でレジスタLの値が大きい場合、レジスタAを1増加し、それ以外はレジスタAとレジスタLの値をセト
E13	○	ICPLL A, n	3	A←n if OY=1 A←A+1 else A←n	○	○	○	○	○	○	1	○	レジスタAとレジスタnの値を比較し、比較結果でレジスタnの値が大きい場合、レジスタAを1増加し、それ以外はレジスタAとレジスタnの値をセト
E14	○	ICPLL B, n	3	B←n if OY=1 B←B+1 else B←n	○	○	○	○	○	○	1	○	レジスタBとレジスタnの値を比較し、比較結果でレジスタnの値が大きい場合、レジスタBを1増加し、それ以外はレジスタBとレジスタnの値をセト
E15	○	ICPLL C, n	3	C←n if OY=1 C←C+1 else C←n	○	○	○	○	○	○	1	○	レジスタCとレジスタnの値を比較し、比較結果でレジスタnの値が大きい場合、レジスタCを1増加し、それ以外はレジスタCとレジスタnの値をセト
E16	○	ICPLL D, n	3	D←n if OY=1 D←D+1 else D←n	○	○	○	○	○	○	1	○	レジスタDとレジスタnの値を比較し、比較結果でレジスタnの値が大きい場合、レジスタDを1増加し、それ以外はレジスタDとレジスタnの値をセト
E17	○	ICPLL E, n	3	E←n if OY=1 E←E+1 else E←n	○	○	○	○	○	○	1	○	レジスタEとレジスタnの値を比較し、比較結果でレジスタnの値が大きい場合、レジスタEを1増加し、それ以外はレジスタEとレジスタnの値をセト

【図 3 3 1】

＜コンペアロード命令・加算（その3）＞

No	新 ニーモニック	バイト 数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
E18	O	3	H ← n if OY = 1 H ← H + 1 else H ← n	0	0	0	0	0	1	0	レジスタHとレジスタnの値を比較し、比較結果で8ビットデータの値が大きい場合、レジスタHを1加算し、それ以外レジスタHに0をセット
E19	O	3	L ← n if OY = 1 L ← L + 1 else L ← n	0	0	0	0	0	0	1	レジスタLとレジスタnの値を比較し、比較結果で8ビットデータの値が大きい場合、レジスタLを1加算し、それ以外レジスタLに0をセット
E20	O	4	BC ← mm if OY = 1 BC ← BC + 1 else BC ← mm(B ← m, C ← n)	0	0	0	0	0	0	1	レジスタBCと16ビットデータの値を比較し、比較結果で16ビットデータの値が大きい場合、レジスタBCを1加算し、それ以外レジスタBCに0をセット
E21	O	4	DE ← mm if OY = 1 DE ← DE + 1 else DE ← mm(D ← m, E ← n)	0	0	0	0	0	0	1	レジスタDEと16ビットデータの値を比較し、比較結果で16ビットデータの値が大きい場合、レジスタDEを1加算し、それ以外レジスタDEに0をセット
E22	O	4	HL ← mm if OY = 1 HL ← HL + 1 else HL ← mm(H ← m, L ← n)	0	0	0	0	0	0	1	レジスタHLと16ビットデータの値を比較し、比較結果で16ビットデータの値が大きい場合、レジスタHLを1加算し、それ以外レジスタHLに0をセット

【図 3 3 2】

＜ICPLD A, r（命令No.E1）:コンペアロード命令・加算＞

No	ニーモニック	バイト数	オペレーション	フラグ								説明
				S	Z	TZ	H	P/V	N	OY		
E1a	ICPLD A, B	2	A←B if OY=1 A←A+1 else A←0	0	0	0	0	0	1	0	レジスタAとレジスタBの値を比較し、比較結果でレジスタBの値が大きい場合、レジスタAを1加算し、それ以外レジスタAに0をセット	
E1b	ICPLD A, C	2	A←C if OY=1 A←A+1 else A←0	0	0	0	0	0	1	0	レジスタAとレジスタCの値を比較し、比較結果でレジスタCの値が大きい場合、レジスタAを1加算し、それ以外レジスタAに0をセット	
E1c	ICPLD A, D	2	A←D if OY=1 A←A+1 else A←0	0	0	0	0	0	1	0	レジスタAとレジスタDの値を比較し、比較結果でレジスタDの値が大きい場合、レジスタAを1加算し、それ以外レジスタAに0をセット	
E1d	ICPLD A, E	2	A←E if OY=1 A←A+1 else A←0	0	0	0	0	0	1	0	レジスタAとレジスタEの値を比較し、比較結果でレジスタEの値が大きい場合、レジスタAを1加算し、それ以外レジスタAに0をセット	
E1e	ICPLD A, H	2	A←H if OY=1 A←A+1 else A←0	0	0	0	0	0	1	0	レジスタAとレジスタHの値を比較し、比較結果でレジスタHの値が大きい場合、レジスタAを1加算し、それ以外レジスタAに0をセット	
E1f	ICPLD A, L	2	A←L if OY=1 A←A+1 else A←0	0	0	0	0	0	1	0	レジスタAとレジスタLの値を比較し、比較結果でレジスタLの値が大きい場合、レジスタAを1加算し、それ以外レジスタAに0をセット	

【図 3 3 3】

＜ICPLD (HL), r（命令No.E3）:コンペアロード命令・加算＞

No	ニーモニック	バイト数	オペレーション	フラグ								説明
				S	Z	TZ	H	P/V	N	OY		
E3a	ICPLD (HL), A	2	(HL)←A if OY=1 (HL)←(HL)+1 else (HL)←0	0	0	0	0	0	1	0	レジスタAが指定するメモリ上のレジスタAの値と比較し、比較結果でレジスタAの値が大きい場合、レジスタHLが指定するメモリ上のレジスタHLの値を1を加算し、それ以外はレジスタHLが指定するメモリに0をセット	
E3b	ICPLD (HL), B	2	(HL)←B if OY=1 (HL)←(HL)+1 else (HL)←0	0	0	0	0	0	1	0	レジスタBが指定するメモリ上のレジスタBの値と比較し、比較結果でレジスタBの値が大きい場合、レジスタHLが指定するメモリ上のレジスタHLの値を1を加算し、それ以外はレジスタHLが指定するメモリに0をセット	
E3c	ICPLD (HL), C	2	(HL)←C if OY=1 (HL)←(HL)+1 else (HL)←0	0	0	0	0	0	1	0	レジスタCが指定するメモリ上のレジスタCの値と比較し、比較結果でレジスタCの値が大きい場合、レジスタHLが指定するメモリ上のレジスタHLの値を1を加算し、それ以外はレジスタHLが指定するメモリに0をセット	
E3d	ICPLD (HL), D	2	(HL)←D if OY=1 (HL)←(HL)+1 else (HL)←0	0	0	0	0	0	1	0	レジスタDが指定するメモリ上のレジスタDの値と比較し、比較結果でレジスタDの値が大きい場合、レジスタHLが指定するメモリ上のレジスタHLの値を1を加算し、それ以外はレジスタHLが指定するメモリに0をセット	
E3e	ICPLD (HL), E	2	(HL)←E if OY=1 (HL)←(HL)+1 else (HL)←0	0	0	0	0	0	1	0	レジスタEが指定するメモリ上のレジスタEの値と比較し、比較結果でレジスタEの値が大きい場合、レジスタHLが指定するメモリ上のレジスタHLの値を1を加算し、それ以外はレジスタHLが指定するメモリに0をセット	

【図 3 3 4】

＜コンペアロード命令・減算（その1）＞

No	新 ニーモニック	バイト 数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
F1	DCPLD A, r	2	A←1 if OY=1 A←r else A←A-1	0	0	0	0	0	1	0	レジスタAとの値を比較し、比較結果が1の値より小さい場合、レジスタAにレジスタrをセットし、それ以外ではレジスタAから1減算
F2	DCPLD A, n	3	A←1 if OY=0 A←A-1 else A←n	0	0	0	0	0	1	0	レジスタAとの値を比較し、比較結果が1の値より小さい場合、レジスタAにレジスタnにセットされた値をセットし、それ以外ではレジスタAから1減算
F3	DCPLD (HL), r	2	(HL)←1 if OY=1 (HL)←r else (HL)←(HL)-1	0	0	0	0	0	1	0	アドレスHLが指定するメモリ上の値とレジスタHLの値を比較し、比較結果が1の値より小さい場合、アドレスHLが指定するメモリにレジスタrをセットし、それ以外ではアドレスHLが指定するメモリから1減算
F4	DCPLD (HL), n	3	(HL)←1 if OY=1 (HL)←n else (HL)←(HL)-1	0	0	0	0	0	1	0	アドレスHLが指定するメモリ上の値とレジスタHLの値を比較し、比較結果が1の値より小さい場合、アドレスHLが指定するメモリにレジスタnをセットし、それ以外ではアドレスHLが指定するメモリから1減算
F5	DCPLD HL, mm	4	HL←1 if OY=1 HL←mm(H←m, L←n) else HL←HL-1	0	0	0	0	0	1	0	アドレスHLの値とレジスタHLの値を比較し、比較結果が1の値より小さい場合、アドレスHLに16ビットでmmをセットし、それ以外ではアドレスHLから1減算
F6	DCPWLD (HL), mm	4	(HL+1)←(HL)-1 if OY=1 (HL+1)←(HL)-mm else (HL+1)←(HL)-1	0	0	0	0	0	1	0	アドレスHLが指定するメモリ上のレジスタHLの値とレジスタHL+1の値を比較し、比較結果が1の値より小さい場合、アドレスHLが指定するメモリ上のレジスタHL+1に16ビットでmmをセットし、それ以外ではアドレスHLが指定するメモリ上のレジスタHL+1から1減算
F7	O DCPHL A, B	2	A←B if OY=0 A←A-1 else A←B	0	0	0	0	0	1	0	レジスタAの値とレジスタBの値を比較し、比較結果でレジスタBの値が小さい場合、レジスタAの値を1減算し、それ以外ではレジスタAにレジスタBの値をセット
F8	O DCPHL A, C	2	A←C if OY=0 A←A-1 else A←C	0	0	0	0	0	1	0	レジスタAの値とレジスタCの値を比較し、比較結果でレジスタCの値が小さい場合、レジスタAの値を1減算し、それ以外ではレジスタAにレジスタCの値をセット

【図 3 3 5】

<コンパイルロード命令・減算(その2)>

No	新	ニーモニック	バイト数	オペレーション	フラグ							説明	
					S	Z	TZ	H	P/V	N	OY		
F9	O	DOPLL A, D	2	A-D if OY=0 A←A-1 else A←D	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタOの値とレジスタDの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタOの値を1減算し、それ以外はレジスタOにレジスタDの値をロード
F10	O	DOPLL A, E	2	A-E if OY=0 A←A-1 else A←E	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタOの値とレジスタEの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタOの値を1減算し、それ以外はレジスタOにレジスタEの値をロード
F11	O	DOPLL A, H	2	A-H if OY=0 A←A-1 else A←H	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタOの値とレジスタHの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタOの値を1減算し、それ以外はレジスタOにレジスタHの値をロード
F12	O	DOPLL A, L	2	A-L if OY=0 A←A-1 else A←L	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタOの値とレジスタLの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタOの値を1減算し、それ以外はレジスタOにレジスタLの値をロード
F13	O	DOPLL A, n	3	A-n if OY=0 A←A-1 else A←n	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタOの値とレジスタnの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタOの値を1減算し、それ以外はレジスタOにレジスタnの値をロード
F14	O	DOPLL B, n	3	B-n if OY=0 B←B-1 else B←n	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタBの値とレジスタnの値を比較し、比較結果でレジスタBの値が小さい場合、レジスタBの値を1減算し、それ以外はレジスタBにレジスタnの値をロード
F15	O	DOPLL C, n	3	C-n if OY=0 C←C-1 else C←n	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタCの値とレジスタnの値を比較し、比較結果でレジスタCの値が小さい場合、レジスタCの値を1減算し、それ以外はレジスタCにレジスタnの値をロード
F16	O	DOPLL D, n	3	D-n if OY=0 D←D-1 else D←n	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタDの値とレジスタnの値を比較し、比較結果でレジスタDの値が小さい場合、レジスタDの値を1減算し、それ以外はレジスタDにレジスタnの値をロード
F17	O	DOPLL E, n	3	E-n if OY=0 E←E-1 else E←n	⊙	⊙	⊙	⊙	⊙	⊙	⊙		レジスタEの値とレジスタnの値を比較し、比較結果でレジスタEの値が小さい場合、レジスタEの値を1減算し、それ以外はレジスタEにレジスタnの値をロード

【図 3 3 6】

<コンパイルロード命令・減算(その3)>

No	ニーモニック	オペレーション	バイト数	フラグ							説明
				S	Z	TZ	H	P/V	N	OY	
F18	DOPLL H, n	H-n if CY=0 HL←H-1 else HL←n	3	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	レジスタHの値とレジスタnの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタHの値を1減算し、それ以外はレジスタHにレジスタnの値をロード	
F19	DOPLL L, n	L-n if CY=0 HL←L-1 else HL←n	3	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	レジスタLの値とレジスタnの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタLの値を1減算し、それ以外はレジスタLにレジスタnの値をロード	
F20	DOPLL B0, mn	BC-mn if CY=0 BC←BC-1 else BC←mn(B←m, C←n)	4	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	レジスタB0の値とレジスタmnの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタB0の値を1減算し、それ以外はレジスタB0にレジスタmnの値をロード	
F21	DOPLL DE, mn	DE-mn if CY=0 DE←DE-1 else DE←mn(D←m, E←n)	4	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	レジスタDEの値とレジスタmnの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタDEの値を1減算し、それ以外はレジスタDEにレジスタmnの値をロード	
F22	DOPLL HL, mn	HL-mn if CY=0 HL←HL-1 else HL←mn(H←m, L←n)	4	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	Ⓜ	レジスタHLの値とレジスタmnの値を比較し、比較結果でレジスタOの値が小さい場合、レジスタHLの値を1減算し、それ以外はレジスタHLにレジスタmnの値をロード	

【図 3 3 7】

<DOPLD A, r (命令No.F1):コンパイルロード命令・減算>

No	ニーモニック	バイト数	オペレーション	フラグ							説明	
				S	Z	TZ	H	P/V	N	OY		
F1a	DOPLD A, B	2	A-1 if OY=1 A-B else A-A-1	0	0	0	0	0	1	0		レジスタAの値と比較し、比較結果が1の値より小さい場合、レジスタAにレジスタBの値をせし、それ以外はレジスタAから1減算
F1b	DOPLD A, C	2	A-1 if OY=1 A-C else A-A-1	0	0	0	0	0	1	0		レジスタAの値と比較し、比較結果が1の値より小さい場合、レジスタAにレジスタCの値をせし、それ以外はレジスタAから1減算
F1c	DOPLD A, D	2	A-1 if OY=1 A-D else A-A-1	0	0	0	0	0	1	0		レジスタAの値と比較し、比較結果が1の値より小さい場合、レジスタAにレジスタDの値をせし、それ以外はレジスタAから1減算
F1d	DOPLD A, E	2	A-1 if OY=1 A-E else A-A-1	0	0	0	0	0	1	0		レジスタAの値と比較し、比較結果が1の値より小さい場合、レジスタAにレジスタEの値をせし、それ以外はレジスタAから1減算
F1e	DOPLD A, H	2	A-1 if OY=1 A-H else A-A-1	0	0	0	0	0	0	1		レジスタAの値と比較し、比較結果が1の値より小さい場合、レジスタAにレジスタHの値をせし、それ以外はレジスタAから1減算
F1f	DOPLD A, L	2	A-1 if OY=1 A-L else A-A-1	0	0	0	0	0	0	1		レジスタAの値と比較し、比較結果が1の値より小さい場合、レジスタAにレジスタLの値をせし、それ以外はレジスタAから1減算

【図 3 3 8】

<DOPLD (HL), r (命令No.F3):コンパイルロード命令・減算>

No	ニーモニック	オペレーション	バイト数	フラグ							説明	
				S	Z	TZ	H	P/V	N	OY		
F3a	DOPLD (HL), A	(HL)-1 if OY=1 (HL)-A else (HL)-(HL)-1	2	0	0	0	0	0	1	0		レジスタHLが指定するメモリとの値と比較し、比較結果が1の値より小さい場合、レジスタHLが指定するメモリにレジスタAの値をロードし、それ以外はレジスタHLが指定するメモリから1を減算
F3b	DOPLD (HL), B	(HL)-1 if OY=1 (HL)-B else (HL)-(HL)-1	2	0	0	0	0	0	1	0		レジスタHLが指定するメモリとの値と比較し、比較結果が1の値より小さい場合、レジスタHLが指定するメモリにレジスタBの値をロードし、それ以外はレジスタHLが指定するメモリから1を減算
F3c	DOPLD (HL), C	(HL)-1 if OY=1 (HL)-C else (HL)-(HL)-1	2	0	0	0	0	0	1	0		レジスタHLが指定するメモリとの値と比較し、比較結果が1の値より小さい場合、レジスタHLが指定するメモリにレジスタCの値をロードし、それ以外はレジスタHLが指定するメモリから1を減算
F3d	DOPLD (HL), D	(HL)-1 if OY=1 (HL)-D else (HL)-(HL)-1	2	0	0	0	0	0	1	0		レジスタHLが指定するメモリとの値と比較し、比較結果が1の値より小さい場合、レジスタHLが指定するメモリにレジスタDの値をロードし、それ以外はレジスタHLが指定するメモリから1を減算
F3e	DOPLD (HL), E	(HL)-1 if OY=1 (HL)-E else (HL)-(HL)-1	2	0	0	0	0	0	1	0		レジスタHLが指定するメモリとの値と比較し、比較結果が1の値より小さい場合、レジスタHLが指定するメモリにレジスタEの値をロードし、それ以外はレジスタHLが指定するメモリから1を減算

【図 3 3 9】

＜シフト命令・左＞

No	新	ニーモニック	オペレーション	ビット 数	フラグ										説明
					S	Z	TZ	H	P/V	N	OY				
G1		SLA r	CY←(Bk7~Bk0)←0	2	0	0	0	0	0	0	0	0	レジスタの内容を1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bk0をセロ		
G2		SLA (HL)	CY←(HL(Bk7~Bk0))←0	2	0	0	0	0	0	0	0	0	アドレスHLで指定するメモリの内容をCYフラグにセットし、シフト前のBk7の内容をCYフラグにセットし、Bk0をセロ		
G3	○	SLBIT A, n	CY←A(Bk7~Bkn)	3	0	0	0	0	0	0	0	0	レジスタAの内容をnで指定されたBk位置から1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bkn-1からBk0はシフト前の値を維持		
G4	○	SLBIT B, n	CY←B(Bk7~Bkn)	3	0	0	0	0	0	0	0	0	レジスタBの内容をnで指定されたBk位置から1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bkn-1からBk0はシフト前の値を維持		
G5	○	SLBIT C, n	CY←C(Bk7~Bkn)	3	0	0	0	0	0	0	0	0	レジスタCの内容をnで指定されたBk位置から1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bkn-1からBk0はシフト前の値を維持		
G6	○	SLBIT D, n	CY←D(Bk7~Bkn)	3	0	0	0	0	0	0	0	0	レジスタDの内容をnで指定されたBk位置から1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bkn-1からBk0はシフト前の値を維持		
G7	○	SLBIT E, n	CY←E(Bk7~Bkn)	3	0	0	0	0	0	0	0	0	レジスタEの内容をnで指定されたBk位置から1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bkn-1からBk0はシフト前の値を維持		
G8	○	SLBIT H, n	CY←H(Bk7~Bkn)	3	0	0	0	0	0	0	0	0	レジスタHの内容をnで指定されたBk位置から1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bkn-1からBk0はシフト前の値を維持		
G9	○	SLBIT L, n	CY←L(Bk7~Bkn)	3	0	0	0	0	0	0	0	0	レジスタLの内容をnで指定されたBk位置から1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bkn-1からBk0はシフト前の値を維持		
G10	○	SLBITQ (k), n	CY←k(Bk7~Bkn)	4	0	0	0	0	0	0	0	0	レジスタQと数値kで指定されたメモリの内容をnで指定されたBk位置から1BIT左にシフトし、シフト前のBk7の内容をCYフラグにセットし、Bkn-1からBk0はシフト前の値を維持		

【図 3 4 0】

＜SLA r (命令No.G1):シフト命令・左＞

No	ニーモニック	オペレーション	ビット 数	フラグ										説明
				S	Z	TZ	H	P/V	N	OY				
G1a	SLA A	CY←A(Bk7~Bk0)←0	2	0	0	0	0	0	0	0	0	レジスタAの内容を1BIT左にシフトし、 シフト前のBk7の内容をCYフラグにセットし、Bk0をリセット		
G1b	SLA B	CY←B(Bk7~Bk0)←0	2	0	0	0	0	0	0	0	0	レジスタBの内容を1BIT左にシフトし、 シフト前のBk7の内容をCYフラグにセットし、Bk0をリセット		
G1c	SLA C	CY←C(Bk7~Bk0)←0	2	0	0	0	0	0	0	0	0	レジスタCの内容を1BIT左にシフトし、 シフト前のBk7の内容をCYフラグにセットし、Bk0をリセット		
G1d	SLA D	CY←D(Bk7~Bk0)←0	2	0	0	0	0	0	0	0	0	レジスタDの内容を1BIT左にシフトし、 シフト前のBk7の内容をCYフラグにセットし、Bk0をリセット		
G1e	SLA E	CY←E(Bk7~Bk0)←0	2	0	0	0	0	0	0	0	0	レジスタEの内容を1BIT左にシフトし、 シフト前のBk7の内容をCYフラグにセットし、Bk0をリセット		
G1f	SLA H	CY←H(Bk7~Bk0)←0	2	0	0	0	0	0	0	0	0	レジスタHの内容を1BIT左にシフトし、 シフト前のBk7の内容をCYフラグにセットし、Bk0をリセット		
G1g	SLA L	CY←L(Bk7~Bk0)←0	2	0	0	0	0	0	0	0	0	レジスタLの内容を1BIT左にシフトし、 シフト前のBk7の内容をCYフラグにセットし、Bk0をリセット		

【図 3 4 1】

＜シフト命令・右＞

No	新	ニーモニック	オペレーション	ビット 数	フラグ										説明
					S	Z	TZ	H	P/V	N	OY				
H1		SRA r	Bk7←(Bk7~Bk0)←CY	2	0	0	0	0	0	0	0	レジスタの内容を1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bk7はシフト前の値を維持			
H2		SRA (HL)	Bk7←(H(Bk7~Bk0))←CY	2	0	0	0	0	0	0	0	アドレスHLで指定するメモリの内容を1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bk7はシフト前の値を維持			
H3	○	SRRBIT A, n	A(Bktn~Bk0)←CY	3	0	0	0	0	0	0	0	レジスタAの内容をnで指定されたBk位置から1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bktn+1からBktnはシフト前の値を維持			
H4	○	SRRBIT B, n	B(Bktn~Bk0)←CY	3	0	0	0	0	0	0	0	レジスタBの内容をnで指定されたBk位置から1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bktn+1からBktnはシフト前の値を維持			
H5	○	SRRBIT C, n	C(Bktn~Bk0)←CY	3	0	0	0	0	0	0	0	レジスタCの内容をnで指定されたBk位置から1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bktn+1からBktnはシフト前の値を維持			
H6	○	SRRBIT D, n	D(Bktn~Bk0)←CY	3	0	0	0	0	0	0	0	レジスタDの内容をnで指定されたBk位置から1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bktn+1からBktnはシフト前の値を維持			
H7	○	SRRBIT E, n	E(Bktn~Bk0)←CY	3	0	0	0	0	0	0	0	レジスタEの内容をnで指定されたBk位置から1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bktn+1からBktnはシフト前の値を維持			
H8	○	SRRBIT H, n	H(Bktn~Bk0)←CY	3	0	0	0	0	0	0	0	レジスタHの内容をnで指定されたBk位置から1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bktn+1からBktnはシフト前の値を維持			
H9	○	SRRBIT L, n	L(Bktn~Bk0)←CY	3	0	0	0	0	0	0	0	レジスタLの内容をnで指定されたBk位置から1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bktn+1からBktnはシフト前の値を維持			
H10	○	SRRBITQ (k), n	k(Bktn~Bk0)←CY	4	0	0	0	0	0	0	0	レジスタQと数値kで指定されたメモリの内容をnで指定されたBk位置から1BIT右にシフトし、シフト前のBk0の内容をCYフラグにセットし、Bktn+1からBktnはシフト前の値を維持			

【図 3 4 2】

＜SRA r (命令No.H1):シフト命令・右＞

No	ニーモニック	ビット 数	オペレーション	フラグ										説明
				S	Z	TZ	H	P/V	N	OY				
H1a	SLA A	2	BK7←A(BK7→BK0)←CY	0	0	0	0	0	0	0	0	0	レジスタAの内容を1BIT右にシフトし、シフト前のBK0の内容をCYフラグにセットし、BK7はシフト前の値を維持	
H1b	SRA B	2	BK7←B(BK7→BK0)←CY	0	0	0	0	0	0	0	0	0	レジスタBの内容を1BIT右にシフトし、シフト前のBK0の内容をCYフラグにセットし、BK7はシフト前の値を維持	
H1c	SRA C	2	BK7←C(BK7→BK0)←CY	0	0	0	0	0	0	0	0	0	レジスタCの内容を1BIT右にシフトし、シフト前のBK0の内容をCYフラグにセットし、BK7はシフト前の値を維持	
H1d	SRA D	2	BK7←D(BK7→BK0)←CY	0	0	0	0	0	0	0	0	0	レジスタDの内容を1BIT右にシフトし、シフト前のBK0の内容をCYフラグにセットし、BK7はシフト前の値を維持	
H1e	SRA E	2	BK7←E(BK7→BK0)←CY	0	0	0	0	0	0	0	0	0	レジスタEの内容を1BIT右にシフトし、シフト前のBK0の内容をCYフラグにセットし、BK7はシフト前の値を維持	
H1f	SRA H	2	BK7←H(BK7→BK0)←CY	0	0	0	0	0	0	0	0	0	レジスタHの内容を1BIT右にシフトし、シフト前のBK0の内容をCYフラグにセットし、BK7はシフト前の値を維持	
H1g	SRA L	2	BK7←L(BK7→BK0)←CY	0	0	0	0	0	0	0	0	0	レジスタLの内容を1BIT右にシフトし、シフト前のBK0の内容をCYフラグにセットし、BK7はシフト前の値を維持	

10

20

30

40

50

【図 3 4 3】

＜コールド・リターン命令（その1）＞

No	新	ニーモニック	オペレ 数	オペレーション	フラグ							説明	
					S	Z	TZ	H	P/V	N	CY		
I1		OALL mm	3	(SP-1)←PCH, (SP-2)←PCI PC←SP-2	•	•	•	•	•	•	•		本命令の次の命令に対応する707ラムダPCの値を、現在のスタック・インデックスで指示される2バイトにブランチし、mmを707ラムダPCに格納してmm番地にジャンプする。
I2	○	CALLD mm	3	IFF=0 (SP-1)←PCH, (SP-2)←PCI PC←mm, SP←SP-2	•	•	•	•	•	•	•		割込み・オペーブルフラグをセットしてINT割込みを無効にし、本命令の次の命令に対応するPCの値を、現在のスタック・インデックスで指示される2バイトにブランチし、mmを707ラムダPCに格納してmm番地にジャンプする。
I3	○	CALLE mm	3	IFF=1 (SP-1)←PCH, (SP-2)←PCI PC←mm, SP←SP-2	•	•	•	•	•	•	•		割込み・オペーブルフラグをセットしてINT割込みを有効にし、本命令の次の命令に対応する707ラムダPCの値を、現在のスタック・インデックスで指示される2バイトにブランチし、mmをPCに格納してmm番地にジャンプする。
I4		CALLF mm	2	(SP-1)←PCH, (SP-2)←PCI PCH←00H~11H, PCH←n SP←SP-2	•	•	•	•	•	•	•		本命令の次の命令に対応する707ラムダPCの値を、現在のSPで指示される2バイトにブランチし、mmを707ラムダPCに格納してmm番地にジャンプする。
I5		CALLED mm	2	IFF=0 (SP-1)←PCH, (SP-2)←PCI PCH←00H~11H, PCH←n SP←SP-2	•	•	•	•	•	•	•		割込み・オペーブルフラグをセットしてINT割込みを無効にし、本命令の次の命令に対応する707ラムダPCの値を、現在のスタック・インデックスで指示される2バイトにブランチし、mmを707ラムダPCに格納してmm番地にジャンプする。
I6	○	CALLEF mm	2	IFF=1 (SP-1)←PCH, (SP-2)←PCI PCH←00H~11H, PCH←n SP←SP-2	•	•	•	•	•	•	•		割込み・オペーブルフラグをセットしてINT割込みを有効にし、本命令の次の命令に対応する707ラムダPCの値を、現在のスタック・インデックスで指示される2バイトにブランチし、mmを707ラムダPCに格納してmm番地にジャンプする。

【図 3 4 4】

＜コールド・リターン命令（その2）＞

No	新	ニーモニック	オペレーション	バイト数	フラグ							説明	
					S	Z	TZ	H	P/V	N	CY		
I7		RET	POH←(SP), POH←(SP+1) SP←SP+2	1	•	•	•	•	•	•	•		スタック・インデックスで指示される2バイトから707ラムダPCにブランチして元の707ラムダに戻る。
I8	○	RETD	IFF←0 POH←(SP), POH←(SP+1) SP←SP+2	1	•	•	•	•	•	•	•		割込み・オペーブルフラグをセットしてINT割込みを無効にし、スタック・インデックスで指示される2バイトから707ラムダPCにブランチして元の707ラムダに戻る。
I9	○	RETE	IFF←1 POH←(SP), POH←(SP+1) SP←SP+2	1	•	•	•	•	•	•	•		割込み・オペーブルフラグをセットしてINT割込みを有効にし、スタック・インデックスで指示される2バイトから707ラムダPCにブランチして元の707ラムダに戻る。

【図 3 4 5】

＜CPUコントリール命令＞

No	新	ニーモニック	オペレーション	オペレ 数	フラグ							説明
					S	Z	TZ	H	P/V	N	CY	
J1		DI	IFF=0	1	・	・	・	・	・	・	割込み・オペーブルフラグをセットしてINT割込みを無効にする。	
J2		EI	IFF=1	1	・	・	・	・	・	・	割込み・オペーブルフラグをセットしてINT割込みを有効にする。	

【図 3 4 6】

＜論理演算命令＞

No	新	ニーモニック	オペレーション	フラグ										説明
				S	Z	TZ	H	P/V	N	CY				
K1		AND A	2	A←A∧A	0	0	0	0	1	0	0	レジスタAの内容とレジスタAの内容で論理積し、結果をレジスタAに格納		
K2		AND r	1	A←A∧r	0	0	0	0	1	0	0	レジスタAの内容とレジスタrの内容で論理積し、結果をレジスタAに格納		
K3		AND n	2	A←A∧n	0	0	0	0	1	0	0	レジスタAの内容とレジスタnの内容で論理積し、結果をレジスタAに格納		
K4		OR r	1	A←A∨r	0	0	0	0	0	0	0	レジスタAの内容とレジスタrの内容で論理和し、結果をレジスタAに格納		
K5		OR n	2	A←A∨n	0	0	0	0	0	0	0	レジスタAの内容とレジスタnの内容で論理和し、結果をレジスタAに格納		

【図 3 4 7】

<AND r (命令No.K2):論理演算命令>

No	ニーモニック	バイト数	オペレーション	フラグ								説明
				S	Z	TZ	H	P/V	N	CY		
K2a	AND B	1	A←A∧B	0	0	0	1	0	0	0	レジスタ0の内容とレジスタBの内容で論理積し、結果をレジスタAに格納	
K2b	AND C	1	A←A∧C	0	0	0	1	0	0	0	レジスタ0の内容とレジスタCの内容で論理積し、結果をレジスタAに格納	
K2c	AND D	1	A←A∧D	0	0	0	1	0	0	0	レジスタ0の内容とレジスタDの内容で論理積し、結果をレジスタAに格納	
K2d	AND E	1	A←A∧E	0	0	0	1	0	0	0	レジスタ0の内容とレジスタEの内容で論理積し、結果をレジスタAに格納	
K2e	AND H	1	A←A∧H	0	0	0	1	0	0	0	レジスタ0の内容とレジスタHの内容で論理積し、結果をレジスタAに格納	
K2f	AND L	1	A←A∧L	0	0	0	1	0	0	0	レジスタ0の内容とレジスタLの内容で論理積し、結果をレジスタAに格納	

【図 3 4 8】

<OR r (命令No.K4):論理演算命令>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	CY	
K4a	OR A	1	A←A∨A	0	0	0	0	0	0	レジスタAの内容とレジスタAの内容で論理和し、結果をレジスタAに格納	
K4b	OR B	1	A←A∨B	0	0	0	0	0	0	レジスタAの内容とレジスタBの内容で論理和し、結果をレジスタAに格納	
K4c	OR C	1	A←A∨C	0	0	0	0	0	0	レジスタAの内容とレジスタCの内容で論理和し、結果をレジスタAに格納	
K4d	OR D	1	A←A∨D	0	0	0	0	0	0	レジスタAの内容とレジスタDの内容で論理和し、結果をレジスタAに格納	
K4e	OR E	1	A←A∨E	0	0	0	0	0	0	レジスタAの内容とレジスタEの内容で論理和し、結果をレジスタAに格納	
K4f	OR H	1	A←A∨H	0	0	0	0	0	0	レジスタAの内容とレジスタHの内容で論理和し、結果をレジスタAに格納	
K4g	OR L	1	A←A∨L	0	0	0	0	0	0	レジスタAの内容とレジスタLの内容で論理和し、結果をレジスタAに格納	

【図 3 4 9】

<8ビット算術命令・比較>

No	新	ニーモニック	バイト数	オペレーション	フラグ							説明
					S	Z	TZ	H	P/V	N	CY	
L1		CP r	1	A←	0	0	0	0	0	0	レジスタAの内容とレジスタrの内容を比較し、結果をもとにフラグを設定	
L2		CP n	2	A←n	0	0	0	0	0	0	レジスタAの内容とレジスタnの内容を比較し、結果をもとにフラグを設定	
L3		CP (DE)	1	A←(DE)	0	0	0	0	0	0	レジスタAの内容とレジスタDEの内容を指定されるレジスタの内容と比較し、結果をもとにフラグを設定	
L4		CP (HL)	2	A←(HL)	0	0	0	0	0	0	レジスタAの内容とレジスタHLの内容で指定されるレジスタの内容と比較し、結果をもとにフラグを設定	
L5		CP r, (HL)	2	r←(HL)	0	0	0	0	0	0	レジスタrの内容とレジスタrの内容で指定されるレジスタの内容を比較し、結果をもとにフラグを設定	
L6		CPQ (k)	2	A←(k)	0	0	0	0	0	0	レジスタAの内容とレジスタkの内容とレジスタrの内容を指定されるレジスタの内容を比較し、結果をもとにフラグを設定	
L7		CPQ r, (k)	3	r←(k)	0	0	0	0	0	0	レジスタrの内容とレジスタkの内容とレジスタrの内容を指定されるレジスタの内容を比較し、結果をもとにフラグを設定	

【図 3 5 0】

<CP r (命令No.L1):8ビット算術命令・比較>

No	ニーモニック	バイト数	オペレーション	フラグ												説明
				S	Z	TZ	H	P/V	N	CY						
L1a	CP B	1	A←B	0	0	0	0	0	0	0	0	0	0	レジスタAの内容とレジスタBの内容を比較し、結果をもとにフラグを設定		
L1b	CP C	1	A←C	0	0	0	0	0	0	0	0	0	0	レジスタAの内容とレジスタCの内容を比較し、結果をもとにフラグを設定		
L1c	CP D	1	A←D	0	0	0	0	0	0	0	0	0	0	レジスタAの内容とレジスタDの内容を比較し、結果をもとにフラグを設定		
L1d	CP E	1	A←E	0	0	0	0	0	0	0	0	0	0	レジスタAの内容とレジスタEの内容を比較し、結果をもとにフラグを設定		
L1e	CP H	1	A←H	0	0	0	0	0	0	0	0	0	0	レジスタAの内容とレジスタHの内容を比較し、結果をもとにフラグを設定		
L1f	CP L	1	A←L	0	0	0	0	0	0	0	0	0	0	レジスタAの内容とレジスタLの内容を比較し、結果をもとにフラグを設定		

【図 3 5 1】

<OP_r (HL) (命令No.L5): 8ビット算術命令・比較>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	CY	
L5a	OP B, (HL)	2	B<(HL)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタBの内容とヘキサダシ74Hの内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L5b	OP C, (HL)	2	C<(HL)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタCの内容とヘキサダシ74Hの内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L5c	OP D, (HL)	2	D<(HL)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタDの内容とヘキサダシ74Hの内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L5d	OP E, (HL)	2	E<(HL)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタEの内容とヘキサダシ74Hの内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L5e	OP H, (HL)	2	H<(HL)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタHの内容とヘキサダシ74Hの内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L5f	OP L, (HL)	2	L<(HL)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタLの内容とヘキサダシ74Hの内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	

【図 3 5 3】

<16ビット算術命令・比較>

No	新	ニーモニック	バイト 数	オペレーション	フラグ							説明
					S	Z	TZ	H	P/V	N	CY	
M1		CP BC, mn	4	BC<mn	Φ	Φ	Φ	Φ	Φ	Φ	Φ	レジスタBCの値と16ビットの値mnを比較し、結果をもとにフラグを設定
M2		CP DE, mn	4	DE<mn	Φ	Φ	Φ	Φ	Φ	Φ	Φ	レジスタDEの値と16ビットの値mnを比較し、結果をもとにフラグを設定
M3		CP HL, mn	4	HL<mn	Φ	Φ	Φ	Φ	Φ	Φ	Φ	レジスタHLの値と16ビットの値mnを比較し、結果をもとにフラグを設定

【図 3 5 2】

<OPQ_r (k) (命令No.L7): 8ビット算術命令・比較>

No	ニーモニック	バイト数	オペレーション	フラグ							説明
				S	Z	TZ	H	P/V	N	CY	
L7a	OPQ B, (k)	3	B<(Qk)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタBの内容とレジスタQと8ビットデータk以外の内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L7b	OPQ C, (k)	3	C<(Qk)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタCの内容とレジスタQと8ビットデータk以外の内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L7c	OPQ D, (k)	3	D<(Qk)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタDの内容とレジスタQと8ビットデータk以外の内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L7d	OPQ E, (k)	3	E<(Qk)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタEの内容とレジスタQと8ビットデータk以外の内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L7e	OPQ H, (k)	3	H<(Qk)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタHの内容とレジスタQと8ビットデータk以外の内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	
L7f	OPQ L, (k)	3	L<(Qk)	Φ	Φ	Φ	Φ	Φ	Φ	レジスタLの内容とレジスタQと8ビットデータk以外の内容で指定される4ビットの内容を比較し、結果をもとにフラグを設定	

【図 3 5 4】

<8ビット算術命令・加算・減算>

No	新	ニーモニック	バイト数	オペレーション	フラグ										説明
					S	Z	TZ	H	P/V	N	CY				
N1		INC r	1	r←r+1	Φ	Φ	Φ	Φ	Φ	0	・	レジスタrの内容を1加算			
N2		INCQ (k)	3	(Qk)←(Qk)+1	Φ	Φ	Φ	Φ	Φ	0	・	レジスタQと8ビットの整数kの内容で指定される4ビットの内容を1加算			
N3		DEC r	1	r←r-1	Φ	Φ	Φ	Φ	Φ	1	・	レジスタrの内容を1減算			
N4		DECQ (k)	3	(Qk)←(Qk)-1	Φ	Φ	Φ	Φ	Φ	1	・	レジスタQと8ビットの整数kの内容で指定される4ビットの内容を1減算			

10

20

30

40

50

【図 3 5 5】

<INC r (命令No.N1), DEC r (命令No.N3):8ビット算術命令・加算・減算>

No	ニーモニック	バイト 数	オペレーション	フラグ								説明
				S	Z	TZ	H	P/V	N	CY		
N1a	INC A	1	A←A+1	0	0	0	0	0	0	0	・	レジスタAの内容を+加算
N1b	INC B	1	B←B+1	0	0	0	0	0	0	0	・	レジスタBの内容を+加算
N1c	INC C	1	C←C+1	0	0	0	0	0	0	0	・	レジスタCの内容を+加算
N1d	INC D	1	D←D+1	0	0	0	0	0	0	0	・	レジスタDの内容を+加算
N1e	INC E	1	E←E+1	0	0	0	0	0	0	0	・	レジスタEの内容を+加算
N1f	INC H	1	H←H+1	0	0	0	0	0	0	0	・	レジスタHの内容を+加算
N1g	INC L	1	L←L+1	0	0	0	0	0	0	0	・	レジスタLの内容を+加算
N3a	DEC A	1	A←A-1	0	0	0	0	0	1	・	・	レジスタAの内容を-減算
N3b	DEC B	1	B←B-1	0	0	0	0	0	1	・	・	レジスタBの内容を-減算
N3c	DEC C	1	C←C-1	0	0	0	0	0	1	・	・	レジスタCの内容を-減算
N3d	DEC D	1	D←D-1	0	0	0	0	0	1	・	・	レジスタDの内容を-減算
N3e	DEC E	1	E←E-1	0	0	0	0	0	1	・	・	レジスタEの内容を-減算
N3f	DEC H	1	H←H-1	0	0	0	0	0	1	・	・	レジスタHの内容を-減算
N3g	DEC L	1	L←L-1	0	0	0	0	0	1	・	・	レジスタLの内容を-減算

【図 3 5 6】

<16ビット算術命令・加算・減算>

No	新	ニーモニック	バイト数	オペレーション	フラグ								説明
					S	Z	TZ	H	P/V	N	CY		
O1		INC BC	1	BC←BC+1	・	・	・	・	・	・	・	・	レジスタBCの内容を1加算
O2		INC DE	1	DE←DE+1	・	・	・	・	・	・	・	・	レジスタDEの内容を1加算
O3		INC HL	1	HL←HL+1	・	・	・	・	・	・	・	・	レジスタHLの内容を1加算
O4		DEC BC	1	BC←BC-1	・	・	・	・	・	・	・	・	レジスタBCの内容を1減算
O5		DEC DE	1	DE←DE-1	・	・	・	・	・	・	・	・	レジスタDEの内容を1減算
O6		DEC HL	1	HL←HL-1	・	・	・	・	・	・	・	・	レジスタHLの内容を1減算
O7		ADDB HL, A	1	HL←HL+A	0	0	0	0	0	0	0	0	レジスタAのHLの内容とレジスタAの内容を加算し、結果をレジスタAに格納
O8		ADDB HL, r	2	HL←HL+r	0	0	0	0	0	0	0	0	レジスタAのHLの内容とレジスタrの内容を加算し、結果をレジスタAに格納
O9		ADDB BC, A	2	BC←BC+A	0	0	0	0	0	0	0	0	レジスタAのBCの内容とレジスタAの内容を加算し、結果をレジスタAに格納
O10		ADDB DE, A	2	DE←DE+A	0	0	0	0	0	0	0	0	レジスタAのDEの内容とレジスタAの内容を加算し、結果をレジスタAに格納

【図 3 5 7】

<ADDB HL, r (命令No.O8):16ビット算術命令・加算>

No	ニーモニック	バイト数	オペレーション	フラグ								説明
				S	Z	TZ	H	P/V	N	CY		
O8a	ADDB HL, B	2	HL←HL+B	0	0	0	0	0	0	0	0	レジスタBの内容とレジスタHLの内容を加算し、結果をレジスタAへ7HLに格納
O8b	ADDB HL, C	2	HL←HL+C	0	0	0	0	0	0	0	0	レジスタCの内容とレジスタHLの内容を加算し、結果をレジスタAへ7HLに格納
O8c	ADDB HL, D	2	HL←HL+D	0	0	0	0	0	0	0	0	レジスタDの内容とレジスタHLの内容を加算し、結果をレジスタAへ7HLに格納
O8d	ADDB HL, E	2	HL←HL+E	0	0	0	0	0	0	0	0	レジスタEの内容とレジスタHLの内容を加算し、結果をレジスタAへ7HLに格納

【図 3 5 8】

A

プログラム1-A(第1, 2実施形態)					バイト数
ラベル	OP1	OP2~	; コメント		
	LD	HL, wAREA0	; wAREA0のアドレスをへアレジスタHLにセット		3
	LD	B, (HL)	; wAREA0の内容をレジスタBにセット		1
	LD	(HL), A	; wAREA0にレジスタAをセット		1
	LD	A, B	; レジスタAにレジスタBをセット		1
合計					6

B

プログラム1-B(第3実施形態)					バイト数
ラベル	OP1	OP2~	; コメント		
	LD	HL, wAREA0	; wAREA0のアドレスをへアレジスタHLにセット		3
	EX	A, (HL)	; wAREA0の内容とレジスタAを交換		2
合計					5

10

20

30

40

50

【図 3 5 9】

A

プログラム2ーA(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	LD	IX, wAREA0	; wAREA0のアドレスをインデックスレジスタIXにセット	4
	LD	A, (IX)	; wAREA0の内容をレジスタAにセット	2
	LD	(IX), B	; wAREA0にレジスタBをセット	2
	LD	B, A	; レジスタBにレジスタAをセット	1
合計				9

B

プログラム2ーB(第3実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	LD	IX, wAREA0	; wAREA0のアドレスをインデックスレジスタIXにセット	4
	EX	B, (IX)	; wAREA0の内容とレジスタBを交換	2
合計				6

【図 3 6 0】

A

プログラム3ーA(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	LDQ	C, (wAREA0)	; wAREA0の内容をレジスタCにセット	3
	LDQ	(wAREA0), A	; wAREA0にレジスタAをセット	3
	LD	A, C	; レジスタAにレジスタCをセット	1
合計				7

B

プログラム3ーB(第3実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	EXQ	A, (wAREA0)	; wAREA0の内容とレジスタAを交換	3
合計				3

【図 3 6 1】

A

プログラム4ーA(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	JR	TZ, LABEL1	; TZ = 1 ならLABEL1にジャンプ	2
	LD	A, 1	; レジスタAに1をセット	2
	JR	LABEL2	; LABEL2にジャンプ	2
LABEL1:	LD	A, 0	; レジスタAに0をセット	2
LABEL2:	LD	(wAREA1), A	; wAREA1にレジスタAを格納	3
合計				11

B

プログラム4ーB(第3実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	LDZ	(wAREA1)	; TZ=1ならwAREA1に0をセットし、 ; TZ=0ならwAREA1に1をセット	4
合計				4

【図 3 6 2】

A

プログラム5ーA(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	LD	HL, wAREA1	; へレジスタHLにwAREA1のアドレスをセット	3
	LD	A, (wAREA0)	; wAREA0の内容をレジスタAにセット	3
	JR	TZ, LABEL1	; TZ = 1 ならLABEL1にジャンプ	2
	LD	A, 1	; レジスタAに1をセット	2
	JR	LABEL2	; LABEL2にジャンプ	2
LABEL1:	LD	A, 0	; レジスタAに0をセット	2
LABEL2:	LD	(wAREA1), A	; wAREA1にレジスタAを格納	3
合計				17

B

プログラム5ーB(第3実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	LD	HL, wAREA1	; へレジスタHLにwAREA1のアドレスをセット	3
	LD	A, (wAREA0)	; wAREA0の内容をレジスタAにセット	3
	LDZ	(HL)	; TZ=1ならwAREA1に0をセットし、 ; TZ=0ならwAREA1に1をセット	2
合計				8

10

20

30

40

50

【図 3 6 3】

プログラム6-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	JR	TZ, LABEL1	; TZ = 1 ならLABEL1にジャンプ	2
	LD	A, 1	; レジスタAに1をセット	2
	JR	LABEL2	; LABEL2にジャンプ	2
LABEL1:	LD	A, 0	; レジスタAに0をセット	2
LABEL2:	LDQ	(wAREA1), A	; wAREA1にレジスタAを格納	2
合計				10

プログラム6-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LDQZ	(wAREA1)	; TZ=1ならwAREA1に0をセットし, ; TZ=0ならwAREA1に1をセット	3
合計				3

【図 3 6 4】

プログラム7-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LD	A, (wAREA2)	; wAREA2の内容をレジスタAにセット	2
	CP	10H	; レジスタAと値を比較	2
	JR	C, LABEL3	; CY=1ならLABEL3にジャンプ	2
	LD	A, 0	; レジスタAに0をセット	2
	JR	LABEL4	; LABEL4にジャンプ	2
LABEL3:	LD	A, 1	; レジスタAに1をセット	2
LABEL4:	LD	(wAREA2), A	; wAREA2にレジスタAをセット	1
合計				13

プログラム7-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LD	A, (wAREA2)	; wAREA2の内容をレジスタAにセット	2
	CP	10H	; レジスタAと値を比較	2
	LDC	(wAREA2)	; CY=1の場合, wAREA2に1をセットし, ; CY=0の場合, wAREA2に0をセット	2
合計				6

10

20

【図 3 6 5】

プログラム8-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LD	HL, wAREA2	; wレジスタHLにwAREA2のアドレスをセット	3
	CP	(HL)	; レジスタAとwAREA2の内容を比較	1
	JR	C, LABEL3	; CY=1ならLABEL3にジャンプ	2
	LD	A, 0	; レジスタAに0をセット	2
	JR	LABEL4	; LABEL4にジャンプ	2
LABEL3:	LD	A, 1	; レジスタAに1をセット	2
LABEL4:	LD	(HL), A	; wAREA2にレジスタAをセット	1
合計				13

プログラム8-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LD	HL, wAREA2	; wレジスタHLにwAREA2のアドレスをセット	3
	CP	(HL)	; レジスタAとwAREA2の内容を比較	1
	LDC	(HL)	; CY=1の場合, wAREA2に1をセットし, ; CY=0の場合, wAREA2に0をセット	2
合計				6

【図 3 6 6】

プログラム9-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	CPQ	(wAREA2)	; レジスタAとwAREA2の内容を比較	3
	JR	C, LABEL3	; CY=1ならLABEL3にジャンプ	2
	LD	A, 0	; レジスタAに0をセット	2
	JR	LABEL4	; LABEL4にジャンプ	2
LABEL3:	LD	A, 1	; レジスタAに1をセット	2
LABEL4:	LDQ	(wAREA2), A	; wAREA2にレジスタAをセット	2
合計				13

プログラム9-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	OPQ	(wAREA2)	; レジスタAとwAREA2の内容を比較	3
	LDQC	(wAREA2)	; CY=1の場合, wAREA2に1をセットし, ; CY=0の場合, wAREA2に0をセット	3
合計				6

30

40

50

【図 3 6 7】

A

プログラム10-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	JR	NC, LABEL1	; CY=0ならLABEL1にジャンプ	2
LABEL1:	LD	A, 10	; レジスタAに10をセット	2
				合計 4

B

プログラム10-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LDF	C, A, 10	; CY=1ならレジスタAに10をセット	3
				合計 3

【図 3 6 8】

A

プログラム11-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	JR	NZ, LABEL1	; Z=0ならLABEL1にジャンプ	2
LABEL1:	LD	BC, 1000	; ヘアレジスタBCに1000をセット	3
				合計 5

B

プログラム11-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LDF	Z, BC, 1000	; Z=1ならヘアレジスタBCに1000をセット	4
				合計 4

10

【図 3 6 9】

A

プログラム12-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LDQ	A, (wAREA4)	; レジスタAにwAREA4の内容をセット	2
	LD	HL, TBL_SUB	; ジャンプ先テーブルの先頭アドレスを取得	3
	ADDWB	HL, A	; アドレスを加算	1
	LD	A, (HL)	; 算出後のアドレスのデータ取得	1
	OR	A	; CYをクリア	1
	ADDWB	HL, A	; 絶対アドレスを生成	1
	JP	(HL)	; wAREA4で指定されたサブルーチンにジャンプ	1
TBL_SUB:	DB	P_SUB_10 - \$	; 1: 第1サブルーチンまでの相対バイト	1
	DB	P_SUB_20 - \$	; 2: 第2サブルーチンまでの相対バイト	1
	DB	P_SUB_30 - \$	; 3: 第3サブルーチンまでの相対バイト	1
	DB	P_SUB_40 - \$	; 4: 第4サブルーチンまでの相対バイト	1
				合計 14

B

プログラム12-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LDQ	A, (wAREA4)	; レジスタAにwAREA4の内容をセット	2
	TJP		; wAREA4で指定されたサブルーチンにジャンプ	1
TBL_SUB:	DB	.LOW.P_SUB_10	; 1: 第1サブルーチンの下位アドレス	1
	DB	.LOW.P_SUB_20	; 2: 第2サブルーチンの下位アドレス	1
	DB	.LOW.P_SUB_30	; 3: 第3サブルーチンの下位アドレス	1
	DB	.LOW.P_SUB_40	; 4: 第4サブルーチンの下位アドレス	1
				合計 7

A

プログラム13-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LDQ	A, (wAREA4)	; レジスタAにwAREA4の内容をセット	2
	LD	HL, TBL_SUB	; ジャンプ先テーブルの先頭アドレスを取得	3
	ADDWB	HL, A	; アドレスを加算	1
	JP	(HL)	; wAREA4で指定されたサブルーチンにジャンプ	1
TBL_SUB:	DW	P_SUB_10	; 1: 第1サブルーチン	2
	DW	P_SUB_20	; 2: 第2サブルーチン	2
	DW	P_SUB_30	; 3: 第3サブルーチン	2
	DW	P_SUB_40	; 4: 第4サブルーチン	2
				合計 15

B

プログラム13-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	; コメント	
	LDQ	A, (wAREA4)	; レジスタAにwAREA4の内容をセット	2
	TJPW		; wAREA4で指定されたサブルーチンにジャンプ	1
TBL_SUB:	DW	P_SUB_10	; 1: 第1サブルーチン	2
	DW	P_SUB_20	; 2: 第2サブルーチン	2
	DW	P_SUB_30	; 3: 第3サブルーチン	2
	DW	P_SUB_40	; 4: 第4サブルーチン	2
				合計 11

20

30

40

50

【図 3 7 1】

プログラム14-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	コメント	
	LDQ	B, (wAREA4)	: レジスタBにwAREA4の内容をセット	3
	LD	HL, TBL_SUB	: ジャンプ先テーブルの先頭アドレスを取得	3
	ADDWB	HL, B	: アドレスを加算	1
	LD	A, (HL)	: 算出後のアドレスのデータ取得	1
	OR	A	: OYをクリア	1
	ADDWB	HL, A	: 絶対アドレスを生成	1
	JP	(HL)	: wAREA4で指定されたサブルーチンにジャンプ	1
TBL_SUB:	DB	P_SUB_10 - \$	: 1: 第1サブルーチンまでの相対バイト	1
	DB	P_SUB_20 - \$	: 2: 第2サブルーチンまでの相対バイト	1
	DB	P_SUB_30 - \$	: 3: 第3サブルーチンまでの相対バイト	1
	DB	P_SUB_40 - \$	: 4: 第4サブルーチンまでの相対バイト	1
合計				15

プログラム14-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	コメント	
	LDQ	B, (wAREA4)	: レジスタBにwAREA4の内容をセット	3
	TJP	B	: wAREA4で指定されたサブルーチンにジャンプ	2
TBL_SUB:	DB	LOW.P_SUB_10	: 1: 第1サブルーチンの下位アドレス	1
	DB	LOW.P_SUB_20	: 2: 第3サブルーチンの下位アドレス	1
	DB	LOW.P_SUB_30	: 3: 第3サブルーチンの下位アドレス	1
	DB	LOW.P_SUB_40	: 4: 第4サブルーチンの下位アドレス	1
合計				9

【図 3 7 2】

プログラム15-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	コメント	
	LDQ	D, (wAREA4)	: レジスタDにwAREA4の内容をセット	3
	LD	HL, TBL_SUB	: ジャンプ先テーブルの先頭アドレスを取得	3
	ADDWB	HL, D	: アドレスを加算	1
	JP	(HL)	: wAREA4で指定されたサブルーチンにジャンプ	1
TBL_SUB:	DW	P_SUB_10	: 1: 第1サブルーチン	2
	DW	P_SUB_20	: 2: 第2サブルーチン	2
	DW	P_SUB_30	: 3: 第3サブルーチン	2
	DW	P_SUB_40	: 4: 第4サブルーチン	2
合計				16

プログラム15-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	コメント	
	LDQ	D, (wAREA4)	: レジスタDにwAREA4の内容をセット	3
	TJPW	D	: wAREA4で指定されたサブルーチンにジャンプ	2
TBL_SUB:	DW	P_SUB_10	: 1: 第1サブルーチン	2
	DW	P_SUB_20	: 2: 第2サブルーチン	2
	DW	P_SUB_30	: 3: 第3サブルーチン	2
	DW	P_SUB_40	: 4: 第4サブルーチン	2
合計				13

10

20

【図 3 7 3】

プログラム16-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	コメント	
	LDQ	A, (wAREA5)	: wAREA5の内容をレジスタAにセット	2
	INC	A	: レジスタAを1加算	1
	CP	10	: レジスタAと10を比較	2
	JR	C, LABEL1	: CY=1ならジャンプ	2
	LD	A, 10	: レジスタAに10をセット	2
LABEL1:	LDQ	(wAREA5), A	: wAREA5にレジスタAをセット	2
合計				11

プログラム16-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	コメント	
	LDQ	A, (wAREA5)	: wAREA5の内容をレジスタAにセット	2
	ICPLL	A, 10	: 命令コード一覧表参照	3
	LDQ	(wAREA5), A	: wAREA5にレジスタAをセット	2
合計				7

【図 3 7 4】

プログラム17-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2~	コメント	
	LDQ	BC, (wAREA5)	: wAREA5+0+1の内容をへアレジスタBCにセット	3
	INC	BC	: へアレジスタBCを1加算	1
	CP	BC, 1000	: へアレジスタBCと1000を比較	4
	JR	C, LABEL1	: CY=1ならジャンプ	2
	LD	BC, 1000	: へアレジスタBCに1000をセット	3
LABEL1:	LDQ	(wAREA5), BC	: wAREA5+0+1にへアレジスタBCをセット	3
合計				16

プログラム17-B(第3実施形態)				バイト数
ラベル	OP1	OP2~	コメント	
	LDQ	BC, (wAREA5)	: wAREA5+0+1の内容をへアレジスタBCにセット	3
	ICPWL	BC, 1000	: 命令コード一覧表参照	4
	LDQ	(wAREA5), BC	: wAREA5+0+1にへアレジスタBCをセット	3
合計				10

30

40

50

【図 3 7 5】

A				
プログラム18-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	LDQ	A, (wAREA5)	: wAREA5の内容をレジスタAにセット	2
	INC	A	: レジスタAを1加算	1
	CP	C	: レジスタAとレジスタCを比較	1
	JR	C, LABEL1	: CY=1ならLABEL1にジャンプ	2
	LD	A, C	: レジスタAにレジスタCをセット	1
LABEL1:	LDQ	(wAREA5), A	: wAREA5にレジスタAをセット	2
合計				9

B				
プログラム18-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	LDQ	A, (wAREA5)	: wAREA5の内容をレジスタAにセット	2
	DCPLL	A, C	: 命令コード一覧表参照	3
	LDQ	(wAREA5), A	: wAREA5にレジスタAをセット	2
合計				7

【図 3 7 6】

A				
プログラム19-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	LDQ	A, (wAREA5)	: wAREA5の内容をレジスタAにセット	2
	DEC	A	: レジスタAを1減算	1
	CP	10	: レジスタAと10を比較	2
	JR	NC, LABEL1	: CY=0ならLABEL1にジャンプ	2
	LD	A, 10	: レジスタAに10をセット	2
LABEL1:	LDQ	(wAREA5), A	: wAREA5にレジスタAをセット	2
合計				11

B				
プログラム19-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	LDQ	A, (wAREA5)	: wAREA5の内容をレジスタAにセット	2
	DCPLL	A, 10	: 命令コード一覧表参照	3
	LDQ	(wAREA5), A	: wAREA5にレジスタAをセット	2
合計				7

10

【図 3 7 7】

A				
プログラム20-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	LDQ	BC, (wAREA5)	: wAREA5+0+1の内容をwレジスタBCにセット	3
	DEC	BC	: wレジスタBCを1減算	1
	CP	BC, 1000	: wレジスタBCと1000を比較	4
	JR	NC, LABEL1	: CY=0ならLABEL1にジャンプ	2
	LD	BC, 1000	: wレジスタBCに1000をセット	3
LABEL1:	LDQ	(wAREA5), BC	: wAREA5+0+1にwレジスタBCをセット	3
合計				16

B				
プログラム20-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	LDQ	BC, (wAREA5)	: wAREA5+0+1の内容をwレジスタBCにセット	3
	DCPWLL	BC, 1000	: 命令コード一覧表参照	4
	LDQ	(wAREA5), BC	: wAREA5+0+1にwレジスタBCをセット	3
合計				10

【図 3 7 8】

A				
プログラム21-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	DECQ	(wAREA5)	: wAREA5の内容を1減算	3
	LDQ	A, (wAREA5)	: wAREA5の内容をレジスタAにセット	2
	CP	C	: レジスタAとレジスタCを比較	1
	JR	NC, LABEL1	: CY=0ならLABEL1にジャンプ	2
	LDQ	(wAREA5), C	: wAREA5にレジスタCをセット	2
LABEL1:				
合計				10

B				
プログラム21-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	LDQ	A, (wAREA5)	: wAREA5の内容をレジスタAにセット	2
	DCPLL	A, C	: 命令コード一覧表参照	2
	LDQ	(wAREA5), A	: wAREA5にレジスタAをセット	2
合計				6

20

30

40

50

【図 3 7 9】

プログラム22-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	LD	B, A	; 元のレジスタAをレジスタBに退避	1
	AND	11111000B	; ビット0～ビット2をクリア	2
	SLA	A	; 左に1ビット分シフト	2
	LD	C, A	; シフト結果をレジスタCに退避	1
	LD	A, B	; 元の値を復帰	1
	AND	00000111B	; シフトしたビットをクリア	2
	OR	C	; ビット7からビット3のシフト結果とシフトされない ; ビット0からビット2を合成	1
合計				10

プログラム22-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	SLBIT	A, 3	; レジスタAのビット3～ビット7を左に1ビット分シフト	3
合計				3

【図 3 8 0】

プログラム23-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	LD	B, A	; 元のレジスタAをレジスタBに退避	1
	AND	00001111B	; ビット7～ビット4をクリア	2
	SRA	A	; 右に1ビット分シフト	2
	LD	C, A	; シフト結果をレジスタCに退避	1
	LD	A, B	; 元の値を復帰	1
	AND	11110000B	; シフトしたビットをクリア	2
	OR	C	; ビット3からビット0のシフト結果とシフトされない ; ビット7からビット4を合成	1
合計				10

プログラム23-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	SRBIT	A, 3	; レジスタAのビット3～ビット0を右に1ビット分シフト	3
合計				3

【図 3 8 1】

プログラム24-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	DI		; 割込み禁止	1
	CALL	SUB_PROC_1	; サブルーチンへ	3
合計				4

プログラム24-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	CALLD	SUB_PROC_1	; 割込み禁止した後、サブルーチンへ	3
合計				3

【図 3 8 2】

プログラム25-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	DI		; 割込み禁止	1
	CALLF	SUB_PROC_1	; サブルーチンへ	2
合計				3

プログラム25-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	; コメント	
	CALLFD	SUB_PROC_1	; 割込み禁止した後、サブルーチンへ	2
合計				2

10

20

30

40

50

【図 3 8 3】

A

プログラム26-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	EI		: 割込み許可	1
	CALL	SUB_PROC.1	: サブルーチンへ	3
合計				4

B

プログラム26-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	CALLE	SUB_PROC.1	: 割込み許可した後、サブルーチンへ	3
合計				3

【図 3 8 4】

A

プログラム27-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	EI		: 割込み許可	1
	CALLF	SUB_PROC.1	: サブルーチンへ	2
合計				3

B

プログラム27-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	CALLFE	SUB_PROC.1	: 割込み許可した後、サブルーチンへ	2
合計				2

10

【図 3 8 5】

A

プログラム28-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	DI		: 割込み禁止	1
	RET		: 戻る	1
合計				2

B

プログラム28-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	RETD		: 割込み禁止した後、戻る	1
合計				1

【図 3 8 6】

A

プログラム29-A(第1, 2実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	EI		: 割込み許可	1
	RET		: 戻る	1
合計				2

B

プログラム29-B(第3実施形態)				バイト数
ラベル	OP1	OP2～	: コメント	
	RETE		: 割込み許可した後、戻る	1
合計				1

20

30

40

50

フロントページの続き

- (56)参考文献 特許第6947806(JP, B2)
特開2019-072136(JP, A)
特開2017-035333(JP, A)
- (58)調査した分野(Int.Cl., DB名)
A63F 5/04