

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)公開番号  
特開2022-38544  
(P2022-38544A)

(43)公開日 令和4年3月10日(2022.3.10)

(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

テーマコード (参考)  
2 C 1 8 2

審査請求 未請求 請求項の数 1 O L (全31頁)

|          |                             |          |                                          |
|----------|-----------------------------|----------|------------------------------------------|
| (21)出願番号 | 特願2020-143115(P2020-143115) | (71)出願人  | 000144153<br>株式会社三共<br>東京都渋谷区渋谷三丁目29番14号 |
| (22)出願日  | 令和2年8月27日(2020.8.27)        | (72)発明者  | 小倉 敏男<br>東京都渋谷区渋谷三丁目29番14号<br>株式会社三共内    |
|          |                             | Fターム(参考) | 2C182 DA23                               |

(54)【発明の名称】 遊技機

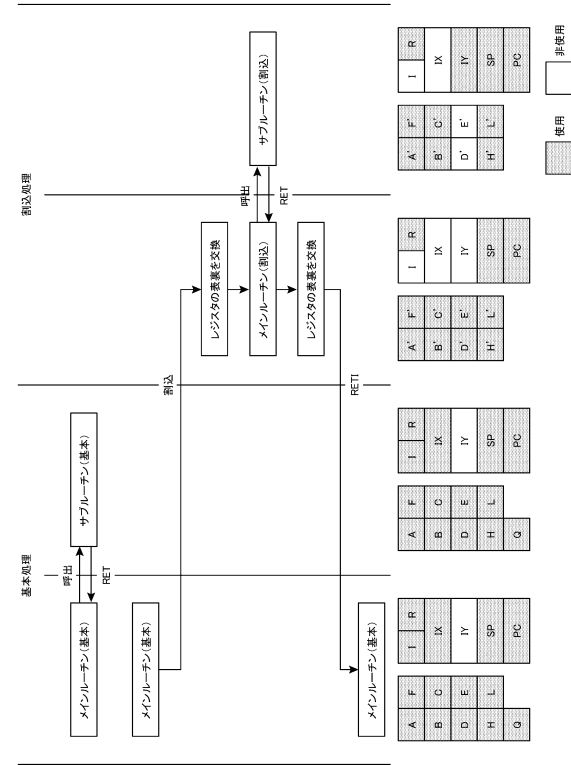
(57)【要約】

【課題】実行されるプログラムが切り替わる制御において改良を施した遊技機を提供することを目的とする。

【解決手段】割込処理は、メインルーチン（割込）と、メインルーチン（割込）から呼び出されて実行されるサブルーチン（割込）と、から構成されるとともに、割込処理において使用される複数のレジスタのうちI Yレジスタは、サブルーチン（割込）で使用され、メインルーチン（割込）で使用されない。

【選択図】図6

【図6】



**【特許請求の範囲】****【請求項 1】**

遊技を行う遊技機において、  
複数のレジスタを使用してプログラムを実行するプログラム実行手段を備え、  
前記プログラムのうち特定プログラムは、メインプログラムと、前記メインプログラムから呼び出されて実行されるサブプログラムと、から構成され、  
前記複数のレジスタのうち所定レジスタは、前記サブプログラムで使用され、前記メインプログラムで使用されない、遊技機。

**【発明の詳細な説明】****【技術分野】**

10

**【0001】**

本発明は、パチンコ遊技機やスロットマシン等の遊技機に関する。

**【背景技術】****【0002】**

この種の遊技機としては、一単位の遊技毎に繰り返し実行される基本処理と、当該基本処理に一定時間間隔毎に割り込んで処理を行う割込処理を実行可能なものが提案されている（例えば、特許文献 1 参照）。

**【先行技術文献】****【特許文献】****【0003】**

20

【特許文献 1】特開 2020 - 39538 号公報

**【発明の概要】****【発明が解決しようとする課題】****【0004】**

特許文献 1 に記載された遊技機のように、実行されるプログラムが切り替わる制御においてデータ混同の観点から改良の余地がある。

**【0005】**

本発明は、実行されるプログラムが切り替わる制御において改良を施した遊技機を提供することを目的とする。

**【課題を解決するための手段】**

30

**【0006】**

請求項 1 の遊技機は、  
遊技を行う遊技機において、  
複数のレジスタを使用してプログラムを実行するプログラム実行手段を備え、  
前記プログラムのうち特定プログラムは、メインプログラムと、前記メインプログラムから呼び出されて実行されるサブプログラムと、から構成され、  
前記複数のレジスタのうち所定レジスタは、前記サブプログラムで使用され、前記メインプログラムで使用されない  
ことを特徴としている。

この特徴によれば、特定プログラムは、メインプログラムと、メインプログラムから呼び出されて実行されるサブプログラムと、から構成されるとともに、複数のレジスタのうち所定レジスタは、サブプログラムで使用され、メインプログラムで使用されないため、メインプログラムとサブプログラムの間で所定レジスタに格納されるデータが混同してしまうことを防止できる。

40

**【0007】**

尚、本発明は、本発明の請求項に記載された発明特定事項のみを有するものであって良いし、本発明の請求項に記載された発明特定事項とともに該発明特定事項以外の構成を有するものであって良い。

**【図面の簡単な説明】****【0008】**

50

【図 1】本発明が適用された実施例のスロットマシンの正面図である。

【図 2】スロットマシンの構成を示すブロック図である。

【図 3】メイン制御部の ROM 及び RAM のメモリマップを示す図である。

【図 4】容量内領域と容量外領域との関係を示す図である。

【図 5】レジスタについて説明するための図である。

【図 6】各種処理と各レジスタとの対応関係について説明するための図である。

【図 7】基本処理と各レジスタとの対応関係について説明するための図である。

【図 8】割込処理と各レジスタとの対応関係について説明するための図である。

【図 9】レジスタの変形例について説明するための図である。

【図 10】レジスタの変形例における容量内処理及び容量外処理と共通レジスタとの対応関係について説明するための図である。 10

【発明を実施するための形態】

【0009】

本発明に係る遊技機を実施するための形態を実施例に基づいて以下に説明する。

【0010】

[形態 1]

形態 1 - 1 の遊技機は、

遊技を行う遊技機（スロットマシン 1）において、

複数のレジスタを使用してプログラムを実行するプログラム実行手段（メイン制御部 41）を備え、 20

前記プログラムのうち特定プログラム（割込処理）は、メインプログラム（メインルーチン（割込））と、前記メインプログラムから呼び出されて実行されるサブプログラム（サブルーチン（割込））と、から構成され、

前記複数のレジスタのうち所定レジスタ（IY レジスタ）は、前記サブプログラム（サブルーチン（割込））で使用され、前記メインプログラム（メインルーチン（割込））で使用されない

ことを特徴としている。

この特徴によれば、特定プログラムは、メインプログラムと、メインプログラムから呼び出されて実行されるサブプログラムと、から構成されるとともに、複数のレジスタのうち所定レジスタは、サブプログラムで使用され、メインプログラムで使用されないため、メインプログラムとサブプログラムの間で所定レジスタに格納されるデータが混同してしまうことを防止できる。 30

【0011】

形態 1 - 2 の遊技機は、形態 1 - 1 に記載の遊技機であって、

前記サブプログラム（サブルーチン（割込））は、複数種類のサブプログラムを含み、

前記所定レジスタ（IY レジスタ）は、前記複数種類のサブプログラムのうちいずれかの種類のサブプログラムで使用される

ことを特徴としている。

この特徴によれば、メインプログラムと複数種類のサブプログラムのうち所定レジスタが使用されるサブプログラムとの間で所定レジスタに格納されるデータが混同してしまうことを防止できる。 40

【0012】

形態 1 - 3 の遊技機は、

遊技を行う遊技機（スロットマシン 1）において、

複数のレジスタを使用してプログラムを実行するプログラム実行手段（メイン制御部 41）を備え、

前記プログラムのうち特定プログラム（割込処理）は、メインプログラム（メインルーチン（割込））と、前記メインプログラムから呼び出されて実行されるサブプログラム（サブルーチン（割込））と、から構成され、

前記複数のレジスタのうち特定レジスタ（D'E レジスタ）は、前記メインプログラム（ 50

メインルーチン（割込））で使用され、前記サブプログラム（サブルーチン（割込））で使用されない

ことを特徴としている。

この特徴によれば、特定プログラムは、メインプログラムと、メインプログラムから呼び出されて実行されるサブプログラムと、から構成されるとともに、複数のレジスタのうち特定レジスタは、メインプログラムで使用され、サブプログラムで使用されないため、メインプログラムとサブプログラムの間で特定レジスタに格納されるデータが混同してしまうことを防止できる。

#### 【 0 0 1 3 】

形態 1 - 4 の遊技機は、形態 1 - 3 に記載の遊技機であって、

10

前記サブプログラム（サブルーチン（割込））は、複数種類のサブプログラムを含み、前記特定レジスタ（D' E' レジスタ）は、前記複数種類のサブプログラムのうちいずれの種類のサブプログラムにおいても使用されない

ことを特徴としている。

この特徴によれば、メインプログラムといずれの種類のサブプログラムとの間においても特定レジスタに格納されるデータが混同してしまうことを防止できる。

#### 【 0 0 1 4 】

形態 1 - 5 の遊技機は、形態 1 - 1 ~ 1 - 4 のいずれかに記載の遊技機であって、

前記特定プログラムは、割込の発生により実行する割込プログラム（割込処理）であることを特徴としている。

20

この特徴によれば、割込プログラムの実行中にメインプログラムとサブプログラムとの間でレジスタに格納されるデータが混同してしまうことを防止できる。

#### 【 0 0 1 5 】

##### [ 形態 2 ]

形態 2 - 1 の遊技機は、

遊技を行う遊技機（スロットマシン 1）において、

複数のレジスタを使用してプログラムを実行するプログラム実行手段（メイン制御部 4 1）を備え、

前記プログラムは、基本プログラム（基本処理）と、前記基本プログラムの実行中に割り込んで実行される割込プログラム（割込処理）と、を含み、

30

前記基本プログラム（基本処理）を実行するときに第 1 レジスタ（表レジスタ）を使用し、

前記割込プログラム（割込処理）を実行するときに使用するレジスタを前記第 1 レジスタ（表レジスタ）から第 2 レジスタ（裏レジスタ）に切り替え、

前記割込プログラム（割込処理）を終了するときに使用するレジスタを前記第 2 レジスタ（裏レジスタ）から前記第 1 レジスタ（表レジスタ）に戻し、

前記複数のレジスタは、前記第 1 レジスタ（表レジスタ）及び前記第 2 レジスタ（裏レジスタ）とは別に前記基本プログラム（基本処理）を実行するときにも前記割込プログラム（割込処理）を実行するときにも使用可能な特別レジスタ（IX レジスタ）を含み、

前記特別レジスタ（IX レジスタ）は、前記基本プログラム（基本処理）で使用され、前記割込プログラム（割込処理）で使用されない

40

ことを特徴としている。

この特徴によれば、基本プログラムでは、第 1 レジスタを使用する一方、基本プログラムに割り込んで実行される割込プログラムの実行中は、使用するレジスタが第 1 レジスタから第 2 レジスタに切り替わるとともに、複数のレジスタは、第 1 レジスタ及び第 2 レジスタとは別に基本プログラムを実行するときにも割込プログラムを実行するときにも使用可能な特別レジスタを含むが、特別レジスタは、基本プログラムで使用され、割込プログラムでは使用されないため、基本プログラムと割込プログラムとの間において特別レジスタに格納されるデータが混同してしまうことを防止できる。

#### 【 0 0 1 6 】

50

形態 2 - 2 遊技機は、形態 2 - 1 に記載の遊技機であって、  
前記複数のレジスタは、前記第 1 レジスタ（表レジスタ）及び前記第 2 レジスタ（裏レジスタ）とは別に前記基本プログラム（基本処理）を実行するときにも前記割込プログラム（割込処理）を実行するときにも使用可能な複数の共用レジスタ（I X レジスタ、I Y レジスタ）を含み、

前記複数の共用レジスタ（I X レジスタ、I Y レジスタ）のうち前記特別レジスタ（I X レジスタ）を含むいずれの共用レジスタも、前記割込プログラム（割込処理）では使用されない

ことを特徴としている。

この特徴によれば、複数のレジスタは、第 1 レジスタ及び第 2 レジスタとは別に基本プログラムを実行するときにも割込プログラムを実行するときにも使用可能な複数の共用レジスタを含むが、複数の共用レジスタのうち特別レジスタを含むいずれの共用レジスタも、割込プログラムでは使用されないので、基本プログラムと割込プログラムとの間において特別レジスタを含む共用レジスタに格納されるデータが混同してしまうことを防止できる

。

#### 【 0 0 1 7 】

形態 2 - 3 遊技機は、形態 2 - 1 に記載の遊技機であって、

前記複数のレジスタは、前記第 1 レジスタ（表レジスタ）及び前記第 2 レジスタ（裏レジスタ）とは別に前記基本プログラム（基本処理）を実行するときにも前記割込プログラム（割込処理）を実行するときにも使用可能な複数の共用レジスタ（I X レジスタ、I Y レジスタ）を含み、

前記複数の共用レジスタ（I X レジスタ、I Y レジスタ）のうち少なくとも前記特別レジスタ（I X レジスタ）は、前記第 2 プログラム（割込処理）では使用されない

ことを特徴としている。

この特徴によれば、複数のレジスタは、第 1 レジスタ及び第 2 レジスタとは別に基本プログラムを実行するときにも割込プログラムを実行するときにも使用可能な複数の共用レジスタを含むが、複数の共用レジスタのうち少なくとも特別レジスタは、割込プログラムでは使用されないので、基本プログラムと割込プログラムとの間において少なくとも特別レジスタに格納されるデータが混同してしまうことを防止できる。

#### 【 0 0 1 8 】

形態 2 - 4 の遊技機は、

遊技を行う遊技機（スロットマシン 1）において、

複数のレジスタを使用してプログラムを実行するプログラム実行手段（メイン制御部 4 1）を備え、

前記プログラムは、基本プログラム（基本処理）と、前記基本プログラムの実行中に割り込んで実行される割込プログラム（割込処理）と、を含み、

前記基本プログラム（基本処理）を実行するときには第 1 レジスタ（表レジスタ）を使用し、

前記割込プログラム（割込処理）を実行するときには使用するレジスタを前記第 1 レジスタ（表レジスタ）から第 2 レジスタ（裏レジスタ）に切り替え、

前記割込プログラム（割込処理）を終了するときには使用するレジスタを前記第 2 レジスタ（裏レジスタ）から前記第 1 レジスタ（表レジスタ）に戻し、

前記複数のレジスタは、前記第 1 レジスタ（表レジスタ）及び前記第 2 レジスタ（裏レジスタ）とは別に前記基本プログラム（基本処理）を実行するときにも前記割込プログラム（割込処理）を実行するときにも使用可能な特殊レジスタ（I Y レジスタ）を含み、

前記特殊レジスタ（I Y レジスタ）は、前記割込プログラム（割込処理）で使用され、前記基本プログラム（基本処理）で使用されない

ことを特徴としている。

この特徴によれば、基本プログラムでは、第 1 レジスタを使用する一方、基本プログラムに割り込んで実行される割込プログラムの実行中は、使用するレジスタが第 1 レジスタか

10

20

30

40

50

ら第 2 レジスタに切り替わるとともに、複数のレジスタは、第 1 レジスタ及び第 2 レジスタとは別に基本プログラムを実行するときにも割込プログラムを実行するときにも使用可能な特殊レジスタを含むが、特殊レジスタは、割込プログラムで使用され、基本プログラムでは使用されないの、基本プログラムと割込プログラムとの間において特殊レジスタに格納されるデータが混同してしまうことを防止できる。

#### 【 0 0 1 9 】

形態 2 - 5 遊技機は、形態 2 - 4 に記載の遊技機であって、  
前記複数のレジスタは、前記第 1 レジスタ（表レジスタ）及び前記第 2 レジスタ（裏レジスタ）とは別に前記基本プログラム（基本処理）を実行するときにも前記割込プログラム（割込処理）を実行するときにも使用可能な複数の共用レジスタ（I X レジスタ、I Y レジスタ）を含み、

前記複数の共用レジスタ（I X レジスタ、I Y レジスタ）のうち前記特殊レジスタ（I Y レジスタ）を含むいずれの共用レジスタも、前記基本プログラム（基本処理）では使用されない

ことを特徴としている。

この特徴によれば、複数のレジスタは、第 1 レジスタ及び第 2 レジスタとは別に基本プログラムを実行するときにも割込プログラムを実行するときにも使用可能な複数の共用レジスタを含むが、複数の共用レジスタのうち特殊レジスタを含むいずれの共用レジスタも、基本プログラムでは使用されないの、基本プログラムと割込プログラムとの間において特殊レジスタを含む共用レジスタに格納されるデータが混同してしまうことを防止できる

#### 【 0 0 2 0 】

形態 2 - 6 遊技機は、形態 2 - 4 に記載の遊技機であって、

前記複数のレジスタは、前記第 1 レジスタ（表レジスタ）及び前記第 2 レジスタ（裏レジスタ）とは別に前記基本プログラム（基本処理）を実行するときにも前記割込プログラム（割込処理）を実行するときにも使用可能な複数の共用レジスタ（I X レジスタ、I Y レジスタ）を含み、

前記複数の共用レジスタ（I X レジスタ、I Y レジスタ）のうち少なくとも前記特殊レジスタ（I Y レジスタ）は、前記基本プログラム（基本処理）では使用されない

ことを特徴としている。

この特徴によれば、複数のレジスタは、第 1 レジスタ及び第 2 レジスタとは別に基本プログラムを実行するときにも割込プログラムを実行するときにも使用可能な複数の共用レジスタを含むが、複数の共用レジスタのうち少なくとも特殊レジスタは、基本プログラムでは使用されないの、基本プログラムと割込プログラムとの間において少なくとも特殊レジスタに格納されるデータが混同してしまうことを防止できる。

#### 【 0 0 2 1 】

形態 2 - 7 遊技機は、形態 2 - 1 ~ 2 - 6 のいずれかに記載の遊技機であって、

前記割込プログラム（割込処理）を実行するときに前記第 2 レジスタ（裏レジスタ）に値を設定する

ことを特徴としている。

この特徴によれば、割込プログラムを実行するときに、もともと第 2 レジスタに格納されていた値ではなく、改めて値を設定するので、意図しない値で割込プログラムが実行されてしまうことを防止できる。

#### 【 0 0 2 2 】

##### [ 形態 3 ]

形態 3 - 1 の遊技機は、

遊技を行う遊技機（スロットマシン 1）において、

複数のレジスタを使用してプログラムを実行するプログラム実行手段（メイン制御部 4 1）を備え、

前記プログラムは、第 1 プログラム（容量内プログラム（容量内処理））と、第 1 プログ

ラムから呼び出されて実行される第 2 プログラム（容量外プログラム（容量外処理））と、を含み、

前記第 2 プログラム（容量外プログラム（容量外処理））を実行するときに前記複数のレジスタのデータを退避させ、前記第 2 プログラム（容量外プログラム（容量外処理））を終了するときに、退避した前記複数のレジスタのデータを復帰させ、

前記第 1 プログラム（容量内プログラム（容量内処理））を実行するときは、前記複数のレジスタのうち特定数のレジスタが使用され、

前記第 2 プログラム（容量外プログラム（容量外処理））を実行するときには、前記複数のレジスタのうち前記特定数よりも少ない数のレジスタ（容量内プログラム（容量内処理））が用いるレジスタのうちの一部のレジスタ）が使用される

10

ことを特徴としている。

この特徴によれば、第 2 プログラムを実行するときに複数のレジスタのデータを退避させ、第 2 プログラムを終了するときに、退避した複数のレジスタのデータを復帰させるとともに、第 2 プログラムを実行するときには、第 1 プログラムを実行するときよりも使用されるレジスタの数が少ないので、第 2 プログラムによる誤ったデータの使用を防止できる。

#### 【 0 0 2 3 】

形態 3 - 2 の遊技機は、形態 3 - 1 に記載の遊技機であって、

前記第 1 プログラム（容量内プログラム（容量内処理））は、第 1 特定プログラム（容量内処理（基本処理））と、第 1 所定プログラム（容量内処理（割込処理））と、を含み、

20

前記第 2 プログラム（容量外プログラム（容量外処理））は、前記第 1 特定プログラム（容量内処理（基本処理））から呼び出されて実行される第 2 特定プログラム（容量外処理（基本処理））と、前記第 1 所定プログラム（容量内処理（割込処理））から呼び出されて実行される第 2 所定プログラム（容量外処理（割込処理））と、を含み、

前記第 1 特定プログラム（容量内処理（基本処理））を実行するときは、前記複数のレジスタのうち第 1 特定数のレジスタが使用され、

前記第 2 特定プログラム（容量外処理（基本処理））を実行するときには、前記複数のレジスタのうち前記第 1 特定数よりも少ない数のレジスタ（容量外処理（基本処理））が用いるレジスタのうちの一部のレジスタ）が使用され、

前記第 1 所定プログラム（容量内処理（割込処理））を実行するときは、前記複数のレジスタのうち第 2 特定数のレジスタが使用され、

30

前記第 2 所定プログラム（容量外処理（割込処理））を実行するときには、前記複数のレジスタのうち前記第 2 特定数よりも少ない数のレジスタ（容量外処理（割込処理））が用いるレジスタのうちの一部のレジスタ）が使用される

ことを特徴としている。

この特徴によれば、第 2 特定プログラムを実行するときには、第 1 特定プログラムを実行するときよりも使用されるレジスタの数が少なく、第 2 所定プログラムを実行するときには、第 1 所定プログラムを実行するときよりも使用されるレジスタの数が少ないので、第 2 特定プログラム及び第 2 所定プログラムによる誤ったデータの使用を防止できる。

#### 【 0 0 2 4 】

40

#### [ 形態 4 ]

形態 4 - 1 の遊技機は、

遊技を行う遊技機（スロットマシン 1）において、

プログラムを実行するプログラム実行手段（メイン制御部 4 1）と、

データを退避するスタック領域と、

を備え、

前記プログラムは、第 1 プログラム（容量内プログラム（容量内処理））と、第 1 プログラムから呼び出されて実行される第 2 プログラム（容量外プログラム（容量外処理））と、を含み、

前記スタック領域は、前記第 1 プログラム（容量内プログラム（容量内処理））によりデ

50

ータを退避する第 1 スタック領域（容量内スタック領域）と、前記第 2 プログラム（容量外プログラム（容量外処理））によりデータを退避する第 2 スタック領域（容量外スタック領域）と、を含み、

前記第 2 プログラム（容量外プログラム（容量外処理））を実行するときにデータの退避アドレスを示すスタックポインタを前記第 1 スタック領域（容量内スタック領域）から前記第 2 スタック領域（容量外スタック領域）に切り替え、前記第 2 プログラム（容量外プログラム（容量外処理））を終了するときに前記スタックポインタを前記第 1 スタック領域（容量内スタック領域）に戻し、

前記第 2 プログラム（容量外プログラム（容量外処理））を終了するときに、前記第 2 プログラムの実行中に前記第 2 スタック領域（容量外スタック領域）に退避したデータを全て復帰させる

10

ことを特徴としている。

この特徴によれば、第 2 プログラムを実行するときにスタックポインタを第 1 スタック領域から第 2 スタック領域に切り替え、第 2 プログラムを終了するときにスタックポインタを第 1 スタック領域に戻すとともに、第 2 プログラムを終了するときに、第 2 プログラムの実行中に第 2 スタック領域に退避したデータを全て復帰させてから、スタックポインタを第 1 スタック領域に戻すので、前回実行した第 2 プログラムで使用されていたデータが退避したまま、さらに第 2 プログラムが実行されることがなく、前回実行した第 2 プログラムで使用されていたデータと、今回実行する第 2 プログラムで使用されるデータと、が混同してしまうことを防止できる。

20

#### 【0025】

形態 4 - 2 の遊技機は、形態 4 - 1 に記載の遊技機であって、

前記第 2 プログラム（容量外プログラム（容量外処理））は、複数種類の第 2 プログラムを含み、

前記第 2 プログラムを終了するときに、いずれの種類の前記第 2 プログラムであっても、前記第 2 プログラムの実行中に前記第 2 スタック領域（容量外スタック領域）に退避したデータを全て復帰させる

ことを特徴としている。

この特徴によれば、第 2 プログラムを終了するときには、いずれの種類の第 2 プログラムであっても、第 2 プログラムの実行中に第 2 スタック領域に退避したデータを全て復帰させてから、スタックポインタを第 1 スタック領域に戻すので、第 2 プログラムの種類に関わらず、前回実行した第 2 プログラムで使用されていたデータと、今回実行する第 2 プログラムで使用されるデータと、が混同してしまうことを防止できる。

30

#### 【実施例】

#### 【0026】

本発明が適用されたスロットマシンの実施例について図面に基づいて説明する。本実施例のスロットマシン 1 は、図 1 に示すように、前面が開口する筐体 1 a と、この筐体 1 a の側端に回転自在に枢支された前面扉 1 b と、から構成されている。スロットマシン 1 の内部には、互いに識別可能な複数種類の図柄が所定の順序で、それぞれ同数ずつ配列されたリール 2 L、2 C、2 R が水平方向に並設されており、図 1 に示すように、これらリール 2 L、2 C、2 R に配列された図柄のうち連続する 3 つの図柄が、スロットマシン 1 の正面の略中央に設けられた透視窓 3 において各々上中下三段に表示されて遊技者側から見えるように配置されている。

40

#### 【0027】

尚、本実施例では、3 つのリールを用いた構成を例示しているが、リールを 1 つのみ用いた構成、2 つのリールを用いた構成、4 つ以上のリールを用いた構成としても良い。また、本実施例では、リール 2 L、2 C、2 R により図柄を可変表示させる可変表示部を構成しているが、可変表示部は、リール以外であっても良く、例えば、外周面に複数の図柄が配置されたベルトを移動させることで図柄を変動表示させることが可能な構成でも良い。また、本実施例では、物理的なリールにて可変表示部を構成しているが、液晶表示器など

50

の画像表示装置にて可変表示部を構成しても良い。

【0028】

スロットマシン1の正面には、図1に示すように、メダルを投入可能なメダル投入部4、メダルが払い出されるメダル払出口9、クレジット（遊技者所有の遊技用価値として記憶されているメダル数）を用いて、その範囲内において遊技状態に応じて定められた規定数の賭数のうち最大の賭数を設定する際に操作されるMAX BETスイッチ6、クレジットとして記憶されているメダル及び賭数の設定に用いたメダルを精算する（クレジット及び賭数の設定に用いた分のメダルを返却させる）際に操作される精算スイッチ10、ゲームを開始する際に操作されるスタートスイッチ7、リール2L、2C、2Rの回転を各々停止する際に操作されるストップスイッチ8L、8C、8R、演出に用いられる演出用スイッチ56が遊技者により操作可能にそれぞれ設けられている。

10

【0029】

スロットマシン1の正面には、図1に示すように、クレジットとして記憶されているメダル枚数が表示されるクレジット表示器11、入賞の発生により払い出されたメダル枚数やエラー発生時にその内容を示すエラーコードや、ストップスイッチ8L、8C、8Rの操作態様に対応する操作情報（ナビ報知）等が表示される遊技補助表示器12、賭数が1設定されている旨を点灯により報知する1 B E T L E D 14、賭数が2設定されている旨を点灯により報知する2 B E T L E D 15、賭数が3設定されている旨を点灯により報知する3 B E T L E D 16、スタートスイッチ7の操作によるゲームのスタート操作が有効である旨を点灯により報知するスタート有効L E D 18、遊技区間が有利区間に制御されている旨を点灯により報知する区間表示L E D 19、リプレイゲーム中である旨を点灯により報知するリプレイ中L E D 20、が設けられた遊技用表示部13が設けられている。

20

【0030】

MAX BETスイッチ6の内部には、MAX BETスイッチ6の操作による賭数の設定操作が有効である旨を点灯により報知するBETスイッチ有効L E D 21（図2参照）が設けられており、ストップスイッチ8L、8C、8Rの内部には、該当するストップスイッチ8L、8C、8Rによるリールを停止させる操作が有効である旨を点灯により報知する左、中、右停止有効L E D 22L、22C、22R（図2参照）がそれぞれ設けられている。

【0031】

また、スロットマシン1の正面には、画像を表示可能な液晶表示器51が設けられている。液晶表示器51は、液晶素子に対して電圧が印加されていない状態で透過性を有する液晶パネルを有しており、前面扉1bの各リール2L、2C、2Rの手前側（遊技者側）に表示領域が配置されるように設けられている。液晶表示器51の背面側の各リール2L、2C、2Rは、液晶表示器51の表示領域のうち透視窓3に対応する透過領域及び透視窓3を介して遊技者側から視認可能である。

30

【0032】

スロットマシン1の前面扉1bの内側には、所定キー操作によりスロットマシン1の外部からのエラー状態を解除するためのリセット操作を検出するリセットスイッチ23、設定値の変更中や設定値の確認中にその時点の設定値が表示される設定値表示器24、前面扉1bの開放状態を検出するドア開放検出スイッチ25、メダル投入部4から投入されたメダルの流路を、スロットマシン1の内部に設けられたホッパータンク側またはメダル払出口9側のいずれか一方に選択的に切り替えるための流路切替ソレノイド30、メダル投入部4から投入されてホッパータンク側に流下したメダルを検出する投入メダルセンサ31a～31cを有するメダルセクタ29が設けられている。

40

【0033】

また、スロットマシン1の内部には、メイン制御部41からの制御信号に応じて前述のリール2L、2C、2Rを回転させたり停止させたりするためのリールユニット34、メイン制御部41からの制御信号に応じてメダルをメダル払出口9より払い出すためのホッパーユニット35が設けられている。

50

## 【 0 0 3 4 】

また、スロットマシン 1 の内部には、設定変更状態または設定確認状態に切り替えるための設定キースイッチ 3 7、通常時においてはエラー状態を解除するためのリセットスイッチとして機能し、設定変更状態においては後述する内部抽選の当選確率（出玉率）の設定値を変更するための設定スイッチとして機能するリセット / 設定スイッチ 3 8、電源を on / off する際に操作される電源スイッチ 3 9 が設けられている。

## 【 0 0 3 5 】

スロットマシン 1 の内部には、遊技制御基板 4 0 が筐体 1 a の内部の所定位置に取り付けられた状態において、遊技制御基板 4 0 の正面側（遊技者側）の下部には 4 桁の 7 セグメント表示器で構成される遊技機情報表示器 5 0 が配置されている（図示略）。遊技機情報表示器 5 0 は、遊技制御基板 4 0 が基板ケースに封入されている状態で、当該基板ケースの外部から表示内容を視認可能となっており、メイン制御部 4 1 により集計されて遊技機情報表示器 5 0 に表示されるスロットマシン 1 における遊技の履歴に基づく遊技機情報を、店員等が認識できるようになっている。

10

## 【 0 0 3 6 】

図 2 に示すように、スロットマシン 1 には、遊技制御基板 4 0、演出制御基板 9 0 が設けられており、遊技制御基板 4 0 によって遊技（ゲーム）の制御が行われ、演出制御基板 9 0 によって遊技状態に応じた演出の制御が行われる。

## 【 0 0 3 7 】

遊技制御基板 4 0 には、前述の MAX BET スイッチ 6、スタートスイッチ 7、ストップスイッチ 8 L、8 C、8 R、精算スイッチ 1 0、リセットスイッチ 2 3、投入メダルセンサ 3 1 a ~ 3 1 c、リールユニット 3 4 のリールセンサ（図示略）、ホッパーユニット 3 5 の払出センサ及び満タンセンサ（図示略）、設定キースイッチ 3 7、リセット / 設定スイッチ 3 8 が接続されており、これら接続されたスイッチ類等の検出信号が入力されるようになっている。

20

## 【 0 0 3 8 】

また、遊技制御基板 4 0 には、前述のクレジット表示器 1 1、遊技補助表示器 1 2、1 ~ 3 B E T L E D 1 4 ~ 1 6、スタート有効 L E D 1 8、区間表示 L E D 1 9、リプレイ中 L E D 2 0、B E T スイッチ有効 L E D 2 1、左、中、右停止有効 L E D 2 2 L、2 2 C、2 2 R、設定値表示器 2 4、流路切替ソレノイド 3 0、リールユニット 3 4、ホッパーユニット 3 5 が接続されており、これら電気部品と、遊技制御基板 4 0 に搭載された遊技機情報表示器 5 0 は、遊技制御基板 4 0 に搭載された後述のメイン制御部 4 1 の制御に基づいて駆動されるようになっている。

30

## 【 0 0 3 9 】

また、遊技制御基板 4 0 には、遊技の制御を行うメイン制御部 4 1 が搭載されている。メイン制御部 4 1 は、演算を行うための C P U 4 1 a、プログラム等が記憶される R O M 4 1 b、ワークデータが一時的に記憶される R A M 4 1 c を備え、R O M 4 1 b に記憶されたプログラムに従って各種の制御を行う。

## 【 0 0 4 0 】

C P U 4 1 a は、R O M 4 1 b に記憶されたプログラムのうちプログラムカウンタが指定するアドレスの命令を実行する。プログラムカウンタは、命令が完了する毎に加算更新されるため、R O M 4 1 b に記憶されたプログラムのうち小さい値のアドレスに規定された命令から順に実行されることとなる。

40

## 【 0 0 4 1 】

メイン制御部 4 1 は、C P U 4 1 a が演算を行うために用いられる複数のレジスタを備える。当該複数のレジスタには、アキュムレータレジスタ（A レジスタ）、フラグレジスタ（F レジスタ）、汎用レジスタ（B レジスタ、C レジスタ、D レジスタ、E レジスタ、H レジスタ、L レジスタ）、インデックスレジスタ（I X レジスタ、I Y レジスタ）、インタラプトレジスタ（I レジスタ）、リフレッシュレジスタ（R レジスタ）、スタックポインタレジスタ（S P レジスタ）、プログラムカウンタレジスタ（P C レジスタ）、送信レ

50

ジスタなどのレジスタが含まれる。また、これらのレジスタのうちアキュムレータレジスタ、フラグレジスタ、汎用レジスタには、それぞれ対になるように構成された表レジスタ及び裏レジスタが含まれる（以下、表レジスタ及び裏レジスタを単にレジスタと呼ぶ場合がある）。CPU 41aは、プログラムに含まれる演算命令や読出命令等の各種の命令を実行することにより、所定のレジスタの値を更新することや、所定のレジスタの値（アドレス）により指定されるRAM 41cの記憶領域に記憶した値（データ）を更新することが可能である。

【0042】

また、メイン制御部41が備えるレジスタのうちフラグレジスタは、その状態が、CPU 41aにより実行された命令による演算結果を示すように変化するように構成されており、フラグレジスタの状態の変化を利用して、先の命令による演算結果に応じた処理をCPU 41aに行わせることができるようになっている。 10

【0043】

また、フラグレジスタは、複数のビット（本実施例では、8ビット）で構成されており、複数のビットには、CPU 41aの処理の負担を減らすために、CPU 41aが実行する演算命令の演算結果に応じて状態が変化可能に構成された第1ビット（以下、ゼロフラグと呼ぶ場合がある。）や第2ビット（以下、キャリーフラグと呼ぶ場合がある。）などが含まれる。

【0044】

また、メイン制御部41は、サブ制御部91に各種のコマンドを送信する。メイン制御部41からサブ制御部91へ送信されるコマンドは一方方向のみで送られ、サブ制御部91からメイン制御部41へ向けてコマンドが送られることはない。 20

【0045】

また、メイン制御部41は、遊技制御基板40に接続された各種スイッチ類の検出状態が変化するまでは制御状態に応じた処理を繰り返しループし、各種スイッチ類の検出状態の変化に応じて段階的に移行する基本処理を実行する。また、メイン制御部41は、タイマ割込が発生する毎（本実施例では、約0.56ミリ秒）毎に基本処理に割り込んで割込処理を実行する。尚、割込処理の実行間隔は、基本処理において制御状態に応じて繰り返す処理が一巡する時間と割込処理の実行時間とを合わせた時間よりも長い時間に設定されており、今回と次の割込処理との間で必ず制御状態に応じて繰り返す処理が最低でも一巡することとなる。 30

【0046】

また、メイン制御部41は、メイン制御部41の備えるレジスタの値を、スタックポインタSPが示すRAM 41cのスタック領域に所定の順序で記憶させることで退避させることが可能であり、RAM 41cのスタック領域に退避させていたレジスタの値を、退避させるときと逆の順序で当該スタック領域から順次読み出して、当該順序に対応するレジスタに設定することで、該当するレジスタの状態を退避させたときの状態に復帰させることが可能である。

【0047】

演出制御基板90には、前述の演出用スイッチ56が接続されており、その検出信号が入力されるようになっている。また、演出制御基板90には、前述の液晶表示器51、音声出力可能なスピーカ、演出に用いられ遊技者側から視認可能な演出用LED 57等の演出装置が接続されており、これら演出装置の出力状態は、演出制御基板90に搭載されたサブ制御部91により制御可能となっている。サブ制御部91は、メイン制御部41から送信されるコマンドや演出用スイッチ56の検出信号を受けて、演出を行うための各種の制御等を行うようになっている。 40

【0048】

本実施例のスロットマシン1は、設定値に応じてメダルの払出率が変わる構成である。詳しくは、内部抽選やAT抽選等の遊技者に対する有利度に影響する抽選において設定値に応じた当選確率を用いることにより、メダルの払出率が変わるようになっている。設定値 50

は 1 ～ 6 の 6 段階からなり、6 が最も払出率が高く、5、4、3、2、1 の順に値が小さくなるほど払出率が低くなる。すなわち設定値として 6 が設定されている場合には、遊技者にとって最も有利度が高く、5、4、3、2、1 の順に値が小さくなるほど有利度が段階的に低くなる。

#### 【0049】

設定値を変更するためには、設定キースイッチ 37 を ON 状態としてからスロットマシン 1 の電源スイッチ 39 を ON にする必要がある。設定キースイッチ 37 を ON 状態として電源を ON にすると、設定値表示器 24 に RAM 41 c から読み出された設定値が表示値として表示され、リセット / 設定スイッチ 38 の操作による設定値の変更が可能な設定変更状態に移行する。設定変更状態において、リセット / 設定スイッチ 38 が操作されると、設定値表示器 24 に表示された表示値が 1 ずつ更新されていく（設定値 6 からさらに操作されたときは、設定値 1 に戻る）。そして、スタートスイッチ 7 が操作されると表示値を設定値として確定する。そして、設定キースイッチ 37 が OFF にされると、確定した表示値（設定値）がメイン制御部 41 の RAM 41 c に格納され、遊技の進行が可能な状態に移行する。

10

#### 【0050】

尚、設定キースイッチ 37、リセット / 設定スイッチ 38 は、スロットマシン 1 の内部に設けられ、所定のキー操作により開放可能な前面扉 1 b を開放しない限り操作不可能とされており、スロットマシン 1 が設置される遊技店の店員のうち所定のキーを所持する店員のみが操作可能となる。特に、設定キースイッチ 37 は、さらにキー操作を要することから、遊技店の店員の中でも、設定キースイッチ 37 の操作を行うためのキーを所持する店員のみが操作可能とされている。また、リセットスイッチ 23 は、前面扉 1 b を開放する必要はないが、所定のキーを用いたキー操作を必要とするため、所定のキーを所持する店員のみが操作可能となる。また、リセット / 設定スイッチ 38 は、通常時においてはエラー状態を解除するためのリセットスイッチとしても機能するようになっている。

20

#### 【0051】

本実施例のスロットマシン 1 においては、メイン制御部 41 は、割込処理において電圧低下が検出されているか否かを判定する停電判定処理を行い、停電判定処理において電圧低下が検出されていると判定した場合に、次回復帰時に RAM 41 c のデータが正常か否かを判定するためのデータを設定する電断処理（メイン）を実行する。

30

#### 【0052】

そして、メイン制御部 41 は、その起動時において RAM 41 c のデータが正常であることを条件に、RAM 41 c に記憶されているデータに基づいてメイン制御部 41 の処理状態を電断前の状態に復帰させることが可能とされている。一方、起動時に RAM 41 c のデータが正常でない場合には、RAM 異常と判定し、RAM 異常を示すエラーフラグを RAM 41 c に設定するとともに、RAM 異常エラー状態に制御し、遊技の進行を不能化させるようになっている。RAM 異常エラー状態は、設定変更状態に移行し、新たに設定値が設定されることで解除され、遊技の進行が可能となる。

#### 【0053】

本実施例のスロットマシン 1 においては、メイン制御部 41 は、遊技の進行に応じて異常を検出した場合に、検出した異常の種類を示すエラーフラグを RAM 41 c に設定するとともに、一般エラー状態に制御し、遊技の進行を不能化させるようになっている。一般エラー状態は、リセットスイッチ 23 またはリセット / 設定スイッチ 38 によるリセット操作により解除され、遊技の進行が可能となる。尚、以下では、RAM 異常エラー状態と、一般エラー状態と、を区別する必要がない場合に、単にエラー状態と呼ぶ。

40

#### 【0054】

本実施例のスロットマシン 1 においてゲームを行う場合には、まず、メダルをメダル投入部 4 から投入するか、あるいは MAX BET スイッチ 6 を操作してクレジットを使用して賭数を設定する。遊技状態に応じて定められた規定数の賭数が設定されると、予め定められた入賞ライン LN（図 1 参照、本実施例では、リール 2 L、2 C、2 R の中段、すなわ

50

ち中段に水平方向に並んだ図柄に跨がって設定されている)が有効となり、スタートスイッチ7の操作が有効な状態、すなわち、ゲームが開始可能な状態となる。尚、遊技状態に対応する規定数のうち最大数を超えてメダルが投入された場合には、その分はクレジットに加算される。また、本実施例では、1本の入賞ラインのみを適用しているが、複数の入賞ラインを適用しても良い。

#### 【0055】

また、本実施例では、入賞ラインLNに入賞を構成する図柄の組合せが揃ったことを認識しやすくするために、入賞ラインLNとは別に、無効ラインLM1~6を設定している。無効ラインLM1~6は、これら無効ラインLM1~6に揃った図柄の組合せによって入賞が判定されるものではなく、入賞ラインLNに入賞を構成する図柄の組合せが揃った際に、無効ラインLM1~6のいずれかに入賞を示唆する示唆図柄の組合せ(例えば、ベルa-ベルa-ベルa)が揃う構成とすることで、入賞ラインLNに入賞を構成する図柄の組合せが揃ったことを認識しやすくするものである。本実施例では、図1に示すように、リール2L、2C、2Rの上段、すなわち上段に水平方向に並んだ図柄に跨がって設定された無効ラインLM1、リール2L、2C、2Rの下段、すなわち下段に水平方向に並んだ図柄に跨って設定された無効ラインLM2、リール2Lの上段、リール2Cの中段、リール2Rの下段、すなわち右下がりに並んだ図柄に跨って設定された無効ラインLM3、リール2Lの下段、リール2Cの中段、リール2Rの上段、すなわち右上がりに並んだ図柄に跨って設定された無効ラインLM4、リール2Lの上段、リール2Cの中段、リール2Rの上段、すなわち小V字状に並んだ図柄に跨って設定された無効ラインLM5、リール2Lの下段、リール2Cの中段、リール2Rの下段、すなわち小山状に並んだ図柄に跨って設定された無効ラインLM6の6種類が無効ラインとして定められている。

10

20

#### 【0056】

ゲームが開始可能な状態でスタートスイッチ7を操作すると、各リール2L、2C、2Rが回転され、各リール2L、2C、2Rの図柄が連続的に変動される。リール2L、2C、2Rが回転されている状態で、いずれかのストップスイッチ8L、8C、8Rを操作すると、対応するリール2L、2C、2Rに対して停止制御が行われ、当該リールの回転が停止され、当該リールの図柄が透視窓3に表示結果として導出表示される。

#### 【0057】

停止制御では、各ストップスイッチについて操作が行われたときから最大停止遅延時間(本実施例では、190ms(ミリ秒))以内に、操作に対応するリール2L、2C、2Rの回転を停止させる制御が行われ、最大停止遅延時間(190ms)が経過するまでの間では、最大で4コマ分の図柄を引き込むことができることとなる。つまり、停止制御では、ストップスイッチ8L、8C、8Rが操作されたときに表示されている図柄と、そこから4コマ先までにある図柄、合計5コマ分(引込範囲)の図柄から一の図柄を選択して、リール2L、2C、2Rに導出させることが可能である。

30

#### 【0058】

これにより、停止制御では、各リール2L、2C、2Rについて対応するストップスイッチ8L、8C、8Rが操作されることで回転を停止させる際に、後述する内部抽選にて当選している入賞役を構成する図柄が、ストップスイッチが操作されたときの図柄から4コマ先までの引込範囲内にある場合には、当該図柄を入賞ラインLN上に引き込んで、ストップスイッチが操作されたリールの回転を停止させる一方で、内部抽選にて当選している入賞役を構成する図柄が、当該引込範囲内にはない場合には、内部抽選にて当選していない入賞役を構成する図柄が、入賞ラインLN上に停止しないように、いずれの入賞役も構成しない図柄を入賞ラインLN上に引き込んで、ストップスイッチが操作されたリールの回転を停止させるように制御することが可能である。

40

#### 【0059】

[メイン制御部のメモリ領域とプログラムについて]

メイン制御部41は、所定のプログラムやデータを記憶するメモリ領域を有するROM41b及びデータを一時的に記憶するRAM41cを備える。

50

## 【 0 0 6 0 】

図 3 は、ROM 4 1 b におけるプログラム / データ領域及び RAM 4 1 c における使用可能領域のアドレスマップである。図 3 ( a ) に示すように、ROM 4 1 b におけるプログラム / データ領域は、遊技の進行に係わる制御を行うための容量内プログラムが記憶される容量内プログラム領域と、容量内プログラムが用いる容量内データが記憶される容量内データ領域と、未使用領域 1 と、エラーの検出や試験信号の作成、払出率等の遊技機の情報表示等、遊技の進行に直接係わらない制御を行うための容量外プログラムが記憶される容量外プログラム領域と、容量外プログラムが用いる容量外データが記憶される容量外データ領域と、未使用領域 2 と、を含む。

## 【 0 0 6 1 】

尚、遊技の進行とは、ゲームを構成する一連のプロセスを進行させることであり、スロットマシンであれば、賭数を設定してゲームを開始可能とする段階、ゲームを開始してリールを回転させる段階、リールを停止させて表示結果を導出させる段階、表示結果に応じてメダル等の価値を付与する段階、を進行させることである。また、パチンコ遊技機であれば、始動入賞を判定する段階、可変表示を開始させる段階、可変表示を停止させる段階、遊技状態を変更する段階、を進行させることである。

## 【 0 0 6 2 】

以下では、容量内プログラムが行う処理を容量内処理と称す場合があり、容量外プログラムが行う処理を容量外処理と称す場合がある。

## 【 0 0 6 3 】

図 3 ( b ) に示すように、RAM 4 1 c の使用可能領域は、容量内プログラムがワークとして用いる容量内 RAM 領域と、容量外プログラムがワークとして用いる容量外 RAM 領域と、未使用領域 4 と、を含んでおり、容量内 RAM 領域には、容量内ワークと、未使用領域 3 と、容量内スタック領域と、が含まれ、容量外 RAM 領域には、容量外ワークと、容量外スタック領域と、が含まれる。尚、本実施例では、容量内スタック領域と容量外スタック領域とをそれぞれ異なる領域に個別に備える構成であるが、容量内プログラム及び容量外プログラムが共用する一のスタック領域を備える構成としても良い。

## 【 0 0 6 4 】

容量内スタック領域は、容量内プログラムがデータを一時的に退避する領域であり、容量内プログラムがデータを退避するにあたって、容量内スタックポインタが示すアドレスの領域に当該データが一時的に格納される。

## 【 0 0 6 5 】

また、容量外スタック領域は、容量外プログラムがデータを一時的に退避する領域であり、容量外プログラムがデータを退避するにあたって、容量外スタックポインタが示すアドレスの領域に当該データが一時的に格納される。

## 【 0 0 6 6 】

また、未使用領域 3 は、容量内プログラム及び容量外プログラムのいずれも使用しない領域であり、予め定められた容量の容量内 RAM 領域に対して余剰となった領域である。

## 【 0 0 6 7 】

また、未使用領域 4 は、容量内プログラム及び容量外プログラムのいずれも使用しない 1 6 バイト以上の領域である。容量内 RAM 領域と、容量外 RAM 領域とが、未使用領域 4 を挟んで連続しない領域に割り当てられているため、遊技の進行に係わる容量内プログラムが用いる容量内 RAM 領域と、遊技の進行に係わらない容量外プログラムが用いる容量外 RAM 領域と、を記憶領域の違いに応じて容易に特定することができる。

## 【 0 0 6 8 】

以下では、容量内プログラム領域、容量内データ領域及び容量内 RAM 領域をまとめて容量内領域と称す場合があり、容量外プログラム領域、容量外データ領域及び容量外 RAM 領域をまとめて容量外領域と称す場合がある。

## 【 0 0 6 9 】

[ 容量内領域と容量外領域の関係について ]

10

20

30

40

50

メイン制御部 4 1 の CPU 4 1 a は、図 4 に示すように、容量内プログラムに基づく処理において容量外プログラムを呼び出して容量外プログラムに基づく処理を実行し、容量外プログラムに基づく処理の終了後、容量内プログラムに基づく処理に復帰する。

【 0 0 7 0 】

図 4 に示すように、CPU 4 1 a は、原則として容量内プログラムに基づく処理を実行するにあたり、容量内データ領域の容量内データを参照して容量内プログラムに基づく処理を実行するとともに、容量内 RAM 領域をワークとして使用し、容量内 RAM 領域の内容を参照及び更新することが可能である。また、CPU 4 1 a は、原則として容量外プログラムに基づく処理を実行するにあたり、容量外データ領域の容量外データを参照して容量外プログラムに基づく処理を実行するとともに、容量外 RAM 領域をワークとして使用し、容量外 RAM 領域の内容を参照及び更新することが可能である。 10

【 0 0 7 1 】

また、CPU 4 1 a は、容量内プログラムに基づく処理を実行するにあたり、容量外 RAM 領域を更新することはないが、容量外 RAM 領域を参照することは可能であり、容量外プログラムに基づく処理を実行するにあたり、容量内 RAM 領域を更新することはないが、容量内 RAM 領域を参照することは可能である。

【 0 0 7 2 】

[ レジスタについて ]

メイン制御部 4 1 は、メイン CPU 4 1 a が演算を行うために用いられる複数のレジスタを備える。当該複数のレジスタは、主に表レジスタ、裏レジスタ、共通レジスタを含む。 20  
表レジスタと裏レジスタは、互いに対の関係にあり、表レジスタ使用時には裏レジスタが使用不可であり、裏レジスタ使用時には表レジスタが使用不可である。また、表レジスタを反転させることで裏レジスタが使用可能となり、裏レジスタを反転させることで表レジスタが使用可能となる。また、共通レジスタは、表レジスタが使用されている場合にも裏レジスタが使用されている場合にも使用可能である。

【 0 0 7 3 】

表レジスタには、A レジスタ、F レジスタ、汎用レジスタ ( B レジスタ、C レジスタ ( B レジスタ、C レジスタをまとめて B C レジスタと呼ぶ。 ) 、D レジスタ、E レジスタ ( D レジスタ、E レジスタをまとめて D E レジスタと呼ぶ。 ) 、H レジスタ、L レジスタ ( H レジスタ、L レジスタをまとめて H L レジスタと呼ぶ。 ) ) 、Q レジスタが含まれる。A 30  
レジスタ、汎用レジスタは、演算に用いられる値が格納され、F レジスタは、演算結果に応じたフラグが格納され、Q レジスタは、特殊命令においてアドレスを指定する際の上位アドレスが格納される。

【 0 0 7 4 】

裏レジスタには、表レジスタの A レジスタ、F レジスタ、汎用レジスタと対となる、A ' レジスタ、F ' レジスタ、汎用レジスタ ( B ' レジスタ、C ' レジスタ ( B ' レジスタ、C ' レジスタをまとめて B ' C ' レジスタと呼ぶ。 ) 、D ' レジスタ、E ' レジスタ ( D ' レジスタ、E ' レジスタをまとめて D ' E ' レジスタと呼ぶ。 ) 、H ' レジスタ、L ' レジスタ ( H ' レジスタ、L ' レジスタをまとめて H ' L ' レジスタと呼ぶ。 ) ) が含まれる。A ' レジスタ、汎 40  
用レジスタは、演算に用いられる値が格納され、F ' レジスタは、演算結果に応じたフラグが格納される。

【 0 0 7 5 】

共通レジスタには、I レジスタ、R レジスタ、I X レジスタ、I Y レジスタ、S P レジスタ、P C レジスタが含まれる。I レジスタは、割込発生時にジャンプするアドレス ( 割込ベクタ ) の上位アドレスが格納され、R レジスタは、メモリをリフレッシュするタイミングをカウントする値が格納され、I X レジスタ、I Y レジスタは、演算に用いられる値が格納され、S P レジスタは、スタック領域を特定するためのスタックポインタの値が格納され、P C レジスタは、実行中のプログラムのアドレスを示すプログラムカウンタの値が格納される。

【 0 0 7 6 】

〔メイン制御部が実行する各種処理について〕

次に、メイン制御部 4 1 が実行する各種処理について説明する。図 6 に示すように、メイン制御部 4 1 は、スロットマシン 1 を制御するための処理として基本処理と、割込処理を実行可能である。

【 0 0 7 7 】

基本処理は、制御状態が移行するまで当該制御状態に応じた処理を繰り返しループする処理であり、メインルーチン（基本）と、メインルーチン（基本）から呼び出される複数種類のサブルーチン（基本）と、から構成されている。

【 0 0 7 8 】

メインルーチン（基本）は、遊技の進行に伴い複数の制御状態を段階的に移行させる処理であり、プログラムの進行に応じて、複数種類のサブルーチン（基本）の中からいずれかの種類のサブルーチン（基本）を呼び出すことが可能である。 10

【 0 0 7 9 】

サブルーチン（基本）は、メインルーチン（基本）に呼び出されることで開始し、当該サブルーチン（基本）が終了することでメインルーチン（基本）の呼出元に復帰させる。

【 0 0 8 0 】

また、図 7 に示すように、基本処理は、容量内プログラムによる容量内処理（基本）と、容量外プログラムによる容量外処理（基本）と、から構成されている。

【 0 0 8 1 】

容量内処理（基本）は、プログラムの進行に応じて、複数種類の容量外処理（基本）の中からいずれかの種類の容量外処理（基本）を呼び出すことが可能である。 20

【 0 0 8 2 】

容量外処理（基本）は、容量内処理（基本）に呼び出されることで開始し、当該容量外処理（基本）が終了することで容量内処理（基本）の呼出元に復帰させる。

【 0 0 8 3 】

容量外処理（基本）は、メインルーチン（容量外・基本）と、メインルーチン（容量外・基本）から呼び出される複数種類のサブルーチン（容量外・基本）と、から構成されている。

【 0 0 8 4 】

メインルーチン（容量外・基本）は、プログラムの進行に応じて、複数種類のサブルーチン（容量外・基本）の中からいずれかの種類のサブルーチン（容量外・基本）を呼び出すことが可能である。 30

【 0 0 8 5 】

サブルーチン（容量外・基本）は、メインルーチン（容量外・基本）に呼び出されることで開始し、当該サブルーチン（容量外・基本）が終了することでメインルーチン（容量外・基本）の呼出元に復帰させる。

【 0 0 8 6 】

割込処理は、タイマ割込が発生する毎に基本処理に割り込んで実行され、当該割込処理が終了することで基本処理の割込元に復帰させる処理であり、メインルーチン（割込）と、メインルーチン（割込）から呼び出される複数種類のサブルーチン（割込）と、から構成されている。 40

【 0 0 8 7 】

メインルーチン（割込）は、プログラムの進行に応じて、複数種類のサブルーチン（割込）の中からいずれかの種類のサブルーチン（割込）を呼び出すことが可能である。

【 0 0 8 8 】

サブルーチン（割込）は、メインルーチン（割込）に呼び出されることで開始し、当該サブルーチン（割込）が終了することでメインルーチン（割込）の呼出元に復帰させる。

【 0 0 8 9 】

また、図 8 に示すように、割込処理は、容量内プログラムによる容量内処理（割込）と、容量外プログラムによる容量外処理（割込）と、から構成されている。 50

## 【 0 0 9 0 】

容量内処理（割込）は、プログラムの進行に応じて、複数種類の容量外処理（割込）の中からいずれかの種類の容量外処理（割込）を呼び出すことが可能である。

## 【 0 0 9 1 】

容量外処理（割込）は、容量内処理（割込）に呼び出されることで開始し、当該容量外処理（割込）が終了することで容量内処理（割込）の呼出元に復帰させる。

## 【 0 0 9 2 】

容量外処理（割込）は、メインルーチン（容量外・割込）と、メインルーチン（容量外・割込）から呼び出される複数種類のサブルーチン（容量外・割込）と、から構成されている。

10

## 【 0 0 9 3 】

メインルーチン（容量外・割込）は、プログラムの進行に応じて、複数種類のサブルーチン（容量外・割込）の中からいずれかの種類のサブルーチン（容量外・割込）を呼び出すことが可能である。

## 【 0 0 9 4 】

サブルーチン（容量外・割込）は、メインルーチン（容量外・割込）に呼び出されることで開始し、当該サブルーチン（容量外・割込）が終了することでメインルーチン（容量外・割込）の呼出元に復帰させる。

## 【 0 0 9 5 】

[ 基本処理及び割込処理において使用されるレジスタについて ]

20

次に、基本処理及び割込処理の実行時にメイン制御部 4 1 が使用するレジスタについて説明する。

## 【 0 0 9 6 】

図 6 に示すように、メイン制御部 4 1 は、基本処理におけるメインルーチン（基本）の実行時に、表レジスタと共通レジスタを使用して各種処理を実行する。詳しくは、メインルーチン（基本）において、表レジスタのうち、A レジスタ、F レジスタ、B C レジスタ、D E レジスタ、H L レジスタ、Q レジスタ、すなわち表レジスタに含まれる全てのレジスタを使用する。また、メインルーチン（基本）において、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、I レジスタ、R レジスタ、I X レジスタ、S P レジスタ、P C レジスタを使用し、I Y レジスタを使用しない。

30

## 【 0 0 9 7 】

また、メイン制御部 4 1 は、基本処理におけるサブルーチン（基本）の実行時に、表レジスタと共通レジスタを使用して各種処理を実行する。詳しくは、サブルーチン（基本）において、表レジスタのうち、A レジスタ、F レジスタ、B C レジスタ、D E レジスタ、H L レジスタ、Q レジスタ、すなわち表レジスタに含まれる全てのレジスタを使用する。また、サブルーチン（基本）において、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、I レジスタ、R レジスタ、I X レジスタ、S P レジスタ、P C レジスタを使用し、I Y レジスタを使用しない。

## 【 0 0 9 8 】

図 6 に示すように、メイン制御部 4 1 は、基本処理の実行中にタイマ割込が発生し、実行中の基本処理に割り込んで割込処理が開始すると、まず使用するレジスタを表レジスタから裏レジスタに交換する。

40

## 【 0 0 9 9 】

メイン制御部 4 1 は、割込処理におけるメインルーチン（割込）の実行時に、裏レジスタと共用レジスタを使用して各種処理を実行する。詳しくは、メインルーチン（割込）において、裏レジスタのうち、A ' レジスタ、F ' レジスタ、B ' C ' レジスタ、D ' E ' レジスタ、H ' L ' レジスタ、すなわち裏レジスタに含まれる全てのレジスタを使用する。また、メインルーチン（割込）において、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、R レジスタ、S P レジスタ、P C レジスタを使用し、I レジスタ、I X レジスタ、I Y レジスタを使用しない。

50

## 【 0 1 0 0 】

また、メイン制御部 4 1 は、使用するレジスタを表レジスタから裏レジスタに交換すると、前回の割込処理で使用されていた裏レジスタの値が格納された状態となるが、今回の割込処理では、前回の割込処理で使用されていた裏レジスタの値を使用せず、改めて裏レジスタに必要な値を格納する。

## 【 0 1 0 1 】

また、メイン制御部 4 1 は、割込処理におけるサブルーチン（割込）の実行時に、裏レジスタと共通レジスタを使用して各種処理を実行する。詳しくは、サブルーチン（割込）において、裏レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、A'レジスタ、F'レジスタ、B' C'レジスタ、H' L'レジスタを使用し、D' E'レジスタを使用しない。また、サブルーチン（割込）において、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、Rレジスタ、I Yレジスタ、S Pレジスタ、P Cレジスタを使用し、Iレジスタ、I Xレジスタを使用しない。

## 【 0 1 0 2 】

また、メイン制御部 4 1 は、割込処理におけるメインルーチン（割込）が終了すると、使用するレジスタを裏レジスタから表レジスタに交換し、タイマ割込発生時に実行していた基本処理に復帰する。

## 【 0 1 0 3 】

このように本実施例では、基本処理において表レジスタを使用し、基本処理に割り込んで実行される割込処理において表レジスタから交換した裏レジスタを使用する。また、基本処理が実行されるときにも、割込処理が実行されるときにも、I Yレジスタを使用可能であるが、I Yレジスタは割込処理において使用され、基本処理では使用されないの、基本処理と割込処理との間においてI Yレジスタに格納されるデータが混同してしまうことを防止できる。

20

## 【 0 1 0 4 】

また、本実施例では、基本処理が実行されるときにも、割込処理が実行されるときにも、I Xレジスタ及びI Yレジスタからなる複数のレジスタを使用可能であるが、I Xレジスタ及びI YレジスタのうちI Xレジスタは基本処理に使用され、I Yレジスタは基本処理に使用されないため、基本処理と割込処理との間において、I Xレジスタ及びI Yレジスタからなる複数のレジスタのうち少なくともI Yレジスタに格納されるデータが混同してしまうことを防止できる。

30

## 【 0 1 0 5 】

尚、本実施例では、I Xレジスタ及びI Yレジスタからなる複数のレジスタのうちI Xレジスタを基本処理で使用する構成であるが、I Xレジスタ及びI Yレジスタからなる複数のレジスタのいずれも基本処理に使用されない構成としても良く、このような構成とすることで、基本処理を実行するときにも、割込処理を実行するときにも使用可能なI Xレジスタ及びI Yレジスタのいずれも基本処理では使用されないの、基本処理と割込処理との間においてI Xレジスタ及びI Yレジスタからなる複数のレジスタに格納されるデータが混同してしまうことを防止できる。

## 【 0 1 0 6 】

本実施例では、基本処理において表レジスタを使用し、基本処理に割り込んで実行される割込処理において表レジスタから交換した裏レジスタを使用する。また、基本処理が実行されるときにも、割込処理が実行されるときにも、I Xレジスタを使用可能であるが、I Xレジスタは基本処理において使用され、割込処理では使用されないの、基本処理と割込処理との間においてI Xレジスタに格納されるデータが混同してしまうことを防止できる。

40

## 【 0 1 0 7 】

また、本実施例では、基本処理が実行されるときにも、割込処理が実行されるときにも、I Xレジスタ及びI Yレジスタからなる複数のレジスタを使用可能であるが、I Xレジスタ及びI YレジスタのうちI Yレジスタは割込処理に使用され、I Xレジスタは割込処理

50

に使用されないため、基本処理と割込処理との間において、I Xレジスタ及びI Yレジスタからなる複数のレジスタのうち少なくともI Xレジスタに格納されるデータが混同してしまうことを防止できる。

【0108】

尚、本実施例では、I Xレジスタ及びI Yレジスタからなる複数のレジスタのうちI Yレジスタを割込処理で使用する構成であるが、I Xレジスタ及びI Yレジスタからなる複数のレジスタのいずれも割込処理に使用されない構成としても良く、このような構成とすることで、基本処理を実行するときにも、割込処理を実行するときにも使用可能なI Xレジスタ及びI Yレジスタのいずれも割込処理では使用しないので、基本処理と割込処理との間においてI Xレジスタ及びI Yレジスタからなる複数のレジスタに格納されるデータが混同してしまうことを防止できる。

10

【0109】

また、本実施例では、割込処理を実行するときに、使用するレジスタを表レジスタから裏レジスタに交換すると、前回の割込処理で使用されていた裏レジスタの値が格納された状態となるが、今回の割込処理では、前回の割込処理で使用されていた裏レジスタの値を使用せず、改めて裏レジスタに必要な値を格納するので、意図しない値で割込処理が実行されてしまうことを防止できる。

【0110】

本実施例では、割込処理は、メインルーチン（割込）と、メインルーチン（割込）から呼び出されて実行されるサブルーチン（割込）と、から構成されるとともに、割込処理において使用される複数のレジスタのうちI Yレジスタは、サブルーチン（割込）で使用され、メインルーチン（割込）で使用されないため、メインルーチン（割込）とサブルーチン（割込）の間でI Yレジスタに格納されるデータが混同してしまうことを防止できる。

20

【0111】

また、メインルーチン（割込）から呼び出されるサブルーチン（割込）は、複数種類のサブルーチン（割込）を含んでおり、I Yレジスタは、複数種類のサブルーチン（割込）の全てで使用されるのではなく、いずれかの種類のサブルーチン（割込）で使用される。このため、メインルーチン（割込）とI Yレジスタを使用するサブルーチン（割込）との間でI Yレジスタに格納されるデータが混同してしまうことを防止できる。

【0112】

30

尚、本実施例では、I Yレジスタは、一部の種類のサブルーチン（割込）において使用される構成であるが、全ての種類のサブルーチン（割込）において使用される構成としても良く、このような構成においても、メインルーチン（割込）とサブルーチン（割込）との間でI Yレジスタに格納されるデータが混同してしまうことを防止できる。

【0113】

また、本実施例では、割込処理が、メインルーチン（割込）と、メインルーチン（割込）から呼び出されて実行されるサブルーチン（割込）と、から構成されるとともに、割込処理において使用される複数のレジスタのうちI Yレジスタは、サブルーチン（割込）で使用され、メインルーチン（割込）で使用されない構成であるが、基本処理において使用される複数のレジスタのうち所定レジスタが、基本処理を構成するサブルーチン（基本）において使用され、メインルーチン（基本）で使用されない構成としても良く、このような構成とすることで、メインルーチン（基本）とサブルーチン（基本）の間で所定レジスタに格納されるデータが混同してしまうことを防止できる。

40

【0114】

本実施例では、割込処理は、メインルーチン（割込）と、メインルーチン（割込）から呼び出されて実行されるサブルーチン（割込）と、から構成されるとともに、割込処理において使用される複数のレジスタのうちD' E' レジスタは、メインルーチン（割込）で使用され、サブルーチン（割込）で使用されないため、メインルーチン（割込）とサブルーチン（割込）の間でD' E' レジスタに格納されるデータが混同してしまうことを防止できる。

。

50

## 【 0 1 1 5 】

また、メインルーチン（割込）から呼び出されるサブルーチン（割込）は、複数種類のサブルーチン（割込）を含んでおり、D' E' レジスタは、いずれの種類のサブルーチン（割込）においても使用されない。このため、メインルーチン（割込）といずれの種類のサブルーチン（割込）との間においてもD' E' レジスタに格納されるデータが混同してしまうことを防止できる。

## 【 0 1 1 6 】

尚、本実施例では、D' E' レジスタは、いずれの種類のサブルーチン（割込）においても使用されない構成であるが、特定の種類のサブルーチン（割込）において使用されず、他の種類のサブルーチン（割込）において使用される構成としても良く、このような構成においても、メインルーチン（割込）と特定の種類のサブルーチン（割込）との間でD' E' レジスタに格納されるデータが混同してしまうことを防止できる。

## 【 0 1 1 7 】

また、本実施例では、割込処理が、メインルーチン（割込）と、メインルーチン（割込）から呼び出されて実行されるサブルーチン（割込）と、から構成されるとともに、割込処理において使用される複数のレジスタのうちD' E' レジスタは、メインルーチン（割込）で使用され、サブルーチン（割込）で使用されない構成であるが、基本処理において使用される複数のレジスタのうち特定レジスタが、基本処理を構成するメインルーチン（基本）において使用され、サブルーチン（基本）で使用されない構成としても良く、このような構成とすることで、メインルーチン（基本）とサブルーチン（基本）の間で特定レジスタに格納されるデータが混同してしまうことを防止できる。

## 【 0 1 1 8 】

[ 容量内処理及び容量外処理において使用されるレジスタについて ]

次に、容量内処理及び容量外処理の実行時にメイン制御部 4 1 が使用するレジスタについて説明する。

## 【 0 1 1 9 】

図 7 に示すように、メイン制御部 4 1 は、容量内処理（基本）の実行時に、表レジスタと共通レジスタを使用して各種処理を実行する。詳しくは、容量内処理（基本）において、表レジスタのうち、A レジスタ、F レジスタ、B C レジスタ、D E レジスタ、H L レジスタ、Q レジスタ、すなわち表レジスタに含まれる全てのレジスタを使用する。また、容量内処理（基本）において、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、I レジスタ、R レジスタ、I X レジスタ、S P レジスタ、P C レジスタを使用し、I Y レジスタを使用しない。

## 【 0 1 2 0 】

図 7 に示すように、メイン制御部 4 1 は、容量外処理（基本）を実行する場合に、容量内処理（基本）からメインルーチン（容量外・基本）を呼び出す。

## 【 0 1 2 1 】

メイン制御部 4 1 は、メインルーチン（容量外・基本）を開始すると、まず S P レジスタに格納されている容量内スタックポインタの値を容量外 R A M 領域の所定領域に退避させた後、S P レジスタに容量外スタック領域の 0 段目を示す値を格納する。これにより、データを退避する領域が容量内スタック領域から容量外スタック領域に切り替わる。

## 【 0 1 2 2 】

その後、表レジスタ及び共通レジスタの全ての値を、容量外スタック領域のうち容量外スタックポインタが示す領域に所定の順序で退避させる。

## 【 0 1 2 3 】

メイン制御部 4 1 は、メインルーチン（容量外・基本）の実行時に、表レジスタと共通レジスタを使用して各種演算を実行する。詳しくは、メインルーチン（容量外・基本）において、表レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、A レジスタ、F レジスタ、B C レジスタ、H L レジスタ、Q レジスタを使用し、D E レジスタを使用しない。また、メインルーチン（容量外・基本）において、共通レジスタに含まれる全て

のレジスタを使用可能であるが、そのうち、Iレジスタ、Rレジスタ、SPレジスタ、PCレジスタを使用し、IXレジスタ、IYレジスタを使用しない。

【0124】

また、メイン制御部41は、メインルーチン（容量外・基本）において、サブルーチン（容量外・基本）を呼び出すことでサブルーチン（容量外・基本）を実行する。この際、容量外スタック領域のうちSPレジスタに格納された値が示す領域に呼出元のアドレスが格納される。また、サブルーチン（容量外・基本）においても容量外スタック領域にデータを退避することがあり、容量外スタック領域は最大N段使用されることとなり、この際、SPレジスタに格納されている容量外スタックポインタは、容量外スタック領域のうちN段目の領域を示す値となる。

10

【0125】

図7に示すように、メイン制御部41は、サブルーチン（容量外・基本）の実行時に、表レジスタと共通レジスタを使用して各種処理を実行する。詳しくは、サブルーチン（容量外・基本）において、表レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、Aレジスタ、Fレジスタ、BCレジスタ、HLレジスタ、Qレジスタを使用し、DEレジスタを使用しない。また、サブルーチン（容量外・基本）において、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、Iレジスタ、Rレジスタ、SPレジスタ、PCレジスタを使用し、IXレジスタ、IYレジスタを使用しない。

【0126】

図7に示すように、メイン制御部41は、メインルーチン（容量外・基本）を終了するとき、容量外スタック領域に退避させていた表レジスタ及び共通レジスタの全ての値を全て復帰させる。これにより、容量外スタック領域に退避していたデータは全て復帰することとなり、容量外スタック領域は空となる。その後、メイン制御部41は、容量外RAM領域の所定領域に退避させた容量内スタックポインタの値をSPレジスタに復帰させる。これにより、データを退避する領域が、容量外スタック領域から容量内スタック領域に切り替わる。そして、メインルーチン（容量外・基本）を終了させて容量内処理（基本）における呼出元に復帰する。

20

【0127】

図8に示すように、メイン制御部41は、容量内処理（割込）の実行時に、裏レジスタと共通レジスタを使用して各種処理を実行する。詳しくは、容量内処理（割込）において、裏レジスタのうち、A'レジスタ、F'レジスタ、B'C'レジスタ、D'E'レジスタ、H'L'レジスタ、すなわち裏レジスタに含まれる全てのレジスタを使用する。また、容量内処理（割込）において、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、Rレジスタ、SPレジスタ、PCレジスタを使用し、Iレジスタ、IXレジスタ、IYレジスタを使用しない。

30

【0128】

図8に示すように、メイン制御部41は、容量外処理（割込）を実行する場合に、容量内処理（割込）からメインルーチン（容量外・割込）を呼び出す。

【0129】

メイン制御部41は、メインルーチン（容量外・割込）を開始すると、まずSPレジスタに格納されている容量内スタックポインタの値を容量外RAM領域の所定領域に退避させた後、SPレジスタに容量外スタック領域の0段目を示す値を格納する。これにより、データを退避する領域が容量内スタック領域から容量外スタック領域に切り替わる。

40

【0130】

その後、裏レジスタ及び共通レジスタの全ての値を、容量外スタック領域のうち容量外スタックポインタが示す領域に所定の順序で退避させる。

【0131】

メイン制御部41は、メインルーチン（容量外・割込）の実行時に、裏レジスタと共通レジスタを使用して各種処理を実行する。詳しくは、裏レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、A'レジスタ、F'レジスタ、B'C'レジスタ、H'L'レジスタ、

50

ジスタを使用し、D'レジスタを使用しない。また、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、Rレジスタ、SPレジスタ、PCレジスタを使用し、Iレジスタ、IXレジスタ、IYレジスタを使用しない。

#### 【0132】

また、メイン制御部41は、メインルーチン（容量外・割込）において、サブルーチン（容量外・割込）を呼び出すことでサブルーチン（容量外・割込）を実行する。この際、容量外スタック領域のうちSPレジスタに格納された値が示す領域に呼出元のアドレスが格納される。また、サブルーチン（容量外・割込）においても容量外スタック領域にデータを退避することがあり、容量外スタック領域は最大N段使用されることとなり、この際、SPレジスタに格納されている容量外スタックポインタは、容量外スタック領域のうちN段目の領域を示す値となる。

#### 【0133】

図8に示すように、メイン制御部41は、サブルーチン（容量外・割込）の実行時に、裏レジスタと共通レジスタを使用して各種処理を実行する。詳しくは、サブルーチン（容量外・割込）においては、裏レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、A'レジスタ、F'レジスタ、B'C'レジスタ、H'L'レジスタを使用し、D'E'レジスタを使用しない。また、共通レジスタに含まれる全てのレジスタを使用可能であるが、そのうち、Rレジスタ、SPレジスタ、PCレジスタを使用し、Iレジスタ、IXレジスタ、IYレジスタを使用しない。

#### 【0134】

図8に示すように、メイン制御部41は、メインルーチン（容量外・基本）を終了するときに、容量外スタック領域に退避させていた裏レジスタ及び共通レジスタの全ての値を全て復帰させる。これにより、容量外スタック領域に退避していたデータは全て復帰することとなり、容量外スタック領域は空となる。その後、メイン制御部41は、容量外RAM領域の所定領域に退避させた容量内スタックポインタの値をSPレジスタに復帰させる。これにより、データを退避する領域が、容量外スタック領域から容量内スタック領域に切り替わる。そして、当該メインルーチン（容量外・割込）を終了させて容量内処理（割込）に復帰する。

#### 【0135】

このように本実施例では、基本処理において、容量内プログラムによる容量内処理（基本）と、容量外プログラムによる容量外処理（基本）と、を実行可能であり、容量内処理（基本）から容量外処理（基本）を呼び出して実行するときに、まず全てのレジスタに格納されている値をRAM41cにおける容量外スタック領域に退避させ、容量外処理（基本）を終了するときに、容量外スタック領域に退避させた全ての値を対応する全てのレジスタに復帰させる。

#### 【0136】

また、容量内処理（基本）を実行するときには、Aレジスタ、Fレジスタ、BCレジスタ、DEレジスタ、HLレジスタ、Qレジスタ、Iレジスタ、Rレジスタ、SPレジスタ、PCレジスタ、すなわち13のレジスタを使用するのに対し、容量外処理（基本）を実行するときには、Aレジスタ、Fレジスタ、BCレジスタ、HLレジスタ、Iレジスタ、Rレジスタ、SPレジスタ、PCレジスタ、すなわち10のレジスタを使用する。このため、容量外処理（基本）では、容量内処理（基本）が使用する複数のレジスタのうち一部のレジスタのみを使用し、容量外処理（基本）を実行するときの方が、容量内処理（基本）を実行するときよりも使用されるレジスタの数が少ないので、容量外処理（基本）の実行による誤ったデータの使用を防止できる。

#### 【0137】

尚、本実施例では、容量外処理（基本）では、容量内処理（基本）が使用する複数のレジスタのうち一部のレジスタのみを使用することで、容量外処理（基本）を実行するときの方が、容量内処理（基本）を実行するときよりも使用されるレジスタの数が少なくなる構成であるが、容量外処理（基本）と、容量内処理（基本）と、でそれぞれ別のレジスタを

使用するとともに、このような構成において、容量内処理（基本）が使用するレジスタの数よりも、容量外処理（基本）が使用するレジスタの数が少ない構成としても良く、このような構成においても、容量内処理（基本）を実行するときよりも使用されるレジスタの数が少ないので、容量外処理（基本）の実行による誤ったデータの使用を防止できる。

【0138】

また、本実施例では、割込処理において、容量内プログラムによる容量内処理（割込）と、容量外プログラムによる容量外処理（割込）と、を実行可能であり、容量内処理（割込）から容量外処理（割込）を呼び出して実行するとき、まず全てのレジスタに格納されている値をRAM41cにおける容量外スタック領域に退避させ、容量外処理（割込）を終了するとき、容量外スタック領域に退避させた全ての値を対応する全てのレジスタに復帰させる。 10

【0139】

また、容量内処理（割込）を実行するときには、A'レジスタ、F'レジスタ、B'C'レジスタ、D'E'レジスタ、H'L'レジスタ、Rレジスタ、SPレジスタ、PCレジスタ、すなわち11のレジスタを使用するのに対し、容量外処理（割込処理）を実行するときには、A'レジスタ、F'レジスタ、B'C'レジスタ、H'L'レジスタ、Rレジスタ、SPレジスタ、PCレジスタ、すなわち9のレジスタを使用する。このため、容量外処理（割込）では、容量内処理（割込）が使用する複数のレジスタのうち一部のレジスタのみを使用し、容量外処理（割込）を実行するときの方が、容量内処理（割込）を実行するときよりも使用されるレジスタの数が少ないので、容量外処理（割込）の実行による誤ったデータの 20

【0140】

尚、本実施例では、容量外処理（割込）では、容量内処理（割込）が使用する複数のレジスタのうち一部のレジスタのみを使用することで、容量外処理（割込）を実行するときの方が、容量内処理（割込）を実行するときよりも使用されるレジスタの数が少なくなる構成であるが、容量外処理（割込）と、容量内処理（割込）と、でそれぞれ別のレジスタを使用するとともに、このような構成において、容量内処理（割込）が使用するレジスタの数よりも、容量外処理（割込）が使用するレジスタの数が少ない構成としても良く、このような構成においても、容量内処理（割込）を実行するときよりも使用されるレジスタの数が少ないので、容量外処理（割込）の実行による誤ったデータの使用を防止できる。 30

【0141】

本実施例では、容量内プログラムによる容量内処理と、容量外プログラムによる容量外処理と、を実行可能であり、容量内処理から容量外処理を呼び出して実行するときSPレジスタに格納されている容量内スタックポインタの値を退避し、容量外スタック領域を示す値に切り替えることによって、データの退避先を容量内スタック領域から容量外スタック領域に切り替える。また、容量外処理（基本処理）を終了するとき、退避していた容量内スタックポインタの値をSPレジスタに戻すことによって、データの退避先を容量外スタック領域から容量内スタック領域に切り替える。

【0142】

そして、容量外処理を終了するとき、容量外処理の実行中に容量外スタック領域に退避させたデータを全て復帰させてから、データの退避先を容量外スタック領域から容量内スタック領域に切り替えるので、前回実行した容量外処理で使用されていたデータが退避したまま、さらに容量外処理が実行されることがなく、前回実行した容量外処理で使用されたデータと、今回実行する容量外処理で使用されるデータと、が混同してしまうことを防止できる。 40

【0143】

また、容量内処理から呼び出される容量外処理は、複数種類の容量外処理を含んでおり、いずれの種類の容量外処理であっても、容量外処理を終了するとき、容量外処理の実行中に容量外スタック領域に退避させたデータを全て復帰させてから、データの退避先を容量外スタック領域から容量内スタック領域に切り替えるので、容量外処理の種類に関わら 50

ず、前回実行した容量外処理で使用されていたデータが退避したまま、さらに容量外処理が実行されることがなく、前回実行した容量外処理で使用されたデータと、今回実行する容量外処理で使用されるデータと、が混同してしまうことを防止できる。

【 0 1 4 4 】

また、容量外処理の種類に関わらず、容量外処理を終了するときには、容量外スタック領域にはデータが格納されていない状態となるため、容量外処理を開始するときに、前回の容量外処理の終了時に設定されていた容量外スタックポインタの値を復帰させる必要がなく、容量外スタックポインタの値として常に容量外スタック領域の 0 段目を示す値を設定すれば良い。

【 0 1 4 5 】

10

[ レジスタの変形例について ]

次に、CPU 4 1 a が演算を行うために用いられる複数のレジスタの変形例について、図 9、図 1 0 に基づいて説明する。

【 0 1 4 6 】

図 5 ~ 図 8 で示した例では、複数のレジスタは、主に表レジスタ、裏レジスタ、共通レジスタを含み、基本処理であれば容量内処理であっても容量外処理であっても表レジスタ及び共通レジスタが使用され、割込処理であれば容量内処理であっても容量外処理であっても裏レジスタ及び共通レジスタが使用される構成であったが、図 9、図 1 0 で示す変形例では、複数のレジスタは、容量内レジスタ、容量外レジスタ、共通レジスタを含む。

【 0 1 4 7 】

20

容量内レジスタは、一組の表レジスタ及び裏レジスタを備え、容量外レジスタは、容量内レジスタが備えるものとは異なる一組の表レジスタ及び裏レジスタを備える。

【 0 1 4 8 】

そして、メイン制御部 4 1 は、容量内処理では、容量内レジスタを使用し、容量外処理においては、使用するレジスタを容量内レジスタから容量外レジスタに交換するとともに、容量外処理の終了時に、使用するレジスタを容量外レジスタから容量内レジスタに戻すことで、容量内レジスタは容量内処理で使用され、容量外レジスタは容量外処理で使用されることとなる。

【 0 1 4 9 】

また、共通レジスタは、容量内処理でも容量外処理でも使用可能な構成である。

30

【 0 1 5 0 】

図 1 0 に示すように、共通レジスタのうち I X レジスタ及び I Y レジスタは、容量内処理においても容量外処理においても使用可能であるが、I X レジスタ及び I Y レジスタのうち I X レジスタは、容量内処理において使用され、容量外処理において使用されない構成となっている。また、I X レジスタ及び I Y レジスタのうち I Y レジスタは、容量外処理において使用され、容量内処理において使用されない構成となっている。

【 0 1 5 1 】

このように、複数のレジスタの変形例では、複数のレジスタは、容量内処理で使用される容量内レジスタ、容量外処理で使用される容量外レジスタ、容量内処理でも容量外処理でも使用可能な共通レジスタを含み、容量内処理において容量内レジスタを使用し、容量内

40

処理から呼び出される容量外処理において容量外レジスタを使用する。

【 0 1 5 2 】

容量内処理でも容量外処理でも I X レジスタ及び I Y レジスタからなる複数のレジスタを使用可能であるが、I X レジスタ及び I Y レジスタからなる複数のレジスタのうち I X レジスタは容量内処理において使用され、容量外処理では使用されないの、容量内処理と容量外処理との間において I X レジスタに格納されるデータが混同してしまうことを防止できる。

【 0 1 5 3 】

また、容量内処理でも容量外処理でも I X レジスタ及び I Y レジスタからなる複数のレジスタを使用可能であるが、I X レジスタ及び I Y レジスタからなる複数のレジスタのうち

50

I Yレジスタは容量外処理において使用され、容量内処理では使用されないの、容量内処理と容量外処理との間においてI Yレジスタに格納されるデータが混同してしまうことを防止できる。

【0154】

尚、I Xレジスタ及びI Yレジスタからなる複数のレジスタのうちI Xレジスタを容量内処理のみで使用し、I Yレジスタを容量外処理のみで使用する構成であるが、I Xレジスタ及びI Yレジスタからなる複数のレジスタを容量内処理のみに使用する構成、またはI Xレジスタ及びI Yレジスタからなる複数のレジスタを容量外処理のみに使用する構成としても良く、このような構成とすることで、容量内処理と容量外処理との間においてI Xレジスタ及びI Yレジスタからなる複数のレジスタに格納されるデータが混同してしまうことを防止できる。 10

【0155】

[作用効果1]

本実施例では、割込処理は、メインルーチン(割込)と、メインルーチン(割込)から呼び出されて実行されるサブルーチン(割込)と、から構成されるとともに、割込処理において使用される複数のレジスタのうちI Yレジスタは、サブルーチン(割込)で使用され、メインルーチン(割込)で使用されないため、メインルーチン(割込)とサブルーチン(割込)の間でI Yレジスタに格納されるデータが混同してしまうことを防止できる。

【0156】

また、本実施例では、メインルーチン(割込)から呼び出されるサブルーチン(割込)は、複数種類のサブルーチン(割込)を含んでおり、I Yレジスタは、複数種類のサブルーチン(割込)の全てで使用されるのではなく、いずれかの種類のサブルーチン(割込)で使用される。このため、メインルーチン(割込)とI Yレジスタを使用するサブルーチン(割込)との間でI Yレジスタに格納されるデータが混同してしまうことを防止できる。 20

【0157】

また、本実施例では、割込処理は、タイマ割込が発生する毎に基本処理に割り込んで実行され、I Yレジスタは、サブルーチン(割込)で使用され、メインルーチン(割込)で使用されないため、割込処理の実行中にメインルーチン(割込)とI Yレジスタを使用するサブルーチン(割込)との間でI Yレジスタに格納されるデータが混同してしまうことを防止できる。 30

【0158】

本実施例では、割込処理は、メインルーチン(割込)と、メインルーチン(割込)から呼び出されて実行されるサブルーチン(割込)と、から構成されるとともに、割込処理において使用される複数のレジスタのうちD' E'レジスタは、メインルーチン(割込)で使用され、サブルーチン(割込)で使用されないため、メインルーチン(割込)とサブルーチン(割込)の間でD' E'レジスタに格納されるデータが混同してしまうことを防止できる。

【0159】

また、本実施例では、メインルーチン(割込)から呼び出されるサブルーチン(割込)は、複数種類のサブルーチン(割込)を含んでおり、D' E'レジスタは、いずれの種類のサブルーチン(割込)においても使用されない。このため、メインルーチン(割込)といずれの種類のサブルーチン(割込)との間においてもD' E'レジスタに格納されるデータが混同してしまうことを防止できる。 40

【0160】

また、本実施例では、割込処理は、タイマ割込が発生する毎に基本処理に割り込んで実行され、D' E'レジスタは、メインルーチン(割込)で使用され、サブルーチン(割込)で使用されないため、割込処理の実行中にメインルーチン(割込)とサブルーチン(割込)の間でD' E'レジスタに格納されるデータが混同してしまうことを防止できる。

【0161】

[作用効果2]

本実施例では、基本処理において表レジスタを使用し、基本処理に割り込んで実行される割込処理において表レジスタから交換した裏レジスタを使用する。また、基本処理が実行されるときにも、割込処理が実行されるときにも、I Xレジスタを使用可能であるが、I Xレジスタは基本処理において使用され、割込処理では使用されないの、基本処理と割込処理との間においてI Xレジスタに格納されるデータが混同してしまうことを防止できる。

【0162】

また、本実施例では、基本処理が実行されるときにも、割込処理が実行されるときにも、I Xレジスタ及びI Yレジスタからなる複数のレジスタを使用可能であるが、I Xレジスタ及びI YレジスタのうちI Yレジスタは割込処理に使用され、I Xレジスタは割込処理に使用されないため、基本処理と割込処理との間において、I Xレジスタ及びI Yレジスタからなる複数のレジスタのうち少なくともI Xレジスタに格納されるデータが混同してしまうことを防止できる。

10

【0163】

また、本実施例では、基本処理が実行されるときにも、割込処理が実行されるときにも、I Xレジスタ及びI Yレジスタからなる複数のレジスタを使用可能であるが、I Xレジスタ及びI Yレジスタのいずれも割込処理に使用しない構成としても良く、このような構成とすることで、基本処理を実行するときにも、割込処理を実行するときにも使用可能なI Xレジスタ及びI Yレジスタのいずれも割込処理では使用しないの、基本処理と割込処理との間においてI Xレジスタ及びI Yレジスタからなる複数のレジスタに格納されるデータが混同してしまうことを防止できる。

20

【0164】

本実施例では、基本処理において表レジスタを使用し、基本処理に割り込んで実行される割込処理において表レジスタから交換した裏レジスタを使用する。また、基本処理が実行されるときにも、割込処理が実行されるときにも、I Yレジスタを使用可能であるが、I Yレジスタは割込処理において使用し、基本処理では使用しないの、基本処理と割込処理との間においてI Yレジスタに格納されるデータが混同してしまうことを防止できる。

【0165】

また、本実施例では、基本処理が実行されるときにも、割込処理が実行されるときにも、I Xレジスタ及びI Yレジスタからなる複数のレジスタを使用可能であるが、I Xレジスタ及びI YレジスタのうちI Xレジスタは基本処理に使用し、I Yレジスタは基本処理に使用しないため、基本処理と割込処理との間において、I Xレジスタ及びI Yレジスタからなる複数のレジスタのうち少なくともI Yレジスタに格納されるデータが混同してしまうことを防止できる。

30

【0166】

また、本実施例では、基本処理が実行されるときにも、割込処理が実行されるときにも、I Xレジスタ及びI Yレジスタからなる複数のレジスタを使用可能であるが、I Xレジスタ及びI Yレジスタのいずれも基本処理に使用しない構成としても良く、このような構成とすることで、基本処理を実行するときにも、割込処理を実行するときにも使用可能なI Xレジスタ及びI Yレジスタからなる複数のレジスタのいずれも基本処理では使用されないの、基本処理と割込処理との間においてI Xレジスタ及びI Yレジスタからなる複数のレジスタに格納されるデータが混同してしまうことを防止できる。

40

【0167】

また、本実施例では、割込処理を実行するときに、使用するレジスタを表レジスタから裏レジスタに交換すると、前回の割込処理で使用されていた裏レジスタの値が格納された状態となるが、今回の割込処理では、前回の割込処理で使用されていた裏レジスタの値を使用せず、改めて裏レジスタに必要な値を格納するので、意図しない値で割込処理が実行されてしまうことを防止できる。

【0168】

[作用効果3]

50

本実施例では、容量内プログラムによる容量内処理と、容量内プログラムから呼び出されて実行される容量外プログラムによる容量外処理と、を実行可能であり、容量内処理を実行するときに、まず全てのレジスタに格納されている値をRAM 41cにおける容量外スタック領域に退避させ、容量外処理を終了するときに、容量外スタック領域に退避させた全ての値を対応する全てのレジスタに復帰させる。

【0169】

また、容量外処理を実行するときには、容量内処理が使用する複数のレジスタのうち一部のレジスタのみを使用する。このため、容量外処理では、容量内処理が使用する複数のレジスタのうち一部のレジスタのみを使用し、容量外処理を実行するときの方が、容量内処理を実行するときよりも使用されるレジスタの数が少ないので、容量外処理の実行による誤ったデータの使用を防止できる。

10

【0170】

また、本実施例では、基本処理において、容量内プログラムによる容量内処理（基本）と、容量外プログラムによる容量外処理（基本）と、を実行可能であり、容量内処理（基本）から容量外処理（基本）を呼び出して実行するときに、まず全てのレジスタに格納されている値をRAM 41cにおける容量外スタック領域に退避させ、容量外処理（基本）を終了するときに、容量外スタック領域に退避させた全ての値を対応する全てのレジスタに復帰させる。

【0171】

また、容量内処理（基本）を実行するときには、Aレジスタ、Fレジスタ、BCレジスタ、DEレジスタ、HLレジスタ、Qレジスタ、Iレジスタ、Rレジスタ、SPレジスタ、PCレジスタ、すなわち13のレジスタを使用するのに対し、容量外処理（基本）を実行するときには、Aレジスタ、Fレジスタ、BCレジスタ、HLレジスタ、Iレジスタ、Rレジスタ、SPレジスタ、PCレジスタ、すなわち10のレジスタを使用する。このため、容量外処理（基本）では、容量内処理（基本）が使用する複数のレジスタのうち一部のレジスタのみを使用し、容量外処理（基本）を実行するときの方が、容量内処理（基本）を実行するときよりも使用されるレジスタの数が少ないので、容量外処理（基本）の実行による誤ったデータの使用を防止できる。

20

【0172】

また、本実施例では、割込処理において、容量内プログラムによる容量内処理（割込）と、容量外プログラムによる容量外処理（割込）と、を実行可能であり、容量内処理（割込）から容量外処理（割込）を呼び出して実行するときに、まず全てのレジスタに格納されている値をRAM 41cにおける容量外スタック領域に退避させ、容量外処理（割込）を終了するときに、容量外スタック領域に退避させた全ての値を対応する全てのレジスタに復帰させる。

30

【0173】

また、容量内処理（割込）を実行するときには、A'レジスタ、F'レジスタ、B'C'レジスタ、D'E'レジスタ、H'L'レジスタ、Rレジスタ、SPレジスタ、PCレジスタ、すなわち11のレジスタを使用するのに対し、容量外処理（割込処理）を実行するときには、A'レジスタ、F'レジスタ、B'C'レジスタ、H'L'レジスタ、Rレジスタ、SPレジスタ、PCレジスタ、すなわち9のレジスタを使用する。このため、容量外処理（割込）では、容量内処理（割込）が使用する複数のレジスタのうち一部のレジスタのみを使用し、容量外処理（割込）を実行するときの方が、容量内処理（割込）を実行するときよりも使用されるレジスタの数が少ないので、容量外処理（割込）の実行による誤ったデータの使用を防止できる。

【0174】

[作用効果4]

本実施例では、容量内プログラムによる容量内処理と、容量外プログラムによる容量外処理と、を実行可能であり、容量内処理から容量外処理を呼び出して実行するときにSPレジスタに格納されている容量内スタックポインタの値を退避し、容量外スタック領域を示

50

す値に切り替えることによって、データの退避先を容量内スタック領域から容量外スタック領域に切り替える。また、容量外処理（基本処理）を終了するとき、退避していた容量内スタックポインタの値をSPレジスタに戻すことによって、データの退避先を容量外スタック領域から容量内スタック領域に切り替える。

【0175】

そして、容量外処理を終了するとき、容量外処理の実行中に容量外スタック領域に退避させたデータを全て復帰させてから、データの退避先を容量外スタック領域から容量内スタック領域に切り替えるので、前回実行した容量外処理で使用されていたデータが退避したまま、さらに容量外処理が実行されることがなく、前回実行した容量外処理で使用されたデータと、今回実行する容量外処理で使用されるデータと、が混同してしまうことを防止できる。

10

【0176】

また、容量内処理から呼び出される容量外処理は、複数種類の容量外処理を含んでおり、いずれの種類の容量外処理であっても、容量外処理を終了するとき、容量外処理の実行中に容量外スタック領域に退避させたデータを全て復帰させてから、データの退避先を容量外スタック領域から容量内スタック領域に切り替えるので、容量外処理の種類に関わらず、前回実行した容量外処理で使用されていたデータが退避したまま、さらに容量外処理が実行されることがなく、前回実行した容量外処理で使用されたデータと、今回実行する容量外処理で使用されるデータと、が混同してしまうことを防止できる。

【0177】

20

以上、本発明の実施例を図面により説明してきたが、本発明はこの実施例に限定されるものではなく、本発明の主旨を逸脱しない範囲における変更や追加があっても本発明に含まれることは言うまでもない。

【0178】

前記実施例では、本発明を遊技用価値としてメダル並びにクレジットを用いて賭数が設定されるスロットマシンに適用した例について説明したが、遊技用価値として遊技球を用いて賭数を設定するスロットマシンや、遊技用価値としてクレジットのみを使用して賭数を設定する完全クレジット式のスロットマシンに適用しても良い。

【0179】

また、前記実施例及び変形例では、本発明を遊技機の一例であるスロットマシン1に適用する例を示したが、本発明はこれに限定されるものではなく、遊技領域に遊技球を発射することにより遊技を行うパチンコ遊技機、さらには、スロットマシンやパチンコ遊技機以外の一般ゲーム機等、所定の遊技を行う遊技機であれば適用可能である。

30

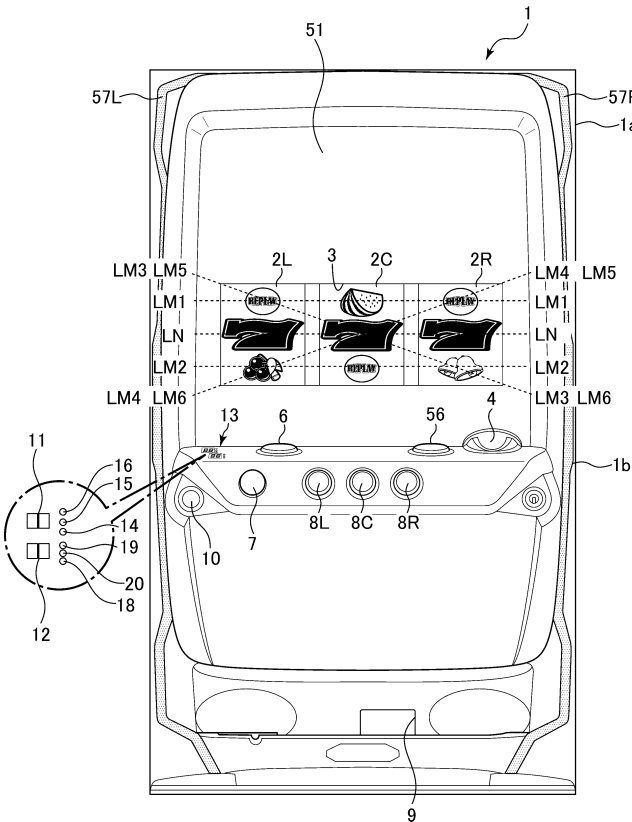
【符号の説明】

【0180】

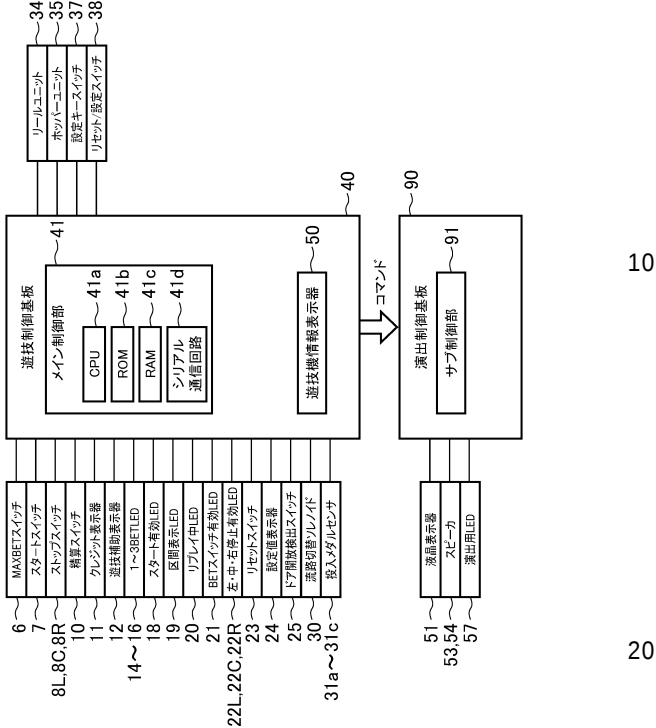
- 1 スロットマシン
- 2 L、2 C、2 R リール
- 6 MAX BETスイッチ
- 7 スタートスイッチ
- 8 L、8 C、8 R ストップスイッチ
- 41 メイン制御部
- 91 サブ制御部

40

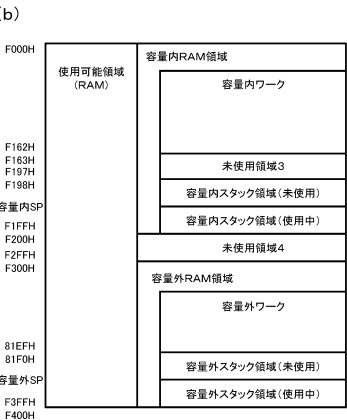
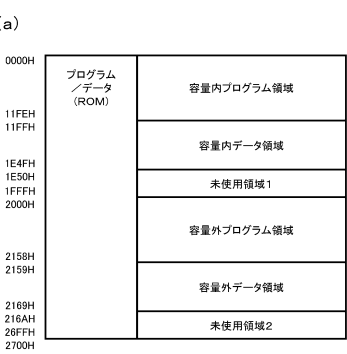
【図面】  
【図 1】  
【図1】



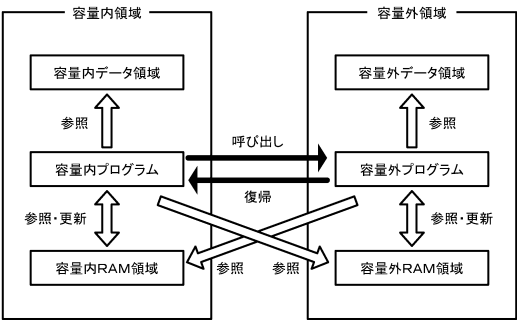
【図 2】  
【図2】



【図 3】  
【図3】

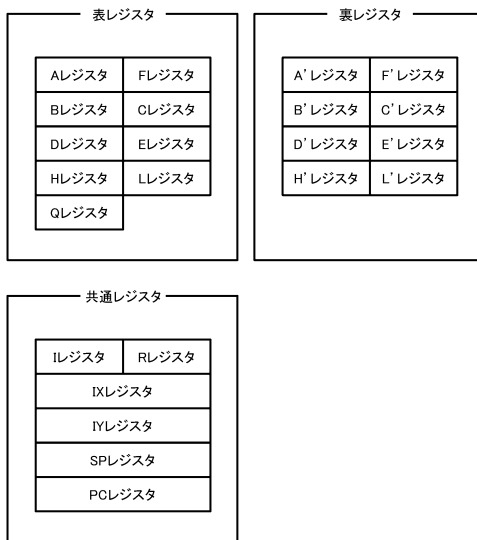


【図 4】  
【図4】



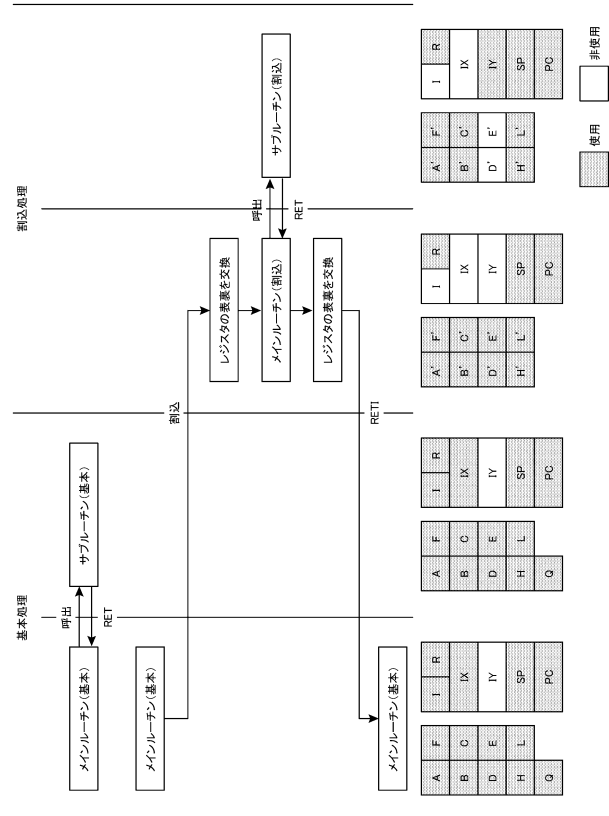
## 【 図 5 】

【図5】



【 図 6 】

【図6】

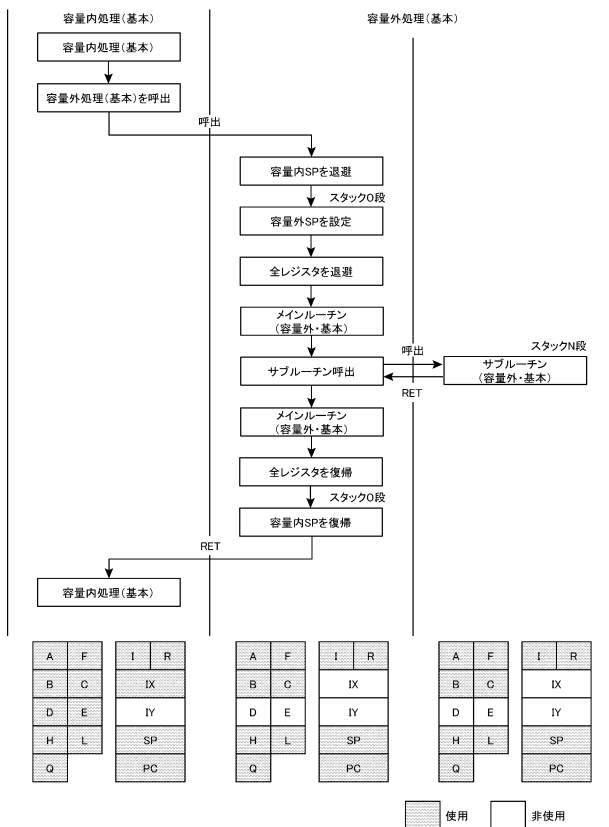


10

20

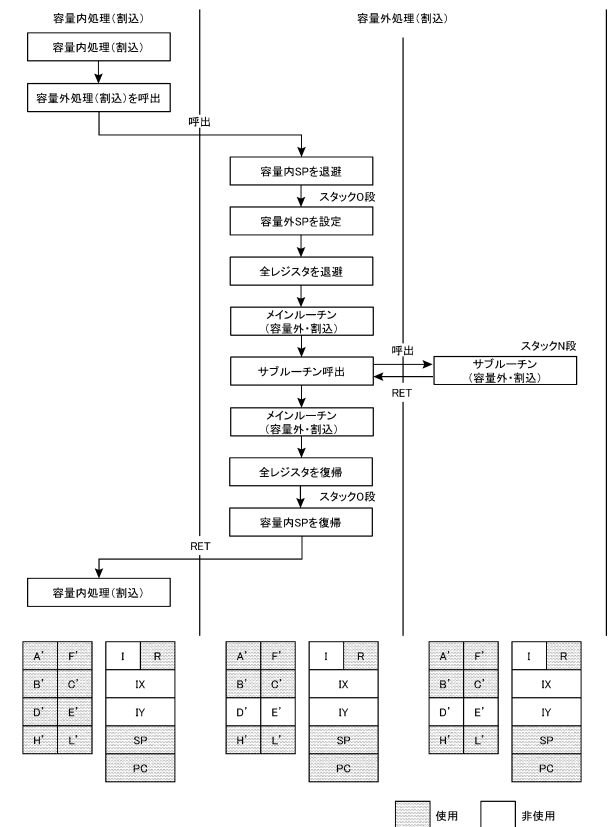
## 【圖 7】

【図7】



【 図 8 】

【図8】

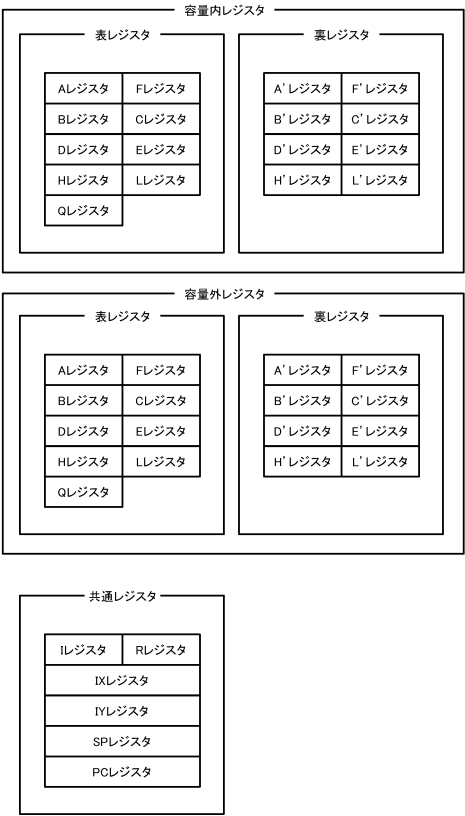


30

40

【図 9】

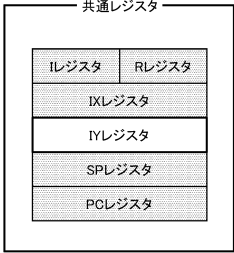
【図9】



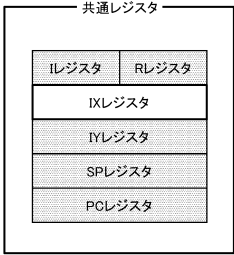
【図 1 0】

【図10】

(a)容量内処理



(b)容量外処理



使用 非使用

10

20

30

40

50