

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6535459号  
(P6535459)

(45) 発行日 令和1年6月26日 (2019.6.26)

(24) 登録日 令和1年6月7日 (2019.6.7)

(51) Int.Cl.

G05B 19/05 (2006.01)

F I

G05B 19/05

J

請求項の数 15 (全 35 頁)

(21) 出願番号	特願2014-248402 (P2014-248402)	(73) 特許権者	000129253
(22) 出願日	平成26年12月8日 (2014.12.8)		株式会社キーエンス
(65) 公開番号	特開2016-110458 (P2016-110458A)		大阪府大阪市東淀川区東中島1丁目3番1
(43) 公開日	平成28年6月20日 (2016.6.20)		4号
審査請求日	平成29年8月31日 (2017.8.31)	(74) 代理人	100076428
			弁理士 大塚 康德
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100131886
			弁理士 坂本 隆志

最終頁に続く

(54) 【発明の名称】 プログラマブル・ロジック・コントローラ、基本ユニット、制御方法およびプログラム

(57) 【特許請求の範囲】

【請求項1】

ユーザプログラムをスキャン毎に繰り返し実行する基本ユニットと、前記基本ユニットに接続される拡張ユニットとを有するプログラマブル・ロジック・コントローラであって、前記拡張ユニットは、

外部機器が接続されるインターフェースと、

前記インターフェースを介した前記外部機器からの入力値または前記インターフェースを介した前記外部機器への出力値に関するデバイス値が格納され、前記基本ユニットとデータ共有するための複数の第一デバイスが割り付けられた記憶部と、

クロックを計数して時間を計測する拡張ユニットタイマと、

を備え、

前記基本ユニットは、

クロックを計数して時間を計測する基本ユニットタイマと、

前記拡張ユニットタイマと前記基本ユニットタイマのタイマ値を同期させる同期手段と、スキャン毎のリフレッシュの期間内で前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させる第1更新手段と、

前記同期手段により前記拡張ユニットタイマのタイマ値と同期した前記基本ユニットタイマのタイマ値に基づいてスキャン中に定周期で実行される処理の期間内で前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させる第2更新手段と、

10

20

を備え、

前記プログラマブル・ロジック・コントローラは、

前記複数の第一デバイスのうち前記ユーザプログラムに記述される一つの第一デバイスに対し、該一つの第一デバイスを更新する更新手段として、前記第1更新手段および前記第2更新手段のいずれか一方を選択する選択手段

を備え、

前記選択手段により前記第1更新手段が選択されている場合には、前記一つの第一デバイスは前記第1更新手段によって更新され、前記選択手段により前記第2更新手段が選択されている場合には、前記一つの第一デバイスは前記第2更新手段によって更新されることを特徴とするプログラマブル・ロジック・コントローラ。

10

【請求項2】

前記基本ユニットは、前記ユーザプログラムと、前記同期手段により同期させた前記基本ユニットタイマのタイマ値に基づいて実行されるユニット間同期プログラムとを記憶するプログラム記憶部を備え、

前記第2更新手段は、前記一つの第一デバイスが前記ユニット間同期プログラムに記述されているときに、前記拡張ユニットが前記一つの第一デバイスに保持しているデータと前記基本ユニットが保持しているデータとを一致させることを特徴とする請求項1に記載のプログラマブル・ロジック・コントローラ。

【請求項3】

前記第2更新手段は、前記スキャン中において前記ユーザプログラムが実行されている期間であっても、前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させることを特徴とする請求項1に記載のプログラマブル・ロジック・コントローラ。

20

【請求項4】

前記拡張ユニットタイマのタイマ値は、前記拡張ユニットの内部処理の開始タイミングとして使用され、

前記拡張ユニットの内部処理は、前記外部機器からの入力値に基づいて前記デバイス値を演算する処理または前記デバイス値から前記外部機器への出力値を演算する処理であることを特徴とする請求項1ないし3のいずれか1項に記載のプログラマブル・ロジック・コントローラ。

30

【請求項5】

前記第1更新手段は、前記同期手段によって前記拡張ユニットタイマのタイマ値と同期した前記基本ユニットタイマのタイマ値に基づくスキャン毎のリフレッシュの期間内で更新を実行することを特徴とする請求項1に記載のプログラマブル・ロジック・コントローラ。

【請求項6】

前記基本ユニットはそれぞれデータを保持する記憶領域である複数の第二デバイスを有し、

前記複数の第一デバイスおよび前記複数の第二デバイスは、前記拡張ユニットのデータを前記基本ユニットへ伝達するための入力デバイスと、前記基本ユニットのデータを前記拡張ユニットに伝達するための出力デバイスとを含み、

40

前記第1更新手段および前記第2更新手段は、前記拡張ユニットの前記第一デバイスのうち前記入力デバイスに記憶されているデータであるデバイス値を前記基本ユニットの前記第二デバイスのうち前記入力デバイスへコピーすることで前記入力デバイスのデータであるデバイス値を更新するとともに、前記基本ユニットの前記第二デバイスのうち前記出力デバイスに記憶されているデータであるデバイス値を前記拡張ユニットの前記第一デバイスである前記出力デバイスへコピーすることで前記出力デバイスのデータであるデバイス値を更新することを特徴とする請求項1ないし5のいずれか1項に記載のプログラマブル・ロジック・コントローラ。

【請求項7】

50

前記入力デバイスは前記拡張ユニットが有する機能の状態を表すデータを記憶し、  
前記出力デバイスは前記拡張ユニットが有する機能を制御するデータを記憶することを特徴とする請求項 6 に記載のプログラマブル・ロジック・コントローラ。

【請求項 8】

前記リフレッシュの期間において前記第 1 更新手段が前記出力デバイスに保持されているデータと前記入力デバイスに保持されているデータを更新し、前記基本ユニットタイマのタイマ値に基づいてスキャン中に定周期で実行される処理の期間において前記第 2 更新手段が前記拡張ユニットに保持されているデータと前記基本ユニットに保持されているデータとを更新することを特徴とする請求項 6 または 7 に記載のプログラマブル・ロジック・コントローラ。

10

【請求項 9】

前記第 2 更新手段は、前記基本ユニットに設けられ、データを保持するメモリである第二デバイスと、前記拡張ユニットに設けられ、前記リフレッシュの対象とならないデータを保持するバッファメモリとの間でデータを一致させることを特徴とする請求項 1 ないし 5 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 10】

前記選択手段は、前記第 2 更新手段による更新の対象となるデータを指定する更新対象指定情報を記憶する記憶手段を有し、

前記第 2 更新手段は、前記更新対象指定情報によって指定されたデータを更新し、

前記第 1 更新手段は、前記更新対象指定情報によって指定されていないデータを前記リフレッシュの期間で更新することを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

20

【請求項 11】

前記拡張ユニットは、前記第 2 更新手段における更新処理の遅延時間に応じて当該拡張ユニットで実行される内部処理の開始を遅延させることを特徴とする請求項 1 ないし 10 のいずれか 1 項に記載のプログラマブル・ロジック・コントローラ。

【請求項 12】

前記拡張ユニットは、

前記第 2 更新手段によって更新される前記データを記憶する第 1 バッファと、

当該拡張ユニットの内部処理によって参照されるデータを記憶する第 2 バッファと、

前記同期手段が同期処理を実行するときに前記第 1 バッファに記憶されているデータと前記第 2 バッファに記憶されているデータとが一致するように更新する第 3 更新手段をさらに有し、

30

前記第 2 更新手段は、前記第 3 更新手段が更新処理を完了した後で、更新処理を実行することを特徴とする請求項 1 に記載のプログラマブル・ロジック・コントローラ。

【請求項 13】

プログラマブル・ロジック・コントローラの拡張ユニットに接続され、ユーザプログラムをスキャン毎に繰り返し実行する基本ユニットであって、

前記拡張ユニットは、

外部機器が接続されるインターフェースと、

40

前記インターフェースを介した前記外部機器からの入力値または前記インターフェースを介した前記外部機器への出力値に関するデバイス値が格納され、前記基本ユニットとデータ共有するための複数のデバイスが割り付けられた記憶部と、

クロックを計数して時間を計測する拡張ユニットタイマと、

を備えており、

前記基本ユニットは、

クロックを計数して時間を計測する基本ユニットタイマと、

前記拡張ユニットタイマと前記基本ユニットタイマのタイマ値を同期させる同期手段と、

スキャン毎のリフレッシュの期間内で前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させる第 1 更新手段と、

50

前記同期手段により前記拡張ユニットタイマのタイマ値と同期した前記基本ユニットタイマのタイマ値に基づいてスキャン中に定周期で実行される処理の期間内で前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させる第2更新手段と、

前記複数のデバイスのうち前記ユーザプログラムに記述される一つのデバイスに対し、該一つのデバイスを更新する更新手段として、前記第1更新手段および前記第2更新手段のいずれか一方を選択する選択手段と

を備え、

前記選択手段により前記第1更新手段が選択されている場合には、前記一つのデバイスは前記第1更新手段によって更新され、前記選択手段により前記第2更新手段が選択されている場合には、前記一つのデバイスは前記第2更新手段によって更新されることを特徴とする基本ユニット。

10

【請求項14】

プログラマブル・ロジック・コントローラの拡張ユニットに接続され、ユーザプログラムをスキャン毎に繰り返し実行する基本ユニットの制御方法であって、

前記拡張ユニットは、

外部機器が接続されるインターフェースと、

前記インターフェースを介した前記外部機器からの入力値または前記インターフェースを介した前記外部機器への出力値に関するデバイス値が格納され、前記基本ユニットとデータ共有するための複数のデバイスが割り付けられた記憶部と、

20

クロックを計数して時間を計測する拡張ユニットタイマと、

を備えており、

前記制御方法は、

クロックを計数して時間を計測する基本ユニットタイマと、前記拡張ユニットタイマと前記基本ユニットタイマのタイマ値を同期させる同期工程と、

スキャン毎のリフレッシュの期間内で前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させる第1更新工程と、

前記同期工程により前記拡張ユニットタイマのタイマ値と同期した前記基本ユニットタイマのタイマ値に基づいてスキャン中に定周期で実行される処理の期間内で前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させる第2更新工程と、

30

前記複数のデバイスのうち前記ユーザプログラムに記述される一つのデバイスに対し、該一つのデバイスを更新する更新工程として、前記第1更新工程および前記第2更新工程のいずれか一方を選択する選択工程と

を有し、

前記選択工程において前記第1更新工程が選択されている場合には、前記一つのデバイスは前記第1更新工程において更新され、前記選択工程において前記第2更新工程が選択されている場合には、前記一つのデバイスは前記第2更新工程において更新されることを特徴とする基本ユニットの制御方法。

40

【請求項15】

請求項14に記載の制御方法の各工程をコンピュータに実行させるプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プログラマブル・ロジック・コントローラ、基本ユニット、制御方法およびプログラムに関する。

【背景技術】

【0002】

一般にプログラマブル・ロジック・コントローラ(PLC)は基本ユニット(CPUユニットとも呼ぶ)と拡張ユニットを備えている(特許文献1)。基本ユニットと拡張ユニ

50

ットは、通常のラダープログラムが実行されるタイミングと同期した、リフレッシュ期間と呼ばれる期間においてデータの交換（いわゆるリフレッシュ通信）を行う。

【 0 0 0 3 】

ところで、基本ユニットと拡張ユニットは、ユニット間同期処理によって相互に制御の同期をとることもできる。具体的には、1台又は複数台の拡張ユニットの内部処理と、基本ユニットにおける所定の処理（ラダー実行処理など）とを同期させて実行することもできる。

【 0 0 0 4 】

一般に、ユニット間同期処理において基本ユニットと拡張ユニットの間でデータ交換を行うためには、ユーザは別途、ユニット間同期用のラダープログラムの中に、ダイレクト通信の命令語を書き込まなければならない。ダイレクト通信の命令語とは、基本ユニットがその命令語を実行したタイミングで、基本ユニットと拡張ユニットの間でデータ交換される命令語である。このダイレクト通信の命令語があることで、毎スキャンで繰り返し実行されるリフレッシュ通信のタイミングを待たなくてもデータ交換されるため、ユニット間同期処理を用いた外部機器の高速制御などのメリットが得られる。

【 先行技術文献 】

【 特許文献 】

【 0 0 0 5 】

【 特許文献 1 】 特開 2 0 1 4 - 0 5 2 6 7 2 号 公 報

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 0 6 】

しかし、ユニット間同期用のラダープログラム中にダイレクト通信の命令語を逐一記述することは、ユーザにとって煩雑な作業であり、手間が掛かるものである。例えば、ユニット間同期ではない、毎スキャンで繰り返し実行される通常のラダープログラム（いわゆる平場のラダープログラム）の記述の一部または全部をユニット間同期用のラダープログラムに流用する場合であっても、別途、ダイレクト通信の命令語を書き加えなければならず、結局のところ、プログラム変更の作業が必要になる。

【 0 0 0 7 】

そこで、毎スキャンで繰り返し実行されるリフレッシュ通信と同様に、ユニット間同期用のラダープログラムが実行されるタイミングと同期して、ユニット間同期の対象となっているデータを一括でデータ交換する方法（いわゆるユニット間同期リフレッシュ）が考えられる。このユニット間同期リフレッシュは、毎スキャン実行される通常のリフレッシュとは別概念のものであるため、ユニット間同期リフレッシュの対象となるデバイスとして、通常のリフレッシュの対象となるデバイスとは異なる専用デバイスが用意される。

【 0 0 0 8 】

しかし、基本ユニットと拡張ユニットとの間で共有すべきデータは数多く存在するため、ユニット間同期リフレッシュの期間でデータのすべてをユニット間で一致するように更新処理（リフレッシュ）を実行しようとする、上述した専用デバイスを数多く用意しなければならない。その結果、専用デバイスを用いてユニット間同期用のラダープログラムを作成することは、ユーザにとって煩雑な作業であり、手間が掛かるものである。また、上述同様、平場のラダープログラムの記述の一部または全部をユニット間同期用のラダープログラムに流用する場合には、別途、通常のリフレッシュの対象となるデバイスを、ユニット間同期リフレッシュの対象となる専用デバイスに書き換えなければならず、この方法も結局のところ、プログラムの変更作業が必要になる。

【 0 0 0 9 】

また、ユニット間同期リフレッシュの対象となる専用デバイスの数が多い場合、ユニット間同期リフレッシュの期間が長くなってしまふ。その結果、外部機器の高速制御などのユニット間同期処理のメリットが相殺されることになってしまふ。もしこのようなデータの一部を、ユニット間同期リフレッシュの期間以外の期間で実行できるように選択でき

10

20

30

40

50

ば、リフレッシュ期間の増大を抑制できるであろう。そこで、本発明は、基本ユニットと拡張ユニットとの間で共有すべきデータの更新タイミングを選択可能なデバイスを用意することで、ユニット間同期処理のメリットが低下するのを防ぎながら、プログラムの変更作業の負担を軽減してユーザビリティを向上させることを目的とする。

【課題を解決するための手段】

【0010】

本発明は、たとえば、ユーザプログラムをスキャン毎に繰り返し実行する基本ユニットと、前記基本ユニットに接続される拡張ユニットとを有するプログラマブル・ロジック・コントローラであって、前記拡張ユニットは、

10

外部機器が接続されるインターフェースと、

前記インターフェースを介した前記外部機器からの入力値または前記インターフェースを介した前記外部機器への出力値に関するデバイス値が格納され、前記基本ユニットとデータ共有するための複数の第一デバイスが割り付けられた記憶部と、

クロックを計数して時間を計測する拡張ユニットタイマと、

を備え、

前記基本ユニットは、

クロックを計数して時間を計測する基本ユニットタイマと、

前記拡張ユニットタイマと前記基本ユニットタイマのタイマ値を同期させる同期手段と、スキャン毎のリフレッシュの期間内で前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させる第1更新手段と、

20

前記同期手段により前記拡張ユニットタイマのタイマ値と同期した前記基本ユニットタイマのタイマ値に基づいてスキャン中に定周期で実行される処理の期間内で前記拡張ユニットが保持しているデータと前記基本ユニットが保持しているデータとを一致させる第2更新手段と、

を備え、

前記プログラマブル・ロジック・コントローラは、

前記複数の第一デバイスのうち前記ユーザプログラムに記述される一つの第一デバイスに対し、該一つの第一デバイスを更新する更新手段として、前記第1更新手段および前記第2更新手段のいずれか一方を選択する選択手段

30

を備え、

前記選択手段により前記第1更新手段が選択されている場合には、前記一つの第一デバイスは前記第1更新手段によって更新され、前記選択手段により前記第2更新手段が選択されている場合には、前記一つの第一デバイスは前記第2更新手段によって更新されることを特徴とするプログラマブル・ロジック・コントローラを提供する。

【発明の効果】

【0011】

本発明によればユーザビリティを向上させることが可能になる。

【図面の簡単な説明】

【0012】

40

【図1】PLCシステムの一例を示す図

【図2】ユーザプログラムの一例を示す図

【図3】プログラム作成支援装置の一例を示す図

【図4】PLCの一例を示す図

【図5】スキャンタイムを説明するための図

【図6】スキャンタイムと同期周期との関係を示す図

【図7】基本ユニットの機能を示す図

【図8】拡張ユニットの機能を示す図

【図9】プログラム作成支援装置の機能を示す図

【図10】スキャンタイムにおけるユニット間同期とリフレッシュを含む処理を示すフロ

50

ーチャート

【図 1 1】拡張ユニットで実行される同期リフレッシュを示すフローチャート

【図 1 2】同期リフレッシュと内部処理との時間的な関係を示す図

【図 1 3】ダブルバッファ構成と複数の拡張ユニット間でのデータの送受信を示す図

【図 1 4】ユニット間同期のタイミングに沿って実行されるスキャン処理を示すフローチャート

【図 1 5】ユニット間同期のタイミングに沿って実行されるスキャン処理を示すフローチャート

【図 1 6】同期リフレッシュと一括リフレッシュとのいずれかを選択するための GUI の一例を示す図

10

【発明を実施するための形態】

【0013】

以下に本発明の一実施形態を示す。以下で説明される個別の実施形態は、本発明の上位概念、中位概念および下位概念など種々の概念を理解するために役立つであろう。また、本発明の技術的範囲は、特許請求の範囲によって確定されるのであって、以下の個別の実施形態によって限定されるわけではない。

【0014】

はじめにプログラマブル・ロジック・コントローラ（PLC、単にプログラマブルコントローラと呼ばれてもよい）を当業者にとってよりよく理解できるようにするために、一般的な PLC の構成とその動作について説明する。

20

【0015】

図 1 は、本発明の実施の形態によるプログラマブル・ロジック・コントローラシステムの一構成例を示す概念図である。図 1 に示すように、このシステムは、ラダープログラムなどのユーザプログラムの編集を行うためのプログラム作成支援装置 1 と、工場等に設置される各種制御装置を統括的に制御するための PLC（プログラマブル・ロジック・コントローラ）2 とを備えている。PLC 2 は、CPU が内蔵された基本ユニット 3 と 1 つないし複数の拡張ユニット 4 を備えている。基本ユニット 3 に対して 1 つないし複数の拡張ユニット 4 が着脱可能となっている。基本ユニット 3 は CPU ユニットと呼ばれることもある。

【0016】

30

基本ユニット 3 には、表示部 5 及び操作部 6 が備えられている。表示部 5 には、基本ユニット 3 に取り付けられている各拡張ユニット 4 の動作状況などを表示することができ、表示部 5 の表示内容は、操作部 6 を操作することにより切り替えることができる。表示部 5 には、通常、PLC 2 内のデバイスの現在値（デバイス値）や PLC 2 内で生じたエラー情報などが表示される。なお、デバイスとは、デバイス値を格納するために設けられたメモリ上の領域を指す名称であり、デバイスメモリと呼ばれてもよい。デバイス値とは、入力機器からの入力状態、出力機器への出力状態およびユーザプログラム上で設定される内部リレー（補助リレー）、タイマ、カウンタ、データメモリ等の状態を示す情報である。

【0017】

40

拡張ユニット 4 は、PLC 2 の機能を拡張するために用意されており、基本ユニット 3 に対して側方から取り付けられる。1 つ目の拡張ユニット 4 は、基本ユニット 3 に対して側方から直接的に取り付けられる。2 つ目以降の拡張ユニット 4 は、既に取り付けられている拡張ユニット 4 に対して、側方から直列的に取り付けられる。たとえば、基本ユニット 3 の右側面と拡張ユニット 4 の左側面とが連結面になっている。同様に、1 つ目の拡張ユニット 4 の右側面の形状等は基本ユニット 3 の右側面とほぼ同じであるため、1 つ目の拡張ユニット 4 の右側面に 2 つ目の拡張ユニット 4 の左側面が連結される。このような連結方式は、数珠つなぎ方式とかデジチェーン方式と呼ばれてもよい。各連結面にはコネクタが設けられており、通信や電力供給を行うためのバスもコネクタを介して連結される。このようにして、基本ユニット 3 と複数の拡張ユニット 4 が直列的に取り付けられる

50

と、各拡張ユニット4内に備えられた配線(例:バス)を介して、各拡張ユニット4が基本ユニット3に対して通信可能に接続される。各拡張ユニット4には、その拡張ユニット4の機能に対応する被制御装置16(図4)が接続され、これにより、各被制御装置16が拡張ユニット4を介して基本ユニット3に接続される。被制御装置16には、センサなどの入力装置や、アクチュエータなどの出力装置が含まれる。

【0018】

プログラム作成支援装置1は、たとえば、携帯可能ないわゆるノートタイプやタブレットタイプのパーソナルコンピュータであって、表示部7及び操作部8を備えている。PLC2を制御するためのユーザプログラムの一例であるラダープログラムは、プログラム作成支援装置1を用いて作成され、その作成されたラダープログラムは、プログラム作成支援装置1内でニモニックコードに変換される。そして、プログラム作成支援装置1を、USB(Universal Serial Bus)などの通信ケーブル9を介してPLC2の基本ユニット3に接続し、ニモニックコードに変換されたラダープログラムをプログラム作成支援装置1から基本ユニット3に送ると、そのラダープログラムが基本ユニット3内でマシンコードに変換され、基本ユニット3に備えられたメモリ内に記憶される。

【0019】

なお、図1では示していないが、プログラム作成支援装置1の操作部8には、プログラム作成支援装置1に接続されたマウスなどのポインティングデバイスが含まれていてもよい。また、プログラム作成支援装置1は、USB以外の他の通信ケーブル9を介して、PLC2の基本ユニット3に対して着脱可能に接続されるような構成であってもよい。

【0020】

図2は、ラダープログラムの作成時にプログラム作成支援装置1の表示部7に表示されるラダー図17の一例を示す図である。図2に示すように、PLC2を制御するためのラダープログラムは、プログラム作成支援装置1の表示部7にマトリックス状に表示される複数のセル18内に仮想デバイスのシンボル19を適宜配置して、視覚的なリレー回路を表すラダー図17を構築することにより作成される。

【0021】

ラダー図17には、たとえば、10列×N行(Nは任意の自然数)のセル18が配置されている。そして、各行のセル18内に、図2に示す左側から右側に向かって、時系列的に仮想デバイスのシンボル19を適宜配置することにより、視覚的なリレー回路を作成することができる。作成されるリレー回路は、1行で表される直列的なリレー回路であってもよいし、複数行に並列的に表されたリレー回路を互いに結合することにより作成された、並列的なリレー回路であってもよい。

【0022】

図2に示すリレー回路は、入力装置からの入力信号に基づいてオン/オフされる3つの仮想デバイス(以下、「入力デバイス」と呼ぶ。)のシンボル19a, 19b, 19cと、出力装置の動作を制御するためにオン/オフされる仮想デバイス(以下、「出力デバイス」と呼ぶ。)のシンボル19dとが適宜結合されることにより構成されている。

【0023】

各入力デバイスのシンボル19a, 19b, 19cの上方に表示されている文字(「R0001」、「R0002」及び「R0003」)は、その入力デバイスのデバイス名(アドレス名)21を表している。各入力デバイスのシンボル19a, 19b, 19cの下方に表示されている文字(「フラグ1」、「フラグ2」及び「フラグ3」)は、その入力デバイスに対応付けられたデバイスコメント22を表している。出力デバイスのシンボル19dの上方に表示されている文字(「原点復帰」)は、その出力デバイスの機能を表す文字列からなるラベル23である。

【0024】

図2に示す例では、デバイス名「R0001」及び「R0002」にそれぞれ対応する2つの入力デバイスのシンボル19a, 19bが直列的に結合されることにより、AND回路が構成されている。また、これらの2つの入力デバイスのシンボル19a, 19bが

10

20

30

40

50



らなるAND回路に対して、デバイス名「R0003」に対応する入力デバイスのシンボル19cが並列的に結合されることにより、OR回路が構成されている。すなわち、このリレー回路では、2つのシンボル19a, 19bに対応する入力デバイスがいずれもオンした場合、又は、シンボル19cに対応する入力デバイスがオンした場合にのみ、シンボル19dに対応する出力デバイスがオンされるようになっている。

#### 【0025】

図3は、図1のプログラム作成支援装置1の電氣的構成について説明するためのブロック図である。図3に示すように、プログラム作成支援装置1には、CPU24、表示部7、操作部8、記憶装置25及び通信部26が備えられている。表示部7、操作部8、記憶装置25及び通信部26は、それぞれCPU24に対して電氣的に接続されている。記憶装置25は、少なくともRAMを含む構成であり、ラダープログラム記憶部25aと、編集ソフト記憶部25bとを備えている。

10

#### 【0026】

ユーザは、編集ソフト記憶部25bに記憶されている編集ソフトをCPU24に実行させて、操作部8を通じてラダープログラムを編集する。ここで、ラダープログラムの編集には、ラダープログラムの作成及び変更が含まれる。編集ソフトを用いて作成されたラダープログラムは、ラダープログラム記憶部25aに記憶される。また、ユーザは、必要に応じてラダープログラム記憶部25aに記憶されているラダープログラムを読み出し、そのラダープログラムを、編集ソフトを用いて変更することができる。通信部26は、通信ケーブル9を介してプログラム作成支援装置1を基本ユニット3に通信可能に接続するためのものである。

20

#### 【0027】

図4は、PLC2の電氣的構成について説明するためのブロック図である。図4に示すように、基本ユニット3には、CPU10、表示部5、操作部6、記憶装置12及び通信部14が備えられている。表示部5、操作部6、記憶装置12、及び通信部14は、それぞれCPU10に電氣的に接続されている。記憶装置12は、RAMやROM、メモ리카ードなどを含んでもよく、ラダープログラムなどを記憶する。記憶装置12には、プログラム作成支援装置1から入力されたラダープログラムやユーザデータが上書きして記憶される。また、記憶装置12には基本ユニット用の制御プログラムも格納されている。

#### 【0028】

30

図5は、本発明の実施の形態に係るプログラマブルコントローラの基本ユニット3でのスキャンタイムの構成を示す模式図である。図5が示すように1つのスキャンタイムTは、入出力のリフレッシュを行うためのユニット間通信201、プログラム実行202、END処理204により構成されている。ユニット間通信201で、基本ユニット3は、ラダープログラムを実行して得られた出力データを基本ユニット3内の記憶装置12から外部機器などに送信するとともに、受信データを含めた入力データを基本ユニット3内の記憶装置12に取り込む。たとえば、基本ユニット3のデバイスに記憶されているデバイス値はリフレッシュによって拡張ユニット4のデバイスに反映される。同様に、拡張ユニット4のデバイスに記憶されているデバイス値はリフレッシュによって基本ユニット3のデバイスに反映される。なお、リフレッシュ以外のタイミングでデバイス値をユニット間で更新する仕組みが採用されてもよい。ただし、基本ユニット3のデバイスは基本ユニット3が随時書き換えており、同様に、拡張ユニット4のデバイスは拡張ユニット4が随時書き換えている。つまり、基本ユニット3のデバイスは基本ユニット3の内部の装置によって随時アクセス可能であり、同様に、拡張ユニット4のデバイスは拡張ユニット4の内部の装置によって随時アクセス可能になっている。基本ユニット3と拡張ユニット4との間では基本的にリフレッシュのタイミングにおいて相互にデバイス値を更新して同期する。プログラム実行202で、基本ユニット3は、更新された入力データを用いてプログラムを実行(演算)する。基本ユニット3はプログラムの実行によりデータを演算処理する。なお、END処理とは、プログラム作成支援装置1や基本ユニット3に接続された表示器(図示せず)等の外部機器とのデータ通信、システムのエラーチェック等の周辺サービス

40

50

に関する処理全般を意味する。E N D 処理は周辺処理と呼ばれることもある。

【 0 0 2 9 】

このように、プログラム作成支援装置 1 はユーザの操作に応じたラダープログラムを作成し、作成したラダープログラムを P L C 2 に転送する。P L C 2 は、入出力リフレッシュ、ラダープログラムの実行および E N D 処理を 1 サイクル ( 1 スキャン ) として、このサイクルを周期的、すなわちサイクリックに繰り返し実行する。これにより、各種入力機器 ( センサ等 ) からのタイミング信号に基づいて、各種出力機器 ( モータ等 ) を制御する。よって、P L C 2 は汎用のパーソナルコンピュータ ( P C ) とは全く異なる動きをする。

【 0 0 3 0 】

< ユニット間同期 >

ユニット間同期とは、基本ユニット 3 の内部処理の制御周期と拡張ユニット 4 の内部処理の制御周期とを同期させる処理のことである。基本ユニット 3 と拡張ユニット 4 はそれぞれタイマを有し、タイマ値に応じて内部処理の開始タイミングなどを決定する。基本ユニット 3 と拡張ユニット 4 が同じ周期のタイマを有することにより、両者の制御周期が同期する。なお、基本ユニット 3 の制御周期を基準として拡張ユニット 4 の制御周期を同期させる方法と、拡張ユニット 4 の制御周期を基準として基本ユニット 3 の制御周期を同期させる方法とがある。いずれにしても拡張ユニット 4 は、拡張ユニット 4 の制御周期にしたがって内部処理を実行する。つまり、内部処理の開始タイミングは拡張ユニット 4 の制御周期の開始タイミングに一致している。なお、同期処理はユーザプログラムであるラダープログラムのスキャンタイム内における特定の期間に実行されてもよいし、ラダープログラムに対して割り込むことで実行されてもよい。ここでは説明の便宜上、一定周期の割り込みによってユニット間同期が実行されるものとして説明する。割り込みを実行するための割り込みモジュールが基本ユニット 3 に設けられていてもよい。たとえば、図 6 ( A ) が示すように、スキャンタイムが 5 ミリ秒だとすると、ユニット間同期の同期周期は 2 0 0 マイクロ秒 (  $\mu s$  ) に設定されることがある。この場合は、1 つのスキャンタイム内で 2 5 回のユニット間同期が実行されることになる。なお、スキャンはスキャンモジュールによって制御されており、これとは並行してユニット間同期は同期モジュールによって実行されている。そのため、リフレッシュの期間 ( ユニット間通信 2 0 1 )、ラダープログラムを実行している期間 ( プログラム実行 2 0 2 )、E N D 処理を実行している期間 ( E N D 処理 2 0 4 ) のいずれにおいてもユニット間同期が定周期で実行される。

【 0 0 3 1 】

上述した内部処理は内部機能が実行する処理のことである。拡張ユニット 4 はインターフェースを備え、当該インターフェースを介して外部機器である被制御装置 1 6 と接続されている。拡張ユニット 4 は外部機器から取得したデータに対して演算を実行して入力デバイス値を求め、入力デバイスに格納する。また、拡張ユニット 4 は出力デバイスのデバイス値に対して演算を実行して出力値を求め、外部機器に出力する。内部処理とはこれらの演算処理などである。なお、演算処理は、数式に基づく算出処理だけでなく、テーブルから値を取得するような処理も含まれる概念である。

【 0 0 3 2 】

< 一括リフレッシュと同期リフレッシュ >

図 5 に示したように入出力リフレッシュは 1 つのスキャンタイム T において一回だけ実行される ( これを一括リフレッシュと呼ぶことにする。 )。しかし、1 つのスキャンタイムにおいて一回だけのリフレッシュを行うだけでは、十分ではないようなデータも存在する。このようなデータについてはデバイスに割り付けずに、バッファメモリに割り付け、バッファメモリに対してダイレクト通信を行うことで、データを読み書きすることが可能となる。しかし、この場合は、ラダープログラム内でダイレクト通信を実行するための命令を記述しておかなければならない。スキャンタイム ( 例 : 5 m s ) に対してずっと短い周期 ( 例 : 2 0 0  $\mu s$  ) で更新される必要があるデータについてはラダープログラム内の多数の個所にデータの更新命令を記述しておかねばならず、ユーザビリティにかける面が

10

20

30

40

50

あった。また、「発明が解決しようとする課題」の欄に記述したような問題があった。

【0033】

ところで、上述したようにユニット間同期（ユニット間同期プログラム）は非常に短い同期周期で繰り返し実行されるように設定されることがある。よって、図6（B）が示すように、ユニット間同期の際にデータをリフレッシュすれば、リアルタイム性の要求されるデータをフレッシュな状態に維持できるようになる（これを同期リフレッシュと呼ぶことにする）。しかし、一括リフレッシュの対象となっていたすべてのデバイスについて同期リフレッシュを実行してしまうと、ユニット間同期に付随した同期リフレッシュによって平場のラダープログラムが中断される時間が長くなり（別の言い方をすると、同期リフレッシュに伴って実行されるユニット間同期プログラムの記述量が制限されることになり）、基本ユニット3が拡張ユニット4を制御する上で不都合が生じる。これは、基本ユニット3がユニット間同期のタイミングでラダープログラムに対して割り込みを発生させ、同期リフレッシュを実行する必要があるからである。なお、リフレッシュの期間（ユニット間通信201）、ラダープログラムを実行している期間（プログラム実行202）、END処理を実行している期間（END処理204）のいずれにおいてもユニット間リフレッシュが定周期の割り込みで実行される。基本ユニット3に対して多数の拡張ユニット4が接続されていれば、その接続台数に概ね比例してリフレッシュ対象のデバイスの数も増加するため、中断時間も増加する。

10

【0034】

そこで、複数あるデバイスをそれぞれ一括リフレッシュの対象とするか、同期リフレッシュの対象とするかを選択できれば、ラダープログラムが中断される時間を削減できるであろう。つまり、基本ユニット3と拡張ユニット4との間で共有すべきデータの更新タイミングを選択可能とすれば、ユーザは自己が共用する範囲の中断時間に収まるように各デバイスについて更新タイミングを選択できるようになり、ユーザビリティが向上することになる。なお、一部のデバイスは一括リフレッシュの対象として設定可能とされるだけでなく、同期リフレッシュの対象として設定可能であってもよい。

20

【0035】

なお、これまでユーザプログラムであるラダープログラム内でダイレクト通信によって更新されていたようなバッファメモリについても同期リフレッシュの対象とされてもよい。つまり、同期リフレッシュの対象として選択されたバッファメモリについては、ラダープログラム内ではダイレクト通信のコードを記述する必要がなくなる。つまり、同期リフレッシュの対象として選択されたバッファメモリについては一括リフレッシュの対象とされているデバイスと同様にラダープログラム内で扱うことが可能となる。ラダープログラムの作成が容易になる点でもユーザビリティが向上しよう。また、ラダープログラム内でダイレクト通信により更新されていたデバイスについても同様に同期リフレッシュの対象として選択することが可能となる。

30

【0036】

<基本ユニットの機能>

図7は基本ユニット3の機能の一例を示すブロック図である。CPU10は制御プログラム77を実行することで様々な機能を実現する。なお、機能の一部はASICなどの論理回路によって実装されてもよい。

40

【0037】

基本ユニット制御部70は基本ユニット3の各部を統括的に制御する機能である。たとえば、基本ユニット制御部70はスキャンの管理やユニット間リフレッシュなどのための割り込みの管理などを実行する。基本ユニット制御部70はCPU10が制御プログラム77を実行することで実現される機能である。プログラム実行部71は、主にラダープログラム78などを実行する機能である。プログラム実行部71はASICなどにより実装されると、制御の高速化を実現しやすい。デバイス制御部72は、デバイス80に対する読み書きを制御する機能である。デバイス制御部72は、たとえば、同期リフレッシュが実行されている間は、デバイス80の内容が他の手段によって書き換えられたり、読み出

50

されたりしないように、排他・調停処理（禁止処理）を実行する。デバイス制御部 72 は、第 2 更新部 76 が同期リフレッシュを実行している間は、第 2 更新部 76 以外の手段による割り込みを禁止してもよい。なお、同期リフレッシュを開始したときにすでに発生している割り込みについては禁止されずに、そのまま実行される。

#### 【0038】

ユニット間同期部 73 は、予め定められた同期周期にしたがってユニット間同期プログラムを実行する。ユニット間同期プログラムは、平場のラダープログラムに対して割り込みをかけて実行されるプログラムであり、基本的に平場のラダープログラムよりも短い周期で実行される。ユニット間同期部 73 は、拡張ユニットタイマ 119（図 8）のタイマ値と基本ユニットタイマ 88 のタイマ値とを同期させるタイマ同期を実行する。タイマ同期を実行する機能はタイマ同期部としてユニット間同期部 73 から独立して設けられていてもよい。これは拡張ユニット 4 に関しても同様である。タイマ同期の周期はたとえば 1000 マイクロ秒などである。これにより、基本ユニット 3 の制御周期に対して拡張ユニット 4 の制御周期が同期する。一般に、ユニット間同期が実行されない場合、基本ユニット 3 の制御周期と拡張ユニット 4 の制御周期が徐々にずれて行き、基本ユニット 3 が拡張ユニット 4 を精度よく制御できなくなることがある。そのため、ユニット間同期プログラムを実行することで、基本ユニット 3 の制御周期と拡張ユニット 4 の制御周期が位相レベルで一致するようになる。基本ユニットタイマ 88 はクロックを計数して時間を計測するタイマまたはカウンタである。ユニット間同期部 73 は、基本ユニットタイマ 88 のタイマ値を基準とした同期周期にしたがってユニット間同期プログラムを実行する。ユニット間同期部 73 は、通信部 14 に接続された拡張バスを介して拡張ユニット 4 に対して同期タイミングを通知するための同期信号を送信する。通信部 14 には拡張バスによる通信を行うための通信機能やプログラム作成支援装置 1 と通信するための通信機能とが含まれているものとする。

#### 【0039】

更新部 74 は、基本ユニット 3 がデバイスに記憶しているデータと拡張ユニット 4 がデバイスに記憶しているデータとが一致するように相互に更新する機能である。たとえば、基本ユニット 3 から拡張ユニット 4 に転送されるデータは出力デバイスに格納される。拡張ユニット 4 から基本ユニット 3 に転送されるデータは入力デバイスに格納される。一般に、複数の出力デバイスはまとめて出力リフレッシュによって更新され、複数の入力デバイスはまとめて入力リフレッシュによって更新される。入力リフレッシュと出力リフレッシュは一括リフレッシュと同期リフレッシュの双方に存在する。第 1 更新部 75 は一括リフレッシュを実行する機能である。つまり、第 1 更新部 75 はスキャン毎のリフレッシュの期間内で拡張ユニット 4 が保持しているデータと基本ユニット 3 が保持しているデータとを一致させる。第 2 更新部 76 は同期リフレッシュを実行する機能である。つまり、第 2 更新部 76 はユニット間同期部 73 により拡張ユニットタイマ 119 のタイマ値と同期した基本ユニットタイマ 88 のタイマ値に基づいてスキャン中に定周期で実行される処理の期間内で拡張ユニット 4 が保持しているデータと基本ユニット 3 が保持しているデータとを一致させる。なお第 2 更新部 76 はスキャンタイム内のいずれの期間であっても割り込んで同期リフレッシュを実行しうる。

#### 【0040】

ユーザプログラムであるラダープログラム 78 に記述される拡張ユニット 4 のデバイスは、第 1 更新部 75 による更新の対象と第 2 更新部 76 による更新の対象とのうち少なくともいずれかの更新対象として設定可能に構成される。基本ユニット制御部 70 は、第 1 更新部 75 による更新の対象となるデータ / デバイスや第 2 更新部 76 による更新の対象となるデータ / デバイスを選択する選択手段として機能してもよい。つまり、基本ユニット制御部 70 はデバイスを更新する更新手段として、第 1 更新部 75 および第 2 更新部 76 のうち少なくとも一方を選択する選択手段として機能してもよい。たとえば、基本ユニット制御部 70 は、操作部 6 からの指示にしたがって各デバイスを一括リフレッシュの対象とするか、それとも同期リフレッシュの対象とするかを選択し、選択結果を設定データ

79に保持する。つまり、設定データ79には、同期リフレッシュの対象となるデバイスやバッファメモリを指定する更新対象指定情報が含まれる。なお、一括リフレッシュの対象となるデバイスやバッファメモリについても更新対象指定情報として設定データ79に含まれてもよい。なお、プログラム作成支援装置1においてこのような設定データ79が作成され、ラダープログラム78とともに基本ユニット3に転送されてきてもよい。なお、同期リフレッシュの対象だけが選択可能とされ、一括リフレッシュの対象は選択不可能にされてもよい。また、一括リフレッシュの対象だけが選択可能とされ、同期リフレッシュの対象は選択不可能にされてもよい。

#### 【0041】

記憶装置12には制御プログラム77が記憶されている。制御プログラム77にはスキャンを実行するスキャンプログラムやユニット間同期を実行するユニット間同期プログラム、割り込みを実行する割り込みモジュールなどが含まれている。なお、ユニット間同期プログラムは、ユニット間同期処理によって拡張ユニットタイマ119(図8)のタイマ値と同期した基本ユニットタイマ88のタイマ値に基づいて実行される。記憶装置12はプログラム記憶部としても機能する。スキャンとはスキャンタイム内でユーザプログラムとEND処理とを周期的に繰り返し実行させることをいう。END処理には一括リフレッシュと周辺処理とが含まれる。スキャンタイムTはスキャンに要する時間であり、一定に設定されるケースと可変とされるケースがあるが、本実施形態ではそのどちらのケースにも適用可能である。

#### 【0042】

スキャン制御プログラムや同期プログラムもラダー言語により記述されていてもよい。ユーザプログラムの一種であるラダープログラム78はラダー言語により記述され、基本ユニット3の内部機能や拡張ユニット4の内部機能を制御するためのユーザプログラムである。設定データ79には基本ユニット3や拡張ユニット4の接続順序や内部機能を制御するために必要となるパラメータなどが含まれている。デバイス80は、たとえば、拡張ユニット4の内部機能に割り付けられたデータを機能する記憶領域であり、デバイスメモリと呼ばれることもある。ここではデバイス80は複数のデバイスの集合として記載されている。第1デバイス81は一括リフレッシュの対象として選択されているデバイスである。第2デバイス82は同期リフレッシュの対象として選択されているデバイスである。第1デバイス81の数は一般に第2デバイス82の数よりも多いが、これは逆であってもよい。第3デバイス83は、オプションであり、拡張ユニット4のバッファメモリUGに割り付けられたデバイスメモリである。第3デバイス83については、第1更新部75が一括リフレッシュしてもよいし、第2更新部76が同期リフレッシュしてもよい。第3デバイス83とバッファメモリUGの割り付け関係や、どちらのリフレッシュが適用されるかを示す情報についても設定データ79に保持されている。複数の第3デバイス83が設けられていてもよい。どちらのリフレッシュが適用されるかに応じて、更新部74では、第3デバイス83が第1デバイス81として扱われたり、第2デバイス82として扱われたりすることになる。

#### 【0043】

##### < 拡張ユニットの機能 >

図8は拡張ユニットの機能の一例を示すブロック図である。CPU110は記憶装置111に記憶されているシステムプログラム117を実行することで様々な機能を実現する。拡張ユニット制御部113は、第1デバイス118や第2デバイス120に書き込まれる基本ユニット3からの指示に応じて被制御装置16を制御する機能である。記憶装置111は、ROMやRAMなどを有し、記憶装置111における一部のメモリは書き換え可能な不揮発性メモリであってもよい。CPU110は被制御装置16と通信するためのインターフェース123を有している。ユニット間同期部114は、拡張バスおよび通信部112を介して基本ユニット3から送信される同期信号にしたがって拡張ユニット4における内部処理の制御周期を基本ユニット3の制御周期に同期させるものである。拡張ユニットタイマ119はクロックを計数して時間を計測するタイマまたはカウンタである。ユニ

ット間同期部 1 1 4 は、拡張ユニットタイマ 1 1 9 のタイマ値を基本ユニットタイマ 8 8 のタイマ値に一致させるように動作してもよい。なお、ユニット間同期部 1 1 4 は、第 2 更新部 7 6 における更新処理の遅延時間に応じて拡張ユニット 4 で実行される内部処理の開始を遅延（延期）させる延期部 1 1 5 を備えていてもよい。更新処理が完了していないにもかかわらず、拡張ユニット 4 が内部処理を開始してしまうと、最新のデバイス値が内部処理に反映されなくなってしまう。そこで、更新処理が完了するまで内部処理の開始タイミングを延期することで、内部処理は最新のデバイス値を利用できるようになる。

#### 【 0 0 4 4 】

記憶装置 1 1 1 にはデバイス値を格納するデバイスやダイレクト通信の対象となるバッファなどの記憶領域が確保される。第 1 デバイス 1 1 8 は、一括リフレッシュされるデバイスである。第 1 デバイス 1 1 8 のデバイス値は、基本ユニット 3 の第 1 デバイス 8 1 と一致するデバイス値が格納される。第 2 デバイス 1 2 0 は、同期リフレッシュされるデバイスである。第 2 デバイス 1 2 0 のデバイス値は、基本ユニット 3 の第 2 デバイス 8 2 と一致するデバイス値が格納される。なお、第 1 デバイス 8 1 や第 2 デバイス 1 2 0 は複数のバッファにより実現されてもよい。第 1 バッファ 1 2 1 は、基本ユニット 3 とデバイス値を交換するためのバッファである。第 2 バッファ 1 2 2 は、拡張ユニット制御部 1 1 3 がデバイス値を入力したり、出力したりするためのバッファである。バッファ更新部 1 1 6 はオプションのユニットであり、1 つのデバイスを複数のバッファにより実現するときに必要となる。第 2 デバイス 1 2 0 が出力デバイスである場合、バッファ更新部 1 1 6 は、基本ユニット 3 の第 2 デバイス 8 2 のデバイス値を第 1 バッファ 1 2 1 に格納し、第 1 バッファ 1 2 1 に格納されたデバイス値をさらに第 2 バッファ 1 2 2 に格納する。第 2 デバイス 1 2 0 が入力デバイスである場合、バッファ更新部 1 1 6 は、内部処理の結果であるデバイス値を第 2 バッファ 1 2 2 に格納し、第 2 バッファ 1 2 2 に格納されたデバイス値を第 1 バッファ 1 2 1 に格納し、第 1 バッファ 1 2 1 に格納されたデバイス値を基本ユニット 3 の第 2 デバイス 8 2 に転送する。このように第 1 バッファ 1 2 1 は同期リフレッシュの際に参照され、第 2 バッファ 1 2 1 は拡張ユニット 4 の内部処理によって参照される。このようなダブルバッファ構成を採用することで、第 2 デバイス 1 2 0 のデバイス値が確定してから、リフレッシュや内部処理に利用されるようになる。つまり、リフレッシュの途中でデバイス値が書き換えられてしまったり、内部処理の途中でデバイス値が書き換えられてしまったりすることが起きにくくなる。デバイス値には、一緒のタイミングで更新される必要がある複数のデバイス値の集合や（同時性）、更新の順序が決まっている複数のデバイス値の集合が存在する（順序性）。また、基本ユニット 3 と拡張ユニット 4 との間での伝達遅延時間がそれぞれほぼ同じになることを要求される複数のデバイス値の集合も存在する（伝達遅延時間の一定性）。たとえば、1 つ目のデバイスには分母が格納され、2 つ目のデバイスには分子が格納されており、この分母と分子とによって制御値が決定されることがある。このような制御値において、拡張ユニット制御部 1 1 3 が分母を取得した後で、分子が同期リフレッシュによって書き換えられてしまうと、分母と分子のペアが破たんし（いわゆる泣き別れ現象）、制御値が異常となる。よって、分母と分子のペアのように同時性を要求されるデバイスに関してはダブルバッファ構成が有効である。バッファ更新部 1 1 6 は、同時性を要求される複数のデバイス値が揃った後で第 1 バッファ 1 2 1 から第 2 バッファ 1 2 2 へ、または、第 2 バッファ 1 2 2 から第 1 バッファ 1 2 1 へバッファ更新を行うことで、複数のデバイス値の泣き別れを抑制できるようになる。なお、ダブルバッファ構成は順序性や伝達遅延時間の一定性を確保するためにも役立つ。バッファメモリ U G は、通常はダイレクト通信によってアクセスされるメモリである。本実施形態では、複数のバッファメモリ U G の 1 つ以上について第 3 デバイス 8 3 に割り付けられうる。第 3 デバイス 8 3 に割り付けられたバッファメモリ U G は一括リフレッシュされたり、同期リフレッシュされたりするため、ダイレクト通信による更新が不要となる。

#### 【 0 0 4 5 】

< プログラム作成支援装置の機能 >

10

20

30

40

50

図9はプログラム作成支援装置1のCPU24が編集ソフトウェア86を実行することで実現する機能を示している。編集ソフトウェア86は、ラダープログラム78や設定データ79などをユーザが編集するために使用するプログラムである。

【0046】

CPU24は編集ソフトウェア86を実行することで様々な手段として機能するが、ここでは代表的なものについて説明する。ラダープログラム作成部87は表示部7にラダープログラム78を作成および編集するためのユーザインタフェースを表示し、操作部8から入力される指示にしたがってラダープログラム78を作成し、記憶装置25に書き込む。設定データ作成部84は、設定データ79を作成するためのユーザインタフェースを表示部7に表示し、操作部8から入力される指示にしたがって設定データ79を作成し、記憶装置25に書き込む。たとえば、設定データ作成部84は、基本ユニット3に接続される複数の拡張ユニット4のデバイスを列挙したリストを表示部7に表示し、デバイスごと一括リフレッシュの対象とするか、同期リフレッシュの対象となるかを選択する。なお、1つの拡張ユニット4あたりのデバイスの数が少ない場合、設定データ作成部84は、拡張ユニット4を単位として、一括リフレッシュの対象とするか、同期リフレッシュの対象となるかを選択してもよい。たとえば、設定データ作成部84は、基本ユニット3に接続される複数の拡張ユニット4を列挙したリストを表示部7に表示し、拡張ユニット4ごと一括リフレッシュの対象とするか、同期リフレッシュの対象となるかを選択する。このように設定データ79には、各デバイスごと一括リフレッシュの対象とするか、同期リフレッシュの対象となるかを示す情報が含まれていてもよい。プロジェクト転送部85は、ラダープログラム78および設定データ79などをプロジェクトデータとしてまとめ、通信部26を介して基本ユニット3に転送する。通信部26は、基本ユニット3と通信するためのUSBインターフェースやLANインターフェースを有している。

【0047】

<フローチャート>

図10は基本ユニット3が実行する処理を示すフローチャートである。基本ユニット3がメンテナンスモード(プログラミングモード)からランモードに切り替えられると、CPU10は制御プログラム77にしたがって本フローチャートに係る処理を実行する。なお、本実施形態ではランモードにおいてユニット間同期処理が実行されるが、メンテナンスモードにおいてもユニット間同期処理が実行されるようにしてもよい。

【0048】

S1でCPU10(第1更新部75)は一括リフレッシュを実行する。たとえば、第1更新部75は複数あるデバイスのうち設定データ79によって一括リフレッシュの対象として選択されているデバイスについて一括リフレッシュを実行する。

【0049】

S2でCPU10(プログラム実行部71)はユーザプログラムであるラダープログラム78を開始する。以降、プログラム実行部71は、ラダープログラム78が終了するまで、制御プログラム77と並列にラダープログラム78を実行する。

【0050】

S3でCPU10(ユニット間同期部73)はユニット間同期をトリガーするイベントが発生したかどうかを判定する。このイベントは割り込みモジュール(ユニット間同期部73または基本ユニット制御部70)などによって所定期間ごとに発生するものとする。たとえば、ユニット間同期部73は基本ユニットタイマ88のタイマ値が所定値増えるごとにユニット間同期を実行してもよい。ユニット間同期の通知は信号線などを通じて拡張ユニット4にも発生する。ユニット間同期のイベントが発生するとS4に進む。ユニット間同期のイベントが発生していなければ、ラダープログラム78が継続され、S8に進む。

【0051】

S4でCPU10(ユニット間同期部73、プログラム実行部71)は、基本ユニットタイマ88のタイマ値に基づき定周期で発生する同期リフレッシュのための割り込みによ

ってラダープログラムを中断する。たとえば、ユニット間同期部 7 3 は基本ユニットタイマ 8 8 のタイマ値に基づきプログラム実行部 7 1 にラダープログラムを中断するよう指示する。これによりプログラム実行部 7 1 はラダープログラムを中断する。なお、割り込み自体が中断の指示として認識されてもよい。

【 0 0 5 2 】

S 5 で C P U 1 0 ( 第 2 更新部 7 6 ) は同期リフレッシュを実行する。第 2 更新部 7 6 は複数あるデバイスのうち設定データ 7 9 によって同期リフレッシュの対象として選択されているデバイスについて一括リフレッシュを実行する。

【 0 0 5 3 】

S 6 で C P U 1 0 ( ユニット間同期部 7 3 ) はユニット間同期プログラムを実行する。基本ユニット 3 のユニット間同期部 7 3 と拡張ユニット 4 のユニット間同期部 1 1 4 は拡張バスを通じて通信し、相互にタイマ値を一致させることで双方の制御周期を同期させる。

10

【 0 0 5 4 】

S 7 で C P U 1 0 ( ユニット間同期部 7 3 、プログラム実行部 7 1 ) はラダープログラム 7 8 を再開する。たとえば、ユニット間同期部 7 3 はプログラム実行部 7 1 にラダープログラムを再開するよう指示する。これによりプログラム実行部 7 1 はラダープログラムを再開する。

【 0 0 5 5 】

S 8 で C P U 1 0 ( 基本ユニット制御部 7 0 ) はラダープログラム 7 8 が終了したかどうかを判定する。ラダープログラム 7 8 が終了していなければ、S 3 に戻る。ラダープログラム 7 8 が終了したのであれば、S 9 に進む。

20

【 0 0 5 6 】

S 9 で C P U 1 0 ( 基本ユニット制御部 7 0 ) は上述した E N D 処理を実行する。これにより 1 つのキャンタイムが終了する。その後、次のスキャンを開始するために、S 1 に戻る。以降、ランモードからメンテナンスモードに切り替わるまで、図 5 に示したように、スキャン T にしたがって本フローチャートに係る処理が繰り返し実行される。

【 0 0 5 7 】

ここではラダープログラム 7 8 の実行中にユニット間同期が実行されるものとして説明したが、一括リフレッシュ ( S 1 ) 中や E N D 処理 ( S 9 ) 中のいずれにおいてもユニット間同期は実行される。つまり、S 1 および S 9 には S 3 ないし S 7 に相当するステップが含まれている。ここで、S 4、S 7 に記載のラダープログラムは一括リフレッシュと E N D 処理に読み替えられるものとする。

30

【 0 0 5 8 】

図 1 1 は拡張ユニット 4 が実行する処理を示すフローチャートである。ここでは一括リフレッシュについては省略している。なお、C P U 1 1 0 はシステムプログラム 1 1 7 にしたがって以下の処理を実行する。

【 0 0 5 9 】

S 1 1 で C P U 1 1 0 ( ユニット間同期部 1 1 4 ) はユニット間同期の通知が発生するのを待つ。拡張バスを通じてユニット間同期の通知を受信すると、S 1 2 に進む。

40

【 0 0 6 0 】

S 1 2 で C P U 1 1 0 ( ユニット間同期部 1 1 4 、バッファ更新部 1 1 6 ) は同期リフレッシュを実行する。バッファ更新部 1 1 6 は、第 2 デバイス 1 2 0 のうち入力デバイスに格納されているデバイス値を基本ユニット 3 に転送し、基本ユニット 3 から受信したデバイス値を出力デバイスに格納する。なお、ダブルバッファ構成が採用される場合、バッファ更新部 1 1 6 は、第 1 バッファ 1 2 1 と第 2 バッファ間でデバイス値の更新を行う。出力デバイスに関しては、基本ユニット 3 から受信したデバイス値が第 1 バッファ 1 2 1 に格納され、その後、第 1 バッファ 1 2 1 から第 2 バッファ 1 2 2 にコピーされる。拡張ユニット制御部 1 1 3 によって実行される内部処理は第 2 バッファ 1 2 2 に格納されているデバイス値を参照する。入力デバイスに関しては、拡張ユニット制御部 1 1 3 が実行す

50



る内部処理によって第2バッファ122に格納されたデバイス値が第1バッファ121にコピーされ、その後、第1バッファ121から基本ユニット3に転送される。

【0061】

S13でCPU110(拡張ユニット制御部113)は第1デバイス118や第2デバイス120のうち入力デバイスからデバイス値を読み出し、デバイス値にしたがって内部処理を実行する。拡張ユニット制御部113は内部処理の実行結果を第1デバイス118や第2デバイス120のうち出力デバイス(とりわけ、出力デバイスの第2バッファ122)に格納する。

【0062】

このようにユニット間同期のタイミングで同期リフレッシュを実行することで、一括リフレッシュを前提とした通信用プログラム(基本ユニットが拡張ユニットと通信するためのプログラム)を同期リフレッシュの対象となったデバイスについても適用できるようになる。従来技術では、一括リフレッシュの対象となっていなかったデバイスなどについて任意のタイミングで最新のデバイス値が必要になるときは、平場のラダープログラム(メインのラダープログラム)内でダイレクト通信によってデバイス値を更新する必要があった。しかし、これはユーザにとって面倒なプログラミング作業であった。一方で、本実施例によれば、一括リフレッシュの対象となっていなかったデバイスについても同期リフレッシュによって更新されるため、ダイレクト通信のためのコードが不要となり、一括リフレッシュを前提とした通信用プログラムを同期リフレッシュに関しても流用できるようになる。

【0063】

< バッファメモリへの同期リフレッシュの適用 >

なお、PLC2ではデバイスとは別にバッファメモリUGと呼ばれる記憶領域が存在する。デバイスはリフレッシュの対象となる記憶領域であり、バッファメモリUG(単にバッファと呼ばれることもある)はリフレッシュの対象とならず、ダイレクト通信によりアクセスされていた。このようなバッファメモリUGについても同期リフレッシュの対象として選択されてもよい。これにより、デバイスと比較してより複雑なコードが必要となっていたバッファメモリUGについても、同期リフレッシュにより、デバイスと同様なコードにより扱うことが可能となる。とりわけ、複数のバッファメモリUGに格納される値(バッファ値と呼ぶことにする)には、一緒のタイミングで更新される必要がある複数のバッファ値の集合や(同時性)、更新の順序が決まっている複数のバッファ値の集合が存在する(順序性)。たとえば、1つ目のバッファメモリには分母が格納され、2つ目のデバイスには分子が格納されており、この分母と分子とによって制御値が決定されることがある。このような制御値において、拡張ユニット制御部113が分母を取得した後で、分子が同期リフレッシュによって書き換えられてしまうと、分母と分子のペアが破たんし(泣き別れ)、制御値が異常となる。よって、分母と分子のペアのように同時性を要求されるバッファ値に関しては同期リフレッシュにおいて同時性や順序性を維持しつつ、更新される。また、伝達遅延時間の一定性が要求されるバッファメモリUGについても、第3デバイス83を割り付けて同期リフレッシュが適用されてもよい。これにより伝達遅延時間の一定性が確保されよう。なお、バッファメモリUGについてもダブルバッファ構成が採用されてもよい。なお、同期リフレッシュの対象となるバッファメモリUGおよび第3デバイス83については明細書の全体において第2デバイスと同様に扱われるものとする。つまり、第2デバイスに関する説明はそのままバッファメモリUGと第3デバイス83に適用される。また、バッファメモリUGおよび第3デバイス83が一括リフレッシュの対象として選択された場合、第1デバイスに関する説明はそのままバッファメモリUGと第3デバイス83に適用される。

【0064】

< 同期リフレッシュの対象となるデバイスの選択 >

上述したようにユニット間同期はリフレッシュタイミングよりも短い周期(例: 50  $\mu$ s ~ 5 ms)で実行されることがある。これは、基本ユニット3と拡張ユニット4との応

10

20

30

40

50

答性（別の言い方をすれば、外部入力機器に対する基本ユニットの応答性、又は、基本ユニットに対する外部出力機器の応答性）のバラツキを小さくするためである。ユニット間同期を実行するタイミングで同期リフレッシュを実行するには、ユニット間同期の同期周期内で同期リフレッシュを完了しなければならない。よって、同期リフレッシュの対象となるデバイスの数はある程度制限される必要があるだろう。たとえば、一括リフレッシュの対象となっていたすべてのデバイスを同期リフレッシュの対象として選択してしまうと、ユニット間同期の周期を短くできなくなり、基本ユニット3と拡張ユニット4との応答性のバラツキを小さくすることが困難となろう。そこで、ユニット間同期の同期周期の長さを考慮して同期リフレッシュの対象が予め選択される。この選択は、プログラム作成支援装置1で実行されたり、または、基本ユニット3の操作部6を通じて実行されたりする。また、同期リフレッシュを割り込みプログラムにより実現し、選択されたデバイスについて同期リフレッシュが実行される。これは、スキャンタイムが長くなったりするなど、変動したりしても、安定的にユニット間同期と同期リフレッシュとを実現する。この場合、ユニット間同期は、スキャンタイム内のいずれの場所でも実行されうる。たとえば、ユニット間同期部73および第2更新部76は、ラダープログラムの実行期間だけでなく、一括リフレッシュの期間でもユニット間同期と同期リフレッシュを実行してもよい。定周期でユニット間同期と同期リフレッシュが実行されるようになり、基本ユニット3と拡張ユニット4との応答性のバラツキが小さくならう。

#### 【0065】

なお、プログラム作成支援装置1の設定データ作成部84は、同期リフレッシュの対象となるデバイスやバッファメモリUGを選択するが、そもそもユニット間同期の対象となる拡張ユニット4を選択してもよい。これにより、ユニット間同期の対象となる拡張ユニット4の数が多く、ユニット間同期リフレッシュの対象になるデータの量が多い場合であっても、ユニット同期を実現しつつ、ユニット間同期の同期周期（例えば200 $\mu$ s）を短くすることができる。図16に例示するように、設定データ作成部84は、基本ユニット3に対して接続される複数の拡張ユニット4を列挙したリスト90などを表示し、操作部8から入力された指示に応じて、複数の拡張ユニット4のそれぞれについてユニット間同期の対象とするか否かを選択してもよい。たとえば、基本ユニット3と拡張ユニット4との応答性のバラツキを小さくする必要がある位置決めユニット（モーションユニット）についてはユニット間同期の対象として選択されてもよい。LAN通信ユニットなどはそれほどリアルタイム性を要求されないため、ユニット間同期の対象として選択されなくてもよい。なお、設定データ作成部84は、ユニット間同期に対応していない拡張ユニット4については強制的にユニット間同期の対象外を選択し、ユーザに対して選択の余地を与えなくてもよい。たとえば、設定データ作成部84はそのような拡張ユニットについてはグレーアウト表示し、操作部8による選択を不能にしてもよい。

#### 【0066】

設定データ作成部84は、ユニット間同期の対象として選択された拡張ユニットについては、さらに、占有デバイスなどの特定のデバイスのリフレッシュを一括リフレッシュの対象とするか、それとも同期リフレッシュの対象とするかを選択するためのGUIを表示部7に表示してもよい。占有デバイスとは、拡張ユニットごとに予め割り付けられたデバイスのことである。たとえば、基本ユニット3との間での通信量が多い拡張ユニット4についてはその占有デバイスを同期リフレッシュの対象とすることで、一括リフレッシュ用のプログラムコードを同期リフレッシュ用に流用することが容易になる。なお、設定データ作成部84は、ユニット間同期の周期を短くすることが必要な拡張ユニット4の占有デバイスについてはあえて同期リフレッシュの対象外として選択してもよい。このようなデバイスについてはユニット間同期のタイミングで同期リフレッシュを行わないため、同期周期を短くすることが可能となる。

#### 【0067】

< 複数のデバイス値の泣き別れの抑制 >

上述したように分子と分母とにより定まる比率などを正しく伝達するには、分子と分母

10

20

30

40

50

とが同時性の要件を満たしてユニット間でリフレッシュされる必要がある。これを実現する手法としてすでにダブルバッファ構成について説明したが、他の排他調停方法が使用されてもよい。

【 0 0 6 8 】

上述したように、拡張ユニット 4 の内部処理は複数の出力デバイスの値を参照する。また、拡張ユニット 4 の内部処理は複数の処理結果を入力デバイス値に格納する。複数の出力デバイスの値には同時性を要求されるものがある。同様に、複数の入力デバイス値にも同時性を要求されるものがある。

【 0 0 6 9 】

そこで、図 1 2 が示すように、基本ユニット 3 の第 2 更新部 7 6 がこれのデバイスについて同期リフレッシュを完了するまで待ってから、拡張ユニット 4 は内部処理を開始する。つまり、図 1 2 に示すように、第 1 拡張ユニットおよび第 2 拡張ユニットのユニット間同期部 1 1 4 は、ユニット間同期によって同期リフレッシュのための割り込みが発生すると所定の待ち時間（リフレッシュ待ち時間と呼ばれてもよい）が経過するのを待ってから、拡張ユニット制御部 1 1 3 に内部処理を開始させる。なお、この待ち時間の間に、基本ユニット 3 の第 2 更新部 7 6 がデバイスやバッファメモリ U G の同期リフレッシュを実行する。これにより、デバイス値が同期リフレッシュされている期間においては、拡張ユニット制御部 1 1 3 の内部処理がデバイス値を読み込んだり、書き込んだりしないようになり、泣き別れが発生しないようになる。このように拡張ユニット 4 のユニット間同期部 1 1 4 が同期リフレッシュと内部処理とを同期させることで、同期リフレッシュによるデバイスへのアクセスと、内部処理によるそのデバイスへのアクセスとが同時には発生しないようになる。つまり、拡張ユニット制御部 1 1 3 は待ち時間中のデバイスへのアクセスが排他または調停される。この排他・調停機能は、ユニット間同期部 1 1 4 に内蔵されてもよいが、拡張ユニット制御部 1 1 3 に内蔵されてもよい。排他・調停機能（禁止機能）は、同期周期の開始タイミングから所定の待ち時間を計時するタイマやカウンタを備えていてもよい。

【 0 0 7 0 】

なお、所定の待ち時間内に同期リフレッシュが完了しない場合や、そもそも所定の待ち時間を設けない場合などでは、同期リフレッシュが完了する前に内部処理が開始されてしまうことがある。この場合は、上述したダブルバッファ構成が有効である。つまり、同期リフレッシュは第 1 バッファ 1 2 1 に対して実行され、拡張ユニット 4 の内部処理による第 1 バッファ 1 2 1 へのアクセスは排他または調停される。つまり、同期リフレッシュによるアクセスよりも後に発生したアクセスは同期リフレッシュによるアクセスが完了するまで待たされることになる。同様に、拡張ユニット 4 の内部処理は第 2 バッファ 1 2 2 にアクセスし、同期リフレッシュによる第 2 バッファ 1 2 2 へのアクセスは排他または調停される。つまり、内部処理によるアクセスよりも後に発生した同期リフレッシュによるアクセスは内部処理によるアクセスが完了するまで待たされることになる。第 1 バッファ 1 2 1 と第 2 バッファ 1 2 2 のデバイス値の更新はバッファ更新部 1 1 6 によってユニット間同期タイミング（同期周期の開始タイミング）で実行される。これによって、複数のデバイス値の泣き別れが抑制される。上述したように、バッファメモリ U G についてもダブルバッファ構成が採用されてもよい。

【 0 0 7 1 】

< デバイス値の伝達遅延時間 >

デバイス値の伝達遅延時間については一定時間に制御されることが望ましいことがある。上述したようにユニット間同期は、基本ユニット 3 と拡張ユニット 4 との間における応答性のバラツキを小さくするために実行される。しかし、基本ユニット 3 の制御周期と拡張ユニット 4 の制御周期とを同期させるだけでは不十分なことがある。つまり、基本ユニット 3 と拡張ユニット 4 との間におけるデバイス値の伝達遅延時間が一定になれば、さらに応答性のバラツキが小さくなる。上述したように複数のデバイス値の泣き別れは、これらのデバイス値それぞれの伝達遅延時間が一致しないために発生することが多い。特に、

これらのデバイス値に対する内部処理によるアクセスが非同期で発生することが原因となりやすい。

【 0 0 7 2 】

そこで、上述したように同期リフレッシュの実行タイミングと内部処理の実行タイミングとを同期させることで、デバイスに対する非同期アクセスが抑制され、伝達遅延時間のバラツキが小さくなる。なお、伝達遅延時間のバラツキを小さくするためには、上述したダブルバッファ構成も有効である。あるデバイスに対する同期リフレッシュによるアクセスと、拡張ユニット4の内部処理によるアクセスとが競合しなくなるため、基本ユニット3と内部処理による非同期アクセスが生じないようになる。よって、伝達遅延時間のバラツキが小さくなる。

10

【 0 0 7 3 】

図13はダブルバッファの動作を説明する図である。ここでは基本ユニット3に対して入力ユニットである第1拡張ユニットと出力ユニットである第2拡張ユニットが接続されているものとする。同期周期の開始時点では、第1拡張ユニットの第1バッファ121にはデバイス値としてX0が格納されており、第2バッファ122にはX1が格納されている。同様に、第2拡張ユニットの第1バッファ121にはデバイス値としてY0が格納されており、第2バッファ122にもY0が格納されている。

【 0 0 7 4 】

ユニット間同期のための割り込みが発生すると、第1拡張ユニットのバッファ更新部116は第2バッファ122のデバイス値を第1バッファ121にコピーする。なお、出力ユニットである第2拡張ユニットはこの時点でバッファ間のコピーを実行しなくてもよい。基本ユニット3の第2更新部76はまず、同期リフレッシュのうち入力リフレッシュを実行する。つまり、第2更新部76は第1拡張ユニットの第1バッファ121からデバイス値を読み出し、第1拡張ユニットに割り付けられた第2デバイスに82コピーする。

20

【 0 0 7 5 】

次に、第2更新部76は、同期リフレッシュのうち出力リフレッシュを実行する。つまり、第2更新部76は第2拡張ユニットの第1バッファ121からデバイス値を読み出し、第2拡張ユニットに割り付けられた第2デバイス82にコピーする。

【 0 0 7 6 】

割り込みが発生したタイミングから所定時間が経過すると、基本ユニット3のユニット間同期部73と第1拡張ユニットおよび第2拡張ユニットのユニット間同期部114がお互いの制御周期を位相レベルで一致させる。この同期処理が完了した時点、またはこの同期処理の開始時点に、第1拡張ユニットおよび第2拡張ユニットの拡張ユニット制御部113は、内部処理を開始する。図13では、同期処理の開始時点で内部処理が開始されるように制御されている。なお、第2拡張ユニットのバッファ更新部116は、内部処理の開始タイミングで第1バッファ121のデバイス値を第2バッファ122にコピーする。

30

【 0 0 7 7 】

その後、第1拡張ユニットの拡張ユニット制御部113は内部処理の結果に応じて第2バッファ122のデバイス値を更新する。第2拡張ユニットの拡張ユニット制御部113は第2バッファ122のデバイス値を参照し、デバイス値に応じて内部処理を実行する。

40

【 0 0 7 8 】

なお、入力リフレッシュを実行し、次に、ユニット間同期(ユニット間同期プログラム)や内部処理を実行し、最後に出力リフレッシュを実行することもある。この場合、第1拡張ユニットと第2拡張ユニットとの間でデバイス値を伝達すると、2つの同期周期が必要になる。つまり、1つ目の同期周期において第1拡張ユニットから基本ユニット3にデバイス値を伝達し、その後、内部処理が実行されて、基本ユニット3からデバイス値が第2拡張ユニットに伝達される。第2拡張ユニットは次の同期周期においてデバイス値を参照し、内部処理を実行する。よって、2つの同期周期が必要になる。一方で、図13に示すように、入力リフレッシュを実行し、次に出力リフレッシュを実行し、最後にユニット間同期や内部処理を実行すると、1つの同期周期の遅延で済むといった利点がある

50

。つまり、1つ目の同期周期において第1拡張ユニットから基本ユニット3にデバイス値が伝達され、その後、基本ユニット3からデバイス値が第2拡張ユニットに伝達され、そのデバイス値が内部処理によって参照可能となる。よって、1つの同期周期内で、第1拡張ユニットと第2拡張ユニットとがデバイス値を送受信できるようになる。ただし、内部処理は割り込みが発生したタイミングから所定の待ち時間が経過するまで待ってから実行される必要がある。

#### 【0079】

第1拡張ユニットと第2拡張ユニットとが通信する必要がある事例としては位置決めユニットと、位置決めユニットにトリガーを与える入力ユニットとを含むPLC2が考えられる。より具体的には、入力ユニットのセンサでワーク（検査対象物）を検知すると、検知したことを示すデバイス値が基本ユニット3に伝達され、さらに位置決めユニットに伝達される。位置決めユニットはこのデバイス値に基づいてモータなどを起動し、位置決めを実行する。

#### 【0080】

このような複数の拡張ユニット間でのデバイス値の通信における遅延を短くするには、入力リフレッシュと出力リフレッシュを実行してからユニット間同期を実行する方が有利であろう。とりわけ、位置決めユニットのように高速動作が要求されるユニットが関与する、拡張ユニット間でのデバイス値の通信では、このような処理の順番が向いている。

#### 【0081】

##### <ビットデバイスとワードデバイスのリフレッシュ>

デバイスやバッファメモリには1ビットのデバイス値を記憶するものと、1ワードのデバイス値を記憶するものがある。つまり、リフレッシュの対象となるものにはビット型のものやワード型のものが混在している。このような場合にも上述した順序性が必要となることがある。順序性の要件を満たす方法としては、デバイス値の送信側が送信すべきデバイス値をデバイスに格納すると、フラグをセットする。伝送路においては、デバイス値が先に伝送され、その後でフラグが伝送される。受信側は、フラグがセットされたことを認識すると、デバイスからデバイス値を読み出す。これにより、受信側はデバイス値を少ない遅延時間でもって読み出せるようになる。なお、フラグはビット型のデバイスで実現されうる。つまり、ワードデバイスに格納された伝送すべきワードデータ（ワードデバイス値）を先に転送し、フラグを格納したビットデバイスのデバイス値を後から転送することが要求されることがある。もし、フラグを先に転送してしまうと、古いワードデバイス値が読み出されてしまうことがあるからである。

#### 【0082】

ユニット間同期よりも高い優先度の割り込みが同期リフレッシュの実行中に発生することがある。この場合、入力デバイスに関してはワードデバイスが先に更新され、その後でビットデバイスが更新される。出力デバイスに関しても同様に順序性が要求される。この場合は、ビットデバイスが先に転送され、ワードデバイスが後に転送される。なお、基本ユニット3の基本ユニット制御部70や拡張ユニット4の拡張ユニット制御部113は、同期リフレッシュが実行されている期間での他の手段による割り込みを禁止してもよい。

#### 【0083】

なお、フラグとなるデバイスにダブルバッファ構成を適用してもよい。この場合、上述したビットデバイスとワードデバイスの更新順序が逆になる。ここでは、ビットデバイスとワードデバイスについて説明したが、ワードデバイスをビットデバイスとして使用するケースでも本実施例は適用可能である。つまり、本実施例は、ワードデバイスしか存在しないケースでもワードデバイス間の更新順序を維持するためにも使用可能である。

#### 【0084】

##### <内部処理の延期>

拡張ユニット4のユニット間同期部114は延期部115を備えていてもよい。同期リフレッシュが完了していないにもかかわらず、内部処理が開始されてしまうと、上述した泣き別れ問題が発生しうる。そこで、ユニット間同期部114は同期リフレッシュが所定

10

20

30

40

50

の待ち時間で完了しなかったことを検知すると、内部処理の発生タイミングを延期させてもよい。なお、延期時間は動的に決定されてもよいし、静的に決定されていてもよい。前者では、ユニット間同期部 114 が同期リフレッシュの完了を検知すると、内部処理を開始させる。後者では、ユニット間同期部 114 が所定の延期時間の経過を待って、内部処理を開始させる。

#### 【0085】

同期リフレッシュの遅延は、フロー制御に伴うリトライなどが原因で発生することがある。このような場合に延期部 115 は有用であるが、代わりに上述したダブルバッファ構成が採用されてもよい。ダブルバッファ構成を採用することで、同時性の問題や順序性の問題、さらには伝達遅延時間のバラツキ問題も緩和される。

10

#### 【0086】

＜一括リフレッシュやスキャンタイムを拡張ユニットの制御周期に同期させること＞

上述した実施例はユニット間同期により基本ユニット 3 の制御周期と拡張ユニット 4 の制御周期とが整合されるが、一括リフレッシュやスキャンタイムはこれらとは非同期で実施されてもよい。しかし、ユニット間同期を実行するためには割り込みが必要となる。もし、拡張ユニット 4 の制御周期に対して、一括リフレッシュやスキャンタイムも同期させることができれば、割り込みが不要となろう。

#### 【0087】

図 14 は一括リフレッシュやスキャンタイムを拡張ユニットの制御周期に同期させる処理を示すフローチャートである。この処理は基本ユニット 3 の CPU 10 で実行される。

20

#### 【0088】

S15 で CPU 10 (第 1 更新部 75) は一括入力リフレッシュを実行する。一括入力リフレッシュとは、入力デバイスについての上述した一括リフレッシュである。S16 で CPU 10 (プログラム実行部 71) はラダープログラム 78 を実行する。S17 で CPU 10 (第 1 更新部 75) は一括出力リフレッシュを実行する。一括出力リフレッシュとは、出力デバイスについての上述した一括リフレッシュである。S18 で CPU 10 (基本ユニット制御部 70) は END 処理を実行する。END 処理は周辺処理とも呼ばれ、外部インターフェースからのデータの取得やロギング処理によるデータの収集などである。END 処理も同期周期に連動して実行されるようになる。S19 で CPU 10 (ユニット間同期部 73) はユニット間同期を実行するタイミングになったかどうかを判定する。ユニット間同期タイミングが到来すると、ユニット間同期部 73 は、ユニット間同期の対象として指定されている拡張ユニット 4 のユニット間同期部 114 と同期処理を実行し、お互いの制御周期を一致させる。その後、S15 に戻る。つまり、ユニット間同期によって、基本ユニット 3 と拡張ユニット 4 の制御周期が一致するとともに、拡張ユニット 4 の制御周期とスキャンタイミングも同期することになる。

30

#### 【0089】

このようにユニット間同期をトリガーとしてラダープログラム 78 のスキャンが実行されるため、スキャンタイムは拡張ユニット 4 の制御周期に対して同期することになる。なお、図 14 に示してフローチャートからわかるように、同期周期とスキャンタイムはほぼ同じ長さとなる。

40

#### 【0090】

図 15 は一括リフレッシュやスキャンタイムを拡張ユニットの制御周期に同期させる他の処理を示すフローチャートである。この処理は基本ユニット 3 の CPU 10 で実行される。図 14 と比較すると、同期リフレッシュも実行される点が特に異なっている。

#### 【0091】

S20 で CPU 10 (第 2 更新部 76) は同期リフレッシュを実行する。S21 で CPU 10 (第 1 更新部 75) は一括リフレッシュを実行する。一括リフレッシュには一括入力リフレッシュと一括出力リフレッシュとが含まれている。S22 で CPU 10 (プログラム実行部 71) はラダープログラム 78 を実行する。S23 で CPU 10 (基本ユニット制御部 70) は END 処理を実行する。S24 で CPU 10 (ユニット間同期部 73)

50

は基本ユニットタイマ 8 8 のタイマ値に基づき、ユニット間同期を実行するタイミングになったかどうかを判定する。ユニット間同期タイミングが到来すると、ユニット間同期部 7 3 は、ユニット間同期の対象として指定されている拡張ユニット 4 のユニット間同期部 1 1 4 と同期処理を実行し、お互いの制御周期を一致させる。その後、S 2 0 に戻る。このようにユニット間同期に連動して同期リフレッシュと一括リフレッシュとが実行されてもよい。これにより割り込みプログラムを使用することなく、一括リフレッシュやスキャンタイムを拡張ユニットの制御周期に同期させることが可能となる。

#### 【 0 0 9 2 】

なお、スキャンタイムは固定長のものと可変長のものとがある。プログラム作成支援装置 1 の設定データ作成部 8 4 は、固定長のスキャンタイムにしたがってラダープログラム 7 8 を実行することをユーザによって指示されたときは、さらに、スキャンタイムをユニット間同期によって同期させるか、同期させないかをユーザの指示にしたがって選択してもよい。どちらが選択されたかは設定データ 7 9 に格納される。基本ユニット 3 の CPU 1 0 は設定データ 7 9 を参照することで、どちらが選択されたかを認識し、選択された動作モードにしたがってスキャンを実行する。なお、スキャンタイムをユニット間同期によって同期させることが選択された場合、設定データ作成部 8 4 は、ユーザ指示にしたがって同期周期を設定してもよい。スキャンタイムをユニット間同期によって同期させることが選択されていない場合、設定データ作成部 8 4 は、ユーザ指示にしたがってスキャンタイムを設定してもよい。

#### 【 0 0 9 3 】

##### < バッファメモリの同期リフレッシュ >

すでに所々で触れたように、デバイスだけでなく、バッファメモリ U G についても同期リフレッシュが適用されてもよい。一般にデバイスはリフレッシュにより更新されるが、バッファメモリ U G は更新されない。そのため、従来は、バッファメモリ U G に関しては、必要に応じてラダープログラム 7 8 内でダイレクト通信を実行するための命令が記述される必要があった。なお、基本ユニット 3 は複数の占有デバイスを有しており、そのうちの幾つかが拡張ユニット 4 に対して割り付けられる。よって、拡張ユニット 4 のデバイスに対して割り付けられていない基本ユニット 3 のデバイス（第 3 デバイス 8 3 ）については、バッファメモリ U G を割り付けることでバッファメモリ U G もリフレッシュすることが可能となる。これにより、バッファメモリ U G についても同期リフレッシュの対象となる。

#### 【 0 0 9 4 】

プログラム作成支援装置 1 の設定データ作成部 8 4 はユーザ指示に応じてバッファメモリ U G に対応する第 3 デバイス 8 3 を選択する。たとえば、設定データ作成部 8 4 は、拡張ユニットごとの設定データを作成するために、デバイスのリストを表示部 7 に表示する。このリストには、各デバイスの識別情報、たとえば、デバイス番号（例：DM 3 1 0 0 0 など）が列挙される。また、各デバイスの隣の欄には、対応するバッファメモリ U G を指定するための欄が設けられてもよい。この欄には、バッファメモリ U G の識別情報（例：UG 2 1 5 8 など）が操作部 8 から入力される。なお、設定データ作成部 8 4 はバッファメモリ U G に対応する第 3 デバイス 8 3 を一括リフレッシュの対象とするか、同期リフレッシュの対象とするかを、ユーザ指示に応じて選択する。設定データ作成部 8 4 はユーザによって選択されたリフレッシュの方式についても設定データ 7 9 に書き込む。

#### 【 0 0 9 5 】

このように、バッファメモリ U G が第 3 デバイス 8 3 に関連付けられるため、第 2 更新部 7 6 は設定データ 7 9 にしたがって、バッファメモリ U G も第 2 デバイスとして同期リフレッシュを実行する。また、一括リフレッシュの対象として選択されたバッファメモリ U G については第 1 デバイスとして第 1 更新部 7 5 が一括リフレッシュを実行する。

#### 【 0 0 9 6 】

バッファメモリ U G に関しても上述した同期リフレッシュを実行することで同時性や順序性、伝達遅延時間の一定性などの問題が解決ないしは緩和される。

## 【 0 0 9 7 】

占有デバイスの同期リフレッシュとバッファメモリUGの同期リフレッシュが同期周期内のどのタイミングで実行されるかについて説明する。たとえば、第2更新部76は、入力デバイスについて同期リフレッシュを実行し、次に、入力バッファメモリについて同期リフレッシュを実行する。第2更新部76は、出力バッファメモリについて同期リフレッシュを実行し、次に、出力デバイスについて同期リフレッシュを実行する。このように入力系が先にリフレッシュされ、その後に、出力系がリフレッシュされてもよい。占有デバイスが先にリフレッシュされ、その後にバッファメモリがリフレッシュされてもよい。第2更新部76は、入力デバイスについて同期リフレッシュを実行し、次に、出力デバイスについて同期リフレッシュを実行する。さらに、第2更新部76は、出力バッファメモリについて同期リフレッシュを実行し、次に入力バッファメモリについて同期リフレッシュを実行する。最後に、ユニット間同期部73がユニット間同期を実行する。なお、上述したように拡張ユニット間での通信に関しては後者の事例の方が有利であろう。

10

## 【 0 0 9 8 】

ところでバッファメモリUGについてダイレクト通信で更新しようとする、泣き別れ問題が発生することがあった。これはバッファメモリUGのサイズが32ビットなどに制限されているときに発生しうる。そこで、上述したダブルバッファ構成などの泣き別れ対策をバッファメモリUGにも適用することで、泣き別れ問題が緩和される。

## 【 0 0 9 9 】

< まとめ >

20

図1などを用いて説明したように、PLC2は、ユーザプログラムをスキャン毎に繰り返し実行する基本ユニット3と、基本ユニット3に接続される拡張ユニット4とを有する。図8などを用いて説明したように、拡張ユニット4は、外部機器が接続されるインターフェース123と、インターフェース123を介した外部機器からの入力値またはインターフェース123を介した外部機器への出力値に関するデバイス値が格納され、基本ユニット3とデータ共有するためのデバイスが割り付けられた記憶部である記憶装置111と、クロックを計数して時間を計測する拡張ユニットタイマ119を有している。図7などを用いて説明したように、基本ユニット3は基本ユニットタイマ88、ユニット間同期部73、第1更新部75および第2更新部76を有している。ユニット間同期部73は拡張ユニットタイマ119のタイマ値と基本ユニットタイマ88のタイマ値を同期させる。第1更新部75はスキャン毎のリフレッシュの期間内で拡張ユニット4が保持しているデータと基本ユニット3が保持しているデータとを一致させる。第2更新部76はユニット間同期部73により拡張ユニットタイマ119のタイマ値と同期した基本ユニットタイマ88のタイマ値に基づいてスキャン中に定周期で実行される処理の期間内で拡張ユニット4が保持しているデータと基本ユニット3が保持しているデータとを一致させる。なお、ユーザプログラムであるラダープログラム78に記述される拡張ユニット4に割り付けられたデバイスは、第1更新部75による更新の対象および/または第2更新部76による更新の対象として設定可能に構成される。これによりユーザのプログラミングの負担が軽減され、ユーザビリティが向上する。

30

## 【 0 1 0 0 】

40

見方を変えれば、本実施例では、一括リフレッシュの対象に設定可能なデバイスや同期リフレッシュの対象に設定可能なデバイス、あるいは、一括リフレッシュの対象にも同期リフレッシュの対象にも設定可能なデバイスが用意されたことになる。その結果、平場のラダープログラム78の記述をユニット間同期プログラムにコピーした場合であっても、平場のラダープログラム78に記述されたデバイスを、ユニット間同期リフレッシュの対象に設定変更するだけで、リフレッシュが実行される。よって、プログラムの変更作業が軽減される。また、ユニット間同期リフレッシュの対象に含めるかどうかはユーザが自由に設定可能であるため、一部のデバイスをユニット間同期リフレッシュの対象から外すことで(図16には拡張ユニットに割り付けられた複数のデバイスを一括して対象に設定したり、対象外に設定したりする例が示されている。)、ユニット間同期リフレッシュ期間

50



の増大を抑制することができる。

【 0 1 0 1 】

このように、デバイスは、第 1 更新部 7 5 による更新の対象として設定されてもよいし、第 2 更新部 7 6 による更新の対象として設定されてもよい。基本ユニット制御部 7 0 や設定データ作成部 8 4 は、デバイスを更新する手段として第 1 更新部 7 5 と第 2 更新部 7 6 とのうち少なくとも一方を選択する選択手段として機能してもよい。この更新部の選択は一括リフレッシュの対象とするか、それとも同期リフレッシュの対象とするかの選択に相当し、更新タイミングの選択にも相当しよう。このように基本ユニット 3 と拡張ユニット 4 との間で共有すべきデータの更新タイミングを選択可能とすることで、ユーザビリティを向上させることが可能となる。つまり、一括リフレッシュ期間だけでなく、ユーザプログラムが実行されている期間などの他の期間において同期リフレッシュが実行されるようになる。その結果、一括リフレッシュ期間が長くなりすぎるといったことは起きにくくなる。なお、いくつかのデバイスについては同期リフレッシュと一括リフレッシュの双方が適用されてもよい。

10

【 0 1 0 2 】

基本ユニット 3 は、基本ユニット 3 が有する内部機能の制御タイミングと拡張ユニット 4 の制御タイミングとを同期させる同期処理を実行するユニット間同期部 7 3 をさらに有している。ユニット間同期部 7 3 は、例えばタイマ等によって実現される。第 2 更新部 7 6 は、ユーザプログラムが実行されている期間内でユニット間同期部 7 3 が同期処理を実行するタイミングになると、拡張ユニット 4 が保持しているデータと基本ユニット 3 が保持しているデータとを一致させるよう同期リフレッシュを実行する。このようにユニット間同期のタイミングで同期リフレッシュが実行されるようになる。ユーザプログラム内でデバイスやバッファメモリのダイレクト通信を行うことも可能であるが、これはプログラムコード（ラダー図）の流用性が乏しく、また複雑なプログラミングが必要になる。一方で、ユニット間同期のタイミングで同期リフレッシュを実行すれば、一括リフレッシュ用のプログラムコードと同様の簡易なコードによってデバイスやバッファメモリの同期リフレッシュが可能となる。よって、ユーザのプログラミングに関する負担が軽減されよう。

20

【 0 1 0 3 】

図 6 や図 1 0 などを用いて説明したように、ユニット間同期部 7 3 は、ユニット間同期のタイミングが到来するとユニット間同期を実行する。これによりユーザプログラムの実行期間中であってもユニット間同期が実行されるため、基本ユニット 3 の制御タイミングと拡張ユニット 4 の制御タイミングとのずれを小さくすることが可能となる。また、ユーザプログラムの実行期間中においてユニット間同期に付随して同期リフレッシュが実行されるようになり、一括リフレッシュの期間を待つことなく、デバイスやバッファメモリを更新できるようになる。

30

【 0 1 0 4 】

図 1 4 や図 1 5 を用いて説明したように、基本ユニット 3 は、ユニット間同期をトリガー（トリガ）としてスキャンを実行してもよい。つまり、基本ユニット 3 は、拡張ユニット 4 に適用されている制御周期に対して第 1 更新部 7 5 が更新を実行するタイミングを同期させるユニット間同期部 7 3 をさらに有していてもよい。ユニット間同期部 7 3 は基本ユニット 3 の制御周期に対して拡張ユニット 4 に適用されている制御周期を同期させるものである。しかし、ユニット間同期部 7 3 は拡張ユニット 4 に適用されている制御周期に対して第 1 更新部 7 5 が更新を実行するタイミングを同期させるように動作してもよい。拡張ユニット 4 の内部処理は拡張ユニット 4 の制御周期に同期して実行される。また、デバイスやバッファメモリに格納されるデータは内部処理の動作に必要なものであったり、内部処理の結果であったりする。拡張ユニット 4 の制御周期にしたがってデータをリフレッシュすれば、データの伝達遅延時間が削減されよう。

40

【 0 1 0 5 】

基本ユニット 3 および拡張ユニット 4 はそれぞれデータを保持するメモリであるデバイスメモリを有し、デバイスメモリは、拡張ユニット 4 のデータを基本ユニット 3 へ伝達す

50

るための入力デバイスメモリと、基本ユニット3のデータを拡張ユニット4に伝達するための出力デバイスメモリとを含む。第1更新部75および第2更新部76は、拡張ユニット4のデバイスメモリに記憶されているデータを基本ユニット3のデバイスメモリへコピーすることで入力デバイスメモリのデータを更新するとともに、基本ユニット3のデバイスメモリに記憶されているデータを拡張ユニット4のデバイスメモリへコピーすることで出力デバイスメモリのデータを更新する。これにより入力デバイスや出力デバイスについて一括リフレッシュと同期リフレッシュとが実行されるようになる。このように第1更新部75および第2更新部76のリフレッシュ動作はよく似ているため、一括リフレッシュ用のコードを同期用のリフレッシュ用のコードとして流用しやすくなる。

#### 【0106】

10

入力デバイスメモリは拡張ユニット4が有する機能の状態を表すデータを記憶するものである。たとえば、拡張ユニット4の被制御装置16がセンサなどであれば、センサの検知結果(検知状態)を示すデバイス値が入力デバイスメモリに格納されることになる。出力デバイスメモリは拡張ユニット4が有する機能の制御するデータを記憶するものである。たとえば、拡張ユニット4の被制御装置16がモータなどであれば、そのモータの回転数などを示すデバイス値が出力デバイスメモリに格納されるようになる。

#### 【0107】

図6などを用いて説明したように、一括リフレッシュの期間において第1更新部75が出力デバイスメモリに保持されているデータを更新する。次に、第1更新部75が入力デバイスメモリに保持されているデータを更新する。その次にユーザプログラムが実行されている期間内において第2更新部76がデータを更新してもよい。出力デバイスメモリには内部処理によって参照されるデータや命令が格納されているため、一括リフレッシュの期間においては出力デバイスメモリが先にリフレッシュされてもよい。もちろん、入力デバイスメモリが先にリフレッシュされてもよい。

20

#### 【0108】

第2更新部76は、基本ユニット3に設けられ、データを保持するメモリであるデバイスメモリと、拡張ユニット4に設けられ、一括リフレッシュの対象とならないデータを保持するバッファメモリとの間でデータを一致させてもよい。このようにデバイスとバッファメモリとを対応付けることで、バッファメモリについてもあたかもデバイスであるかのようにリフレッシュ可能となる。

30

#### 【0109】

基本ユニット制御部70は、第2更新部76による更新の対象となるデータを指定する更新対象指定情報(設定データ79)を記憶する記憶装置12を有していてもよい。第2更新部76は、設定データ79によって指定されたデータをユーザプログラムが実行されている期間内で同期リフレッシュする。第1更新部75は、設定データ79によって指定されていないデータを一括リフレッシュの期間で一括リフレッシュする。このようにデバイスやバッファメモリについて適用されるリフレッシュのタイミングは設定データ79によって保持されてもよい。

#### 【0110】

設定データ作成部84に関して説明したように、設定データ79は、基本ユニット3に接続され、ユーザプログラムを作成して当該基本ユニット3に転送するプログラム作成支援装置上で作成された情報であってもよい。プログラム作成支援装置1の操作部8は基本ユニット3の操作部6よりも操作がしやすい。また、プログラム作成支援装置1の表示部7は基本ユニット3の表示部5よりも表示可能な情報量が多い。よって、デバイスやバッファメモリについて適用されるリフレッシュのタイミングを選択しやすくなるであろう。

40

#### 【0111】

第2更新部76がデータを更新している間は第2更新部76以外の他の手段による当該データの読み書きを排他または調停する排他・調停手段として基本ユニット制御部70やデバイス制御部72が機能してもよい。上述したように同期リフレッシュの実行中にデータが変更されてしまうと、同時性の問題が発生しうる。よって、他の手段によるデータの

50

読み書きを排他または調停することで、同時性の問題を解決できるようになる。

【0112】

デバイス制御部72は、相互に更新の連動性を要求される複数のデータが第2更新部76によって更新されている期間においては第2更新部76以外の他の手段による当該複数のデータの読み書きを排他または調停してもよい。これにより同時性の問題が解決しよう。

【0113】

デバイス制御部72は、それぞれ基本ユニット3と拡張ユニット4との間におけるデータの伝達時間（伝達遅延時間）をほぼ一定にすることが要求される複数のデータが第2更新部76によって更新されている期間においては第2更新部76以外の他の手段による当該複数のデータの読み書きを排他または調停してもよい。これにより複数のデバイス値についての伝達遅延時間のバラツキが小さくならう。

【0114】

デバイス制御部72は、それぞれ更新順序が定められている複数のデータが第2更新部76によって更新されている期間においては第2更新部76以外の他の手段による当該複数のデータの読み書きを排他または調停してもよい。たとえば、複数のデータは、ビット型のデータとワード型のデータであってもよい。第2更新部76は、基本ユニット3のビット型のデータとワード型のデータを拡張ユニット4へコピーするときはビット型のデータをコピーしてからワード型のデータをコピーしてもよい。また、第2更新部76は、拡張ユニット4のビット型のデータとワード型のデータを基本ユニット3へコピーするときはワード型のデータをコピーしてからビット型のデータをコピーしてもよい。これにより上述した順序性の問題が解決するであろう。なお、ワード型のデータの一部がビット型のデータとして使用されるケースにも本実施例は適用可能である。

【0115】

拡張ユニット4は、第2更新部76における更新処理の遅延時間に応じて当該拡張ユニット4で実行される内部処理の開始を遅延させてもよい。同期リフレッシュの対象となるデバイスやバッファメモリの数が多くなったり、フロー制御によるデータの再送が多く発生したりすると、同期リフレッシュの時間が長くなる。その結果、所定の待ち時間を過ぎても同期リフレッシュが完了しなくなってしまう。同期リフレッシュが完了していないにもかかわらず、内部処理を開始してしまえば、上述した同時性の問題が発生しう。よって、このよう場合は内部処理を遅延させて、同時性等の問題を解決してもよい。

【0116】

デバイス制御部72は、第2更新部76がデータを更新している間は割り込み処理を禁止してもよい。同期リフレッシュの最中に割り込みが入ってしまうと、同時性の問題や伝達遅延時間のバラツキの問題などが発生しやすくなる。同期リフレッシュの最中は割り込みを禁止することで、同時性の問題や伝達遅延時間のバラツキの問題などが緩和されよう。

【0117】

拡張ユニット4は、基本ユニット3に設けられている記憶装置12にデータを保持していてもよい。一般にデバイスやバッファメモリは基本ユニット3と拡張ユニット4との双方に設けられるために、リフレッシュが必要となる。しかし、拡張ユニット4の第1デバイス118や第2デバイス120が基本ユニット3の記憶装置12に設けられ、記憶装置12内でリフレッシュが実行されてもよい。これは、拡張ユニット4のデバイスメモリやバッファメモリがどこにあるかは問題とならないことを意味する。このように、基本ユニット3のデバイスメモリやバッファメモリと拡張ユニット4のデバイスメモリやバッファメモリとが同一の記憶装置内に配置されていたとしても、上述した実施例は適用可能である。

【0118】

ところで、基本ユニット3の記憶装置12は、ユーザプログラム（ラダープログラム78）と、拡張ユニット4の内部処理の実行タイミングと同期して実行される同期プログラ

ム（制御プログラム 77 の一部）とを記憶するプログラム記憶手段として機能してもよい。これらはプログラム実行手段である CPU 10 によって実行される。とりわけ、ラダープログラム 78 についてプログラム実行部 71 で実行される。第 1 デバイス 81 や第 2 デバイス 82 はユーザプログラムにおいて拡張ユニット 4 の内部処理に割り付けられている第 1 デバイスメモリの一例である。

#### 【0119】

PLC 2 の基本ユニット 3 または拡張ユニット 4 は、第 1 デバイスメモリに記憶されているデータと同一のデータを記憶するように制御される第 2 デバイスメモリを有している。第 2 デバイスメモリは、たとえば、記憶装置 12 に確保され、拡張ユニット 4 によってアクセスされる。または、第 2 デバイスメモリは、拡張ユニット 4 の記憶装置 111 に確保された第 1 デバイス 118 や第 2 デバイス 120 であってもよい。第 1 更新部 75 や第 2 更新部 76 は第 1 デバイスメモリに記憶されているデータと第 2 デバイスメモリに記憶されているデータとが一致するように更新処理を実行する更新手段として機能する。ただし、第 1 更新部 75 や第 2 更新部 76 は拡張ユニット 4 の CPU 110 によって実現されていてもよい。拡張ユニット 4 のユニット間同期部 114 は自己または拡張ユニット制御部 113 が管理する同期周期にしたがって基本ユニット 3 に割り込みを実行し、同期リフレッシュとユニット間同期を実行してもよい。なお、プログラム作成支援装置 1 の設定データ作成部 84 や基本ユニット 3 の操作部 6 などは、一括リフレッシュの期間と同期リフレッシュの期間との中のいずれの期間で更新処理を実行するかを設定する設定手段として機能する。設定データ 79 は、拡張ユニット 4 からアクセス可能な記憶装置 12、記憶装置 111 に格納され、拡張ユニット制御部 113 やユニット間同期部 114 により読み出されてもよい。

#### 【0120】

基本ユニット 3 は、基本ユニット 3 の制御周期と拡張ユニット 4 の制御周期とを同期させるユニット間同期部 73 と、ユニット間同期部 73 が同期処理を実行する同期周期における所定期間において、拡張ユニット 4 が保持しているデータと基本ユニット 3 が保持しているデータとが一致するように更新する第 2 更新部 75 とを有する。第 2 更新部 75 は、相互に更新の連動性を要求される複数のデータまたは予め定められた更新順序にしたがって更新されることを要求される複数のデータを更新してもよい。これにより同時性の問題や順序性の問題などが緩和されよう。

#### 【0121】

なお、リアルタイム性が要求されないようなデータを扱う拡張ユニットについては、従来通り、一括リフレッシュされればよい。一方で、モーションユニット（位置決めユニット）などリアルタイム性や高速動作が要求される拡張ユニットについては、同期リフレッシュの対象とされてもよい。とりわけ、同期周期が短くなると、1 つのスキャンタイムにおいて多数の回数にわたって同期リフレッシュが実行されることになる。よって、同期リフレッシュの対象となる拡張ユニットやそのデバイスが増えれば増えるほど、スキャンタイムが延びてしまったり、内部処理の開始が遅れてしまったりする。よって、同期周期がスキャンタイムより短い場合は、同期リフレッシュの対象を絞ることが望ましいだろう。これは、第 2 拡張ユニット 4 のデータのうち第 2 更新部 76 によって更新されるデータと第 1 更新部 75 によって更新されるデータとを選択する選択手段を設けることで、ユーザが同期リフレッシュの対象を絞ってもよい。

#### 【0122】

拡張ユニット 4 は、第 2 更新部 76 がデータの更新処理を完了した後で当該データを参照して内部処理を実行してもよい。これは、データの更新処理が完了する前にそのデータを内部処理が使用してしまうと同時性の問題などが発生するからである。なお、第 2 更新部 76 は、拡張ユニット 4 が内部処理を実行していない期間にデータの更新処理を実行してもよい。これにより同時性の問題が緩和されよう。

#### 【0123】

拡張ユニット 4 は、第 2 更新部 76 によって更新されるデータを記憶する第 1 バッファ

1 2 1 と、当該拡張ユニット 4 の内部処理によって参照されるデータを記憶する第 2 バッファ 1 2 2 とを備えていてもよい。この場合、バッファ更新部 1 1 6 は、ユニット間同期処理が実行されるときに第 1 バッファ 1 2 1 に記憶されているデータと第 2 バッファ 1 2 2 に記憶されているデータとが一致するように更新する第 3 更新手段として機能する。第 2 更新部 7 6 は、バッファ更新部 1 1 6 が更新処理を完了した後で、更新処理を実行する。たとえば、第 2 更新部 7 6 は、同期周期の開始タイミングから所定期間が経過してから同期リフレッシュを実行する。所定期間は、バッファ更新部 1 1 6 が更新処理を完了するのに十分な期間であり、予め設定される。

#### 【0124】

拡張ユニット 4 は、ユニット間同期部 1 1 4 による同期処理が遅延すると、内部処理の開始タイミングを延期する延期部 1 1 5 をさらに有していてもよい。同期処理の遅延は、伝送路でのエラー等、突発的な事象が原因となることが多い。そのため、常に一定期間を待たせるよりも、遅延時間に応じて内部処理を遅らせた方が有利であろう。

#### 【0125】

図 1 4 や図 1 5 を用いて説明したように、基本ユニット 3 は、ユニット間同期部 7 3 が同期処理を実行するタイミングになると、一括リフレッシュの期間に遷移してもよい。これにより、ユニット間同期により獲得された同期周期に対してスキャンの開始タイミングを整合させることが可能となる。

#### 【0126】

上述したように、基本ユニット 3 はユーザプログラムにおいて拡張ユニット 4 の内部処理に割り付けられているデバイスメモリを有している。また、基本ユニット 3 または拡張ユニット 4 は、デバイスメモリに記憶されているデータと同一のデータを記憶するように制御されるバッファメモリ U G を有していてもよい。なお、基本ユニット 3 にバッファメモリ U G が配置される場合、拡張ユニット 4 からこのバッファメモリ U G はアクセスされ、データを読み書きされることになる。第 1 更新部 7 5 や第 2 更新部 7 6 は、拡張ユニット 4 の制御周期に同期して、デバイスメモリに記憶されているデータとバッファメモリ U G に記憶されているデータとが一致するように更新処理を実行してもよい。このようにバッファメモリ U G をデバイスに割り付けることで、バッファメモリ U G についてもあたかもデバイスのように一括リフレッシュや同期リフレッシュが適用可能となる。

#### 【0127】

上述したように複数の拡張ユニット 4 が基本ユニット 3 を介してデバイス値を送受信するときにも本発明は有利であろう。この場合、基本ユニット 3 は、少なくとも、基本ユニット 3 の制御周期と複数の拡張ユニット 4 の制御周期とを同期させるユニット間同期部 7 3 と、ユーザプログラムにおいて第 1 拡張ユニット 4 の内部処理に割り付けられており、当該第 1 拡張ユニット 4 から基本ユニット 3 に転送すべきデータを記憶する第 1 デバイスメモリと、ユーザプログラムにおいて第 2 拡張ユニット 4 の内部処理に割り付けられており、基本ユニット 3 から当該第 2 拡張ユニット 4 へ転送すべきデータを記憶する第 2 デバイスメモリとを有している。なお、図 7 では第 1 デバイス 8 1 や第 2 デバイス 8 2 はそれぞれ 1 つずつしか記載されていないが、これらは複数個あってもよい。よって、第 1 デバイスメモリや第 2 デバイスメモリとは、複数の第 1 デバイス 8 1 や複数の第 2 デバイス 8 2 に相当する。

#### 【0128】

第 1 拡張ユニット 4 は、第 1 デバイスメモリに記憶されているデータと同一のデータを記憶するように制御される第 1 バッファメモリを有していてもよい。第 2 拡張ユニット 4 は、第 2 デバイスメモリに記憶されているデータと同一のデータを記憶するように制御される第 2 バッファメモリを有していてもよい。図 8 には、第 1 デバイス 1 1 8 と第 2 デバイス 1 2 0 が例示されているが、これとは別にバッファメモリ U G が記憶装置 1 1 1 に設けられている。図 1 3 が示すように、第 1 拡張ユニット 4 から入力されたデバイス値が基本ユニット 3 を介して第 2 拡張ユニット 4 に転送されてもよい。これを実現するために、第 1 更新部 7 5 または第 2 更新部 7 6 は、拡張ユニット 4 の制御周期に同期して、第 1 デ

10

20

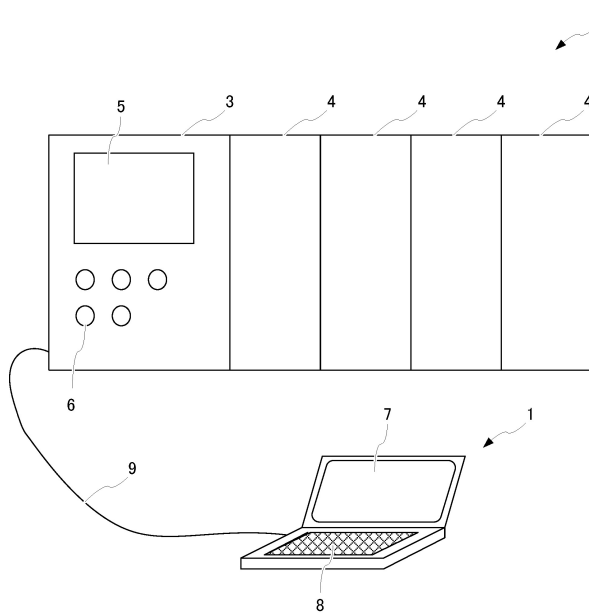
30

40

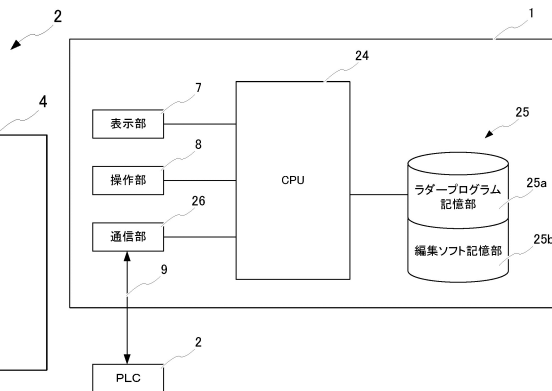
50

バースメモリに記憶されているデータと第1バッファメモリに記憶されているデータとが一致するように更新処理を実行し、次に、第2デバイスメモリに記憶されているデータと第2バッファメモリに記憶されているデータとが一致するように更新処理を実行する。なお、第1更新部75または第2更新部76は拡張ユニット4側に設けられていてもよい。第1拡張ユニット4は、第2更新部76が同期リフレッシュを完了した後で内部処理を開始する。同様に、第2拡張ユニット4は、第2更新部76が同期リフレッシュを完了した後で内部処理を開始する。これにより一つの同期周期の伝達遅延時間だけで複数の拡張ユニット間でのデバイス値の送受信が実現可能となる。

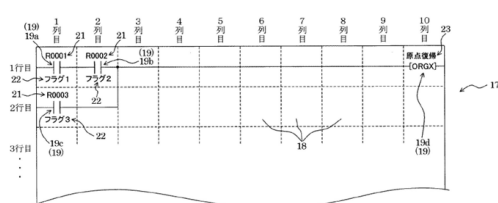
【図1】



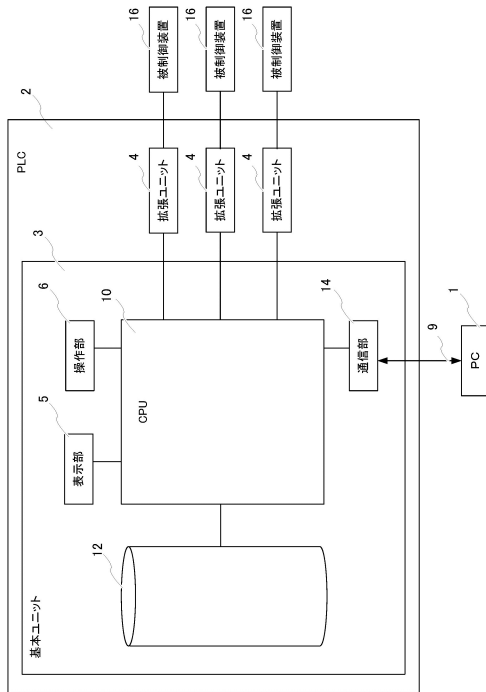
【図3】



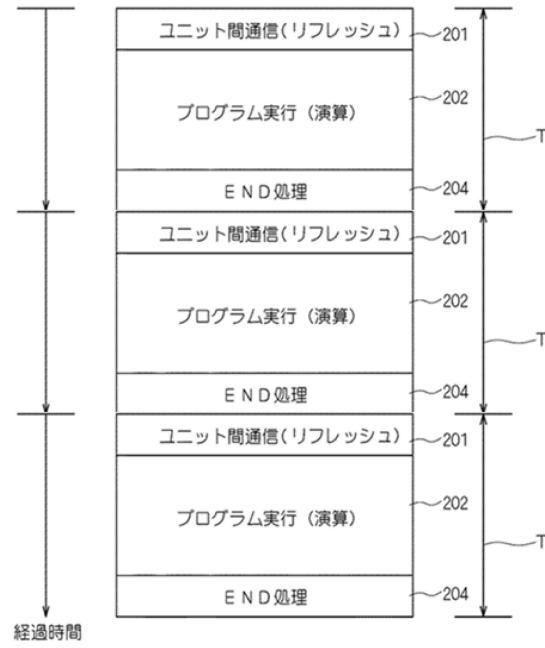
【図2】



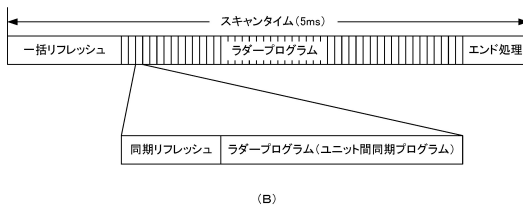
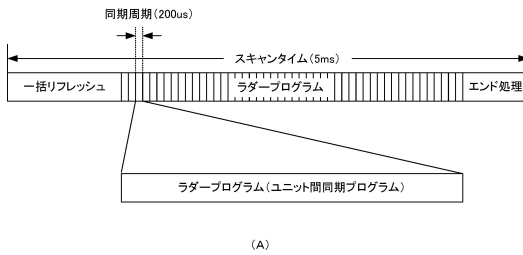
【図4】



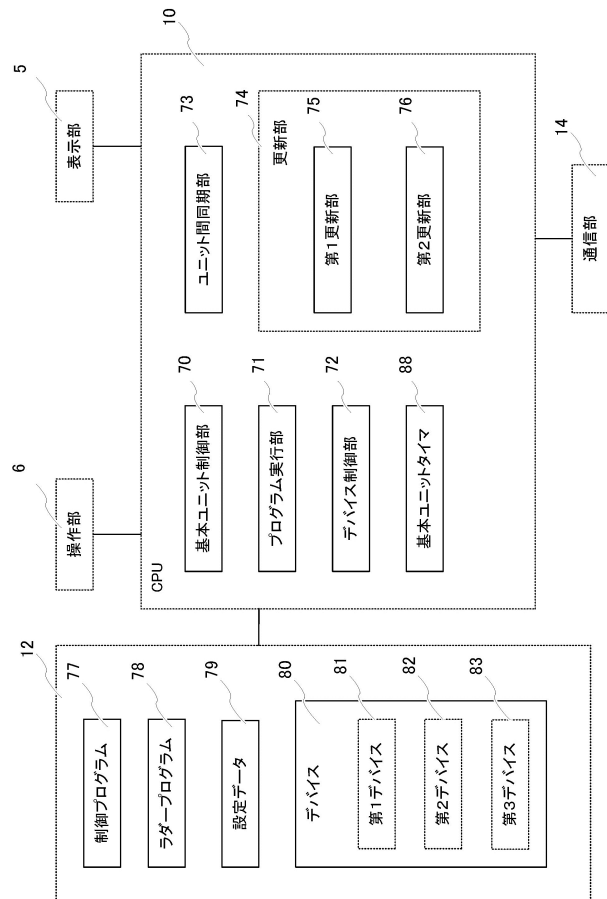
【図5】



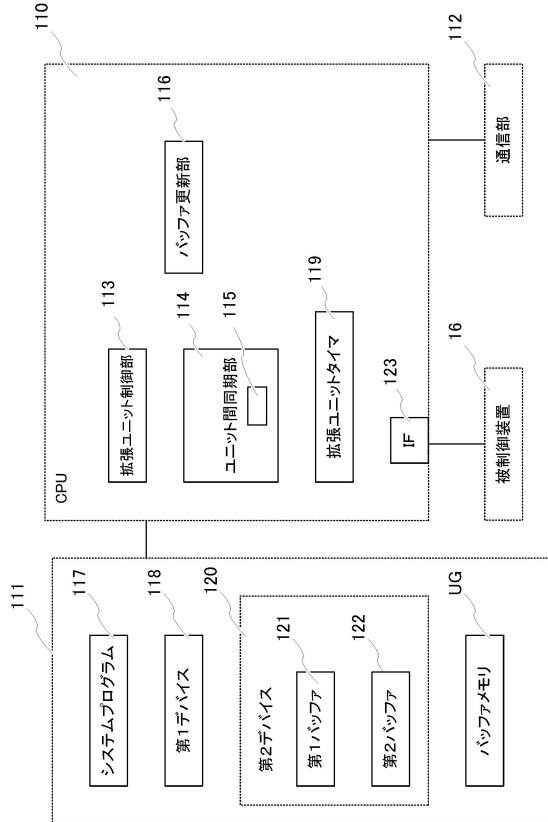
【図6】



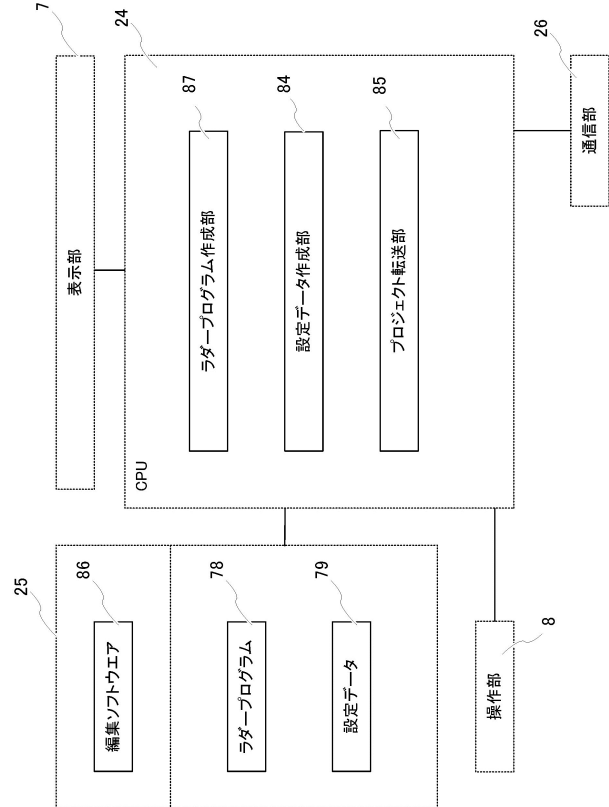
【図7】



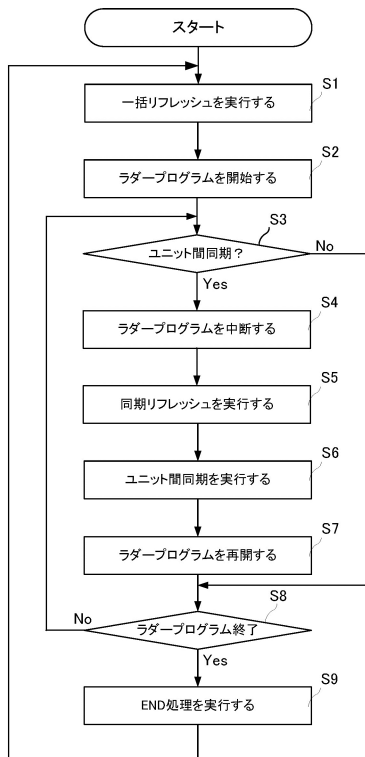
【圖 8】



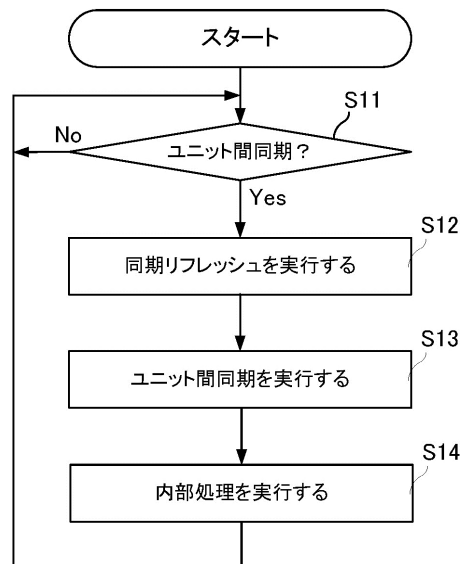
【 図 9 】



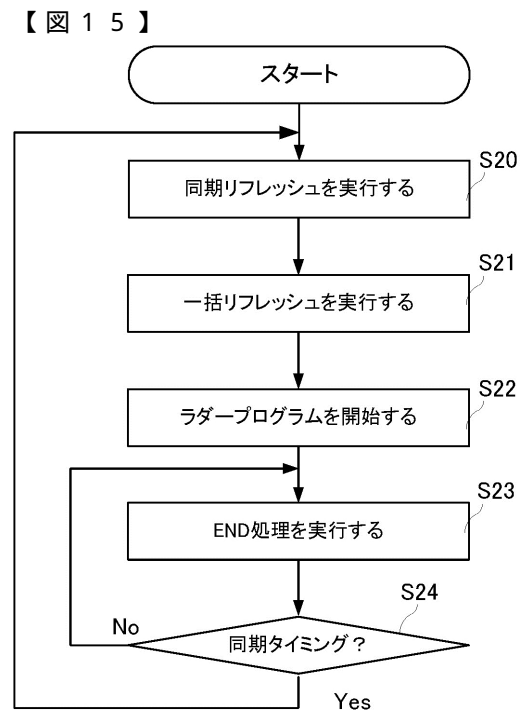
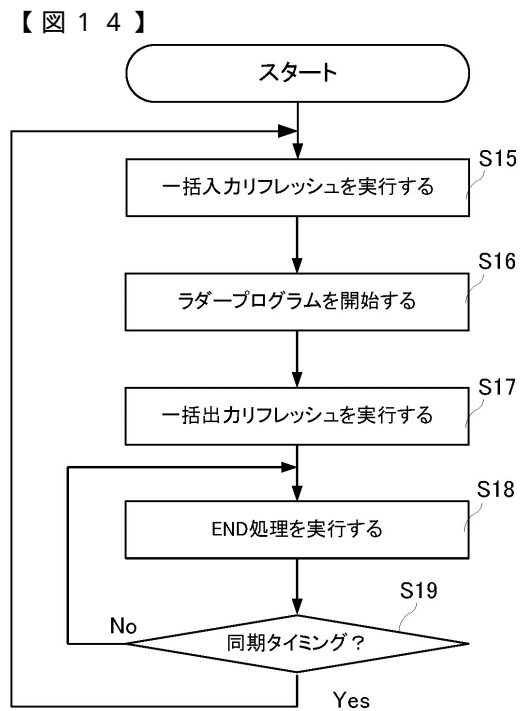
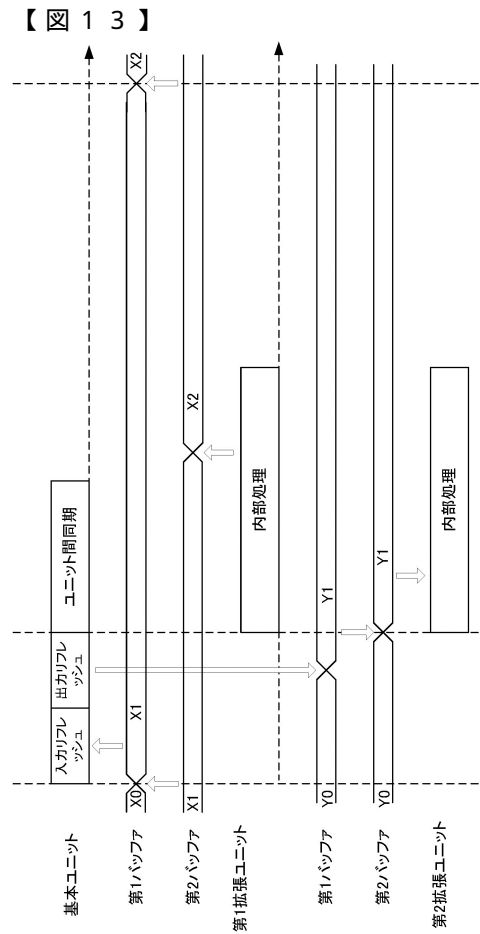
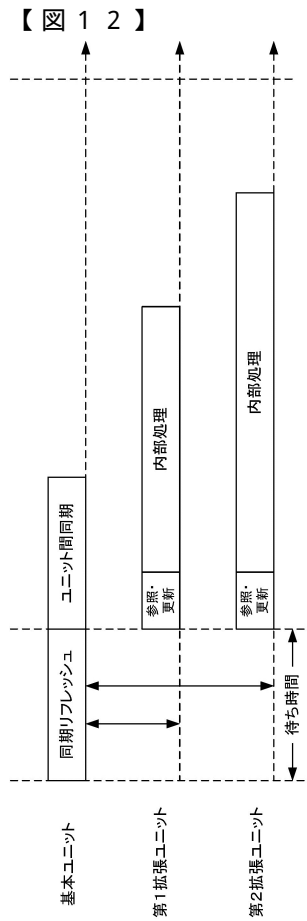
【 図 1 0 】



【 図 1 1 】







【図 16】

90

ユニット種別	リフレッシュの選択
LAN通信ユニット	一括リフレッシュ
32点入力ユニット	同期リフレッシュ
32点出力ユニット	
4chアナログ出力ユニット	同期リフレッシュ
温度調整ユニット	
モーションユニット	同期リフレッシュ
-	-

---

フロントページの続き

(72)発明者 北川 知伸  
大阪府大阪市東淀川区東中島 1 - 3 - 14 株式会社キーエンス内

審査官 永田 和彦

(56)参考文献 特開 2010 - 198600 (JP, A)  
特開 2003 - 29809 (JP, A)  
特開 2008 - 251052 (JP, A)  
国際公開第 2014 / 108999 (WO, A1)  
特開 2014 - 52672 (JP, A)

(58)調査した分野(Int.Cl., DB名)  
G05B 19/05