

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6295914号
(P6295914)

(45) 発行日 平成30年3月20日(2018.3.20)

(24) 登録日 平成30年3月2日(2018.3.2)

(51) Int.Cl. F I
G05B 19/05 (2006.01) G05B 19/05 A

請求項の数 11 (全 12 頁)

(21) 出願番号	特願2014-208552 (P2014-208552)	(73) 特許権者	000005234
(22) 出願日	平成26年10月10日(2014.10.10)		富士電機株式会社
(65) 公開番号	特開2016-81097 (P2016-81097A)		神奈川県川崎市川崎区田辺新田1番1号
(43) 公開日	平成28年5月16日(2016.5.16)	(74) 代理人	100074099
審査請求日	平成29年2月14日(2017.2.14)		弁理士 大菅 義之
		(72) 発明者	佐藤 直己
			神奈川県川崎市川崎区田辺新田1番1号
			富士電機株式会社内
		審査官	黒田 暁子

最終頁に続く

(54) 【発明の名称】 プログラマブルコントローラシステム、その支援装置、プログラマブルコントローラ

(57) 【特許請求の範囲】

【請求項1】

プログラマブルコントローラに接続可能な支援装置であって、
ファンクションブロックが記述された任意の制御プログラムのソースコードをコンパイルして該ファンクションブロックに係わる各F Bインスタンスを生成する際に、ファンクションブロックの呼出元のタスクのレベルに応じたインデックスメモリ領域にアクセスする為のアクセス情報を生成して、該アクセス情報を当該F Bインスタンスに埋め込むコンパイラ手段、

を有することを特徴とする支援装置。

【請求項2】

複数の前記インデックスメモリ領域は、前記プログラマブルコントローラが有するインデックスレジスタの記憶領域を、複数に分割して成るものであり、

各インデックスメモリ領域は、各タスクレベルに対応して設けられ、

前記アクセス情報は、該インデックスレジスタの記憶領域の先頭からのオフセット値であることを特徴とする請求項1記載の支援装置。

【請求項3】

前記コンパイラ手段は、前記インデックスメモリ領域の記憶容量に、前記呼出元のタスクのレベル値を乗じることで、前記オフセット値を算出することを特徴とする請求項2記載の支援装置。

【請求項4】

10

20

前記インデックスメモリ領域へのアクセスをメモリ間接指定命令によって命令することを特徴とする請求項 1 ~ 3 の何れかに記載の支援装置。

【請求項 5】

ファンクションブロックが記述された任意の制御プログラムのソースコードをコンパイルして該ファンクションブロックに係わる各 F B インスタンスを生成する際に、ファンクションブロックの呼出元のタスクのレベルに応じたインデックスメモリ領域にアクセスする為のアクセス情報を生成して、該アクセス情報を当該 F B インスタンスに埋め込むコンパイラ手段を有する支援装置に接続可能に構成されるプログラマブルコントローラであって、

前記コンパイラ手段による前記コンパイルの結果である機械語オブジェクトを保持する記憶手段と、

前記記憶手段に保持された前記機械語オブジェクトを実行するプログラム実行手段と、
を有することを特徴とするプログラマブルコントローラ。

【請求項 6】

前記支援装置が有する転送手段によって転送されてくる前記機械語オブジェクトを受信して、該受信した機械語オブジェクトを前記記憶手段に記憶する受信手段を更に有することを特徴とする請求項 5 記載のプログラマブルコントローラ。

【請求項 7】

前記プログラム実行手段は、前記記憶手段に保持された前記機械語オブジェクトをプログラム実行メモリにコピーし、当該プログラム実行メモリにコピーしたプログラムを実行することを特徴とする請求項 5 または 6 記載のプログラマブルコントローラ。

【請求項 8】

前記インデックスメモリ領域は複数あり、該複数のインデックスメモリ領域は、前記プログラマブルコントローラが有するインデックスレジスタの記憶領域を、複数に分割して成るものであり、

前記 F B インスタンス実行の際には呼出元のタスクレベルに応じたインデックスメモリ領域にアクセスすることを特徴とする請求項 6 記載のプログラマブルコントローラ。

【請求項 9】

インデックスレジスタを有し、

前記プログラム実行手段が前記機械語オブジェクトを実行することで、前記 F B インスタンス実行の際には呼出元のタスクレベルに応じたインデックスメモリ領域にアクセスすることを特徴とする請求項 5 記載のプログラマブルコントローラ。

【請求項 10】

プログラマブルコントローラと、該プログラマブルコントローラに接続可能な支援装置を有するプログラマブルコントローラシステムにおいて、

前記支援装置は、

ファンクションブロックが記述された任意の制御プログラムのソースコードをコンパイルして該ファンクションブロックに係わる各 F B インスタンスを生成する際に、ファンクションブロックの呼出元のタスクのレベルに応じたインデックスメモリ領域にアクセスする為のアクセス情報を生成して、該アクセス情報を当該 F B インスタンスに埋め込むコンパイラ手段を有し、

前記プログラマブルコントローラは、

前記コンパイラ手段による前記コンパイルの結果である機械語オブジェクトを実行するプログラム実行手段、

を有することを特徴とするプログラマブルコントローラシステム。

【請求項 11】

前記プログラマブルコントローラは、

インデックスレジスタを有し、前記プログラム実行手段が前記機械語オブジェクトを実行することで、前記 F B インスタンス実行の際には呼出元のタスクレベルに応じたインデックスメモリ領域にアクセスすることを特徴とする請求項 10 記載のプログラマブルコン

10

20

30

40

50

トローラシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、IEC61131-3に準拠したプログラマブルコントローラの制御プログラム生成に関する。

【背景技術】

【0002】

プログラマブルコントローラ（PLC）の技術分野において、IEC61131-3に準拠したソフトウェアは、図4に示すように、コンフィグレーション（PLCシステム）、リソース（PLCの単一CPU）、タスク（Task；PLCプログラムの型）から構成される。

10

【0003】

Taskは複数定義可能であり、各Taskにレベルが設定される。プログラムはTaskに登録する事で、Taskの実行時にCallされ実行される。ファンクションブロック（FB）は、プログラムの実行時にプログラム（PG）もしくは他のFBからCallされて実行される。プログラム（PG）やFB上では様々な命令が使用される。また、後述するように、任意の1つのFBが複数のプログラム（PG）からCallされる場合もある。

【0004】

また、プログラマブルコントローラのプログラム（制御プログラム等）に関して、近年、旧システムのプログラムを、IEC61131-3に準拠したシステムに移植する要望が増えてきている。

20

【0005】

ここで、IEC61131-3に規定されている命令群には、メモリ間接指定命令（インデックス命令など）は、存在していない。そもそも、IEC61131-3には、インデックスレジスタに係わる処理の概念がない。その一方で、この様な制約が無かった旧システムのプログラムの中には、インデックス命令（インデックスレジスタ命令）があるプログラムも少なくない。更に、旧システムでは、インデックスレジスタのデータ入出力制御に係わるサポート機能が、そのOS（オペレーティングシステム）に組み込まれている場合がある。これについては後述する。

【0006】

30

上記旧システムでは、インデックス命令（LEA命令、PUSH命令、POP命令など）を使用してインデックスレジスタのデータ操作を行っている。LEA命令は、インデックスレジスタからデータ（アドレス）をロードする処理を行い、PUSH命令はインデックスレジスタにデータをセットする処理を行い、POP命令はインデックスレジスタからデータを取り出す処理を行う。

【0007】

また、上記インデックスレジスタからロードしたアドレスを用いて、当該アドレスが示す領域にアクセスしてその格納データを取得／更新等する。あるいは、このアドレスに対して所定のオフセットを加算／減算する等して求めたアドレスを用いて、当該アドレスが示す領域にアクセスしてその格納データを取得／更新等する。あるいは、上記インデックスレジスタの格納アドレスの値自体を、更新する場合もある。尚、これは、インデックス命令の一般的な処理例であるが、この例に限らない。

40

【0008】

ここで、上記インデックス命令が、任意のタスク（そのタスクに設定されているプログラム）から呼び出されて実行される場合がある。タスクにはレベルが設定されている。この場合、異なるレベルの複数のタスクから、インデックスレジスタにアクセスすることになり、特にインデックスレジスタの格納データ（上記格納アドレス）が更新されると、問題が生じることになる。例えば、任意のレベルのタスクの処理を実行中に上位タスクの割り込みを受けた場合、この上位タスクによるインデックス命令によってインデックスレジスタの格納データ（上記格納アドレス）が書き換えられると、問題が生じる場合がある。つ

50

まり、上位レベルタスクによってインデックスレジスタの内容が書き換えられた場合、元々実行中であったタスクの動作に影響する可能性がある。

【0009】

この為、旧システムでは、この様な問題を解消する為の機能が、そのOS（オペレーティングシステム）に組み込まれている。例えば、旧システムでは上記OSのサポート機能によって、上記POP命令やPUSH命令を用いてインデックスレジスタを操作することで、上記問題を解消している。

【0010】

しかしながら、IEC61131-3に準拠したシステムの場合、そもそもインデックスレジスタに係わる処理の概念がないので、そのOSには上記の様なサポート機能は存在していない。

10

【0011】

この為、上記のように、旧システムのプログラムを、IEC61131-3に準拠したシステムに移植しようとしても、インデックス命令に関して問題が生じることになる。

また、例えば、特許文献1の従来技術が知られている。

【0012】

特許文献1の発明は、インデックスレジスタ命令をもたないプログラマブルコントローラにおいて、インデックスレジスタ命令を含む命令からインデックスレジスタ命令を用いない命令を機械的に作成するものである。

20

【0013】

特許文献1の技術では、インデックスレジスタ命令を使用して作成されたプログラムを、機械的作業によってインデックスレジスタ命令を持たない命令に変換し、インデックスレジスタ命令を持たないプログラマブルコントローラを動作させる。

【先行技術文献】

【特許文献】

【0014】

【特許文献1】特開2006-323609号公報

【発明の概要】

【発明が解決しようとする課題】

【0015】

30

上記従来の問題を解決する為に、従来では例えば、インデックスレジスタのメモリ領域を、各タスクレベル毎に分割することが考えられている。これによって、例えばタスクレベル1のタスクによるインデックス命令は、タスクレベル1用の分割領域にしかアクセスできない。例えばタスクレベル2のタスクによるインデックス命令は、タスクレベル2用の分割領域にしかアクセスできない。尚、分割領域を設けることは、インデックスレジスタを複数設けることと同義であると見做してもよい。

【0016】

しかしながら、図5に示すように、異なるレベルのタスクに登録された各プログラムが、同一のFBをコールする場合、このFBからインデックス命令を実行する際に、どのレベル用の分割領域にアクセスするのかを指定することが出来なかった。

40

【0017】

このように、IEC61131-3に準拠したプログラマブルコントローラシステムでは、インデックスレジスタに係わる処理の概念がないので、上記旧システムのOSの様なサポート機能がなく、複数の異なるレベルのタスクから同一のFBをコールする場合に、上述した問題が生じることになる。

【0018】

尚、上記の通り、IEC61131-3に準拠したプログラマブルコントローラシステムでは、インデックスレジスタに係わる処理の概念がないが、実質的にインデックス命令に相当する処理が、行えないわけではない。上記のように、インデックス命令とは、メモリ間接指定命令であり、特定のメモリ領域に格納データに基づいて、アクセス先のアドレスを求めて

50

アクセスするものである。IEC61131-3ではこのようなアクセス方法を禁じる思想であるが、技術的にこのようなアクセスが出来ないわけではない。

【0019】

本発明の課題は、メモリ間接指定命令を有するFBを複数のレベルのタスクからコールし得るプログラマブルコントローラシステムに関して、タスクレベル毎に対応するインデックスメモリ領域を定義すると共に、上記FBに係わる各FBインスタンスが呼出元のタスクレベルに応じたインデックスメモリ領域にアクセスすることを実現させる開発支援装置、プログラマブルコントローラシステム等を、提供することである。

【課題を解決するための手段】

【0020】

本発明の支援装置は、プログラマブルコントローラに接続可能な支援装置であって、ファンクションブロックが記述された任意の制御プログラムのソースコードをコンパイルして該ファンクションブロックに係わる各FBインスタンスを生成する際に、ファンクションブロックの呼出元のタスクのレベルに応じたインデックスメモリ領域にアクセスする為のアクセス情報を生成して、該アクセス情報を当該FBインスタンスに埋め込むコンパイラ手段を有する。

【発明の効果】

【0021】

本発明の開発支援装置、プログラマブルコントローラシステム等によれば、メモリ間接指定命令を有するFBを複数のレベルのタスクからコールし得るプログラマブルコントローラシステムに関して、タスクレベル毎に対応するインデックスメモリ領域を定義すると共に、上記FBに係わる各FBインスタンスが呼出元のタスクレベルに応じたインデックスメモリ領域にアクセスすることを実現させることができる。

【図面の簡単な説明】

【0022】

【図1】開発支援装置の機能構成図である。

【図2】インデックス領域の分割使用例である。

【図3】機械語オブジェクトのPLCにおける実行イメージを示す。

【図4】IEC61131-3に準拠したソフトウェアの構成例である。

【図5】複数のタスクレベルのプログラムがFBをCallする図である。

【発明を実施するための形態】

【0023】

以下、図面を参照して本発明の実施の形態について説明する。

本発明は、プログラマブルコントローラ(PLC)の開発支援装置に係わる。

図1は、開発支援装置の機能構成図である。

【0024】

開発支援装置10は、例えばコンパイラ機能部11を有する。

開発支援装置10は、コンパイラ機能部11によって、任意の制御プログラム1(ソースコード)をコンパイルして、機械語オブジェクトコード2を生成する。そして、不図示の転送機能部によって、この機械語オブジェクトコード2をPLC(プログラマブルコントローラ)3にダウンロード(転送)する。PLC3は、不図示のプログラム実行機能部によってこの機械語オブジェクトコード2を実行することで、不図示の制御対象機器等の制御処理等を実現する。

【0025】

上記制御プログラム1(ソースコード)には、ファンクションブロックが記述されている。コンパイラ機能部11は、制御プログラム1のコンパイル処理において、任意のファンクションブロックに係わる各FBインスタンスを生成する際に、該ファンクションブロックの呼出元のタスクのレベルに応じたインデックスメモリ領域にアクセスする為のアクセス情報を生成して、該アクセス情報を当該FBインスタンスに埋め込む。1つのファンクションブロックに対して複数のタスクが呼出しを行う場合、呼出元のタスク毎に応じた

10

20

30

40

50

F B インスタンスが生成される。これら各 F B インスタンス毎に、呼出元のタスクのレベルに応じたアクセス情報が、埋め込まれることになる。

【 0 0 2 6 】

上記インデックスメモリ領域は、P L C 3 が有するインデックスレジスタの記憶領域を、複数に分割して成るものである。各インデックスメモリ領域は、各タスクレベルに対応して設けられている。上記アクセス情報は、例えば、該インデックスレジスタの記憶領域の先頭からのオフセット値である。コンパイラ機能部 1 1 は、例えば、インデックスメモリ領域の記憶容量に、呼出元のタスクのレベル値を乗じることで、オフセット値を算出するが、この例に限らない。尚、オフセット値の算出は、図示のオフセットアドレス算出部 1 1 a が行う。

10

【 0 0 2 7 】

P L C 3 は、不図示のプログラム実行機能部によって上記機械語オブジェクトコード 2 を実行することで、F B インスタンス実行の際には呼出元のタスクレベルに応じたインデックスメモリ領域にアクセスすることになる。例えばインデックス命令によって、インデックスメモリ領域へのアクセスが行われる。

【 0 0 2 8 】

ここで、上述したことから、P L C 3 は、IEC61131-3に準拠したプログラマブルコントローラシステムである。また、制御プログラム 1 (ソースコード) は、F B (ファンクションブロック) 等を用いて作成されているものとする。上記のように、1つのF B が、複数のタスク(プログラム(P G))から呼び出される(Call)場合もあり、これら複数のタスクのレベルが異なる場合もある。

20

【 0 0 2 9 】

また、上述したことから、上記制御プログラム 1 (ソースコード) は、例えば、不図示の旧システム(上述したIEC61131-3に準拠していないプログラマブルコントローラシステム)における制御プログラムに基づいて、これと略同様の処理内容となるように、開発者等が作成したものである。制御プログラム 1 には、インデックス命令(インデックスレジスタ命令; L E A 命令、P U S H 命令、P O P 命令など)も、記述されていてよい。

【 0 0 3 0 】

ここで、上述したように、IEC61131-3準拠のプログラマブルコントローラシステムは、インデックス命令に対応していない。しかし、上記P L C 3 は、IEC61131-3準拠のプログラマブルコントローラであるが、インデックス命令に対応している。これについて、詳しくは後述するが、例えば上記旧システム等におけるインデックス命令を、本システムでも利用可能とすることもできる。但し、この例に限らず、例えば、上記旧システム等におけるインデックス命令に相当する処理を本システムで実現する命令を、開発者等が新たに作成してもよい。ここでは、この様なインデックス命令やインデックス命令に相当する処理を実現する命令を、“メモリ間接指定命令”と呼ぶものとする。

30

【 0 0 3 1 】

そして、ここでは特に、レベルが異なる複数のタスク(プログラム(P G))から呼び出されるF B が、上記“メモリ間接指定命令”を実行するものである場合について説明する。

40

【 0 0 3 2 】

ここで、一般的に、制御プログラムを構成するF B をコンパイルすると、呼出元タスクに応じた各F B インスタンスが生成される。本手法では、上記コンパイル機能部 1 1 によるコンパイル処理において、F B の実行コードをマッピングする際に(F B インスタンスの生成時に)、呼出元のタスクのレベルに応じたオフセットアドレスを算出して、このオフセットアドレスをF B インスタンスに保有させるものである。

【 0 0 3 3 】

これより、図示のように、コンパイラ機能部 1 1 の処理結果である機械語オブジェクト 2 に含まれる各F B インスタンス 2 a には、オフセットアドレスが含まれることになる。詳しくは後述する。

50

【 0 0 3 4 】

ここで、本例では、タスクレベル毎に対応してインデックスレジスタ領域（インデックスメモリ領域）を設ける。例えば、図 2 に示すように、インデックスレジスタの記憶領域を複数に分割して、これら各分割領域それぞれを任意のタスクレベルに応じたインデックスレジスタ領域とする。つまり、図示の例では、例えば“Level3退避領域”は、タスクレベル‘3’に対応するインデックスレジスタ領域であり、“Level0退避領域”は、タスクレベル‘0’に対応するインデックスレジスタ領域であり、“Default退避領域”は、デフォルトレベルに対応するインデックスレジスタ領域である。

【 0 0 3 5 】

尚、ここではレベル‘3’が最上位レベル、デフォルトレベルが最下位レベルであるものとする。そして、上記“メモリ間接指定命令”を有する任意の F B が、例えばタスクレベル 3 のタスク（そのプログラム）から Call された場合には“Level3退避領域”にアクセスし、あるいは例えばデフォルトレベルのタスク（そのプログラム）から Call された場合には“Default退避領域”にアクセスする。これを実現する為に、コンパイル機能部 1 1（オフセットアドレス算出部 1 1 a）の機能が設けられている。

【 0 0 3 6 】

上記のように、呼び出し元のタスクのレベルに応じたインデックスレジスタ領域にアクセスするので、例えばレベル 3 のタスクに係わる処理によって例えば“Default退避領域”の格納データが書き換えられてしまうような事態は、生じることはない。

【 0 0 3 7 】

ここで、各 F B は、特に図示しない任意のプログラムコード群から成るものである。任意の 1 つの F B を例にすると、本例のコンパイル機能部 1 1 によれば、この F B に係わる各 F B インスタンスには、呼出元のタスクのレベルに応じたオフセットアドレス値が、含まれるようになる。すなわち、この F B に係わる上記プログラムコード群の中にメモリ間接指定命令があった場合には、コンパイル機能部 1 1 は、当該 F B の呼出元のタスクのレベルに応じたオフセットアドレス値を求めて、これを F B インスタンス中に埋め込む。

【 0 0 3 8 】

尚、特に説明しないが、当然、コンパイラ機能部 1 1 の機能によって、F B インスタンスには、自己に埋め込まれたオフセットアドレス値を用いて、該当するインデックスレジスタ領域にアクセスする処理が、記述されることになる。例えば、後述する図 3 の例のように、「アクセス先アドレス = インデックスレジスタの記憶領域の先頭アドレス + オフセットアドレス値」等として、当該アクセス先アドレスにアクセスする処理が、F B インスタンス中に記述されることになる。

【 0 0 3 9 】

コンパイル機能部 1 1 には、上記のようなオフセットアドレス値を算出する機能も含まれており、これが上記オフセットアドレス算出部 1 1 a である。コンパイル機能部 1 1 は、例えば、呼出元のタスクは分かるので、そのタスクレベルも分かる。これより、このタスクレベルを、予め決められた算出式に適用することで、オフセットアドレス値を算出することができる。この算出式の一例を以下に示す。

【 0 0 4 0 】

$$\text{オフセットアドレス値} = \text{所定記憶容量} \times (\text{タスクレベル} + 1)$$

但し、これは図示の例に応じた一例に過ぎない。例えば、仮にデフォルトレベルが無いとする場合には（この場合、Level0が最下位レベルとなる）、

$$\text{オフセットアドレス値} = \text{所定記憶容量} \times \text{タスクレベル}$$

等となることになる。

【 0 0 4 1 】

上記所定記憶容量は、例えば図 2 に示す各退避領域（各タスクレベル毎に対応するインデックス記憶領域；インデックスレジスタの記憶領域を分割した領域と見做すこともできる）の記憶容量である。尚、ここでは全ての退避領域の記憶容量が同一であるものとする。尚、上記所定記憶容量は、図 2 の例では 1 WORD（ワード）である。

10

20

30

40

50

【 0 0 4 2 】

コンパイル機能部 1 1 は、上記メモリ間接指定命令のコンパイル処理結果として、「上記インデックスレジスタの記憶領域の先頭アドレス + 上記オフセットアドレス値」で得られるアドレスから始まる上記“退避領域”にアクセスする処理を生成する。尚、インデックスレジスタの記憶領域（インデックス領域）の先頭アドレスは、予め設定されているものとする。

【 0 0 4 3 】

この様にして生成された機械語オブジェクト 2 は、P L C 3 にダウンロードされて実行されることになる。P L C 3 は、この機械語オブジェクトコード 2 を実行する。尚、特に図示しないが、P L C 3 は、機械語オブジェクトコード 2 を記憶する記憶部と、この記憶されている機械語オブジェクトコード 2 を実行するプログラム実行機能部等を有する。

10

【 0 0 4 4 】

開発支援装置 1 0 で生成された上記機械語オブジェクトコード 2 を、P L C 3 に記憶させる方法は、様々であってよい。例えば開発支援装置 1 0 は、機械語オブジェクトコード 2 を P L C 3 にダウンロード（転送）する転送機能部（不図示）を有し、P L C 3 は、この機械語オブジェクトコード 2 を受信して上記記憶部に記憶する受信機能部（不図示）を有する。但し、この例に限らず、例えばメモリカード等の可搬型記憶媒体を介する方法であっても構わない。

【 0 0 4 5 】

また、例えば一例としては、P L C 3 は更に不図示のプログラム実行メモリを有し、上記不図示のプログラム実行機能部は、上記記憶部に保持された上記機械語オブジェクトコード 2 を、上記プログラム実行メモリにコピーし、当該プログラム実行メモリにコピーしたプログラムを実行するように構成してもよい。

20

【 0 0 4 6 】

図 3 に、上記のように生成された機械語オブジェクトの P L C 3 における実行イメージを示す。

図 3 では、まず図上左側に、任意の F B に関して複数の F B インスタンスが生成されたイメージを示す。そして、図上中央には、そのなかの“F B インスタンス n”の機械語コード群のイメージを示すが、この機械語コード群の殆どは省略して示しており、上記オフセットアドレス値のみを示している。ここでは図示のように、オフセットアドレス値は‘m WORD’であるものとする。尚、ここでは、上記各“退避領域”の記憶容量は全て 1 WORD であるものとする。

30

【 0 0 4 7 】

そして、上記省略している機械語コード群の一部として、このオフセットアドレス値は‘mWORD’を用いて、「インデックス領域の先頭アドレス + オフセットアドレス値‘mWORD’」のアドレスから始まる退避領域に、アクセスする処理が記述されることになる。

【 0 0 4 8 】

これより、この例では、図上右側に示すように、インデックス領域の先頭から‘mWORD’分、すなわち m 個の退避領域の分をオフセットし、その次の退避領域（Level2 退避領域）にアクセスすることになる。

40

【 0 0 4 9 】

本発明では、上述したように、IEC61131-3 に準拠した P L C に係わる開発支援装置において、レベルが異なる複数のタスクからコールされる F B であって実質的にインデックス命令を含む F B をコンパイルして F B インスタンスを生成する際に、呼出元タスクのレベルに応じたオフセットアドレス値を求めて、これを F B インスタンスに埋め込む。これによって、P L C 側でインデックスメモリアクセスを実行する際には、呼出元のタスクレベルに応じた分割領域にアクセスすることができる。つまり、任意のタスク実行中に、より上位レベルのタスク割込みがあった場合でも、この上位タスクによって元のタスクのインデックス値が書き換えられる（メモリ値が破壊される）ことはない。

【 0 0 5 0 】

50

尚、図1に示す開発支援装置10(プログラマブルコントローラの支援装置)は、例えば例えばパソコン等の汎用の一般的なコンピュータ装置により実現される。よって、開発支援装置10は、特に図示しないがハードウェア的には一般的なパソコン等の構成を有している。すなわち、CPU、ハードディスク/メモリ等の記憶装置、入出力インタフェース、ディスプレイ等の表示部、キーボード等のユーザ操作部、通信機能部等の不図示の各構成を有している。

【0051】

上不図示の記憶装置には、予め所定のアプリケーションプログラムが記憶されている。上記不図示のCPUが、このアプリケーションプログラムを実行することで、コンパイラ機能部11(そのオフセットアドレス算出部11a等)の各種処理機能や上記転送機能部等が実現される。

10

【0052】

また、PLC3も、例えば不図示のCPUや記憶装置を有し、この記憶装置に予め記憶されている所定のアプリケーションプログラムを、CPUが実行することにより、上記不図示のプログラム実行機能部や受信機能部等を実現するものである。

【0053】

ここで、ここまで説明していなかったが、PLC3は、IEC61131-3に準拠した既存のプログラマブルコントローラそのままではなく、そのファームウェアに改良を施してある。

まず、上記背景技術や課題で説明したように、旧システムでは、インデックス命令(LEA命令、PUSH命令、POP命令)を使用出来たが、IEC61131-3準拠のプログラマブルコントローラシステムでは、そもそもインデックスレジスタに係わる処理の概念がない。この為、上記PLC3は、インデックス命令を使用できるようにする為の改良をファームウェアに施してある。

20

【0054】

まず、PLC(プログラマブルコントローラ)のファームウェアは、通常、サポートしている命令群のリストを保持している。IEC61131-3準拠の既存のプログラマブルコントローラの場合、上記のことから当然、命令群リスト中にインデックス命令(LEA命令、PUSH命令、POP命令)は無い。これに対して、PLC3の場合、開発者等が事前に命令群リスト中にインデックス命令(LEA命令、PUSH命令、POP命令)を追加する作業を行っている。

30

【0055】

更に、これらLEA命令、PUSH命令、POP命令に応じたプログラムコード(マイクロプログラム等)を、予め開発者等が任意に作成して、PLC3に保持させている。このプログラムコード(マイクロプログラム等)は、旧システムにおけるLEA命令、PUSH命令、POP命令に応じたプログラムコード(マイクロプログラム等)と同一ではないが、同じ処理が実行されるように開発者等が作成している。尚、この様なプログラムコード(マイクロプログラム等)の作成は、当業者であれば問題なく実現できるものである。

【0056】

PLC3は、上記機械語オブジェクト2中にインデックス命令がある場合には、これに応じた上記プログラムコード(マイクロプログラム等)を実行することで、当該インデックス命令の処理を実現させる。

40

【0057】

例えば上述したようにして、PLC3は、IEC61131-3準拠のプログラマブルコントローラであるが、旧システムと略同様にインデックス命令(LEA命令、PUSH命令、POP命令)を実行できるようになっている。しかしながら、上述したように、IEC61131-3に準拠したシステムの場合、そもそもインデックスレジスタに係わる処理の概念がないので、そのOSには旧システムの上述したサポート機能は存在していない。本手法は、上述したようにして、この問題を解消している。

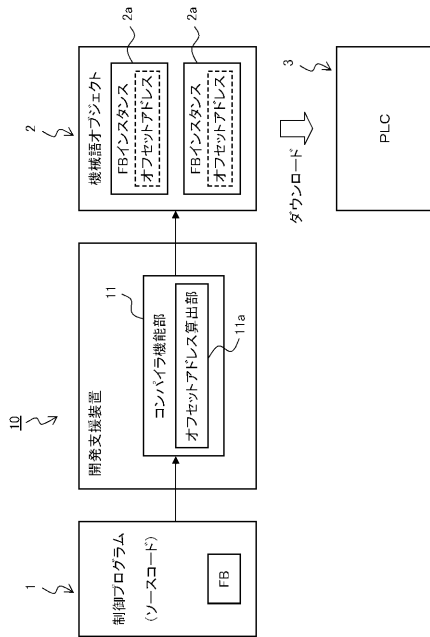
【符号の説明】

50

- 【 0 0 5 8 】
- 1 制御プログラム
- 2 機械語オブジェクト
- 3 P L C
- 1 0 開発支援装置
- 1 1 コンパイラ機能部
- 1 1 a オフセットアドレス算出部

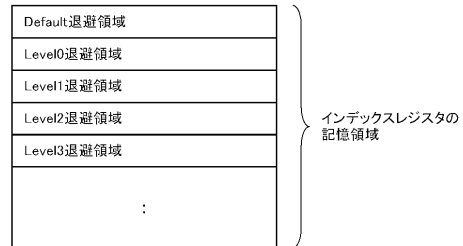
【 図 1 】

開発支援装置の機能構成図



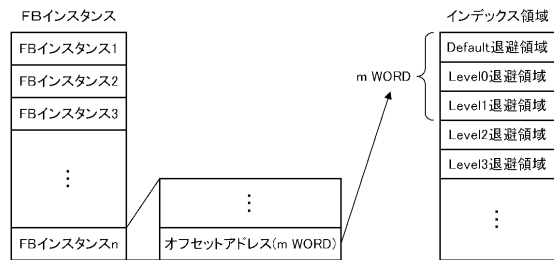
【 図 2 】

インデックス領域の分割使用例



【 図 3 】

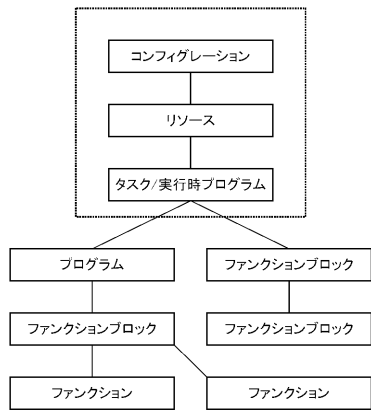
機械語オブジェクトのPLCにおける実行イメージを示す



TASKレベル2のプログラムから呼ばれたFBインスタンス

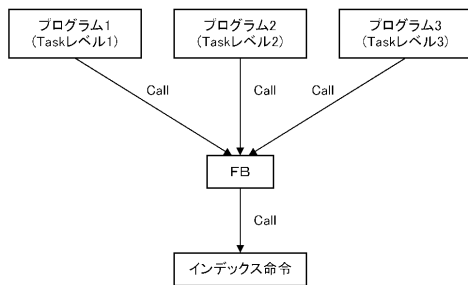
【 図 4 】

IEC61131-3に準拠したソフトウェアの構成例



【 図 5 】

複数のタスクレベルのプログラムがFBをCallする図



フロントページの続き

- (56)参考文献 特開2012-234272(JP,A)
特開平07-334214(JP,A)
特開2001-022413(JP,A)
特開2012-194682(JP,A)
特開2006-323609(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G05B 19/05