

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5771114号
(P5771114)

(45) 発行日 平成27年8月26日(2015. 8. 26)

(24) 登録日 平成27年7月3日(2015. 7. 3)

(51) Int.Cl.

F I

G O 5 B 19/05 (2006.01)

G O 5 B 19/05

W

G O 5 B 19/05

F

請求項の数 12 (全 20 頁)

(21) 出願番号 特願2011-224540 (P2011-224540)
 (22) 出願日 平成23年10月12日(2011.10.12)
 (65) 公開番号 特開2013-84169 (P2013-84169A)
 (43) 公開日 平成25年5月9日(2013.5.9)
 審査請求日 平成26年1月30日(2014.1.30)

(73) 特許権者 000005108
 株式会社日立製作所
 東京都千代田区丸の内一丁目6番6号
 (74) 代理人 110000350
 ポレール特許業務法人
 (72) 発明者 益子 直也
 茨城県日立市大みか町五丁目2番1号 株
 式会社日立製作所 情報制御システム社内
 (72) 発明者 寺江 尚
 茨城県日立市大みか町五丁目2番1号 株
 式会社日立製作所 情報制御システム社内
 (72) 発明者 靖 輝記
 茨城県日立市大みか町五丁目2番1号 株
 式会社日立製作所 情報制御システム社内

最終頁に続く

(54) 【発明の名称】 コントローラ、およびタスクとラダーの処理方法

(57) 【特許請求の範囲】

【請求項 1】

ラダー処理を実行する第1のプロセッサと、タスク処理を実行する第2のプロセッサと、プログラムとしてラダー処理のための複数のラダープログラムとタスク処理のための複数のタスクプログラムを収納する主メモリを備え、前記複数のラダープログラムと複数のタスクプログラムについて、処理の優先度に応じてラダープログラムとタスクプログラムの組み合わせを定めておき、

前記第2のプロセッサは、処理の優先度ごとの処理周期での割り込みに応じて、当該優先度でのラダープログラムの実行を前記第1のプロセッサに指示し、

前記第1のプロセッサは、前記第2のプロセッサの指示に従って主メモリ内のラダープログラムから当該優先度のラダープログラムを得、これに従ってラダー処理を実行し、

前記第2のプロセッサは、主メモリ内のタスクプログラムから当該優先度のタスクプログラムを得、これに従ってタスク処理を実行し、

指定された割り込みの優先度ごとの処理周期内に、当該優先度でのタスクプログラムによるタスク処理と、当該優先度でのラダープログラムによるラダー処理を完了するとともに、前記処理の優先度ごとの処理周期は、処理の優先度が上位であるほど短い処理周期とされており、

前記第2のプロセッサは、前記割り込みの新たな発生を監視して実行中の優先度よりも上位の優先度の第2の割り込みであることを検知し、

前記第1のプロセッサおよび、または前記第2のプロセッサは、実行中の前記プログラ

10

20

ムを用いた処理を中断して、上位の優先度の第 2 の割り込みで定まるプログラムを用いた処理を実行し、

前記第 1 のプロセッサおよび、または前記第 2 のプロセッサは、前記上位の優先度の第 2 の割り込みで定まるプログラムを用いた処理の完了後に、第 2 の割り込み発生前に実行中のプログラムを用いた処理を前記の中断した個所から再開し、

指定された割り込みの優先度ごとの処理周期内に、当該優先度でのタスクプログラムによるタスク処理と、当該優先度でのラダープログラムによるラダー処理を完了することを特徴とするコントローラ。

【請求項 2】

請求項 1 に記載のコントローラにおいて、

前記第 1 のプロセッサは、ラダー処理の実行中に異常が発生したことを検知して前記第 2 のプロセッサに報告し、

前記第 2 のプロセッサは、当該優先度のタスク処理を実行すると共に、次回割り込み時の当該優先度のラダー処理についての処理の可否を決定することを特徴とするコントローラ。

【請求項 3】

請求項 1 または請求項 2 に記載のコントローラにおいて、

前記主メモリに収納されたラダー処理のためのラダープログラムは、複数の一連のラダー命令で構成され、

前記第 1 のプロセッサは、前記ラダー命令に従い当該のラダー処理を実行すると共に、実行できないラダー命令を検出した際には、ラダー処理を一時停止し、

前記第 2 のプロセッサは、前記停止したラダー命令を実行し、

前記第 1 のプロセッサは、前記第 2 のプロセッサにおける前記停止したラダー命令の実行後に、ラダー処理を停止したアドレスの次の番地から継続してラダー処理を再開することを特徴とするコントローラ。

【請求項 4】

請求項 1 から請求項 3 のいずれか 1 項に記載のコントローラにおいて、

ラダー処理を行う前記第 1 のプロセッサは、ソフトウェアを使用して実行されていることを特徴とするコントローラ。

【請求項 5】

請求項 1 から請求項 4 のいずれか 1 項に記載のコントローラにおいて、

割り込みにより指定された同一優先度の前記第 1 のプロセッサのラダー処理と、前記第 2 のプロセッサのタスク処理は、その処理周期内で一方の処理が完了後に他方の処理を実行する直列処理とされる

ことを特徴とするコントローラ。

【請求項 6】

請求項 1 から請求項 5 のいずれか 1 項に記載のコントローラにおいて、

割り込みにより指定された同一優先度の前記第 1 のプロセッサのラダー処理と、前記第 2 のプロセッサのタスク処理は、その処理周期内で双方の処理が同時に実行される期間を含む並列処理とされる

ことを特徴とするコントローラ。

【請求項 7】

ラダー処理を実行する第 1 のプロセッサと、タスク処理を実行する第 2 のプロセッサとを備え、複数のラダー処理と複数のタスク処理について、処理の優先度に応じてラダー処理とタスク処理の組み合わせを定めておき、処理の優先度ごとに定められた処理周期での割り込みを行い、

割り込みに応じて、当該優先度でのラダー処理を前記第 1 のプロセッサで行い、当該優先度でのタスク処理を前記第 2 のプロセッサで行い、

割り込みの優先度ごとの処理周期内に、当該優先度でのタスク処理と、当該優先度でのラダー処理を完了するとともに、前記処理の優先度ごとの処理周期は、処理の優先度が上

10

20

30

40

50

位であるほど短い処理周期とされており、

処理の優先度ごとの処理周期の期間内に、実行中の優先度よりも上位の優先度の第2の割り込みで発生したときに、

第1のプロセッサと第2のプロセッサは、第2の割り込みで定まる優先度のラダー処理とタスク処理を実行し、

前記上位の優先度の第2の割り込みで定まる処理の完了後に、第2の割り込み発生前に実行中の処理を前記中断した個所から再開し、

指定された割り込みの優先度ごとの処理周期内に、当該優先度でのタスク処理と、当該優先度でのラダー処理を完了する

ことを特徴とするタスクとラダーの処理方法。

10

【請求項8】

請求項7に記載のタスクとラダーの処理方法において、

前記第1のプロセッサが、ラダー処理の実行中に異常が発生したことを検知したとき、

前記第2のプロセッサは、当該優先度のタスク処理を実行すると共に、次回割り込み時の当該優先度のラダー処理についての処理の可否を決定する

ことを特徴とするタスクとラダーの処理方法。

【請求項9】

請求項7または請求項8に記載のタスクとラダーの処理方法において、

前記第1のプロセッサは、一連のラダー命令に従いラダー処理を実行すると共に、実行できないラダー命令を検出した際には、ラダー処理を一時停止し、

前記第2のプロセッサは、前記停止したラダー命令をソフト処理により実行し、

前記第1のプロセッサは、ラダー処理を停止したアドレスの次の番地から継続してラダー処理を再開する

ことを特徴とするタスクとラダーの処理方法。

20

【請求項10】

請求項7から請求項9のいずれか1項に記載のタスクとラダーの処理方法において、

ラダー処理を行う前記第1のプロセッサは、ソフトウェアを使用して実行されていることを特徴とするタスクとラダーの処理方法。

【請求項11】

請求項7から請求項10のいずれか1項に記載のタスクとラダーの処理方法において、

割り込みにより指定された同一優先度の前記第1のプロセッサのラダー処理と、前記第2のプロセッサのタスク処理は、その処理周期内で一方の処理が完了後に他方の処理を実行する直列処理とされる

ことを特徴とするタスクとラダーの処理方法。

30

【請求項12】

請求項7から請求項11のいずれか1項に記載のタスクとラダーの処理方法において、

割り込みにより指定された同一優先度の前記第1のプロセッサのラダー処理と、前記第2のプロセッサのタスク処理は、その処理周期内で双方の処理が同時に実行される期間を含む並列処理とされる

ことを特徴とするタスクとラダーの処理方法。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明はタスク処理とラダー処理を実行するコントローラ、およびタスクとラダーの処理方法に関する。

【背景技術】

【0002】

一般にコントローラは、ラダー処理を実行するソフトウェア、または、ラダー処理プロセッサを設けており、定周期で、または連続してラダー処理の実行を行っている。またコントローラは、定周期でラダー処理を実行した後の空いた時間でラダー処理以外のプログ

50

ラム実行（タスク処理）を行っている。

【0003】

また、リアルタイムOSのマルチタスクスケジューリングでは、タスクは優先度に基づいてプリエンティブスケジューリングされ、最高優先度の実行可能タスクを実行する。仮に、現在実行中のタスクよりも高い優先度のタスクが実行可能となった場合、即座にタスクの切り替えを行っている。

【0004】

上記の処理を可能とするコントローラにおけるラダー処理実行方法に関し、以下の特許文献のように処理することが知られている。

【0005】

特許文献1では、第2のプロセッサから第1のプロセッサに対して、異なるサイクルで異なるレベルの割り込みを入れ、割り込みに応じあらかじめ登録済みのラダー処理を実行している。

【0006】

特許文献2では、定周期で実行されるラダー処理実行後に条件の成立したタスクを実行している。

【0007】

特許文献3では、複数のラダープログラムを必要に応じ実行させ、プロセッサの演算量を減らしコントローラの応答性能向上を行っている。

【0008】

特許文献4では、ラダー処理によるタスクのスケジューリングを行っている。

【先行技術文献】

【特許文献】

【0009】

【特許文献1】特開2001-005505号公報

【特許文献2】特開平01-281503号公報

【特許文献3】特開平01-140301号公報

【特許文献4】特開平08-272630号公報

【発明の概要】

【発明が解決しようとする課題】

【0010】

上記の特許文献による処理では、以下の問題に対応することができない。

【0011】

第1に、複数の組のラダー処理を実行させることができていない。上記の、ラダー処理プロセッサを持ちラダー処理の実行を行うコントローラでは、1組のラダー処理を周期的に実行し、ラダー処理実行の空いた時間でラダー処理以外のタスクの実行を行っているにすぎない。

【0012】

第2に、タスクを先に実行させ、タスク処理からラダー処理を呼び出して動作させること、あるいは1つのタスクから複数のラダー処理を呼び出すといったことができない。これは、マイクロプロセッサが実行するタスク処理とラダー処理を組み合わせで動作させた場合は、マルチタスク処理等で発生するタスクの中断等により、タスクの演算時間が延びた場合にラダー処理のリアルタイム演算（所定の時間内に演算終了することを保障する）ができないからである。

【0013】

第3に、ラダー処理、または、タスク処理において異常を検出した場合、コントローラ自体を停止させるためコントローラとしての可用性が低いことが問題であった。

【0014】

本発明では、前記した問題を解決し、複数の組のラダー処理の実行とタスク処理を優先順位に応じたマルチタスクスケジューリングを実現するコントローラ、およびタスクとラ

10

20

30

40

50

ダーの処理方法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】

【0015】

前記課題を解決するために、本発明のコントローラでは、ラダー処理を実行する第1のプロセッサと、タスク処理を実行する第2のプロセッサと、プログラムとしてラダー処理のための複数のラダープログラムとタスク処理のための複数のタスクプログラムを収納する主メモリを備え、複数のラダープログラムと複数のタスクプログラムについて、処理の優先度に応じてラダープログラムとタスクプログラムの組み合わせを定めておき、第2のプロセッサは、処理の優先度ごとの処理周期での割り込みに応じて、当該優先度でのラダープログラムの実行を前記第1のプロセッサに指示し、第1のプロセッサは、第2のプロセッサの指示に従って主メモリ内のラダープログラムから当該優先度のラダープログラムを得、これに従ってラダー処理を実行し、第2のプロセッサは、主メモリ内のタスクプログラムから当該優先度のタスクプログラムを得、これに従ってタスク処理を実行し、指定された割り込みの優先度ごとの処理周期内に、当該優先度でのタスクプログラムによるタスク処理と、当該優先度でのラダープログラムによるラダー処理を完了する。

10

【0016】

また、処理の優先度ごとの処理周期は、処理の優先度が上位であるほど短い処理周期とされている。

【0017】

また、第2のプロセッサは、割り込みの新たな発生を監視して、実行中の優先度よりも上位の優先度の第2の割り込みであることを検知し、第1のプロセッサおよび、または第2のプロセッサは、実行中のプログラムを用いた処理を中断して、上位の優先度の第2の割り込みで定まるプログラムを用いた処理を実行し、第1のプロセッサおよび、または第2のプロセッサは、上位の優先度の第2の割り込みで定まるプログラムを用いた処理の完了後に、第2の割り込み発生前に実行中のプログラムを用いた処理を再開し、指定された割り込みの優先度ごとの処理周期内に、当該優先度でのタスクプログラムによるタスク処理と、当該優先度でのラダープログラムによるラダー処理を完了する。

20

【0018】

また、第1のプロセッサは、ラダー処理の実行中に異常が発生したことを検知して第2のプロセッサに報告し、第2のプロセッサは、当該優先度のタスク処理を実行すると共に、次回割り込み時の当該優先度のラダー処理についての処理の可否を決定する。

30

【0019】

また、主メモリに収納されたラダー処理のためのラダープログラムは、複数の一連のラダー命令で構成され、第1のプロセッサは、ラダー命令に従い当該のラダー処理を実行すると共に、実行できないラダー命令を検出した際には、ラダー処理を一時停止し、第2のプロセッサは、停止したラダー命令を実行し、第1のプロセッサは、第2のプロセッサにおける停止したラダー命令の実行後に、ラダー処理を停止したアドレスの次の番地から継続してラダー処理を再開する。

【0020】

また、第1のプロセッサはハードウェアで構成され、第2のプロセッサは、ソフトウェアで構成されている。

40

【0021】

また、割り込みにより指定された同一優先度の第1のプロセッサのラダー処理と、第2のプロセッサのタスク処理は、その処理周期内で一方の処理が完了後に他方の処理を実行する直列処理とされる。

【0022】

また、割り込みにより指定された同一優先度の第1のプロセッサのラダー処理と、第2のプロセッサのタスク処理は、その処理周期内で双方の処理が同時に実行される期間を含む並列処理とされる。

【0023】

50

前記課題を解決するために、本発明のタスクとラダーの処理方法では、ラダー処理を実行する第1のプロセッサと、タスク処理を実行する第2のプロセッサとを備え、複数のラダー処理と複数のタスク処理について、処理の優先度に応じてラダー処理とタスク処理の組み合わせを定めておき、処理の優先度ごとに定められた処理周期での割り込みを行い、割り込みに応じて、当該優先度でのラダー処理を第1のプロセッサで行い、当該優先度でのタスク処理を第2のプロセッサで行い、割り込みの優先度ごとの処理周期内に、当該優先度でのタスク処理と、当該優先度でのラダー処理を完了する。

【0024】

また、処理の優先度ごとの処理周期は、処理の優先度が上位であるほど短い処理周期とされている。

10

【0025】

また、処理の優先度ごとの処理周期の期間内に、実行中の優先度よりも上位の優先度の第2の割り込みで発生したときに、第1のプロセッサと第2のプロセッサは、第2の割り込みで定まる優先度のラダー処理とタスク処理を実行し、上位の優先度の第2の割り込みで定まる処理の完了後に、第2の割り込み発生前に実行中の処理を再開し、指定された割り込みの優先度ごとの処理周期内に、当該優先度でのタスク処理と、当該優先度でのラダー処理を完了する。

【0026】

また、第1のプロセッサが、ラダー処理の実行中に異常が発生したことを検知したとき、第2のプロセッサは、当該優先度のタスク処理を実行すると共に、次回割り込み時の当該優先度のラダー処理についての処理の可否を決定する。

20

【0027】

また、第1のプロセッサは、一連のラダー命令に従いラダー処理を実行すると共に、実行できないラダー命令を検出した際には、ラダー処理を一時停止し、第2のプロセッサは、停止したラダー命令をソフト処理により実行し、第1のプロセッサは、ラダー処理を停止したアドレスの次の番地から継続してラダー処理を再開する。

【0028】

また、第1のプロセッサはハードウェアで構成され、第2のプロセッサは、ソフトウェアで構成されている。

【0029】

30

また、割り込みにより指定された同一優先度の第1のプロセッサのラダー処理と、第2のプロセッサのタスク処理は、その処理周期内で一方の処理が完了後に他方の処理を実行する直列処理とされる。

【0030】

また、割り込みにより指定された同一優先度の第1のプロセッサのラダー処理と、第2のプロセッサのタスク処理は、その処理周期内で双方の処理が同時に実行される期間を含む並列処理とされる。

【発明の効果】

【0031】

本発明によれば、リアルタイム演算において複数の組のラダー処理を実行することが可能となる。また本発明の実施例によれば、1つの組のラダー処理が異常となった場合においても、該当のラダー処理のみを停止することで、他の組のラダー処理は影響を受けず継続してラダー処理を実行でき可用性を向上させることができる。

40

【図面の簡単な説明】

【0032】

【図1】図2の処理を実現する為のコントローラの全体構成を示す図。

【図2】コントローラで実現する処理内容の概念を説明するための図。

【図3】ラダー処理プロセッサ3と主メモリ4の各部の詳細を示す図。

【図4】ラダープログラム44内の具体的な記憶内容を示す図。

【図5】コントローラ1全体としての処理手順を示す図。

50

- 【図 6】タイマー種別、優先度、ラダー処理並びにタスク処理の関連を示す図。
【図 7】ラダー処理とタスク処理の直列実行に関する説明図。
【図 8】ラダー処理中に異常発生したときの処理（ラダー処理中段）を示す図。
【図 9】ラダー処理中に異常発生したときの処理（ラダー処理再開）を示す図。
【図 10】ラダー処理とタスク処理の並列実行に関する説明図。
【図 11】ラダー処理とタスク処理の直列実行に関する説明図。
【図 12】タスク処理、ラダー処理の順番の並列実行に関する説明図。
【図 13】上位割り込みのときのラダー処理プロセッサ 3 の設定に関する説明図。
【図 14】ソフト実効命令検出時のソフト実行手順を示す図。
【図 15】マイクロプロセッサでソフト処理を行うときの各部処理内容を示す図。

10

【発明を実施するための形態】

【0033】

以下本発明の実施例を、図面を用いて説明する。

【実施例】

【0034】

最初に、本発明のコントローラで実現する処理の概念について図 2 を用いて説明する。これは、複数の周期で動作するラダー処理を 1 つの環境でマルチタスク動作させるための考え方を示している。

【0035】

ここでは、複数のラダー処理が優先度に応じて区別されている。優先度が 1 位のラダー処理（ラダー 1）に対して、タイマー 1 による割り込み周期 T_1 が設定されており、タイマー 1 による割り込みの都度、優先度 1 位のラダー処理（ラダー 1）が実行される。

20

【0036】

以下同様に、優先度が 2 位のラダー処理（ラダー 2）に対して、タイマー 2 による割り込み周期 T_2 が設定されており、タイマー 2 による割り込みの都度、優先度 2 位のラダー処理（ラダー 2）が実行される。また、優先度が 3 位のラダー処理（ラダー 3）に対して、タイマー 3 による割り込み周期 T_3 が設定されており、タイマー 3 による割り込みの都度、優先度 3 位のラダー処理（ラダー 3）が実行される。

【0037】

なお、タスク処理も優先度に応じて区別されており、各優先度のラダー処理実行後にタスク処理が実行される。また、優先度が高いほど、タイマーによる割り込み周期 T が短く設定されている。

30

【0038】

従って、図 2 において例えば時刻 t_1 から優先度が 3 位のラダー処理（ラダー 3）実行中に、時刻 t_2 においてタイマー 2 による割り込みが発生した場合、これを中断して優先度が 2 位のラダー処理（ラダー 2）の実行に移る。

【0039】

また、この優先度が 2 位のラダー処理（ラダー 2）に続くタスク 2 の実行中に、時刻 t_3 においてタイマー 1 による割り込みが発生した場合、これを中断して優先度が 1 位のラダー処理（ラダー 1）の実行に移る。

40

【0040】

また、優先度が 2 位のラダー処理（ラダー 2）に続くタスク 2 の残余部分の実行は、上位割り込みによるラダー処理（ラダー 1）とこれに続くタスク 1 が完了した時刻 t_4 のタイミング以降に実行される。同様に、優先度が 3 位のラダー処理（ラダー 3）の残余部分とこれに続くタスク 3 の実行は、上位割り込みによるラダー処理（ラダー 2）とこれに続くタスク 2 が完了した時刻 t_5 のタイミング以降に実行される。

【0041】

なお、図 2 の時刻 t_6 から時刻 t_9 の間でも上位割り込みが発生し、この間下位側が待機している状況が発生している。この部分の動作説明については、上記説明から容易に理解できることなので説明を省略する。

50

【 0 0 4 2 】

図 1 は、図 2 の処理を実現する為のコントローラの全体構成を示している。

【 0 0 4 3 】

コントローラ 1 は、マイクロプロセッサ 2、ラダー処理プロセッサ 3、主メモリ 4、並びにこれらを相互に接続するバス 5 で構成される。このうち、主メモリ 4 は、管理ソフトウェア 4 1、タスクプログラム 4 2、タスク管理テーブル 4 3、ラダープログラム 4 4、ラダー管理テーブル 4 5 を含む。

【 0 0 4 4 】

このように図 1 には、マイクロプロセッサ 2 と、ラダー処理プロセッサ 3 を備えており、これらそれぞれのプロセッサは以下の役割を担う。

10

【 0 0 4 5 】

まず、ラダー処理プロセッサ 3 は、ラダー命令で記述されたラダープログラム 4 4 を実行する専用のプロセッサである。本発明では、ハードウェアで命令を解釈し実行する。このために、ラダープログラム 4 4 内のラダー命令はラダー処理プロセッサ 3 に取り込まれ使用される。

【 0 0 4 6 】

他方、マイクロプロセッサ 2 は、ラダープログラム 4 4 で処理できないような複雑な処理（ネットワーク通信処理、I/O デバイス制御処理、二重化構成制御など）を、例えば C 言語を用いてプログラムし、そのプログラムを実行するプロセッサである。

【 0 0 4 7 】

20

マイクロプロセッサ 2 は、基本的に管理ソフトウェア 4 1 である OS が動作している。OS はタスクプログラム 4 2 を用いて、タスク管理テーブル 4 3 内のタスクのマルチタスクスケジューリングを行い、タスクを制御する。このため、タスク管理テーブル 4 3 内のタスクは、マイクロプロセッサ 2 に取り込まれ使用される。

【 0 0 4 8 】

本発明では、このマルチタスクスケジューリングとラダー処理の起動・停止を連携させ動作させる。このためマイクロプロセッサ 2 は、ラダー処理プロセッサ 3 に対して、ラダー処理の実行（起動）と停止を指示する。また、ラダー処理プロセッサ 3 は、指示に応じたラダー処理の完了後に、ラダー処理の終了を報告する。

【 0 0 4 9 】

30

また、ラダー処理プロセッサ 3 は、ラダー命令の実行を監視しており、異常停止したこと、或いは処理遅延などの異常をマイクロプロセッサ 2 に報告する。

【 0 0 5 0 】

また、ラダー処理プロセッサ 3 がハードウェアで実行できないラダー命令を検出した場合には、検出場所でラダー処理実行を停止し、マイクロプロセッサ 2 のソフトウェアに連絡する。連絡を受けたマイクロプロセッサ 2 上のソフトウェアはラダー処理の実行を行い、ラダー処理実行後ラダー処理プロセッサ 3 の再開を指示する。或いは、原因が特定されるまで以降の当該優先度のラダー処理を停止するなどの処理を実行する。

【 0 0 5 1 】

なお、マイクロプロセッサ 2 におけるタスクの実行は、図 2 の例では同一優先度のラダー処理の実行後に行われる。従って、ラダー処理プロセッサ 3 のラダー処理終了報告を受けてから、この優先度のタスク処理が実行される運びになる。

40

【 0 0 5 2 】

本発明のコントローラ 1 においては、上記したようにラダー処理プロセッサ 3 とマイクロプロセッサ 2 が連携している。このように、ラダー処理プロセッサ 3、マイクロプロセッサ 2 の 2 つのプロセッサを持つことにより、C 言語で記述されたプログラムを実行中に、ラダー処理をラダー処理プロセッサで平行して動作させることもできる。並行処理する事例については後で図 10 を用いて説明する。

【 0 0 5 3 】

なお、ラダー処理をラダー処理プロセッサ 3 とハードウェアで実行してもよく、インタ

50

ブリタのようにソフトウェア的に実行することも考えられる。この処理については種々実現可能である。

【 0 0 5 4 】

図 1 のラダー処理プロセッサ 3 と主メモリ 4 の各部の詳細が図 3 に記載されているので、これらの詳細構成を図 3 で説明する。

【 0 0 5 5 】

図 3 において、まずラダー処理プロセッサ 3 の内部には、プログラムカウンタ 3 1、管理テーブル番号 3 2、ワークレジスタ 3 3、ラダー制御レジスタ 3 4、実行ステータス 3 5などを備えている。これらが如何なる機能を果たすのかは、以後の動作説明の中で明らかにする。

10

【 0 0 5 6 】

また図 3 において、主メモリ 4 内のタスクプログラム 4 2 の記憶エリア内には図 2 で説明した優先度別のタスクが記憶されている。ここでは、タスク 1 が 4 2 1、タスク 2 が 4 2 2、タスク 3 が 4 2 3 で表示されている。これらのタスクは C 言語で記述されており、マイクロプロセッサ 2 に読み込まれて使用される。

【 0 0 5 7 】

主メモリ 4 内のタスク管理テーブル 4 3 の記憶エリア内には、タスクプログラム 4 2 内の各タスクの管理テーブルが設けられている。ここでは、タスク 1 の管理テーブルが 4 3 1、タスク 2 の管理テーブルが 4 3 2、タスク 3 の管理テーブルが 4 3 3 で表示されている。

20

【 0 0 5 8 】

また図 3 において、主メモリ 4 内のラダープログラム 4 4 の記憶エリア内には図 2 で説明した優先度別のラダーが記憶されている。ここでは、ラダー 1 が 4 4 1、ラダー 2 が 4 4 2、ラダー 3 が 4 4 3 で表示されている。これらのラダーは、ラダー処理プロセッサ 3 に読み込まれて使用される。

【 0 0 5 9 】

主メモリ 4 内のラダー管理テーブル 4 5 の記憶エリア内には、ラダープログラム 4 4 内の各ラダーの管理テーブルが設けられている。ここでは、ラダー 1 の管理テーブルが 4 5 1、ラダー 2 の管理テーブルが 4 5 2、ラダー 3 の管理テーブルが 4 5 3 で表示されている。

30

【 0 0 6 0 】

ラダープログラム 4 4 内の具体的な記憶内容が図 4 に示されている。ここでは、例えばラダー 3 (L 3) が、複数の一連のラダー命令 (ここでは L 3 1 から L 3 n まで) で構成されていることがわかる。他のラダー L 1、L 2 も同様に複数の一連のラダー命令で構成されている。

【 0 0 6 1 】

また図 2 において、各周期内でのラダー処理では、一時中断されることがあるにしても、この複数の一連のラダー命令の全てが同一周期内で実行される必要がある。従って、各ラダーの管理テーブル 4 5 1、4 5 2、4 5 3 では、複数の一連のラダー命令について現在時点での処理内容、処理順番を記憶管理することになる。

40

【 0 0 6 2 】

なお、図示していないが、タスクプログラム 4 2 内の各タスクも基本的には、図 4 のラダーのように構成されている。例えばタスク 1 は、複数の一連のタスク命令で構成されている。タスク命令についても各周期内でのタスク処理では、一時中断されることがあるにしても、この複数の一連のタスク命令の全てが同一周期内で実行される必要がある。従って、各タスクの管理テーブル 4 3 1、4 3 2、4 3 3 では、複数の一連のタスク命令について、現在時点での処理内容、処理順番を記憶管理することになる。

【 0 0 6 3 】

図 3、図 4 に示すようにラダー処理プロセッサ 3 あるいは主メモリ 4 内各部が構成されている。係る前提の構成において、コントローラ 1 は以下のように作動する。コントロー

50

ラ 1 全体としての処理手順を図 5 に示す。

【 0 0 6 4 】

まずマイクロプロセッサ 2 は、ラダープログラムを先頭から動作させるとき、以下の手順でラダー処理プロセッサ 3 を起動する。

【 0 0 6 5 】

まずステップ S 1 0 0 では、タイマーによる割り込みの受理と内容解析を行い、例えばこれが優先度 3 の割り込みであることを確認する。このことのために、マイクロプロセッサ 2 内には、タイマー種別と、優先度と、このときのラダー処理並びにタスク処理の関連が例えば表形式で記憶されている。

【 0 0 6 6 】

図 6 がこの表の一例であり、タイマー 1 が優先度 1 で、ラダー 1 並びにタスク 1 を実行する。同様にタイマー 2 が優先度 2 で、ラダー 2 並びにタスク 2 を実行し、タイマー 3 が優先度 3 で、ラダー 3 並びにタスク 3 を実行するという関係にある。因みにこの表には、ラダー並びにタスクに関連して、このプログラムを収納する主メモリ 4 内のアドレスが記憶されているのが望ましい。

【 0 0 6 7 】

ステップ S 1 0 1 では、この割り込みが優先度下位の処理実行中に生じた上位割り込みかどうかを確認する。上位割り込みであるときステップ S 1 0 2、上位割り込みでないときにステップ S 1 0 3 の処理に入る。

【 0 0 6 8 】

この場合には優先度 3 の実行なので、上位割り込みではない。ステップ S 1 0 3 において、マイクロプロセッサ 2 はラダー処理プロセッサ 3 起動準備のための各種設定を実行する。

【 0 0 6 9 】

ここでは、最初に主メモリ 4 に格納されているラダープログラム 4 4 の実行すべきアドレス 4 0 1 を、図 3 に示すラダー処理プロセッサ 3 のプログラムカウンタ 3 1 に設定する。例えば、図 4 のラダー L 3 を先頭から動作させようとするときには、これを収納する 4 4 3 内の最初のラダー命令 L 3 1 の格納位置を示すアドレスを設定する。

【 0 0 7 0 】

次に、使用する管理テーブル 4 0 2 の番号（例えばラダー 3 管理テーブル 4 5 3）を、ラダー処理プロセッサ 3 の管理テーブル番号 3 2 に設定する。また、ワークレジスタ 3 3 のクリアを行い、以後の処理に備える。

【 0 0 7 1 】

ステップ S 1 0 4 では、マイクロプロセッサ 2 による以上の設定処理を受けて、ラダー処理プロセッサ 3 は、マイクロプロセッサ 2 に指定されたアドレス 4 0 1 からラダープログラムを読み込み、ラダー処理を実行し、順次の命令（図 4 の複数の一連のラダー命令）を読み込み、ラダー処理を実行する。

【 0 0 7 2 】

一方、マイクロプロセッサ 2 側では引き続き割り込み監視を継続しており、上位割り込みが発生し、ステップ S 1 0 2 の処理が実行されたものとする。このとき、マイクロプロセッサ 2 は、ラダー処理の停止をするために、ラダー処理プロセッサ 3 のラダー制御レジスタ 3 4 に停止指示を行う。

【 0 0 7 3 】

ラダー処理プロセッサ 3 では、ラダー処理（ステップ S 1 0 4）での逐次命令の処理を継続しながら、ステップ S 1 0 5 において停止指示を確認している。この停止指示を受けたときに、ラダー処理プロセッサ 3 がラダー処理実行中である場合、ラダー処理プロセッサ 3 のプログラムカウンタ 3 1 は、図 4 に示すラダー処理の途中で停止する。つまり、下位ラダーを停止し、この状態を保持する。

【 0 0 7 4 】

例えば、図 4 のラダー L 3 を先頭から実行しており、ラダー命令 L 3 m の格納位置まで

10

20

30

40

50

進んだところでラダー制御レジスタ34に停止指示を受けたとすると、プログラムカウンタ31にはこのラダー命令L3mの格納エリアを示すアドレスが設定され、記憶される。係る停止状態は、図2の処理の流れにおいて、時刻t2でラダー3が上位割り込みを受けた状態に相当する。

【0075】

ステップS107では、上位割り込みによる上位のラダー処理、タスク処理が実行される。つまり、図5において、ステップS103の設定から始まり、ラダー処理（ステップS104）、タスク処理（ステップS111）までの一連の上位の処理が新たに実行される。ここでは、この全てが滞りなく処理されて、処理ステップS108に至ったものとする。

10

【0076】

なお、上位割り込み発生時及び上位割り込み回復時のラダー処理プロセッサ3の具体的な内部処理については、後で図13を参照して説明する。

【0077】

この回復状態は、例えば図1の時刻t2から始まったラダー3の停止状態が、時刻t5になりラダー3に対する上位割り込みが解除されたときに相当する。処理ステップS108では、ラダー処理を再開する。

【0078】

このときマイクロプロセッサ2は、ラダー処理の再実行指示を、ラダー制御レジスタ34に設定する。これを受けてラダー処理プロセッサ3はラダー処理を再開する。このときに、停止時にプログラムカウンタ31に記憶されたラダー命令L3mの格納エリアを示すアドレスが参照され、ここから再実行がされることは言うまでもない。ステップS108での再開準備完了後は、ステップS104に移り、残余のラダー処理を継続する。

20

【0079】

なお、ステップS105で、停止が指示されない場合、ラダー命令は図4のラダー終了命令L3nまでラダー処理を実行する。ステップS109において、ラダー終了命令L3nを検出すると、ラダー処理プロセッサ3によるラダー処理は停止し、実行ステータス35にラダー処理終了のステータスを設定して完了する。

【0080】

ステップS110では、これを受けてラダー処理プロセッサ3は、マイクロプロセッサ2に処理の完了を報告する。

30

【0081】

ステップS111でマイクロプロセッサ2は、この優先度のタスクを主メモリ4のタスクプログラム42から導入して、タスク処理を実行する。この場合には、タスク3が実行される。タスク3の実行完了をもって最初に設定された優先度3の割り込みに対するラダーとタスクに関する一連の処理が完了する。

【0082】

図7は、マルチタスクスケジューリングの優先度に従ったラダー処理のスケジューリングの例を示す。この例は、ラダー処理とタスク処理の直列実行を行う場合である。本説明では優先度の若い数値の方が、優先度が高いものとして説明する。タスク421とラダー441が優先度1、タスク422とラダー442が優先度2、タスク423とラダー443が優先度3に設定されている。このラダー処理とタスク処理の組み合わせは、図6に示すように優先度を持たせ関連付けて動作させる。

40

【0083】

そして、より優先度の高いタスクが動作可能状態となったとき、実行中のラダー処理を停止し、優先度の高いタスクと関連付けされたラダー処理を起動する。優先度の高いタスク、ラダー処理の動作が完了した後、割り込まれたラダー処理は中断された番地から継続して処理を実行する。

【0084】

図7の場合には、時刻t10のタイマー3割り込みによりラダー3処理443の実行が

50

開始される。ラダー 3 処理 4 4 3 実行中に、時刻 t_{11} のタイマー 2 割り込みにより、より優先度の高いラダー 2 処理 4 4 2 へ切り替えを行うため、ラダー 3 処理 4 4 3 を停止し、ラダー 2 処理 4 4 2 を実行する。

【0085】

優先度の高いラダー 2 処理後は、引き続き優先度の高いタスク 2 処理 4 2 2 を実行する。その途中に、時刻 t_{12} のタイマー 1 割り込みにより、より優先度の高いラダー 1 処理 4 4 1 へ切り替えを行い、ラダー 1 処理 4 4 1、タスク 1 処理 4 2 1 を行う。時刻 t_{13} のタスク 1 終了後、待ち状態であったタスク 2 処理 4 2 2 を再開する。時刻 t_{14} のタスク 2 処理 4 2 2 の終了後、待ち状態であったラダー 3 処理 4 4 3 を再開し、その後、タスク 3 処理が実行される。

10

【0086】

図 8 のマルチタスクスケジューリングの例では、ラダー処理プロセッサ 3 におけるラダー処理中に、異常発生を検知してこれをマイクロプロセッサ 2 に報告したときの処理を示している。図 5 処理ステップ S 1 1 2 から処理ステップ S 1 1 5 には、この場合の処理フローが記載されている。

【0087】

この図 8 の例では、時刻 t_{20} での割り込みによるラダー 3 の処理 4 4 3 を実行中に、異常発生 4 4 3 1 した。このことは、図 5 の処理ステップ S 1 1 2 で検知されラダー処理プロセッサ 3 がこれを検出して時刻 t_{21} においてマイクロプロセッサ 2 に報告（処理ステップ S 1 1 3）したとする。この場合に、マイクロプロセッサ 2 は、異常発生となったラダー 3 の処理 4 4 3 について次周期実行時、スケジューリングから外し実行を行わない（処理ステップ S 1 1 4）事を決定する。なお、この優先度のタスク処理は、今回も次回以降も引き続き実行する。

20

【0088】

つまり、マイクロプロセッサ 2 でのタスク処理 4 2 3 は異常発生側のラダー処理プロセッサ 3 と独立に実行可能であるので、このタスク処理のみ継続実行する。マイクロプロセッサ 2 は、割り込み時刻 t_{20} から周期 T_3 経過後の割り込み時刻 t_{26} の処理では、ラダー処理 3 をマルチタスクスケジューリングの対象から除外している。

【0089】

このようにすることで、該当のラダー 3 処理 4 4 3 のみを停止することが可能となり、他のラダー処理への影響はなくなり、さらに優先度の高いラダー処理（4 4 1、4 4 2）の定周期性が保障される。

30

【0090】

但し、異常を検出した次周期以降の処理については、図 9 に示すようにラダー 3 処理 4 4 3、タスク 3 処理 4 2 3 を実行することも可能である。この場合には、割り込み周期 T_3 の間にラダー 3 処理 4 4 3 の健全性が確認されていることが望ましい。

【0091】

図 2、図 7、図 8、図 9 は、同一優先度のラダー処理とタスク処理をこの順番で直列に実行する説明図であるが、本発明ではラダー処理プロセッサ 3 とマイクロプロセッサ 2 を有し、それぞれが処理を分担しているため、ラダー処理とタスク処理は同時刻に並列で実行することも可能である。

40

【0092】

図 10 では、各優先度において、ラダー処理とタスク処理を並列に実行している。この場合、同じ優先度のラダー処理とタスク処理は割り込みにより同じタイミングで処理開始するが、それぞれの処理時間長は一般には相違する。このため、上位割り込みによりラダー処理側が中断する場合（時刻 t_{31} での割り込み）と、タスク処理側が中断する場合（時刻 t_{32} での割り込み）とが発生し得る。この場合の中断並びに再開処理は、ラダー処理プロセッサ 3 とマイクロプロセッサ 2 で個別に準備され、実行されることになる。

【0093】

また図 11 で示すように、基本的にはラダー処理とタスク処理をこの順番で直列に実行

50

しながら、上位割り込みが発生した側の処理のみを遅らせ、あるいは中断させることもできる。図 11 の例では、基本的にラダー処理とタスク処理は、この順番で直列に実行している。これは、時刻 t_{40} 、 t_{41} 、 t_{45} 、 t_{49} の処理から明らかであり、優先度の高い処理が存在しない状態ではこの関係が維持されている。

【0094】

ラダー 3 について、優先度の高いラダー 2 の割り込み（時刻 t_{41} ）により、時刻 t_{41} から t_{42} の期間中断して、その後 t_{43} に一連の処理を完了する。図 7 などの例では、ラダー処理終了に続いてタスク 3 が開始されるタイミングになるが、図 11 では優先度の高いタスク 2 の終了まで開始が遅れている。また開始後も、より上位のタスク 1 の処理期間（ t_{46} - t_{47} ）中断する。

10

【0095】

また、図 7 では、タイマー割り込みにより、ラダー処理が実行され、その後タスク処理の実行の順番で実行を行う説明図であるが、この順番に限定する必要はなく、図 12 に示すように、タイマー割り込みにより、タスク処理が実行され、その後ラダー処理の実行を順番で実行してもよい。

【0096】

また、図 7 の説明図において、タイマー割り込みから、複数のラダー処理の実行を行ってもよく、図 12 の説明図において、1 つのタスク処理から、複数のラダー処理の実行を行ってもよい。

【0097】

20

図 5 の処理フローのステップ S107 の説明において、上位割り込み、さらには上位割り込み解除のときのラダー処理プロセッサ 3 の設定の説明を後述することにしていた。図 13 を用いて、優先度の低いラダー 3 処理から、より優先度の高いラダー 2 処理への切り替えを行うときの具体的な処理内容について説明する。図 13 は、基本的に図 3 と同じ回路部分（ラダー処理プロセッサ 3、主メモリ 4）を主体に記載しているが、説明の都合上ラダープログラムのみ記述した点と、ワークレジスタ 33 の内容を記述した点で相違している。

【0098】

ここでは、ラダー 3 処理 443 を実行中、より優先度の高いラダー 2 処理 442 が動作可能状態となったとする。マイクロプロセッサ 2 は、ラダー 3 処理 443 のラダー処理を停止するよう、ラダー制御レジスタ 34 にラダー処理停止の指示を行う。

30

【0099】

ラダー処理停止の指示を受けたラダー処理プロセッサ 3 は、実行中のラダー処理を停止する。停止したときのラダー処理プロセッサ 3 のプログラムカウンタ 31 がアドレス 404（ラダー命令 L_{3m+1} ）を指しているとする。

【0100】

プロセッサ 2 は、プログラムカウンタ 31 とワークレジスタ 33 の内容を管理テーブル番号 32 が指し示すラダー 3 管理テーブル 453 へ退避する。

【0101】

次に実行すべきラダー 2 処理 442 の、管理テーブル番号を管理テーブル番号 32 へ設定し、ラダー 2 管理テーブル 452 からプログラムカウンタ 21、ワークレジスタ 33 を回復し、ラダー 2 処理を実行する環境を整える。ラダー制御レジスタ 34 にラダー処理実行の指示を行うことでラダー 2 処理 442 を実行する。

40

【0102】

ラダー 2 処理の実行終了後、プロセッサ 2 は、実行待ちとなっているラダー 3 処理 423 の、管理テーブル番号を管理テーブル番号 32 へ設定し、ラダー 3 管理テーブル 453 から退避していたプログラムカウンタ 21、ワークレジスタ 33 を回復し、ラダー 3 処理を実行する環境を整える。このときプログラムカウンタは退避していたアドレス 404 を指し、ワークレジスタは演算中のレジスタ値となる。ラダー制御レジスタ 34 にラダー処理実行の指示を行うことでラダー 3 処理 443 をアドレス 404 から再開する。

50

【 0 1 0 3 】

図 1 において、ラダー処理プロセッサ 3 の処理が命令解読不能により実行不可能状態に陥ったときに、これをマイクロプロセッサ 2 に報告し、マイクロプロセッサ 2 がソフト処理することについて説明した。

【 0 1 0 4 】

図 1 4 は、ソフト実効命令の検出時の、ソフト実行手順を示す。ラダー処理プロセッサ 3 は、主メモリ 4 のラダープログラム 4 4 を用いて、優先度 3 のラダー 3 を読み込み実行中に、ラダー命令 L 3 m の解読不能により実行不可能状態に陥ったとする。

【 0 1 0 5 】

ラダー処理プロセッサ 3 は、ソフト実行の命令を検出したときは、アドレス 4 0 5 にてラダー処理を停止し、ラダー命令 L 3 m にこれがソフト実効命令の対象であることと、この時の各種のパラメータを保持する。またマイクロプロセッサ 2 に対して割り込みを行うか、あるいは実行ステータス 3 5 を用いてプロセッサ 2 に対してソフト実効命令の検出を知らせる。

10

【 0 1 0 6 】

マイクロプロセッサ 2 は、ソフト実効命令の連絡を受け、主メモリ 4 のラダープログラム 4 4 のアドレス 4 0 5 からソフト実効命令の読み出し、パラメータの読み出しを行い、プログラムカウンタ 3 1 を 4 0 6 (ラダー命令 L 3 m + 1) にする。そのうえで、ラダー命令 L 3 m に対するソフト実効命令、パラメータによりソフト処理を行い、ソフト処理が終了した時点で、ラダー制御レジスタ 3 4 に起動指示を行い、再びアドレス 4 0 6 (ラダー命令 L 3 m + 1) からラダー処理を継続する。

20

【 0 1 0 7 】

図 1 5 は、マイクロプロセッサ 2 におけるソフト処理を行うときの各部処理内容を示す図である。この図で、横方向にはマイクロプロセッサ 2、ラダー処理プロセッサ 3、主メモリを示し、縦方向には時間を示している。

【 0 1 0 8 】

この図によれば、マイクロプロセッサ 2 は時刻 t 1 0 0 の割り込みを受けて、ラダー処理プロセッサ 3 を起動 (時刻 t 1 0 1) する。以後ラダー処理プロセッサ 3 は、メモリ 4 内のラダー命令を逐次取り込んで処理を進める。この間の処理では、ラダー処理プロセッサ 3 とメモリ 4 の間でラダー演算データの入出力を行っている。

30

【 0 1 0 9 】

これに対し、実行できないラダー命令が生じると、ラダー処理プロセッサ 3 は時刻 t 1 0 2 でマイクロプロセッサ 2 に対してソフト処理を要求する。マイクロプロセッサ 2 は、改めてメモリ 4 にアクセスし、メモリ 4 の間でラダー演算データの入出力を行ない、ラダー命令のソフト処理を実行する。ラダー命令のソフト処理を完了すると、時刻 t 0 3 においてソフト処理完了をラダー処理プロセッサ 3 に通知する。

【 0 1 1 0 】

ラダー処理プロセッサ 3 は、再度メモリ 4 内のラダー命令を逐次取り込んで処理を進める。このときに再開するラダー命令は、中断し、マイクロプロセッサ 2 でソフト処理されたラダー命令の次の順番のラダー命令とされる。このために、再開以後の処理では、再度ラダー処理プロセッサ 3 とメモリ 4 の間でラダー演算データの入出力を行っている。

40

【 0 1 1 1 】

ラダー処理プロセッサ 3 は、ラダー終了命令までの一連のラダー処理を実行完了した時点 t 1 0 4 でラダー処理完了をマイクロプロセッサ 2 に報告し、以後マイクロプロセッサ 2 における同じ優先度レベルのタスク処理を実行する。

【 0 1 1 2 】

タスク処理では、マイクロプロセッサ 2 は、改めてメモリ 4 にアクセスし、メモリ 4 の間でタスク処理データの入出力を行ない、タスク命令のソフト処理を実行する。

【 0 1 1 3 】

以上述べたように、本発明ではラダー処理プロセッサ 3 とマイクロプロセッサ 2 にタス

50

クとラダーの処理を分散し、かつ優先度に応じて処理するタスクとラダーの組み合わせを変更したことにより、コントローラの同一環境下でマルチタスクスケジューリングが可能である。

【符号の説明】

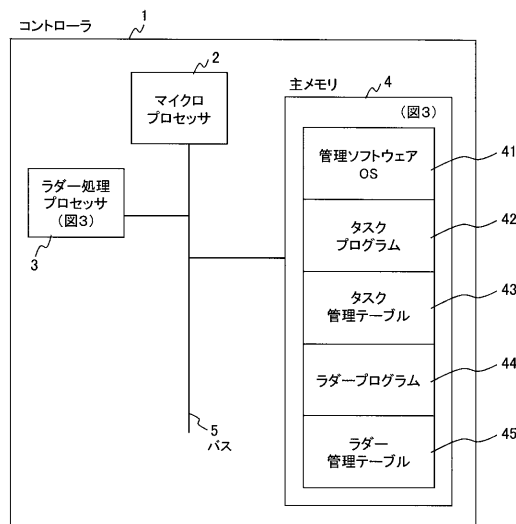
【 0 1 1 4 】

- 1 : コントローラ
- 2 : マイクロプロセッサ
- 3 : ラダー処理プロセッサ
- 4 : 主メモリ
- 5 : バス
- 4 1 : 管理ソフトウェア
- 4 2 : タスクプログラム
- 4 3 : タスク管理テーブル
- 4 4 : ラダープログラム
- 4 5 : ラダー管理テーブル

10

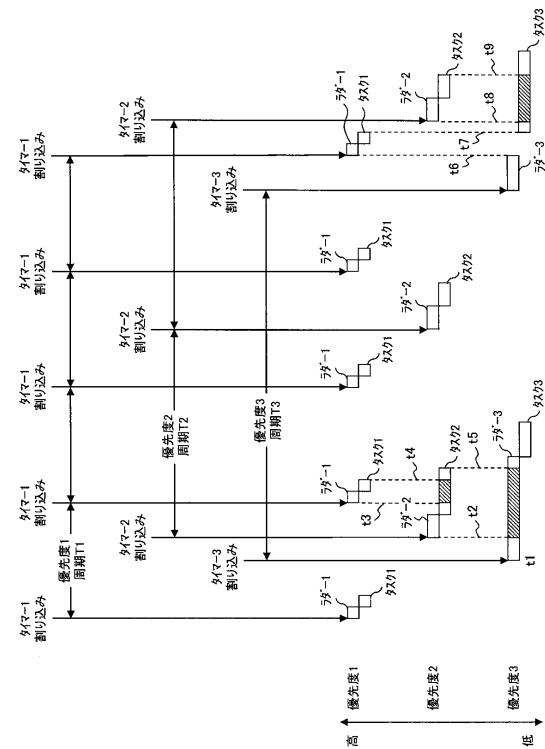
【図 1】

図 1

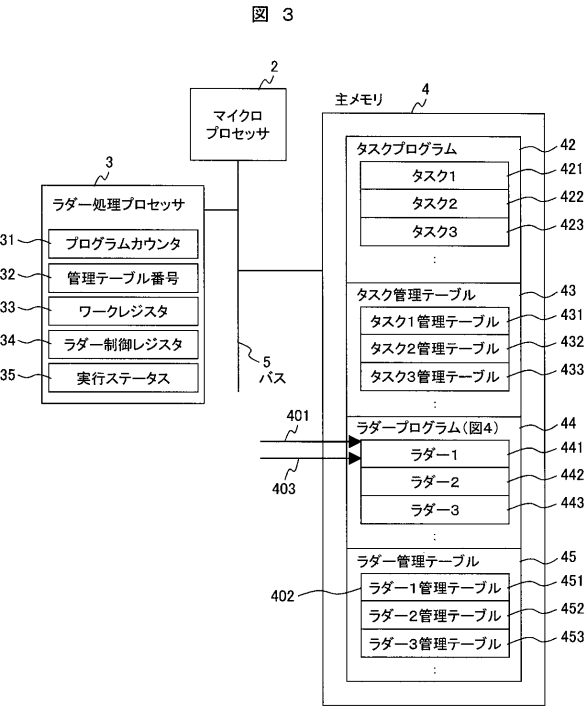


【図 2】

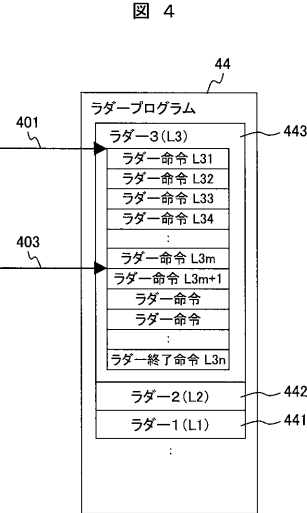
図 2



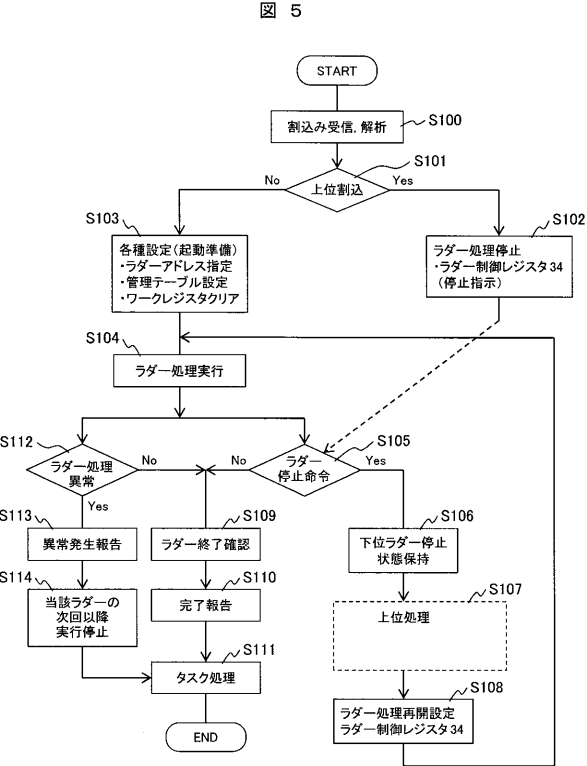
【図 3】



【図 4】



【図 5】



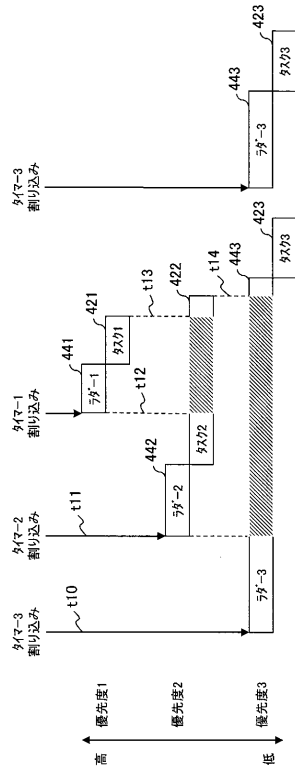
【図 6】

図 6

タイマー種別	優先度	処理内容	
		ラダー	タスク
タイマー1	1	ラダー1	タスク1
タイマー2	2	ラダー2	タスク2
タイマー3	3	ラダー3	タスク3

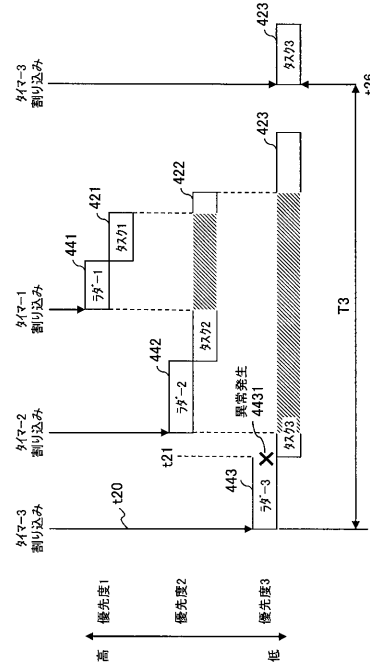
【図 7】

図 7



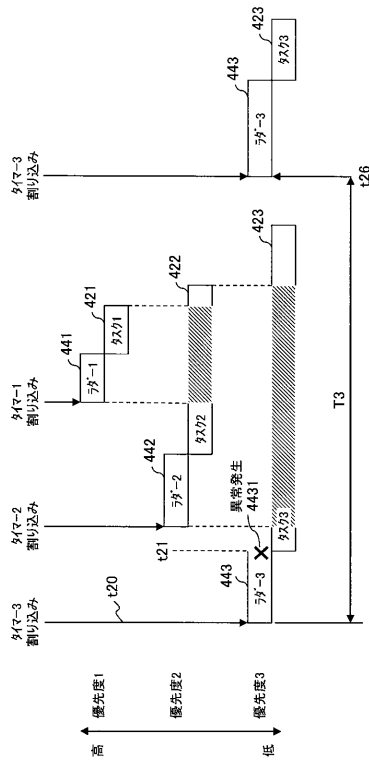
【図 8】

図 8



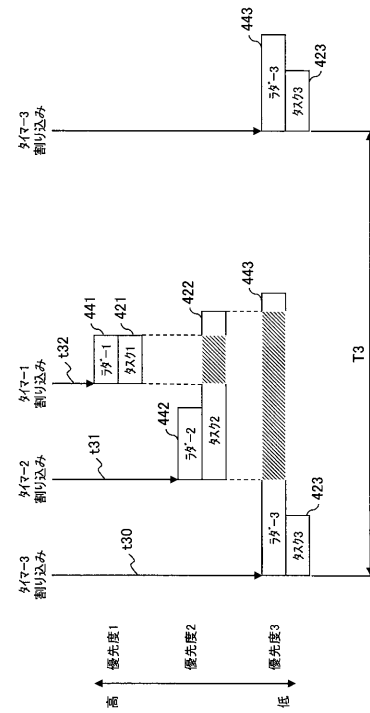
【図 9】

図 9

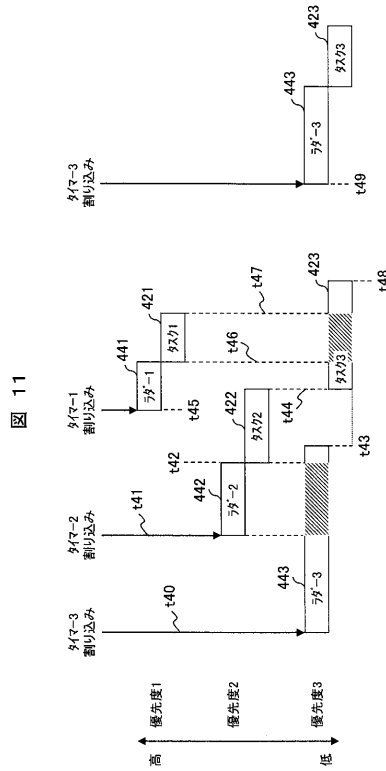


【図 10】

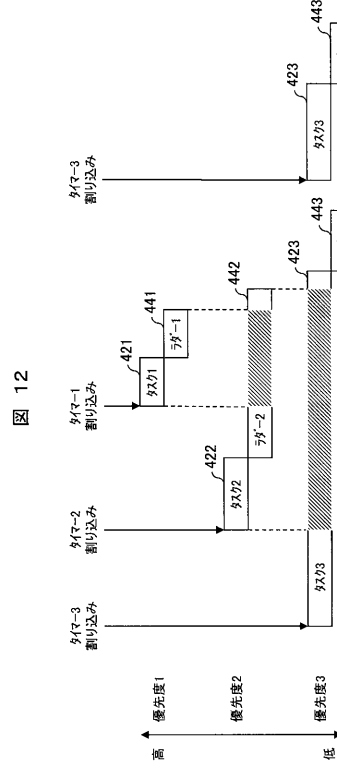
図 10



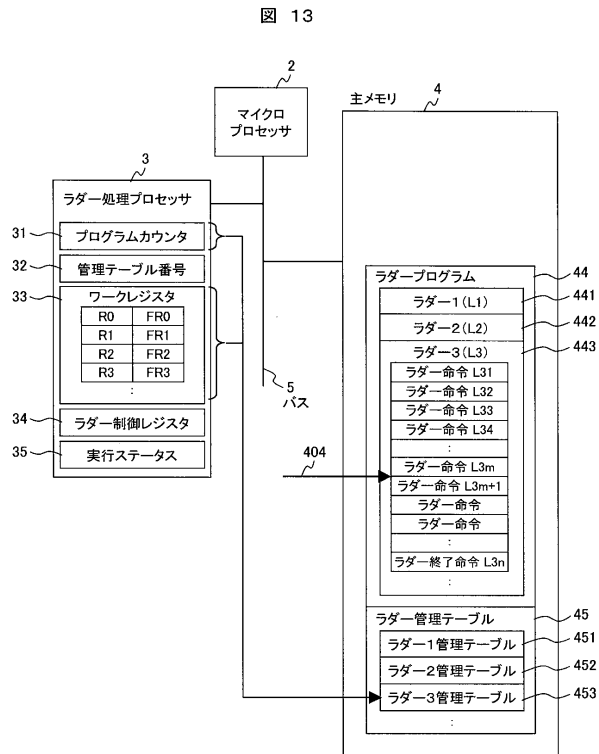
【図 11】



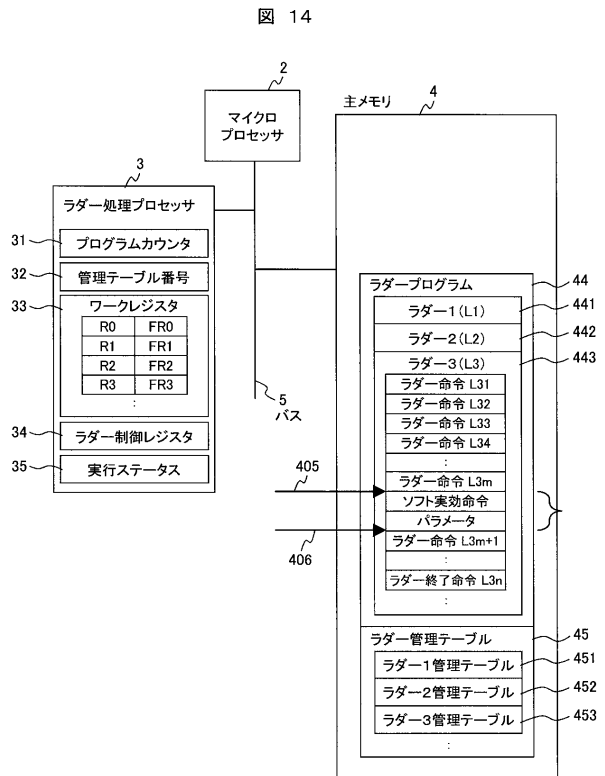
【図 12】



【図 13】

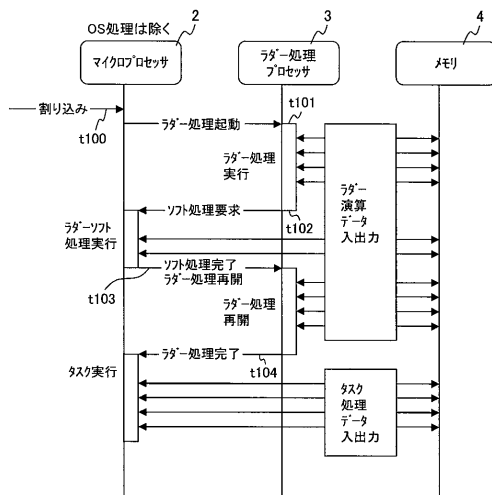


【図 14】



【図 15】

図 15



フロントページの続き

(72)発明者 清水 勝人

茨城県日立市大みか町五丁目2番1号 株式会社日立製作所 情報制御システム社内

審査官 青山 純

(56)参考文献 特開2001-005505(JP,A)

特開昭63-049902(JP,A)

特開平02-311903(JP,A)

特開平03-237502(JP,A)

特開2009-223417(JP,A)

(58)調査した分野(Int.Cl., DB名)

G05B 19/05