

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号
特開2012-123719
(P2012-123719A)

(43) 公開日 平成24年6月28日 (2012.6.28)

(51) Int.Cl.	F I	テーマコード (参考)
GO6F 9/32 (2006.01)	GO6F 9/32 360A	5B005
GO6F 12/08 (2006.01)	GO6F 12/08 523C	5B033
	GO6F 12/08 571	
	GO6F 12/08 573	
	GO6F 12/08 515P	

審査請求 未請求 請求項の数 6 O L (全 17 頁) 最終頁に続く

(21) 出願番号	特願2010-275719 (P2010-275719)	(71) 出願人	000005108
(22) 出願日	平成22年12月10日 (2010.12.10)		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	100064414
			弁理士 磯野 道造
		(74) 代理人	100111545
			弁理士 多田 悦夫
		(72) 発明者	中三川 哲明
			茨城県日立市大みか町七丁目1番1号 株
			式会社日立製作所日立研究所内
		(72) 発明者	上脇 正
			茨城県日立市大みか町七丁目1番1号 株
			式会社日立製作所日立研究所内

最終頁に続く

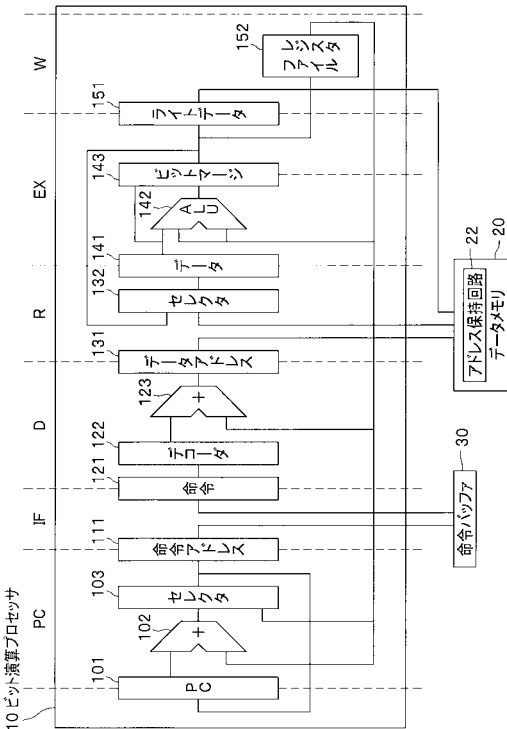
(54) 【発明の名称】 プログラマブルコントローラ

(57) 【要約】

【課題】ラダー言語用のビット演算プロセッサを備えるプログラマブルコントローラにおいて多発する、リードモディファイライトによるパイプライン処理の中断を防ぐ。

【解決手段】リードモディファイライトの対象データをバッファレジスタにロードするとともに対象データのアドレスをレジスタファイル152に保持しておくパイプラインステージR（リードステージ）の後ろに、ビット演算及びビットデータのマージを行うパイプラインステージEX（実行ステージ）を設け、その後ろにリードステージRにて保持したアドレスにマージ結果をストアするパイプラインステージW（ライトステージ）を設ける。

。 【選択図】 図1



【特許請求の範囲】**【請求項 1】**

ビット演算処理の対象となる 1 ビットのデータを複数個まとめたワードの単位でメモリの読み書きを行うプログラマブルコントローラであって、

プログラムに含まれるビット演算処理命令列を、パイプライン処理機構によって並列に実行するビット演算プロセッサを備え、

前記ビット演算プロセッサが備えるパイプラインステージは、演算対象となるデータをワード単位でメモリから読み込むリードステージの次に演算ステージがあり、前記演算ステージの次に、前記演算ステージによって演算されたビット演算の結果を含むワードデータを前記リードステージで読み込んだデータと同じアドレスに書き込むライトステージがある

ことを特徴とするプログラマブルコントローラ。

【請求項 2】

前記メモリは、

少なくとも 2 ウェイ以上のセットアソシアティブ方式又はフルアソシアティブ方式のキャッシュメモリであり、

前記リードステージにて読み込んだキャッシュエントリのインデックス情報及びウェイ情報を、前記ライトステージまで保持するアドレス保持回路を備える

ことを特徴とする請求項 1 に記載のプログラマブルコントローラ。

【請求項 3】

前記ビット演算プロセッサは、

前記ビット演算処理命令のなかのビットデータのストア命令を実行するときに、

前記リードステージにてストア対象のビットを含む元のワードデータを読み込んで記憶部に保持しておき、

前記演算ステージにて演算された前記ビット演算の結果のビットデータを、前記保持した元のワードデータにマージする

ことを特徴とする請求項 1 または請求項 2 に記載のプログラマブルコントローラ。

【請求項 4】

前記メモリは、単一のパイプラインステージ処理サイクル時間内に 1 回以上のリードと 1 回以上のライトとを行うことができる 2 ポート以上のメモリである

ことを特徴とする請求項 1 または請求項 2 に記載のプログラマブルコントローラ。

【請求項 5】

キャッシュヒット判定を前記リードステージで行い、前記ライトステージでは前記アドレス保持回路に保持された前記インデックス情報及びウェイ情報によって示されるキャッシュエントリに、前記ビット演算の結果を含むワードデータがマージされたラインデータを書き込む

ことを特徴とする請求項 2 に記載のプログラマブルコントローラ。

【請求項 6】

ストアするアドレス範囲によってライトスルーモードとライトバックモードを切り替える

ことを特徴とする請求項 2 に記載のプログラマブルコントローラ。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、鉄鋼、電力、上下水道等のプラントや各種機械に対してシーケンス制御を高速に行うビット演算プロセッサを搭載したプログラマブルコントローラに関する。

【背景技術】**【0002】**

プログラマブルコントローラにおいては、従来からシーケンス制御を効率よく記述できるラダー言語が使われている。ラダー言語では、スイッチの開閉状態等、1 ビットで表わ

10

20

30

40

50

される情報を入力とし、リレー（継電器）出力等、1ビットで表わされる情報を出力とする場合が多い。そのため、プログラマブルコントローラは、ラダー言語特有の1ビットデータ処理を高速に行うため、専用のビット演算プロセッサを搭載することが多い。

【0003】

ビット演算プロセッサは、1ビットのデータを扱うのに適した専用の命令セットをサポートしているが、演算結果を格納する先は汎用のメモリ素子であるため、8ビットや16ビットといったメモリ素子のアクセス単位サイズに合わせなければならない。汎用のメモリ素子に対して1ビットのデータを書き込むには、リードモディファイライト動作、即ち、メモリをアクセス単位サイズでリードし、リードしたデータの一部を変更し、再びアクセス単位サイズでライトする動作、が必要になるため、ラダー言語処理では汎用言語処理よりもメモリアクセス回数が多くなる特徴がある。

10

【0004】

ビット演算プロセッサの高速化に関連する従来技術として、特許文献1には、過去複数回分のワード単位のメモリアクセスアドレスとデータを高速アクセスが可能なバッファに記憶しておき、アクセスしようとするビットを含むワードデータのアドレスがバッファに記憶されているアドレスと一致する場合には、メモリでなくバッファ上のデータを使用することによりメモリアクセス回数を減らす工夫が記載されている。

【0005】

一方、汎用コンピュータの分野では、メモリアクセスの高速化のために従来からキャッシュメモリが使用されてきたが、近年の半導体微細化に伴い、キャッシュメモリ用内部SRAM（Static Random Access Memory）のソフトエラー対策が必要になり、キャッシュメモリにECC（Error Check and Correct：誤り検出・訂正符号）を付加して保護する場合も多くなってきている。ECCの付加単位は4バイト程度で設計されることが多いため、キャッシュメモリへの書き込み単位も4バイト以上になってしまう。汎用コンピュータの命令でも1バイトや2バイトの書き込みがサポートされるが、このようなECC付加単位以下のサイズの書き込みを行おうとするとキャッシュメモリに対してリードモディファイライトを行うことになる。

20

【0006】

単純な構成のキャッシュメモリでリードモディファイライトを行う場合、キャッシュメモリが2サイクル以上占有され、その間プロセッサのパイプライン処理が中断されてしまう。リードモディファイライトがあってもパイプライン処理を中断させないための工夫としては、複数段のストアバッファを設け、ストア処理を置き去りにすることが行われている。特許文献2には、ECCを付加したキャッシュメモリのストアバッファの構成が記載されている。なお、ストアバッファによるストア処理の置き去りは、ロード・ストア命令以外を実行する際のキャッシュメモリの空き時間を利用してストア処理を行うものであるため、パイプライン処理の中断を確定的に無くすものではなく、中断の確率をできるだけ減らす工夫である。

30

【0007】

また、特許文献3には、キャッシュメモリを多ポート化し、オペランドフェッチ（リード）のパイプラインステージとオペランドライトのパイプラインステージを分けることにより、メモリ間データ転送を行う命令を効率よく実行するための工夫が記載されているが、キャッシュミス時の動作やオペランドフェッチとライトのアドレス競合時の動作については記載されていないし、リードモディファイライトの処理にも全く触れられていない。

40

【先行技術文献】

【特許文献】

【0008】

【特許文献1】特開平11-39160号公報

【特許文献2】国際公開第2007/088597号

【特許文献3】特開平4-40524号公報

【発明の概要】

50

【発明が解決しようとする課題】

【0009】

ラダー言語処理用のビット演算プロセッサにおいても高速化のためにパイプライン処理やキャッシュメモリを採用することが考えられるが、前述のようにラダー言語処理ではリードモディファイライトが多発するため、リードモディファイライトによるパイプライン処理のストール（中断）を防ぐ必要がある。特許文献2のようにアドレス比較器の付いた多段のストアバッファを使用すれば、パイプライン処理が中断する確率を減らすことができるが、ラダー言語で記述されるプログラムは汎用言語に比べてビット単位でのストア命令の比率が高い傾向があり、ストアバッファの段数をより大きくしなければならず、実現に必要な回路規模が大きくなってしまう。

10

【0010】

一方、現在のほとんどの汎用プロセッサの命令セットは、RISC（Reduced Instruction Set Computer：縮小命令セットコンピュータ）の思想が取り入れられているため、演算処理とロード・ストア処理とはそれぞれ異なる命令で実行されるようになっている。即ち、演算処理は演算命令によって汎用レジスタ間で行い、メモリとレジスタ間のロード・ストアは別の命令で行うようになっている。このため、汎用プロセッサ用のパイプラインは、レジスタ間演算に適した構成になっているが、ラダー言語処理はメモリとアキュムレータ間の演算が中心であり、汎用プロセッサ用のパイプライン構成は必ずしも処理効率のよいものではない。

【0011】

20

従来技術によるパイプライン及びメモリの構成によってラダー言語処理を行った場合の課題を具体例によって示す。図14に典型的なRISCプロセッサのパイプライン構成を示す。PC（Program Counter）は命令アドレスを計算するステージ、IF（Instruction Fetch）は命令をフェッチするステージ、D（Decode）はフェッチした命令をデコードするステージ、EX（Execute）はレジスタ間演算またはメモリアドレス計算を行うステージ、M（Memory）はメモリをリード又はメモリにライトするステージ、WB（Write Back）は演算結果又はリード値をレジスタに書き戻すステージである。ビットデータ演算に伴うリードモディファイライトを行う場合には、MステージをM1とM2の2つに分け、M1サイクルでリードを、M2サイクルでビットデータのマージとライトを行う。

【0012】

30

ここで、図7（a）に例示したラダー図を前記従来技術のパイプライン構成で実行する場合を考える。図7（b）は、図7（a）に例示したラダー図を通常のラダー言語の命令列に変換したプログラムの例である。また、図7（c）は、それを前記従来技術のパイプライン構成で実行するための命令列に変換したプログラムの例である。

【0013】

変数X0～X3は入力、変数Y2～Y3は出力を表し、それぞれ1ビット長のデータとしてメモリ内の2つのワードデータ内に記憶されているものとする。図8はこれらの変数がメモリに割り付けられて格納される様子を示している。図7（b）において、LD（Load）はメモリからアキュムレータへのロード命令、ANDはメモリからロードした値とアキュムレータ値の論理積をアキュムレータに記憶する命令、ST（Store）はアキュムレータ値をメモリにストアする命令、ORはメモリからロードした値とアキュムレータ値の論理和をアキュムレータに記憶する命令である。また、図7（c）においては、LDはメモリからレジスタへのロード命令、ANDは2つのレジスタ値の論理積をレジスタに記憶する命令、STはレジスタ値をメモリにストアする命令、ORは2つのレジスタ値の論理和をレジスタに記憶する命令である。

40

【0014】

図15は、図7（c）に示した命令列を図14のパイプライン構成で実行した場合の動作を示すタイムチャートである。図15に示すように、1番目及び2番目のLD命令はそれぞれサイクルt0及びt1から開始され、サイクルt5及びt6までそのまま実行され、レジスタR1とR2にデータがロードされる。3番目のAND命令はサイクルt2から

50

t 4 まではそのまま実行されるが、サイクル t 5 の E X ステージで必要な変数 X 1 のデータがまだリードされていないためパイプラインが 1 サイクルの間ストールする（「-」印）。変数 X 1 のデータは 2 番目の L D 命令の M ステージの実行が完了する次のサイクル t 6 で利用可能となり、その後はサイクル t 8 までそのまま実行される。4 番目の S T 命令はサイクル t 3 から開始されるが、サイクル t 5 での前命令のストールに伴い一緒にストールする。続いてサイクル t 6 及び t 7 が実行されたのち、M ステージではサイクル t 8 と t 9 との 2 サイクルをかけてリードモディファイライトが実行される。そのため、5 番目の L D 命令は、サイクル t 5 に加えてサイクル t 9 でもストールする。

【 0 0 1 5 】

以下同様にして、6 番目の L D 命令はサイクル t 5 及び t 9 が、7 番目の O R 命令及び 8 番目の S T 命令はサイクル t 9 及び t 1 1 が、それぞれストールすることになる。また、8 番目の S T 命令はサイクル t 1 4 と t 1 5 の 2 サイクルをかけてリードモディファイライトが実行される。以上説明したように、従来技術のパイプライン構成では、通常のラダー言語では 6 命令で記述される処理を実行するのに 8 命令を要し、かつ、3 つのサイクルにてパイプラインのストールが発生するため、全ての処理を完了するまでに計 1 7 サイクルを要することとなる。

【 0 0 1 6 】

本発明の目的は、ラダー言語を処理するビット演算プロセッサを備えたプログラマブルコントローラに好適な、リードモディファイライトなどによるパイプライン処理のストールを起こさないパイプラインの構成を提案することにある。さらには、キャッシュメモリを備えるビット演算プロセッサにおいて好適なパイプラインとキャッシュメモリの構成を提案することを目的とする。

【課題を解決するための手段】

【 0 0 1 7 】

ラダー言語を処理するビット演算プロセッサにおいて、パイプライン処理のストールを起こさないためには、E X ステージの演算結果を次のサイクルで実行される次命令の E X ステージで利用可能とするとともに、リードモディファイライトの対象となるメモリの内容を事前にリードする新たなステージである R (Read) ステージを設け、その次の E X ステージにてビット演算及びビットデータのマージを行い、その結果を最後の W (Write) ステージにてメモリにストアするようにすればよい。

【 0 0 1 8 】

そこで、前記の目的を達成するために、本発明は、ビット演算処理の対象となる 1 ビットのデータを複数個まとめたワードの単位でメモリの読み書きを行うプログラマブルコントローラであって、プログラムに含まれるビット演算処理命令列を、パイプライン処理機構によって並列に実行するビット演算プロセッサを備え、前記ビット演算プロセッサが備えるパイプラインステージは、ビット演算対象となるデータをワード単位でメモリから読み込むリードステージの次に演算ステージがあり、前記演算ステージの次に、前記演算ステージによって演算されたビット演算の結果を含むワードデータを前記リードステージで読み込んだデータと同じアドレスに書き込むライトステージがあることを特徴とする。

【 0 0 1 9 】

また、本発明は、前記メモリが、少なくとも 2 ウェイ以上のセットアソシアティブ方式又はフルアソシアティブ方式のキャッシュメモリであり、前記リードステージにて読み込んだキャッシュエントリのインデックス情報及びウェイ情報を、前記ライトステージまで保持するインデックス保持レジスタ及びウェイ保持レジスタを備えることを特徴とする。

【 0 0 2 0 】

また、メモリにアクセスする命令が連続した場合でもパイプライン処理をストールさせないために、前記メモリは単一のパイプライン処理サイクル時間内に 1 回以上のリードと 1 回以上のライトが独立に行える 2 ポート以上のメモリであることが好ましい。

【 0 0 2 1 】

なお、特許文献 3 には、キャッシュリード、演算、キャッシュライト、という順番のパ

10

20

30

40

50

パイプライン構成が記載されているが、リードとライトを独立に行うことが目的であるため、リードとライトには別なアドレスを指定でき、そのためキャッシュライト時にもキャッシュヒット判定が行われる。すなわち、リードモディファイライトを行うことは考慮されておらず、2リード+1ライトの3ポートメモリが必要となるなど、本発明とは目的と構成が異なっている。

【発明の効果】

【0022】

本発明によれば、ビット演算プロセッサを備えたプログラマブルコントローラにおいて実行されるプログラムにおいて、リードモディファイライトに伴うパイプライン処理のストールが起こらないため、ラダー言語によって記述されたプログラムを効率よく処理することができる。また、キャッシュメモリを備えたビット演算プロセッサにおいても同様にパイプライン処理のストールが起こらないため、ラダー言語によって記述されたプログラムを効率よく処理することができる。また、ストア命令の実行時に先行するリードステージでキャッシュヒット判定を行うことにより、ライトステージでのキャッシュ入れ替え処理が不要となるので、パイプライン制御が単純になる。

10

【図面の簡単な説明】

【0023】

【図1】第1実施形態に係るプログラマブルコントローラが備えるビット演算プロセッサの内部構成及びパイプライン構成を示す図

20

【図2】第1実施形態に係るプログラマブルコントローラの全体構成を示す図

【図3】データメモリとして2ポートメモリを用いた場合の構成を示すブロック図

【図4】第1実施形態に係るビット演算プロセッサの演算ステージの詳細構成を示す図

【図5】第2実施形態に係るキャッシュメモリの構成を示す図

【図6】第2実施形態に係るビット演算プロセッサの演算ステージの詳細構成を示す図

【図7】ビット演算プロセッサの動作を説明するためのラダー図とそのプログラム例

【図8】図7のプログラムで使用される変数のメモリへの割り付けを示す図

【図9】ビット演算プロセッサによって実行される命令の命令フォーマットを示す図

【図10】図7のプログラムに対する命令コード列がメモリに格納された様子を示す図

【図11】第1及び第2実施形態に係るビット演算プロセッサの動作例を示すタイムチャート

30

【図12】第2実施形態に係るビット演算プロセッサにおいてキャッシュミスが発生した場合の動作例を示すタイムチャート

【図13】第1実施形態に係るパイプライン構成を図14と対比して表現した図

【図14】典型的なRISCプロセッサのパイプライン構成を示す図

【図15】従来技術によるビット演算プロセッサの動作例を示すタイムチャート

【発明を実施するための形態】

【0024】

以下、本発明を実施するための形態につき、図1から図13を用いて説明する。

《第1実施形態》

図1は、本発明の第1実施形態に係るプログラマブルコントローラが備えるビット演算プロセッサの内部構成及びパイプライン構成を示す図である。また、図13は、第1実施形態に係るパイプライン構成を図14の従来技術と対比して表現したものである。図1及び図13に示すように、本実施形態に係るパイプライン構成は、(1)プログラムカウンタ(PC)ステージ、(2)命令フェッチ(IF)ステージ、(3)デコード(D)ステージ、(4)メモリリード(R)ステージ、(5)演算実行(EX)ステージ、(6)メモリライト(W)ステージ、の6段からなる。

40

【0025】

PCステージは、直前の命令アドレスを示すPC(プログラムカウンタ)101の値に定数「1」又は指定されたレジスタ値を加算する加算器102、加算結果又は分岐先アドレスを示すレジスタ値を選択するセレクタ103を備え、直前に選択された命令の次に実

50

行すべき命令の命令アドレスを、当該ステージの出力として命令アドレスレジスタ 1 1 1 にセットする。I F ステージは命令アドレスレジスタ 1 1 1 にセットされた命令アドレスに対応する命令を命令バッファ 3 0 から読み出し、命令レジスタ 1 2 1 にセットする。

【 0 0 2 6 】

D ステージは、読み出した命令を解釈するデコーダ 1 2 2、デコーダ 1 2 2 によって命令から抜き出されたアドレスに指定されたレジスタ値を加算する加算器 1 2 3 を備え、デコードした命令がメモリアクセスを伴う場合はデータアドレスを生成してデータアドレスレジスタ 1 3 1 にセットする。また、図示は省略しているが、デコード結果に応じてレジスタ選択や演算機能選択などその後のステージの制御に必要な制御情報を取り出す。

【 0 0 2 7 】

R ステージは、データアドレスレジスタ 1 3 1 で示されたデータアドレスに該当するデータをデータメモリ 2 0 から読み出す。読み出されたデータは次の E X ステージからのバイパスデータとの選択を行うセレクタ 1 3 2 を介してワードバッファ 1 4 1 にセットされる。また、データアドレスレジスタ 1 3 1 で示されたデータアドレスを W ステージで再度利用できるように、データメモリ 2 0 内に備えるアドレス保持回路 2 2 にセットする。

【 0 0 2 8 】

E X ステージは、A L U (Arithmetic Logic Unit : 演算器) 1 4 2、及びビットマージ機構 1 4 3 をもち、ワードバッファ 1 4 1 にセットされたデータ値及び / 又は指定されたレジスタ値を使って命令で指示された演算を行う。このとき、ビットデータの演算を行う場合は、演算対象のワードデータのなかから指定されたビット位置のビットデータを抽出して演算を実行し、ビットマージ機構 1 4 3 を使用して当該ビット位置に演算結果のビットデータを埋め込んでワードデータを生成する。結果を格納する先がレジスタの場合はレジスタファイル 1 5 2 に書き込みを行い、格納する先がデータメモリ 2 0 の場合は格納すべき演算結果のデータをライトバッファ 1 5 1 に書き込む。

【 0 0 2 9 】

W ステージは、R ステージにてアドレス保持回路 2 2 にセットされたデータアドレス、つまり R ステージでデータを読み出したのと同じデータメモリ 2 0 内のデータアドレスにライトバッファ 1 5 1 にセットされたデータを格納する。

【 0 0 3 0 】

図 2 は、第 1 実施形態に係るプログラマブルコントローラの全体構成を示す図である。図 2 に示すように、プログラマブルコントローラ 1 0 0 0 は、C P U (Central Processing Unit) モジュール 1、I / O (Input / Output) モジュール 2 A 及び 2 B、それらを接続する I / O バス 3、C P U モジュール 1 に着脱可能に接続されるプログラム入力装置 4 を備えて構成される。I / O モジュール 2 A、2 B は、それぞれ I / O バス接続回路と I / O インターフェース回路を有しており、必要な I / O の仕様と接点数に応じて種類と数を変えられるようになっている。

【 0 0 3 1 】

C P U モジュール 1 は、ビット演算プロセッサ 1 0、データメモリ 2 0、命令バッファ 3 0、I / O バス制御回路 4 0、メモリコントローラ 5 0、外部 R A M (Random Access Memory) 6 0、R O M (Read Only Memory) 7 0、汎用マイクロプロセッサ 8 0、通信 I / F (Interface) 9 0 を備えて構成される。

【 0 0 3 2 】

データメモリ 2 0 に格納される所定アドレス範囲のデータ群は、それぞれが I / O モジュール 2 A、2 B に接続された外部機器 (図示省略) との間でやり取りされる入力データ又は出力データに対応する。I / O バス制御回路 4 0 は、I / O バス 3 を制御し、I / O モジュール 2 A、2 B に接続された外部機器から得られる入力データをデータメモリ 2 0 に書き込み、また、データメモリ 2 0 から読み出した出力データを I / O バス 3 経由で I / O モジュール 2 A、2 B に接続された外部機器に出力する。

【 0 0 3 3 】

命令バッファ 3 0 は、ビット演算プロセッサ 1 0 から要求される命令アドレスに対する

10

20

30

40

50

命令がバッファ内に蓄えられていればそれを返し、当該命令がバッファ内にない場合にはメモリコントローラ50に外部RAM60からの命令の読み込みを要求する。メモリコントローラ50は、ビット演算プロセッサ10、命令バッファ30、I/Oバス制御回路40、汎用マイクロプロセッサ80からの要求に応じて外部RAM60のリードライト又はROM70のリードを行う。また、汎用マイクロプロセッサ80は、通信IF90を介してプログラム入力装置4からローディングされるラダープログラムを外部RAM60に書き込む等、プログラマブルコントローラ1000全体の制御を司る。汎用マイクロプロセッサ80を動作させるプログラムはROM70に格納されている。

【0034】

なお、本実施形態においては、ビット演算プロセッサ10、データメモリ20、命令バッファ30、I/Oバス制御回路40、及びメモリコントローラ50は、システムLSI (Large Scale Integration) 100に内蔵されているものとする。

【0035】

図3は、本実施形態に係るデータメモリ20として好適な2ポートメモリを用いた場合の構成を示す図である。図3に示すように、データメモリ20は、アドレスセクタ201、アドレス保持レジスタ221及び222を有するアドレス保持回路22、ライトアドレスセクタ215、ライトデータセクタ204、メモリアレイ21を備えて構成される。

【0036】

アドレスセクタ201は、ビット演算プロセッサ10のデータアドレスレジスタ131 (図1) の値とI/Oバス制御回路40から出力されるI/Oデータアドレスのいずれかを選択する。アドレス保持レジスタ221, 222は、Rステージの実行に使用されたデータアドレスを2サイクル後のWステージまで保持する。すなわち、Rステージの出力データとして1段目のアドレス保持レジスタ221にデータアドレスを保持し、その値をEXステージの出力データとして2段目のアドレス保持レジスタ222にコピーして保持することによって、Wステージで使えるようにする。

【0037】

ライトアドレスセクタ215は、データアドレスレジスタ131の値とアドレス保持レジスタ222の値のいずれかを選択する。ライトデータセクタ204は、ライトバッファ151の値とI/Oバス制御回路40から出力されるI/Oデータの値のいずれかを選択する。

【0038】

メモリアレイ21は、リード用のポート1とライト用のポート2の2つが単一のサイクル内でアクセス可能な2ポートメモリで構成される。そのため、パイプライン処理されるある命令のRステージの処理と並列に実行される別命令のWステージの処理とを同一サイクル内に並列に実行することが可能となる。ここでは、メモリアレイ21は2ポートメモリとしたが、単一のサイクル内で1以上のリード用ポートと1以上のライト用ポートがアクセス可能な3ポート以上のメモリとしてもよい。なお、I/Oバス制御回路40からのメモリアクセス時にはリードとライトはそれぞれ独立して行うので、アドレス保持回路22は使用されない。

【0039】

図4は、第1実施形態に係るビット演算プロセッサの演算ステージ (EXステージ、一部他ステージの要素を含む) の詳細構成を示す図である。図4に示すように、演算ステージは、ALU142の他に、ALU142の入力を選択するセクタ144及び145、データメモリ20からリードされた1ワード (16ビット) のデータもしくはこの演算ステージの演算結果を次命令で再び演算に使用するためのバイパスデータのいずれかを選択するセクタ132 (Rステージの要素)、その選択されたワードデータを保持するワードバッファ141、演算結果であるALU142の出力を入力されたワードデータにマージするビットマージ機構143、ビットマージされたデータを保持するライトバッファ151、アキュムレータや汎用レジスタなどのデータを保持するレジスタファイル152 (

10

20

30

40

50

Wステージの要素)を備えて構成される。A L U 1 4 2 は 1 ビット長又は 1 ワード即ち 1 6 ビット長の演算を行うことができる。

【 0 0 4 0 】

図 9 は、ビット演算プロセッサ 1 0 によって実行される命令の命令フォーマットを示す図である。図 9 に示すように、命令には 2 つの形式があるが、いずれも 3 2 ビット固定長である。命令形式 1 は、5 ビットの命令コード (O P : Operation)、4 ビットのビット位置フィールド (B A : Bit Address)、2 3 ビットのワードアドレスフィールド (W A : Word Address) からなり、ビットデータを対象とする。命令形式 2 は、5 ビットの命令コード (O P)、4 ビットのレジスタ指定フィールド (R A : Register Address)、2 3 ビットのワードアドレスフィールド (W A) からなり、ワードデータを対象とする。

10

【 0 0 4 1 】

次に、図 7 及び図 8 に示したラダープログラムを例にとり、図 1 を参照しつつ、本実施形態に係るビット演算プロセッサ 1 0 の動作を図 1 1 のタイムチャートを使って説明する。図 1 0 は、図 7 (b) のラダープログラムに対する前記した命令形式 1 の命令コード列がプログラムメモリ (外部 R A M 6 0) に格納された様子を示したものであり、アドレス A 番地から順に命令コード列が格納されているものとする。

【 0 0 4 2 】

先ず、1 番目の L D 命令は、サイクル t 0 から開始され、サイクル t 5 までそのまま実行される。このとき、サイクル t 0 (P C ステージ) では、L D 命令のアドレスが計算されて命令アドレスレジスタ 1 1 1 にセットされる。サイクル t 1 (I F ステージ) では、命令バッファ 3 0 から当該命令がリードされて命令レジスタ 1 2 1 にセットされる。サイクル t 2 (D ステージ) では、命令がデコードされ、L D 命令であることが認識される。1 番目の L D 命令では、リードすべきワードアドレスは命令中で直接指定されているので、該当するワードアドレスがデータアドレスレジスタ 1 3 1 にセットされる。サイクル t 3 (R ステージ) では、データメモリ 2 0 から、対象となるビットデータを含む 1 ワード分のデータが読み込まれ、ワードバッファ 1 4 1 にセットされる。サイクル t 4 (E X ステージ) では、対象となる変数 X 0 のビットが抜き出されてレジスタファイル 1 5 2 に含まれるアキュムレータにセットされる。サイクル t 5 (W ステージ) では何の動作も行わない (図の網かけ表示部分) 。

20

【 0 0 4 3 】

2 番目の A N D 命令は、サイクル t 1 から開始され、サイクル t 6 までそのまま実行される。なお、サイクル t 6 (W ステージ) では何の動作も行わない。このとき、P C ステージから R ステージまでは 1 番目の L D 命令と同様に実行される。サイクル t 5 (E X ステージ) では、対象となる変数 X 1 のビットを抜き出したのちに A L U 1 4 2 でアキュムレータの内容との論理積を計算し、再びアキュムレータにセットする。本実施例では E X ステージの前に実行される R ステージにおいて、変数 X 1 を含むワードデータがリードされるので、従来技術の例 (図 1 5) のように A N D 命令の E X ステージでパイプライン処理がストールすることはない。

30

【 0 0 4 4 】

3 番目の S T 命令は、サイクル t 2 から開始されサイクル t 7 までそのまま実行される。P C ステージから R ステージまでは 1 番目の L D 命令と同様に実行される。S T 命令においても R ステージにて事前にデータのリードを行うことが本実施形態の大きな特徴である。サイクル t 6 (E X ステージ) では、アキュムレータ値 (1 ビット) が読み出され、ワードバッファ 1 4 1 の値 (1 ワード) にマージされた結果がライトバッファ 1 5 1 にセットされる。サイクル t 7 の W ステージでは、データメモリ 2 0 にライトバッファ 1 5 1 の内容が格納される。その際、格納先のメモリアドレスとしては、R ステージで使用した値を保持しているアドレス保持回路 2 2 に記憶された内容が使用される。このように、本実施形態のパイプライン構成によれば、1 ビットデータの書き込みを行う際においてもパイプライン処理のストールは発生しない。

40

【 0 0 4 5 】

50

以下同様の動作により、4番目のLD命令はサイクルt3からサイクルt8にかけて、5番目のOR命令はサイクルt4からサイクルt9にかけて、6番目のST命令はサイクルt5からサイクルt10にかけて、それぞれパイプライン処理がストールすることなく実行される。このように、本実施形態によれば、6命令を実行する間にパイプライン処理のストールは全く発生せず、図7(b)に示した6命令で記述される処理を実行するのに要するサイクル数を、従来技術の17サイクルから11サイクルに短縮することができる。

【0046】

以上説明したように、第1実施形態に係るビット演算プロセッサを備えたプログラマブルコントローラによれば、ラダー言語処理に特有のビット演算処理をパイプライン処理のストールなしに1サイクルピッチで実行できるので、ラダー言語によって記述されたプログラムを高速に実行することができる。

【0047】

《第2実施形態》

続いて、第1実施形態におけるデータメモリ20(図2参照)をキャッシュメモリによって構成する本発明の第2実施形態について説明する。図5は、第2実施形態に係るキャッシュメモリの構成を示す図である。図5に示すように、データメモリ20としてのキャッシュメモリ20Aは、2ウェイセットアソシアティブ方式のキャッシュメモリであり、アドレスセクタ201A、インデックス保持レジスタ221A及び222A、ウェイセクタ203、ライトデータセクタ204A、ウェイ0タグメモリ205、ウェイ1タグメモリ206、LRU(Least Recently Used)メモリ207、ウェイ0データメモリ208、ウェイ1データメモリ209、ヒット判定回路210、ライトバック制御回路211、ウェイデータセクタ212、ウェイ保持レジスタ213及び214を備えて構成される。図1のアドレス保持回路22は、インデックス保持レジスタ221A、222Aを有するアドレス保持回路1(22A)と、ウェイ保持レジスタ213、214を有するアドレス保持回路2(22B)によって構成される。

【0048】

ウェイデータメモリ208及び209は、それぞれ256個のエントリを持ち、各エントリはそれぞれ16バイト(8ワード)のデータを保持する。キャッシュメモリのアクセス単位は16バイトであり、この16バイト単位のデータを1ラインと呼ぶ。タグメモリ205及び206は、それぞれ256個のエントリを持ち、各エントリは、有効ビットV(Valid)、ダーティビットD(Dirty)及びタグアドレス(図5では「タグ」と略記)からなる。LRUメモリ207は256個のエントリを持ち、最近使用されたウェイを保持する。

【0049】

アドレスセクタ201Aは、ビット演算プロセッサ10のデータアドレスレジスタ131(図1)の値とI/Oバス制御回路40から出力されるI/Oデータアドレスのいずれかを選択し、そのなかから12ビットのタグアドレスと8ビットのインデックス値を抽出する。インデックス保持レジスタ221A、222Aは、Rステージの実行に使用されたデータアドレスから抜き出されたインデックス値の部分(8ビット)を2サイクル後のWステージまで保持する。同様に、ウェイ保持レジスタ213、214は、RステージにてヒットしたウェイをWステージまで保持する。

【0050】

ウェイセクタ203は、2段目のインデックス保持レジスタ222Aに記憶されたインデックス値と2段目のウェイ保持レジスタ214に記憶されたヒットウェイを、Wステージにて書き込みを行うインデックス値とウェイとしてウェイデータメモリ208、209に出力する。ライトデータセクタ204Aは、後記するライトバッファ151Aの値とI/Oバス制御回路40から出力される1ライン分のI/Oデータの値のいずれかを選択する。

【0051】

10

20

30

40

50

ヒット判定回路 210 は、アクセスするデータアドレスの上位 12 ビットとタグメモリ 205, 206 から出力されるタグ値をウェイトごとに比較し、ヒット又はミス进行判定する。ウェイデータセクタ 212 は、ヒット判定回路 210 の出力によってウェイ 0 データ又はウェイ 1 データのいずれかのラインデータを選択する。なお、I/O バス制御回路 40 からのキャッシュアクセス時にはキャッシュのリードとライトをそれぞれ独立して行うため、アドレス保持回路 1, 2 (22A, 22B) は使用されない。

【0052】

ライトバック制御回路 211 は、キャッシュミス時に LRU メモリ 207 の出力が示す追い出しウェイ (最近使用されたウェイでない方のウェイ) の対象エントリがライトバックモードの対象となるアドレス範囲に含まれていてダーティであった場合、つまり、データの更新が行われていた場合は、タグと同時に読み出されているデータをメモリコントローラ 50 に送って外部 RAM 60 (図 2) の記憶内容を更新させる。なお、I/O データアドレスに相当する所定のアドレス範囲については、動作をライトスルーモードに切り替えて直ちに記憶内容の更新を行わせるので、ライトバック動作は行わない。また、不図示のリード制御回路により、当該追い出した対象エントリにキャッシュミスとなったラインデータを読み込む。

10

【0053】

なお、これらキャッシュメモリ 20A を構成するタグメモリ 205, 206 及びウェイデータメモリ 208, 209 は、単一サイクル内でリード用のポートとライト用のポートの 2 つが並列にアクセス可能な 2 ポート以上のメモリで構成することが好ましく、それによってメモリへのアクセスが競合することによって生じるパイプライン処理のストールをなくすることができる。また、ここではキャッシュメモリ 20A は 2 ウェイセットアソシアティブ方式としたが、3 ウェイ以上のセットアソシアティブ方式やフルアソシアティブ方式としてもよい。

20

【0054】

図 6 は、第 2 実施形態に係るビット演算プロセッサの演算ステージ (EX ステージ、一部他ステージの要素を含む) の詳細構成を示す図である。図 6 に示すように、この演算ステージ構成は、前記の第 1 実施形態に係る演算ステージ構成 (図 4) における入力及び出力データのサイズを 1 ライン長 (16 バイト = 128 ビット) に変更するとともに、1 ライン長のデータを保持しておいて演算結果のワードデータをマージするための手段を付加したものであり、その他の構成要素は図 4 と同様であるので重複する説明を省略する。

30

【0055】

セクタ 132A (R ステージの要素) は、キャッシュメモリ 20A からリードされた 1 ライン長 (16 バイト) のデータのなかの 1 ワード (16 ビット) もしくはこの演算ステージの演算結果を次命令で再び演算に使用するためのバイパスデータのいずれかを選択する。ラインバッファ 146 は、リードされた 1 ライン長のデータを保持する。また、ワードマージ機構 147 は、ビットマージ機構 143 の出力である演算結果のワードデータをラインバッファ 146 に保持されているラインデータにマージする。ライトバッファ 151A はマージされた 1 ライン分のデータを保持する。

40

【0056】

ここで、前記と同様に、図 7 及び図 8 に示したラダープログラムを例にとり、本実施形態に係るビット演算プロセッサの動作を説明する。キャッシュメモリ 20A がリード動作とライト動作を単一のサイクルで実行可能な 2 ポートメモリによって構成されており、かつ、対象データが全てキャッシュヒットするものと仮定した場合の動作は、前記した図 11 のタイムチャートと同様である。本実施形態の動作が前記の第 1 実施形態と異なるのは、R ステージ及び W ステージにおけるデータのリード及びライトが 1 ラインの単位で行われ、EX ステージでは、セクタ 132A によって抽出された対象ワードに対して演算を実行し、演算結果のワードデータを、ラインバッファ 146 に保持しておいたラインデータにマージしてライトするラインデータを生成する点にある。

50

【0057】

次に、図 7 (b) のラダープログラムにおける 3 番目の S T 命令でキャッシュミスが起こった場合の動作を図 1 2 のタイムチャートを使って説明する。1 番目の L D 命令と 2 番目の A N D 命令の動作は図 1 1 の場合と同じである。3 番目の S T 命令の R ステージ (サイクル t 5) でキャッシュミスが発生すると、メモリコントローラ 5 0 を介して外部 R A M 6 0 のデータを読み込むために、例えばサイクル t 5 からサイクル t 1 4 までパイプライン処理がストールする。しかし、サイクル t 1 5 で R ステージが再開されたのちは、パイプライン処理のストールなしに実行される。このように S T 命令に対するキャッシュミスが発生した場合であっても、R ステージにおいてキャッシュラインの入れ替えを行うだけで済むのでキャッシュの制御が単純になる。

【 0 0 5 8 】

10

なお、S T 命令の次命令がアクセスするデータアドレスが S T 命令と同一インデックス値を有しており、かつ、次命令がキャッシュミスとなった場合でも、2 ウェイセットアソシアティブ以上のキャッシュメモリであれば、当該 S T 命令が W ステージでラインデータを上書きする前に当該エントリがパージされてしまっていることはない。

【 0 0 5 9 】

以上説明したように、第 2 実施形態に係るキャッシュメモリを内蔵したビット演算プロセッサを備えたプログラマブルコントローラによれば、ラダー言語処理に特有のビット演算処理をパイプライン処理のストールなしに 1 サイクルピッチで実行できるので、ラダー言語によって記述されたプログラムを高速に実行することができる。

【 0 0 6 0 】

20

以上にて本発明を実施する形態の説明を終えるが、本発明の実施の態様はこれに限られるものではなく、本発明の趣旨を逸脱しない範囲内で各種の変更が可能である。

【 符号の説明 】

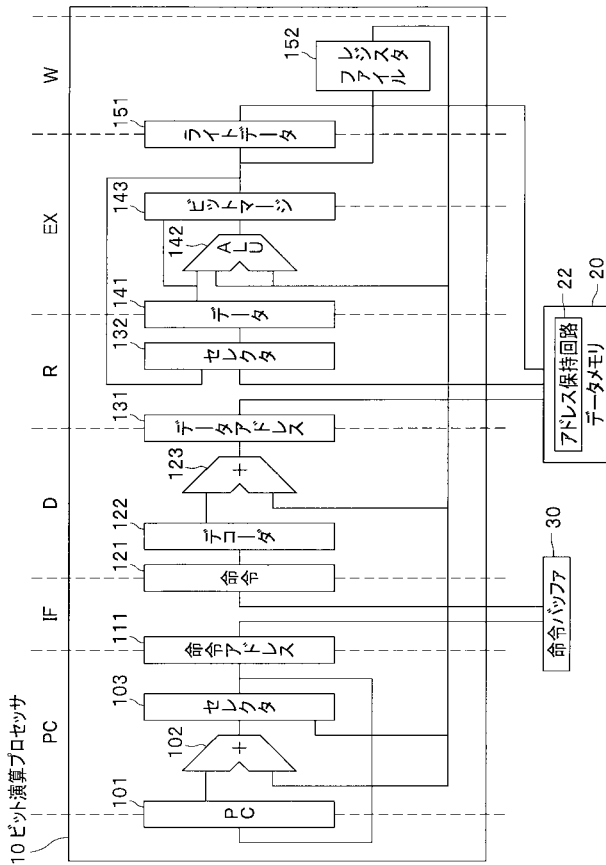
【 0 0 6 1 】

- 1 CPU モジュール
- 2 A , 2 B I / O モジュール
- 3 I / O バス
- 4 プログラム入力装置
- 1 0 ビット演算プロセッサ
- 2 0 データメモリ
- 2 0 A キャッシュメモリ
- 2 1 メモリアレイ
- 2 2 , 2 2 A , 2 2 B アドレス保持回路
- 3 0 命令バッファ
- 4 0 I / O バス制御回路
- 5 0 メモリコントローラ
- 6 0 外部 R A M
- 7 0 R O M
- 8 0 汎用マイクロプロセッサ
- 9 0 通信 I / F
- 1 0 0 システム L S I
- 1 0 0 0 プログラマブルコントローラ

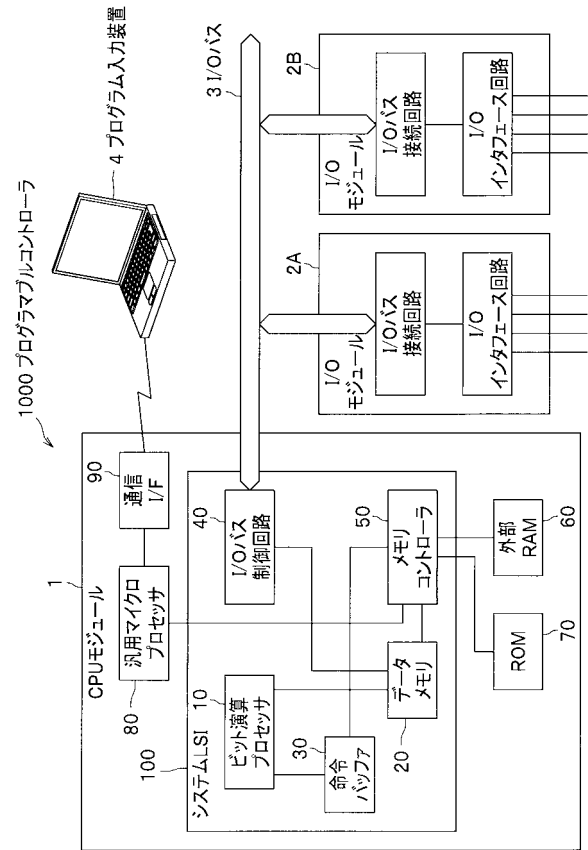
30

40

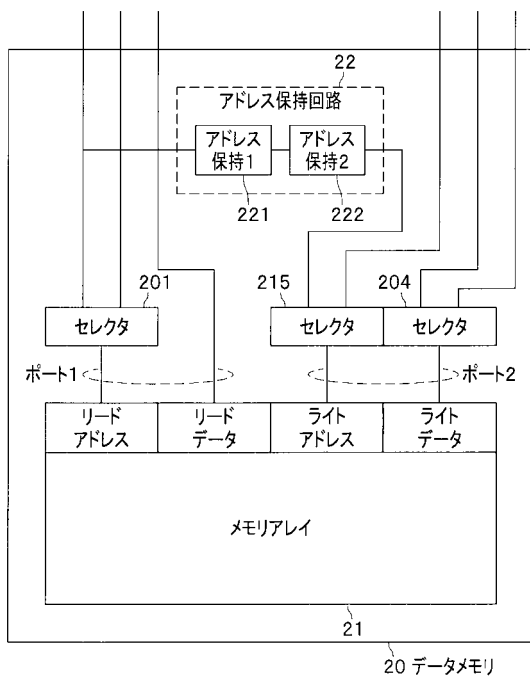
【図 1】



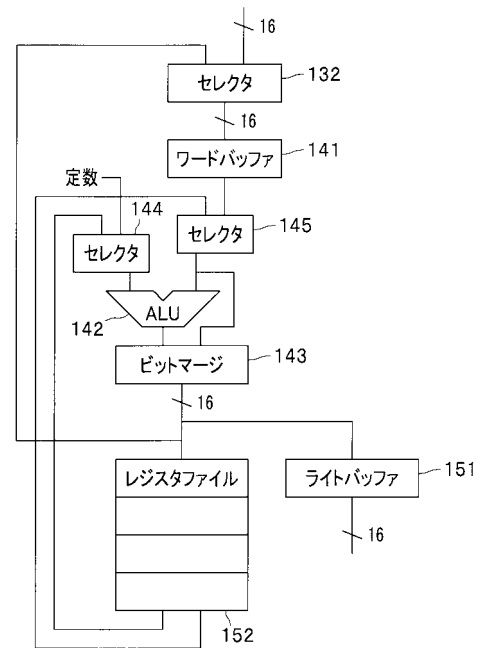
【図 2】



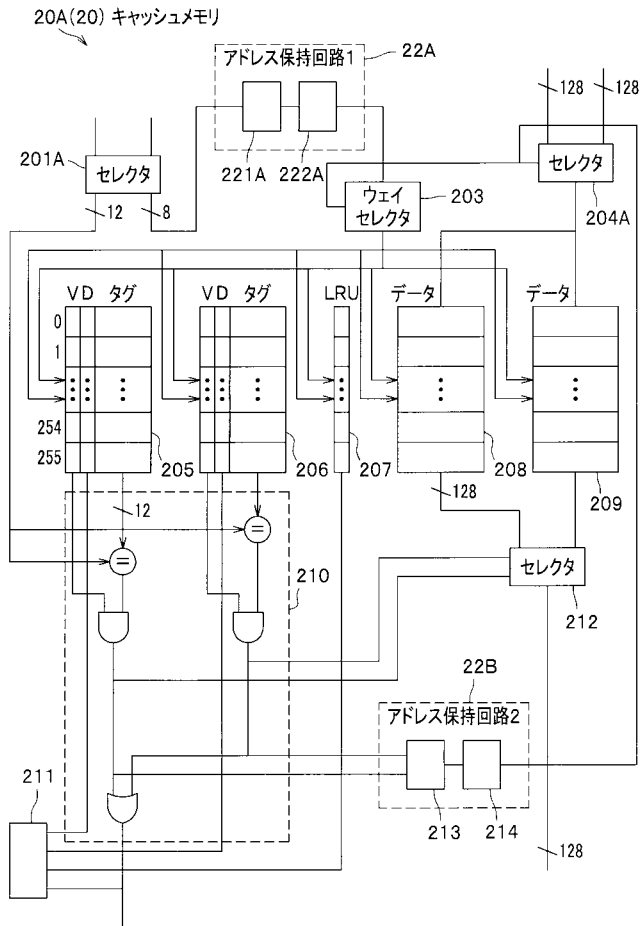
【図 3】



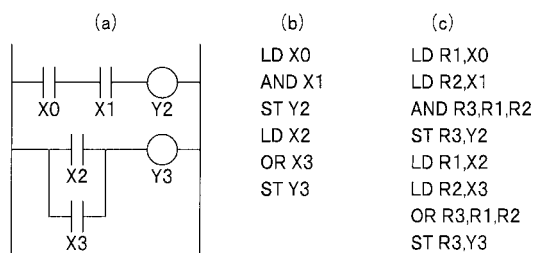
【図 4】



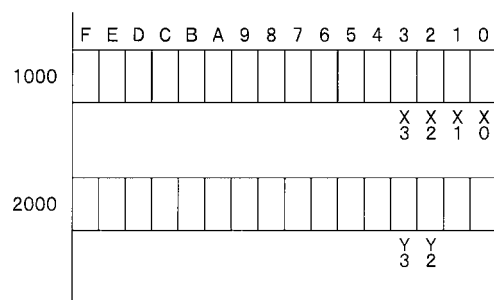
【図 5】



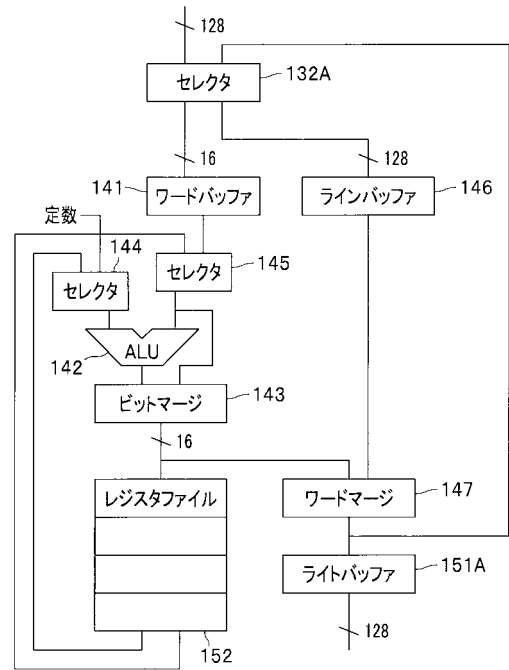
【図 7】



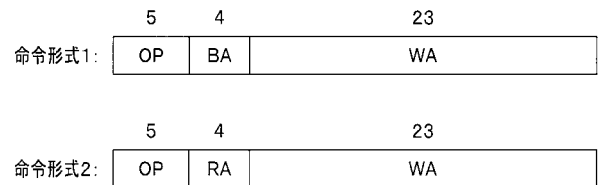
【図 8】



【図 6】



【図 9】



【図 10】

A:	LD	0	1000
A+1:	AND	1	1000
A+2:	ST	2	2000
A+3:	LD	2	1000
A+4:	OR	3	1000
A+5:	ST	3	2000

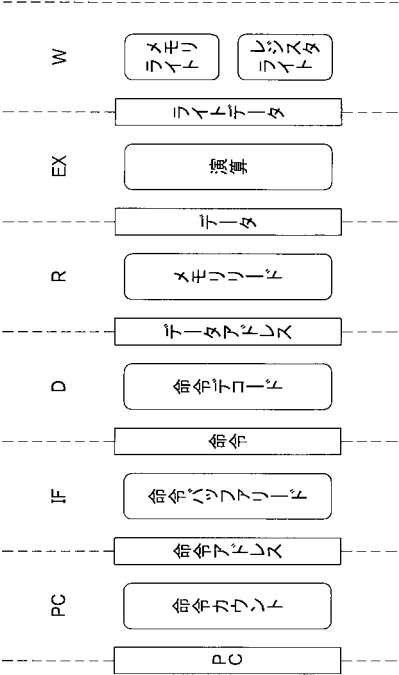
【図 1 1】

	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
LD X0	PC	IF	D	R	EX	W					
AND X1		PC	IF	D	R	EX	W				
ST Y2			PC	IF	D	R	EX	W			
LD X2				PC	IF	D	R	EX	W		
OR X3					PC	IF	D	R	EX	W	
ST Y3						PC	IF	D	R	EX	W

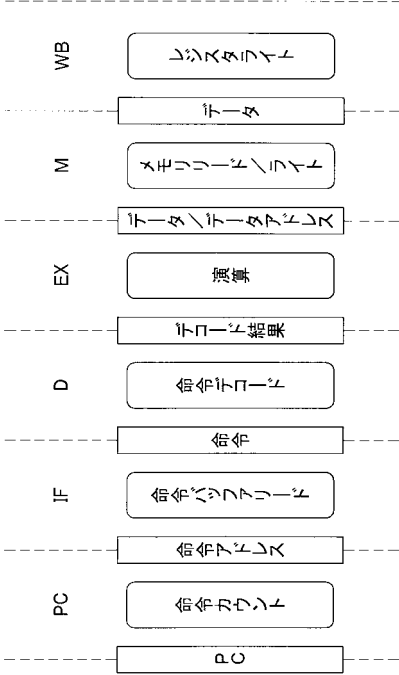
【図 1 2】

	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20
LD X0	PC	IF	PC																		
AND X1		PC	IF	PC																	
ST Y2			PC	IF	PC																
LD X2				PC	IF	PC															
OR X3					PC	IF	PC														
ST Y3						PC	IF	PC													

【図 1 3】



【図 1 4】



【 図 1 5 】

	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16
LD R1X0	PC	IF	D	EX	M	WB											
LD R2X1		PC	IF	D	EX	M	WB										
AND R3R1,R2			PC	IF	D	-	EX	M	WB								
ST R3Y2			PC	IF	D	-	D	EX	M1	M2	WB						
LD R1X2				PC	IF	-	IF	D	EX	-	M	WB					
LD R2X3						-	PC	IF	D	-	EX	M	WB				
OR R3R1,R2								PC	IF	-	D	-	EX	M	WB		
ST R3Y3									PC	-	IF	-	D	EX	M1	M2	WB

フロントページの続き

(51)Int.Cl. F I テーマコード(参考)
G 0 6 F 12/08 5 1 1 E

(72)発明者 山田 勉
茨城県日立市大みか町七丁目1番1号 株式会社日立製作所日立研究所内
(72)発明者 白石 雅裕
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 大谷 辰幸
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
Fターム(参考) 5B005 JJ12 KK12 MM01 NN01 TT02 UU41
5B033 CA24