(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2012-18435
(P2012-18435A)
(43)公開日　平成24年1月26日(2012.1.26)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 9/45	(2006.01)	G06F 9/44	322G	5B081

審査請求 未請求　請求項の数 4　ＯＬ　（全 14 頁）

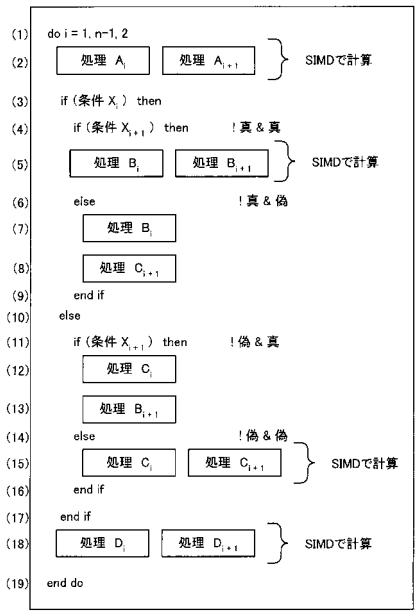
(21)出願番号　特願2010-153530 (P2010-153530)
(22)出願日　平成22年7月6日 (2010.7.6)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100094330
弁理士　山田　正紀
(74)代理人　100109689
弁理士　三上　結
(72)発明者　鈴木　清文
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　竹重　和明
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

Ｆターム(参考)　5B081 CC32

(54)【発明の名称】コンパイル装置およびコンパイルプログラム

(57)【要約】
【課題】　コンパイル装置等に関し、ループ内の判定文の下で実行される処理についてＳＩＭＤ化による演算装置の高速化が図られたオブジェクトプログラムを作成する。
【解決手段】　ＳＩＭＤ化可能と判定されたループについて、該ループ内に、条件に応じて異なる処理を実行させる判定文が存在する場合の該判定文の下で実行される処理のうち、処理対象の複数データについて該判定文により同一条件と判定される場合にＳＩＭＤ処理を実行させるＳＩＭＤ命令を生成し、該判定文により異なる条件と判定される場合に逐次的な処理を実行させる命令を生成する。
【選択図】　図３



【特許請求の範囲】
【請求項１】
プログラム言語を用いて作成されたソースプログラムをコンピュータ上で実行可能なオブジェクトプログラムに変換するコンパイルを行なうコンパイル装置であって、
コンパイル対象の、ループを含むプログラム内の各ループについて、複数のデータに対して一度の命令で同じ処理を実行させるＳＩＭＤ化が可能なループであるか否かを判定するＳＩＭＤ化解析部と、
前記ＳＩＭＤ化解析部でＳＩＭＤ化可能と判定されたループについて、該ループ内に、条件に応じて異なる処理を実行させる判定文が存在する場合の該判定文の下で実行される処理のうち、処理対象の複数データについて該判定文により同一条件と判定される場合にＳＩＭＤ処理を実行させるＳＩＭＤ命令を生成し、該判定文により異なる条件と判定される場合に逐次的な処理を実行させる命令を生成するＳＩＭＤ命令生成部とを有することを特徴とするコンパイル装置。
【請求項２】
前記ＳＩＭＤ命令生成部が、前記ＳＩＭＤ化解析部でＳＩＭＤ化可能と判定されたループ内に存在する判定文が条件の真偽に応じて異なる処理を実行させるＩＦ文であって、該ＩＦ文の下で実行される処理のうち、処理対象の複数のデータについて該ＩＦ文によりいずれも真と判定される場合、およびいずれも偽と判定される場合に、ＳＩＭＤ命令を生成するものであることを特徴とする請求項１記載のコンパイル装置。
【請求項３】
プログラムを実行する演算処理装置内で実行され、該演算処理装置内に、プログラム言語を用いて作成されたソースプログラムをコンピュータ上で実行可能なオブジェクトプログラムに変換するコンパイルを行なうコンパイル装置を構築するコンパイルプログラムであって、
前記演算処理装置内に、
コンパイル対象の、ループを含むプログラム内の各ループについて、複数のデータに対して一度の命令で同じ処理を実行させるＳＩＭＤ化が可能なループであるか否かを判定するＳＩＭＤ化解析部と、
前記ＳＩＭＤ化解析部でＳＩＭＤ化可能と判定されたループについて、該ループ内に、条件に応じて異なる処理を実行させる判定文が存在する場合の該判定文の下で実行される処理のうち、処理対象の複数データについて該判定文により同一条件と判定される場合にＳＩＭＤ処理を実行させるＳＩＭＤ命令を生成し、該判定文により異なる条件と判定される場合に逐次的な処理を実行させる命令を生成するＳＩＭＤ命令生成部とを有するコンパイル装置を構築することを特徴とするコンパイルプログラム。
【請求項４】
前記ＳＩＭＤ命令生成部が、前記ＳＩＭＤ化解析部でＳＩＭＤ化可能と判定されたループ内に存在する判定文が条件の真偽に応じて異なる処理を実行させるＩＦ文であって、該ＩＦ文の下で実行される処理のうち、処理対象の複数のデータについて該ＩＦ文によりいずれも真と判定される場合、およびいずれも偽と判定される場合に、ＳＩＭＤ命令を生成するものであることを特徴とする請求項３記載のコンパイルプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本願は、コンパイルを実行するコンパイル装置およびコンパイルプログラムに関する。
【背景技術】
【０００２】
近年、処理能力向上のために、複数のデータに対して一度の命令で同じ処理を実行させる、いわゆるＳＩＭＤ（Ｓｉｎｇｌｅ Ｉｎｓｔｒｕｃｔｉｏｎ Ｍｕｌｔｉｐｌｅ Ｄａｔａ）命令を有するプロセッサ（例えば、Ｉｎｔｅｌ ＸｅｏｎのＳＳＥ２など）が登場してきている。

【０００３】
ＳＩＭＤ命令を使うと、複数のデータに対する演算を、１つの命令で、命令の実行時間を変えることなく実行することができる。このため、ＳＩＭＤ命令を使うことにより演算処理速度を高めることができる。
【０００４】
ここでループ内の処理がＳＩＭＤ化可能か否かを判定する方法が提案されている。
【０００５】
しかしながら、プログラム中のループ内の判定文の下で実行される処理については、単純にＳＩＭＤ化しても演算処理速度が向上しないばかりかかえって低下する場合がある。
【０００６】
また、ベクトル演算子へのデータ依存関係を解析しアンローリングの可否を判定してアンローリングを行なう手法が提案されている。しかしながらこの提案はアンローリングに関する提案であり、ＳＩＭＤ化とは無関係である。
【先行技術文献】
【特許文献】
【０００７】
【特許文献１】特開平６－３２４８８１号公報
【特許文献２】特開昭６２－１６９２７２号公報
【発明の概要】
【発明が解決しようとする課題】
【０００８】
１つの側面では、本発明は、ループ内の判定文の下で実行される処理についてＳＩＭＤ化による演算処理の高速化を図ることを目的とする。
【課題を解決するための手段】
【０００９】
１つの態様では、コンパイル装置は、プログラム言語を用いて作成されたソースプログラムをコンピュータ上で実行可能なオブジェクトプログラムに変換するコンパイルを実行するコンパイル装置である。
【００１０】
ここで、このコンパイル装置は、ＳＩＭＤ化解析部とＳＩＭＤ命令生成部とを有する。
【００１１】
ＳＩＭＤ化解析部は、コンパイル対象の、ループを含むプログラム内の各ループについて、複数のデータに対して一度の命令で同じ処理を実行させるＳＩＭＤ化が可能なループであるか否かを判定する。
【００１２】
またＳＩＭＤ命令生成部は、ＳＩＭＤ化解析部でＳＩＭＤ化可能と判定されたループについて以下の処理を実行する。すなわち、このＳＩＭＤ命令生成部は、そのループ内に、条件に応じて異なる処理を実行させる判定文が存在する場合のその判定文の下で実行される処理に着目する。そしてこのＳＩＭＤ命令生成部は、その処理のうち、処理対象の複数データについて判定文により同一条件と判定される場合にＳＩＭＤ処理を実行させるＳＩＭＤ命令を生成する。一方、このＳＩＭＤ命令生成部は、その判定文により異なる条件と判定される場合には逐次的な処理を実行させる命令を生成する。
【００１３】
また１つの態様では、コンパイルプログラムは、プログラムを実行する演算処理装置内で実行され、その演算処理装置内に、ソースプログラムをオブジェクトプログラムに変換するコンパイルを実行するコンパイル装置を構築するプログラムである。
【発明の効果】
【００１４】
１つの態様によれば、ＳＩＭＤ化のメリットが生かされ、高速演算処理可能なオブジェクトプログラムが生成される。

【図面の簡単な説明】
【００１５】
【図１】ソースプログラムのループ部分の一例を示す図である。
【図２】マスク方式を採用してＳＩＭＤ化したプログラムの例を示す図である。
【図３】後述する実施形態により作成されるプログラム例を示した図である。
【図４】コンパイル装置の一実施形態として動作するコンピュータの外観図である。
【図５】図４に外観を示すコンピュータのハードウエア構成図である。
【図６】実施形態としてのコンパイル装置の機能説明図である。
【図７】ＳＩＭＤ条件認識部の処理を表わしたフローチャートである。
【図８】図７に示す処理により作成される条件管理テーブルの一例を示す図である。
【図９】図６のＳＩＭＤ命令生成部の処理管理テーブル作成部、および処理埋込部における処理を表わしたフローチャートである。
【図１０】処理管理テーブル作成部１３２（図９のステップＳ２４）の詳細フローを示した図である。
【図１１】処理管理テーブルの一例を示す図である。
【発明を実施するための形態】
【００１６】
以下、実施形態を説明する。
【００１７】
図１はソースプログラムのループ部分の一例を示す図である。
【００１８】
ここには、「ｄｏ　ｉ＝１，ｎ」から「ｅｎｄ　ｄｏ」までのループ内に、「処理A_i」と、「処理B_i」と、「処理C_i」と、「処理D_i」との４つの処理が示されている。「処理A_i」は条件に因らず実行される処理である。「処理B_i」は、条件判定文「ｉｆ（条件X_i）」中の「条件X_i」が真のときに実行される処理である。また「処理C_i」は、その「条件X_i」が偽のときに実行される処理である。さらに「処理D_i」は条件に因らず実行される処理である。
【００１９】
すなわち、この図１のループは、処理A_iを実行した後、X_iが真であれば処理B_iを実行し処理C_iはスキップして処理D_iを実行し、X_iが偽であれば処理B_iはスキップし処理C_iを実行して処理D_iを実行する。図１のループは、これを、先ずはｉ＝１と置いて実行し、ｉを１ずつインクリメントしながらｉ＝ｎまで順次に繰り返すことを意味している。
【００２０】
図２は、マスク方式を採用してＳＩＭＤ化したプログラムの例を示す図である。この図２のプログラムは図１のソースプログラムと同一内容の処理を実行するプログラムである。この図２は、本件に対する比較例に相当する。
【００２１】
ここではＳＩＭＤ実行のデータ数を２としている。
【００２２】
「ｄｏ　ｉ＝１，ｎ－１，２」は、以下の（ａ）～（ｃ）の処理を実行することを意味している。すなわち、
（ａ）先ずｉ＝１と置いて「ｅｎｄ　ｄｏ」までの間の処理を実行する。
【００２３】
（ｂ）次にｉを２インクリメントすることによりｉ＝３として「ｅｎｄ　ｄｏ」までの間の処理を実行する。
【００２４】
（ｃ）これをｉを２ずつインクリメントしながらｉ＝ｎ－１まで繰り返す。
【００２５】
「ｄｏ　ｉ＝１，ｎ－１，２」と「ｅｎｄ　ｄｏ」との間の処理において、先ずは、

「処理A_i」と「処理A_{i+1}」をＳＩＭＤで一度に計算する。

【００２６】

次に、条件X_i、X_{i+1}からマスクデータＭ１を生成し、次に「処理B_i」と「処理B_{i+1}」をマスクデータＭ１によるマスク付きＳＩＭＤで計算する。ここで、マスクデータＭ１は、条件X_iが真であれば、「処理B_i」の計算結果を採用し、条件X_{i+1}が真であれば、「処理B_{i+1}」の計算結果を採用することを表わすデータである。これはX_iとX_{i+1}の双方が真であれば「処理B_i」と「処理B_{i+1}」の双方の計算結果を採用し、X_i、X_{i+1}の双方が偽であれば「処理B_i」と「処理B_{i+1}」の双方の計算結果を破棄することを意味している。

【００２７】

次にマスクデータＭ１とは逆条件のマスクデータＭ２を生成し、このマスクデータＭ２によるマスク付きＳＩＭＤで「処理C_i」と「処理C_{i+1}」を計算する。

【００２８】

このマスクデータＭ２は、マスクデータＭ１とは逆に、条件X_iが偽であれば「処理C_i」の計算結果を採用し、条件X_{i+1}が偽であれば「処理C_{i+1}」の計算結果を採用することを意味する。

【００２９】

「処理C_i」、「処理C_{i+1}」の計算の後は、条件X_i、X_{i+1}の真偽とは無関係に、「処理D_i」、「処理D_{i+1}」をＳＩＭＤで計算する。

【００３０】

図２に示すようなマスク方式を使ったＳＩＭＤの場合、条件式が偽のときに実行されるｅｌｓｅ句が無く（図２ではｅｌｓｅ句が存在する）、かつ条件式が真になる割合いが多ければ、ＳＩＭＤ化による処理速度の向上が期待できる。

【００３１】

しかしながら、このマスク方式の場合、冗長な計算、すなわち、判定文の真偽にかかわらず真の場合の計算と偽の場合の計算の双方が行なわれる。このため、このマスク方式の場合、この部分の演算量が多く、性能向上が見込めない場合がある。またマスクデータを生成する処理自体がオーバヘッドとなり、ＳＩＭＤ化の効果が落ちてしまう結果となる。

【００３２】

さらに、判定文が真の場合の計算と偽の場合の計算との双方を常に実行するため、本来計算してはいけない計算も行なってしまう場合がある。例えば分母がゼロか否かを判定してゼロでない場合に除算を計算する場合に、真偽の双方で計算を行なうと分母がゼロの場合も計算を行なう結果となり、エラーとなってしまいプログラムが正常に実行できない。このような場合はＳＩＭＤ化自体が不可能である。

【００３３】

図３は、後述する実施形態により作成されるプログラム例を示した図である。この図３も図１のソースプログラムと同一内容の処理を実行するプログラムである。ここでもＳＩＭＤ実行のデータ数を２としている。

【００３４】

「ｄｏ　ｉ＝１，ｎ－１，２」は、図２の場合と同じく、以下の（ａ）～（ｃ）の処理を実行することを意味している。

【００３５】

（ａ）先ずｉ＝１と置いて「ｅｎｄ　ｄｏ」までの間の処理を実行する。

【００３６】

（ｂ）次にｉを２インクリメントしｉ＝３として「ｅｎｄ　ｄｏ」までの間の処理を実行する。

【００３７】

（ｃ）さらにｉを２ずつインクリメントしながらｉ＝ｎ－１まで繰り返す。

【００３８】

（２）行目の「処理A_i」、「処理A_{i+1}」は条件の真偽とは無関係であり、ＳＩＭ

Ｄにより計算される。

【００３９】

（３）行目から（１７）行目までは、２段階のｉｆ文により処理内容が分かれている。このようなｉｆ文等の判定文が複数段に形成されているプログラム構造を、ここでは「入れ子構造」と称する。以下では、この図３に基づいてこの入れ子構造を説明する。

【００４０】

（３）行目の「ｉｆ（条件X_i）」では条件X_iの真偽が判定され、（４）行目の「ｉｆ（条件X_{i+1}）」では条件X_{i+1}の真偽が判定される。したがって（５）行目の「処理B_i」、「処理B_{i+1}」は、条件X_i、X_{i+1}の双方が真の場合に実行される。ここでは、これら「処理B_i」、「処理B_{i+1}」がＳＩＭＤで実行される。

【００４１】

（４）行目の「ｉｆ（条件X_{i+1}）」が偽のときは、（６）行目にスキップされ、「条件X_iが真、かつ条件X_{i+1}が偽」に対応する２つの処理、すなわち（７）行目の「処理B_i」と（８）行目の「処理C_{i+1}」が逐次的に実行される。

【００４２】

また、（３）行目のｉｆ文で条件X_iが偽であると判定されると（１０）行目にスキップし、（１１）行目において条件X_{i+1}の判定が行なわれる。条件X_{i+1}が真であると判定されると、「条件X_iが偽かつ条件X_{i+1}が真」に対応する２つの処理、すなわち（１２）行目の「処理C_i」と（１３）行目の「処理B_{i+1}」が逐次的に実行される。

【００４３】

（１１）行目の「ｉｆ（条件X_{i+1}）」で条件X_iが偽であると判定されると（１４）行目にスキップし、「条件X_i、X_{i+1}の双方が偽」に対応する２つの処理、すなわち（１５）行目の２つの「処理C_i」、「処理C_{i+1}」が、ＳＩＭＤで実行される。

【００４４】

その後、（１８）行目の、条件には無関係の「処理D_i」、「処理D_{i+1}」が、ＳＩＭＤで実行される。

【００４５】

この図３のプログラムによれば、ｉｆ文の下で行なわれる処理について、条件X_i、X_{i+1}の双方が真、又は双方が偽のときにＳＩＭＤ命令を使った処理が行なわれ、真と偽、又は偽と真のときにはＳＩＭＤ命令は使わずに複数の処理が逐次的に行なわれる。この場合、図２に示すマスク方式のような冗長な計算、すなわち真と偽の双方での計算を実行する必要がなく、処理時間を短縮できる。図３のプログラムの場合、条件X_i、X_{i+1}が真、偽又は偽、真の場合は、ＳＩＭＤ命令を使わず逐次処理を行なうことになるが、この実行時間はＳＩＭＤ処理を行なわない場合と同等程度の処理時間となる。

【００４６】

さらに、マスクデータを作る必要はないので、オーバーヘッド無しにＳＩＭＤ化の効果をダイレクトに得ることができる。

【００４７】

また、判定結果に応じて、判定結果が真の場合の処理、又は偽の場合の処理の一方のみ、すなわち実行すべき処理のみ実行されるため、真偽双方の処理を実行することによるエラーなどが起こることもない。

【００４８】

次に、本件の実施形態を説明する。以下に説明する実施形態は、一例として図１に示すソースプログラムを元にして図３に示すプログラムに相当するオブジェクトプログラムを作成するコンパイル装置である。

【００４９】

図４は、コンパイル装置の一実施形態として動作するコンピュータの外観図、図５は図４に外観を示すコンピュータのハードウエア構成図である。

【００５０】

図４に示すコンピュータ１００は、後述するＣＰＵ（Ｃｏｎｔｒｏｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）、メモリ、ハードディスク装置等を内蔵した本体部１１０を有する。またこのコンピュータ１００は、その本体部１１０からの指示により表示画面１２１上に画像を表示する画像表示部１２０を有する。さらに、このコンピュータ１００は、ユーザによるキー操作に応じてこのコンピュータ１００内に指示や文字情報等を入力するキーボード１３０を有する。さらに、このコンピュータ１００は、ユーザ操作により、表示画面１２１上に表示されたカーソルの移動や、表示画面１２１上に表示されたアイコン等の指定による指示入力等を行なうマウス１４０を有する。

【００５１】

本体部１１０は、ＣＤ（Ｃｏｍｐａｃｔ　Ｄｉｓｋ）やＤＶＤ（Ｄｉｇｉｔａｌ　Ｖｅｒｓａｔｉｌｅ　Ｄｉｓｋ）（以下では、ＣＤとＤＶＤを区別せずにＣＤ／ＤＶＤと称する）が装填されるＣＤ／ＤＶＤ装填口１１１を有する。その本体部１１０の内部には、そのＣＤ／ＤＶＤ装填口１１１から装填されたＣＤ／ＤＶＤをドライブする、ＣＤ／ＤＶＤドライブが内蔵されている。

【００５２】

本体部１１０には、さらに、図５に示すように、ＣＰＵ１１２、メモリ１１３、ハードディスク装置１１４およびネットワークインタフェイス１１５を有する。また、この図５には、ＣＤ／ＤＶＤ１１７をドライブする上述のＣＤ／ＤＶＤドライブ１１６も示されている。

【００５３】

これらの各要素の相互間、さらに、これらの各要素と、図４に外観を示した画像表示部１２０、キーボード１３０およびマウス１４０との相互間は、バス１５０で接続されている。

【００５４】

ここで、ハードディスク装置１１４には、各種プログラムやデータ等が保存されている。このハードディスク装置１１４に保存されているプログラムが読み出されてメモリ１１３に展開され、そのメモリ１１３上に展開されたプログラムがＣＰＵ１１２で実行される。また、ネットワークインタフェイス１１５は、ＬＡＮ（Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）やインタネット等を介して外部と通信する要素である。

【００５５】

ここでハードディスク装置１１４には、ソースプログラム（例えば図１参照）を元にオブジェクトプログラム（例えば図３参照）を作成するコンパイルプログラムが保存されている。

【００５６】

ソースプログラムは、ＣＤ／ＤＶＤ１１７に格納されてＣＤ／ＤＶＤドライブ１１６で読み込まれる。あるいは、ネットワークインタフェース１１５を介して受信してもよい。ハードディスク装置１１４に保存されているコンパイルプログラムがメモリ１１３に展開されＣＰＵ１１２で実行されると、このコンピュータ１００はコンパイル装置の一例として動作する。このコンピュータ１００がコンパイル装置として動作すると、上記のようにして取得したソースプログラムを元にオブジェクトプログラムが作成される。作成されたオブジェクトプログラムは一旦はハードディスク装置１１４内に保存された後、ＣＤ／ＤＶＤドライブ１１６によりＣＤ／ＤＶＤに書き込まれたり、あるいは、ネットワークインタフェース１１５を介して送信される。このようにして、そのオブジェクトプログラムが実行先のコンピュータ（図示せず）に渡され、そのコンピュータで実行される。

【００５７】

図６は、実施形態としてのコンパイル装置の機能説明図である。

【００５８】

この図６のコンパイル装置１０は、図４，図５に示すコンピュータ内でのコンパイルプログラムの実行によりそのコンピュータ内に実現する機能を表わしたものである。

【００５９】

この図６のコンパイル装置１０は、プログラム言語で作成されたソースプログラム１を取り込み、そのソースプログラム１を、実行予定のコンピュータ上で実行可能なオブジェクトプログラム２に変換するコンパイルを実行する装置である。ここで、このオブジェクトプログラム２が実行されるコンピュータは、このコンパイル装置１０の機能実現に用いるコンピュータと同一のコンピュータであることを妨げるものではないが、別のコンピュータであってもよい。

【００６０】

このコンパイル装置１０は、ソースプログラム解析部１１と、ＳＩＭＤ化解析部１２と、ＳＩＭＤ命令生成部１３と、オブジェクトプログラム生成部１４とを有する。このうちＳＩＭＤ命令生成部１３は、ＳＩＭＤ条件認識部１３１と、処理管理テーブル作成部１３２と、処理埋込部１３３とを有する。

【００６１】

ソースプログラム解析部１１は、フォートラン言語やＣ言語、Ｃ＋＋言語等のプログラム言語を用いて作成されたソースプログラムを、それらのプログラム言語から離れた共通の中間言語のプログラムに変換する。以降の処理は、この中間言語に対して行なわれる。ただし、前述の図３は、分かり易さのため、プログラム言語に近い様式で示されている。

【００６２】

このソースプログラム解析部１１の処理は従来のコンパイル装置における処理と変わるところはなく、ここでの詳細説明は省略する。

【００６３】

ＳＩＭＤ化解析部１２は、ループがＳＩＭＤ化可能か否かをデータ依存解析を行なって判定する。

【００６４】

この解析処理はベクトル化のためのデータ依存解析と同様であり、例えば以下の文献に示されている。

【００６５】

“The parallel execution of DO loops, Leslie Lamport”Communications of the ACM, Volume 17, Issue 2（February 1974）, Pages：83－93

あるいは、前掲の特許文献１に記載された手法を採用してもよい。

【００６６】

このデータ依存解析も従来から知られている手法であり、ここでのこれ以上の説明は省略する。

【００６７】

ＳＩＭＤ命令生成部１３は、ＳＩＭＤ化解析部１２でＳＩＭＤ化可能と判定されたループについて以下の処理を実行する。すなわち、このＳＩＭＤ命令生成部１３は、そのループ内に、条件に応じて異なる処理を実行させる判定文が存在する場合のその判定文の下で実行される処理に着目する。そしてこのＳＩＭＤ命令生成部１３は、その処理のうち、処理対象の複数データについて判定文により同一条件と判定される場合にＳＩＭＤ処理を実行させるＳＩＭＤ命令を生成する。一方、このＳＩＭＤ命令生成部は１３、その判定文により処理対象の複数のデータについて異なる条件と判定される場合には逐次的な処理を実行させる命令を生成する。

【００６８】

ここで、ＳＩＭＤ命令生成部１３を構成するＳＩＭＤ条件認識部１３１、処理管理テーブル作成部１３２および処理埋込部１３３は、上記の命令生成の処理を実行するために、それぞれ以下の処理を分担している。すなわち、ＳＩＭＤ条件認識部１３１は、ＳＩＭＤ化解析部１２によりＳＩＭＤ化可能と判定されたループの内部を調べて、後述する図８に示すような条件管理テーブルを作成する。また、処理管理テーブル作成部１３２は、図８の条件管理テーブルを参照して図１１の処理管理テーブルを作成する。この処理管理テー

ブルについても後述する。処理埋込部１３３では、図１１の処理管理テーブルを元にＳＩＭＤ命令や逐次的な命令を生成して、例えば図１に示すソースプログラムの場合に図３に示すようなプログラムを作成する。尚、この処理埋込部１３３は、判定文の影響を受けない処理については、従来のＳＩＭＤ化処理と同様、通常のＳＩＭＤ命令を生成する。

【００６９】

このＳＩＭＤ命令生成部１３は、本実施形態の特徴的な処理を実行する部分である。

【００７０】

また、図６のコンパイル装置１０を構成するオブジェクトプログラム生成部１４は、ＳＩＭＤ命令生成部１３で生成された命令を含む、中間言語で記述されたプログラムを元に、コンピュータでの実行が可能なオブジェクトプログラム１を生成する。このオブジェクトプログラム生成部１４における、オブジェクトプログラム生成処理自体は従来と変わるところがなく、このオブジェクトプログラム生成部１４についての詳細説明は省略する。

【００７１】

以下では、本実施形態の特徴的な部分であるＳＩＭＤ命令生成部１３について詳述する。

【００７２】

以下の説明にあたっては、図１のソースプログラムから図３のプログラムに変換する場合を例に挙げながら説明する。

【００７３】

図７は、ＳＩＭＤ条件認識部１３１の処理を表わしたフローチャートである。

【００７４】

また、図８は、図７に示す処理により作成される条件管理テーブルの一例を示す図である。この図８は、図１に示す、ソースプログラムの一部分としてのループに、図７に示す処理を適用した場合に作成される条件管理テーブルを示している。尚、図１に示すループは、ＳＩＭＤ化解析部１２によりＳＩＭＤ化可能と判定されたループであるとする。

【００７５】

図７に示すフローチャートのステップＳ１１では、ＳＩＭＤ化可能と判定されたループ（ここでは、図１に示すループ）から、ｉｆ,ｅｌｓｅ，ｅｎｄｉｆ等の制御文や「処理$Ａ_i$」，「処理$Ｂ_i$」等の処理のうちの１つが取り出される。ここでは先ず、図１に示す「処理$Ａ_i$」が取り出される。

【００７６】

次いで、今回取り出した「処理$Ａ_i$」が制御文であるか否かが判定される（ステップＳ１２）。「処理$Ａ_i$」は制御文でなく計算処理なのでステップＳ１３に進む。このステップＳ１３では、図８の条件管理テーブルの「種類」および「処理」の欄に、それぞれ「計算処理」および「処理$Ａ_i$」が登録される。

【００７７】

図１の例では、条件管理テーブルへの登録処理は未だ終了していないので（ステップＳ１５）、ステップＳ１１に戻り、今度はｉｆ文が取り出される。このｉｆ文は制御文の１種であり（ステップＳ１２）、今度はステップＳ１４に進んで、図８の条件管理テーブルの「種類」および「条件」の各欄にそれぞれ「ｉｆ」および「$Ｘ_i$」が登録される。

【００７８】

以下同様にして、条件管理テーブルに、「処理$Ｂ_i$」，「ｅｌｓｅ」，「処理$Ｃ_i$」，「ｅｎｄｉｆ」および「処理$Ｄ_i$」が登録される。「処理$Ｂ_i$」，「処理$Ｃ_i$」および「処理$Ｄ_i$」については、「処理$Ａ_i$」と同様、ステップＳ１３において条件管理テーブル（図８参照）の「種類」の欄に「計算処理」が入力され「処理」の欄に「処理$Ｂ_i$」，「処理$Ｃ_i$」および「処理$Ｄ_i$」のそれぞれが登録される。一方、「ｅｌｓｅ」と「ｅｎｄｉｆ」については、「ｉｆ」と同様、ステップＳ１４において「種類」の欄にそれぞれ「ｅｌｓｅ」および「ｅｎｄｉｆ」が登録され、さらに「ｅｌｓｅ」の場合は「条件」の欄に「ｎｏｔ$Ｘ_i$」が登録される。

【００７９】

図９は、図６のＳＩＭＤ命令生成部１３の処理管理テーブル作成部１３２および処理埋込部１３３における処理を表わしたフローチャートである。

【００８０】

この図９のフローチャートにおけるステップＳ２４が処理管理テーブル作成部１３２の処理に対応し、ステップＳ２３およびステップＳ２５が処理埋込部１３３の処理に対応する。

【００８１】

また図１０は処理管理テーブル作成部１３２（図９のステップＳ２４）の詳細フローを示した図である。

【００８２】

さらに図１１は、処理管理テーブルの一例を示す図である。

【００８３】

ここでは、図９に示すフローチャートに従い、先ず、図８に示す条件管理テーブルから「処理A_i」が取り出される（ステップＳ２１）。この「処理A_i」は条件判定文ではないため（ステップＳ２２）、ステップＳ２３に進み、通常のＳＩＭＤ処理の命令が生成される（図３（２）参照）。

【００８４】

次に条件管理テーブルから「ｉｆ」文が取り出される（ステップＳ２１）。この「ｉｆ」文は条件判定文であり（ステップＳ２２）、ステップＳ２４に進み、処理管理テーブル（図１１参照）が作成される。

【００８５】

このステップＳ２４では、図１０に示すフローチャートに従う処理が実行される。

【００８６】

ここでは、先ず、条件管理テーブル（図８参照）から条件X_iが取り出され、条件X_{i+1}と合わせて処理管理テーブル（図１１）の１行目に設定される（ステップＳ３１）。

【００８７】

次に、条件管理テーブル（図８）から次の「処理B_i」が取り出され、その「処理B_i」が条件X_iが真の欄に設定される。また条件X_{i+1}が真の欄に「処理B_{i+1}」が設定される（ステップＳ３２）。

【００８８】

次に、条件管理テーブル（図８）中の「ｅｌｓｅ」が読みとばされる（ステップＳ３３）。ただし「ｅｌｓｅ」の存在は認識され、これにより「ｅｌｓｅ」の次の処理が条件X_i、X_{i+1}が偽のときに実行される処理であることが認識される。

【００８９】

次に条件管理テーブル（図８）から次の「処理C_i」が取り出され、条件X_iが偽の欄に設定される。また「処理C_{i+1}」が「条件X_{i+1}」が偽の欄に設定される（ステップＳ３４）。

【００９０】

次の「ｅｎｄｉｆ」により、ｉｆ文の下で実行された処理の終了を認識する。「ｅｎｄｉｆ」自身は読みとばす（ステップＳ３５）。

【００９１】

図９に戻って説明を続ける。

【００９２】

図９のステップＳ２４では、上記の通り、処理管理テーブル（図１１参照）が作成される。ステップＳ２５では、今度はその作成された処理管理テーブルから処理が取り出され、ｉｆ文の入れ子構造に埋め込まれる（ステップＳ２５）。これにより、「処理B_i」，「処理B_{i+1}」，「処理C_i」，「処理C_{i+1}」の各処理が、図３の（５），（７），（８），（１２），（１３），（１５）に埋め込まれ、（１７）までのフローが作成される。

【００９３】

図８の条件管理テーブルには未だ「処理Ｄ$_{i}$」が残っており、したがって図９のステップＳ２６からもう一度ステップＳ２１に戻り、ステップＳ２２を経由して今度はステップＳ２３に進み、「処理Ｄ$_{i}$」，「処理Ｄ$_{i+1}$」について通常のＳＩＭＤ処理の命令が生成される（図３（１８）参照）。

【００９４】

上記の手順を経ることにより、図１のソースプログラムから図３のプログラムが作成される。ただし、ここでの処理は中間言語上で行なわれる。

【００９５】

図６のオブジェクトプログラム作成部１４では、中間言語で作成された図３のプログラムを元に、コンピュータで実行可能なオブジェクトプログラム２が作成される。このオブジェクトプログラム２は、図３のプログラムを元に作成されたプログラムであり、ループ内のｉｆ文等の判定文の下で実行される処理について、ＳＩＭＤ命令を使った高速処理可能なオブジェクトプログラムである。

【００９６】

尚、ここでは条件に応じて真偽の２値に分かれるｉｆ文を取り挙げて説明したが、ｓｗｉｔｃｈ文やｓｅｌｅｃｔ文といった、条件に応じて３値以上に分岐する命令についても論理的にｉｆ文に変換することができ、したがってｓｗｉｔｃｈ文やｓｅｌｅｃｔ文等の判定文についても本件の適用が可能である。

【００９７】

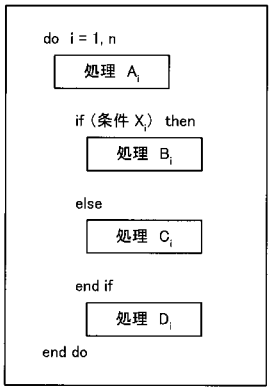
またここではＳＩＭＤ実行のデータ数を２としているが、データ数が３以上であっても組合せの数が増えるだけで上記と同様のＳＩＭＤ化処理が可能である。

【符号の説明】

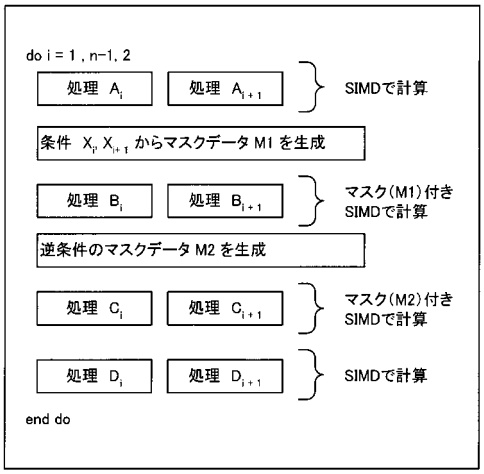
【００９８】

１　ソースプログラム
２　オブジェクトプログラム
１０　コンパイル装置
１１　ソースプログラム解析部
１２　ＳＩＭＤ化解析部
１３　ＳＩＭＤ命令生成部
１４　オブジェクトプログラム生成部
１００　コンピュータ
１１０　本体部
１１１　ＣＤ／ＤＶＤ装填口
１１２　ＣＰＵ
１１３　メモリ
１１４　ハードディスク装置
１１５　ネットワークインタフェイス
１１６　ＣＤ／ＤＶＤドライブ
１１７　ＣＤ／ＤＶＤ
１２０　画像表示部
１２１　表示画面
１３０　キーボード
１３１　ＳＩＭＤ条件認識部
１３２　処理管理テーブル作成部
１３３　処理埋込部
１４０　マウス
１５０　バス
Ｍ１　マスクデータ
Ｍ２　マスクデータ

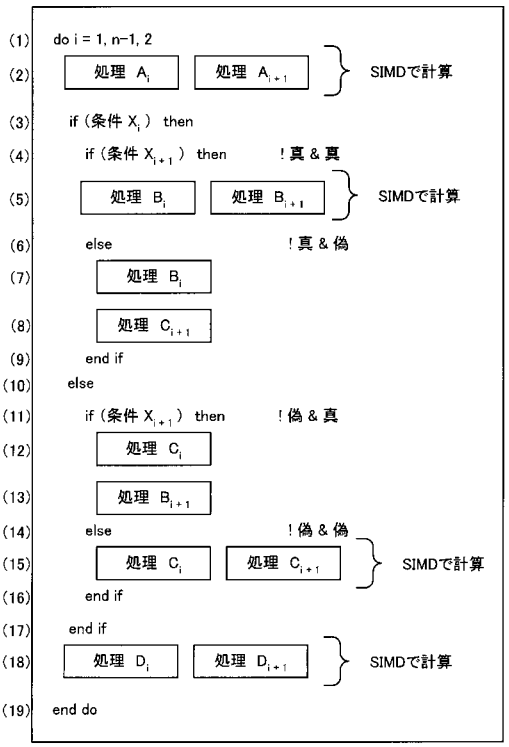
【図１】


do i = 1, n
処理 A_i
if（条件 X_i） then
処理 B_i
else
処理 C_i
end if
処理 D_i
end do

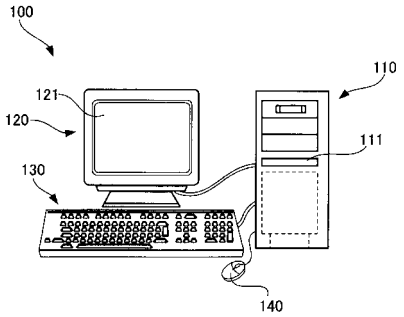
【図２】


do i = 1 , n−1, 2
処理 A_i
処理 A_{i+1}
SIMDで計算
条件 X_i, X_{i+1} からマスクデータ M1 を生成
処理 B_i
処理 B_{i+1}
マスク(M1)付き SIMDで計算
逆条件のマスクデータ M2 を生成
処理 C_i
処理 C_{i+1}
マスク(M2)付き SIMDで計算
処理 D_i
処理 D_{i+1}
SIMDで計算
end do

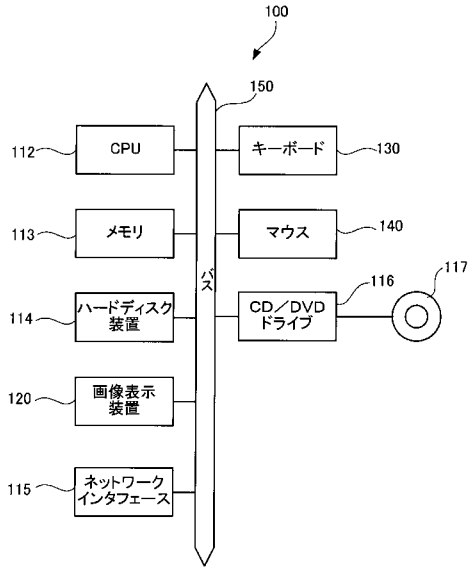
【図３】


(1) do i = 1, n−1, 2
(2) 処理 A_i 処理 A_{i+1} SIMDで計算
(3) if（条件 X_i） then
(4) if（条件 X_{i+1}） then ！真 & 真
(5) 処理 B_i 処理 B_{i+1} SIMDで計算
(6) else ！真 & 偽
(7) 処理 B_i
(8) 処理 C_{i+1}
(9) end if
(10) else
(11) if（条件 X_{i+1}） then ！偽 & 真
(12) 処理 C_i
(13) 処理 B_{i+1}
(14) else ！偽 & 偽
(15) 処理 C_i 処理 C_{i+1} SIMDで計算
(16) end if
(17) end if
(18) 処理 D_i 処理 D_{i+1} SIMDで計算
(19) end do

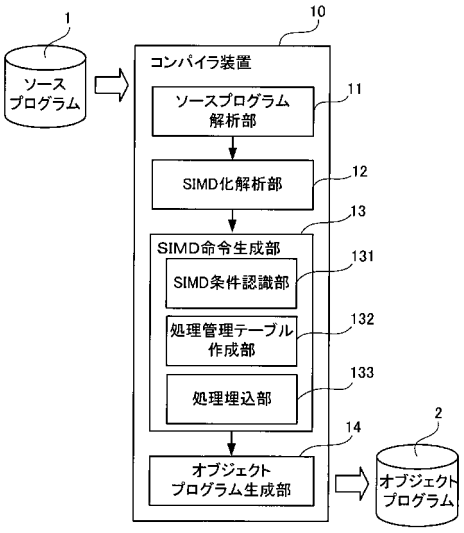
【図４】


100
110
121
120
111
130
140

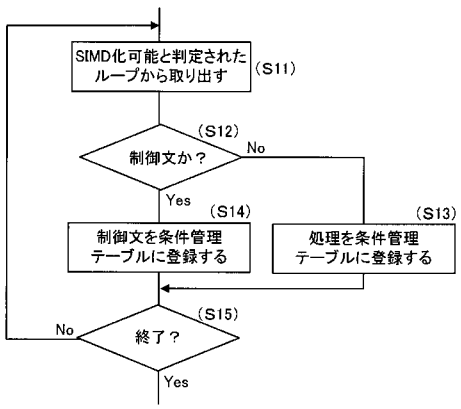
【図５】



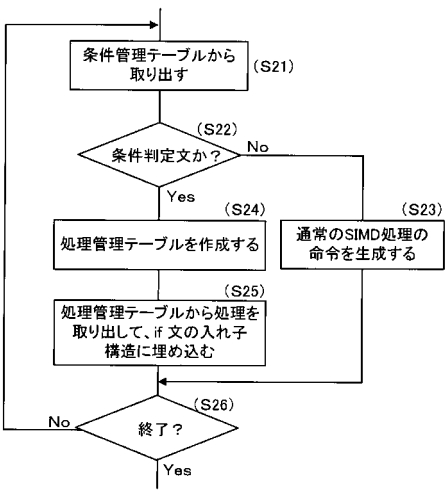
【図６】



【図７】



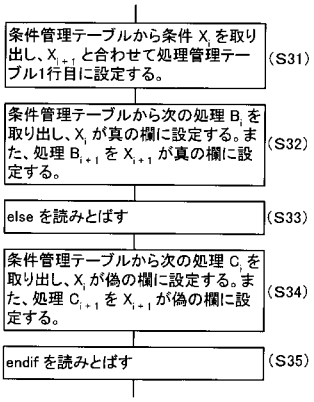
【図９】



【図８】

番号	種類	処理	条件
1	計算処理	処理 A_i	—
2	if	—	X_i
3	計算処理	処理 B_i	
4	else	—	notX_i
5	計算処理	処理 C_i	
6	endif	—	—
7	計算処理	処理 D_i	—

【図１０】



【図１１】

	X_i	X_{i+1}
真真	処理 B_i	処理 B_{i+1}
真偽	処理 B_i	処理 C_{i+1}
偽真	処理 C_i	処理 B_{i+1}
偽偽	処理 C_i	処理 C_{i+1}