

(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2012-168660

(P2012-168660A)

(43) 公開日 平成24年9月6日(2012.9.6)

(51) Int.Cl.		F I			テーマコード (参考)
G06F 1/04 (2006.01)		G06F 1/04	301C		5B042
G06F 11/34 (2006.01)		G06F 11/34	B		5B079
		G06F 11/34	S		

審査請求 未請求 請求項の数 6 O L (全 23 頁)

(21) 出願番号 特願2011-27948 (P2011-27948)
 (22) 出願日 平成23年2月10日 (2011.2.10)

(71) 出願人 00005223
 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 100104190
 弁理士 酒井 昭徳
 (72) 発明者 三輪 真弘
 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 成瀬 彰
 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) 5B042 GA05 GA38 HH20 LA08 MA08
 MA14 MB03 MC28
 5B079 BA01 BC01

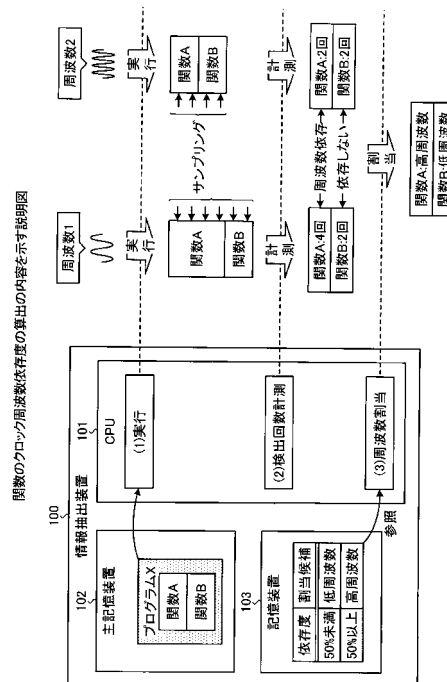
(54) 【発明の名称】 情報抽出プログラム、情報抽出方法、および情報抽出装置

(57) 【要約】

【課題】ソースコードに手を加えることなしに、関数のクロック周波数依存度を算出する。

【解決手段】情報抽出装置100は、CPU101を2種類のクロック周波数で動作させた場合のそれぞれの場合において、従来技術であるタイムベースサンプリングを用いて関数ごとの実行時間を計測する。そして、情報抽出装置100は、2種類のクロック周波数の変化量と実行時間(処理効率)の変化量との割合をクロック周波数依存度として関数ごとに算出する。これにより、情報抽出装置100は、ソースコードに手を加えることなしにクロック周波数依存度を算出することができる。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

第 1 のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとに前記プログラム内の関数群のうちの実行中の関数を検出する第 1 の検出工程と、

前記第 1 のクロック周波数とは異なる第 2 のクロック周波数で動作中の前記プロセッサによる前記プログラムの実行中において、前記所定時間ごとに前記関数群のうちの実行中の関数を検出する第 2 の検出工程と、

前記関数群のうちいずれかの関数について、前記第 1 の検出工程による前記いずれかの関数の第 1 の検出回数を計測する第 1 の計測工程と、

前記いずれかの関数について、前記第 2 の検出工程による前記いずれかの関数の第 2 の検出回数を計測する第 2 の計測工程と、

前記第 1 の検出回数から前記第 2 の検出回数への変化量に基づいて、クロック周波数の変化による前記いずれかの関数の実行時間の変化の度合いを示す依存度を算出する算出工程と、

前記算出工程によって算出された依存度を出力する出力工程と、

をコンピュータに実行させることを特徴とする情報抽出プログラム。

【請求項 2】

前記算出工程によって算出された依存度が閾値以下であるか否かを判定する判定工程と、

前記判定工程によって依存度が閾値以下であると判定された場合に、所定の低周波数の割当クロック周波数を前記いずれかの関数と関連付けて、割当テーブルに格納し、前記判定工程によって依存度が閾値より大きいと判定された場合に、所定の高周波数の割当クロック周波数を前記いずれかの関数と関連付けて、前記割当テーブルに格納する格納工程と、

を前記コンピュータに実行させることを特徴とする請求項 1 に記載の情報抽出プログラム。

【請求項 3】

前記依存度ごとに対応する割当クロック周波数を保持するテーブルに基づいて、前記算出工程によって算出された依存度に対応する割当クロック周波数を前記いずれかの関数と関連付けて割当テーブルに格納する格納工程を前記コンピュータに実行させることを特徴とする請求項 1 に記載の情報抽出プログラム。

【請求項 4】

前記プログラムをあらたに実行した場合において、前記関数群のうちの実行中の関数を検出する第 3 の検出工程と、

前記割当テーブルに基づいて、前記第 3 の検出工程によって検出された関数に対応する前記割当クロック周波数を前記プロセッサのクロック周波数として設定する設定工程と、

を前記コンピュータに実行させることを特徴とする請求項 2 または 3 に記載の情報抽出プログラム。

【請求項 5】

第 1 の検出手段と、第 2 の検出手段と、第 1 の計測手段と、第 2 の計測手段と、算出手段と、出力手段と、を備えるコンピュータが、

前記第 1 の検出手段により、第 1 のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとに前記プログラム内の関数群のうちの実行中の関数を検出する第 1 の検出工程と、

前記第 2 の検出手段により、前記第 1 のクロック周波数とは異なる第 2 のクロック周波数で動作中の前記プロセッサによる前記プログラムの実行中において、前記所定時間ごとに前記関数群のうちの実行中の関数を検出する第 2 の検出工程と、

前記第 1 の計測手段により、前記関数群のうちいずれかの関数について、前記第 1 の検出工程による前記いずれかの関数の第 1 の検出回数を計測する第 1 の計測工程と、

10

20

30

40

50

前記第2の計測手段により、前記いずれかの関数について、前記第2の検出工程による前記いずれかの関数の第2の検出回数を計測する第2の計測工程と、

前記算出手段により、前記第1の検出回数から前記第2の検出回数への変化量に基づいて、クロック周波数の変化による前記いずれかの関数の実行時間の変化の度合いを示す依存度を算出する算出工程と、

前記出力手段により、前記算出工程によって算出された依存度を出力する出力工程と、
を実行することを特徴とする情報抽出方法。

【請求項6】

第1のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとに前記プログラム内の関数群のうちの実行中の関数を検出する第1の検出手段と

10

、
前記第1のクロック周波数とは異なる第2のクロック周波数で動作中の前記プロセッサによる前記プログラムの実行中において、前記所定時間ごとに前記関数群のうちの実行中の関数を検出する第2の検出手段と、

前記関数群のうちいずれかの関数について、前記第1の検出手段による前記いずれかの関数の第1の検出回数を計測する第1の計測手段と、

前記いずれかの関数について、前記第2の検出手段による前記いずれかの関数の第2の検出回数を計測する第2の計測手段と、

前記第1の検出回数から前記第2の検出回数への変化量に基づいて、クロック周波数の変化による前記いずれかの関数の実行時間の変化の度合いを示す依存度を算出する算出手段と、

20

前記算出手段によって算出された依存度を出力する出力手段と、
を備えることを特徴とする情報抽出装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、アプリケーションプログラムから情報を抽出する情報抽出プログラム、情報抽出方法、および情報抽出装置に関する。

【背景技術】

【0002】

近年、情報機器の分野において、バッテリー持続時間を延長する観点、環境保全の観点、情報機器の発熱を抑えて信頼性を向上させる観点から、情報機器の省電力化が図られている。

30

【0003】

情報機器のCPU(Central Processing Unit)の消費電力はクロック周波数を高くするにつれ増大するため、情報機器のCPUのクロック周波数を低くすれば省電力化が可能である。しかしながら、クロック周波数を低くするとクロック周波数に律速される情報機器のパフォーマンスが低下してしまう。

【0004】

ここで、CPUで実行されるアプリケーションプログラム(以下、単に「プログラム」という)は関数群から構成され、関数ごとにクロック周波数に律速される度合い(CPUクロック周波数依存度、以下、単に「依存度」という)が異なる。従って、クロック周波数に律速される関数(依存度の高い関数)は高いクロック周波数で実行し、クロック周波数に律速されない関数(依存度の低い関数)は低いクロック周波数で実行すれば、情報機器のパフォーマンスを低下させずに省電力化できる。

40

【0005】

従来、依存度算出の技術として、プログラムのソースコード内の各関数の前後に関数の実行時間を算出する命令を書き込んでおき、CPUのクロック周波数を変更したときの実行時間の変化の割合を調べて、依存度として算出する技術がある(例えば、下記非特許文献1参照)。

50

【先行技術文献】

【非特許文献】

【0006】

【非特許文献1】堀田 義彦, 佐藤 三久, 木村 英明, 松岡 聡, 朴 泰祐, 高橋 大介, “PC クラスタにおける電力実行プロファイル情報を用いたDV S制御による電力性能の最適化”, 情報処理学会論文誌 コンピューティングシステム, Vol. 47, No. SIG12 (ACS 15), pp. 272 - 284, (2006).

【発明の概要】

【発明が解決しようとする課題】

10

【0007】

しかしながら、上述した従来技術では、ソースコードに命令を書き込む必要があるため、ソースコードに手を加えられないプログラムには適用できず、依存度を算出できないという問題があった。ソースコードに手を加えられないプログラムとは、例えば、他者による改変を防ぐためにバイナリ形式で提供されているプログラムである。

【0008】

従って、従来では、プログラム内の関数の依存度が分からず、関数ごとにクロック周波数を変更できないため、一様なクロック周波数でプログラム全体を実行していた。

【0009】

そのため、例えば、低いクロック周波数で実行してもパフォーマンスが下がらない関数がプログラム内に存在するにもかかわらず、一様に高いクロック周波数でプログラム全体を実行してしまい消費電力が多くなるという問題があった。また、例えば、高いクロック周波数で実行しなければパフォーマンスが下がる関数がプログラム内に存在するにもかかわらず、一様に低いクロック周波数でプログラム全体を実行してしまいパフォーマンスが悪くなるという問題があった。

20

【0010】

1つの側面では、本発明は、アプリケーションプログラム内の関数ごとにクロック周波数の変化による実行時間の変化の度合い(依存度)を特定することができる情報抽出プログラム、情報抽出方法、および情報抽出装置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】

30

【0011】

1つの案では、第1のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとにプログラム内の関数群のうちの実行中の関数を検出し、第1のクロック周波数とは異なる第2のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとに関数群のうちの実行中の関数を検出し、関数群のうちのいずれかの関数について、該関数の第1の検出回数を計測し、該関数の第2の検出回数を計測し、第1の検出回数から第2の検出回数への変化量に基づいて、クロック周波数の変化による該関数の実行時間の変化の度合いを示す依存度を算出し、算出された依存度を出力する情報抽出プログラム、情報抽出方法、および情報抽出装置を用いる。

【発明の効果】

40

【0012】

本発明にかかる情報抽出プログラム、情報抽出方法、および情報抽出装置によれば、アプリケーションプログラム内の関数ごとにクロック周波数の変化による実行時間の変化の度合い(依存度)を特定することができる。

【図面の簡単な説明】

【0013】

【図1】図1は、関数のクロック周波数依存度の算出の内容を示す説明図である。

【図2】図2は、情報抽出装置のハードウェア構成を示すブロック図である。

【図3】図3は、依存度テーブルの記憶内容を示す説明図である。

【図4】図4は、割当テーブルの記憶内容を示す説明図である。

50

【図 5】図 5 は、アドレステーブルの記憶内容を示す説明図である。
【図 6】図 6 は、特定情報テーブルの記憶内容を示す説明図である。
【図 7】図 7 は、プロファイルテーブルの記憶内容を示す説明図である。
【図 8】図 8 は、周波数候補テーブルの記憶内容を示す説明図である。
【図 9】図 9 は、情報抽出装置の機能的構成を示す機能ブロック図である。
【図 10】図 10 は、情報抽出装置による依存度の算出の内容を示す説明図である。
【図 11】図 11 は、特定情報の内容を示す説明図である。
【図 12】図 12 は、特定情報を用いてサンプリング回数を計測する具体例を示す説明図である。

【図 13】図 13 は、プログラムの測定の内容を示す説明図である。 10
【図 14】図 14 は、関数実行時のクロック周波数の制御の内容を示す説明図である。
【図 15】図 15 は、依存度算出処理の詳細を示すフローチャートである。
【図 16】図 16 は、プログラム実行処理の詳細を示すフローチャートである。
【図 17】図 17 は、割当テーブル作成処理の詳細を示すフローチャートである。
【図 18】図 18 は、クロック周波数制御処理の詳細を示すフローチャートである。
【発明を実施するための形態】

【0014】

以下に添付図面を参照して、本発明の実施の形態にかかる情報抽出プログラム、情報抽出方法、および情報抽出装置を詳細に説明する。

【0015】

(関数のクロック周波数依存度の算出の内容)

まず、図 1 を用いて関数のクロック周波数依存度の算出の内容を示す説明図について説明する。図 1 は、関数のクロック周波数依存度の算出の内容を示す説明図である。ここで、クロック周波数依存度とは、関数が CPU 101 のクロック周波数によってどの程度律速されるかを示す度合いであり、クロック周波数の変化による関数の実行時間(処理効率)の変化の度合いとして算出される。

【0016】

図 1 において、情報抽出装置 100 は、自装置の CPU 101 のクロック周波数を変更可能な装置である。例えば、CPU 101 のクロック周波数を変更可能な装置としては、サーバ、PC(Personal Computer)、携帯端末が挙げられる。 30

【0017】

情報抽出装置 100 の主記憶装置 102 には、プログラム X が保持されている。プログラム X とは、依存度の算出対象となる関数を含むプログラムであって、プログラム実行時には、情報抽出装置 100 が CPU 101 のクロック周波数を制御することにより省電力化される対象となるプログラムである。ここでは、プログラム X は、例として関数 A および関数 B から構成されるプログラムであるとする。

【0018】

さらに、情報抽出装置 100 の記憶装置 103 には、閾値が記憶されている。閾値とは、関数の実行時に、低周波数のクロック周波数(以下、「低クロック周波数」という)で CPU 101 を動作させるか、高周波数のクロック周波数(以下、「高クロック周波数」という)で CPU 101 を動作させるかを判定する境界となる値である。この閾値は予め情報抽出装置 100 のユーザによって設定される。 40

【0019】

ただし、情報抽出装置 100 の記憶装置 103 には、閾値を記憶せずに、依存度に対応する CPU 101 のクロック周波数と依存度とを対応付けて保持するテーブル(以下、「周波数候補テーブル」という)を記憶してもよい。周波数候補テーブルでは、高クロック周波数と低クロック周波数だけではなく、依存度に合わせて複数のクロック周波数の候補を保持してもよい。

【0020】

ここで、情報抽出装置 100 は、プログラム X のソースコードに変更を加えることなし 50

に、プログラム X を構成する関数 A および関数 B のクロック周波数依存度を算出する。まず、情報抽出装置 100 は、CPU 101 を 2 種類のクロック周波数で動作させた場合のそれぞれの場合において、タイムベースサンプリングを用いて関数ごとの実行時間を計測する。

【0021】

具体的には、タイムベースサンプリングでは、CPU 101 が実行している関数を特定する特定情報を所定時間間隔（以下、「サンプリング間隔」という）で取得することにより、実行中の関数を検出する（以下、「サンプリング」という）。そして、関数ごとに検出回数（以下、「サンプリング回数」という）を計測する。サンプリングは所定時間ごとに行われているため、サンプリング回数は、各関数の実行時間（または処理効率）を示す値として採用することができる。従って、情報抽出装置 100 は、関数ごとの実行時間を間接的に計測することができる。

10

【0022】

プログラム X を構成する関数 A および関数 B のクロック周波数依存度の算出では、情報抽出装置 100 は、まず任意の周波数 1（以下、「サンプリング条件 1」という）が CPU 101 のクロック周波数として設定されている状態において（1）プログラム X を実行する。プログラム X の実行後、情報抽出装置 100 は、（2）実行中の関数がプログラム X 内のどの関数であるかを、所定時間間隔でサンプリングし、関数ごとにサンプリング回数を計測する。ここでは、関数 A は 4 回サンプリングされ、関数 B は 2 回サンプリングされたとする。

20

【0023】

次に、情報抽出装置 100 は、周波数 1 とは異なる値である周波数 2（以下、「サンプリング条件 2」という）へと CPU 101 のクロック周波数を変更した状態において（1）プログラム X を実行する。プログラム X の実行後、情報抽出装置 100 は、（2）実行中の関数がプログラム X 内のどの関数であるかを所定時間間隔でサンプリングし、関数ごとにサンプリング回数を計測する。ここでは、関数 A は 2 回サンプリングされ、関数 B は 2 回サンプリングされたとする。

【0024】

クロック周波数を変化させる前後でのサンプリング回数の変化量は、関数の実行時間（処理効率）の変化量を示している。従って、情報抽出装置 100 は、サンプリング回数の変化量に基づき、クロック周波数の変化による実行時間（処理効率）の変化の度合い（すなわち、上述したクロック周波数依存度、以下では単に「依存度」という）を関数ごとに算出することができる。これにより、情報抽出装置 100 は、ソースコードに手を加えることなしに依存度を算出することができる。

30

【0025】

ここで、関数 A は、クロック周波数の変化前後でサンプリング回数が増えたため、依存度の高い関数であると分かる。詳細な算出式は後述するが、ここでは、仮に関数 A の依存度は 100%であったとする。一方、関数 B は、クロック周波数の変化前後でサンプリング回数が増えなかったため、依存度の低い関数であると分かる。詳細な算出式は後述するが、ここでは、仮に関数 B の依存度は 0%であったとする。

40

【0026】

このように依存度を算出した情報抽出装置 100 は、（3）関数を実行する際に CPU 101 のクロック周波数として設定すべき周波数を、関数ごとに割り当てる。割り当てられる周波数は上述した閾値（または周波数候補テーブル）を参照して決定される。

【0027】

ここでは、図 1 に示すように閾値が 50%であったとする。関数 A は閾値より大きい依存度の関数であるから、関数 A には高クロック周波数が割り当てられる。一方、関数 B は閾値以下の依存度の関数であるから、関数 B には低クロック周波数が割り当てられる。そして、情報抽出装置 100 は、各関数に割り当てられたクロック周波数（以下、「割当クロック周波数」という）を保持するテーブル（以下、「割当テーブル」という）として主

50

記憶装置 102 に記憶する。

【0028】

その後、情報抽出装置 100 において、あらたにプログラム X を実行する際には、割当テーブルを参照して、実行中の関数に対応する割当クロック周波数へと、CPU 101 のクロック周波数を動的に変更させる。ここで、情報抽出装置 100 は、上述した特定情報を取得して実行中の関数を検出するため、ソースコード中の関数部分にクロック周波数を変更する命令を書き込まなくても、実行中の関数に対応したクロック周波数へと CPU 101 のクロック周波数を変更できる。

【0029】

このように、情報抽出装置 100 は、特定情報を取得することにより実行中の関数を検出し、実行中の関数に対応する割当クロック周波数へと CPU 101 のクロック周波数を変更する。これにより、情報抽出装置 100 は、プログラムのソースコードに手を加えることなしに、CPU 101 のクロック周波数を消費電力がより小さくなるクロック周波数へと動的に設定し、クロック周波数を低くすることによる性能の低下を抑えつつ、省電力化を図ることができる。

10

【0030】

(情報抽出装置のハードウェア構成例)

次に、図 2 を用いて情報抽出装置のハードウェア構成例について説明する。

【0031】

図 2 は、情報抽出装置のハードウェア構成例を示すブロック図である。図 2 において、情報抽出装置 100 は、CPU (Central Processing Unit) 101 と、主記憶装置 102 と、記憶装置 103 と、I/F (Interface) 104 と、を備えている。また、各構成部は、バス 110 によってそれぞれ接続されている。

20

【0032】

CPU 101 は、情報抽出装置 100 全体の制御を司る。主記憶装置 102 は、情報抽出プログラムを保持している。そして、主記憶装置 102 は、OS (Operating System) を保持している。

【0033】

また、主記憶装置 102 は、依存度の算出対象となる関数を含むプログラム (例えば、上述したプログラム X) と割当テーブル 102a を保持している。そして、記憶装置 103 は、依存度テーブル 103a と、アドレステーブル 103b と、特定情報テーブル 103c と、プロファイルテーブル 103d と、周波数候補テーブル 103e と、を記憶している。記憶装置 103 としては、不揮発性メモリやフラッシュメモリ、ハードディスクドライブなどを採用することができる。

30

【0034】

インターフェース (以下、「I/F」と略する。) 104 は、通信回線を通じて LAN (Local Area Network)、WAN (Wide Area Network)、インターネットなどのネットワーク 120 に接続され、このネットワーク 120 を介して他の装置に接続される。そして、I/F 104 は、ネットワーク 120 と内部のインターフェースを司り、外部装置からのデータの入出力を制御する。I/F 104 には、例えばモデムや LAN アダプタなどを採用することができる。

40

【0035】

(依存度テーブルの記憶内容)

次に、図 2 に示した依存度テーブル 103a の記憶内容について説明する。

【0036】

図 3 は、依存度テーブルの記憶内容を示す説明図である。図 3 に示すように、依存度テーブル 103a は、関数名項目のそれぞれに対応付けて、CPU 動作周波数依存度項目を有し、関数名ごとにレコードを構成する。

【0037】

関数名項目には、上述したプログラム X を構成する関数群のそれぞれの関数名が記憶さ

50

れる。ただし、関数名はプログラム内で関数に付与されている名称でなくてもよく、情報抽出装置 100 が関数を一意に特定する名称を任意に付与してもよい。関数名については、以下のテーブルにおいても同様である。CPU動作周波数依存度項目には、関数がCPUのクロック周波数によってどの程度律速されるかを示す度合いである依存度が記憶される。

【0038】

(割当テーブルの記憶内容)

次に、図2に示した割当テーブル102aの記憶内容について説明する。

【0039】

図4は、割当テーブルの記憶内容を示す説明図である。図4に示すように、割当テーブル102aは、関数名項目のそれぞれに対応付けて、割当クロック周波数項目を有し、関数名ごとにレコードを構成する。

10

【0040】

関数名項目には、上述したプログラムを構成する関数群のそれぞれの関数名が記憶される。割当クロック周波数項目には、関数の実行中にCPUのクロック周波数として設定すべき割当クロック周波数が記憶される。例えば、割当テーブル102aでは、上述の依存度テーブル103aに基づき、依存度の低い関数に対しては低クロック周波数が割当クロック周波数として記憶され、依存度の高い関数に対しては高クロック周波数が割当クロック周波数として記憶される。

【0041】

20

(アドレステーブルの記憶内容)

次に、図2に示したアドレステーブル103bの記憶内容について説明する。アドレステーブル103bは情報抽出装置100のユーザによって予め保持されている。

【0042】

図5は、アドレステーブルの記憶内容を示す説明図である。図5に示すように、アドレステーブル103bは、関数名項目のそれぞれに対応付けて、関数のアドレス範囲項目を有し、関数名ごとにレコードを構成する。

【0043】

関数名項目には、上述したプログラムを構成する関数群のそれぞれの関数名が記憶される。関数のアドレス範囲項目には、OSに用意されたコマンドによって特定した関数の命令アドレスの範囲が記憶される。例えば、OSがUNIX(登録商標)であるPCにおいては、nmコマンドを用いることにより関数の命令アドレスの範囲を特定できる。命令アドレスとは、CPU101が実行する命令が主記憶上へロードされた際の主記憶上の格納位置を示す値である。

30

【0044】

(特定情報テーブルの記憶内容)

次に、図2に示した特定情報テーブル103cの記憶内容について説明する。

【0045】

図6は、特定情報テーブルの記憶内容を示す説明図である。図6に示すように、特定情報テーブル103cは、時間項目のそれぞれに対応付けて、特定情報項目と、関数名項目と、を有し、サンプリングを行った時間ごとにレコードを構成する。

40

【0046】

時間項目には、上述したサンプリングを行った時間が記憶される。特定情報項目には、関数名を一意に特定するための特定情報が記憶される。特定情報とは、PID(Process Identification)および命令アドレスである。関数名項目には、特定情報により特定された関数の関数名が記憶される。

【0047】

(プロファイルテーブルの記憶内容)

次に、図2に示したプロファイルテーブル103dの記憶内容について説明する。

【0048】

50

図7は、プロファイルテーブルの記憶内容を示す説明図である。図7に示すように、プロファイルテーブル103dは、関数名項目のそれぞれに対応付けて、プロファイル結果項目を有し、関数名ごとにレコードを構成する。

【0049】

関数名項目には、上述したプログラムを構成する関数群のそれぞれの関数名が記憶される。プロファイル結果項目には、関数ごとのサンプリング回数が記憶される。関数ごとのサンプリング回数は、上述した特定情報テーブル103cにおいて、関数ごとのレコードの数を計測することで算出できる。

【0050】

(周波数候補テーブルの記憶内容)

次に、図2に示した周波数候補テーブル103eの記憶内容について説明する。周波数候補テーブル103eは情報抽出装置100のユーザによって予め保持されている。

【0051】

図8は、周波数候補テーブルの記憶内容を示す説明図である。図8に示すように、周波数候補テーブル103eは、依存度範囲項目のそれぞれに対応付けて、依存度範囲項目と、周波数候補項目と、を有し、依存度範囲ごとにレコードを構成する。

【0052】

依存度範囲項目には、上述した依存度の範囲が記憶される。周波数候補項目には、依存度の範囲ごとに、関数に割り当てる割当クロック周波数が記憶される。

【0053】

(情報抽出装置の機能的構成)

次に、図9を用いて情報抽出装置の機能的構成例について説明する。

【0054】

図9は、情報抽出装置の機能的構成例を示す機能ブロック図である。図9に示すように、情報抽出装置100は、第1の検出部901と、第2の検出部902と、第1の計測部903と、第2の計測部904と、算出部905と、出力部906と、判定部907と、格納部908と、第3の検出部909と、設定部910と、を備える。また、各機能部(第1の検出部901～設定部910)の処理結果は、特に指定する場合を除いて、例えば、記憶装置103に記憶される。

【0055】

第1の検出部901は、第1のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとにプログラム内の関数群のうちの実行中の関数を検出する機能を有する。

【0056】

ここで、プログラムとは、上述したプログラムXである。プロセッサとは、上述したCPU101である。第1のクロック周波数とは、サンプリングの際にCPU101に設定されるクロック周波数であり、例えば、上述したサンプリング条件1である。所定時間とは、上述したサンプリング間隔である。第1のクロック周波数と所定時間は、情報抽出装置100のユーザによって、予め記憶装置103に保持されている。

【0057】

具体的には、例えば、第1の検出部901は、周波数1で動作中のCPU101において、上述したPIDおよび命令アドレスを検出する。これにより、第1の検出部901は、検出したPIDおよび命令アドレスを上述したアドレス範囲と比較することにより、実行中の関数がプログラムXを構成する関数群のうちのどの関数であるかを検出できる。

【0058】

第1の検出部901は、具体的には、例えば、図2に示した主記憶装置102に保持されたプログラムをCPU101に実行させることにより、その機能を実現する。

【0059】

第2の検出部902は、第1のクロック周波数とは異なる第2のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとに関数群のうちの実行

10

20

30

40

50

中の関数を検出する機能を有する。

【0060】

ここで、第2のクロック周波数とは、サンプリングの際にCPUに設定されるクロック周波数であり、第1のクロック周波数と異なる周波数である。例えば、上述したサンプリング条件2である。所定時間とは、上述したサンプリング間隔であり、第1の検出部901のサンプリング間隔と同一値である。第2のクロック周波数と所定時間とは、情報抽出装置100のユーザによって、予め記憶装置103に保持されている。

【0061】

具体的には、例えば、第2の検出部902は、周波数2で動作中のCPU101において実行中の関数について、上述したPIDおよび命令アドレスを検出する。これにより、第2の検出部902は、検出したPIDおよび命令アドレスを上述したアドレス範囲と比較することにより、実行中の関数がプログラムXを構成する関数群のうちどの関数であるかを検出できる。

10

【0062】

第2の検出部902は、具体的には、例えば、図2に示した主記憶装置102に記憶されたプログラムをCPU101に実行させることにより、その機能を実現する。

【0063】

第1の計測部903は、関数群のうちいずれかの関数について、第1の検出部901による該関数の第1の検出回数を計測する機能を有する。ここで、いずれかの関数とは、プログラムX内の関数であって、依存度を算出する対象となる関数である。

20

【0064】

具体的には、例えば、第1の計測部903は、プログラムXの終了までに、プログラムX内の関数群のうち依存度を算出する対象となる関数を第1の検出部901が検出した回数を計測する。これにより、第1の計測部903は、第1のクロック周波数で動作中のCPU101における依存度を算出する対象となる関数のサンプリング回数を計測することができる。

【0065】

第1の計測部903は、具体的には、例えば、図2に示した主記憶装置102に保持されたプログラムをCPU101に実行させることにより、その機能を実現する。

【0066】

第2の計測部904は、いずれかの関数について、第2の検出部902による該関数の第2の検出回数を計測する機能を有する。ここで、いずれかの関数とは、プログラムX内の関数であって、依存度を算出する対象となる関数である。

30

【0067】

具体的には、例えば、第2の計測部904は、プログラムXの終了までに、プログラムX内の関数群のうち依存度を算出する対象となる関数を第2の検出部902が検出した回数を計測する。これにより、第2の計測部904は、第2のクロック周波数で動作中のCPU101における依存度を算出する対象となる関数のサンプリング回数を計測することができる。

【0068】

第2の計測部904は、具体的には、例えば、図2に示した主記憶装置102に保持されたプログラムをCPU101に実行させることにより、その機能を実現する。

40

【0069】

算出部905は、第1の検出回数から第2の検出回数への変化量に基づいて、クロック周波数の変化によるいずれかの関数の実行時間の変化の割合を示す依存度を算出する機能を有する。ここで、依存度とは、関数がCPU101のクロック周波数によってどの程度律速されるかを示す割合である。

【0070】

具体的には、算出部905は、第1のクロック周波数の逆数を第1のクロック周期として算出し、第2のクロック周波数の逆数を第2のクロック周期として算出しておく。そし

50

て、算出部 905 は、 $((\text{第 1 の 検 出 回 数} - \text{第 2 の 検 出 回 数}) / \text{第 2 の 検 出 回 数}) / ((\text{第 1 の ク ロ ッ ク 周 波 数} - \text{第 2 の ク ロ ッ ク 周 波 数}) / \text{第 2 の ク ロ ッ ク 周 波 数})) \times 100$ の式により、依存度を算出する。これにより、算出部 905 は、依存度を算出する対象となる関数の依存度を算出することができる。

【0071】

算出部 905 は、具体的には、例えば、図 2 に示した主記憶装置 102 に保持されたプログラムを CPU 101 に実行させることにより、その機能を実現する。

【0072】

出力部 906 は、算出部 905 によって算出された依存度を出力する機能を有する。具体的には、出力部 906 は、関数と依存度とをディスプレイ（不図示）に出力する。これにより、ユーザは、出力された関数と依存度とからプログラム X 内の関数の依存度を把握することができる。

10

【0073】

出力部 906 は、具体的には、例えば、図 2 に示した主記憶装置 102 に保持されたプログラムを CPU 101 に実行させることにより、その機能を実現する。

【0074】

判定部 907 は、算出部 905 によって算出された依存度が閾値以下であるか否かを判定する機能を有する。ここで、閾値とは、関数の実行時に、低クロック周波数で CPU 101 を動作させるか、高クロック周波数で CPU 101 を動作させるかを判定する境界となる値である。閾値は予め記憶装置 103 に保持されている。

20

【0075】

判定部 907 は、具体的には、例えば、図 2 に示した主記憶装置 102 に保持されたプログラムを CPU 101 に実行させることにより、その機能を実現する。

【0076】

具体的には、例えば、判定部 907 は、算出部 905 によって算出された依存度が閾値以下であるか否かを判定する。これにより、判定部 907 は、関数が、依存度が低いため低クロック周波数で CPU 101 を動作させても処理効率が大幅に下がらず電力消費を少なくできる関数であるか、依存度が高いため低クロック周波数で CPU 101 を動作させると処理効率が下がる関数であるか、を判定することができる。

【0077】

格納部 908 は、判定部 907 によって依存度が閾値以下であると判定された場合に、所定の低周波数の割当周波数をいずれかの関数と関連付けて、割当テーブル 102 a に格納し、判定部 907 によって依存度が閾値より大きいと判定された場合に、所定の高周波数の割当周波数をいずれかの関数と関連付けて、割当テーブル 102 a に格納する機能を有する。

30

【0078】

ここで、所定の低周波数の割当クロック周波数とは、依存度の低い関数の実行時に CPU 101 のクロック周波数として設定する割当クロック周波数である。所定の高周波数の割当クロック周波数とは、依存度の高い関数の実行時に CPU 101 のクロック周波数として設定する割当クロック周波数である。所定の低クロック周波数および所定の高クロック周波数は予め記憶装置 103 に保持されている。

40

【0079】

具体的には、例えば、格納部 908 は、関数に割り当てるべき割当クロック周波数を関数と関連付けて割当テーブル 102 a に格納することができる。これにより、格納部 908 は、高周波数の割当クロック周波数と低周波数の割当クロック周波数とのうち、関数に割り当てるべき割当クロック周波数を関数と関連付けて割当テーブル 102 a に格納することができる。

【0080】

また、格納部 908 は、依存度ごとに対応する割当クロック周波数を保持するテーブルに基づいて、算出部 905 によって算出された依存度に対応する割当クロック周波数を任

50

意の関数と関連付けて割当テーブル102aに格納する機能を有する。依存度ごとに対応する割当クロック周波数を保持するテーブルとは、上述した周波数候補テーブル103eである。

【0081】

具体的には、例えば、格納部908は、算出部905によって算出された依存度が、テーブルのどの範囲に含まれる依存度であるかに基づき、対応する割当クロック周波数を抽出し、抽出した割当クロック周波数と関数とを関連付けて割当テーブル102aに格納する。これにより、格納部908は、関数に割り当てるべき割当クロック周波数の候補を複数用意しておき、その候補のうちから依存度に基づき抽出した割当クロック周波数を関数と関連付けて割当テーブル102aに格納することができる。

10

【0082】

格納部908は、具体的には、例えば、図2に示した主記憶装置102に保持されたプログラムをCPU101に実行させることにより、および、記憶装置103により、その機能を実現する。

【0083】

第3の検出部909は、プログラムをあらたに実行した場合において、関数群のうちの実行中の関数を検出する機能を有する。ここで、あらたに実行したプログラムとは、上述したサンプリングのためのプログラムXの実行の終了後に、再度実行されたプログラムXである。

【0084】

20

具体的には、例えば、第3の検出部909は、実行中の関数について、上述したPIDおよび命令アドレスを検出する。これにより、第3の検出部909は、検出したPIDおよび命令アドレスを上述したアドレス範囲と比較することにより、実行中の関数がプログラムXを構成する関数群のうちどの関数であるかを検出できる。

【0085】

第3の検出部909は、具体的には、例えば、図2に示した主記憶装置102に保持されたプログラムをCPU101に実行させることにより、その機能を実現する。

【0086】

設定部910は、前記割当テーブル102aに基づいて、第3の検出部909によって検出された関数に対応する割当クロック周波数をプロセッサのクロック周波数として設定する機能を有する。

30

【0087】

具体的には、例えば、設定部910は、割当テーブル102aに基づき、実行中の関数に対応する割当クロック周波数をCPU101のクロック周波数として設定する。これにより、設定部910は、実行中の関数に対して設定すべきクロック周波数をCPU101に設定できる。例えば、設定部910は、依存度の低い関数の実行中は低周波数の割当クロック周波数でCPU101を動作させ、省電力化を図ることができる。

【0088】

設定部910は、具体的には、例えば、図2に示した主記憶装置102に保持されたプログラムをCPU101に実行させることにより、その機能を実現する。

40

【0089】

(依存度の算出の内容)

次に、図10～図13を用いて、情報抽出装置による依存度の算出の内容について説明する。

【0090】

図10は、情報抽出装置による依存度の算出の内容を示す説明図である。依存度を算出する場合には、情報抽出装置100は、第1の計測部903および第2の計測部904によってサンプリングを行い、特定情報を計測する。次に、情報抽出装置100は、計測された特定情報に基づき、算出部905によって依存度を算出する。そして、算出された依存度が依存度テーブル103aに格納される。

50

【 0 0 9 1 】

図 1 1 は、特定情報の内容を示す説明図である。特定情報には、P I D と命令アドレスが含まれている。P I D とは、実行中のプログラムごとに C P U 1 0 1 から一意に割り振られる識別符号であり、例えば、同一のプログラムが同時に複数実行されている場合であってもそれぞれの実行中のプログラムを識別することができる。命令アドレスは、実行中の命令の主記憶上の格納位置を示す値であり、この特定情報を用いて情報抽出装置 1 0 0 はサンプリング回数を計測する。

【 0 0 9 2 】

図 1 2 は、特定情報を用いてサンプリング回数を計測する具体例を示す説明図である。ここで、情報抽出装置 1 0 0 は、 t 、 $t + 1$ 、 $t + 2$ 、 $t + 3$ のそれぞれの時間において特定情報を取得する。また、情報抽出装置 1 0 0 の記憶装置 1 0 3 には、予めプログラム X を構成する関数のアドレス範囲を保持したアドレステーブル 1 0 3 b が記憶されている。

10

【 0 0 9 3 】

ここで、アドレステーブル 1 0 3 b から、特定情報内の命令アドレスに対応する関数を抽出することにより、実行中の関数がどの関数であるかを判別することができる。また、ここでは、実行中のプログラムは 1 つだけであったが、同時に実行されている他のプログラムがあっても、特定情報内の P I D を用いることによりプログラムごとにサンプリングできるため、他のプログラムのサンプリング回数を誤って合算することを防ぐことができる。

20

【 0 0 9 4 】

このように、プログラム X において実行中の関数をサンプリングし、特定情報テーブル 1 0 3 c として記憶しておくことができる。また、特定情報テーブル 1 0 3 c に基づき関数ごとにサンプリング回数を計測することができる。そして、情報抽出装置 1 0 0 は、このサンプリング回数を関数ごとに保持するプロファイルテーブル 1 0 3 d を作成する。

【 0 0 9 5 】

次に、プロファイルテーブル 1 0 3 d に基づく依存度の算出の内容について説明する。図 1 3 は、プログラムの測定の内容を示す説明図である。ここでは、関数 A を 5 0 0 0 回、関数 B を 5 0 0 0 回実行すると終了するプログラム X を例に挙げる。また、C P U 1 0 1 のクロック周波数を 1 [G H z] (クロック周期は 1 [n s]) および 2 [G H z] (クロック周期は 0 . 5 [n s]) に変更した場合のサンプリング回数を用いて依存度を算出するとする。

30

【 0 0 9 6 】

図 1 3 において、A および B は関数名を示し、関数名の下に記した時間は、関数の 1 回の実行時間である。従って、1 [G H z] 時では、このプログラム X の実行時間は 1 5 0 [s e c] となり、2 [G H z] 時では 1 0 0 [s e c] となる。

【 0 0 9 7 】

この例において、各クロック周波数において 1 [m s e c] ごとのサンプリングを行う。1 [G H z] 時においては、約 1 5 0 0 0 0 回サンプリングが行われ、関数 A は約 1 0 0 0 0 0 回、関数 B は約 5 0 0 0 0 回サンプリングされる。一方、2 [G H z] 時においては、約 1 0 0 0 0 0 回サンプリングが行われ、関数 A は約 5 0 0 0 0 回、関数 B は約 5 0 0 0 0 回サンプリングされる。

40

【 0 0 9 8 】

このサンプリング回数を用いて上述した依存度の算出式に基づいて依存度を算出すると、関数 A の依存度は、以下ようになる。

【 0 0 9 9 】

$$[\{ (1 0 0 0 0 0 - 5 0 0 0 0) / 5 0 0 0 0 \} / \{ (1 - 0 . 5) / 0 . 5 \}] \times 1 0 0 = 1 0 0 [\%]$$

【 0 1 0 0 】

また、関数 B の依存度は、以下ようになる。

50

【 0 1 0 1 】

$[\{ (5 0 0 0 0 - 5 0 0 0 0) / 5 0 0 0 0 \} / \{ (1 - 0 . 5) / 0 . 5 \}] \times 1 0 0 = 0 [\%]$

【 0 1 0 2 】

次に、図 1 4 を用いて、作成した依存度テーブル 1 0 3 a に基づき関数実行時のクロック周波数の制御の内容について説明する。

【 0 1 0 3 】

図 1 4 は、関数実行時のクロック周波数の制御の内容を示す説明図である。情報抽出装置 1 0 0 は、格納部 9 0 8 により、依存度テーブル 1 0 3 a に基づき関数の実行時に CPU 1 0 1 のクロック周波数として設定する割当クロック周波数を決定する。

10

【 0 1 0 4 】

例えば、割当クロック周波数を高割当クロック周波数と低割当クロック周波数とのどちらかとする場合、依存度に対する閾値を設定しておき、閾値以下か否かによって関数に対応する割当クロック周波数を決定する。そして、割当テーブル 1 0 2 a に関数と割当クロック周波数とを関連付けて格納し、割当テーブル 1 0 2 a を作成する。

【 0 1 0 5 】

割当テーブル 1 0 2 a の作成後、再度プログラム X を実行する場合には、情報抽出装置 1 0 0 は、第 3 の検出部 9 0 9 によって実行中の関数を検出する。そして、実行中の関数に対応する割当クロック周波数を割当テーブル 1 0 2 a から抽出し、抽出した割当クロック周波数を設定部 9 1 0 により CPU 1 0 1 のクロック周波数として設定する。

20

【 0 1 0 6 】

このようにして、情報抽出装置 1 0 0 は、実行中の関数に対応する割当クロック周波数によって CPU 1 0 1 を動作させることができる。また、情報抽出装置 1 0 0 は、一定時間ごとに実行中の関数を検出し、実行中の関数に対応する割当クロック周波数を CPU 1 0 1 のクロック周波数として設定する。従って、関数の実行終了または実行開始に合わせて、クロック周波数を動的に変更していくことができる。

【 0 1 0 7 】

次に、図 1 5 に示した依存度算出処理の詳細について説明する。

【 0 1 0 8 】

図 1 5 は、依存度算出処理の詳細を示すフローチャートである。図 1 5 に示すように、CPU 1 0 1 は、クロック周波数を変更する変更前後において、タイムベースサンプリングによって、プログラム X 内の関数のサンプリング回数を測定する（ステップ S 1 5 0 1 ）。次に、CPU 1 0 1 は、上述した算出式を用いて、プログラム X 内の関数の依存度を算出して（ステップ S 1 5 0 2 ）、依存度算出処理を終了する。

30

【 0 1 0 9 】

これにより、関数のクロック周波数への依存度が算出され、依存度テーブル 1 0 3 a が作成される。

【 0 1 1 0 】

次に、図 1 6 に示したプログラム実行処理の詳細について説明する。

【 0 1 1 1 】

図 1 6 は、プログラム実行処理の詳細を示すフローチャートである。図 1 6 に示すように、CPU 1 0 1 は、後述する割当テーブル作成処理によって、関数ごとに対応する割当クロック周波数を保持する割当テーブル 1 0 2 a を作成する（ステップ S 1 6 0 1 ）。次に、CPU 1 0 1 は、プログラム X を開始し（ステップ S 1 6 0 2 ）、後述するクロック周波数制御処理によって、一定時間ごとに CPU 1 0 1 のクロック周波数を制御する（ステップ S 1 6 0 3 ）。そして、プログラム実行処理を終了する。

40

【 0 1 1 2 】

これにより、CPU 1 0 1 は、関数ごとに割当クロック周波数を関連付けて保持する割当テーブル 1 0 2 a を作成することができる。また、所定時間ごとに実行中の関数を検出し、割当テーブル 1 0 2 a に基づき、検出した関数に対応付けられた割当クロック周波数

50

をCPU101のクロック周波数として設定することができる。

【0113】

次に、図17に示した割当テーブル作成処理の詳細について説明する。

【0114】

図17は、割当テーブル作成処理の詳細を示すフローチャートである。図17に示すように、まず、CPU101は、依存度が閾値より大きいか否かを判定する(ステップS1701)。依存度が閾値より大きい場合は(ステップS1701: Yes)、CPU101は、割当クロック周波数として高クロック周波数を選択する(ステップS1702)。一方、依存度が閾値以下の場合は(ステップS1701: No)、CPU101は、割当クロック周波数として低クロック周波数を選択する(ステップS1703)。

10

【0115】

次に、CPU101は、選択された割当クロック周波数を割当テーブル102aに格納する(ステップS1704)。そして、CPU101は、依存度テーブル103aに依存度が保持されている関数のうち割当クロック周波数が未決定の関数があるか否かを判定する(ステップS1705)。割当クロック周波数が未決定の関数がある場合には(ステップS1705: Yes)、CPU101は、ステップS1701に戻り、未決定の関数について割当クロック周波数を決定する。一方、割当クロック周波数が未決定の関数がない場合には(ステップS1705: No)、割当テーブル作成処理を終了する。

【0116】

これにより、関数ごとに割当クロック周波数を関連付けて保持する割当テーブル102aを作成することができる。

20

【0117】

次に、図18に示したクロック周波数制御処理の詳細について説明する。

【0118】

図18は、クロック周波数制御処理の詳細を示すフローチャートである。図18に示すように、CPU101は、サンプリングした特定情報から実行中の関数を検出する(ステップS1801)。次に、CPU101は、検出した実行中の関数に対応する割当クロック周波数を、割当テーブル102aから抽出する(ステップS1802)。

【0119】

そして、CPU101は、CPU101の現在のクロック周波数と、抽出した割当クロック周波数とが異なるか否かを判定する(ステップS1803)。現在のCPU101のクロック周波数と、抽出した割当クロック周波数とが同じ場合(ステップS1803: No)、クロック周波数制御処理を終了する。

30

【0120】

一方、現在のCPU101のクロック周波数と、抽出した割当クロック周波数とが異なる場合(ステップS1803: Yes)、CPU101は、割当クロック周波数をCPU101のクロック周波数として設定して(ステップS1804)、クロック周波数制御処理を終了する。

【0121】

これにより、実行中の関数を検出し、割当テーブル102aに基づき、検出した関数に対応付けられた割当クロック周波数をCPU101のクロック周波数として設定することができる。また、一定時間ごとに実行中の関数を検出しているため、実行中の関数が終了し、あらたな関数が実行された場合にも、あらたに実行された関数を検出して、動的にCPU101のクロック周波数を変更していくことができる。

40

【0122】

このように、上述した実施の形態では、情報抽出装置100は、クロック周波数を変化させ、変化前後での関数の実行時間を測定して、クロック周波数の変化に対する実行時間の変化を調べて関数の依存度を算出する。これにより情報抽出装置100は、ソースコードの不明なプログラム(例えば、他者からの改変を防ぐためにバイナリ形式で提供されるプログラム)に対しても、プログラム内の関数の依存度を算出することができる。

50

【 0 1 2 3 】

一方、ソースコードに手を加えることが可能なプログラムであっても、プログラムのソースコードに手を加えることなしに関数の依存度を算出することができる。そのため、プログラマが手作業でソースコードに命令を書き込む必要がなく、書き込みに要していた時間や手間を削減することができる。また、書き込み時のヒューマンエラーを回避することができる。

【 0 1 2 4 】

そして、情報抽出装置 1 0 0 のユーザが記憶装置 1 0 3 に閾値（または周波数候補テーブル 1 0 3 e）を保持しておくことで、情報抽出装置 1 0 0 は、関数に対応付ける割当クロック周波数を自動的に決定し、割当テーブル 1 0 2 a を作成することができる。そのため、割当テーブル 1 0 2 a の作成に要する時間を削減することができる。また、予め閾値（または周波数候補テーブル 1 0 3 e）が保持されていれば、情報抽出装置 1 0 0 のユーザが依存度と割当クロック周波数の対応付けに詳しくなくても、割当テーブル 1 0 2 a を作成することができる。

10

【 0 1 2 5 】

また、情報抽出装置 1 0 0 は、算出した依存度を出力する。そのため、情報抽出装置 1 0 0 のユーザは、一律な基準に基づいて割当クロック周波数を決定するのではなく、出力された関数の依存度に基づいて、各関数の実行に最適な割当クロック周波数を関数ごとに決定することができる。

【 0 1 2 6 】

ここで、最適なクロック周波数とは、最も消費電力の少なくなるクロック周波数を採用してもよいし、他のプログラムの処理効率を下げないように高クロック周波数を採用してもよい。

20

【 0 1 2 7 】

そして、情報抽出装置 1 0 0 は、特定情報から実行中の関数を検出し、主記憶装置 1 0 2 に保持された割当テーブル 1 0 2 a に基づき、実行中の関数の依存度に対応した割当クロック周波数を CPU 1 0 1 に動的に設定していく。従って、プログラムに対して「関数を実行する際にクロック周波数を変更する」という命令を書き加えることができない場合であっても、情報抽出装置 1 0 0 は、実行中の関数を検出しクロック周波数を変更することができる。

30

【 0 1 2 8 】

さらに、情報抽出装置 1 0 0 は、一定時間ごとに実行中の関数を検出しているため、実行中の関数が終了し、あらたな関数が実行された場合にも、あらたに実行された関数を検出して、動的に CPU 1 0 1 のクロック周波数を変更していくことができる。

【 0 1 2 9 】

このように、情報抽出装置 1 0 0 は、実行中の関数に対応した割当クロック周波数によって CPU 1 0 1 を動作させるようにしたため、依存度の低い関数の実行時には低クロック周波数で CPU 1 0 1 を動作させて省電力化を図ることができる。一方、依存度の高い関数の実行時には高クロック周波数で CPU 1 0 1 を動作させてパフォーマンスの低下を抑えることができる。

40

【 0 1 3 0 】

上述した実施の形態に関し、さらに以下の付記を開示する。

【 0 1 3 1 】

（付記 1）第 1 のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとに前記プログラム内の関数群のうちの実行中の関数を検出する第 1 の検出工程と、

前記第 1 のクロック周波数とは異なる第 2 のクロック周波数で動作中の前記プロセッサによる前記プログラムの実行中において、前記所定時間ごとに前記関数群のうちの実行中の関数を検出する第 2 の検出工程と、

前記関数群のうちいずれかの関数について、前記第 1 の検出工程による前記いずれか

50

の関数の第 1 の検出回数を計測する第 1 の計測工程と、

前記いずれかの関数について、前記第 2 の検出工程による前記いずれかの関数の第 2 の検出回数を計測する第 2 の計測工程と、

前記第 1 の検出回数から前記第 2 の検出回数への変化量に基づいて、クロック周波数の変化による前記いずれかの関数の実行時間の変化の度合いを示す依存度を算出する算出工程と、

前記算出工程によって算出された依存度を出力する出力工程と、

をコンピュータに実行させることを特徴とする情報抽出プログラム。

【0132】

(付記 2) 前記算出工程によって算出された依存度が閾値以下であるか否かを判定する判定工程と、

前記判定工程によって依存度が閾値以下であると判定された場合に、所定の低周波数の割当クロック周波数を前記いずれかの関数と関連付けて、割当テーブルに格納し、前記判定工程によって依存度が閾値より大きいと判定された場合に、所定の高周波数の割当クロック周波数を前記いずれかの関数と関連付けて、前記割当テーブルに格納する格納工程と、

を前記コンピュータに実行させることを特徴とする付記 1 に記載の情報抽出プログラム

。

【0133】

(付記 3) 前記依存度ごとに対応する割当クロック周波数を保持するテーブルに基づいて、前記算出工程によって算出された依存度に対応する割当クロック周波数を前記いずれかの関数と関連付けて割当テーブルに格納する格納工程を前記コンピュータに実行させることを特徴とする付記 1 に記載の情報抽出プログラム。

【0134】

(付記 4) 前記プログラムをあらたに実行した場合において、前記関数群のうちの実行中の関数を検出する第 3 の検出工程と、

前記割当テーブルに基づいて、前記第 3 の検出工程によって検出された関数に対応する前記割当クロック周波数を前記プロセッサのクロック周波数として設定する設定工程と、

を前記コンピュータに実行させることを特徴とする付記 2 または 3 に記載の情報抽出プログラム。

【0135】

(付記 5) 第 1 の検出手段と、第 2 の検出手段と、第 1 の計測手段と、第 2 の計測手段と、算出手段と、出力手段と、を備えるコンピュータが、

前記第 1 の検出手段により、第 1 のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとに前記プログラム内の関数群のうちの実行中の関数を検出する第 1 の検出工程と、

前記第 2 の検出手段により、前記第 1 のクロック周波数とは異なる第 2 のクロック周波数で動作中の前記プロセッサによる前記プログラムの実行中において、前記所定時間ごとに前記関数群のうちの実行中の関数を検出する第 2 の検出工程と、

前記第 1 の計測手段により、前記関数群のうちいずれかの関数について、前記第 1 の検出工程による前記いずれかの関数の第 1 の検出回数を計測する第 1 の計測工程と、

前記第 2 の計測手段により、前記いずれかの関数について、前記第 2 の検出工程による前記いずれかの関数の第 2 の検出回数を計測する第 2 の計測工程と、

前記算出手段により、前記第 1 の検出回数から前記第 2 の検出回数への変化量に基づいて、クロック周波数の変化による前記いずれかの関数の実行時間の変化の度合いを示す依存度を算出する算出工程と、

前記出力手段により、前記算出工程によって算出された依存度を出力する出力工程と、

を実行することを特徴とする情報抽出方法。

【0136】

(付記 6) 判定手段と、格納手段と、をさらに備える前記コンピュータが、

10

20

30

40

50

前記判定手段により、前記算出工程によって算出された依存度が閾値以下であるか否かを判定する判定工程と、

前記格納手段により、前記判定工程によって依存度が閾値以下であると判定された場合に、所定の低周波数の割当クロック周波数を前記いずれかの関数と関連付けて、割当テーブルに格納し、前記判定工程によって依存度が閾値より大きいと判定された場合に、所定の高周波数の割当クロック周波数を前記いずれかの関数と関連付けて、前記割当テーブルに格納する格納工程と、

を実行することを特徴とする付記 5 に記載の情報抽出方法。

【0137】

(付記 7) 格納手段をさらに備える前記コンピュータが、

前記格納手段により、前記依存度ごとに対応する割当クロック周波数を保持するテーブルに基づいて、前記算出工程によって算出された依存度に対応する割当クロック周波数を前記いずれかの関数と関連付けて割当テーブルに格納する格納工程を実行することを特徴とする付記 5 に記載の情報抽出方法。

【0138】

(付記 8) 第 3 の検出手段と、設定手段と、をさらに備える前記コンピュータが、

前記検出手段により、前記プログラムをあらたに実行した場合において、前記関数群のうちの実行中の関数を検出する第 3 の検出工程と、

前記設定手段により、前記割当テーブルに基づいて、前記第 3 の検出工程によって検出された関数に対応する前記割当クロック周波数を前記プロセッサのクロック周波数として設定する設定工程と、

を実行することを特徴とする付記 6 または 7 に記載の情報抽出方法。

【0139】

(付記 9) 第 1 のクロック周波数で動作中のプロセッサによるプログラムの実行中において、所定時間ごとに前記プログラム内の関数群のうちの実行中の関数を検出する第 1 の検出手段と、

前記第 1 のクロック周波数とは異なる第 2 のクロック周波数で動作中の前記プロセッサによる前記プログラムの実行中において、前記所定時間ごとに前記関数群のうちの実行中の関数を検出する第 2 の検出手段と、

前記関数群のうちいずれかの関数について、前記第 1 の検出手段による前記いずれかの関数の第 1 の検出回数を計測する第 1 の計測手段と、

前記いずれかの関数について、前記第 2 の検出手段による前記いずれかの関数の第 2 の検出回数を計測する第 2 の計測手段と、

前記第 1 の検出回数から前記第 2 の検出回数への変化量に基づいて、クロック周波数の変化による前記いずれかの関数の実行時間の変化の度合いを示す依存度を算出する算出手段と、

前記算出手段によって算出された依存度を出力する出力手段と、

を備えることを特徴とする情報抽出装置。

【0140】

(付記 10) 前記算出手段によって算出された依存度が閾値以下であるか否かを判定する判定手段と、

前記判定手段によって依存度が閾値以下であると判定された場合に、所定の低周波数の割当クロック周波数を前記いずれかの関数と関連付けて、割当テーブルに格納し、前記判定手段によって依存度が閾値より大きいと判定された場合に、所定の高周波数の割当クロック周波数を前記いずれかの関数と関連付けて、前記割当テーブルに格納する格納手段と、

を備えることを特徴とする付記 9 に記載の情報抽出装置。

【0141】

(付記 11) 前記依存度ごとに対応する割当クロック周波数を保持するテーブルに基づいて、前記算出手段によって算出された依存度に対応する割当クロック周波数を前記いずれ

10

20

30

40

50

かの関数と関連付けて割当テーブルに格納する格納手段を備えることを特徴とする付記9に記載の情報抽出装置。

【0142】

(付記12)前記プログラムをあらたに実行した場合において、前記関数群のうちの実行中の関数を検出する第3の検出手段と、

前記割当テーブルに基づいて、前記第3の検出手段によって検出された関数に対応する前記割当クロック周波数を前記プロセッサのクロック周波数として設定する設定手段と、を備えることを特徴とする付記10または11に記載の情報抽出装置。

【符号の説明】

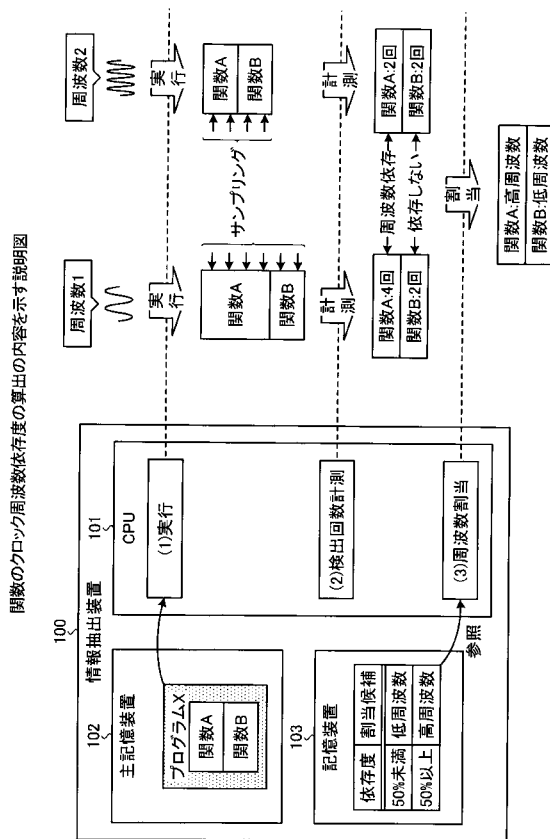
【0143】

- 100 情報抽出装置
- X プログラム
- 901 第1の検出部
- 902 第2の検出部
- 903 第1の計測部
- 904 第2の計測部
- 905 算出部
- 906 出力部
- 907 判定部
- 908 格納部
- 909 第3の検出部
- 910 設定部

10

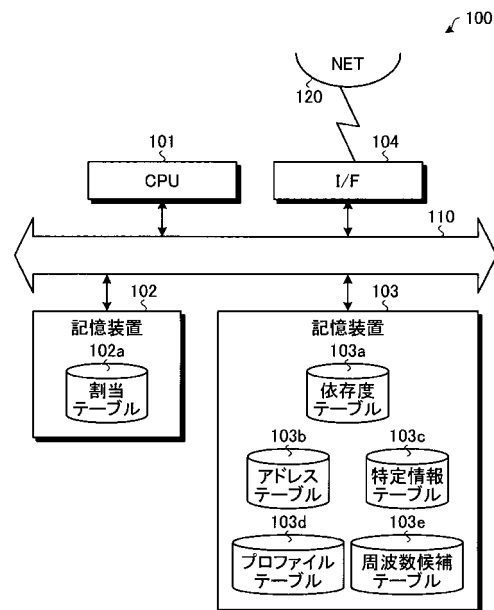
20

【図1】



【図2】

情報抽出装置のハードウェア構成例を示すブロック図



【 図 3 】

依存度テーブルの記憶内容を示す説明図

↙ 103a

関数名	CPU動作周波数依存度
関数A	100%
関数B	0%

【 図 4 】

割当テーブルの記憶内容を示す説明図

↙ 102a

関数名	割当クロック周波数
関数A	2GHz
関数B	1GHz

【 図 5 】

アドレステーブルの記憶内容を示す説明図

↙ 103b

関数名	関数のアドレス範囲
関数A	0x00~0x20
関数B	0x21~0x40

【 図 6 】

特定情報テーブルの記憶内容を示す説明図

↙ 103c

時間	特定情報 (PID, 命令アドレス)	関数名
t	(100, 0x12)	関数A
t+1	(100, 0x26)	関数B
t+2	(100, 0x14)	関数A
t+3	(100, 0x18)	関数A

【 図 7 】

プロフィールテーブルの記憶内容を示す説明図

↙ 103d

関数名	プロフィール結果
関数A	3
関数B	1

【 図 8 】

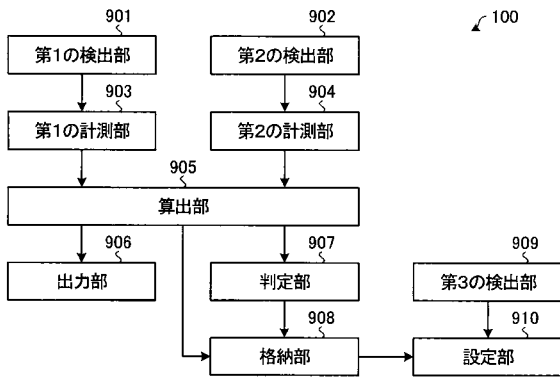
周波数候補テーブルの記憶内容を示す説明図

↙ 103e

依存度範囲	周波数候補
0%以上30%未満	1GHz
30%以上60%未満	2GHz
60%以上100%以下	3GHz

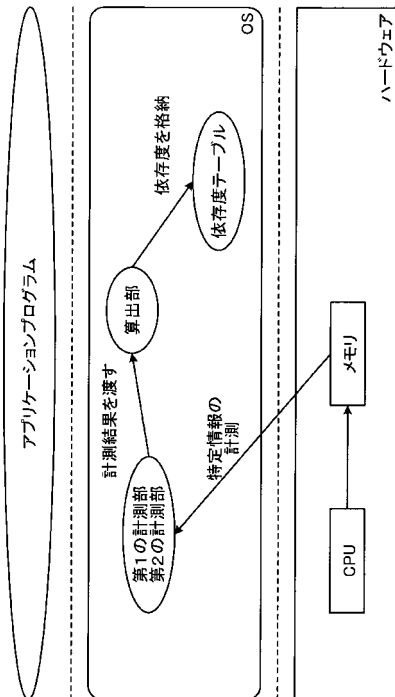
【 図 9 】

情報抽出装置の機能的構成を示す機能ブロック図



【 図 10 】

情報抽出装置による依存度の算出の内容を示す説明図



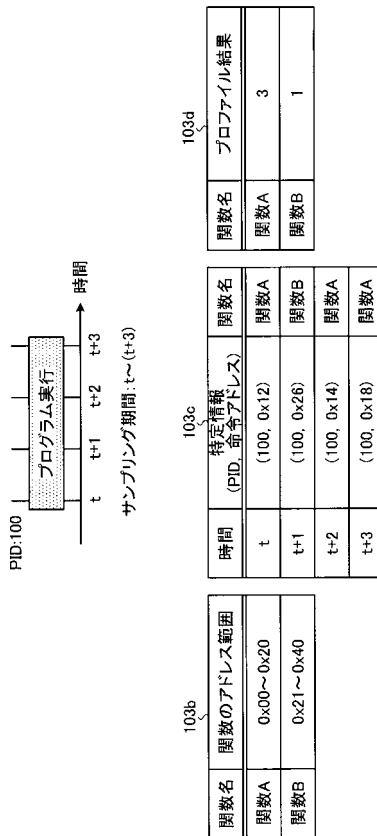
【 図 1 1 】

特定情報の内容を示す説明図

PID:100
命令アドレス:0x10

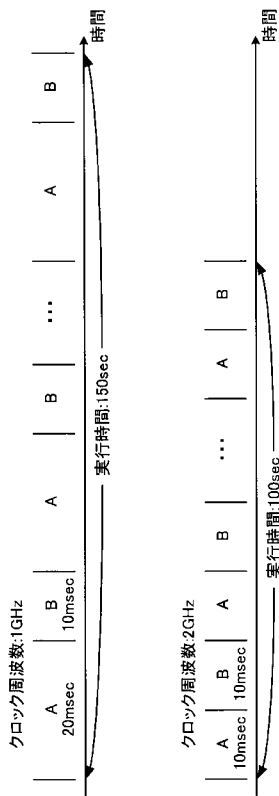
【 図 1 2 】

特定情報を用いてサンプリング回数を計測する具体例を示す説明図



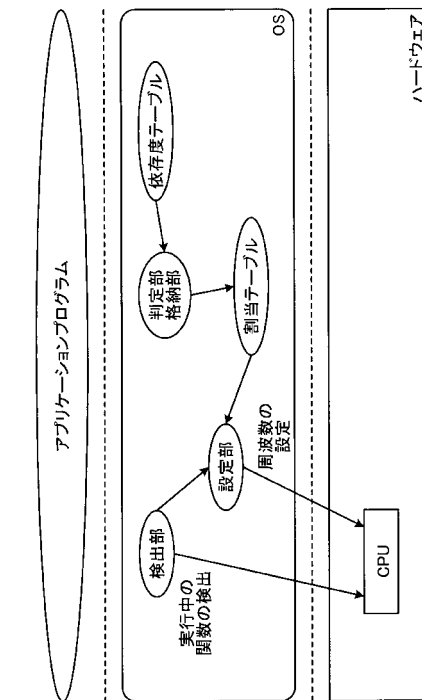
【 図 1 3 】

プログラムの判定の内容を示す説明図



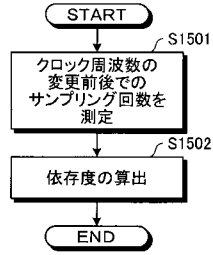
【 図 1 4 】

関数実行時のクロック周波数の制御の内容を示す説明図



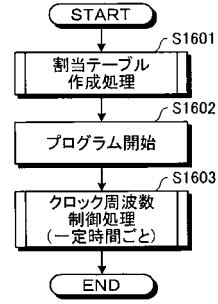
【 図 1 5 】

依存度算出処理の詳細を示すフローチャート



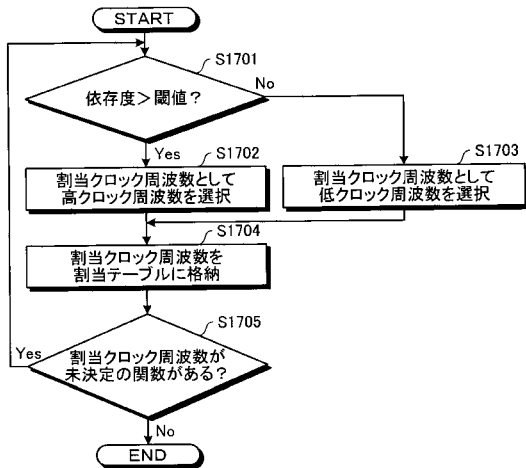
【 図 1 6 】

プログラム実行処理の詳細を示すフローチャート



【 図 1 7 】

割当テーブル作成処理の詳細を示すフローチャート



【 図 1 8 】

クロック周波数制御処理の詳細を示すフローチャート

