

Notice

This translation is machine-generated. It cannot be guaranteed that it is intelligible, accurate, complete, reliable or fit for specific purposes. Critical decisions, such as commercially relevant or financial decisions, should not be based on machine-translation output.

DESCRIPTION JPH11149489A

[0001]

13 BACKGROUND OF THE INVENTION 1. Field of the Invention The present invention relates to a simulation apparatus that can automatically simulate operation timing, which has been difficult in the past, when simulating a processor or the like.

17 【発明の属する技術分野】 本発明は、プロセッサ等のシミュレーションを行う際に、従来困難であった動作タイミングのシミュレーションを自動的に行うことのできるシミュレーション装置に関する。

[0002]

25 [Prior Art] Conventionally, processor instruction set simulators (also referred to as simulation models) have been broadly divided into simulation models written in HDL (Verilog, VHDL, etc.) and simulation models written in C language. .

29 【従来の技術】 従来では、プロセッサの命令セット・シミュレータ（シミュレーション・モデルともいう）には、大きく分けてHDL（Verilog,VHDL等）記述によるシミュレーション・モデルとC言語によるシミュレーション・モデルとがあった。

[0003]

37 However, simulation models written in HDL (Verilog, VHDL, etc.) have low processing

performance and usually lack a debugging function, making it difficult to use them for software development.

41 しかし、HDL（Verilog,VHDL等）記述によるシミュレーション・モデルでは、処理性能が低く、また通常はデバッグ機能が無いためにソフト開発に利用することは難しかった。

[0004]

48 Therefore, simulation models written in C language are generally used for software development.

51 そこで、ソフト開発には、一般にC言語によるシミュレーション・モデルが使用されている。

54 This simulation model in C language further includes an instruction-level simulator that can simulate only the functional aspect, and a cycle-level simulator that can simulate not only the functions but also the operational timing of which processor instructions are executed and when. It can be classified into two types: accurate simulators and accurate simulators.

59 このC言語によるシミュレーション・モデルは、さらに、機能面のみをシミュレーション可能な命令レベル・シミュレータと、機能に併せて、プロセッサのどの命令が、いつ実行されるかという動作タイミングについてもシミュレーション可能なcycle-accurateなシミュレータとの2つに分類することができる。

[0005]

68 When developing software with strict timing constraints and co-verifying HW (hardware)/SW (software), an instruction-level simulator that can only simulate functionality cannot perform sufficient simulations, but it is also possible to simulate operational timing. A cycle-accurate simulator was needed.

73 タイミング制約の厳しいソフトの開発やHW（ハードウェア）／SW（ソフトウェア）協調検証においては、機能面についてのシミュレーションしかできない命令レベル・シミュレータだけでは十分なシミュレーションが行えず、動作タイミングについてもシミュレーション可能なcycle-accurateなシミュレータが必要であった。

[0006]

82 Here, a conventional cycle-accurate simulator will be explained.

84 ここで、従来のcycle-accurateなシミュレータについて説明する。

[0007]

90 Conventionally, when simulating operation timing with a cycle-accurate simulator, a program for simulating the operation timing was created and the program was executed on the cycle-accurate simulator to simulate the operation timing.

94 従来ではcycle-accurateなシミュレータで動作タイミングをシミュレーションする場合には、動作タイミングをシミュレーションするためのプログラムを作成し、そのプログラムをcycle-accurateなシミュレータで実行することにより動作タイミングをシミュレーションしていた。

99 There were two ways to create the program: one was created by converting the HDL description into C language, and the other was created by hand. However, when converting from HDL description to C language, software simulation Because they contain unnecessary structural descriptions, they have the problem of slowing down processing speed, so they are often created manually.

105 そのプログラムの作成方法としては、HDL記述からC言語に変換して作られる場合と人手により作成される場合との2つがあったが、HDL記述からC言語に変換される場合にはソフトのシミュレーションには不要な構造の記述が含まれているために、処理速度が低くなるという問題点があり、そのために人手で作成される場合が多かった。

[0008]

114 [Problem to be Solved by the Invention] However, in order to create a program that simulates the operation timing manually as described above, there is a problem in that a large number of man-hours are required for its development (particularly the implementation and verification of the operation timing). There was a point.

119 【発明が解決しようとする課題】しかしながら、上述したような人手で動作タイミングをシミュレーションするプログラムを作成するには、その開発（特に、動作タイミングについての実装と検証）に大きな工数が必要となる問題点があった。

[0009]

127 When creating a program manually, the program is first created based on the specifications of the processor to be simulated, but the functional aspects of simulation are usually clearly stated in the specifications. , it was possible to easily implement/verify it manually (for example, it could be implemented/verified in about a week).

132 それは人手でプログラムを作成する場合には、まずシミュレーションしようとしているプロセッサの仕様書に基づいてプログラムを作成するのであるが、シミュレーションのうち機能面については、通常は仕様書に明確に記載されており、人手でも容易に実装／検証をすることができた（例えば、1週間程度で実装／検証が可能である）。

138 On the other hand, regarding the operational timing, a large number of man-hours were required for implementation/verification (for example, it may take about one year).

141 これに対し、動作タイミングの面については、実装／検証に大きな工数が必要であった（例えば、1年程度必要となる場合もある）。

[0010]

148 The reason why this large amount of man-hour is required is that, firstly, it takes a lot of time for the person creating the program to understand the specifications, and secondly, all the operation timings are based on the specifications. It may not always be written down in books, and it may take a great deal of effort to find the parts that are not written down.

153 この大きな工数が必要となる理由としては、まず第1に、プログラムを作成する人が仕様を理解することに多くの時間が必要であることが考えられ、第2には動作タイミングは全てが仕様書に記されているとは限らず、記されていない部分を見つけることに大きな労力を要するということがある。

159 Furthermore, there are cases in which specifications for operation timing are not written down at all, and in such cases it is very difficult to implement operation timing itself.

162 また、動作タイミングの仕様が全く記されていない場合もあり、このような場合には動作タイミングを実装すること自体が大変困難であった。

[0011]

169 The present invention has been made in view of the above circumstances, and its purpose is to automate the implementation of operation timing, which previously required a lot of effort, in the case of simulation, and to make it possible to automate the implementation of operation timing, which previously required a lot of effort. An object of the present invention is to provide a simulation device that enables simulation of operation timing even when the operation timing is unknown.

176 本発明は上記事情に鑑みてなされたものであり、その目的は、シミュレーションする場合において、多くの労力を必要としていた動作タイミングの実装を自動化することを可能にすると共に、動作タイミングの仕様が難解であったり、不明な場合であっても動作タイミングのシミュレーションを可能にするシミュレーション装置を提供することにある。

[0012]

185 [Means for Solving the Problems] In order to achieve the above object, a simulation device according to the first invention includes an input means for inputting a program for setting operation timing and a program to be simulated, and an input means for inputting a program for setting operation timing and a program to be simulated. Program counter/clock cycle number correspondence that executes the input operation timing setting program and generates a program counter/clock cycle number correspondence table that shows the relationship between the number of clock cycles and the program counter of each stage of the pipeline. An instruction to execute a program for setting operation timing and a program to be simulated by referring to the table generation means and the program counter/clock cycle number correspondence table generated by the program counter/clock cycle number correspondence table generation means. Operation timing information that represents the number of stall stages that occur due to the relationship between the preceding instruction and the following instruction, by referring to the level simulator, the information output from this instruction level simulator, and the program counter/clock cycle number correspondence table. and a stage state reproduction means that reproduces the stage state at each clock cycle number by referring to the operation timing information generated by the operation timing information generation means and information from the instruction level simulator. and an output means for outputting the stage state reproduced by the stage state reproduction means.

205 【課題を解決するための手段】 上記目的を達成するために、第1の発明であるシミュレーショ

ン装置は、動作タイミング設定用のプログラムとシミュレーションしようとするプログラムを入力する入力手段と、この入力手段により入力された動作タイミング設定用のプログラムを実行して、クロックサイクル数とパイプラインの各ステージのプログラムカウンタとの関係を表したプログラムカウンタ・クロックサイクル数対応表を生成するプログラムカウンタ・クロックサイクル数対応表生成手段と、このプログラムカウンタ・クロックサイクル数対応表生成手段によって生成されたプログラムカウンタ・クロックサイクル数対応表を参照して、動作タイミング設定用のプログラムとシミュレーションしようとするプログラムとを実行する命令レベルシミュレータと、この命令レベルシミュレータから出力される情報とプログラムカウンタ・クロックサイクル数対応表とを参照して、先行する命令と後続する命令との関係により発生するストールの段数を表した動作タイミング情報を生成する動作タイミング情報生成手段と、この動作タイミング情報生成手段によって生成された動作タイミング情報と命令レベルシミュレータからの情報とを参照して、各クロックサイクル数におけるステージ状態を再現するステージ状態再現手段と、このステージ状態再現手段によって再現されたステージ状態を出力する出力手段とを具備することを特徴とする。

[0013]

225 According to this first invention, it is possible to automate the development of a simulation program in an instruction set simulator, which conventionally took a lot of time and effort, especially the simulation of operation timing, so the time required for simulation can be automated. It can be significantly shortened.

230 この第1の発明によれば、従来多くの時間と手間のかかっていた命令セット・シミュレータにおけるシミュレーションするプログラムの開発、特に動作タイミングのシミュレーションを自動化することができるので、シミュレーションするのに要する時間を大幅に短縮することができる。

236 Furthermore, in the past, it was impossible to simulate the operation timing of a processor to be simulated or a processor for which only an HDL description existed and specifications regarding the operation timing were unknown; however, the simulation apparatus of the present invention According to this, it is possible.

241 さらに、従来はシミュレーションの対象となるプロセッサやHDL記述のみが存在し、動作タイミングに関する仕様が不明であるようなプロセッサに対しては動作タイミングのシミュレーションが不可能であったが、本発明のシミュレーション装置によれば可能になる。

[0014]

249 In the simulation device according to the second aspect of the invention, when the instruction level simulator does not have a cache table, the operation timing information includes information on the instruction cache miss rate and information on the data cache miss rate. The feature is that an expected latency value is generated by multiplying the cache miss rate or the data cache miss rate by the memory latency, and this expected latency value is added to the number of clock cycles.

256 第2の発明であるシミュレーション装置では、命令レベルシミュレータがキャッシュテーブルを有していない場合に、動作タイミング情報に命令キャッシュのミス率の情報とデータキャッシュのミス率の情報とを含めて、この命令キャッシュのミス率またはデータキャッシュのミス率とメモリのレイテンシとを掛け合わせてレイテンシ期待値を生成し、このレイテンシ期待値をクロックサイクル数に加算することを特徴とする。

[0015]

266 According to this second invention, even if the instruction level simulator does not have a cache table, by maintaining the cache miss rate, a real number can be used for the number of clock cycles and added in the form of an expected latency value. I can do it.

270 この第2の発明によれば、命令レベルシミュレータがキャッシュテーブルを有していない場合でも、キャッシュのミス率を保持することにより、クロックサイクル数に実数を用いてレイテンシ期待値というかたちで加算することができる。

275 This makes it possible to obtain an accurate number of clock cycles even when a cache table is not provided, speed up simulation processing, and reduce memory usage.

278 これによりキャッシュテーブルを有していない場合でも正確なクロックサイクル数を得ることができると共にシミュレーションの処理を高速化でき、メモリ使用量を軽減できる。

[0016]

285 The simulation device according to the third invention is characterized in that the program counter/clock cycle number correspondence table generation means adds a refill request from the data cache or the instruction cache to the program counter/clock cycle number correspondence table.

290 第3の発明であるシミュレーション装置では、プログラムカウンタ・クロックサイクル数対応表生成手段において、データキャッシュまたは命令キャッシュからのR e f i l l要求をプログラムカウンタ・クロックサイクル数対応表に追加することを特徴とする。

[0017]

298 According to this third invention, by detecting Refill requests from the data cache and instruction cache, the simulation device itself can recognize bus contention without preparing a special program. In addition to saving time and effort, it becomes easy to identify the number of stall stages and the cause thereof.

303 この第3の発明によれば、データキャッシュと命令キャッシュとからのR e f i l l要求を検知することにより、特別なプログラムを用意することなく、バス競合についてシミュレーション装置自体が認識できるので、特別なプログラムを用意する手間を省くことができると共に、ストール段数およびその原因の特定が容易になる。

[0018]

312 The fourth aspect of the invention is a simulation device, characterized by comprising comparison means for comparing the number of clock cycles of the stage state outputted from the stage state reproduction means and the number of clock cycles in the program counter/clock cycle number correspondence table. do.

317 第4の発明であるシミュレーション装置では、ステージ状態再現手段から出力されるステージ状態のクロックサイクル数とプログラムカウンタ・クロックサイクル数対応表のクロックサイクル数とを比較する比較手段を具備することを特徴とする。

[0019]

325 According to the fourth invention, the comparison means compares the number of clock cycles of the PC/CC correspondence table generated by the PC/CC correspondence table generation means and the number of clock cycles of the stage state reproduced by the stage state reproduction means. By providing , a more accurate number of clock cycles can be obtained.

331 この第4の発明によれば、P C ・ C C対応表生成手段で生成されたP C ・ C C対応表のクロッ

クサイクル数とステージ状態再現手段で再現されたステージ状態のクロックサイクル数とを比較する比較手段を設けることによって、より正確なクロックサイクル数を得ることができる。

[0020]

340 DESCRIPTION OF THE PREFERRED EMBODIMENTS An embodiment of a simulation apparatus according to the present invention will be described below with reference to the drawings.

343 【発明の実施の形態】 以下、本発明に係るシミュレーション装置の一実施形態を図面に基づいて説明する。

[0021]

350 FIG. 1 is a block diagram showing the configuration of a simulation apparatus according to this embodiment.

353 図1は本実施形態のシミュレーション装置の構成を示すブロック図である。

[0022]

359 As shown in FIG. 1, the simulation apparatus of this embodiment includes an input means 100 for inputting a program for setting operation timing, a program to be simulated, and architecture information necessary for processing; A PC/CC correspondence table generation means 210 generates a PC (program counter)/CC (clock cycle number) correspondence table according to a program for setting the input operation timing, and a PC/CC correspondence table generation means 210 generates the correspondence table. An instruction level simulator 220 executes the program input by the input means 100 using the PC/CC correspondence table, and obtains operation timing information from the information output by the instruction level simulator 220 and the PC/CC correspondence table. and a stage state that reproduces the stage state of the pipeline at each clock from the operation timing information generated by the operation timing information generation means 230 and the information output by the instruction level simulator 220. It consists of a reproduction means 240 and an output means 300 that outputs the stage state reproduced by the stage state reproduction means 240.

374 図1に示すように、本実施形態のシミュレーション装置は、動作タイミングを設定するためのプログラムやシミュレーションしようとするプログラム、さらに処理に必要なアーキテクチャ情報を入力する入力手段100と、この入力手段100により入力された動作タイミングを設定するためのプログラムによってPC（プログラムカウンタ）・CC（クロックサイクル数）対応表を生成するPC・CC対応表生成手段210と、このPC・CC対応表生成手段210により生成されたPC・CC対応表を利用して、入力手段100によって入力されたプログラムを実行する命令レベルシミュレータ220と、この命令レベルシミュレータ220により出力された情報とPC・CC対応表とから動作タイミング情報を生成する動作タイミング情報生成手段230と、この動作タイミング情報生成手段230によって生成された動作タイミング情報と命令レベルシミュレータ220によって出力された情報とから各クロックにおけるパイプラインのステージ状態を再現するステージ状態再現手段240と、このステージ状態再現手段240によって再現されたステージ状態を出力する出力手段300とから構成されている。

[0023]

392 The instruction level simulator 220 also has a cache table 225.

394 また、命令レベルシミュレータ220はキャッシュテーブル225を有している。

[0024]

400 Note that the input means 100 includes a keyboard, a mouse, a light pen, a flexible disk device, etc., and the simulation means 200 is an ordinary type having a CPU for performing various processes and a storage means for storing instructions for this processing. The device includes a computer system, and the output means 300 includes a display device, a printer device, and the like.

406 なお、入力手段100は、キーボード、マウス、ライトペン、又はフレキシブルディスク装置等が含まれ、シミュレーション手段200は各種の処理を行うためのCPUとこの処理の命令を記憶する記憶手段とを有する通常のコンピュータシステムを含む装置であり、また、出力手段300は、ディスプレイ装置やプリンタ装置等が含まれる。

412 Instructions and timing constraints for each process of the PC/CC correspondence table generation means 210, instruction level simulator 220, operation timing information generation means 230, and stage state reproduction means 240 included in the simulation means 200 are held in a storage means, It is loaded into the CPU and executed as necessary.

417 上記シミュレーション手段200に含まれるPC・CC対応表生成手段210、命令レベルシミュレータ220、動作タイミング情報生成手段230、ステージ状態再現手段240の各処理の命令やタイミング制約は記憶手段に保持されており、必要に応じてCPUにロードされ、実行がなされる。

423 Normally, execution is controlled by an operator inputting a command to the input means 100.

426 通常の場合、実行の制御は、操作者が入力手段100に対して命令（コマンド）の入力を行うことにより行われる。

[0025]

433 Next, the operation of the simulation apparatus of this embodiment will be explained.

435 次に、本実施形態のシミュレーション装置の動作について説明する。

438 FIG. 2 is a flowchart showing the operation of the simulation apparatus of this embodiment.

440 図2は本実施形態のシミュレーション装置の動作を示したフローチャートである。

[0026]

446 First, a program P1 for setting operation timing and architecture information are input from the input means 100 (S01).

449 まず、動作タイミング設定用のプログラムP1とアーキテクチャ情報を入力手段100から入力する（S01）。

453 The architecture information includes information such as the number of pipeline stages of the processor, information on input or output operands for each instruction, and cache size and method.

457 アーキテクチャ情報にはプロセッサのパイプライン段数、各命令についての入力オペランドあ

るいは出力オペランドの情報、キャッシュ・サイズや方式などの情報が含まれている。

[0027]

464 The program P1 for setting operation timing is a program that includes various stall causes, such as cache misses and data hazards, and is input to obtain information on how many stages of stalls occur due to each stall cause. It is something that will be done.

468 また、動作タイミング設定用のプログラム P 1 とは、さまざまなストール原因、例えばキャッシュミスやデータハザードを含んだプログラムであって、各ストール原因によって何段のストールが発生するかという情報を得るために入力されるものである。

[0028]

476 Here, an example of the program P1 for setting the operation timing is shown in FIG.

478 ここで、この動作タイミング設定用のプログラム P 1 の例を図 3 に示す。

481 Program A in Figure 3 is an example of a program with no cache misses and a data hazard, program B is an example of a program with a data cache miss and a data hazard, and program C is an example of a program with a data cache miss. This is an example of a program where there is an instruction cache miss.

486 図 3 のプログラム A はキャッシュミスがなく、データハザードがある場合のプログラムの例であり、プログラム B はデータキャッシュミスがあり、データハザードもある場合のプログラムの例であり、プログラム C はデータキャッシュミスがあり、命令キャッシュミスもある場合のプログラムの例である。

[0029]

495 In this way, when the program P1 for setting operation timing including various stall causes is input, it is sent to the PC/CC correspondence table generation means 210 and the instruction level simulator 220.

499 このように、さまざまなストール原因を含んだ動作タイミング設定用のプログラム P 1 は入力

されると、PC・CC対応表生成手段210と命令レベルシミュレータ220とに送られる

。

[0030]

507 The PC/CC correspondence table generation means 210 to which the program P1 is sent generates a PC/CC correspondence table corresponding to the stall causes included in each program.

511 プログラムP1の送られたPC・CC対応表生成手段210では、それぞれのプログラムが含んでいるストール原因に対応したPC・CC対応表を生成する。

[0031]

518 This PC/CC correspondence table generation means 210 is configured by a processor to be simulated, an application system including the processor, or an HDL simulator, so that when the program P1 for setting operation timing is inputted, it is possible to obtain information about how stalls caused by each stall cause affect each stage of the pipeline, and create a PC/CC correspondence table based on this information.

524 このPC・CC対応表生成手段210は、シミュレーションを行おうとしている対象のプロセッサやそのプロセッサを含む応用システム、あるいはHDLシミュレータによって構成されているので、動作タイミング設定用のプログラムP1が入力されると、各ストール原因によって発生するストールがパイプラインの各ステージにどのような影響を与えるかについて情報を得ることができ、この情報をもとにPC・CC対応表を作成する。

531 For example, when program A in Figure 3 is input, a PC/CC correspondence table is generated with no cache miss and data hazard, and similarly when program B is input, there is a data cache miss, a PC/CC correspondence table is generated when there is also a data hazard, and a PC/CC correspondence table is generated when there is a data cache miss and an instruction cache miss when program C is input (S02).

537 例えば、図3のプログラムAが入力された場合にはキャッシュミスがなく、データハザードがある場合のPC・CC対応表が生成され、同様にプログラムBが入力された場合にはデータキャッシュミスがあり、データハザードもある場合のPC・CC対応表が生成され、プログラムCが入力された場合にはデータキャッシュミスがあり、命令キャッシュミスもある場合のPC・CC対応表が生成される(S02)。

[0032]

547 Here, the PC/CC correspondence table will be explained based on the example of FIG. 4.

549 ここで、PC・CC対応表について図4の例をもとに説明する。

[0033]

555 The PC/CC correspondence table in FIG. 4(a) shows the stage state of the pipeline at each point in time of the number of clock cycles.

558 図4（a）のPC・CC対応表はクロックサイクル数の各時点のパイプラインのステージ状態を示している。

562 Each stage of the pipeline is F for fetch, D for decode, E for execute, M for memory access, and W for write back.

565 パイプラインの各ステージは、Fがフェッチ、Dがデコード、Eがエクゼキュート、Mはメモリアクセス、Wはライトバックを示している。

569 FIG. 4(a) is a PC/CC correspondence table for the program in FIG. 4(b), and the program counter value of each instruction is the value shown in FIG. 4(b). - For simplicity, the CC correspondence table shows only the last three digits of the program counter value in FIG. 4(b).

574 図4（a）は図4（b）のプログラムについてのPC・CC対応表であり、各命令のプログラムカウンタの値は図4（b）に示す値であるが、図4（a）のPC・CC対応表では簡単のため図4（b）のプログラムカウンタの値の下3桁のみで表している。

[0034]

582 The PC/CC correspondence table in FIG. 4(a) shows that when the number of clock cycles is 101, the 13C instruction, that is, the addi instruction, cannot perform the E stage and a one-

stage stall occurs. I understand.

586 図4 (a) のPC・CC対応表では、クロックサイクル数が101の時点で13Cの命令、即ちaddi命令がEの段を行うことができずに1段のストールが発生していることが分かる。

[0035]

594 Apart from the PC/CC correspondence table generation means 210 that generates such a PC/CC correspondence table, the operation timing setting program P1 and architecture information inputted from the input means 100 in step S01 are sent to the instruction level simulator 220. will also be sent.

599 このようなPC・CC対応表を生成するPC・CC対応表生成手段210とは別に、ステップS01で入力手段100から入力された動作タイミング設定用のプログラムP1とアーキテクチャ情報とは命令レベルシミュレータ220にも送られる。

604 The instruction level simulator 220 reads one instruction of the sent operation timing setting program P1, obtains the program counter of the PC/CC correspondence table, decodes the instruction, and sends the information to the operation timing information generation means 230. (S03).

609 命令レベルシミュレータ220では送られてきた動作タイミング設定用のプログラムP1の1命令を読み込むとともに、PC・CC対応表のプログラムカウンタを入手して当該命令をデコードし、その情報を動作タイミング情報生成手段230に送る(S03)。

[0036]

617 The operation timing information generating means 230 to which this information has been sent reads the corresponding line of the PC/CC correspondence table and obtains the number of clock cycles at the time of the relevant instruction (S04).

621 この情報の送られた動作タイミング情報生成手段230ではPC・CC対応表の該当する1行を読み込み、当該命令時点におけるクロックサイクル数を得る(S04)。

625 Then, operation timing information at the time of the instruction is created and updated from

information such as the instruction, the number of clock cycles at the time of the instruction, and the previous instructions and the number of clock cycles (S05).

629 そして、当該命令と当該命令時点におけるクロックサイクル数、それ以前の命令とそのクロックサイクル数などの情報から、当該命令時点における動作タイミング情報を作成、更新する（S05）。

634 The processing from step S03 to step S05 is repeated for all instructions of the program for setting operation timing (S06), and when all instructions are completed, operation timing information is generated (S07).

638 上記ステップS03からステップS05までの処理を動作タイミング設定用のプログラムの全ての命令について繰り返し行い（S06）、全ての命令について終了すると動作タイミング情報が生成される（S07）。

[0037]

646 Here, the operation timing information will be explained based on the example of FIG. 5.

648 ここで、動作タイミング情報について図5の例をもとに説明する。

651 The operation timing information represents the relationship between the preceding instruction and the following instruction, and in Figure 5, the preceding instruction is shown vertically and the following instruction is shown horizontally, and the numbers in the figure are stalled. Represents the minimum number of stages.

656 動作タイミング情報は先行する命令と後続する命令の関係を表したものであり、図5では縦には先行する命令が、横には後続する命令が表されており、図中の数字はストールする段数の最小値を表している。

661 Additionally, if there is a cache miss or bus contention, a list is added.

663 また、キャッシュミスやバス競合があった場合にはリストを付加していく。

666 FIG. 5 shows that when the preceding instruction is a mult instruction and the following instruction is also a mult instruction, a stall with a stage number of "1" occurs.

669 図5では先行する命令が`mult`命令で後続する命令も`mult`命令である場合には段数が「1」のストールが発生することを示している。

673 Further, when the preceding instruction is an `lw` instruction and the following instruction is a `mult` instruction, a cache miss occurs and a two-stage stall occurs.

676 また、先行する命令が`lw`命令で後続する命令が`mult`命令である場合にはキャッシュミスが発生し、2段のストールが発生することを示している。

[0038]

683 Next, a description will be given of how this operation timing information is created using the PC/CC correspondence table and information from the instruction level simulator.

686 次に、この動作タイミング情報がPC・CC対応表と命令レベルシミュレータからの情報とによって、どのように作成されるかについて説明する。

[0039]

693 Taking the PC/CC correspondence table of FIG. 4(a) and the program of FIG. 4(b) as examples, first referring to the program of FIG. 4(b), the `lw` instruction and the `mult` instruction are executed in the order of Referring to the PC/CC correspondence table in 4(a), there is a particular stall between "134", the last three digits of the program counter of the `lw` instruction, and "138", the last three digits of the `mult` instruction. I can see that there isn't.

699 図4（a）のPC・CC対応表と図4（b）のプログラムを例にすると、まず図4（b）のプログラムを参照すると、`lw`命令、`mult`命令の順で実行されており、図4（a）のPC・CC対応表を参照すると`lw`命令のプログラムカウンタの下3桁である「134」と`mult`命令の下3桁である「138」との間にはとくにストールが発生していないことが分かる。

[0040]

709 Therefore, the operation timing information generating means 230 writes "0" as the number

of stall stages in the column where the preceding instruction is the lw instruction and the following instruction is the mult instruction, as shown in the operation timing information of FIG.

714 そこで、動作タイミング情報生成手段230では、図5の動作タイミング情報に示すように、先行する命令がlw命令、後続する命令がmult命令となる欄にストール段数として「0」を書き込むことになる。

[0041]

722 Next, when referring to the program in FIG. 4(b), the addi instruction is executed next to the mult instruction, and when referring to the PC/CC correspondence table in FIG. 4(a), the number under the program counter of the mult instruction is It can be seen that one stage of stall occurs in the row where the number of clock cycles is "101" between the digit "138" and the last three digits of the addi instruction "13C".

728 次に、図4（b）のプログラムを参照すると、mult命令の次にはaddi命令が実行されており、図4（a）のPC・CC対応表を参照するとmult命令のプログラムカウンタの下3桁である「138」とaddi命令の下3桁である「13C」との間にはクロックサイクル数が「101」の行において、1段のストールが発生していることが分かる。

[0042]

737 Therefore, the operation timing information generating means 230 writes "1" as the stall stage number in the column where the preceding instruction is the mult instruction and the following instruction is the addi instruction, as shown in the operation timing information of FIG.

742 そこで、動作タイミング情報生成手段230では、図5の動作タイミング情報に示すように、先行する命令がmult命令、後続する命令がaddi命令となる欄にストール段数として「1」を書き込む。

747 However, the number of stall stages written here represents the minimum value, and if there is another cause of stall, the number of stall stages will increase.

750 ただし、ここで書き込んだストールの段数は最小値を表すものであり、その他にストールの原因が存在した場合にはストールする段数は増えていく。

[0043]

757 Further, when receiving information that a cache miss has occurred from the cache table 225 of the instruction level simulator 220, the operation timing information generation means 230 adds the information as a list to the operation timing information, as shown in FIG. 5.

761 また、命令レベルシミュレータ 220 のキャッシュテーブル 225 からキャッシュミスが発生したという情報を受けた場合には、動作タイミング情報生成手段 230 では、図 5 に示すように、その情報をリストとして動作タイミング情報に付加する。

766 In this way, the operation timing information generation means 230 creates operation timing information as shown in FIG. 5.

769 このようにして、動作タイミング情報生成手段 230 では図 5 のような動作タイミング情報が作成される。

[0044]

776 As described above, by entering the presence or absence of a stall and the number of stall stages in the operation timing information, it becomes possible to automatically simulate a stall due to a data hazard.

780 以上説明したように、動作タイミング情報にストールの有無、ストール段数を記入することによって、データハザードによるストールを自動的にシミュレーションすることができるようになる。

[0045]

788 In addition, as shown in FIG. 6, the operation timing information generation means 230 generates program counters 231 (F stage program counter, D stage program counter, E stage program counter) of each stage that operate by obtaining information from the PC/CC correspondence table, an M stage program counter, and a W stage program counter), and an instruction buffer 232.

794 また、動作タイミング情報生成手段230は図6に示すように、PC・CC対応表からの情報を得て動作する各ステージのプログラムカウンタ231（Fステージプログラムカウンタ、Dステージプログラムカウンタ、Eステージプログラムカウンタ、Mステージプログラムカウンタ、Wステージプログラムカウンタ）と、命令バッファ232とを具備している。

800 The instruction buffer 232 obtains and stores instructions from the instruction level simulator 220, and in particular, when receiving information that a branch has occurred (whether the address has jumped or not), it can record the instructions as shown in the figure. can.

804 そして、命令バッファ232では命令レベルシミュレータ220からの命令を入手して蓄えておき、とくに分岐が発生したという情報（アドレスがジャンプしたか否か）を受けたときには図に示すように記録することができる。

809 By this recording, when a stall occurs in a subsequent instruction, it is possible to identify whether the cause is a branch hazard or not.

812 この記録により後続する命令にストールが発生したときに、その原因が分岐ハザードであるか否かを特定することができる。

[0046]

819 Next, after executing steps up to step S07 as described above and generating operation timing information, the program P2 to be simulated is input from the input means 100 (S08).

822 次に、以上説明したようなステップS07までを実行して動作タイミング情報が生成されたところでシミュレーションしようとするプログラムP2を入力手段100から入力する（S08）。

827 The input program P2 is sent to the instruction level simulator 220, and the instruction level simulator 220 reads one instruction of the sent program P2 to be simulated, obtains the program counter of the PC/CC correspondence table, and executes the instruction. is decoded (S09) and executed (S10).

832 入力されたプログラムP2は命令レベルシミュレータ220に送られ、命令レベルシミュレータ220では送られてきたシミュレーションしようとするプログラムP2の1命令を読み込むとともに、PC・CC対応表のプログラムカウンタを入手して当該命令をデコードして（S09）実行する（S10）。

838 Then, when information that the instruction has been executed by the instruction level simulator 220 is sent to the stage state reproduction means 240, the stage state reproduction means 240 receives the operation timing information and calculates the instruction and the number of clock cycles at the time of the instruction. Using information such as the previous instruction and the number of clock cycles at the time of the instruction, the stage state at the time of the instruction is reproduced and output (S11).

845 そして、当該命令が命令レベルシミュレータ２２０で実行されたという情報がステージ状態再現手段２４０に送られると、ステージ状態再現手段２４０では動作タイミング情報を受けて、当該命令と当該命令時点のクロックサイクル数、それ以前の命令とその命令時点のクロックサイクル数等の情報により、当該命令時点のステージ状態を再現して出力する（Ｓ１１）。

[0047]

855 The processes from step S09 to step S11 are repeated for all instructions of the program P2 to be simulated (S12), and upon completion, the stage state is reproduced (S13), and this stage state is outputted from the output means 300 as a processing result. Then (S14) the simulation regarding the operation timing ends.

860 シミュレーションしようとするプログラムＰ２の全ての命令について、ステップＳ０９からステップＳ１１までの処理を繰り返し行い（Ｓ１２）、終了するとステージ状態が再現され（Ｓ１３）、このステージ状態を出力手段３００から処理結果として出力すると（Ｓ１４）動作タイミングについてのシミュレーションが終了することになる。

[0048]

869 Here, the output stage status is a reproduction of all the statuses of each stage of the pipeline, as shown in the PC/CC correspondence table shown in Figure 4, and the corresponding number of clock cycles at that time. , for simple representation, only the number of clock cycles may be output.

874 ここで、出力されるステージ状態は、図４に示したＰＣ・ＣＣ対応表のようにパイプラインの各ステージの状態を全て再現して、そのときのクロックサイクル数を対応させたものであるが、簡単に表す場合にはクロックサイクル数のみを出力してもよい。

879 When outputting only this number of clock cycles, the stage state reproduction means 240
can have a simple configuration such as a cycle number counter.

882 このクロックサイクル数のみを出力する場合には、ステージ状態再現手段240はサイクル数
カウンタのように簡単な構成にすることができる。

[0049]

889 Next, a case where the instruction level simulator 220 does not have the cache table 225 will
be described.

892 次に、命令レベルシミュレータ220がキャッシュテーブル225を有していない場合につい
て説明する。

[0050]

899 In the simulation apparatus of FIG. 1, since the instruction level simulator 220 has the cache
table 225, the operation timing information generation means 230 receives accurate
information from the instruction level simulator 220 regarding information on whether a
cache miss has occurred. I was able to get

904 図1のシミュレーション装置では、命令レベルシミュレータ220がキャッシュ・テーブル
225を有していたので、キャッシュミスが発生したか否かの情報について、動作タイミン
グ情報生成手段230は命令レベルシミュレータ220から正確な情報を得ることができた
。

[0051]

913 However, if you want to shorten the simulation time, if the instruction level simulator 220 has
a cache table 225, the simulation speed may be slowed down because it takes time to refer to
the cache table for each instruction. there were.

917 しかし、シミュレーションする時間を短くしたい場合には、命令レベルシミュレータ220が
キャッシュ・テーブル225を有していると、各命令毎にキャッシュテーブルを参照する時
間がかかるためにシミュレーションするスピードが遅くなる場合があった。

922 Therefore, if it is desired to shorten the simulation time, it may be better for the instruction level simulator 220 not to have the cache table 225.

925 従って、シミュレーションする時間を短くしたい場合には命令レベルシミュレータ 220 がキャッシュテーブル 225 を有しないほうがよい場合がある。

929 Furthermore, there are cases where the instruction level simulator does not originally have a cache table.

932 また、もともと命令レベルシミュレータがキャッシュテーブルを有していない場合もある。

[0052]

938 In such a case, it is not possible to obtain information on whether a cache miss has occurred from the instruction level simulator 220, so the instruction cache miss rate as shown in FIG. 7 and the instruction cache miss rate as shown in FIG. 8 are included in the operation timing information. Information about the data cache miss rate should be included.

943 このような場合には、命令レベルシミュレータ 220 からキャッシュミスが発生したか否かの情報を得ることができないので、動作タイミング情報に図 7 に示すような命令キャッシュのミス率と図 8 に示すようなデータキャッシュのミス率に関する情報を含める必要がある。

[0053]

951 The graph in FIG. 7 shows the relationship between the amount of address jump and the instruction cache miss rate when a branch instruction occurs.

954 図 7 のグラフは分岐命令が発生したときのアドレス・ジャンプの量と命令キャッシュのミス率との関係を示したグラフである。

958 Therefore, by having the operation timing information generation means 230 and the stage state reproduction means 240 hold the amount of address jump for the immediately preceding branch instruction, the miss rate of the instruction cache can be calculated from the amount of address jump using the graph of FIG. can be found.

963 従って、動作タイミング情報生成手段 230 とステージ状態再現手段 240 が直前の分岐命令
に対するアドレスジャンプの量を保持しておくことによって、図 7 のグラフを利用してアド
レスジャンプの量から命令キャッシュのミス率を求めることができる。

[0054]

971 Further, the graph in FIG. 8 is a graph showing the relationship between the number of
accesses to the data cache (and access range) and the miss rate of the data cache.

974 また、図 8 のグラフはデータキャッシュへのアクセス数（及びアクセス範囲）とデータキャッ
シュのミス率との関係を示したグラフである。

978 Therefore, by holding the number of accesses and the access range by the operation timing
information generating means 230 and the stage state reproducing means 240, the data
cache miss rate can be determined from the number of accesses using the graph of FIG.

982 従って、動作タイミング情報生成手段 230 とステージ状態再現手段 240 がアクセス数及び
アクセス範囲を保持しておくことによって、図 8 のグラフを利用してアクセス数からデータ
キャッシュのミス率を求めることができる。

[0055]

990 Then, by multiplying the instruction cache and data cache miss rates by the memory latency,
the expected latency value can be generated since the cache miss rate is a probability (1 or
less).

994 そして、この命令キャッシュ、データキャッシュのミス率とメモリのレイテンシとを掛け合わ
せると、キャッシュのミス率は確率（1 以下）であるのでレイテンシ期待値を生成すること
ができる。

999 By adding this expected latency value to the number of clock cycles each time an instruction
is executed, operation timing can be simulated even when there is no information regarding
cache misses from the cache table.

1003 このレイテンシ期待値を命令の実行の度に、クロックサイクル数に加算していくことによっ

て、キャッシュテーブルからのキャッシュミスに関する情報がない場合でも動作タイミングのシミュレーションを行うことができるようになる。

[0056]

1011 Conventionally, the number of clock cycles was an integer (for example, 100, 101, etc. in FIG. 4) and was therefore added by an integer, but in this case, the influence of a cache miss would be ignored.

1015 従来、クロックサイクル数は、整数（例えば、図4では100、101など）だったので整数加算されていたものであるが、それではこの場合キャッシュミスの影響は無視されてしまう。

1020 Therefore, in order to avoid this problem, when there is no cache table, the number of clock cycles is expressed as a real number (for example, 100.8, etc.) so that the expected latency value can be added as a real number.

1024 そこで、この問題を回避するためにキャッシュテーブルがない場合には、クロックサイクル数を実数値（例えば、100.8など）で表してレイテンシ期待値を実数で加算できるようにする。

1029 In this way, by using a real number for the number of clock cycles, the effect of a data cache miss is added in the form of an expected latency value each time data is accessed, and even when the instruction level simulator 220 does not have a cache table 225. It is possible to obtain an accurate number of clock cycles.

1034 このように、クロックサイクル数に実数を用いることによって、データへのアクセスの度にデータキャッシュのミスの影響がレイテンシ期待値というかたちで加算され、命令レベルシミュレータ220がキャッシュテーブル225を有しない場合であっても正確なクロックサイクル数を得ることができる。

[0057]

1043 Also, as shown in Figures 7 and 8, the cache miss rate is expressed as a function of specific parameters (address jump amount, number of accesses, access range, etc.), but the cache miss rate is simply expressed as the average. It may also be expressed as a value (fixed

value).

1048 また、ここでは図7や図8のように、特定のパラメタ（アドレスジャンプの量、アクセス数、アクセス範囲等）の関数としてキャッシュのミス率を表したが、キャッシュのミス率を簡単にその平均値（一定値）で表してもよい。

[0058]

1056 Next, a case will be described in which the simulation device itself can detect whether or not bus contention is occurring.

1059 次に、バス競合が生じているか否かをシミュレーション装置自体が検知することができる場合について説明する。

[0059]

1066 In the simulation device shown in Figure 1, in order to obtain operation timing information when there is bus contention, a special program as shown in Figure 3 must be prepared to create a PC/CC correspondence table when there is bus contention. did not become.

1070 図1のシミュレーション装置では、バス競合がある場合の動作タイミング情報を得るためには図3に示すような特別なプログラムを用意してバス競合がある場合のPC・CC対応表を作成しなければならなかった。

1075 However, if the simulation device itself can detect whether or not bus contention is occurring, this effort can be omitted.

1078 しかし、バス競合が生じているか否かをシミュレーション装置自体が検知することができるようにすれば、その手間を省略することができる。

[0060]

1085 Therefore, as shown in FIG. 9, the refill request signal from the data cache and instruction cache is configured so that it can be detected and used by the PC/CC correspondence table generation means 210 via a BIU (Bus Interface Unit).

1089 そこで、図 9 に示すようにデータ・キャッシュと命令キャッシュからのrefill要求信号を
BIU (Bus Interface Unit) を介してPC・CC対応表生成手段210で検知して利用できる
ように構成する。

1094 Since the PC/CC correspondence table generation means 210 is constituted by a processor
to be simulated, an application system including the processor, or an HDL simulator, it
includes a data cache and an instruction cache.

1098 PC・CC対応表生成手段210はシミュレーションを行おうとしている対象のプロセッサ
やそのプロセッサを含む応用システム、あるいはHDLシミュレータによって構成されてい
るので、データキャッシュや命令キャッシュを含んでいる。

1103 Therefore, the configuration is such that instructions can be directly detected from the data
cache and instruction cache.

1106 そこで、このデータキャッシュと命令キャッシュから直接命令を検知できるように構成した
ものである。

[0061]

1113 Then, the PC/CC correspondence table generation means 210 adds the value of the detected
refill request signal to the PC/CC correspondence table and sends it to the operation timing
information generation means 230. Based on the value of the refill request signal, it is
possible to correctly recognize the presence or absence of bus contention and generate
operation timing information.

1119 そして、PC・CC対応表生成手段210では、検知したrefill要求信号の値をPC・CC対
応表に追加して動作タイミング情報生成手段230に送るので、動作タイミング情報生成手
段230では、送られてきたrefill要求信号の値によってバス競合の有無を正しく認識して動
作タイミング情報を生成することができる。

1125 Furthermore, the stage state reproducing means 240 which has received the operation
timing information can also output a stage state in which the influence of bus contention is
taken into consideration.

1129 さらに、その動作タイミング情報を受け取ったステージ状態再現手段240でもバス競合の

影響について考慮されたステージ状態を出力することができる。

[0062]

1136 In this way, by enabling the simulation apparatus itself to directly detect the presence or absence of bus contention in the PC/CC correspondence table generation means 210, the effort of creating a special program can be omitted.

1140 このように、シミュレーション装置自体がP C・C C対応表生成手段2 1 0においてバス競合の有無を直接検知できるようにしたことにより、特別なプログラムを作る手間を省略できるようになった。

[0063]

1148 Next, a simulation device that can check whether the stage state output as a processing result is correct will be described.

1151 次に、処理結果として出力されたステージ状態が正しいか否かを確認することのできるシミュレーション装置について説明する。

[0064]

1158 FIG. 10 is a block diagram showing the configuration of a simulation device that can confirm whether the stage state is correct.

1161 図1 0はステージ状態が正しいか否かを確認することのできるシミュレーション装置の構成を示すブロック図である。

[0065]

1168 In the simulation apparatus of FIG. 10, a comparison means compares the number of clock cycles of the PC/CC correspondence table generated by the PC/CC correspondence table generation means 210 and the number of clock cycles of the stage state reproduced by the stage state reproduction means 240. 250.

1173 図10のシミュレーション装置では、PC・CC対応表生成手段210で生成されたPC・CC対応表のクロックサイクル数とステージ状態再現手段240で再現されたステージ状態のクロックサイクル数とを比較する比較手段250をさらに具備している。

[0066]

1181 Since the PC/CC correspondence table generation means 210 is configured by a processor to be simulated, an application system including the processor, or an HDL simulator, when the program P2 to be simulated is inputted, It can be expected to obtain an accurate number of clock cycles.

1186 PC・CC対応表生成手段210は、シミュレーションを行おうとしている対象のプロセッサやそのプロセッサを含む応用システム、あるいはHDLシミュレータによって構成されているので、シミュレーションをしようとするプログラムP2を入力した場合には正確なクロックサイクル数を得ることが期待できる。

1192 Therefore, by comparing the number of clock cycles in the PC/CC correspondence table and the number of clock cycles in the stage state generated by the stage state reproduction means 240, it is possible to check whether the number of clock cycles in the stage state is correct. can.

1197 従って、PC・CC対応表のクロックサイクル数とステージ状態再現手段240で生成されたステージ状態のクロックサイクル数とを比較することによって、ステージ状態のクロックサイクル数が正しいか否かを確認することができる。

[0067]

1205 The simulation device in FIG. 10 performs the same processing as the simulation device in FIG. 1 as shown in the flowchart in FIG. 2, outputs the stage state, and then checks whether the output stage state is correct. Perform the processing shown below.

1209 この図10のシミュレーション装置は、図2のフローチャートに示すような図1のシミュレーション装置と同様の処理を行ってステージ状態を出力した後に、出力されたステージ状態が正しいか否かを確認する以下に示すような処理を行う。

[0068]

1217 First, when the program P2 to be simulated is input again from the input means 100, it is executed by the PC/CC correspondence table generation means 210 to generate a PC/CC correspondence table.

1221 まず、シミュレーションしようとするプログラム P 2 を入力手段 1 0 0 から再び入力すると、P C ・ C C 対応表生成手段 2 1 0 で実行され、P C ・ C C 対応表を生成する。

[0069]

1228 Next, when the program P2 is executed one instruction at a time by the instruction level simulator 220, the stage state reproduction means 240 receives the information sent from the instruction level simulator 220 and the already generated operation timing information, and receives the first instruction. The stage status is output in the same way as in the embodiment.

1234 次に、プログラム P 2 が命令レベルシミュレータ 2 2 0 で 1 命令ずつ実行されると、ステージ状態再現手段 2 4 0 では、命令レベルシミュレータ 2 2 0 から送られる情報とすでに生成されている動作タイミング情報とを受けて、第 1 の実施形態と同様にステージ状態を出力する。

[0070]

1243 The stage state output by the stage state reproduction means 240 and the PC/CC correspondence table generated by the PC/CC correspondence table generation means 210 are sent to the comparison means 250, and the stage state clock cycle number and the PC/CC correspondence table are sent to the comparison means 250. Compare with the number of clock cycles in the correspondence table.

1249 このステージ状態再現手段 2 4 0 で出力されたステージ状態と P C ・ C C 対応表生成手段 2 1 0 で生成された P C ・ C C 対応表とが比較手段 2 5 0 に送られ、ステージ状態のクロック・サイクル数と P C ・ C C 対応表のクロックサイクル数とを比較する。

1254 As a result of this comparison, if the respective numbers of clock cycles are different, the immediately preceding instruction sequence, that is, the contents of the instruction buffer

of the stage state reproduction means 240 is output as a log file.

1258 この比較した結果、それぞれのクロックサイクル数が違っていた場合にはその直前の命令列、すなわちステージ状態再現手段240の命令バッファの内容をログ・ファイルとして出力する。

1263 The developer of the simulator analyzes the contents, creates an additional program for correcting the error in the number of clock cycles, and inputs it from the input means 100.

1266 シミュレータの開発者は、その内容を解析しクロック・サイクル数の誤差を修正するような追加プログラムを作成し、入力手段100から入力する。

1270 This additional program is called (dynamically linked) and used by the stage state reproduction means 240.

1273 この追加プログラムは、ステージ状態再現手段240により呼び出されて（ダイナミック・リンクがなされ）利用される。

[0071]

1280 Furthermore, if they do not match, the value of the clock cycle number in the stage state is replaced with the clock cycle number in the PC/CC correspondence table.

1283 また、さらに一致しない場合には、ステージ状態のクロック・サイクル数の値をPC・CC対応表のクロックサイクル数で置き換える。

1287 This replacement makes it possible to prevent future comparisons from unnecessarily failing, thereby reducing the analysis work of the person performing the simulation.

1290 この置き換えにより、以後の比較がいたずらに失敗することを防ぐことができるので、シミュレーションをする者の解析作業が軽減されることになる。

[0072]

1297 [Effects of the Invention] As explained above, according to the simulation device of the

present invention, it is possible to automate the development of a simulation program in an instruction set simulator, which conventionally took a lot of time and effort, especially the simulation of operation timing. Therefore, the time required for simulation can be significantly reduced.

1303 【発明の効果】以上説明してきたように、本発明のシミュレーション装置によれば、従来多くの時間と手間のかかっていた命令セット・シミュレータにおけるシミュレーションするプログラムの開発、特に動作タイミングのシミュレーションを自動化することができるので、シミュレーションするのに要する時間を大幅に短縮することができる。

1309 Furthermore, in the past, it was impossible to simulate the operation timing of a processor to be simulated or a processor for which only an HDL description existed and specifications regarding the operation timing were unknown; however, the simulation apparatus of the present invention According to this, it is possible.

1314 さらに、従来はシミュレーションの対象となるプロセッサやHDL記述のみが存在し、動作タイミングに関する仕様が不明であるようなプロセッサに対しては動作タイミングのシミュレーションが不可能であったが、本発明のシミュレーション装置によれば可能になる。

[0073]

1322 Furthermore, even if the instruction level simulator does not have a cache table, by maintaining the cache miss rate, a real number can be used for the number of clock cycles and added in the form of an expected latency value.

1326 また、命令レベルシミュレータがキャッシュテーブルを有していない場合でも、キャッシュのミス率を保持することにより、クロックサイクル数に実数を用いてレイテンシ期待値というかたちで加算することができる。

1331 This makes it possible to obtain an accurate number of clock cycles even when a cache table is not provided, speed up simulation processing, and reduce memory usage.

1334 これによりキャッシュテーブルを有していない場合でも正確なクロックサイクル数を得ることができると共にシミュレーションの処理を高速化でき、メモリ使用量を軽減できる。

[0074]

1341 In addition, by detecting Refill request signals from the data cache and instruction cache, the simulation device itself can recognize bus contention without preparing a special program, thereby saving the effort of preparing a special program. At the same time, it becomes easy to identify the number of stall stages and the cause thereof.

1346 また、データキャッシュと命令キャッシュとからの R e f i l l 要求信号を検知することにより、特別なプログラムを用意することなく、バス競合についてシミュレーション装置自体が認識できるので、特別なプログラムを用意する手間を省くことができると共に、ストール段数およびその原因の特定が容易になる。

[0075]

1355 Furthermore, by providing a comparison means for comparing the number of clock cycles in the PC/CC correspondence table generated by the PC/CC correspondence table generation means and the number of clock cycles in the stage state reproduced by the stage state reproduction means, more accurate The number of clock cycles can be obtained.

1360 また、P C ・ C C 対応表生成手段で生成された P C ・ C C 対応表のクロックサイクル数とステージ状態再現手段で再現されたステージ状態のクロックサイクル数とを比較する比較手段を設けることによって、より正確なクロックサイクル数を得ることができる。

[0076]

1368 Brief description of the drawing

1370 図面の簡単な説明

[0077]

1376 FIG. 1 is a block diagram showing the configuration of an embodiment of a simulation apparatus according to the present invention.

1379 図 1 本発明によるシミュレーション装置の一実施の形態の構成を示すブロック図である。

[0078]

1385 FIG. 2 is a flowchart showing the operation of the simulation apparatus according to the present invention.

1388 図2 本発明によるシミュレーション装置の動作を示すフローチャートである。

[0079]

1394 FIG. 3 is a diagram showing an example of a special program for setting operation timing.

1396 図3 動作タイミグを設定するための特別なプログラムの例を示す図である。

[0080]

1402 FIG. 4 is a diagram showing an example of a PC/CC correspondence table.

1404 図4 P C ・ C C 対応表の例を示す図である。

[0081]

1410 FIG. 5 is a diagram showing an example of operation timing information.

1412 図5 動作タイミグ情報の例を示す図である。

[0082]

1418 FIG. 6 is a block diagram showing the configuration of the operation timing information generation means 210.

1421 図6 動作タイミグ情報生成手段210の構成を示すブロック図である。

[0083]

1427 FIG. 7 is a graph showing the relationship between the data cache miss rate and the amount of address jump.

1430 図7 データキャッシュのミス率とアドレスジャンプの量との関係を示したグラフである。

[0084]

1436 FIG. 8 is a graph showing the relationship between the instruction cache miss rate and the number of accesses.

1439 図8 命令キャッシュのミス率とアクセス数との関係を示したグラフである。

[0085]

1445 FIG. 9 is a block diagram showing the configuration inside the PC/CC correspondence table generation means 210 when a Refill request is detected.

1448 図9 R e f i l l 要求を検知する場合のP C ・ C C 対応表生成手段2 1 0 内の構成を示したブロック図である。

[0086]

1455 10 is a block diagram showing a case where the simulation apparatus of FIG. 1 is further provided with comparison means.

1458 図1 0 図1 のシミュレーション装置に比較手段をさらに具備した場合について示したブロック図である。

[0087]

1465 Explanation of symbols

[0088]

1473 100 Input means 200 Simulation means 210 PC/CC correspondence table generation means
220 Instruction level simulator 225 Cache table 230 Operation timing information
generation means 231 Program counter 232 Instruction buffer 240 Stage state
reproduction means 250 Comparison means 300 Output means

1478 1 0 0 入力手段 2 0 0 シミュレーション手段 2 1 0 PC・CC対応表生成手段 2 2 0 命令
レベルシミュレータ 2 2 5 キャッシュテーブル 2 3 0 動作タイミング情報生成手段 2 3 1
プログラムカウンタ 2 3 2 命令バッファ 2 4 0 ステージ状態再現手段 2 5 0 比較手段
3 0 0 出力手段