

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 JP2008204111A

10 Semiconductor integrated circuit design support device, design support method, manufacturing method, program, and recording medium

13 半導体集積回路の設計支援装置、その設計支援方法、その製造方法、プログラム、及び記録媒体

[0001]

20 The present invention relates to a semiconductor integrated circuit design support device, a design support method thereof, a manufacturing method thereof, a program, and a recording medium, and in particular, a technique suitably applied in semiconductor integrated circuit design performed using a behavioral synthesis tool. It is related to.

25 本発明は、半導体集積回路の設計支援装置、その設計支援方法、その製造方法、プログラム、及び記録媒体に関し、特に、動作合成ツールを用いて行われる半導体集積回路の設計において好適に適用される技術に関するものである。

[0002]

33 With advances in semiconductor manufacturing technology, the degree of integration of semiconductors is increasing at an extremely rapid rate, and as a result, the scale and complexity of LSIs (Large Scale Integrations) are increasing year by year. On the other hand, since the life cycle of electronic equipment products equipped with LSIs tends to become shorter, there is a desire to shorten the circuit design period. However, as LSIs become larger

and more complex, conventional design methods are considered to be reaching their limits, and in recent years, methods of describing designs at a higher level of abstraction have been attracting attention. Furthermore, various EDA (Electronic Design Automation) tools for this purpose have been proposed.

43 半導体製造技術の進歩により、半導体の集積度は非常に速い速度で高まってきており、これにより L S I (Large Scale Integration) は規模、複雑さが年々増している。一方で、L S I を搭載する電子機器製品のライフサイクルは短くなる傾向にあることから、回路設計期間の短縮化が望まれている。ところが、L S I が大規模化、複雑化するに従い、従来の設計手法は限界にきていると考えられており、近年では、より高い抽象度で設計を記述する手法に注目が集まっている。また、そのための E D A (Electronic Design Automation：電子設計自動化) ツールも各種提案されている。

[0003]

55 Behavioral synthesis tools are attracting attention as one of these EDA tools.

57 この E D A ツールの 1 つとして注目を浴びているのが動作合成ツールである。

60 A behavioral synthesis tool is a tool that generates an RTL (Register Transfer Level) description to be input to a logic synthesis tool from a behavioral description that expresses the behavior of hardware. A behavioral description is a direct description of the processing algorithm to be executed by the hardware at a behavioral level, and there is no concept of clock cycles. On the other hand, RTL descriptions are written with consideration of hardware-specific concepts such as registers and clock synchronization. A behavioral synthesis tool can increase the abstraction level of a design by automatically synthesizing an RTL description from a behavioral description.

69 動作合成ツールは、ハードウェアの動作を表現する動作記述から、論理合成ツールに入力する R T L (Register Transfer Level：レジスタ転送レベル) 記述を生成するツールである。動作記述は、ハードウェアに実行させる処理のアルゴリズムを直接的に動作レベルで記述したものでクロックサイクルの概念はない。他方、R T L 記述は、レジスタやクロック同期等のハードウェアに特有の概念を意識して記述したものである。動作合成ツールでは、動作記述から R T L 記述を自動的に合成することにより、設計の抽象度を上げることができる。

[0004]

80 The most important feature of behavioral description is that it does not explicitly describe what kind of operation (addition, multiplication, comparison, etc.) is to be performed in each clock cycle, but simply describes the operation.

84 動作記述の最大の特徴は、各クロックサイクルでどのような演算（加算、乗算、比較等）をするかについては明示的には記述せず、単に演算のみを記述するという点である。

88 It is said that behavioral level description has the following advantages over RTL description.
・ The amount of description is smaller than RTL description. - Simulation at the behavioral level is faster than RTL simulation.

92 R T L 記述に比べて動作レベル記述には以下の利点があるとされている。 ・ R T L 記述に比べて記述量が少なくて済む。 ・ 動作レベルでのシミュレーションの方が、R T L シミュレーションよりシミュレーション速度が速い。

[0005]

100 By the way, when an RTL description is generated using a behavioral synthesis tool, it is necessary to check the latency of the RTL description.

103 ところで、動作合成ツールを使って R T L 記述を生成した場合、その R T L 記述のレイテンシを調べる必要がある。

107 Latency is the total number of clock cycles required to complete all processing on the total input data. Typically, this latency depends on the type of data being input. For this reason, a logic simulation (hereinafter referred to as RTL simulation) for the RTL description is performed, and the latency is determined from the simulation result. Although there are cases where it is possible to determine the latency without performing this RTL simulation, there are many cases where this is not possible.

114 レイテンシとは、入力された総データに対して処理をすべて完了するのに必要な総クロックサイクル数のことである。通常、このレイテンシは入力するデータの種類の依存する。このため、R T L 記述に対する論理シミュレーション（以下、R T L シミュレーションという）を実施し、そのシミュレーション結果からレイテンシを求める。この R T L シミュレーションをせずにレイテンシを求めることができる場合もあるが、そうでない場合が多い。

[0006]

124 A case in which latency cannot be determined without performing an RTL simulation will be described using the example of FIG. 4.

127 R T L シミュレーションを実施しないとレイテンシがわからない場合について、図 4 の例を使って説明する。

131 FIG. 4 is an example of description at the behavioral level. When inputting this into a behavioral synthesis tool and generating an RTL description, in the RTL description, the total number of processing cycles of block A41 and the total number of processing cycles of block B42 are not known before RTL simulation (note that blocks are (the part that represents the operation of the unit)). This is because the number of times block A41 and block B42 are executed may differ depending on the input data to be processed.

138 図 4 は動作レベルの記述例である。これを動作合成ツールに入力して、R T L 記述を生成したとき、R T L 記述において、ブロック A 4 1 の総処理サイクル数とブロック B 4 2 の総処理サイクル数は R T L シミュレーション前には分からない（なお、ブロックは所定単位の動作を表す部分）。これは、ブロック A 4 1 やブロック B 4 2 の実行される回数が、処理すべき入力データによって異なる場合があるからである。

[0007]

148 For example, assume that processing block A41 requires two clock cycles, and processing block B42 requires three clock cycles.

151 例えば、ブロック A 4 1 の処理では 2 クロックサイクルが必要で、ブロック B 4 2 の処理では 3 クロックサイクルが必要であるとする。

155 An RTL description is generated by a behavioral synthesis tool, and as a result of inputting certain data to be processed and executing a logic simulation on this RTL description, block A41 is executed 1 million times and block B42 is executed 10,000 times. Suppose that As a result, the total latency in block A41 is 2×1 million times = 2 million clock cycles, and the total latency in block B42 is $3 \times 10,000$ times = 30,000 clock cycles. In other words, if it is desired to reduce the total latency for this data, it is necessary to take measures to reduce the processing cycles of block A41 instead of block B42. Furthermore, if RTL simulation is performed using different input data, there is a possibility that the total latency of block B42 will be greater than the total latency of block A41.

165 動作合成ツールによりRTL記述が生成され、このRTL記述に対して、ある処理すべきデータを入力して論理シミュレーションを実行した結果、ブロックA 4 1が100万回実行され、ブロックB 4 2は1万回実行されたとする。この結果、ブロックA 4 1における総レイテンシは、 $2 \times 100 \text{万回} = 200 \text{万クロックサイクル}$ であり、ブロックB 4 2における総レイテンシは、 $3 \times 1 \text{万回} = 3 \text{万クロックサイクル}$ となる。つまり、このデータに対して総レイテンシを減らしたい場合には、ブロックB 4 2ではなくブロックA 4 1の処理サイクルを減らすための対策を採ることが必要となる。また、別の入力データを使ってRTLシミュレーションした場合は、逆にブロックB 4 2の総レイテンシがブロックA 4 1の総レイテンシよりも多くなる可能性もある。

[0008]

179 Additionally, it seems that a method of determining worst-case latency without performing RTL simulation is being considered, but it has not been put into practical use, and latency measurement using RTL simulation is still the main method.

183 また、RTLシミュレーションを行わずに最悪ケースのレイテンシを求める手法も考えられているようだが、実用には至らず、依然RTLシミュレーションによるレイテンシ計測が主要となっている。

188 Furthermore, when latency is measured by RTL simulation, the intended function may not be completed with the desired latency. In this case, it is necessary to go back to the behavioral description and find out which part of the behavioral description is consuming a large number of clock cycles. However, currently it is a very troublesome task to find out which part of the behavioral description has a lot of latency.

194 さらに、RTLシミュレーションによるレイテンシを計測した場合、目的とする機能が所望のレイテンシで完了しない場合がある。この場合は、動作記述に立ち戻り、該動作記述においてどの部分で多数のクロックサイクルを消費しているかを調べる必要がある。しかし、動作記述のどの部分でレイテンシが多くなっているのかを調べることは、現状では非常に面倒な作業となっている。

[0009]

204 For example, Patent Document 1 discloses a method for specifying the correspondence between a path in an RTL description and a path in a behavioral description.

207 例えば特許文献 1 では、RTL 記述上のパスと動作記述上のパスとの対応関係を特定する方法が開示されている。

211 In this invention, in static timing analysis of a gate description (gate-level logic circuit) converted from an RTL description by logic synthesis, correction is efficiently performed to reduce the delay of a critical path (path with the longest delay in the circuit). That's what I'm aiming for. <patcit num="1"><text>特開 2 0 0 6 – 2 8 5 8 6 5 号公報</text></patcit>

216 当該発明では、論理合成により RTL 記述から変換されたゲート記述（ゲートレベル論理回路）での静的タイミング解析において、クリティカルパス（回路中最も遅延が大きいパス）の遅延を小さくする修正を効率良く行うことを狙いとしている。<patcit num="1"><text>特開 2 0 0 6 – 2 8 5 8 6 5 号公報</text></patcit>

[0010]

225 In view of the above-mentioned situation, the present invention makes it possible to efficiently verify the logical functions and timing of a designed logic circuit using RTL simulation in the design of a semiconductor integrated circuit, and also enables the upstream design process to be performed based on the verification results. The purpose is to enable corrections on the side.

231 上述した状況に鑑み、本発明は、半導体集積回路の設計において、設計した論理回路の論理機能やタイミングを RTL シミュレーションで効率的に検証できるようにするとともに、該検証結果に基づいて設計工程の上流側での修正を可能とすることを目的とする。

[0011]

239 In order to achieve such an objective, the present invention reads a behavioral description that describes the algorithm of processing performed by hardware at a behavioral level, and creates an RTL description that is written with consideration of hardware-specific concepts including registers and clock synchronization. The design support device for a semiconductor integrated circuit that generates a semiconductor integrated circuit is characterized by having a latency analysis means for analyzing the results of logic simulation for the RTL description and determining the latency in each block representing a predetermined unit of operation in the behavioral description. do.

248 かかる目的を達成するために、本発明は、ハードウェアが行う処理のアルゴリズムを動作レベルで記述した動作記述を読み込み、レジスタやクロック同期を含むハードウェア特有の概念を意識して記述したRTL記述を生成する半導体集積回路の設計支援装置において、前記RTL記述に対する論理シミュレーションの結果を解析して、前記動作記述における所定単位の動作を表す各ブロックでのレイテンシを求めるレイテンシ解析手段を有することを特徴とする。

[0012]

259 Further, in addition to the above, the apparatus may further include a correspondence table generating means for generating a correspondence table between each block in the behavioral description and the state in the RTL description.

263 また、上記に加えて、前記動作記述における前記各ブロックと、前記RTL記述における状態との対応表を生成する対応表生成手段をさらに有するように構成してもよい。

[0013]

270 Furthermore, the latency analysis means is configured to calculate the latency in each block in the behavioral description by referring to the result of the logic simulation and the correspondence table generated by the correspondence table generation means. You can.

274 また、さらに、前記レイテンシ解析手段は、前記論理シミュレーションの結果と、前記対応表生成手段により生成された対応表とを参照して、前記動作記述における前記各ブロックでのレイテンシを求めるように構成してもよい。

[0014]

282 In addition, as another aspect, a semiconductor integrated circuit reads a behavioral description that describes an algorithm of processing performed by hardware at a behavioral level, and generates an RTL description that is written with consideration of hardware-specific concepts including registers and clock synchronization. The design support method is characterized by comprising a latency analysis step of analyzing the results of logical simulation for the RTL description to obtain the latency in each block representing a predetermined unit of operation in the behavioral description.

290 また、他の態様として、ハードウェアが行う処理のアルゴリズムを動作レベルで記述した動作記述を読み込み、レジスタやクロック同期を含むハードウェア特有の概念を意識して記述したRTL記述を生成する半導体集積回路の設計支援方法において、前記RTL記述に対する論理シミュレーションの結果を解析して、前記動作記述における所定単位の動作を表す各ブロックでのレイテンシを求めるレイテンシ解析工程を有することを特徴とする。

[0015]

300 In addition to the above, the present invention may further include a correspondence table generation step of generating a correspondence table between each block in the behavioral description and the state in the RTL description.

304 また、上記に加えて、前記動作記述における前記各ブロックと、前記RTL記述における状態との対応表を生成する対応表生成工程をさらに有するように構成してもよい。

[0016]

311 Furthermore, the latency analysis step is configured to calculate the latency in each block in the behavioral description by referring to the result of the logic simulation and the correspondence table generated by the correspondence table generation step. It's okay.

315 また、さらに、前記レイテンシ解析工程は、前記論理シミュレーションの結果と、前記対応表生成工程により生成された対応表とを参照して、前記動作記述における前記各ブロックでのレイテンシを求めるように構成してもよい。

[0017]

323 Further, as another aspect, a design step of designing a circuit and creating a layout using the above design support method, a generation step of generating a photomask based on the layout of the circuit designed by the design step, The method for manufacturing a semiconductor integrated circuit may include a transfer step of transferring a photomask pattern generated in the generation step onto a semiconductor wafer.

329 また、他の態様として、上記の設計支援方法を用いて回路を設計しレイアウトを作成する設計工程と、前記設計工程により設計された回路のレイアウトに基づいてフォトマスクを生成する生成工程と、前記生成工程により生成されたフォトマスクのパターンを半導体ウエハに転

写する転写工程と、を有することを特徴とする半導体集積回路の製造方法であってもよい。

[0018]

338 Furthermore, as another aspect, the present invention may be a program for causing a semiconductor integrated circuit design support apparatus to execute the above-described design support method.

342 また、他の態様として、上記の設計支援方法を半導体集積回路の設計支援装置に実行させるためのプログラムであってもよい。

[0019]

349 Furthermore, as another aspect, it may be a recording medium on which a host program is recorded.

352 また、他の態様として、上位プログラムを記録した記録媒体であってもよい。

[0020]

358 According to the present invention, when designing a semiconductor integrated circuit, it is possible to efficiently verify the logic function and timing of a designed logic circuit using RTL simulation, and it is also possible to make corrections at the upstream side of the design process based on the verification results. It becomes possible.

363 本発明によれば、半導体集積回路の設計において、設計した論理回路の論理機能やタイミングをRTLシミュレーションで効率的に検証できるとともに、該検証結果に基づいて設計工程の上流側での修正することが可能となる。

[0021]

371 Embodiments of the present invention will be described below with reference to the drawings.

373 以下、図面を参照しながら本発明の実施形態について説明する。

[0022]

379 FIG. 1 is a functional block diagram showing a schematic configuration of a semiconductor integrated circuit design support apparatus according to an embodiment of the present invention.

383 図 1 は、本発明の実施形態に係る半導体集積回路の設計支援装置の概略構成を示した機能ブロック図である。

387 The design support apparatus of this embodiment includes a behavioral synthesis means 2, a logic simulation means 6, and a latency information display means 9.

390 本実施形態の設計支援装置は、動作合成手段 2、論理シミュレーション手段 6、及びレイテンシ情報表示手段 9 を備える。

[0023]

397 The behavioral synthesis means 2 has a correspondence table generation means 3, which reads the behavioral description 1 and generates an RTL description 4 and a correspondence table 3.

401 動作合成手段 2 は、対応表生成手段 3 を有しており、動作記述 1 を読み込んで R T L 記述 4 と対応表 3 とを生成する。

405 Note that an existing behavioral synthesis algorithm is used.

407 なお、動作合成のアルゴリズムは既存のものを使用する。

410 The correspondence table 5 is information indicating which state in the state machine (state transition machine) in the RTL description 4 each block in the behavioral description 1 corresponds to.

414 対応表 5 は、動作記述 1 における各ブロックが R T L 記述 4 内でのステートマシン（状態遷移機械）におけるどの状態に対応するのかわかる情報である。

418 The block here refers to a part that represents a predetermined unit of operation, and for example, as shown in FIG. means.

421 ここでいうブロックは、所定単位の動作を表す部分で、例えば図4に示すように、C言語での中括弧 (" {","} ") にて囲まれた部分（ブロック41及び42）を意味する。

[0024]

428 An example of correspondence table 5 is shown in FIG.

430 対応表5の例を図2に示す。

433 Here, the correspondence table 5 is composed of three pieces of information: block name, start state, and end state.

436 ここで対応表5は、ブロック名、開始状態、終了状態の3つの情報から構成される。

439 The block name indicates each block in behavioral description 1.

441 ブロック名は、動作記述1における各ブロックを示している。

444 An example of how to name a block is "file name_line number_block type."

446 ブロック名のつけ方の例としては、「ファイル名_行番号_ブロックの種類」がある。

449 Examples of block types include "if" and "case"; however, this block naming method is merely an example, and the present invention is not limited to these.

452 ブロックの種類としては、「i f」や「c a s e」といったものが挙げられるが、このブロック名のつけ方は例示であって、本発明ではこれに限るものではない。

456 The start state indicates a state in which execution of a block is started.

458 開始状態とは、ブロックの実行が開始される状態を示す。

461 Examples of the states here include each state in the state machine (finite state machine) in RTL description 4.

464 ここでいう状態の例として、RTL記述4中のステートマシン（有限状態機械）における各状態が挙げられる。

468 Of course, the starting state may be expressed using the file name, line number, etc. in the RTL description 4 instead of the state of the state machine.

471 もちろん、ステートマシンの状態ではなく、RTL記述4中におけるファイル名や行番号等を用いて開始状態を表現してもよい。

475 Further, the end state indicates a state in which execution of a block is completed.

477 また、終了状態とは、ブロックの実行が完了する状態を示す。

480 In the following description, it is assumed that the block "file1_102_if" in FIG. 2 corresponds to the block 41 in FIG. 4, and the block "file1_104_else" corresponds to the block 42 in FIG. 4.

484 以後の説明では、図2中のブロック「file1_102_if」は図4のブロック41に対応し、ブロック「file1_104_else」は図4のブロック42に対応しているものとする。

[0025]

492 The correspondence table generating means 3 generates a correspondence table 5 of each block in the behavioral description 1 and the state in the RTL description 4 from the behavioral description 1 and the RTL description 4 automatically converted from the behavioral description 1 by the behavioral synthesis means 2.

497 対応表生成手段3は、動作記述1と、動作合成手段2が動作記述1から自動変換したRTL記述4とから、動作記述1における各ブロック及びRTL記述4における状態の対応表5を生成する。

502 Specifically, the behavioral description 1 is first divided into blocks, which are predetermined units of behavior, and then each process in behavioral synthesis (CDFG (Control Data Flow Graph) conversion, scheduling, allocation (data path assignment), state machine generation, etc.)) to identify which block in behavioral description 1 the state in RTL description 4 corresponds to.

508 具体的には、まず動作記述 1 を所定単位の動作であるブロックに分け、次いで、動作合成における各過程（C D F G（Control Data Flow Graph）変換、スケジューリング、アロケーション（データパス割り当て）、ステートマシン生成等）の内容を参照し、R T L記述 4 での状態が動作記述 1 のどのブロックに該当するかを特定する。

514 Then, the first state and last state of the block are specified as the start state and end state so that the range of states corresponding to the block can be known.

517 そして、該当ブロックに相当する状態の範囲がわかるように、ブロック最初の状態と最後の状態を開始状態及び終了状態として特定する。

[0026]

524 The logic simulation means 6 has a latency analysis means 7, which performs logic simulation on the RTL description 4, and uses the latency analysis means 7 to analyze each of the behavioral descriptions 1 while referring to the logic simulation results and the correspondence table 5. Find the block latency.

529 論理シミュレーション手段 6 は、レイテンシ解析手段 7 を有しており、R T L記述 4 に対して論理シミュレーションを実施し、レイテンシ解析手段 7 により、論理シミュレーション結果及び対応表 5 を参照しながら動作記述 1 における各ブロックのレイテンシを求める。

534 Then, the latency information display means 9 displays the determined latency of each block.

536 そして、レイテンシ情報表示手段 9 は、求められた各ブロックのレイテンシを表示する。

[0027]

542 The latency analysis means 7 monitors state transitions from the logic simulation results for the RTL description.

545 レイテンシ解析手段7では、RTL記述に対する論理シミュレーション結果から状態遷移を監視する。

549 A detailed explanation will be given below with reference to FIG. 2.

551 以下、図2を参照して具体的に説明する。

554 In the correspondence table of FIG. 2, state S10 is shown as a start state, and state S21 is shown as an end state.

557 図2の対応表では状態S10が開始状態、状態S21が終了状態と示されている。

560 Therefore, when the state S10 is reached, it is recognized that the execution of the block 41 (file_102_if) has started, and when the simulation progresses further and the state S21 is reached, it is recognized that the execution of the block has been completed.

564 そこで、状態S10に到達したらブロック41 (file_102_if) の実行が開始されたと認識し、そしてさらにシミュレーションが進み状態S21に到達したら同ブロックの実行が完了したと認識する。

569 Then, the difference between the time in state S21 and the time in state S10 is calculated using a timer or the like, and is stored as the latency in block 41 (file_102_if).

572 そして、状態S21の時刻と状態S10の時刻の差分をタイマ等で計算し、ブロック41 (file_102_if) でのレイテンシとして記憶しておく。

576 Furthermore, when block 41 (file_102_if) is executed again, the latency is calculated in the same manner and added to the previously calculated latency.

579 また、ブロック41 (file_102_if) が再度実行された場合は、同様の方法でレイテンシを計算し、以前に求めたレイテンシに加算していく。

583 As shown in FIG. 2, in addition to the correspondence between state S10 and state S21, the

correspondence between the start state of state S12 and the end state of state 30 is specified, and the start state of a block is not limited to one.

587 図2に示すように、状態S10と状態S21の対応のほか、状態S12の開始状態と状態30の終了状態の対応が特定されており、ブロックの開始状態は1つとは限らない。

[0028]

594 FIG. 3 shows an example in which the latency information display means 9 displays the total latency of each block determined by the latency analysis means 7.

597 図3は、レイテンシ解析手段7で求めた各ブロックの総レイテンシをレイテンシ情報表示手段9が表示する場合の例を示したものである。

601 Although only blocks are displayed in FIG. 3, it is also possible to display latency information and the like for each function.

604 図3ではブロックのみが表示されているが、さらに関数ごとのレイテンシ情報等を表示することもできる。

[0029]

611 (Flow of Processing) FIG. 5 is a flowchart showing the flow of processing operations for determining the total latency of each block in the behavioral description in this embodiment.

614 (処理のフロー) 図5は、本実施形態において動作記述における各ブロックの総レイテンシを求める処理動作の流れを示したフローチャートである。

[0030]

621 First, a behavioral description that directly expresses an algorithm for executing a certain process at a behavioral level is created (step S101).

624 まず、ある処理を実行するためのアルゴリズムを直接的に動作レベルで表現する動作記述を作成する(ステップS101)。

628 Next, the behavioral synthesis means automatically converts the behavioral description into an RTL description (step S102).

63 1 次に、動作合成手段により動作記述から R T L 記述への自動変換を行う（ステップ S 1 0 2）。

635 Note that, as described above, behavioral synthesis is performed through various steps such as CDFG (Control Data Flow Graph) conversion, scheduling, allocation (data path assignment), and state machine generation.

639 なお、動作合成は、先述したとおり、C D F G（Control Data Flow Graph）変換、スケジューリング、アロケーション（データパス割り当て）、ステートマシン生成等の各工程を経て行う。

644 After obtaining the RTL description by behavioral synthesis (step S103), the correspondence table generation means generates a correspondence table by referring to each step of the behavioral description, RTL description, and behavioral synthesis (step S104).

648 動作合成により R T L 記述を得た後（ステップ S 1 0 3）、対応表生成手段により、動作記述、R T L 記述、動作合成の各工程を参照して対応表の生成を行う（ステップ S 1 0 4）。

[0031]

655 Once the correspondence table is obtained (step S105), logic simulation is performed on the RTL description to verify the logic function and timing of the designed logic circuit (step S106).

659 対応表が得られたら（ステップ S 1 0 5）、R T L 記述に対して論理シミュレーションを行い、設計した論理回路の論理機能やタイミングの検証を行う（ステップ S 1 0 6）。

663 Then, a latency analysis is performed to obtain the total latency of each block in the behavioral level description from the correspondence between each block of the behavioral description and the state in the RTL description according to the correspondence table and the verification result (step S107) (step S108).

668 そして、対応表による動作記述の各ブロック及びRTL記述における状態の対応関係と、該検証結果（ステップS107）とから、動作レベル記述における各ブロックの総レイテンシを求めるべくレイテンシ解析を行う（ステップS108）。

673 After the total latency of each block is obtained (step S109), this is displayed as latency information by the latency information display means (step S110).

676 各ブロックの総レイテンシが得られた後（ステップS109）、これをレイテンシ情報としてレイテンシ情報表示手段により表示する（ステップS110）。

[0032]

683 In this embodiment, the correspondence table generation means 3 is included in the behavioral synthesis means 2, and the logic simulation means 6 is configured to include the latency analysis means 7, but the behavioral synthesis means 2 and the correspondence table generation means 3 may be configured independently, or the logic simulation means 6 and the latency analysis means 7 may be configured separately.

689 なお、本実施形態では、対応表生成手段3が動作合成手段2に含まれ、論理シミュレーション手段6がレイテンシ解析手段7を包含するように構成しているが、動作合成手段2と対応表生成手段3とを独立して構成してもよいし、また、論理シミュレーション手段6とレイテンシ解析手段7とを別個に構成してもよい。

695 In this case, the automatic conversion function of the RTL description and the correspondence table generation function are separated, and each function is possessed by the behavioral synthesis means 2 and the correspondence table generation means 3, and the logic simulation function and the latency analysis function are separated, respectively. The logic simulation means 6 and the latency analysis means 7 have this.

701 この場合、RTL記述の自動変換機能と対応表生成機能とは分離され、各機能を動作合成手段2及び対応表生成手段3が有することとなり、論理シミュレーション機能とレイテンシ解析機能とは分離され、それぞれを論理シミュレーション手段6及びレイテンシ解析手段7が有することとなる。

[0033]

710 According to the above embodiment, by creating a correspondence table between each block in the behavioral description and the state in the RTL description and comparing it with the RTL simulation results, the latency in each block in the behavioral description can be easily determined.

715 上記の実施形態によれば、動作記述における各ブロックとRTL記述における状態との対応表を作成し、RTLシミュレーション結果と照らし合わせるにより、動作記述における各ブロックにおけるレイテンシを容易に求めることができる。

720 This makes it easy to know which part of the behavioral description is effective to modify in order to reduce latency.

723 これにより、レイテンシを削減するために動作記述のどの部分を修正するのが効果的かを容易に知ることができる。

[0034]

730 The embodiments described above are preferred embodiments of the present invention, and the scope of the present invention is not limited to the above embodiments, and various modifications may be made without departing from the gist of the present invention. It is possible to implement it in

735 なお、上述する実施形態は、本発明の好適な実施形態であり、上記実施形態のみに本発明の範囲を限定するものではなく、本発明の要旨を逸脱しない範囲において種々の変更を施した形態での実施が可能である。

[0035]

743 That is, the design support apparatus of the above-described embodiment operates by processing executed by a computer (CPU, etc.) according to instructions from a program.

746 すなわち、上記した実施形態の設計支援装置は、プログラムの命令によりコンピュータ（CPU等）で実行される処理等によって動作する。

750 The program sends commands to each component of the computer, performs the predetermined processing described above, for example, generates an RTL description and

correspondence table from the behavioral description by the CPU, and generates the RTL description and correspondence table in a logical simulation. Refer to this to find the total latency of each block in the behavioral description.

756 当該プログラムは、コンピュータの各構成要素に指令を送り、先に述べたような所定の処理、例えば、CPUにより、動作記述からRTL記述と対応表を生成し、論理シミュレーションにおいてRTL記述及び対応表を参照して動作記述の各ブロックの総レイテンシを求める。

761 In this way, each process in the design support apparatus of the above embodiment is realized by specific means in which a program and a computer cooperate.

764 このように、上記実施形態の設計支援装置における各処理は、プログラムとコンピュータとが協働した具体的手段によって実現されるものである。

[0036]

771 Then, the computer of the design support apparatus reads and executes the program code stored in the storage medium via a computer-readable recording medium, that is, the storage medium, on which the program code of the software that realizes the functions of the above embodiment is recorded. The object of the present invention can also be achieved.

776 そして、上記実施形態の機能を実現するソフトウェアのプログラムコードを記録したコンピュータ読み取り可能な記録媒体、すなわち記憶メディアを介して、設計支援装置のコンピュータが記憶メディアに格納されたプログラムコードを読み出し実行することによっても、本発明の目的は達成される。

782 Further, the program can be directly loaded and executed on the computer of the design support apparatus through a communication line without using a recording medium, and the object of the present invention can also be achieved in this way as well.

786 また、プログラムは、記録メディアを介さず、通信回線を通じて直接に設計支援装置のコンピュータにロードし実行することもでき、これによっても同様に本発明の目的は達成される。

[0037]

793 In this case, the program code itself read from the storage medium or loaded through the

communication line and executed realizes the processing functions of the embodiments described above.

797 この場合、記憶メディアから読み出された又は通信回線を通じてロードし実行されたプログラムコード自体が前述の実施形態の処理機能を実現することになる。

801 A storage medium storing the program code constitutes the present invention.

803 そして、そのプログラムコードを記憶した記憶メディアは本発明を構成する。

806 Note that storage media for supplying program codes include, for example, floppy (registered trademark) disks, hard disks, optical disks, magneto-optical disks, CD-ROMs, CD-Rs, non-volatile memory cards, ROMs, magnetic tapes, etc. can be used.

810 なお、プログラムコードを供給するための記憶メディアとしては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、不揮発性のメモリカード、ROM、磁気テープ等を用いることができる。

[0038]

818 In addition, a circuit is designed and a layout is created using the design support method mentioned above, a photomask is generated based on the designed circuit layout, and the pattern of the generated photomask is transferred to a semiconductor wafer. The present invention can also be effectively implemented by manufacturing a semiconductor integrated circuit.

824 また、先に述べた設計支援方法を用いて回路を設計してレイアウトを作成し、設計された回路のレイアウトに基づいてフォトマスクを生成して、生成されたフォトマスクのパターンを半導体ウエハに転写して半導体集積回路を製造することによっても、本発明は実効的に実施される。

[0039]

833 1 is a functional block diagram showing a schematic configuration of a semiconductor integrated circuit design support apparatus according to an embodiment of the present invention.

837 本発明の実施形態に係る半導体集積回路の設計支援装置の概略構成を示した機能ブロック図である。

841 FIG. 3 is a diagram illustrating an example of a correspondence table between each block of a behavioral description and a state in an RTL description in an embodiment of the present invention.

845 本発明の実施形態における動作記述の各ブロック及びRTL記述における状態の対応表の例を示した図である。

849 FIG. 3 is a diagram showing an example of displaying latency information in an embodiment of the present invention.

852 本発明の実施形態におけるレイテンシ情報の表示の例を示した図である。

855 FIG. 3 is a diagram showing an example of a behavioral description in an embodiment of the present invention.

858 本発明の実施形態における動作記述の例を示した図である。

861 7 is a flowchart showing the flow of processing operations for determining the total latency of each block of a behavioral description in an embodiment of the present invention.

864 本発明の実施形態における動作記述の各ブロックの総レイテンシを求める処理動作の流れを示したフローチャートである。

868 Explanation of symbols

870 符号の説明

[0040]

876 1 Behavioral description 2 Behavioral synthesis means 3 Correspondence table generation means 4 RTL description 5 Correspondence table 6 Logic simulation means 7 Latency

analysis means 8 Latency information 9 Latency information display means 41 Block A42
Block B

881 1 動作記述 2 動作合成手段 3 対応表生成手段 4 R T L記述 5 対応表 6 論理シミュレーション手段 7 レイテンシ解析手段 8 レイテンシ情報 9 レイテンシ情報表示手段 4 1 ブロック A 4 2 ブロック B