

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2018年9月20日(20.09.2018)



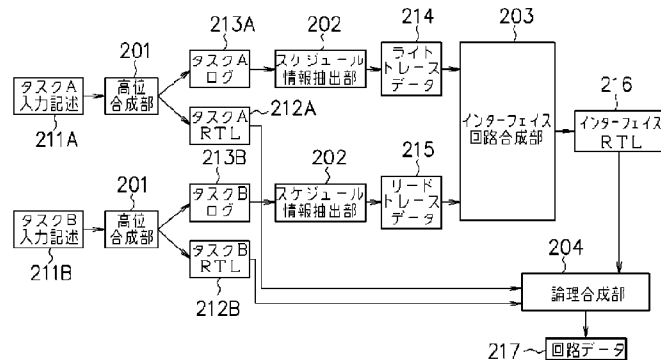
(10) 国際公開番号

WO 2018/168211 A1

- (51) 国際特許分類:
G06F 17/50 (2006.01)
- (21) 国際出願番号: PCT/JP2018/002091
- (22) 国際出願日: 2018年1月24日(24.01.2018)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2017-048999 2017年3月14日(14.03.2017) JP
- (71) 出願人: 富士通株式会社 (FUJITSU LIMITED)
[JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 Kanagawa (JP).
- (72) 発明者: 富田 憲範 (TOMITA, Yoshinori);
〒2118588 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 Kanagawa (JP).
- (74) 代理人: 向山 直樹 (MUKOUYAMA Naoki);
〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 Kanagawa (JP).
- (81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ,

(54) Title: INFORMATION PROCESSING DEVICE, INFORMATION PROCESSING METHOD, AND PROGRAM

(54) 発明の名称: 情報処理装置、情報処理方法及びプログラム



- 201 High level synthesis unit
202 Schedule information extraction unit
203 Interface circuit synthesis unit
204 Logical synthesis unit
211A Task A input description
211B Task B input description
212A Task A RTL
212B Task B RTL
213A Task A log
213B Task B log
214 Write trace data
215 Read trace data
216 Interface RTL
217 Circuit data

(57) Abstract: [Problem] To allow automated interface circuit synthesis with which, while preventing an increase in circuit area, it is possible to improve performance. [Solution] Provided is an information processing device, comprising: a high level synthesis unit which high level synthesizes a circuit module which executes a process of a task A on the basis of an input description and a circuit module which accepts output data of the task A and executes a process of a task B; and an interface circuit synthesis unit which, on the basis of write trace data and read trace data which relate to an



NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT,
QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA,
UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類：

- 国際調査報告（条約第21条(3)）

interface circuit, synthesizes an interface circuit between the circuit modules. If, when synthesizing the interface circuit, the minimum execution commencement gap of the interface circuit which has been derived on the basis of the write trace data and the read trace data is greater than the minimum execution commencement gap of the circuit modules, a storage element is provided which is separate from a memory which is internal to the interface circuit and the interface circuit execution commencement gap is shortened.

(57) 要約：【課題】回路面積の増加を抑えつつ性能を向上できるインターフェイス回路を自動合成できるようにする。【解決手段】入力記述に基づいてタスクAの処理を実行する回路モジュール、及びタスクAの出力データを受けてタスクBの処理を実行する回路モジュールを高位合成する高位合成部と、インターフェイス回路に係るライトトレースデータ及びリードトレースデータに基づいて回路モジュール間のインターフェイス回路を合成するインターフェイス回路合成部とを有し、インターフェイス回路を合成する際、ライトトレースデータ及びリードトレースデータに基づいて求めたインターフェイス回路の最小の実行開始間隔が、回路モジュールの最小の実行開始間隔よりも大きい場合、インターフェイス回路内メモリとは別の記憶素子を設けてインターフェイス回路の実行開始間隔を短縮する。

明 細 書

発明の名称： 情報処理装置、情報処理方法及びプログラム

技術分野

[0001] 本発明は、情報処理装置、情報処理方法及びプログラムに関する。

背景技術

[0002] 回路設計において、R T L (Register Transfer Level) 設計ツールや高位合成ツール

の進歩により、回路モジュールの設計は設計工数を削減できている。その一方で、回路モジュール間でのデータ転送を担う、回路モジュール間を接続するためのインターフェイス回路の設計については、設計ツールによる設計支援があまりなされていない。回路モジュール間のインターフェイス回路は、システム全体の性能と回路面積との両方に影響を与えるので、設計時にはどのように実現するかを検討することは重要である。

[0003] 回路モジュール間のインターフェイス回路の作成方法としては、回路モジュールの動作にあわせた専用回路として人手により設計する方法がある。この方法では、インターフェイス回路は専用化されるため、性能や回路面積の点では優れたものが作成できるが、設計に多大な時間及び労力を要する。また、回路モジュールのインターフェイスが変更された場合、インターフェイス回路も変更が必要となり手間がかかってしまう。

[0004] また、インターフェイス回路の別の作成方法としては、回路モジュール間のインターフェイスとして標準的な方式を採用し、I P マクロを使って作成する方法がある。この方法では、設計の工数は低減できるが、性能向上や回路面積削減を目的としたカスタマイズの実現可能範囲が限られてしまう。また、インターフェイス回路の別の作成方法として、回路モジュール間のインターフェイス回路を自動合成する技術が提案されている（例えば、特許文献1 参照）。

先行技術文献

特許文献

[0005] 特許文献1：特開2004-54641号公報

発明の概要

発明が解決しようとする課題

[0006] 図26に示すようにタスクAの処理を実行する回路モジュール2601と、回路モジュール2601から出力されるデータを用いてタスクBの処理を実行する回路モジュール2603とが、インターフェイス回路2602により接続されている場合、回路モジュール2601、インターフェイス回路2602、回路モジュール2603のいずれかがボトルネックとなることでシステム全体の性能が決まる。インターフェイス回路2602は、回路モジュール2601、2603間でデータを転送するためのメモリを有している。

[0007] ここで、インターフェイス回路2602の実行開始間隔（同じ処理を繰り返す間隔）は、回路モジュール2601による書き込み動作と、回路モジュール2603による読み出し動作と、インターフェイス回路2602のメモリの容量とによって定まる。回路面積が大きくなることを避けるため、インターフェイス回路2602のメモリの容量を小さくすると、インターフェイス回路2602の実行開始間隔が大きくなる。インターフェイス回路2602の実行開始間隔が、回路モジュール2601及び回路モジュール2603の実行開始間隔よりも大きい場合、インターフェイス回路2602がボトルネックとなる。その結果、回路モジュール2601から出力されたデータが回路モジュール2603に入力されるまでの期間が長くなり、システム全体の性能が低下してしまう。

[0008] 1つの側面では、本発明の目的は、回路面積の増加を抑えつつ性能を向上できるインターフェイス回路を自動合成することができる情報処理装置を提供することにある。

課題を解決するための手段

[0009] 情報処理装置の一態様は、それぞれの入力記述に基づいて、第1のタスク

の処理を実行する第１の回路モジュール、及び第１の回路モジュールから出力されるデータを受けて第２のタスクの処理を実行する第２の回路モジュールを高位合成する高位合成部と、データが入出力されるメモリを有して回路モジュール間のデータの転送を担うインターフェイス回路を、インターフェイス回路に対する第１の回路モジュールによるデータの書き込み情報及び第２の回路モジュールによるデータの読み出し情報に基づいて合成する回路合成部とを有する。回路合成部は、データの書き込み情報及びデータの読み出し情報に基づいてインターフェイス回路の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、第１の回路モジュール及び第２の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、第１の回路モジュール及び第２の回路モジュールの最小の実行開始間隔に基づいて、インターフェイス回路内にメモリに入出力されるデータを記憶する、メモリとは別の記憶素子を設ける。

発明の効果

[0010] 発明の一態様においては、回路面積の増加を抑えつつ実行開始間隔を短縮し性能を向上させることができるインターフェイス回路を自動合成することが可能となる。

図面の簡単な説明

[0011] [図1]図１は、本発明の実施形態における情報処理装置の構成例を示す図である。

[図2]図２は、本実施形態における情報処理装置の機能構成例を示す図である。

[図3]図３は、本実施形態における情報処理装置の処理例を示すフローチャートである。

[図4]図４は、入力記述の一例を示す図である。

[図5]図５（Ａ）は、タスクＡに係る回路モジュールの例を示すブロック図であり、図５（Ｂ）は、タスクＡのスケジュール情報を説明する図である。

[図6]図６（Ａ）は、タスクＢに係る回路モジュールの例を示すブロック図で

あり、図 6（B）は、タスク B のスケジュール情報を説明する図である。

[図7]図 7（A）は、ライトトレースデータを説明する図であり、図 7（B）は、リードトレースデータを説明する図である。

[図8]図 8 は、インターフェイス回路に対する動作を説明する図である。

[図9]図 9 は、タスク A、インターフェイス回路、及びタスク B の処理タイミングの一例を示す図である。

[図10]図 10（A）～図 10（C）は、インターフェイス回路の実行開始間隔の改善方法を説明する図である。

[図11]図 11 は、インターフェイス回路に対する動作を説明する図である。

[図12]図 12 は、インターフェイス回路の構成例を示す図である。

[図13]図 13 は、制御回路の構成例を示す図である。

[図14]図 14 は、インターフェイス回路の動作例を示すタイミングチャートである。

[図15]図 15 は、インターフェイス回路の他の構成例を示す図である。

[図16]図 16（A）及び図 16（B）は、複数のインスタンスを用いる例を説明する図である。

[図17]図 17 は、インターフェイス回路内のメモリ容量を削減する例を説明する図である。

[図18]図 18 は、タスク A のスケジュール情報を示す図である。

[図19]図 19（A）～図 19（C）は、タスク A のタスク実行開始間隔を変化させたときのスケジュール情報を示す図である。

[図20]図 20（A）及び図 20（B）は、インターフェイス回路のスケジュールを示す図である。

[図21]図 21（A）及び図 21（B）は、分離するインターフェイス回路内のメモリ要素の決定方法を説明する図である。

[図22]図 22（A）及び図 22（B）は、改善後のインターフェイス回路のスケジュールを示す図である。

[図23]図 23 は、別に設けるメモリ素子のサイズの決定方法を説明する図で

ある。

[図24]図24は、別に設けるメモリ素子の動作モデルを示す図である。

[図25]図25は、本実施形態における情報処理装置を実現可能なコンピュータの機能ブロック図である。

[図26]図26は、設計対象回路の構成例を示す図である。

発明を実施するための形態

[0012] 以下、本発明の実施形態を図面に基づいて説明する。

なお、以下では、設計対象のハードウェア回路として、図26に示したようにタスクAの処理を実行するとともに、タスクAの出力データを受けてタスクBの処理を実行する回路を一例に説明する。

[0013] 図1は、本発明の一実施形態における情報処理装置の構成例を示す図である。本実施形態における情報処理装置としてのサーバ121は、ユーザー（ハードウェア回路の設計者等）110が行う入力操作等に応じて、設計データ122を基に設計対象のハードウェア回路を自動合成して回路データを提供する。設計データ122は、設計対象の回路の入力記述（動作記述）や合成制約等を含む。

[0014] 設計データ122に基づくハードウェア回路は、高位合成ツール131、論理合成ツール132、スケジュール情報抽出ツール133、インターフェイス回路合成ツール134等のソフトウェアをサーバ121で実行することで自動的に合成される。高位合成ツール131は、入力記述（動作記述）から、その入力記述に応じた処理を実行する回路モジュール（RTL記述）を合成するためのツールである。論理合成ツール132は、RTL記述に基づいてゲートレベルの回路データを生成するためのツールである。

[0015] スケジュール情報抽出ツール133は、スケジュール情報からデータの出入力タイミングの情報（ライトトレースデータやリードトレースデータ）を取得するためのツールである。インターフェイス回路合成ツール134は、スケジュール情報から取得したライトトレースデータ及びリードトレースデータに基づいて、インターフェイス回路（RTL記述）を合成するためのツ

ールである。

[0016] ここで、スケジュール情報は、回路モジュールの合成時に出力されるログ情報の1つとして得られ、回路モジュールの処理の流れを示す情報である。また、ライトトレースデータは、インターフェイス回路内のどのメモリ要素に、どのタイミングでデータを書き込むかを示すデータの書き込み情報であり、リードトレースデータは、インターフェイス回路内のどのメモリ要素から、どのタイミングでデータを読み出すかを示すデータの読み出し情報である。

[0017] 図2は、本実施形態における情報処理装置の機能構成例を示す図である。本実施形態における情報処理装置は、高位合成部201、スケジュール情報抽出部202、インターフェイス回路合成部203、及び論理合成部204を有する。高位合成部201、スケジュール情報抽出部202、インターフェイス回路合成部203、及び論理合成部204の各機能は、それぞれ高位合成ツール131、スケジュール情報抽出ツール133、インターフェイス回路合成ツール134、及び論理合成ツール132によって実現される。

[0018] 高位合成部201は、入力記述（動作記述）に記載された処理を実行する回路モジュール（RTL記述）を合成する。高位合成部201は、パイプライン合成機能を有しており、パイプライン動作可能な回路モジュールを合成する。高位合成部201は、タスクAの入力記述211Aが入力されると、入力記述211Aに基づいてタスクAの処理を実行する回路モジュールを合成し、その回路モジュールのRTL記述212A及びログ情報213Aを出力する。また、高位合成部201は、タスクBの入力記述211Bが入力されると、入力記述211Bに基づいてタスクBの処理を実行する回路モジュールを合成し、その回路モジュール（タスクB）のRTL記述212B及びログ情報213Bを出力する。ログ情報213A、213Bは、回路モジュールでの処理の流れを示すスケジュール情報を含む。

[0019] スケジュール情報抽出部202は、高位合成部201が出力するログ情報

におけるスケジュール情報からデータの入出力タイミングの情報（ライトトレースデータやリードトレースデータ）を取得する。スケジュール情報抽出部202は、データを出力する（データを書き込む）側である回路モジュール（タスクA）のログ情報213Aにおけるスケジュール情報からライトトレースデータ214を取得して出力する。また、スケジュール情報抽出部202は、データが入力される（データを読み出す）側である回路モジュール（タスクB）のログ情報213Bにおけるスケジュール情報からリードトレースデータ215を取得して出力する。

[0020] インターフェイス回路合成部203は、スケジュール情報抽出部202が出力するライトトレースデータ及びリードトレースデータに基づいて、回路モジュール間のデータ転送を担うインターフェイス回路（RTL記述）を合成する。インターフェイス回路合成部203は、ライトトレースデータ214及びリードトレースデータ215を用いてインターフェイス回路2602内のメモリのライフタイム解析を行う。また、インターフェイス回路合成部203は、解析結果に基づいてインターフェイス回路を合成し、インターフェイス回路のRTL記述216を出力する。なお、ライフタイムとは、メモリ要素に対してデータが書き込まれた時に開始され、そのデータが最後に読み出された時に終了するものであり、メモリ要素がデータを保持する期間を示す。

[0021] 論理合成部204は、各回路の回路情報（RTL記述）を論理合成しゲートレベルの回路データを生成する。論理合成部204は、高位合成部201が出力する回路モジュールのRTL記述212A、212Bと、インターフェイス回路合成部203が出力するインターフェイス回路のRTL記述216とに基づいて、設計対象であるハードウェア回路のゲートレベルの回路データ217を生成し出力する。

[0022] 図3は、本実施形態における情報処理装置の処理例を示すフローチャートである。

処理を開始すると、ステップS301にて、情報処理装置の高位合成部2

01が、それぞれの入力記述に基づいて、タスクAの処理を実行する回路モジュール及びタスクBの処理を実行する回路モジュールを高位合成する。次に、ステップS302にて、情報処理装置のスケジュール情報抽出部202が、ステップS301での高位合成により得られたタスクAの処理を実行する回路モジュールのスケジュール情報からライトトレースデータを取得し、タスクBの処理を実行する回路モジュールのスケジュール情報からリードトレースデータを取得する。

[0023] 続いて、ステップS303にて、情報処理装置のインターフェイス回路合成部203が、ステップS302において取得されたライトトレースデータ及びリードトレースデータ

を用いて、インターフェイス回路内メモリのライフタイム解析を行う。次に、ステップS304にて、インターフェイス回路合成部203が、各回路モジュールのスケジュール情報及びライフタイムの解析結果に基づいて、各回路モジュール及びインターフェイス回路の実行開始間隔を決定する。ここで、実行開始間隔とは、同じタスクの処理を繰り返す間隔であり、タスクの処理を開始してから次の同じタスクの処理を開始するまでの時間間隔である。

[0024] 次に、ステップS305にて、インターフェイス回路合成部203が、設計対象の回路全体での処理性能及び回路面積を評価し、処理性能及び回路面積が所定の条件を満たさない場合、ステップS306へ進む。例えば、インターフェイス回路合成部203は、インターフェイス回路の最小の実行開始間隔が各回路モジュールの最小の実行開始間隔よりも大きく、インターフェイス回路がボトルネックとなると判定した場合、ステップS306へ進む。

[0025] ステップS306にて、インターフェイス回路合成部203が、インターフェイス回路内メモリのメモリ要素を1つ選択して別のメモリ素子（記憶素子）へ分離する。そして、別のメモリ素子へ分離した状態での、インターフェイス回路内メモリのライフタイム解析を行い、解析結果を更新して、ステップS304へ戻る。

[0026] ステップS305において、回路全体での処理性能及び回路面積が所定の条件を満たすと判定した場合、ステップS307にて、インターフェイス回路合成部203が、インターフェイス回路の合成を行ってインターフェイス回路のRTL記述を生成する。続いて、ステップS308にて、情報処理装置の論理合成部204が、ステップS301において得られた各回路モジュールのRTL記述、及びステップS307において得られたインターフェイス回路のRTL記述を論理合成して回路データを出力し、処理を終了する。

[0027] 以下、本実施形態における情報処理装置による回路合成について、図4に示す入力記述に記載した処理を実行する回路を合成する場合を例に具体的に説明する。図4は、入力記述の一例を示す図である。図4において、401がタスクAの入力記述であり、402がタスクBの入力記述である。なお、入力記述には、2つの回路モジュールに限らず、他の回路モジュール（図4に示した例ではタスクC）の入力記述も適宜含まれる。

[0028] なお、本実施形態では、タスクは、1つのインターフェイス回路内メモリに対して書き込み及び読み出しの両方を行うことはなく、1つのインターフェイス回路内メモリに対しては書き込み又は読み出しの一方だけを行えるとする。また、タスクは、タスクの1回の実行中、インターフェイス回路内メモリの各メモリ要素に対する書き込みはそれぞれ1回だけ行えるとし、インターフェイス回路内メモリの各メモリ要素に対する読み出しはそれぞれ1回以上行っても良いとする。

[0029] タスクAの入力記述401は、データin[0]～in[7]を入力とし、データtmp12[0]～tmp12[7]を出力として、データin[j]を用いて演算処理を行って演算結果をデータtmp12[j]として出力する処理を、値jを0から1ずつ増加させて8回繰り返し行うことを示している。また、タスクBの入力記述402は、データtmp12[0]～tmp12[7]を入力とし、データtmp23[0]～tmp23[7]を出力として、データtmp12[j]とtmp12[(j+1)%8]（%は剰余演算子）を用いて演算処理を行って演算結果をデータtmp23[j]

］として出力する処理を、値 j を 0 から 1 ずつ増加させて 8 回繰り返し行うことを示している。

[0030] 入力データはメモリ（RAM）に格納されていると指定し、出力データはメモリ（RAM）に格納すると指定して、入力記述 401、402 を高位合成部 201 に入力すると、

高位合成部 201 は、タスク A の処理を実行する回路モジュール及びタスク B の処理を実行する回路モジュールをそれぞれ合成する。高位合成部 201 による高位合成での合成結果の一例を図 5 及び図 6 に示す。

[0031] 図 5（A）は、タスク A の入力記述 401 に基づいて合成された回路モジュールの例を示すブロック図である。なお、図 5（A）においては、データパスに係る部分のみを示している。図 5（A）に示すように、タスク A の処理を実行する回路モジュール 510 は、レジスタ 511、513 及び組み合わせ論理回路 512 を有する。メモリ 520 は、タスク A での入力データ i_n を格納するメモリ（RAM）であり、メモリ 530 は、タスク A での出力データ $tmp12$ を格納するメモリ（RAM）である。

[0032] レジスタ（reg0）511 は、入力データとしてメモリ 520 から読み出したデータを格納するレジスタである。組み合わせ論理回路（logic）512 は、レジスタ（reg0）511 に格納されたデータを用いた演算処理を実行する回路であり、演算結果として得られるデータはレジスタ（reg1）513 に格納される。レジスタ（reg1）513 は、出力データとしてメモリ 530 に書き込むデータを格納するレジスタである。

[0033] 図 5（B）は、合成結果のログ情報の 1 つとして得られるタスク A のスケジュール情報を説明する図である。図 5（B）において、 $in[0] \sim in[7]$ 、reg0、logic、reg1、 $tmp12[0] \sim tmp12[7]$ は、タスク A の処理に係る回路リソースを示している。 $in[0] \sim in[7]$ は入力データ i_n が格納されているメモリ 520 のメモリ要素を示し、 $tmp12[0] \sim tmp12[7]$ は出力データ $tmp12$ を格納するメモリ 530 のメモリ要素を示す。また、reg0、logic、re

g 1 は、回路モジュール 5 1 0 が有するレジスタ 5 1 1、組み合わせ論理回路 5 1 2、レジスタ 5 1 3 をそれぞれ示す。また、図 5 (B) において、r、w、a は回路リソースの動作を示しており、r は読み出し、w は書き込み、a は動作している状態を示す。

[0034] 図 5 (B) に示すようにタスク A の処理では、サイクル 0 でメモリ 5 2 0 のメモリ要素 i n [0] からデータが読み出され、その読み出されたデータがサイクル 1 でレジスタ 5 1 1 に書き込まれる。そして、サイクル 2 で、サイクル 1 にレジスタ 5 1 1 に書き込まれたデータを用いて、組み合わせ論理回路 5 1 2 が演算処理を行い、演算結果のデータがレジスタ 5 1 3 に書き込まれる。次のサイクル 3 で、サイクル 2 にレジスタ 5 1 3 に書き込まれたデータがメモリ 5 3 0 のメモリ要素 t m p 1 2 [0] に書き込まれる。

[0035] 回路モジュール 5 1 0 はパイプライン動作するので、サイクル 1 でメモリ 5 2 0 のメモリ要素 i n [1] からデータが読み出され、サイクル 2 でメモリ 5 2 0 のメモリ要素 i n [2] からデータが読み出され、以降同様にメモリ 5 2 0 のメモリ要素 i n [3] ~ i n [7] からデータが順次読み出されて、前述の処理が繰り返し行われる。そして、サイクル 1 0 で、レジスタ 5 1 3 からメモリ 5 3 0 のメモリ要素 t m p 1 2 [7] へのデータの書き込みが行われて 1 つのタスク A の処理が終了する。

[0036] このようにタスク A の 1 回の実行には 1 1 サイクルを要するが、回路モジュール 5 1 0 はタスクレベルでのパイプライン動作が可能であるので、図 5 (B) に示すように 1 回目のタスク A の処理が実行中であるサイクル 8 から 2 回目のタスク A の処理を開始することができる。すなわち、タスク A の処理を実行する回路モジュール 5 1 0 は、最短で 8 サイクル毎に同じタスク A の処理を繰り返すことが可能であり、回路モジュール 5 1 0 における最小の実行開始間隔は 8 サイクルとなる。

[0037] 図 6 (A) は、タスク B の入力記述 4 0 2 に基づいて合成された回路モジュールの例を示すブロック図である。なお、図 6 (A) においても、データパスに係る部分のみを示し

ている。図6（A）に示すように、タスクBの処理を実行する回路モジュール610は、レジスタ611、613及び組み合わせ論理回路612を有する。メモリ620は、タスクBでの入力データtmp12を格納するメモリ（RAM）であり、メモリ630は、タスクBでの出力データtmp23を格納するメモリ（RAM）である。

[0038] レジスタ（reg0）611は、入力データとしてメモリ620から読み出したデータを格納するレジスタである。組み合わせ論理回路（logic）612は、レジスタ（reg0）611に格納されたデータを用いた演算処理を実行する回路であり、演算結果として得られるデータはレジスタ（reg1）613に格納される。レジスタ（reg1）613は、出力データとしてメモリ630に書き込むデータを格納するレジスタである。

[0039] 図6（B）は、合成結果として得られるタスクBのスケジュール情報を説明する図である。図6（B）において、tmp12[0]～tmp12[7]、reg0、logic、reg1、tmp23[0]～tmp23[7]は、タスクBの処理に係る回路リソースを示している。tmp12[0]～tmp12[7]は入力データtmp12が格納されているメモリ620のメモリ要素を示し、tmp23[0]～tmp23[7]は出力データtmp23を格納するメモリ630のメモリ要素を示す。また、reg0、logic、reg1は、回路モジュール610が有するレジスタ611、組み合わせ論理回路612、レジスタ613をそれぞれ示す。また、図6（B）において、r、w、aは回路リソースの動作を示しており、rは読み出し、wは書き込み、aは動作している状態を示す。

[0040] 図6（B）に示すようにタスクBの処理では、サイクル0でメモリ530のメモリ要素tmp12[0]及びtmp12[1]からデータが読み出され、その読み出されたデータがサイクル1でレジスタ611に書き込まれる。そして、サイクル2で、サイクル1にレジスタ611に書き込まれたデータを用いて、組み合わせ論理回路612が演算処理を行い、演算結果のデータがレジスタ613に書き込まれる。次のサイクル3で、サイクル2にレジ

スタ 6 1 3 に書き込まれたデータがメモリ 6 3 0 のメモリ要素 t m p 2 3 [0] に書き込まれる。

[0041] 回路モジュール 6 1 0 はパイプライン動作するので、サイクル 1 でメモリ 6 2 0 のメモリ要素 t m p 1 2 [1] 及び t m p 1 2 [2] からデータが読み出され、以降同様にメモリ 6 2 0 のメモリ要素からデータが順次読み出されて、前述の処理が繰り返し行われる。そして、サイクル 7 でメモリ 6 2 0 のメモリ要素 t m p 1 2 [1] 及び t m p 1 2 [7] からデータが読み出され、それを用いた演算処理が行われ、サイクル 1 0 でレジスタ 6 1 3 からメモリ 6 3 0 のメモリ要素 t m p 2 3 [7] へのデータの書き込みが行われて 1 つのタスク B の処理が終了する。

[0042] このようにタスク B の 1 回の実行には 1 1 サイクルを要するが、回路モジュール 6 1 0 はタスクレベルでのパイプライン動作が可能であるので、図 6 (B) に示すように 1 回目のタスク B の処理が実行中であるサイクル 8 から 2 回目のタスク B の処理を開始することができる。すなわち、タスク B の処理を実行する回路モジュール 6 1 0 は、最短で 8 サイクル毎に同じタスク B の処理を繰り返すことが可能であり、回路モジュール 6 1 0 における最小の実行開始間隔は 8 サイクルとなる。

[0043] 前述したようなタスク A、タスク B のそれぞれのスケジュール情報が得られると、スケジュール情報抽出部 2 0 2 は、スケジュール情報に基づいて、インターフェイス回路のメモリに係るライトトレースデータ及びリードトレースデータを取得する。ライトトレースデータは、データ出力側であるタスク A のスケジュール情報の中から、回路モジュール 5 1 0 からメモリ 5 3 0 へのデータの書き込みを抽出することで取得でき、図 7 (A) に示すようになる。また、リードトレースデータは、データ入力側であるタスク B のスケジュール情報の中から、メモリ 6 2 0 から回路モジュール 6 1 0 へのデータの読み出しを抽出することで取得でき、図 7 (B) に示すようになる。図 7 (A) に示すライトトレースデータ及び図 7 (B) に示すリードトレースデータは、下記のような数式で表現できる。ここで、 $w[i] = j$ は

、第 i のメモリ要素にサイクル j で書き込むことを示し、 $r[i] = \{j, k\}$ は、第 i のメモリ要素をサイクル j, k で読み出すことを示す。

[0044] [数1]

$w[0]=0$	$r[0]=\{0,7\}$
$w[1]=1$	$r[1]=\{0,1\}$
$w[2]=2$	$r[2]=\{1,2\}$
$w[3]=3$	$r[3]=\{2,3\}$
$w[4]=4$	$r[4]=\{3,4\}$
$w[5]=5$	$r[5]=\{4,5\}$
$w[6]=6$	$r[6]=\{5,6\}$
$w[7]=7$	$r[7]=\{6,7\}$

[0045] 次に、インターフェイス回路合成部 203 は、取得されたインターフェイス回路のメモリに係るライトトレースデータ及びリードトレースデータを用いてメモリのライフタイム解析を行い、メモリ要素毎にどの時間帯に値を保持するか情報を求める。タスク A とタスク B との間には、タスク A の処理を開始した後にタスク B の処理を開始するという依存関係がある。そこで、タスク A の処理がサイクル 0 に開始され、タスク B の処理がサイクル t ($t > 0$) に開始されるとすると、ライトトレースデータ及びリードトレースデータは、下記のように書き換えられる。

[0046] [数2]

$w[0]=0$	$r[0]=\{t+0, t+7\}$
$w[1]=1$	$r[1]=\{t+0, t+1\}$
$w[2]=2$	$r[2]=\{t+1, t+2\}$
$w[3]=3$	$r[3]=\{t+2, t+3\}$
$w[4]=4$	$r[4]=\{t+3, t+4\}$
$w[5]=5$	$r[5]=\{t+4, t+5\}$
$w[6]=6$	$r[6]=\{t+5, t+6\}$
$w[7]=7$	$r[7]=\{t+6, t+7\}$

[0047] インターフェイス回路内のメモリにおけるデータは書き込んだ後に読み出し、また処理性能の観点からタスク B をできるだけ早く実行開始したいので、

[0048] [数3]

$$w[x]=y, \quad y \in r[x]$$

[0049] を満たす最小の値 t を求める。具体的には、下記のように展開される。

[0050] [数4]

$$\begin{array}{ll} x=0: & 0 < t+0 \text{ かつ } 0 < t+7 \\ x=1: & 1 < t+0 \text{ かつ } 1 < t+1 \\ x=2: & 2 < t+1 \text{ かつ } 2 < t+2 \\ x=3: & 3 < t+2 \text{ かつ } 3 < t+3 \\ x=4: & 4 < t+3 \text{ かつ } 4 < t+4 \\ x=5: & 5 < t+4 \text{ かつ } 5 < t+5 \\ x=6: & 6 < t+5 \text{ かつ } 6 < t+6 \\ x=7: & 7 < t+6 \text{ かつ } 7 < t+7 \end{array}$$

[0051] これを解くと、 $t = 2$ が得られる。したがって、メモリのライフタイム解析の結果、インターフェイス回路のメモリに対するデータの書き込み及び読み出しは、図8に示すようなタイミングで実行されることとなる。図8は、インターフェイス回路のメモリに対する動作を説明する図である。第0のメモリ要素ではサイクル0からサイクル9までの10サイクルの間においてデータを保持し、その他のメモリ要素では3サイクルの間においてデータを保持することとなる。したがって、インターフェイス回路は、1回の実行に10サイクルを要し、インターフェイス回路における最小の実行開始間隔は10サイクルとなる。

[0052] ライフタイムの解析結果を基にインターフェイス回路における最小の実行開始間隔が求まると、インターフェイス回路合成部203は、タスクAの処理を実行する回路モジュール510、タスクBの処理を実行する回路モジュール610、及び回路モジュール510、610を接続するためのインターフェイス回路のそれぞれの実行開始間隔を決定する。本例では、回路モジュール510、610における最小の実行開始間隔は8サイクルであるが、インターフェイス回路における最小の実行開始間隔は10サイクルである。そのため、回路モジュール510、回路モジュール610、及びインターフェ

イス回路の実行開始間隔は、これらの最小の実行開始間隔の中で最も大きい 10 サイクルに決定される。

[0053] この場合、タスク A の処理を実行する回路モジュール、インターフェイス回路、及びタスク B の処理を実行する回路モジュールの動作タイミングは、図 9 に示すようになる。図 9 は、タスク A、インターフェイス回路、及びタスク B の処理タイミングの一例を示す図である。図 9 において、1 F 1 2 がインターフェイス回路の処理タイミングを示している。タスク A は 10 サイクル毎に新しい入力データを処理し、タスク B は 10 サイクル毎に新しい出力データを出力しているので、タスク A 及びタスク B を実行するシステムでは、10 サイクルに 1 回の割合で新データを処理する性能を有する。

[0054] 回路モジュール 5 1 0、6 1 0 における最小の実行開始間隔は 8 サイクルであるのに対し、インターフェイス回路における最小の実行開始間隔は 10 サイクルである。このようにインターフェイス回路における最小の実行開始間隔が、回路モジュールにおける最小の実行開始間隔よりも大きい場合、回路モジュールにおける最小の実行開始間隔に基づいて、インターフェイス回路内に別のメモリ素子（記憶素子）を設けることで、インターフェイス回路がボトルネックとなることを回避することができる。

[0055] 以下に、インターフェイス回路内に別のメモリ素子を設けることで、インターフェイス回路における最小の実行開始間隔を改善する手法について説明する。インターフェイス回路合成部 2 0 3 は、インターフェイス回路のメモリに対するアクセスタイミングにおいて、次回のタスクの処理の中で最初に接触するメモリ要素を選択する。本例では、図 8 に示したように、インターフェイス回路内メモリの第 0 のメモリ要素が最初に接触するメモリ要素であるので、第 0 のメモリ要素が選択される。

[0056] そして、サイクル 2 でインターフェイス回路内メモリの第 0 のメモリ要素からデータを読み出した後、そのデータを別のメモリ素子 x 0 に書き込み、サイクル 9 では第 0 のメモリ要素に対する読み出しは行わず、別のメモリ素子 x 0 からデータを読み出すように制御する。このようにすることで、図 1

図 10 (A) に示すインターフェイス回路におけるアクセスタイミングが、図 10 (B) に示すインターフェイス回路におけるアクセスタイミングに変更され、第 0 のメモリ要素のライフタイムが 10 サイクルから 3 サイクルに短縮される。

[0057] 変更後のライフタイムに基づいて、インターフェイス回路における最小の実行開始間隔を求めると 3 サイクルとなる。なお、メモリ素子 x 0 のライフタイムは考慮しない。したがって、パイプライン動作する場合、図 10 (C) に示すようなアクセスタイミングでのインターフェイス回路へのアクセスが可能となる。ここで、図 10 (C) から明らかなように、メモリ素子 x 0 からは書き込まれた順でデータが読み出されるので、メモリ素子 x 0 として F I F O (First In First Out) バッファを用いることで、メモリ素子 x 0 における各データのライフタイムが重なっても、タスク A からタスク B へ正しくデータを受け渡すことが可能である。

[0058] 図 10 (C) に示すようにインターフェイス回路へアクセスを行う場合、サイクル 8 までに 3 個のデータが貯まりサイクル 9 でデータが読み出され、以降はデータの書き込みと読み出しが交互に繰り返されるので、3 段の F I F O バッファをメモリ素子 x 0 とすれば良く、回路面積の増加も抑制できる。なお、F I F O バッファのサイズは、ライトトレースデータ及びリードトレースデータに基づいてインターフェイス回路内メモリのライフタイム解析を行い、解析結果に基づいて決定する。

[0059] 以上のようにインターフェイス回路に別のメモリ素子 x 0 を設けることで、インターフェイス回路における最小の実行開始間隔を改善することができる。この改善後のインターフェイス回路における最小の実行開始間隔で、インターフェイス回路合成部 203 が、回路モジュール 510、回路モジュール 610、及びインターフェイス回路のそれぞれの実行開始間隔を再び決定する。回路モジュール 510、610 における最小の実行開始間隔は 8 サイクルであり、インターフェイス回路における最小の実行開始間隔は 3 サイクルであるので、回路モジュール 510、回路モジュール 610、及びインタ

ーフフェイス回路の実行開始間隔は、これらの最小の実行開始間隔の中で最も大きい8サイクルに決定される。

[0060] この場合、インターフェイス回路における実行開始間隔を8サイクルにすれば良く、インターフェイス回路におけるアクセスタイミングは図11に示すようになる。図11に示されるように、 n 回目の実行と $(n+1)$ 回目の実行とで、インターフェイス回路内メモリの同じメモリ要素に同時にアクセスしていないので、タスクA及びタスクBの処理を依存関係に従ってパイプライン処理できる。また、メモリ素子 $\times 0$ は、ライフタイムの重なりがないことからFIFOバッファでなくともよく、例えばレジスタであってもよい。

[0061] このようにして別のメモリ素子 $\times 0$ を設けることで実行開始間隔を改善したインターフェイス回路のRTL記述をインターフェイス回路合成部203が生成する。そして、そのインターフェイス回路のRTL記述と回路モジュール510、610のRTL記述を論理合成部204が論理合成して、図4に示した入力記述401、402に記載した処理を実行する回路の回路データが生成される。

[0062] 図12は、本実施形態におけるインターフェイス回路の構成例を示す図である。図12

には、インターフェイス回路内にメモリとFIFOバッファを設けた例を示している。インターフェイス回路(1210)は、タスクAの処理を実行する回路モジュール1220、及びタスクBの処理を実行する回路モジュール1230とそれぞれ接続され、回路モジュール1220から回路モジュール1230へデータを受け渡す。

[0063] 回路モジュール1220は、インターフェイス回路1210からライトレディ信号WRDYが入力され、インターフェイス回路1210にライトイネーブル信号WEN、ライトアドレスWA、及びライトデータWDを出力する。回路モジュール1230は、インターフェイス回路1210からリードレディ信号RRDY及びリードデータRDが入力され、インターフェイス回路

1210にリードイネーブル信号REN及びリードアドレスRAを出力する。なお、インターフェイス回路1210及び回路モジュール1230におけるリードアドレスRA及びリードデータRDに係るポートは、タスクBの処理において同時に読み出すデータ数に応じた数が設けられる。

[0064] インターフェイス回路1210は、制御回路1211、メモリ1212、FIFOバッファ1213、及びマルチプレクサ1214、1215を有する。制御回路1211は、インターフェイス回路1210へのデータの書き込みが可能であることをライトレディ信号WRDYにより回路モジュール1220に通知したり、インターフェイス回路1210からのデータの読み出しが可能であることをリードレディ信号RRDYにより回路モジュール1230に通知したりする。また、制御回路1211は、FIFOバッファ1213、及びマルチプレクサ1214、1215を制御し、FIFOバッファ1213に対するデータの書き込み及び読み出しを制御する。ライトレディ信号WRDYは、“1”がデータの書き込みが可能であることを示すものとし、リードレディ信号RRDYは、“1”がデータの読み出しが可能であることを示すものとする。

[0065] メモリ1212及びFIFOバッファ1213は、回路モジュール1220から回路モジュール1230へ受け渡すデータを記憶する。メモリ1212は、入力されるライトイネーブル信号WENが“1”であるとき、ライトアドレスWAによって指定されるメモリ要素にライトデータWDを格納する。また、メモリ1212は、入力されるリードイネーブル信号RENが“1”であるとき、リードアドレスRAによって指定されるメモリ要素からデータを読み出して第1のリードデータRDとして出力する。

[0066] FIFOバッファ1213は、入力される制御回路1211からのFIFOライトイネーブル信号FWENが“1”であるとき、マルチプレクサ1215から出力されるデータを格納する。また、FIFOバッファ1213は、入力される制御回路1211からのFIFOリードイネーブル信号FRE Nが“1”であるとき、格納しているデータのうちで最も古いデータを第2

のリードデータRDとして出力する。

[0067] マルチプレクサ1214は、制御回路1211から出力される制御信号SEL2に応じて、メモリ1212から出力される第1のリードデータRD又はFIFOバッファ1213から出力される第2のリードデータRDの何れかを、回路モジュール1230にリードデータRDとして出力する。マルチプレクサ1214は、制御信号SEL2が“0”であれば第1のリードデータRDを回路モジュール1230にリードデータRDとして出力し、制御信号SEL2が“1”であれば第2のリードデータRDを回路モジュール1230にリードデータRDとして出力する。ここで、制御回路1211は、スケジュール情報に基づいて予め決定された所定のタイミングで、FIFOリードイネーブル信号FREN及び制御信号SEL2を“1”にする。

[0068] マルチプレクサ1215は、制御回路1211から出力される制御信号SEL1に応じて、メモリ1212から出力される第1のリードデータRD又は回路モジュール1220から出力されるライトデータWDの何れかをFIFOバッファ1213に出力する。マルチプレクサ1215は、制御信号SEL1が“0”であれば第1のリードデータRDをFIFOバッファ1213に出力し、制御信号SEL1が“1”であればライトデータWDをFIFOバッファ1213に出力する。マルチプレクサ1215を設けることで、タスクBの処理において複数回読み出されない、すなわち1回だけ読み出されるデータであっても、FIFOバッファ1213に格納することが可能となる。

[0069] 図13は、本実施形態における制御回路1211の構成例を示す図である。図13には、制御回路1211におけるリードレディ信号RRDYを生成する構成を示している。処理の開始を示すスタート信号が組み合わせ回路1301に入力されると、ライトレディ信号WRDY用のフリップフロップ1302の出力が“1”になる。フリップフロップ1302の出力は、ライトレディ信号WRDYとして回路モジュール1220に出力されるとともに、リードレディ信号生成回路1303に出力される。

[0070] リードレディ信号生成回路1303は、ライトレディ信号WRDYが“1”になってから所定のサイクル数が経過した後にリードレディ信号RDYを“1”にするように動作する。リードレディ信号生成回路1303は、組み合わせ回路1304、フリップフロップ1305、及び加算器1307を有するカウンタにより、ライトレディ信号WRDYが“1”になってからの経過サイクル数をカウントする。また、リードレディ信号生成回路1303は、経過サイクル数のカウント値と予め定められるパラメータ値とを比較器1306により比較することで所定のサイクル数が経過したか否かを判定する。

[0071] そして、リードレディ信号生成回路1303は、所定のサイクル数が経過したと判定した場合、リードレディ信号RDY用のフリップフロップ1308の出力を“1”にする。フリップフロップ1308の出力は、リードレディ信号RDYとして回路モジュール1230に出力される。なお、比較器1306での比較に用いるパラメータ値は、スケジュール情報等に基づいて決められる定数値である。制御回路1211が出力する他の信号も、同様に制御回路1211にカウンタ等を設けて、カウンタや状態を用いて生成すれば良い。

[0072] 図14は、本実施形態におけるインターフェイス回路の動作例を示すタイミングチャートである。図12に示した構成において、回路モジュール1220が図4に示したタスクAの処理を実行し、回路モジュール1230が図4に示したタスクBの処理を実行し、インターフェイス回路1210及び回路モジュール1220、1230の実行開始間隔が8サイクルである場合の動作を示している。

[0073] 処理の開始を示すスタート信号STARTが“1”になったら、制御回路1211は、ライトレディ信号WRDYを“1”にする。その後、回路モジュール1220でタスクAの処理が実行され、サイクル1にて回路モジュール1220から出力されるライトイネーブル信号WENが“1”となり、ライトアドレスWAにより指定されるメモリ1212の第0のメモリ要素に回

路モジュール1220の出力データが書き込まれる。以降も同様にして、ライトアドレスWAにより指定されるメモリ1212のメモリ要素に回路モジュール1220の出力データが書き込まれていく。

[0074] タスクAの処理開始後（ライトレディ信号WRDYが“1”になった後）、タスクBの処理を開始可能なデータのメモリ1212への書き込みが行われるまでの期間（本例では5サイクル）が経過したサイクル2にて、制御回路1211は、リードレディ信号RRDYを“1”にする。続く、サイクル3にて、“1”のリードレディ信号RRDYを受けた回路モジュール1230から出力されるリードイネーブル信号RENが“1”となり、リードアドレスRA0、RA1により指定されるメモリ1212の第0のメモリ要素、第1のメモリ要素からデータが読み出され、マルチプレクサ1214を介して回路モジュール1230に出力される。以降も同様にして、リードアドレスRA0、RA1により指定されるメモリ1212のメモリ要素からデータが読み出され、マルチプレクサ1214を介して回路モジュール1230に出力される。

[0075] また、サイクル3においては、制御回路1211は、FIFOライトイネーブル信号FWENを“1”にする。これにより、メモリ1212の第0のメモリ要素から読み出されたデータが、マルチプレクサ1215を介してFIFOバッファ1213に書き込まれる。そして、リードアドレスRA1としてメモリ1212の第0のメモリ要素が再び指定されるサイクル10にて、制御回路1211は、FIFOリードイネーブル信号FREN及び制御信号SEL2とともに“1”にする。これにより、サイクル3においてFIFOバッファ1213に書き込まれたデータが、メモリ1212の第0のメモリ要素のデータとして、マルチプレクサ1214を介して回路モジュール1230に出力される。

[0076] インターフェイス回路1210及び回路モジュール1220、1230の実行開始間隔が8サイクルであるので、以降は8サイクル毎に同じように各

信号が制御され処理が実行される。

[0077] なお、制御回路 1 2 1 1 は、ライトレディ信号 WRDY が “1” になってから回路モジュール 1 2 2 0（タスク A）の実行開始間隔分のサイクルが経過したらライトレディ信号 WRDY を “0” にする。また、制御回路 1 2 1 1 は、ライトレディ信号 WRDY が “0” になってから（インターフェイス回路 1 2 1 0 の実行開始間隔－回路モジュール 1 2 2 0 の実行開始間隔）分のサイクルが経過したらライトレディ信号 WRDY を “1” にする。しかし、本例ではインターフェイス回路 1 2 1 0 の実行開始間隔と回路モジュール 1 2 2 0 の実行開始間隔とが同じであるので、ライトレディ信号 WRDY は “1” のままである。

[0078] また、制御回路 1 2 1 1 は、リードレディ信号 RRDY が “1” になってから回路モジュール 1 2 3 0（タスク B）の実行開始間隔分のサイクルが経過したらリードレディ信号 RRDY を “0” にする。また、制御回路 1 2 1 1 は、リードレディ信号 RRDY が “0” になってから（インターフェイス回路 1 2 1 0 の実行開始間隔－回路モジュール 1 2 3 0 の実行開始間隔）分のサイクルが経過したらリードレディ信号 RRDY を “1” にする。しかし、本例ではインターフェイス回路 1 2 1 0 の実行開始間隔と回路モジュール 1 2 3 0 の実行開始間隔とが同じであるので、リードレディ信号 RRDY は “1” のままである。

[0079] 図 1 5 は、本実施形態におけるインターフェイス回路の他の構成例を示す図である。図 1 5 において、図 1 2 に示した構成要素と同一の機能を有する構成要素には同一の符号を付している。図 1 2 に示したインターフェイス回路 1 2 1 0 は、マルチプレクサ 1 2 1 5 により、メモリ 1 2 1 2 から出力される第 1 のリードデータ RD 又は回路モジュール 1 2 2 0 から出力されるライトデータ WD の何れかを FIFO バッファ 1 2 1 3 に出力するようにしている。しかし、FIFO バッファ 1 2 1 3 に格納するデータが、タスク B の処理において複数回読み出されるデータである場合には、図 1 5 に示すようにマルチプレクサ 1 2 1 5 を設けない構成としても良い。

[0080] なお、前述した実施形態では、タスクAの処理を実行する回路モジュールにおける最小の実行開始間隔及びタスクBの処理を実行する回路モジュールにおける最小の実行開始間隔の中で大きい方の最小の実行開始間隔を、それぞれタスクA、タスクBの処理を実行する回路モジュール及びインターフェイス回路の実行開始間隔としたが、複数のインスタンスを使用することで実行開始間隔を小さくすることも可能である。例えば、実行開始間隔をインターフェイス回路における最小の実行開始間隔である3サイクルにすることも可能である。

[0081] 図16(A)に示すように、インターフェイス回路における実行開始間隔が3サイクルである場合、タスクAの処理によるインターフェイス回路内メモリへのデータの書き込みは、最大で3回分が並列実行されている。タスクBの処理によるインターフェイス回路へのアクセスも同様である。したがって、図16(B)に示すように、タスクAの処理を実行する回路モジュール1601及びタスクBの処理を実行する回路モジュール1603を、それぞれ3つ設けることで3サイクルの実行開始間隔で処理を行うことができる。

[0082] 図16(B)において、1601-1、1601-2、1601-3は、それぞれがタスクAの処理を実行する回路モジュールであり、1602はインターフェイス回路であり、1603-1、1603-2、1603-3は、それぞれがタスクBの処理を実行する回路モジュールである。タスクAの処理を実行する回路モジュール1601-1、1601-2、1601-3は、それぞれメモリ1604から入力データinを読み出して処理を実行する。

[0083] 回路モジュール1601-1、1601-2、1601-3は、処理結果として得られる出力データを、インターフェイス回路1602を介してタスクBの処理を実行する回路モジュール1603-1、1603-2、1603-3に供給する。回路モジュール1603-1、1603-2、1603-3は、それぞれ回路モジュール1601-1、1601-2、1601-3から出力された出力データを用いて処理を実行し、処理結果をメモリ16

05に出力する。

[0084] このように構成することでタスクAの処理を実行する回路モジュールからタスクBの処理を実行する回路モジュールへのデータの転送に要する時間を短縮することができる。また、データの転送を担うインターフェイス回路のインスタンス数が1であるので、回路面積を小さくできる。

[0085] また、タスクAの処理を実行する回路モジュール、タスクBの処理を実行する回路モジュール、及びインターフェイス回路の実行開始間隔を8サイクルとした場合、インターフェイス回路におけるアクセスタイミングは、前述した図11のようになる。さらに、ライフタイムが重ならないメモリ要素を、メモリアドレスの変換を行うことにより共有させるようにしても良い。メモリ要素を共有させた場合のインターフェイス回路におけるアクセスタイミングの一例を図17に示す。図17に示す例では、第0のメモリ要素、第1のメモリ要素、及び第2のメモリ要素を共有させており、これによりインターフェイス回路内のメモリの容量を8から5に削減することができる。

[0086] 前述した説明では、図4に示した入力記述に記載した処理を実行する回路を合成する場合を例に説明したが、以下では、より一般的な場合について説明する。

[0087] 関数 `schedule(task, resource, t)` の値を、行方向に回路リソースを、列方向に時刻（サイクル）をとって行列表現したときの行列 M の i 行 j 列の要素の値を $M[i, j]$ とする。関数 `schedule(task, resource, t)` は、タスク `task` において、時刻 t に回路リソース `resource` が使われるときに1を返し、使われないときに0を返す関数である。例えば、前述したタスクAのスケジュール情報は、図18に示すような19行11列の行列で表現できる。なお、図18においては値1のみを記載し、値0については記載を省略している。

[0088] タスクの実行開始間隔は、タスクレベルで繰り返し動作するとき、どのくらいの時間間隔でタスクを実行するかを示す指標値である。行列 M で表現されるタスクが

時刻 0 で実行開始し、次に時刻 t で実行開始するとき、各時刻でどの回路リソースが使われるかは、 $M + \text{shift right}(M, t)$ を算出することで求められる。関数 $\text{shift right}(M, t)$ は、行列 M の要素を右方向に t 列シフトした行列を返す関数であり、行列 M で表現されるタスクが時刻 t で実行開始した場合の回路リソースの稼働状況表現する。なお、関数 shift right によってシフトして空いた部分の行列要素の値は 0 とする。つまり、行列 M が i 行 j 列のとき、

[0089] [数5]

$$\text{shiftright}(M, t)[i, j] = \begin{cases} M[i, j - t] & (0 \leq j - t) \\ 0 & (j - t < 0) \end{cases}$$

[0090] となる。また、行列 M と行列要素を右方向にシフトした行列とを加算するとき、列数が異なる行列を加算することになるが、対応する行列要素がないときはその要素の値を 0 として加算する。

[0091] 例えば、前述したタスク A について、実行開始間隔を 11 にする場合を示す $M + \text{shift right}(M, 11)$ は図 19 (A) に示すようになる。また、実行開始間隔を 8 にする場合を示す $M + \text{shift right}(M, 8)$ は図 19 (B) に示すようになり、実行開始間隔を 7 にする場合を示す $M + \text{shift right}(M, 7)$ は図 19 (C) に示すようになる。図 19 (C) においては、要素の値が 2 になっている箇所があり、2 以上の値は回路リソースの割り当てが矛盾しており、実行不可能なスケジューリングであることを意味する。つまり、前述したタスク A について、 $M + \text{shift right}(M, t)$ の値は、 t の値が 7 以下である場合、何れかの行列要素の値が 2 以上となるため、タスク A の実行開始間隔の最小値は 8 となる。

[0092] データの出力側の第 1 のタスクのライトトレースデータを $\text{write}(x) = w_x(x) = 0, 1, 2, \dots, n-1$ 、 w_x は時刻を意味する実数) で表す。 $\text{write}(x) = w_x$ は、第 x のメモリ要素 (メモリアドレス x) に時刻 w_x でデータが

書き込まれることを示

す。また、第1のタスクの出力データを受けるデータの入力側の第2のタスクのリードトレースデータを $read(x) = \{r_x^0, r_x^1, \dots, r_x^{k-1}\}$ ($x = 0, 1, 2, \dots, n-1$, r_x^j は時刻を意味する実数) で表す。 $read(x) = \{r_x^0, r_x^1, \dots, r_x^{k-1}\}$ は、第 x のメモリ要素(メモリアドレス x)から時刻 $r_x^0, r_x^1, \dots, r_x^{k-1}$ にデータが読み出されることを示す。なお、メモリ要素(メモリアドレス)毎に読み出し回数が異なっても良い。また、一般性を失うことなく、 $read(x)$ の右辺 $\{r_x^0, r_x^1, \dots, r_x^{k-1}\}$ は昇順でソートされているものとする。すなわち、 $r_x^j < r_x^{j+1}$ ($0 \leq j \leq k-2$) とする。最小値である r_x^0 を $r_x^{\min}$ とし、最大値である r_x^{k-1} を $r_x^{\max}$ とする。

[0093] 第1のタスクの実行開始時刻を t_1 とし、第2のタスクの実行開始時刻を t_2 としたとき、すべてのメモリ要素(メモリアドレス)のデータについて、第1のタスクによって書き込んだ後に第2のタスクによって読み出すという依存関係を満たす必要がある。これは、第1のタスクから第2のタスクへ正しくデータ転送が行われるために必要な条件である。ライトトレースデータ及びリードトレースデータをそれぞれ時刻と解釈して、 $t_1 + write(x) < t_2 + read(x)$ ($x = 0, 1, 2, \dots, n-1$) となるような下記

のような連立不等式を基に t の最小値を求める。

[0094] [数6]

$$\begin{aligned} t_1 + w_x &< t_2 + r_x^0 \\ t_1 + w_x &< t_2 + r_x^1 \\ &\dots \\ t_1 + w_x &< t_2 + r_x^{k-1} \\ x &= 0, 1, 2, \dots, n-1 \end{aligned}$$

[0095] これは、下記のように簡略化できる。

[0096]

[数7]

$$\begin{aligned} t_1 + w_x &< t_2 + r_x^{\min} \\ x &= 0, 1, 2, \dots, n-1 \end{aligned}$$

[0097] ここで、 $t_2 - t_1 = t$ となる変数 t を導入すると以下ようになる。

[0098] [数8]

$$\begin{aligned} (t_2 - t) + w_x &< t_2 + r_x^{\min} \\ w_x &< t + r_x^{\min} \\ x &= 0, 1, 2, \dots, n-1 \end{aligned}$$

[0099] このときの解となる t の最小値を t_{write} とする。例えば、前述したタスク A とタスク

B の場合、値 t_{write} は 5 となり、便宜的に第 0 のメモリ要素（メモリアドレス 0）に時

刻 0 でデータが書き込まれるとするため、 $t_1 = -3$ とし、 $t_2 = 2$ とする。ま

た、第 x のメモリ要素のライフタイム $\text{lifetime}(x)$ は、 $\text{lifetime}(x) = [t_1$

$+ w_x, t_2 + r_x^{\max}]$ で表される。 $r_x^{\max}$ は、 $\text{read}(x)$ の最大値、すなわち

$\max(\{r_x^0, r_x^1, \dots, r_x^{k-1}\})$ である。 $\text{lifetime}(x) = [$

$t_1 + w_x, t_2$

$+ r_x^{\max}]$ は、第 x のメモリ要素（メモリアドレス x ）のデータは、時刻 $t_1 + w_x$ から時刻 $t_2 + r_x^{\max}$ までの間は保持しておく必要があることを示す。

[0100] ここで、第 1 のタスクの処理を実行する回路モジュールと第 2 のタスクの処理を実行する回路モジュールとの間のインターフェイス回路において、第 x のメモリ要素（メモリアドレス x ）にデータがいつ書き込まれ、いつ読み出されるかを示すトレースデータ $\text{trace}(i f 1 2, x)$ は、 $\text{trace}(i f 1 2, x) = [t_1 + w_x, \{t_2 + r_x^0, t_2 + r_x^1, \dots, t_2 + r_x^{k-1}\}]$ とする。つまり、時刻 t_1 で実行開始する第 1 のタスクのライトトレースデータと、時刻 t_2 で実行開始する第 2 のタスクのリードトレースデータとから構成される。

[0101] インターフェイス回路の処理が繰り返し実行されるときの実行開始間隔を、1回目の処理の実行開始が時刻0であり、2回目の処理の実行開始が時刻 t として求める。このとき、第1のタスクから第2のタスクへ正常にデータの受け渡しが行われるためには、1回目の処理の実行時のすべてのメモリ要素のライフタイムが終了してから2回目の処理の実行時のライフタイムが始まるという順序関係が必要となる。つまり、 $lifetime(x) = [t_1 + w_x, t_2 + r_x^{max}]$ であるから、下記の連立不等式が成立する t の値の最小値が、インターフェイス回路の実行開始間隔の最小値となる。

[0102] [数9]

$$\begin{aligned} t_2 + r_x^{max} &< t + t_1 + w_x \\ x &= 0, 1, 2, \dots, n-1 \end{aligned}$$

[0103] 第1のタスクから第2のタスクへ受け渡すデータに対応する配列変数を $tmp12$ とすると、インターフェイス回路のスケジュール情報は、関数 $schedule(if, tmp12[x], t)$ ($x=0, 1, 2, \dots, n-1$)の値を用いて行列表現できる。なお、 $tmp12$ のサイズは n とする。また、関数 $schedule(if, tmp12[x], t)$ は、例えば図20(A)に示すように、時刻 t で回路リソース $tmp12[x]$ にデータを書き込むときに1を返し、時刻 t で回路リソース $tmp12[x]$ からデータを読み出すときに4を返し、時刻 t で回路リソース $tmp12[x]$ のデータを保持するときに16を返し、それ以外は0を返すものとする。つまり、メモリ要素のライフタイム期間外では、関数 $schedule(if, tmp12[x], t)$ は0を返す。

[0104] 第1のタスクの処理を実行する回路モジュールと第2のタスクの処理を実行する回路モジュールとの間のインターフェイス回路のスケジュール情報を、行方向に回路リソースを、列方向に時刻（サイクル）をとって行列表現したときの行列を $M(if12)$ とする。また、行列 $M(if12)$ の列数を、 $c(M(if12))$ で表す。例えば、前述したタスクA及びタスクB間のインターフェイス回路のスケジュール情報は、図20(B)に示すように

なる。

[0105] インターフェイス回路の処理を繰り返し実行する場合の実行開始間隔は、 $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), t)$ に基づいて求められる。 $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), t)$ を計算した結果の行列要素のそれぞれの値が 0, 1, 4, 16 の何れかであれば、そのスケジュールは実行可能であるが、その他の値（具体的には、2, 5, 8, 17, 20）が含まれる場合、そのスケジュールは実行不可能である。

[0106] まず、 $t = c(M(i f 1 2))$ とする。次に、 $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), t)$ のスケジュールが可能か否かを判定する。その結果、 $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), t)$ のスケジュールが可能であれば、 $t = t - 1$ に更新し、 $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), t)$ のスケジュールが可能か否かを判定する。 $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), t)$ のスケジュールが不可能であると判定するまで、 t の値を 1 つずつ小さくして繰り返す。 $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), t)$ のスケジュールが不可能であると判定したとき、行列要素の値が 2, 5, 8, 17, 20 になった行のメモリ要素（メモリアドレス）を分離するメモリ要素として判定する。

[0107] 例えば、前述したタスク A とタスク B との間のインターフェイス回路において、別のメモリ素子を設けずに $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), 10)$ とした場合を図 21 (A) に、 $M(i f 1 2) + s h i f t r i g h t (M(i f 1 2), 9)$ とした場合を図 21 (B) に示す。したがって、図 21 (B) に示されるように、分離するメモリ要素は $t m p 1 2 [0]$ となる。以下では、第 0 のメモリ要素が分離するメモリ要素であるとして説明する。

[0108] 別のメモリ素子を設けて、インターフェイス回路の実行開始間隔を改善する手法について説明する。インターフェイス回路のスケジュール情報を示す

行列 $M(i f 1 2)$ の第0のメモリ要素に対する行ベクトルを $M(i f 1 2)[x 0]$ と表記する。第0のメモリ要素のデータを格納する別のメモリ要素を新たに導入する。

[0109] 第0のメモリ要素が、書き込まれる時間を $w t 0$ とし、最後に読み出される時刻を $r t 0$ とし、最後から2番目に読み出される時刻を $r t 1$ とする。そして、インターフェイス回路の動作を以下のように変更する。時刻 $r t 1$ で第0のメモリ要素から読み出したデータを、別のメモリ素子 $x 0$ へ書き込み、第0のメモリ要素のライフタイムは終了したとみなす。すなわち、 $l i f e t i m e(x 0)=[w t 0, r t 1]$ である。また、時刻 $r t 0$ で別のメモリ要素 $x 0$ からデータを読み出して第2のタスクに渡す。このように動作するインターフェイス回路内のメモリのスケジュール情報を $M 2(i f 1 2)$ とし、別のメモリ素子のスケジュール情報を $M 2(F I F O(x 0))$ と示す。元の $M(i f 1 2)[x 0]$ と、改善後の $M(i f 1 2)[x 0]$ 、 $M 2(F I F O(x 0))$ のスケジュール情報を図22(A)に示す。

[0110] なお、第0のメモリ要素のデータが、第2のタスクによって1回だけ読み出される場合には、インターフェイス回路の動作を、次のように変更する。時刻 $w t 0$ で第0のメモリ要素にはデータの書き込みを行わず、別のメモリ要素 $x 0$ にデータを書き込む。そして、時刻 $r t 0$ で、別のメモリ要素 $x 0$ からデータを読み出して第2のタスクに出力する。このように動作するインターフェイス回路内のメモリのスケジュール情報を $M 2(i f 1 2)$ とし、別のメモリ素子のスケジュール情報を $M 2(F I F O(x 0))$ と示す。元の $M(i f 1 2)[x 0]$ と、改善後の $M(i f 1 2)[x 0]$ 、 $M 2(F I F O(x 0))$ のスケジュール情報を図22(B)に示す。

[0111] 次に、インターフェイス回路内のメモリと別に設けるメモリ素子のサイズを決定する手法について説明する。以下のように、インターフェイス回路の実行開始間隔を示す L は1以上の整数として、行ベクトルを加算し行列要素の最大値を求める。これで、求められた最大値を16で除算して整数の位へ切り上げた値が別のメモリ素子のサイズである。

[0112] [数10]

$$\sum_{i=0}^{n-1} \text{shiftright}(M2(FIFO(x0)), L \times i)$$

[0113] 前述したタスク A 及びタスク B の例において、インターフェイス回路の実行開始間隔を 3 とすると、図 23 に示す sum 行が行ベクトルの要素の値の加算結果である。したがって、36 を 16 で除算して整数の位へ切り上げた値は 3 となるので、別のメモリ素子のサイズは 3 となる。

[0114] 図 24 にインターフェイス回路に設けた別のメモリ素子を制御する制御回路の状態遷移図を示す。図 24 は、 $p = 0$ として書き出した状態を示しているが、 $p = 0$ と $p = 16$ とは対応しており、状態としては、 $p = \alpha + 16$ 、 $\alpha + 17$ 、 $\alpha + 20$ 、 $\alpha + 32$ ($\alpha = 0, 16, 32, \dots$) の状態を繰り返し遷移することとなる。例えば、 $p = \alpha + 16$ の状態でデータを保持する場合にはその状態を維持し、データの書き込みがあれば 1 を加えた値の状態に遷移する。また、 $p = \alpha + 17$ の状態でデータの読み出しがあれば 3 を加えた値の状態に遷移する。

[0115] 本実施形態によれば、インターフェイス回路の最小の実行開始間隔が回路モジュールの最小の実行開始間隔よりも大きい場合、インターフェイス回路内にメモリとは別のメモリ素子（記憶素子）を設けてインターフェイス回路の実行開始間隔を短縮することで、回路面積の増加を抑えつつ処理性能を向上させることができる。また、スケジュール情報からライトトレースデータ及びリードトレースデータを取得し、ライトトレースデータ及びリードトレースデータに基づいてインターフェイス回路を自動合成するので、設計工数を削減できるとともに、適切な回路面積のインターフェイス回路を設計することができる。

[0116] 図 25 は、本実施形態における情報処理装置を実現可能なコンピュータの機能ブロック図である。図 25 に示すようにコンピュータは、CPU (Central Processing Unit) 2

501、各プログラムのワークエリアとなる主記憶装置 2502、各プログラムやデータベースが記録されるハードディスク等の補助記憶装置 2503

、ディスプレイ、キーボード等の入出力装置（Ｉ／Ｏ）２５０４、ネットワークに接続するためのネットワーク接続装置２５０５、及びディスク、磁気テープ等の可搬記憶媒体から記憶内容を読み出す媒体読取装置２５０６を有し、これらが互いにバス２５０８を介して通信可能に接続される。

[0117] 構成合成や論理合成に係る機能やモジュール間インターフェイス回路の自動合成（スケジュール情報抽出を含む）に係る機能をソフトウェアによって実現した場合、ＣＰＵ２５０１がプログラムに基づいて、主記憶装置２５０２をワークエリアとして、主記憶装置２５０２又は補助記憶装置２５０３からデータを読み出して実現する。図２５に示したコンピュータでは、媒体読取装置２５０６によりディスク、磁気テープ等の記憶媒体２５０７に記憶されているプログラム、データを読み出し、これを主記憶装置２５０２又は補助記憶装置２５０３にダウンロードする。そして本実施形態による各処理は、ＣＰＵ２５０１がこのプログラムを実行することによりソフトウェア的に実現させることができる。

[0118] また、図２５に示したコンピュータでは、ディスク、磁気テープ等の記憶媒体２５０７を用いてアプリケーションソフトの交換が行われる場合がある。よって、本実施形態は、コンピュータにより使用されたときに、前述した本実施形態における機能をコンピュータに行わせるためのプログラムやコンピュータ読み出し可能な記憶媒体２５０７として構成することもできる。記録媒体としては、例えばフレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ、磁気テープ、不揮発性のメモリカード、ＲＯＭ等を用いることができる。

[0119] なお、前記実施形態は、何れも本発明を実施するにあたっての具体化のほんの一例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。すなわち、本発明はその技術思想、またはその主要な特徴から逸脱することなく、様々な形で実施することができる。

前述した本実施形態に関し、さらに以下の付記を開示する。

[0120] (付記 1)

それぞれの入力記述に基づいて、第 1 のタスクの処理を実行する第 1 の回路モジュール、及び前記第 1 の回路モジュールから出力されるデータを受けて第 2 のタスクの処理を実行する第 2 の回路モジュールを高位合成する高位合成部と、

前記データが入出力されるメモリを有して回路モジュール間の前記データの転送を担うインターフェイス回路を、前記インターフェイス回路に対する前記第 1 の回路モジュールによる前記データの書き込み情報及び前記第 2 の回路モジュールによる前記データの読み出し情報に基づいて合成する回路合成部とを有し、

前記回路合成部は、前記データの書き込み情報及び前記データの読み出し情報に基づいて前記インターフェイス回路の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、前記第 1 の回路モジュール及び前記第 2 の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、前記第 1 の回路モジュール及び前記第 2 の回路モジュールの最小の実行開始間隔に基づいて、前記インターフェイス回路内に前記メモリに入出力されるデータを記憶する、前記メモリとは別の記憶素子を設けることを特徴とする情報処理装置。

(付記 2)

前記高位合成部は、前記第 1 の回路モジュールでの処理の流れを示す第 1 のスケジュール情報及び前記第 2 の回路モジュールでの処理の流れを示す第 2 のスケジュール情報を出力し、

前記第 1 のスケジュール情報から前記データの書き込み情報を抽出し、前記第 2 のスケジュール情報から前記データの読み出し情報を抽出する情報抽出部を有することを特徴とする付記 1 記載の情報処理装置。

(付記 3)

前記第 1 のスケジュール情報に基づいて前記第 1 の回路モジュールの最小の実行開始間隔を求め、前記第 2 のスケジュール情報に基づいて前記第 2 の

回路モジュールの最小の実行開始間隔を求めることを特徴とする付記 2 記載の情報処理装置。

(付記 4)

前記別の記憶素子は、FIFOバッファであることを特徴とする付記 1～3 の何れか 1 項に記載の情報処理装置。

(付記 5)

前記回路合成部は、前記データの書き込み情報及び前記データの読み出し情報に基づいて前記メモリが前記データを保持する期間を解析し、解析結果に基づいて前記FIFOバッファのサイズを決定することを特徴とする付記 4 記載の情報処理装置。

(付記 6)

前記別の記憶素子は、レジスタであることを特徴とする付記 1～3 の何れか 1 項に記載の情報処理装置。

(付記 7)

前記データの書き込み情報及び前記データの読み出し情報が指定する前記メモリのメモリ要素のうち、前記データを保持する期間が重ならないメモリ要素を共有させることを特徴とする付記 1～6 の何れか 1 項に記載の情報処理装置。

(付記 8)

前記高位合成部が出力する前記第 1 の回路モジュール及び前記第 2 の回路モジュールの回路情報、及び前記回路合成部が出力する前記インターフェイス回路の回路情報を論理合成する論理合成部を有することを特徴とする付記 1～7 の何れか 1 項に記載の情報処理装置。

(付記 9)

情報処理装置の高位合成部が、それぞれの入力記述に基づいて、第 1 のタスクの処理を実行する第 1 の回路モジュール、及び前記第 1 の回路モジュールから出力されるデータを受けて第 2 のタスクの処理を実行する第 2 の回路モジュールを高位合成し、

前記情報処理装置の回路合成部が、前記データが入出力されるメモリを有して回路モジュール間の前記データの転送を担うインターフェイス回路を、前記インターフェイス回路に対する前記第 1 の回路モジュールによる前記データの書き込み情報及び前記第 2 の回路モジュールによる前記データの読み出し情報に基づいて合成し、

前記インターフェイス回路の合成では、前記データの書き込み情報及び前記データの読み出し情報に基づいて前記インターフェイス回路の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、前記第 1 の回路モジュール及び前記第 2 の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、前記第 1 の回路モジュール及び前記第 2 の回路モジュールの最小の実行開始間隔に基づいて、前記インターフェイス回路内に前記メモリに入出力されるデータを記憶する、前記メモリとは別の記憶素子を設けることを特徴とする情報処理方法。

(付記 10)

それぞれの入力記述に基づいて、第 1 のタスクの処理を実行する第 1 の回路モジュール、及び前記第 1 の回路モジュールから出力されるデータを受けて第 2 のタスクの処理を実行する第 2 の回路モジュールを高位合成する高位合成ステップと、

前記データが入出力されるメモリを有して回路モジュール間の前記データの転送を担うインターフェイス回路を、前記インターフェイス回路に対する前記第 1 の回路モジュールによる前記データの書き込み情報及び前記第 2 の回路モジュールによる前記データの読み出し情報に基づいて合成する回路合成ステップとをコンピュータに実行させ、

前記回路合成ステップにて、前記データの書き込み情報及び前記データの読み出し情報

に基づいて前記インターフェイス回路の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、前記第 1 の回路モジュール及び前記第 2 の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、前記第 1 の回路

モジュール及び前記第２の回路モジュールの最小の実行開始間隔に基づいて、前記インターフェイス回路内に前記メモリに入出力されるデータを記憶する、前記メモリとは別の記憶素子を設ける処理をコンピュータに実行させるためのプログラム。

(付記１１)

それぞれの入力記述に基づいて、第１のタスクの処理を実行する第１の回路モジュール、及び前記第１の回路モジュールから出力されるデータを受けて第２のタスクの処理を実行する第２の回路モジュールを高位合成する高位合成ステップと、

前記データが入出力されるメモリを有して回路モジュール間の前記データの転送を担うインターフェイス回路を、前記インターフェイス回路に対する前記第１の回路モジュールによる前記データの書き込み情報及び前記第２の回路モジュールによる前記データの読み出し情報に基づいて合成する回路合成ステップとをコンピュータに実行させ、

前記回路合成ステップにて、前記データの書き込み情報及び前記データの読み出し情報に基づいて前記インターフェイス回路の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、前記第１の回路モジュール及び前記第２の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、前記第１の回路モジュール及び前記第２の回路モジュールの最小の実行開始間隔に基づいて、前記インターフェイス回路内に前記メモリに入出力されるデータを記憶する、前記メモリとは別の記憶素子を設ける処理をコンピュータに実行させるためのプログラムを記録したことを特徴とするコンピュータ読み取り可能な記録媒体。

符号の説明

- [0121] ２０１ 高位合成部
- ２０２ スケジュール情報抽出部
- ２０３ インターフェイス回路合成部
- ２０４ 論理合成部

- 2 1 1 入力記述
- 2 1 2 R T L 記述
- 2 1 3 ログ情報
- 2 1 4 ライトトレースデータ
- 2 1 5 リードトレースデータ
- 2 1 6 R T L 記述
- 2 1 7 回路データ
- 1 2 1 0 インターフェイス回路
- 1 2 1 1 制御回路
- 1 2 1 2 メモリ
- 1 2 1 3 F I F O バッファ（記憶素子）
- 1 2 1 4、1 2 1 5 マルチプレクサ
- 1 2 2 0、1 2 3 0 回路モジュール

請求の範囲

[請求項1] それぞれの入力記述に基づいて、第1のタスクの処理を実行する第1の回路モジュール、及び前記第1の回路モジュールから出力されるデータを受けて第2のタスクの処理を実行する第2の回路モジュールを高位合成する高位合成部と、

前記データが入出力されるメモリを有して回路モジュール間の前記データの転送を担うインターフェイス回路を、前記インターフェイス回路に対する前記第1の回路モジュールによる前記データの書き込み情報及び前記第2の回路モジュールによる前記データの読み出し情報に基づいて合成する回路合成部とを有し、

前記回路合成部は、前記データの書き込み情報及び前記データの読み出し情報に基づいて前記インターフェイス回路の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、前記第1の回路モジュール及び前記第2の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、前記第1の回路モジュール及び前記第2の回路モジュールの最小の実行開始間隔に基づいて、前記インターフェイス回路内に前記メモリに入出力されるデータを記憶する、前記メモリとは別の記憶素子を設けることを特徴とする情報処理装置。

[請求項2] 前記高位合成部は、前記第1の回路モジュールでの処理の流れを示す第1のスケジュール情報及び前記第2の回路モジュールでの処理の流れを示す第2のスケジュール情報を出力し、

前記第1のスケジュール情報から前記データの書き込み情報を抽出し、前記第2のスケジュール情報から前記データの読み出し情報を抽出する情報抽出部を有することを特徴とする請求項1記載の情報処理装置。

[請求項3] 前記第1のスケジュール情報に基づいて前記第1の回路モジュールの最小の実行開始間隔を求め、前記第2のスケジュール情報に基づいて前記第2の回路モジュールの最小の実行開始間隔を求めることを特

徴とする請求項2記載の情報処理装置。

[請求項4] 前記別の記憶素子は、FIFOバッファであることを特徴とする請求項1～3の何れか1項に記載の情報処理装置。

[請求項5] 前記回路合成部は、前記データの書き込み情報及び前記データの読み出し情報に基づいて前記メモリが前記データを保持する期間を解析し、解析結果に基づいて前記FIFOバッファのサイズを決定することを特徴とする請求項4記載の情報処理装置。

[請求項6] 前記別の記憶素子は、レジスタであることを特徴とする請求項1～3の何れか1項に記載の情報処理装置。

[請求項7] 前記データの書き込み情報及び前記データの読み出し情報が指定する前記メモリのメモリ要素のうち、前記データを保持する期間が重ならないメモリ要素を共有させることを特徴とする請求項1～6の何れか1項に記載の情報処理装置。

[請求項8] 前記高位合成部が出力する前記第1の回路モジュール及び前記第2の回路モジュールの回路情報、及び前記回路合成部が出力する前記インターフェイス回路の回路情報を論理合成する論理合成部を有することを特徴とする請求項1～7の何れか1項に記載の情報処理装置。

[請求項9] 情報処理装置の高位合成部が、それぞれの入力記述に基づいて、第1のタスクの処理を実行する第1の回路モジュール、及び前記第1の回路モジュールから出力されるデータを受けて第2のタスクの処理を実行する第2の回路モジュールを高位合成し、

前記情報処理装置の回路合成部が、前記データが入出力されるメモリを有して回路モジュール間の前記データの転送を担うインターフェイス回路を、前記インターフェイス回路に対する前記第1の回路モジュールによる前記データの書き込み情報及び前記第2の回路モジュールによる前記データの読み出し情報に基づいて合成し、

前記インターフェイス回路の合成では、前記データの書き込み情報及び前記データの読み出し情報に基づいて前記インターフェイス回路

の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、前記第 1 の回路モジュール及び前記第 2 の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、前記第 1 の回路モジュール及び前記第 2 の回路モジュールの最小の実行開始間隔に基づいて、前記インターフェイス回路内に前記メモリに入出力されるデータを記憶する、前記メモリとは別の記憶素子を設けることを特徴とする情報処理方法。

[請求項10] それぞれの入力記述に基づいて、第 1 のタスクの処理を実行する第 1 の回路モジュール、及び前記第 1 の回路モジュールから出力されるデータを受けて第 2 のタスクの処理を実行する第 2 の回路モジュールを高位合成する高位合成ステップと、

前記データが入出力されるメモリを有して回路モジュール間の前記データの転送を担うインターフェイス回路を、前記インターフェイス回路に対する前記第 1 の回路モジュールによる前記データの書き込み情報及び前記第 2 の回路モジュールによる前記データの読み出し情報に基づいて合成する回路合成ステップとをコンピュータに実行させ、

前記回路合成ステップにて、前記データの書き込み情報及び前記データの読み出し情報

に基づいて前記インターフェイス回路の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、前記第 1 の回路モジュール及び前記第 2 の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、前記第 1 の回路モジュール及び前記第 2 の回路モジュールの最小の実行開始間隔に基づいて、前記インターフェイス回路内に前記メモリに入出力されるデータを記憶する、前記メモリとは別の記憶素子を設ける処理をコンピュータに実行させるためのプログラム。

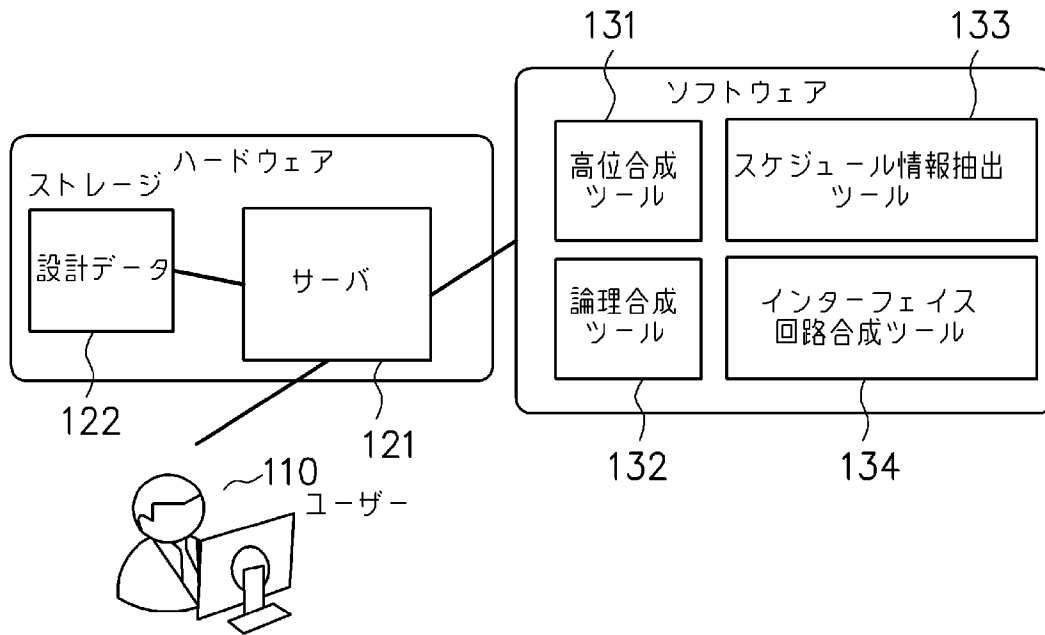
[請求項11] それぞれの入力記述に基づいて、第 1 のタスクの処理を実行する第 1 の回路モジュール、及び前記第 1 の回路モジュールから出力されるデータを受けて第 2 のタスクの処理を実行する第 2 の回路モジュール

を高位合成する高位合成ステップと、

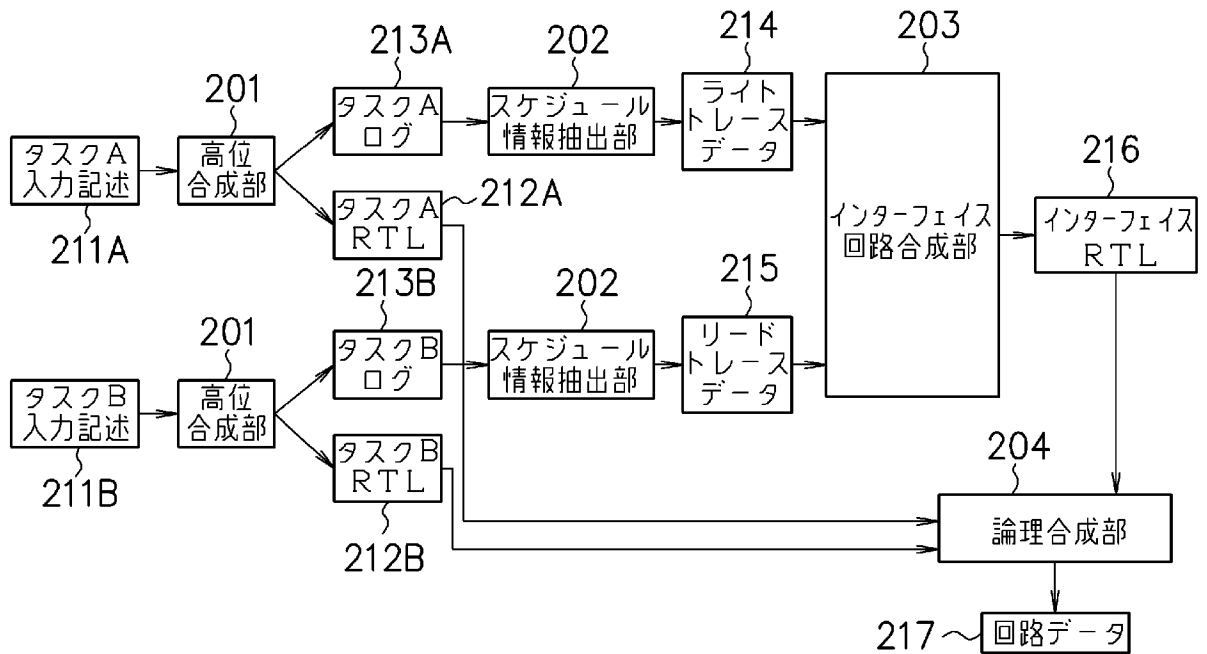
前記データが入出力されるメモリを有して回路モジュール間の前記データの転送を担うインターフェイス回路を、前記インターフェイス回路に対する前記第1の回路モジュールによる前記データの書き込み情報及び前記第2の回路モジュールによる前記データの読み出し情報に基づいて合成する回路合成ステップとをコンピュータに実行させ、

前記回路合成ステップにて、前記データの書き込み情報及び前記データの読み出し情報に基づいて前記インターフェイス回路の最小の実行開始間隔を求め、求めた最小の実行開始間隔が、前記第1の回路モジュール及び前記第2の回路モジュールのそれぞれの最小の実行開始間隔よりも大きい場合、前記第1の回路モジュール及び前記第2の回路モジュールの最小の実行開始間隔に基づいて、前記インターフェイス回路内に前記メモリに入出力されるデータを記憶する、前記メモリとは別の記憶素子を設ける処理をコンピュータに実行させるためのプログラムを記録したことを特徴とするコンピュータ読み取り可能な記録媒体。

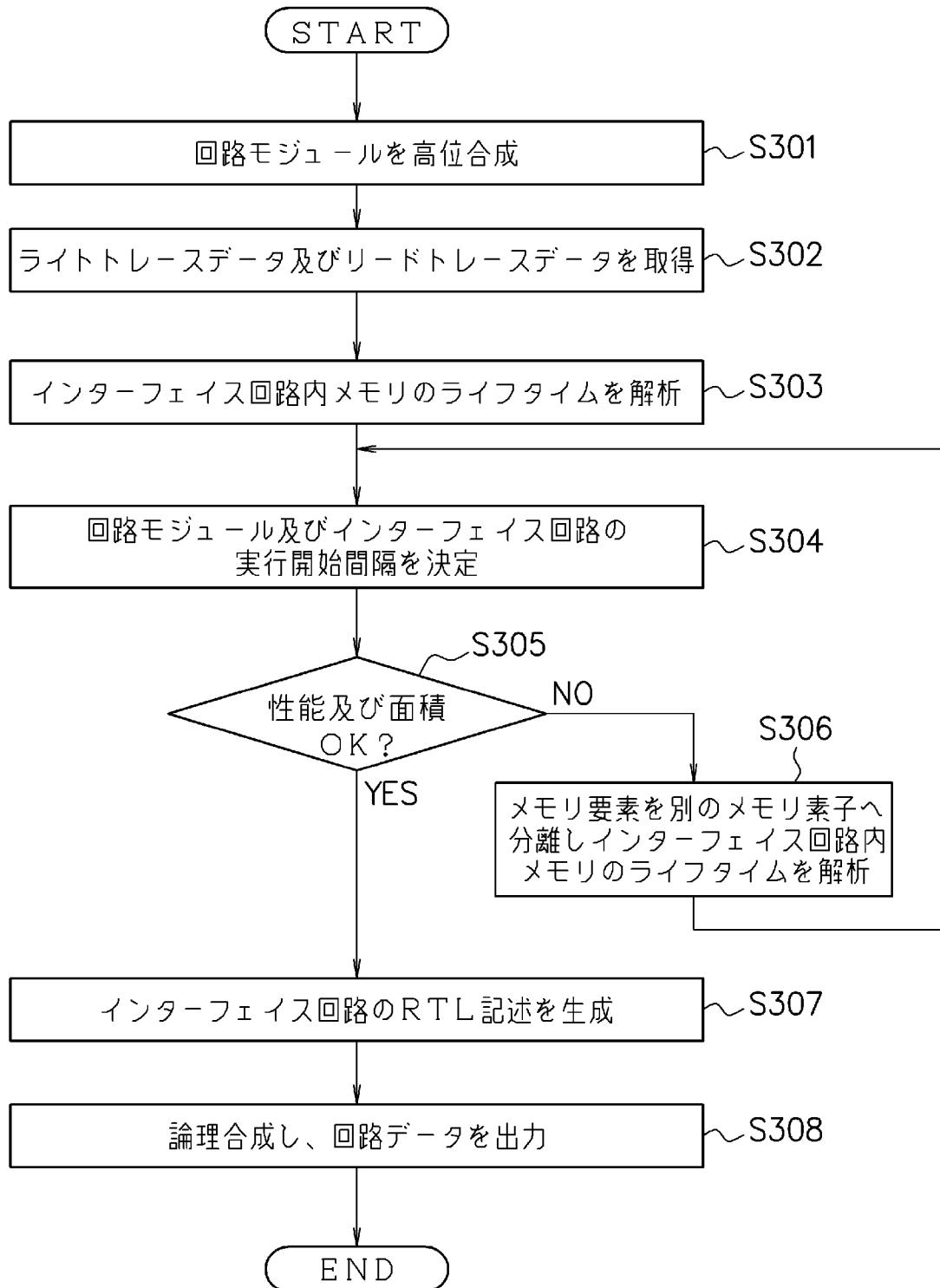
[図1]



[図2]



[図3]



[図4]

```

#include <stdio.h>
#include <math.h>
void target(int in[8], int out[24])
{
    int, j;

    // taskA
    int tmp12[8];
    for (j=0; j<8; j++) {
        tmp12[j]=(in[j]/2)*2;
    }

    // taskB
    int tmp23[8];
    for (j=0; j<8; j++) {
        tmp23[j]=(tmp12[j] + tmp12[(j+1)%8])/2;
    }

    // taskC
    for (j=0; j<8; j++) {
        out[j*3+0] = (tmp23[j] & 0x0000ff);
        out[j*3+1] = (tmp23[j] & 0x00ff00) >> 8;
        out[j*3+2] = (tmp23[j] & 0xff0000) >> 16;
    }
}

int main(int argc, char* argv[])
{
    int in[8] = {11, 23, 37, 47, 59, 67, 71, 87 };
    int out[24];
    target(in, out);
    return 0;
}

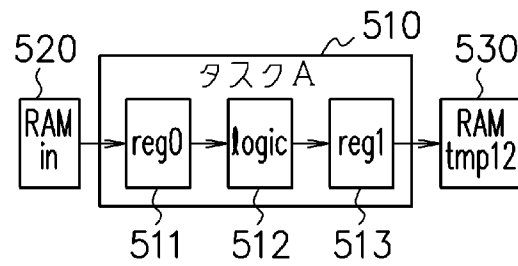
```

401

402

[図5]

(A)



(B)

	time																		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
in [0]	r								r										
in [1]		r								r									
in [2]			r								r								
in [3]				r								r							
in [4]					r								r						
in [5]						r								r					
in [6]							r								r				
in [7]								r								r			
reg0		w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
logic			a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
reg1			w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
tmp12 [0]				w								w							
tmp12 [1]					w								w						
tmp12 [2]						w								w					
tmp12 [3]							w								w				
tmp12 [4]								w								w			
tmp12 [5]									w								w		
tmp12 [6]										w								w	
tmp12 [7]											w								w

[図7]

(A)

タスクA		time							
Mem		0	1	2	3	4	5	6	7
0		w							
1			w						
2				w					
3					w				
4						w			
5							w		
6								w	
7									w

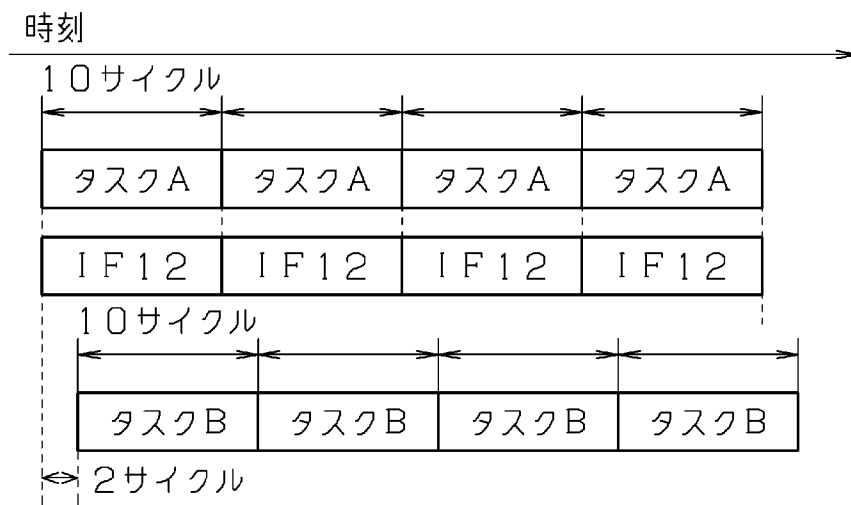
(B)

タスクB		time							
Mem		0	1	2	3	4	5	6	7
0		r							r
1		r	r						r
2			r	r					
3				r	r				
4					r	r			
5						r	r		
6							r	r	
7								r	r

[図8]

		time																				
Mem		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0		w		r							r	w		r							r	w
1			w	r	r								w	r	r							
2				w	r	r								w	r	r						
3					w	r	r								w	r	r					
4						w	r	r								w	r	r				
5							w	r	r								w	r	r			
6								w	r	r								w	r	r		
7									w	r	r								w	r	r	

[図9]



[図10]

(A)

		time									
Mem		0	1	2	3	4	5	6	7	8	9
0		w		r							r
1			w	r	r						
2				w	r	r					
3					w	r	r				
4						w	r	r			
5							w	r	r		
6								w	r	r	
7									w	r	r

(B)

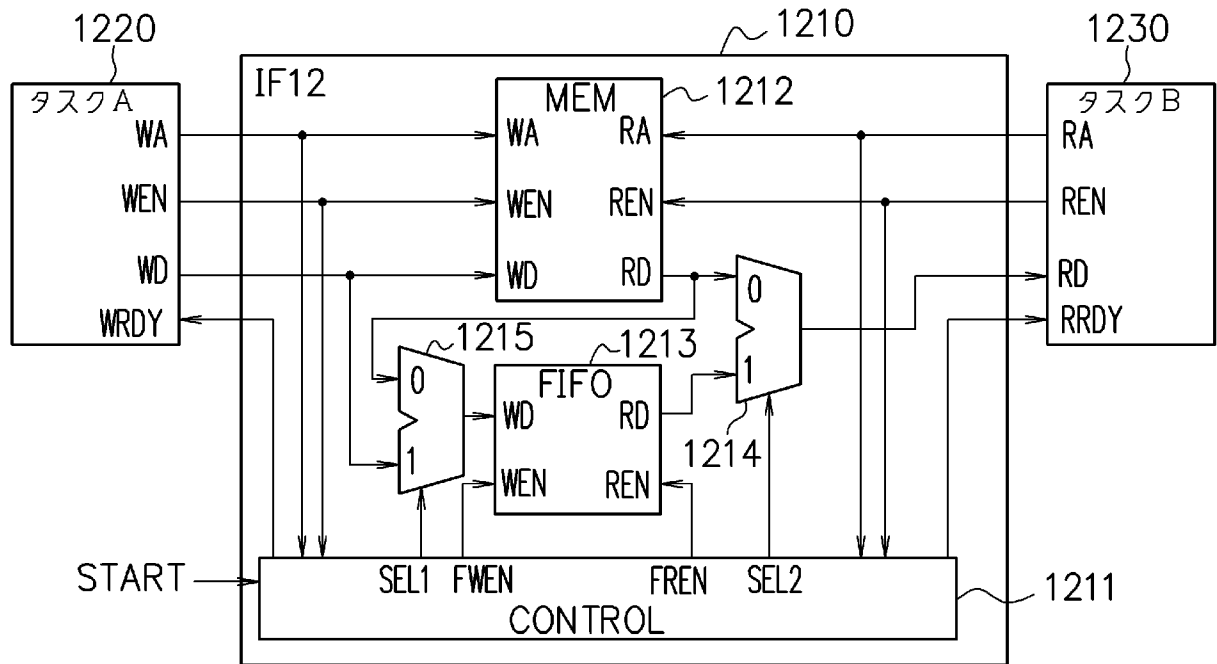
		time									
Mem		0	1	2	3	4	5	6	7	8	9
0		w		r							
1			w	r	r						
2				w	r	r					
3					w	r	r				
4						w	r	r			
5							w	r	r		
6								w	r	r	
7									w	r	r
x0				w							r

(C)

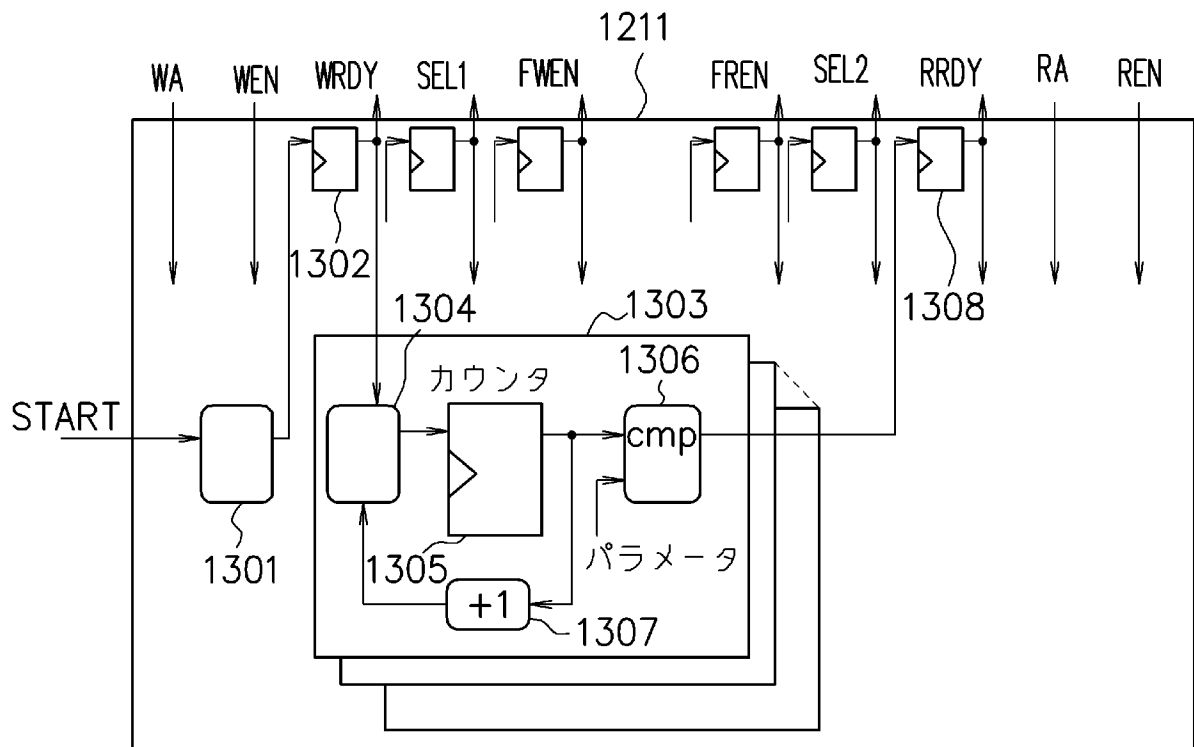
		time																		
Mem		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0		w		r	w		r	w		r	w		r							
1			w	r	r	w	r	r	w	r	r	w	r	r						
2				w	r	r	w	r	r	w	r	r	w	r	r					
3					w	r	r	w	r	r	w	r	r	w	r	r				
4						w	r	r	w	r	r	w	r	r	w	r	r			
5							w	r	r	w	r	r	w	r	r	w	r	r		
6								w	r	r	w	r	r	w	r	r	w	r	r	
7									w	r	r	w	r	r	w	r	r	w	r	r
x0				w							r									
						w								r						
								w								r				
										w								r		
												w								r

[illegible]

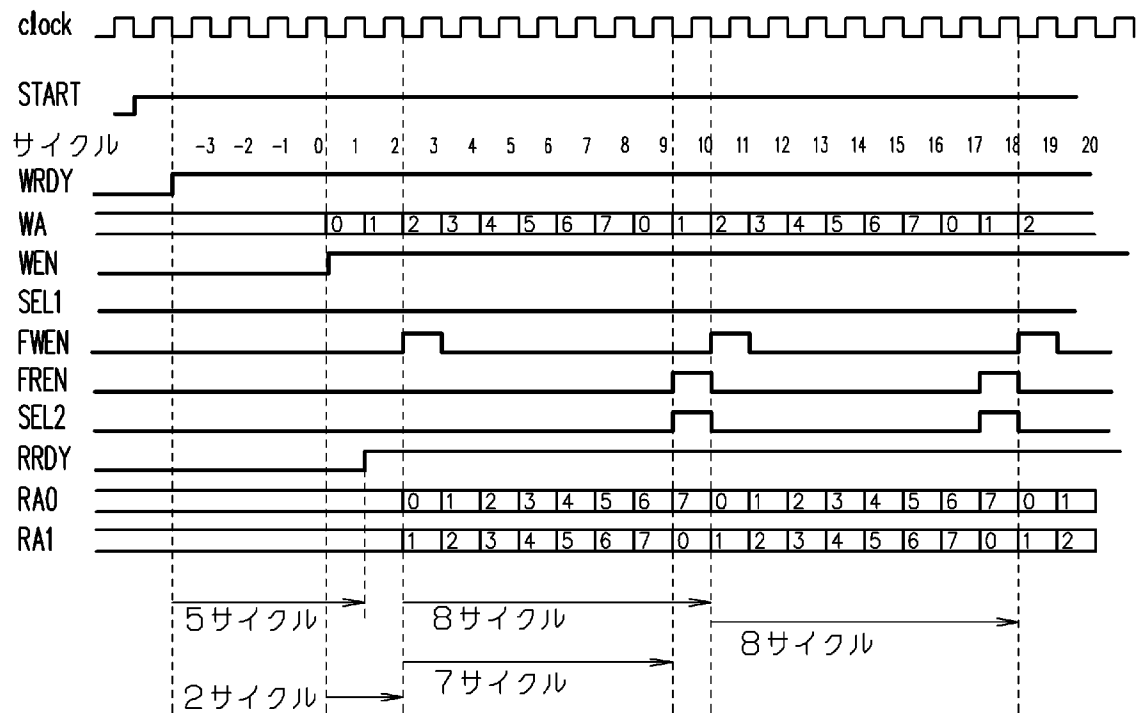
[図12]



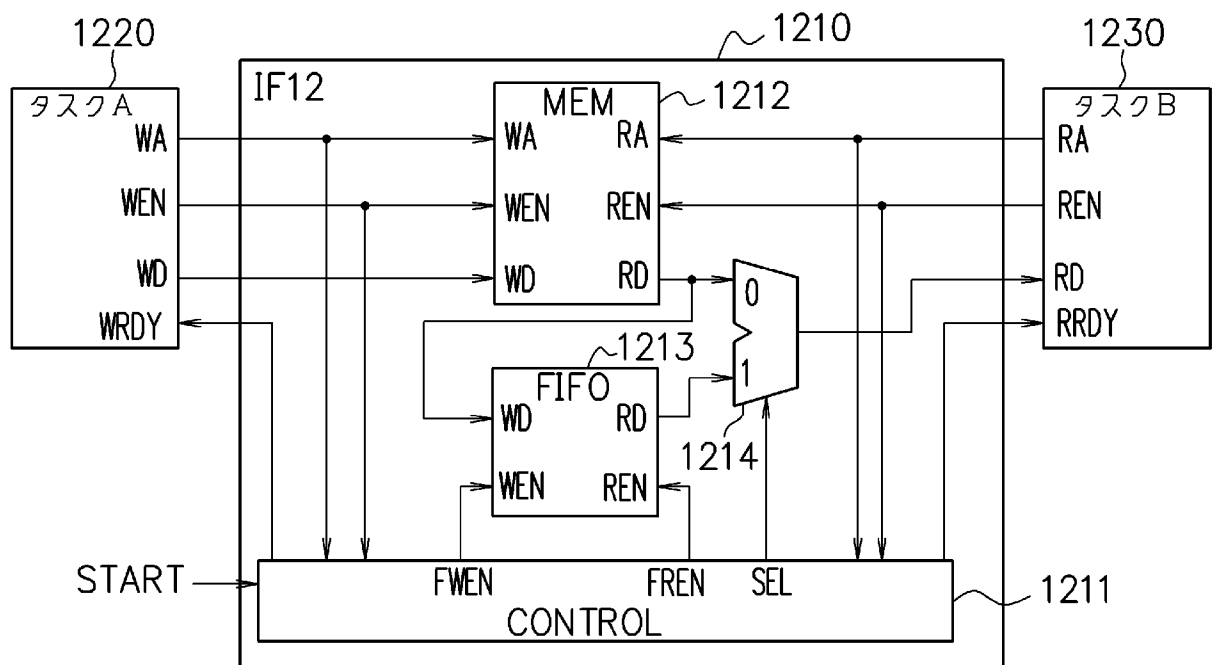
[図13]



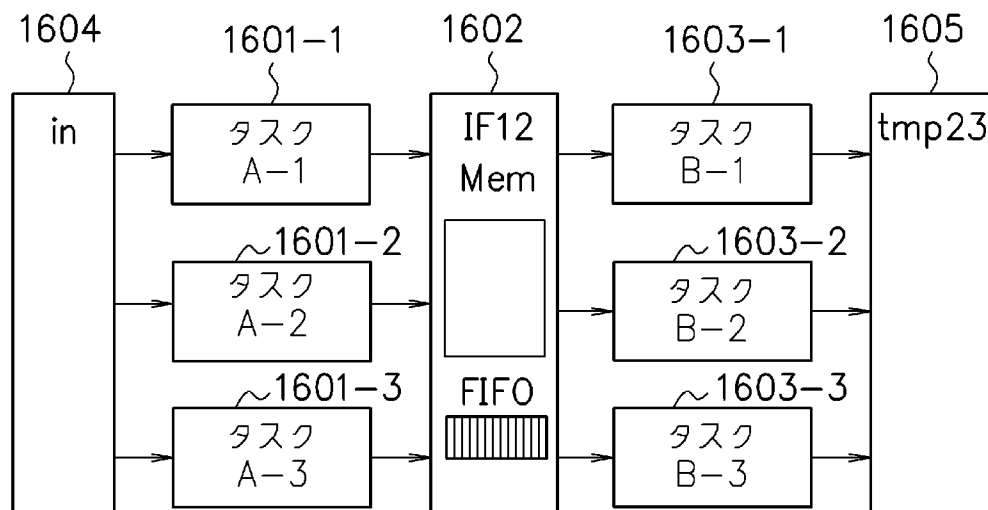
[図14]



[図15]



(A)



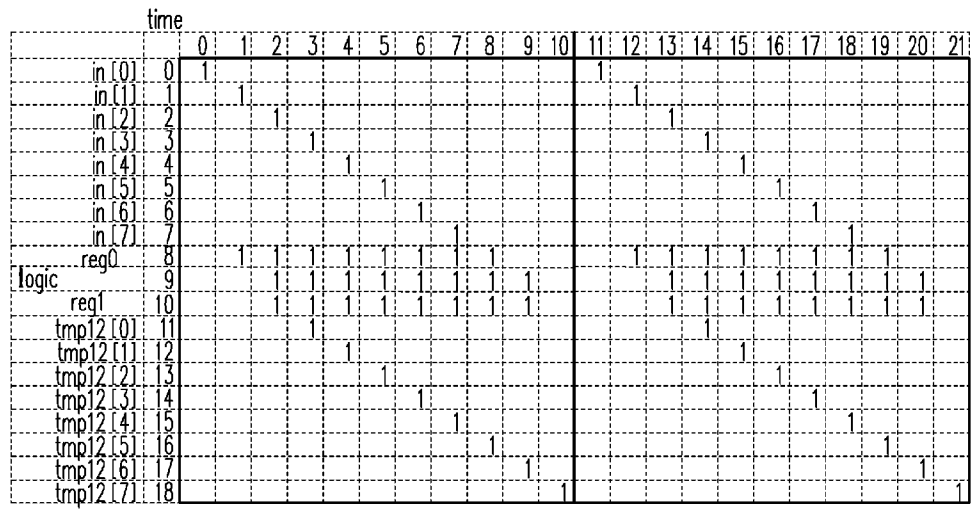
[図17]

[illegible]

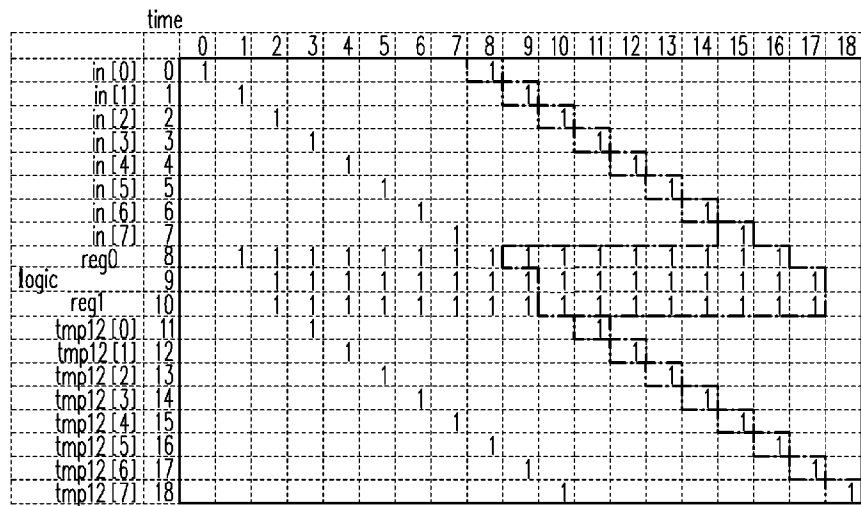
[illegible]

[図19]

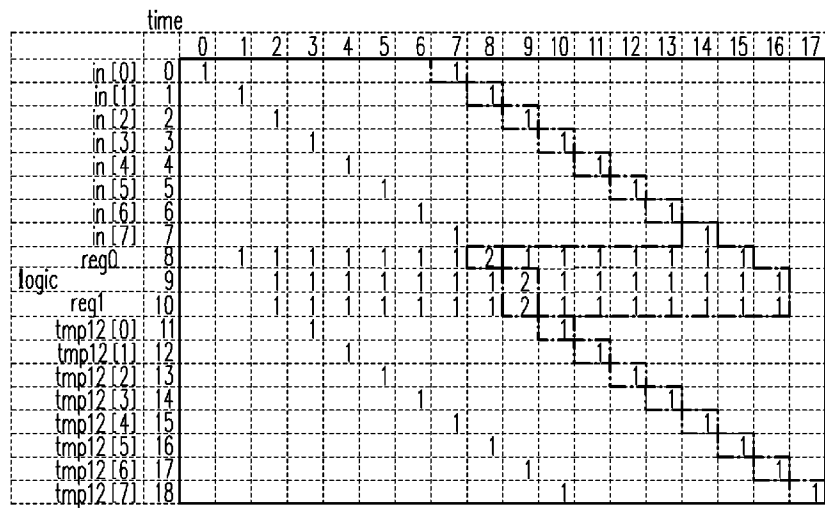
(A)



(B)



(C)



[図20]

(A)

time

	0	1	...	t	...
tmp[0]	1	16		4	
tmp[1]	0	1		16	
...					
tmp[n-1]	0	0		0	

(B)

time

tmp12	0	1	2	3	4	5	6	7	8	9
0	1	16	4	16	16	16	16	16	16	4
1		1	4	4						
2			1	4	4					
3				1	4	4				
4					1	4	4			
5						1	4	4		
6							1	4	4	
7								1	4	4

[図21]

(A)

$M(\text{if12}) + \text{shiftright}(M(\text{if12}), 10)$

time

tmp12	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	1	16	4	16	16	16	16	16	16	4	1	16	4	16	16	16	16	16	16	4
1		1	4	4								1	4	4						
2			1	4	4								1	4	4					
3				1	4	4								1	4	4				
4					1	4	4								1	4	4			
5						1	4	4								1	4	4		
6							1	4	4								1	4	4	
7								1	4	4								1	4	4

(B)

$M(\text{if12}) + \text{shiftright}(M(\text{if12}), 9)$

time

tmp12	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	1	16	4	16	16	16	16	16	16	5	16	4	16	16	16	16	16	16	4	
1		1	4	4							1	4	4							
2			1	4	4							1	4	4						
3				1	4	4							1	4	4					
4					1	4	4							1	4	4				
5						1	4	4							1	4	4			
6							1	4	4							1	4	4		
7								1	4	4							1	4	4	

[図22]

(A)

	time															
	0			wt0					rt0					rt1		
M (if12) [x0]	0	...	0	1	16	...	16	~	4	16	...	16		4	0	...
M2 (if12) [x0]	0	...	0	1	16	...	16	~	4	0	...	0		0	0	...
M2 (FIFO (x0))	0	...	0	0	0	...	0	~	1	16	...	16		4	0	...

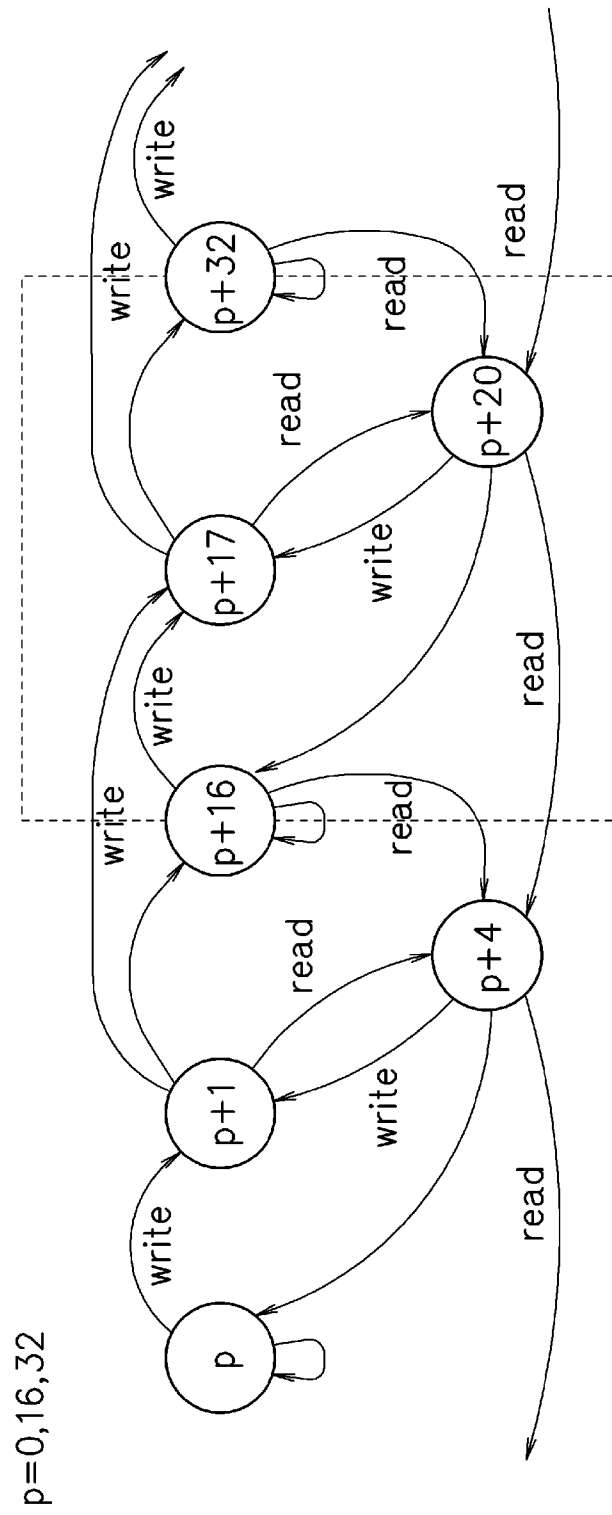
(B)

	time															
	0			wt0					rt0							
M (if12) [x0]	0	...	0	1	16	...	16	~	4	16	...	16				
M2 (if12) [x0]	0	...	0	0	0	...	0	~	0	0	...	0				
M2 (FIFO (x0))	0	...	0	1	16	...	16	~	4	0	...	0				

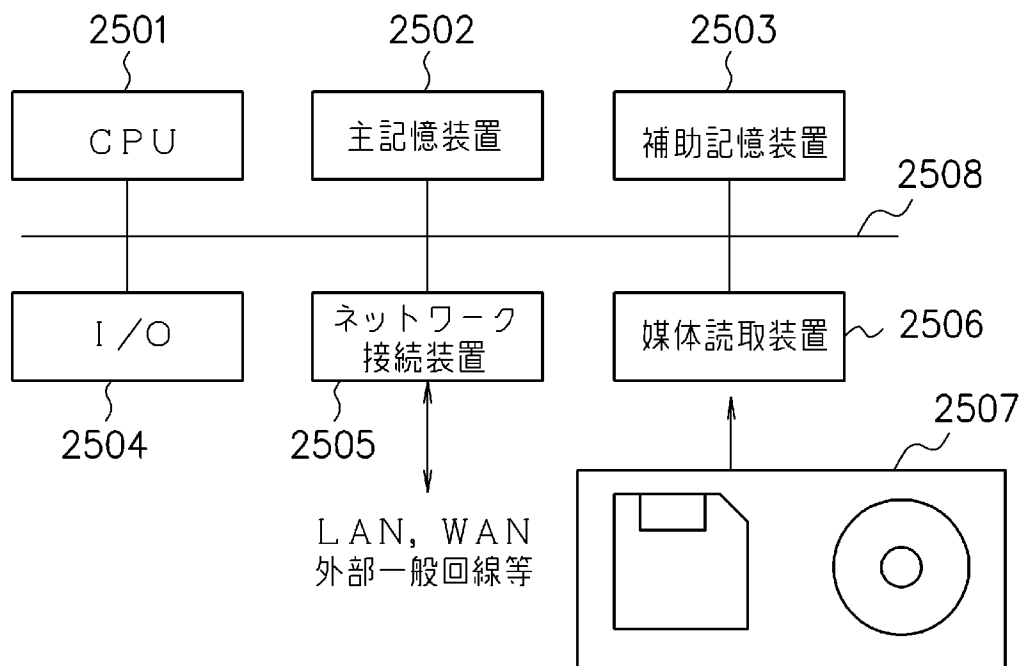
[図23]

x_0	w	r	w	r	w	r	w	r												
$M_2(FIFO(x_0))$	1	16	16	16	16	4														
$shiftright(M_2(FIFO(x_0)), 3)$		1	16	16	16	16	4													
$shiftright(M_2(FIFO(x_0)), 6)$				1	16	16	16	4												
$shiftright(M_2(FIFO(x_0)), 9)$					1	16	16	16	4											
$shiftright(M_2(FIFO(x_0)), 12)$						1	16	16	16	4										
$shiftright(M_2(FIFO(x_0)), 15)$							1	16	16	16	4									
sum	0	1	16	16	17	32	32	33	36	32	33	36	32	33	36	32	20	16	16	4

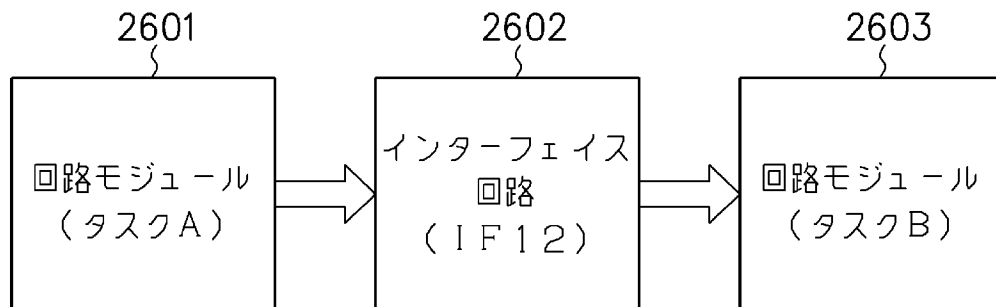
[図24]



[図25]



[図26]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2018/002091

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06F17/50 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. G06F17/50

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan	1922-1996
Published unexamined utility model applications of Japan	1971-2018
Registered utility model specifications of Japan	1996-2018
Published registered utility model applications of Japan	1994-2018

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

IEEE Xplore

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2013-235474 A (MITSUBISHI ELECTRIC CORP.) 21 November 2013, entire text, all drawings (Family: none)	1-11
A	JP 2006-092561 A (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.) 06 April 2006, entire text, all drawings (Family: none)	1-11
A	JP 2004-054641 A (FUJITSU LTD.) 19 February 2004, entire text, all drawings (Family: none)	1-11

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
27 February 2018 (27.02.2018)

Date of mailing of the international search report
13 March 2018 (13.03.2018)

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

A. 発明の属する分野の分類（国際特許分類（IPC））

Int.Cl. G06F17/50(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（IPC））

Int.Cl. G06F17/50

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2018年
日本国実用新案登録公報	1996-2018年
日本国登録実用新案公報	1994-2018年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

IEEE Xplore

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2013-235474 A（三菱電機株式会社） 2013.11.21, 全文, 全図（ファミリーなし）	1-11
A	JP 2006-092561 A（松下電器産業株式会社） 2006.04.06, 全文, 全図（ファミリーなし）	1-11
A	JP 2004-054641 A（富士通株式会社） 2004.02.19, 全文, 全図（ファミリーなし）	1-11

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

27.02.2018

国際調査報告の発送日

13.03.2018

国際調査機関の名称及びあて先

日本国特許庁（ISA/J P）

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

平野 崇

5M

3657

電話番号 03-3581-1101 内線 3599