

(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
2011 年 10 月 13 日(13.10.2011)

PCT

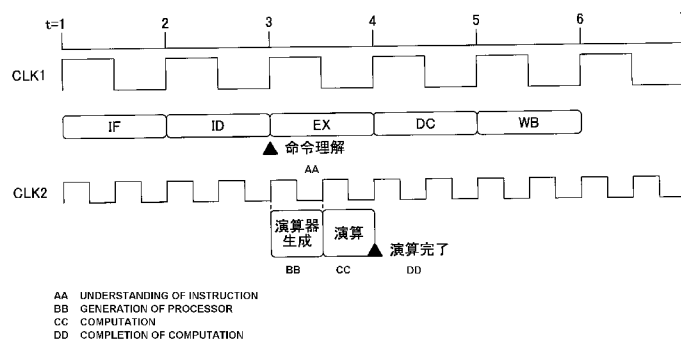
(10) 国際公開番号
WO 2011/125174 A1

- (51) 国際特許分類:
G06F 9/38 (2006.01) G06F 9/30 (2006.01)
- (21) 国際出願番号: PCT/JP2010/056227
- (22) 国際出願日: 2010 年 4 月 6 日(06.04.2010)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人 (米国を除く全ての指定国について): トヨタ自動車株式会社 (TOYOTA JIDOSHA KABUSHIKI KAISHA) [JP/JP]; 〒4718571 愛知県豊田市トヨタ町 1 番地 Aichi (JP).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 磯村 俊郎 (ISOMURA, Toshio) [JP/JP]; 〒4718571 愛知県豊田市トヨタ町 1 番地 トヨタ自動車株式会社内 Aichi (JP). 嵩本 益三 (DAKEMOTO, Masumi) [JP/JP]; 〒4718571 愛知県豊田市トヨタ町 1 番地 トヨタ自動車株式会社内 Aichi (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- (74) 代理人: 伊東 忠彦 (ITO, Tadahiko); 〒1506032 東京都渋谷区恵比寿 4 丁目 2 0 番 3 号 恵比寿ガーデンプレイスタワー 3 2 階 Tokyo (JP).
- 添付公開書類:
— 国際調査報告 (条約第 21 条(3))

(54) Title: DYNAMIC RECONSTRUCTION PROCESSOR AND OPERATING METHOD OF SAME

(54) 発明の名称: 動的再構成プロセッサ及びその動作方法

[図20]



(57) Abstract: Disclosed is a dynamic reconstruction processor for executing, for each instruction, each of a series of steps to execute each instruction, provided with a dynamic construction processor and a clock generation circuit. For each of the series of steps, excluding the instruction execution step for executing an instruction using the dynamic construction processor, the start timing is defined on the basis of a main clock. An instruction execution step for executing an instruction using the dynamic construction processor includes a processor generation sub-step for dynamically constructing a processor corresponding to an instruction by way of the dynamic construction processor, and a computation sub-step for performing computation corresponding to the instruction by way of the processor constructed at the processor generation sub-step. For the processor generation sub-step and the computation sub-step, the start timings are defined on the basis of a sub-clock. For the sub-clock, the processor generation sub-step and the instruction execution sub-step are generated in such a modality as to complete before the start timing of the step immediately following the instruction execution step.

(57) 要約:

[続葉有]



命令毎に一連の各工程を実行して各命令を実行する動的再構成プロセッサであって、動的構成演算器と、クロック生成回路とを備え、一連の各工程は、動的構成演算器を用いて命令を実行する命令実行工程を除き、メインクロックに基づいて開始タイミングが規定され、動的構成演算器を用いて命令を実行する命令実行工程は、命令に対応する演算器を動的構成演算器により動的に構成する演算器生成サブ工程と、演算器生成サブ工程で構成した演算器により命令に対応した演算を行う演算サブ工程とを含み、演算器生成サブ工程及び演算サブ工程は、サブクロックに基づいて開始タイミングが規定され、サブクロックは、演算器生成サブ工程及び命令実行サブ工程が、命令実行工程の直後の工程の開始タイミングよりも前に完了する態様で生成される。

明 細 書

発明の名称： 動的再構成プロセッサ及びその動作方法

技術分野

[0001] 本発明は、命令毎に一連の各工程を実行して各命令を実行する動的再構成プロセッサ及びその動作方法に関する。

背景技術

[0002] 従来から、演算器構成情報を格納する書き換え可能なメモリ（RAM）と、メモリ内の演算器構成情報に基づいて所定の専用演算器を構成する専用演算ユニットとを備える演算処理装置が知られている（例えば、特許文献1参照）。この専用演算器はFPGA（Field Programmable Gate Array）により構成されている。

先行技術文献

特許文献

[0003] 特許文献1：特開平7-175631号公報

発明の概要

発明が解決しようとする課題

[0004] ところで、RISC（Reduced Instruction Set Computer）プロセッサ等では、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）というサイクルで処理が行われており、エクゼキュートは、CPUのハードウェアとして命令毎に用意された演算器を用いて実行される。また、高速化のためにパイプライン処理などが実行されている。

[0005] しかしながら、命令毎に演算器をハードウェアで用意する構成では、ハードウェアの面積が増加するという問題がある。例えば、代表的な命令として、ロード／ストア、整数算術演算／論理演算命令、分岐命令、ビット操作命令等がある。これらの命令のそれぞれには、数種類から数十種類の命令があり、更に、オペランド数に応じた命令や、ワード長に応じた命令を用意して

いる場合もあり、32ビットマイコンでは数百にも及ぶ場合がある。

[0006] これら1つ1つの命令に対して、CPU内部に演算器（ハードウェア）を予め用意しなければならないが、実際には、同時刻に1つの演算器しか動作しておらず、他の演算器は全て停止している。

[0007] この点、特許文献1に記載の構成では、FPGAにより専用演算器を構成することができるので、基本演算ユニット内に用意すべき基本演算器の数を減らすことができ、演算の高速化及び装置の小型化を図ることができる。

[0008] しかしながら、命令に応じて動的にFPGAにより演算器を構成する構成において、命令を遅れ無しで実行するためには、命令に応じて動的にFPGAにより演算器を構成する処理と、該構成した演算器により演算を行う処理とを、データキャッシュのクロックタイミングよりも前に完了させる必要がある。

[0009] そこで、本発明は、命令に応じて動的に演算器を構成する処理と、該構成した演算器により演算を行う処理と遅れの無いタイミングで完了させることができる動的再構成プロセッサ及びその動作方法の提供を目的とする。

課題を解決するための手段

[0010] 上記目的を達成するため、本発明の一局面によれば、命令毎に一連の各工程を実行して各命令を実行する動的再構成プロセッサであって、

命令に対応する演算器を動的に構成する動的構成演算器と、

メインクロック及び前記メインクロックとは別のサブクロックを生成するクロック生成回路とを備え、

前記一連の各工程は、前記動的構成演算器を用いて命令を実行する命令実行工程を除き、前記メインクロックに基づいて開始タイミングが規定され、

前記動的構成演算器を用いて命令を実行する命令実行工程は、命令に対応する演算器を前記動的構成演算器により動的に構成する演算器生成サブ工程と、前記演算器生成サブ工程で構成した演算器により前記命令に対応した演算を行う演算サブ工程とを含み、

前記演算器生成サブ工程及び前記演算サブ工程は、前記サブクロックに基

づいて開始タイミングが規定され、

前記サブクロックは、前記演算器生成サブ工程及び前記命令実行サブ工程が、前記命令実行工程の直後の工程の開始タイミングよりも前に完了する態様で生成されることを特徴とする、動的再構成プロセッサが提供される。

[0011] 本発明のその他の一局面によれば、命令を取り出すフェッチ工程と、取り出した前記命令をデコードするデコード工程と、エクゼキュート工程と、データキャッシュ工程とを含むプロセッサの動作方法であって、

前記エクゼキュート工程は、前記命令に対応する演算器を動的に構成する演算器生成サブ工程及び前記演算器生成サブ工程で構成した演算器により前記命令に対応した演算を行う演算サブ工程を含み、

当該方法は、

前記メインクロックで規定される第1のタイミングで前記フェッチ工程を実行し、

前記メインクロックで規定される第2のタイミングで前記デコード工程を実行し、

前記メインクロックで規定される第3のタイミングに代えて、前記メインクロックとは別のサブクロックで規定される第1のタイミングで、前記演算器生成サブ工程を実行すると共に、前記サブクロックで規定される第2のタイミングで、前記演算サブ工程を実行し、

前記メインクロックで規定される第4のタイミングで前記データキャッシュ工程を実行することを特徴とする、方法が提供される。

発明の効果

[0012] 本発明によれば、命令に応じて動的に演算器を構成する処理と、該構成した演算器により演算を行う処理と遅れの無いタイミングで完了させることができる動的再構成プロセッサ及びその動作方法が得られる。

図面の簡単な説明

[0013] [図1]本発明の実施例1による動的再構成プロセッサ1の構成を概略的に示す図である。

[図2]最小セット演算器 1 1 の設定方法の一例を示す表図である。

[図3]最小セット演算器 1 1 の設定方法のその他の一例を示す表図である。

[図4]最小セット演算器 1 1 の設定方法の更なるその他の一例を示す表図である。

[図5]本実施例による単一の最小セット演算器 1 1 によりシングルスレッド（パイプラインなし）を実現した場合の時系列の一例を示す図である。

[図6]図 5 に対応した最小セット演算器 1 1 の遷移態様を示す図である。

[図7]本実施例による 2 つの最小セット演算器 1 1 A, 1 1 B によりマルチスレッド（2 段のパイプライン）を実現した場合の時系列の一例を示す図である。

[図8]図 7 に対応した最小セット演算器 1 1 A, 1 1 B で構成される演算器の遷移態様を示す図である。

[図9]本実施例による 2 つの最小セット演算器 1 1 A, 1 1 B によりマルチスレッド（5 段のパイプライン）を実現した場合の時系列の一例を示す図である。

[図10]本実施例による 2 つの最小セット演算器 1 1 A, 1 1 B によりスーパー scaler アーキテクチャを実現した場合の時系列の一例を示す図である。

[図11]図 1 0 に対応した最小セット演算器 1 1 A, 1 1 B で構成される演算器の遷移態様を示す図である。

[図12]本発明の実施例 2 による動的再構成プロセッサ 2 の構成を概略的に示す図である。

[図13]本発明のその他の一実施例（実施例 3）による動的再構成プロセッサ 3 の構成を概略的に示す図である。

[図14]CPU 2 2 によりシングルスレッド（パイプラインなし）を実現した場合の時系列の一例を示す図である。

[図15]CPU 2 2 によりマルチスレッド（2 段のパイプライン）を実現した場合の時系列の一例を示す図である。

[図16]CPU 2 2 によりスーパー scaler アーキテクチャを実現した場合の

時系列の一例を示す図である。

[図17]パイプラインがストールするような状況を示す図である。

[図18]パイプラインのストールを防止できる態様での最小セット演算器 1 1 の適用例を示す図である。

[図19]クロック生成回路 1 2 の構成の一例を（第 1 の遅れ防止方法）示す図である。

[図20]図 1 9 に示すクロック生成回路 1 2 により実現される遅れ防止機能の原理を示す図である。

[図21]クロック CLK 1 のみを用いる場合に生じる遅れを示す図である。

[図22]クロック生成回路 1 2 の構成のその他の一例（第 2 の遅れ防止方法）を示す図である。

[図23]図 2 2 に示すクロック生成回路 1 2 により実現される遅れ防止機能の原理を示す図である。

[図24]第 2 の遅れ防止方法単独では遅れを完全に防止できない状況を示す図である。

[図25]第 1 の遅れ防止方法及び第 2 の遅れ防止方法の組み合わせにより実現される遅れ防止機能の原理を示す図である。

発明を実施するための形態

[0014] 以下、図面を参照して、本発明を実施するための最良の形態の説明を行う。

[0015] 図 1 は、本発明の一実施例（実施例 1）による動的再構成プロセッサ 1 の構成を概略的に示す図である。

[0016] 動的再構成プロセッサ 1 は、CPU 1 0 と、クロック生成回路 1 2 とを含む。クロック生成回路 1 2 は、CPU 1 0 の動作に必要な 2 つのクロック CLK 1, CLK 2 を発生する。クロック CLK 1 は、メインクロックである。クロック CLK 2 は、後述の遅れ防止のために生成される特別なクロックである。クロック生成回路 1 2 の構成及びクロック CLK 2 の機能については、後に詳説する。尚、以下の説明において、図 1 8 までの説明までは、ク

ロックという用語は、メインクロックを指す。図 19 以降の説明において、2つのクロック CLK 1, CLK 2 を区別して説明する。

[0017] CPU 10 は、命令実行部（主に演算回路）を構成する最小セット演算器 11 を含む。CPU 10 は、演算回路以外については、命令デコーダ制御回路、命令キャッシュ、レジスタファイル、データキャッシュ等（図示せず）の一般的な構成を含んでよい。CPU 10 には、メモリ（ROM, RAM 等）が接続される。

[0018] 最小セット演算器 11 は、全命令セットに対応する演算器を生成可能な最小限のゲート（又は素子）を備える。全命令セットとは、動的再構成プロセッサ 1 に組み込まれるソフトウェアに含まれる全命令セットであってよいし、汎用性を持たせるために、動的再構成プロセッサ 1 に組み込まれるソフトウェアに含まれる命令以外の命令を含む全命令セットであってもよい。生成可能とは、実際に生成されるか否かを問わない趣旨であり、理論的に生成できる状態をいう。

[0019] 図 2 は、最小セット演算器 11 の設定方法の一例を示す表図である。図 2 に示す例では、最小セット演算器 11 は、全命令セットに対応する演算器を生成可能な最小限のゲートを備える FPG A（Field Programmable Gate Array）からなる。即ち、最小セット演算器 11 は、いわゆる FPG A 合成用のゲートレベルのゲート単位で最小限のゲートを備えるように構成される。FPG A 合成用のゲートは、NAND, NOR, NOT のようなASIC（application specific integrated circuit）論理合成用のゲートに加えて、AND, OR などの複雑なゲート（ASIC 論理合成用のゲートを組み合わせて構成されるゲート）を含む。例えば、AND は、NAND と NOT を組み合わせて構成されるゲートであり、OR は、NOR と NOT を組み合わせて構成されるゲートである。

[0020] 図 2 には、全命令セットに含まれる各命令に対応する各演算器が示されている。例えば、演算器 1C は、16 ビットの桁上げの無い加算命令を実行するための演算器であり、例えば 2 入力の AND ゲートが 30 個、OR ゲート

が20個、NOTゲートが40個、MUXゲートが4個、DFF（Dフリップフロップ）が17個、等から構成されることを意味する。演算器 $C_2, \dots, C_n$ （ n は全命令セットのそれぞれの命令に対応した演算器の数）は、同様に、全命令セット（上記の演算器 C_1 に係る加算命令を除く全命令）のそれぞれの命令に対応した他の演算器を表す。尚、表図に示されている数は、例示的に挙げているだけであり、正しい数でない。

[0021] 図2に示す例では、最小セット演算器 C_1 を構成するに当たり、2入力のANDゲートについては、全命令セットに対応する全演算器 $C_1, \dots, C_n$ のそれぞれを構成するのに必要な数（本例では、30, 20, ..., 25）の最大数（本例では、30）の2入力のANDゲートが用意され、同様に、3入力のANDゲートについては、全命令セットに対応する全演算器 $C_1, \dots, C_n$ のそれぞれを構成するのに必要な数（本例では、0, 20, ..., 15）の最大数（本例では、20）の3入力のANDゲートが用意され、同様に、ORゲートについては、全命令セットに対応する全演算器 $C_1, \dots, C_n$ のそれぞれを構成するのに必要な数（本例では、20, 30, ..., 15）の最大数（本例では、30）のORゲートが用意され、同様に、NOTゲートについては、全命令セットに対応する全演算器 $C_1, \dots, C_n$ のそれぞれを構成するのに必要な数（本例では、40, 30, ..., 20）の最大数（本例では、40）のNOTゲートが用意され、同様に、XORゲートについては、全命令セットに対応する全演算器 $C_1, \dots, C_n$ のそれぞれを構成するのに必要な数（本例では、0, 4, ..., 0）の最大数（本例では、4）のXORゲートが用意され、同様に、MUXゲートについては、全命令セットに対応する全演算器 $C_1, \dots, C_n$ のそれぞれを構成するのに必要な数（本例では、4, 8, ..., 5）の最大数（本例では、8）のMUXゲートが用意され、同様に、DFFゲートについては、全命令セットに対応する全演算器 $C_1, \dots, C_n$ のそれぞれを構成するのに必要な数（本例では、17, 8, ..., 16）の最大数（本例では、17）のDFFゲートが用意され、といった具合に、各ゲートについて、全演算器 $C_1, \dots, C_n$ の任意の1つを生成で

きるようにするのに必要な最小限のゲート数が用意される。

[0022] 図3は、最小セット演算器11の設定方法のその他の一例を示す表図である。図3に示す例では、最小セット演算器11は、FPGA合成用のゲートレベルのゲート単位よりも小さい単位で最小限のゲートを備えるように構成される。具体的には、最小セット演算器11は、いわゆるASIC論理合成用のゲートレベルのゲート単位で最小限のゲートを備えるように構成される。即ち、最小セット演算器11は、NAND、NOR及びNOTのゲート単位で最小限のゲートを備えるように構成される。

[0023] 図3には、図2と同様に、全命令セットに含まれる各命令に対応する各演算器が示されている。図3の表図の見方は、図2と同様である。全命令セットに対応する全演算器C1, ..., Cnのそれぞれに対して、NAND、NOR及びNOTのそれぞれのゲートの必要数が示されている。尚、表図に示されている数は、例示的に挙げているだけであり、正しい数でない。

[0024] 図3に示す例では、図2に示す例と同様に、最小セット演算器11を構成するに当たり、2入力のNANDゲートについては、全命令セットに対応する全演算器C1, ..., Cnのそれぞれを構成するのに必要な数（本例では、30, 20, ..., 25）の最大数（本例では、30）の2入力のNANDゲートが用意され、といった具合に、NAND、NOR及びNOTのそれぞれのゲートについて、全演算器C1, ..., Cnの任意の1つを生成できるようにするのに必要な最小限のゲート数が用意される。

[0025] 図4は、最小セット演算器11の設定方法の更なるその他の一例を示す表図である。尚、表図に示されている数は、例示的に挙げているだけであり、正しい数でない。

[0026] 図4に示す例では、最小セット演算器11は、ASIC論理合成用のゲートレベルのゲート単位よりも更に小さい素子単位で最小限の素子を備えるように構成される。具体的には、最小セット演算器11は、PchMOSFET（Metal-Oxide-Semiconductor Field-Effect Transistor）及びNchMOSFETレベルの素子単位で最小限の素子を備えるように構成される。即ち

、最小セット演算器 11 は、全演算器 $C_1, \dots, C_n$ の任意の 1 つを生成できるようにするのに必要な最小限の PchMOSFET 及び NchMOSFET を備えるように構成される。

[0027] ここで、図 2 に示す例に比べて図 3 に示す例の方が粒度が小さく、図 3 に示す例に比べて図 4 に示す例の方が粒度が小さい。粒度が小さくなるほど、無駄が少なくなる。但し、その反面として、粒度が小さくなるほど、最小セット演算器 11 を用いて後述の演算器を構成するのに要する時間が長くなる。

[0028] このように構成された最小セット演算器 11 は、全命令セットに対応する全演算器 $C_1, \dots, C_n$ のうちの任意の 1 つの演算器を構成することができる。即ち、最小セット演算器 11 は、対応する結線情報に基づいて、ゲート（又は素子）の結線を行うことで、全演算器 $C_1, \dots, C_n$ のうちの任意の 1 つの演算器を構成することができる。結線情報は、全演算器 $C_1, \dots, C_n$ のそれぞれに対応して予め用意され（即ち全命令セットのそれぞれに対応して予め用意され）、メモリに記憶されていてよい。尚、結線情報は、最小セット演算器 11 の最小単位に応じて規定される。例えば、最小セット演算器 11 が図 2 に示した例のように FPG A 合成用のゲート単位を最小単位として構築されている場合は、FPG A 合成用のゲート単位で結線情報（即ち、各 AND、各 OR 等のゲート間の結線態様を表す情報）が生成・記憶される。また、最小セット演算器 11 が図 3 に示した例のように ASIC 論理合成用のゲート単位を最小単位として構築されている場合は、ASIC 論理合成用のゲート単位で結線情報（即ち、各 NAND、各 NOR 及び各 NOT のゲート間の結線態様を表す情報）が生成・記憶される。また、最小セット演算器 11 が図 4 に示した例のように PchMOSFET 及び NchMOSFET の素子単位を最小単位として構築されている場合は、PchMOSFET 及び NchMOSFET の素子単位で結線情報（即ち、各 PchMOSFET のソース・ドレイン及び各 NchMOSFET のソース・ドレイン間の結線態様を表す情報）が生成・記憶される。

- [0029] 図5は、本実施例による単一の最小セット演算器11によりシングルスレッド（パイプラインなし）を実現した場合の時系列の一例を示す図である。図6は、図5に対応した最小セット演算器11で構成される演算器の遷移態様を示す図である。図5において、 $t=4$ 、 $t=9$ は、命令1のIFのクロックを1番目として何番目のクロックかを表し、それぞれの命令1、2に係るデータキャッシュ（DC）のクロックの発生時点を表す。
- [0030] 図5に示すように、本例では、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）というサイクルで処理が実行される。
- [0031] フェッチ（IF）では、命令キャッシュから命令を取り出す。デコード（ID）では、取り出した命令をデコードすると共にレジスタオペランドをフェッチする。エクゼキュート（EX）では、デコード結果とフェッチしたレジスタの値を元に命令（演算等）を実行する。また、ロード／ストア命令の場合は実行アドレスの計算を行い、分岐命令の場合は分岐先のアドレスの計算を行う。但し、エクゼキュート工程は、これらの演算処理に加えて、後述のように最小セット演算器11による演算器生成処理を含む。データキャッシュ（DC）では、エクゼキュート工程で計算したアドレスに対応するメモリの値をデータキャッシュからリードする。ライトバック（WB）では、エクゼキュート工程で計算した結果又はデータキャッシュ工程でフェッチしたオペランドをレジスタに格納する。また、ストア命令の場合はデータキャッシュにライトする。
- [0032] ここでは、一例として、命令1がADD（加算）命令であり、命令2がMUL（乗算）命令であるとする。本実施例では、命令1がフェッチされ、命令1がデコードされると（命令1が理解されると）、命令1（加算）に応じた演算器（加算器）が最小セット演算器11により構成される（図6の命令1の後の加算器参照）。そして、最小セット演算器11により構成された加算器により演算が実行される（即ち、命令1が実行される）。この最小セット演算器11による加算器の結線及び構成された加算器による演算は、命令

1に係るDCのクロックの発生時（ t_4 ）までに完了するように構成される（この構成の詳細は、後述する）。命令1が実行されると、演算結果がレジスタに格納され、命令1に対する処理が終了する。

[0033] 命令1に対する処理が終了すると、命令2がフェッチされ、命令2がデコードされると（命令2が理解されると）、命令2（乗算）に応じた演算器（乗算器）が最小セット演算器11により構成される（図6の命令2の後の乗算器参照）。そして、最小セット演算器11により構成された乗算器により演算が実行される（即ち、命令2が実行される）。この最小セット演算器11による乗算器の結線及び構成された乗算器による演算は、命令2に係るDCのクロックの発生時（ t_9 ）までに完了するように構成される（この構成の詳細は、後述する）。命令2が実行されると、演算結果がレジスタに格納され、命令2に対する処理が終了する。尚、最小セット演算器11の結線は、各命令に対する処理が終了する毎に一旦クリア（リセット）されてもよいし、若しくは、上書き形式で命令毎に変更されてもよい。このようにして、本実施例による単一の最小セット演算器11によるシングルスレッド処理が実行されていく。

[0034] 図7は、本実施例による2つの最小セット演算器11（ここでは、区別のため、それぞれ参照符号11A、11Bを付す）によりマルチスレッド（2段のパイプライン）を実現した場合の時系列の一例を示す図である。図8は、図7に対応した最小セット演算器11A、11Bで構成される演算器の遷移態様を示す図である。図7において、 $t=3$ 、 $t=4$ 、 $t=5$ は、命令1のIFのクロックを1番目として何番目のクロックかを表し、命令1に係るエクゼキュート（EX）のクロックの発生時点、及び、それぞれの命令1、2に係るデータキャッシュ（DC）のクロックの発生時点を表す。

[0035] 同様に、本例では、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）というサイクルで処理が実行される。

[0036] ここでは、一例として、命令1がADD（加算）命令であり、命令2がM

UL（乗算）命令であるとする。

[0037] 命令1に関しては、命令1がフェッチされ、命令1がデコードされると（命令1が理解されると）、命令1（加算）に応じた演算器（加算器）が最小セット演算器11Aにより構成される（図8の命令1の後の加算器参照）。そして、最小セット演算器11Aにより構成された加算器により演算が実行される（即ち、命令1が実行される）。この最小セット演算器11Aによる加算器の結線及び構成された加算器による演算は、命令1に係るDCのクロックの発生時（t4）までに完了するように構成される（この構成の詳細は、後述する）。命令1が実行されると、演算結果がレジスタに格納され、命令1に対する処理が終了する。

[0038] 命令2に関しては、命令2がフェッチされ、命令2がデコードされると（命令2が理解されると）、命令2（乗算）に応じた演算器（乗算器）が最小セット演算器11Bにより構成される（図8の命令2の後の乗算器参照）。そして、最小セット演算器11Bにより構成された乗算器により演算が実行される（即ち、命令2が実行される）。この最小セット演算器11Bによる乗算器の結線及び構成された乗算器による演算は、命令2に係るDCのクロックの発生時（t5）までに完了するように構成される（この構成の詳細は、後述する）。命令2が実行されると、演算結果がレジスタに格納され、命令2に対する処理が終了する。このようにして、本実施例による最小セット演算器11A、11Bによるマルチスレッド（2段のパイプライン）処理が実行されていく。

[0039] 尚、マルチスレッドのパイプラインの段数（本数）は、上述の2段に限られず、3段以上の任意の段数であってよい。最小セット演算器11は、パイプラインの段数に応じた個数設けられてもよいが、図9を参照して後述するように最小セット演算器11は、必要最小限の個数が望ましい。

[0040] 図9は、本実施例による2つの最小セット演算器11（ここでは、区別のため、それぞれ参照符号11A、11Bを付す）によりマルチスレッド（5段のパイプライン）を実現した場合の時系列の一例を示す図である。図9に

において、 $t = 1 \sim 9$ は、命令1のIFのクロックを1番目として何番目のクロックかを表す。

[0041] ここでは、一例として、命令1がADD（加算）命令であり、命令2がMUL（乗算）命令であり、命令3がSUB（減算）命令であり、命令4がADD（加算）命令であり、命令5がMUL（乗算）命令であるとする。

[0042] 命令1に関しては、 $t = 1$ で命令1がフェッチされ、命令1がデコードされると（命令1が理解されると）、命令1（加算）に応じた演算器（加算器）が最小セット演算器11Aにより構成される。そして、最小セット演算器11Aにより構成された加算器により演算が実行される（即ち、命令1が実行される）。この最小セット演算器11Aによる加算器の結線及び構成された加算器による演算は、命令1に係るDCのクロックの発生時（ t_4 ）までに完了するように構成される（この構成の詳細は、後述する）。命令1が実行されると、演算結果がレジスタに格納され、命令1に対する処理が終了する。

[0043] 命令2に関しては、 $t = 2$ で命令2がフェッチされ、命令2がデコードされると（命令2が理解されると）、命令2（乗算）に応じた演算器（乗算器）が最小セット演算器11Bにより構成される。そして、最小セット演算器11Bにより構成された乗算器により演算が実行される（即ち、命令2が実行される）。この最小セット演算器11Bによる乗算器の結線及び構成された乗算器による演算は、命令2に係るDCのクロックの発生時（ t_5 ）までに完了するように構成される（この構成の詳細は、後述する）。命令2が実行されると、演算結果がレジスタに格納され、命令2に対する処理が終了する。

[0044] 命令3に関しては、 $t = 3$ で命令3がフェッチされ、命令3がデコードされると（命令3が理解されると）、命令3（減算）に応じた演算器（減算器）が最小セット演算器11Aにより構成される。そして、最小セット演算器11Aにより構成された減算器により演算が実行される（即ち、命令3が実行される）。この最小セット演算器11Aによる減算器の結線及び構成され

た減算器による演算は、命令 3 に係る DC のクロックの発生時 (t 6) までに完了するように構成される (この構成の詳細は、後述する)。命令 3 が実行されると、演算結果がレジスタに格納され、命令 3 に対する処理が終了する。ここで、命令 3 に関しては、命令 1 に関して使用された最小セット演算器 1 1 A が減算器を構成するのに使用されている。これは、命令 3 のデコード (ID) の処理が完了するまでには、命令 1 のエクゼキュート (EX) は完了しており、命令 1 に関して使用された最小セット演算器 1 1 A が解放された状態 (利用可能な状態) となっているためである。

[0045] 命令 4 に関しては、t = 4 で命令 4 がフェッチされ、命令 4 がデコードされると (命令 4 が理解されると)、命令 4 (加算) に応じた演算器 (加算器) が最小セット演算器 1 1 B により構成される。そして、最小セット演算器 1 1 B により構成された加算器により演算が実行される (即ち、命令 4 が実行される)。この最小セット演算器 1 1 B による加算器の結線及び構成された加算器による演算は、命令 4 に係る DC のクロックの発生時 (t 7) までに完了するように構成される (この構成の詳細は、後述する)。命令 4 が実行されると、演算結果がレジスタに格納され、命令 4 に対する処理が終了する。同様に、この命令 4 に関しては、命令 2 に関して使用された最小セット演算器 1 1 B が加算器を構成するのに使用されている。これは、命令 4 のデコード (ID) の処理が完了するまでには、命令 2 のエクゼキュート (EX) は完了しており、命令 2 に関して使用された最小セット演算器 1 1 B が解放された状態 (利用可能な状態) となっているためである。

[0046] 同様にして命令 5 に関しては、命令 1 及び命令 3 に関して使用された最小セット演算器 1 1 A を利用して演算器を構成し、演算が実行される。

[0047] 尚、図 9 に示す例では、5 段のパイプラインのマルチスレッドに対して、2 つの最小セット演算器 1 1 A、1 1 B を命令の順に交互に使用することで、演算器不足によるパイプラインのストールの発生を防止しつつ、ハードウェアリソースの低減を図っている。しかしながら、5 段のパイプラインのマルチスレッドに対して、3 つの最小セット演算器 1 1 又は 4 つの最小セット

演算器 11 を順に周期的に使用することも可能である。このような考え方は、パイプラインの段数に応じて適宜適用可能である。

[0048] 図 10 は、本実施例による 2 つの最小セット演算器 11（ここでは、区別のため、それぞれ参照符号 11A, 11B を付す）によりスーパースcalar（superscalar）（並列実行）を実現した場合の時系列の一例を示す図である。

図 11 は、図 10 に対応した最小セット演算器 11A, 11B で構成される演算器の遷移態様を示す図である。

[0049] 同様に、本例では、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）というサイクルで処理が実行される。ここでは、一例として、命令 1 が ADD（加算）命令であり、命令 2 が ADD（加算）命令であるとする。

[0050] 図 10 に示す例では、命令 1 がフェッチされ、命令 1 がデコードされると（命令 1 が理解されると）、命令 1（加算）に応じた演算器（加算器）が最小セット演算器 11A により構成される（図 11 の命令 1 の後の加算器参照）。命令 1 と同時に、命令 2 がフェッチされ、命令 2 がデコードされると（命令 2 が理解されると）、命令 2（加算）に応じた演算器（加算器）が最小セット演算器 11B により構成される（図 10 の命令 2 の後の加算器参照）。そして、最小セット演算器 11A、11B により構成された加算器のそれぞれにより演算が実行される（即ち、命令 1 及び命令 2 が同時に実行される）。この最小セット演算器 11A、11B による加算器の結線及び構成された加算器による演算は、命令 1 及び命令 2 に係る DC のクロックの発生時（t4）までに完了するように構成される（この構成の詳細は、後述する）。命令 1 及び命令 2 が実行されると、各演算結果がレジスタに格納され、命令 1 及び命令 2 に対する処理が終了する。このようにして、本実施例による最小セット演算器 11A, 11B によるスーパースcalar 処理が実行されている。

[0051] 尚、並列処理の数（並列数）は、上述の 2 の並列数に限られず、3 以上の任意の並列数であってよい。いずれの場合も、最小セット演算器 11 は、並

列数に応じた個数設けられる。これにより、演算器不足によるパイプラインのストールの発生を防止することができる。

[0052] 図 12 は、本発明のその他の一実施例（実施例 2）による動的再構成プロセッサ 2 の構成を概略的に示す図である。

[0053] 本実施例の動的再構成プロセッサ 2 は、CPU 10 及びクロック生成回路 12 に加えて、故障修復用のゲート 20 を備える。CPU 10、特に最小セット演算器 11 の構成や動作例については、上述の実施例 1 と同様であってよい。

[0054] 故障修復用のゲート 20 は、最小セット演算器 11 の一部のゲートに故障が発生した場合に、故障したゲートに代えて使用される。即ち、最小セット演算器 11 の一部のゲートに故障が発生した場合に、故障したゲートを停止し、故障修復用のゲート 20 に接続（結線）を変更することで、動作を継続することができる。尚、ゲートの故障検出方法及びゲートの停止方法は、一般的に故障修復技術で使用される方法が利用されてもよい。

[0055] この目的のため、故障修復用のゲート 20 は、最小セット演算器 11 を構成する全ゲートよりも少ない数のゲートから構成され、最小セット演算器 11 の最小単位に対応する単位のゲートから構成される。例えば、最小セット演算器 11 が図 2 に示した例のように FPG A 合成用のゲート単位を最小単位として構築されている場合は、故障修復用のゲート 20 は、FPG A 合成用のゲート単位で各ゲートを含む。また、最小セット演算器 11 が図 3 に示した例のように ASIC 論理合成用のゲート単位を最小単位として構築されている場合は、故障修復用のゲート 20 は、ASIC 論理合成用のゲート単位で各ゲートを含む。また、また、最小セット演算器 11 が図 4 に示した例のように Pch MOSFET 及び Nch MOSFET の素子単位を最小単位として構築されている場合は、故障修復用のゲート 20 に代えて、Pch MOSFET 及び Nch MOSFET の素子単位で各素子を含む故障修復用の素子が設けられてよい。

[0056] 最小セット演算器 11 が図 2 又は図 3 に示した例のようにゲート単位を最

小単位として構築されている場合は、故障修復用のゲート 20 は、最小セット演算器 11 を構成する全ゲートのうちの所定のゲート（例えば、使用頻度の高いゲート）のみから構成されてもよい。或いは、最小セット演算器 11 が図 2 又は図 3 に示した例のようにゲート単位を最小単位として構築されている場合は、故障修復用のゲート 20 は、最小セット演算器 11 を構成する全種類のゲートを種類毎に一つずつ備えてもよい。

[0057] このように、実施例 2 によれば、ゲートレベルの単位又は素子レベルの単位で故障修復用のゲート 20 又は素子を構成するため、演算器単位で故障修復用の演算器を用意しておく構成に比べて、故障修復用に用意するゲート又は素子の数を低減することができ、小面積で故障修復用の構成を実現することができる。尚、当然ながら、図 12 では説明上、故障修復用のゲート 20 は最小セット演算器 11 とは別に示されているが、最小セット演算器 11 と一体に構成されてもよい（即ち、最小セット演算器 11 に組み込まれてよい）。

[0058] 図 13 は、本発明のその他の一実施例（実施例 3）による動的再構成プロセッサ 3 の構成を概略的に示す図である。

[0059] 本実施例の動的再構成プロセッサ 3 は、CPU 10 及びクロック生成回路 12 に加えて、CPU（演算器）22 を備える。CPU 10、特に最小セット演算器 11 の構成や動作例については、上述の実施例 1 と同様であってよい。

[0060] CPU 22 は、一般的な CPU であってよく、ハードウェアとして複数の演算器（再構成可能でない演算器）を備える。尚、CPU 22 は、CPU 10 と一体に構成されてもよい。即ち、CPU 22 内の複数の演算器（再構成可能でない演算器）が、CPU 10 内の最小セット演算器 11 とは別に、CPU 10 内に組み込まれてもよい。この場合、共有可能なハードウェア（演算器以外のハードウェアであって、例えば命令デコーダ制御回路等）は 1 つにまとめられてもよい。

[0061] 図 14、図 15 及び図 16 は、CPU 22 の各動作例（シングルスレッド

、マルチスレッド、スーパースカラー）をそれぞれ示し、最小セット演算器 11 の同動作例を示す図 5、図 7 及び図 10 とそれぞれ対照となる図である。CPU 22 の各動作は、図 14、図 15 及び図 16 に示すように、一般的な態様であってよい。

[0062] 例えばシングルスレッドの場合は、図 14 に示すように、命令 1（加算命令）がフェッチされ、命令 1 がデコードされると（命令 1 が理解されると）、エクゼキュート（EX）のクロック（ $t=3$ ）にて CPU 22 内の加算器を用いて演算を行う。このようにして命令 1 が実行されると、演算結果がレジスタに格納され、命令 1 に対する処理が終了する。次いで、命令 2（乗算命令）がフェッチされ、命令 2 がデコードされると（命令 2 が理解されると）、エクゼキュート（EX）のクロック（ $t=8$ ）にて CPU 22 内の乗算器を用いて演算を行う。このようにして命令 2 が実行されると、演算結果がレジスタに格納され、命令 2 に対する処理が終了する。このようにして、各種命令に応じて予めハードウェアとして用意されている CPU 22 内の各種演算器を用いて各種演算を行い、シングルスレッド処理が実行されていく。

[0063] マルチスレッドの場合も同様に、図 15 に示すように、各種命令に応じて予めハードウェアとして用意されている CPU 22 内の各種演算器を用いて各種演算が実行される。スーパースカラーの場合も同様に、図 16 に示すように、各種命令に応じて予めハードウェアとして用意されている CPU 22 内の各種演算器を用いて各種演算が実行される。尚、図 14、図 15 及び図 16 では、CPU 22 内の特定種類の演算器が示されているが、実際にはより他の種類の演算器が含まれる。尚、図 16 に示す CPU 22 は、スーパースカラー（並列実行）のために、図 14 及び図 15 に示す CPU 22 よりも多数の演算器を含んでいる。その際、並列数が 2 であることから、図 16 に示す CPU 22 は、図 14 及び図 15 に示す CPU 22 に対して完全に 2 倍の演算器を搭載してもよいが、ある程度多くの演算器が搭載されていてもよい。

[0064] 本実施例 3 の動的再構成プロセッサ 3 は、命令に応じて、最小セット演算

器 1 1 と CPU 2 2 とを使い分けるように構成されている。この使い分けの態様は、任意の態様であってよい。

[0065] 一例として、頻繁に使う命令は、CPU 2 2 内の演算器により実行し、低頻度の命令のみ、最小セット演算器 1 1 により動的に構成される演算器により実行することとしてもよい。これにより、CPU 2 2 により高速演算を維持しつつ、最小セット演算器 1 1 により小面積化を図ることができる。尚、コンパイラに依存するが、実際には、高頻度命令は限られており、面積低減効果が大きく損なわれることはない。高頻度の命令か低頻度の命令かの区分けは、相対的な基準であってよく、高速演算の要求と小面積化の要求との兼ね合いで決定されてもよい。各命令の頻度は、動的再構成プロセッサ 3 が最も利用される用途における命令解析を行うことで判断されてもよい。このようにして、コンパイラ技術と連携してアーキテクチャ設計を行い、コストと速度のバランスを取ることが可能である。

[0066] その他の一例として、ハイプラインのストールが生じるような状況が発生した場合、即ち同一の命令が同時に CPU 2 2 内の演算器以上に発行された場合に（CPU 2 2 内の演算器で賄いきれない数の命令が発行された場合に）、最小セット演算器 1 1 が臨時的に使用されることとしてもよい。即ち、CPU 2 2 は、通常時に処理を行い、CPU 2 2 の演算器で実行できない態様の命令群が発行された場合に、CPU 2 2 の演算器で実行できない命令に応じた演算器が、最小セット演算器 1 1 により動的に構成されてよい。そして、CPU 2 2 の演算器で処理できない命令は、このようにして最小セット演算器 1 1 により構成された演算器により実行される。

[0067] 具体例として、例えば図 1 7 に示すように、加算命令の命令 1, 2, 3 が同時に発行されたが、CPU 2 2 内に加算器が 2 個しかない場合は、本来であれば、命令 3 のパイプラインがストールし、待ち状態が発生する。これに対して、本例では、図 1 8 に示すように、CPU 2 2 が保有している演算器以上の命令が同時に発行されたことが分かった時点で、最小セット演算器 1 1 を用いて加算器を生成し、ストールを回避する。尚、図 1 8 に示す例では

、命令 1 及び命令 2 については、CPU 22 が保有している演算器（2 個の加算器）により実行され、命令 3 については最小セット演算器 11 で構成された加算器により実行される。図 18 に示す例においても、この最小セット演算器 11 による加算器の結線及び構成された加算器による演算は、DC のクロックの発生時（ t_4 ）までに完了するように構成される（この構成の詳細は、後述する）。

[0068] 次に、最小セット演算器 11 による演算器の生成（結線）及び生成された加算器による演算を、遅くとも DC のクロックの発生時（即ち、演算結果を記憶する処理のためのクロック発生時）までに完了させるための構成（特に、クロック生成回路 12 の構成及び機能）について説明する。

[0069] 図 19 は、クロック生成回路 12 の構成の一例（第 1 の遅れ防止方法）を示す図である。クロック生成回路 12 は、発信回路 13 と、第 1 の通倍回路 15 と、第 2 の通倍回路 17 とを含む。発信回路 13 には、外部に設けられる発信子 14 が接続される。尚、発信子 14 は、動的再構成プロセッサ 1, 2, 3 の内部に設けられてもよい。発信回路 13 の出力は、第 1 の通倍回路 15 に接続される。第 1 の通倍回路 15 の出力は、第 2 の通倍回路 17 に接続される。実施例 1, 2 による動的再構成プロセッサ 1, 2 の場合、第 1 の通倍回路 15 の出力は、CPU 10 に接続される。実施例 3 による動的再構成プロセッサ 3 の場合、第 1 の通倍回路 15 の出力は、CPU 10 及び CPU 22 に接続される。

[0070] 第 1 の通倍回路 15 は、典型的には PLL（Phase Locked Loop）で構成され、発信回路 13 で励起したクロック源信号の周波数 f_{org} （内部クロック周波数）を通倍する。即ち、 $f_{PLL1} = d \times f_{org}$ 。ここで、 f_{PLL1} は、第 1 の通倍回路 15 からのクロック CLK 1 の周波数を表し、 d は定数である。尚、第 1 の通倍回路 15 は、低周波数の場合は省略されてもよいが、一般的に、数十 MHz 以上の場合は、発信回路 13 により励起される周波数を通倍して使用するために必要とされる。

[0071] 第 1 の通倍回路 15 の出力は、CPU 10（又は CPU 10 及び CPU 22

2)に入力され、メインクロックであるクロックCLK1として機能する。

[0072] 第2の通倍回路17は、典型的にはPLLで構成され、第1の通倍回路15の出力であるクロックCLK1の周波数を通倍（本例では、2通倍）する。即ち、 $f_{PLL2} = 2 \times f_{PLL1}$ 。これにより、クロックCLK1に同期したクロックCLK2であって、クロックCLK1の周波数の2倍の周波数のクロックCLK2が生成される。クロックCLK2は、CPU10に入力される。尚、第2の通倍回路17は、第1の通倍回路15と並列に設けられてもよい。この場合、第2の通倍回路17は、発信回路13で励起したクロック源信号の周波数 f_{org} （内部クロック周波数）を、第1の通倍回路15に対して2倍の係数 d で通倍する。即ち、 $f_{PLL2} = 2 \times d \times f_{org}$ 。

[0073] 図20は、図19に示すクロック生成回路12により実現される遅れ防止機能（第1の遅れ防止方法）の原理を示す図である。図20には、クロックCLK1の波形と共に1つのサイクル処理（フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB））が時系列で示されている。図20において、 $t = 1 \sim 7$ は、命令1のIFのクロックを1番目として何番目のクロックかを表す。また、図20には、クロックCLK2の波形と共に、最小セット演算器11による演算器の生成処理（演算器生成）及び最小セット演算器11により生成された演算器による演算処理（演算）の各処理のタイミングが示されている。また、図20には、デコード（ID）における命令の理解が完了したタイミングが矢印で示されている。

[0074] フェッチ（IF）、デコード（ID）、データキャッシュ（DC）、ライトバック（WB）の各工程は、クロックCLK1に基づいて実行される。即ち、フェッチ（IF）、デコード（ID）、データキャッシュ（DC）、ライトバック（WB）の各工程は、それぞれ、クロックCLK1の立ち上がりエッジ（ $t = 1, 2, 4, 5$ ）で開始される。

[0075] 他方、本実施例では、エクゼキュート（EX）は、最小セット演算器11による演算器の生成（結線）及び生成された加算器による演算の2つの処理

を含むため、本来であれば、CLK 1の2クロックが必要である。しかしながら、図21に対照的に示すように、エクゼキュート（EX）にCLK 1の2クロックを与えると、その分だけ（CLK 1の1ブロック分だけ）データキャッシュ（DC）、ライトバック（WB）の各工程が遅れることになる。

[0076] そこで、図19及び図20に示す例では、最小セット演算器11による演算器の生成処理（結線情報に基づく結線）、及び、最小セット演算器11により生成された演算器による演算処理は、クロックCLK 1を2通倍したクロックCLK 2に基づいて実行される。これにより、図20に示すように、最小セット演算器11による演算器の生成処理（演算器生成）及び最小セット演算器11により生成された演算器による演算処理（演算）を、データキャッシュ（DC）用のクロックCLK 1の立ち上がりエッジ（ $t=4$ ）までに完了することができる。即ち、通倍クロックを用いて演算器生成及び演算を高速に行うことで、図21に示すような遅れが発生することなく、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）の各工程を実現することができる。

[0077] 尚、以上の図20の説明は、実施例1、2、3による動的再構成プロセッサ1、2、3におけるCPU 10の動作に関する。実施例3による動的再構成プロセッサ3におけるCPU 22の動作は、通常通りであってよい。即ち、動的再構成プロセッサ3におけるCPU 22では、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）の各工程は、通常通り、クロックCLK 1に基づいて実行される。

[0078] 図22は、クロック生成回路12の構成のその他の一例（第2の遅れ防止方法）を示す図である。図22に示すクロック生成回路12は、図19に示した例に対して、第2の通倍回路17に代えて位相調整回路18を備える点が主に異なる。他の構成は同様であってよい。

[0079] 位相調整回路18は、第1の通倍回路15の出力であるクロックCLK 1の位相を所定位相量だけずらしたクロックCLK 2を生成する。所定位相量

は、デコード（ID）の処理に要する時間（実処理時間）の最長時間（考えられうるワースト時間） ΔT に基づいて設定される。所定位相量は、デコード（ID）の最長時間 ΔT （図23参照）よりも長くクロックCLK1の1クロック分の時間よりも短い時間に相当する位相範囲内で決定されてもよい。但し、好ましくは、所定位相量は、最小セット演算器11による演算器の生成処理（演算器生成）が可能な限り早く開始できるように、デコード（ID）の最長時間 ΔT に相当する位相に設定される。ここでは、所定位相量は、デコード（ID）の最長時間 ΔT に相当する位相に設定された場合を想定して説明を続ける。

[0080] 図23は、図22に示すクロック生成回路12により実現される遅れ防止機能（第2の遅れ防止方法）の原理を示す図である。図23には、クロックCLK1の波形と共に1つのサイクル処理（フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB））が時系列で示されている。図23において、 $t=1\sim 7$ は、命令1のIFのクロックを1番目として何番目のクロックかを表す。また、図23には、クロックCLK2の波形と共に、最小セット演算器11による演算器の生成処理（演算器生成）及び最小セット演算器11により生成された演算器による演算処理（演算）の各処理のタイミングが示されている。また、図23には、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）の各工程に要する時間（実処理時間）の最長時間が示されている。また、デコード（ID）における命令の理解が完了したタイミング（最も遅い場合のタイミング）が矢印で示されている。尚、上述のデコード（ID）の最長時間 ΔT は、デコード（ID）用のクロックCLK1の立ち上がりエッジ（ $t=2$ ）から命令の理解が完了したタイミングまでの最長時間である。

[0081] フェッチ（IF）、デコード（ID）、データキャッシュ（DC）、ライトバック（WB）の各工程は、同様に、クロックCLK1に基づいて実行される。他方、図22及び図23に示す例では、最小セット演算器11による

演算器の生成処理（演算器生成）及び最小セット演算器 1 1 により生成された演算器による演算処理（演算）の各処理は、クロック CLK 1 の位相をずらしたクロック CLK 2 に基づいて実行される。即ち、最小セット演算器 1 1 による演算器の生成処理（演算器生成）は、クロック CLK 2 に基づいて、命令の理解が完了したタイミングで実行開始され、従って、デコード（ID）用のクロック CLK 1 の立ち上がりエッジ（ $t=2$ ）の次の立ち上がりエッジ（ $t=3$ ）よりも前に実行開始される。また、最小セット演算器 1 1 により生成された演算器による演算処理（演算）は、次のクロック CLK 2 の立ち上がりエッジで実行開始される。これにより、図 2 3 に示すように、最小セット演算器 1 1 による演算器の生成処理（演算器生成）及び最小セット演算器 1 1 により生成された演算器による演算処理（演算）を、データキャッシュ（DC）用のクロック CLK 1 の立ち上がりエッジ（ $t=4$ ）までに完了することができる。即ち、2 相クロックを用いることで、図 2 1 に示すような遅れが発生することなく、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）の各工程を実現することができる。

[0082] 尚、以上の図 2 3 の説明は、実施例 1、2、3 による動的再構成プロセッサ 1、2、3 における CPU 10 の動作に関する。実施例 3 による動的再構成プロセッサ 3 における CPU 22 の動作は、通常通りであってよい。即ち、動的再構成プロセッサ 3 における CPU 22 では、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）の各工程は、通常通り、クロック CLK 1 に基づいて実行される。後述の図 2 4、2 5 の説明についても同様である。

[0083] ところで、クロック CLK 1 の 1 クロック分の時間（即ち周期）とデコード（ID）の最長時間 ΔT との関係や、最小セット演算器 1 1 による演算器の生成処理（演算器生成）に要する時間、最小セット演算器 1 1 により生成された演算器による演算処理（演算）に要する時間等によっては、上述の第 1 及び第 2 の遅れ防止方法によっては遅れを防止できない場合も生じうる。

かかる場合には、第1及び第2の遅れ防止方法を組み合わせたり、及び／又は、第1の遅れ防止方法において3通倍以上の通倍を行ったりすることで、遅れを防止することができる。

[0084] 例えば、図24に示すように、図23に示す例よりもデコード（ID）の最長時間 ΔT が長い場合には、その分だけクロックCLK1に対するクロックCLK2の位相ずれ量が大きくなり、最小セット演算器11による演算器の生成処理（演算器生成）及び最小セット演算器11により生成された演算器による演算処理（演算）を、データキャッシュ（DC）用のクロックCLK1の立ち上がりエッジ（ $t=4$ ）までに完了することができなくなる。この場合は、例えば図25に示すように、第1及び第2の遅れ防止方法を組み合わせることで、最小セット演算器11による演算器の生成処理（演算器生成）及び最小セット演算器11により生成された演算器による演算処理（演算）を、データキャッシュ（DC）用のクロックCLK1の立ち上がりエッジ（ $t=4$ ）までに完了することができる。

[0085] 以上、本発明の好ましい実施例について詳説したが、本発明は、上述した実施例に制限されることはなく、本発明の範囲を逸脱することなく、上述した実施例に種々の変形及び置換を加えることができる。

[0086] 例えば、上述した実施例では、2つのクロックCLK1、CLK2を用いて、最小セット演算器11による演算器の生成処理（演算器生成）及び最小セット演算器11により生成された演算器による演算処理（演算）を、データキャッシュ（DC）の開始時まで完了させている。しかしながら、3つ以上のクロックを用いてもよい。例えば、クロックCLK1に対して異なる位相ずれた2つのクロックを生成し、それぞれのクロックに基づいて、最小セット演算器11による演算器の生成処理（演算器生成）及び最小セット演算器11により生成された演算器による演算処理（演算）のそれぞれを、実施してもよい。

[0087] また、上述した実施例では、最小セット演算器11によるエクゼキュート（EX）は、最小セット演算器11による演算器の生成処理（演算器生成）

及び最小セット演算器 11 により生成された演算器による演算処理（演算）の 2 つの処理に分離されている。しかしながら、3 つ以上の処理に分解してもよい。例えば、最小セット演算器 11 による演算器の生成処理を、更に、命令に応じた結線情報の読み出し処理と、読み出した結線情報に基づいて最小セット演算器 11 により演算器を生成する処理とに分解してもよい。この場合も、同様に、3 相クロックを用いたり、逡倍クロックを用いたりすることで、エクゼキュート（EX）をデータキャッシュ（DC）の開始時まで完了させることができる。

[0088] また、クロック CLK 1、CLK 2 のそれぞれは、必ずしも常に同一周期であるクロックである必要はなく、上述のような遅れの生じないようなタイミングで各処理のトリガを与えるものであればよい。また、クロック CLK 1 自体が、例えば周波数スプレッタにより周波数を変動されていてもよい。

[0089] また、上述した実施例では、フェッチ（IF）、デコード（ID）、エクゼキュート（EX）、データキャッシュ（DC）、ライトバック（WB）というサイクルで処理が実行されているが、他の形態であってよい。特に、エクゼキュート（EX）の直後の工程は任意である。また、データキャッシュ（DC）及びライトバック（WB）は、エクゼキュート（EX）での演算結果をメモリ又はレジスタファイル等へ書き込む処理であればよい。また、データキャッシュ（DC）は、メモリアクセス（MA又はMEM）と称される場合もあり、呼び名は任意である。

[0090] また、上述した実施例では、好ましい実施例として、全命令セットに対応する演算器を生成可能な最小限のゲート又は素子を備える最小セット演算器 11 が動的構成演算器として使用されているが、最小セット演算器 11 に代えて、最小セット演算器 11 よりも多くのゲート又は素子を備える動的構成演算器が使用されてもよい（図 12 参照）、最小セット演算器 11 よりも少ないゲート又は素子を備える動的構成演算器が使用されてもよい。

符号の説明

[0091] 1, 2, 3 動的再構成プロセッサ

1 0	C P U
1 1	最小セット演算器
1 2	クロック生成回路
1 3	発信回路
1 4	発信子
1 5	第 1 の通倍回路
1 7	第 2 の通倍回路
1 8	位相調整回路
2 0	故障修復用のゲート
2 2	C P U

請求の範囲

- [請求項1] 命令毎に一連の各工程を実行して各命令を実行する動的再構成プロセッサであって、
- 命令に対応する演算器を動的に構成する動的構成演算器と、
- メインクロック及び前記メインクロックとは別のサブクロックを生成するクロック生成回路とを備え、
- 前記一連の各工程は、前記動的構成演算器を用いて命令を実行する命令実行工程を除き、前記メインクロックに基づいて開始タイミングが規定され、
- 前記動的構成演算器を用いて命令を実行する命令実行工程は、命令に対応する演算器を前記動的構成演算器により動的に構成する演算器生成サブ工程と、前記演算器生成サブ工程で構成した演算器により前記命令に対応した演算を行う演算サブ工程とを含み、
- 前記演算器生成サブ工程及び前記演算サブ工程は、前記サブクロックに基づいて開始タイミングが規定され、
- 前記サブクロックは、前記演算器生成サブ工程及び前記命令実行サブ工程が、前記命令実行工程の直後の工程の開始タイミングよりも前に完了する態様で生成されることを特徴とする、動的再構成プロセッサ。
- [請求項2] 前記命令実行工程の直後の工程の開始タイミングは、前記命令実行工程の直前の工程の開始タイミングに対して前記メインクロックの2クロック後に規定される、請求項1に記載の動的再構成プロセッサ。
- [請求項3] 前記サブクロックは、前記メインクロックを逡倍したクロック、前記メインクロックの位相をずらしたクロック、又は、前記メインクロックの位相をずらし且つ逡倍したクロックである、請求項1に記載の動的再構成プロセッサ。
- [請求項4] 前記動的構成演算器は、前記演算器生成サブ工程で生成されうる全ての演算器を生成可能な最小限のゲート又は素子を備える、請求項1

に記載の動的再構成プロセッサ。

[請求項5] 前記動的構成演算器は、前記演算器生成サブ工程で生成されうる全ての演算器を生成可能な最小限のゲート又は素子を備える最小セット演算器からなり、

前記最小セット演算器を用いてシングルスレッドで動作する、請求項1に記載の動的再構成プロセッサ。

[請求項6] 前記動的構成演算器は、前記演算器生成サブ工程で生成されうる全ての演算器を生成可能な最小限のゲート又は素子を備える最小セット演算器を複数個備え、

各最小セット演算器を利用して並列処理又はパイプライン処理を行う、請求項1に記載の動的再構成プロセッサ。

[請求項7] 再構成不能な演算器を更に備え、

命令に応じて、前記動的構成演算器と前記再構成不能な演算器とを使い分け、

前記再構成不能な演算器を用いて命令を実行する命令実行工程は、前記メインクロックに基づいて開始タイミングが規定される、請求項1に記載の動的再構成プロセッサ。

[請求項8] 比較的高い頻度で発生する所定の命令に対しては、前記再構成不能な演算器を用い、比較的低い頻度で発生する命令に対しては、前記動的構成演算器を用いる、請求項7に記載の動的再構成プロセッサ。

[請求項9] 同一命令が、該命令に対応する前記再構成不能な演算器の数よりも多い命令数で同時に発行された場合に、前記再構成不能な演算器の数の命令に対しては、前記再構成不能な演算器を用い、前記再構成不能な演算器の数よりも多い分の命令に対しては、前記動的構成演算器を用いる、請求項7に記載の動的再構成プロセッサ。

[請求項10] 前記動的構成演算器は、前記演算器生成サブ工程で生成されうる全ての演算器を生成可能な最小限のゲート又は素子を備える最小セット演算器からなり、

前記最小セット演算器のゲート又は素子に故障が発生した場合に使用される故障補修用ゲート又は故障補修用素子を更に備える、請求項1に記載の動的再構成プロセッサ。

[請求項11] 前記動的構成演算器は、前記演算器生成サブ工程で生成されうる全ての演算器を生成可能な最小限のゲートを、NAND, NOR, NOTのゲート単位で備える最小セット演算器からなり、

前記演算器生成サブ工程は、前記NAND, NOR, NOTのゲート単位で結線を行うことで、前記命令に対応する演算器を動的に構成する、請求項1に記載の動的再構成プロセッサ。

[請求項12] 前記動的構成演算器は、前記演算器生成サブ工程で生成されうる全ての演算器を生成可能な最小限の素子を、PchMOSFET及びNchMOSFETレベルの素子単位で備える最小セット演算器からなり、

前記演算器生成サブ工程は、前記PchMOSFET及びNchMOSFETレベルの素子単位で結線を行うことで、前記命令に対応する演算器を動的に構成する、請求項1に記載の動的再構成プロセッサ。

[請求項13] 命令を取り出すフェッチ工程と、取り出した前記命令をデコードするデコード工程と、エクゼキューション工程と、データキャッシュ工程とを含むプロセッサの動作方法であって、

前記エクゼキューション工程は、前記命令に対応する演算器を動的に構成する演算器生成サブ工程及び前記演算器生成サブ工程で構成した演算器により前記命令に対応した演算を行う演算サブ工程を含み、

当該方法は、

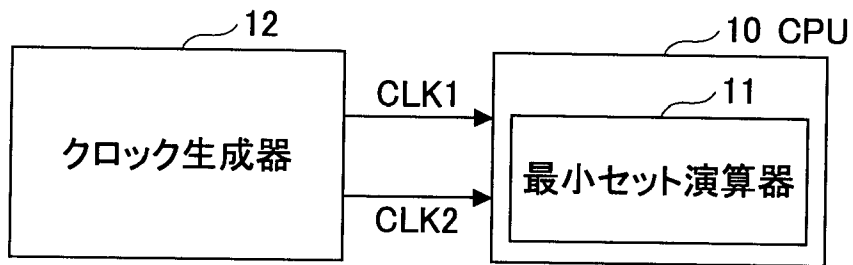
前記メインクロックで規定される第1のタイミングで前記フェッチ工程を実行し、

前記メインクロックで規定される第2のタイミングで前記デコード工程を実行し、

前記メインクロックで規定される第3のタイミングに代えて、前記メインクロックとは別のサブクロックで規定される第1のタイミングで、前記演算器生成サブ工程を実行すると共に、前記サブクロックで規定される第2のタイミングで、前記演算サブ工程を実行し、

前記メインクロックで規定される第4のタイミングで前記データキャッシュ工程を実行することを特徴とする、方法。

[図1]

1

[図2]

命令に対応する演算器		演算器C1	演算器C2	...	演算器Cn
ゲート	属性				
AND	2入力	<u>30</u>	20	...	25
	3入力	0	<u>20</u>	...	15
	...	.	.	...	.
OR	...	20	<u>30</u>	...	15
NOT	...	<u>40</u>	30	...	20
XOR	...	0	<u>4</u>	...	0
MUX	...	4	<u>8</u>	...	5
DFF	...	<u>17</u>	8	...	16
...	...	.	.	...	.

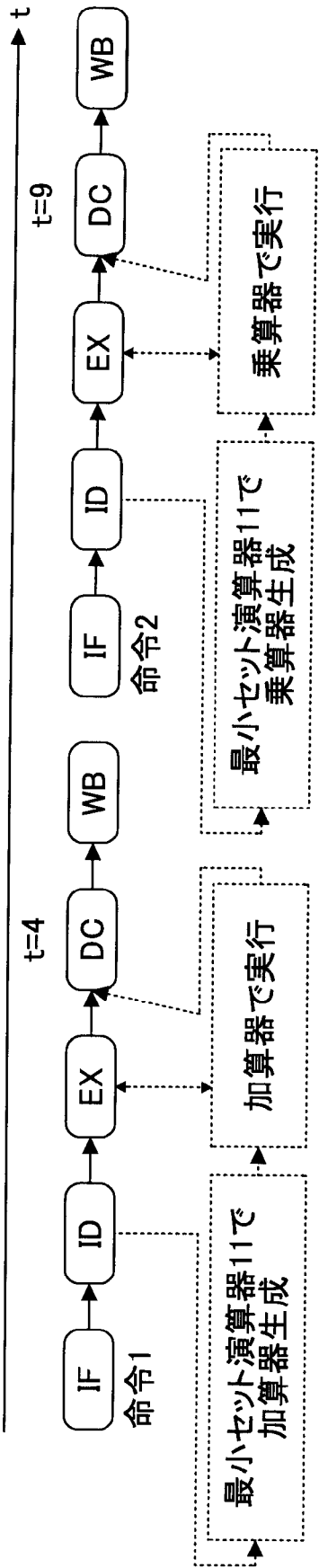
[図3]

命令に対応する演算器		演算器C1	演算器C2	...	演算器Cn
ゲート	属性				
NAND	2入力	30	20	...	25
	3入力	.	.	...	.
	...	.	.	...	.
NOR	...	.	.	...	.
NOT	...	.	.	...	.

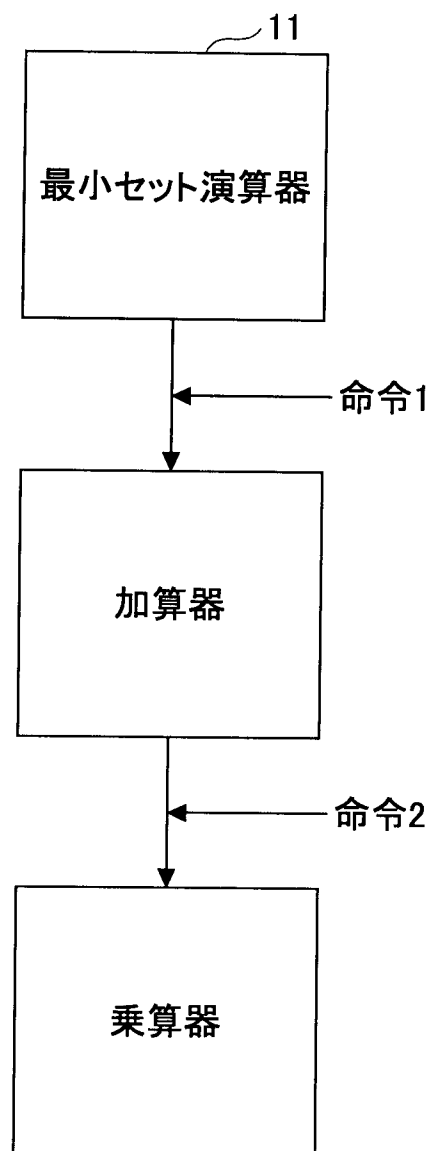
[図4]

命令に対応する演算器		演算器C1	演算器C2	...	演算器Cn
素子	属性				
Pch MOS	...	3000	2000	...	2500
Nch MOS	...	.	.	...	.

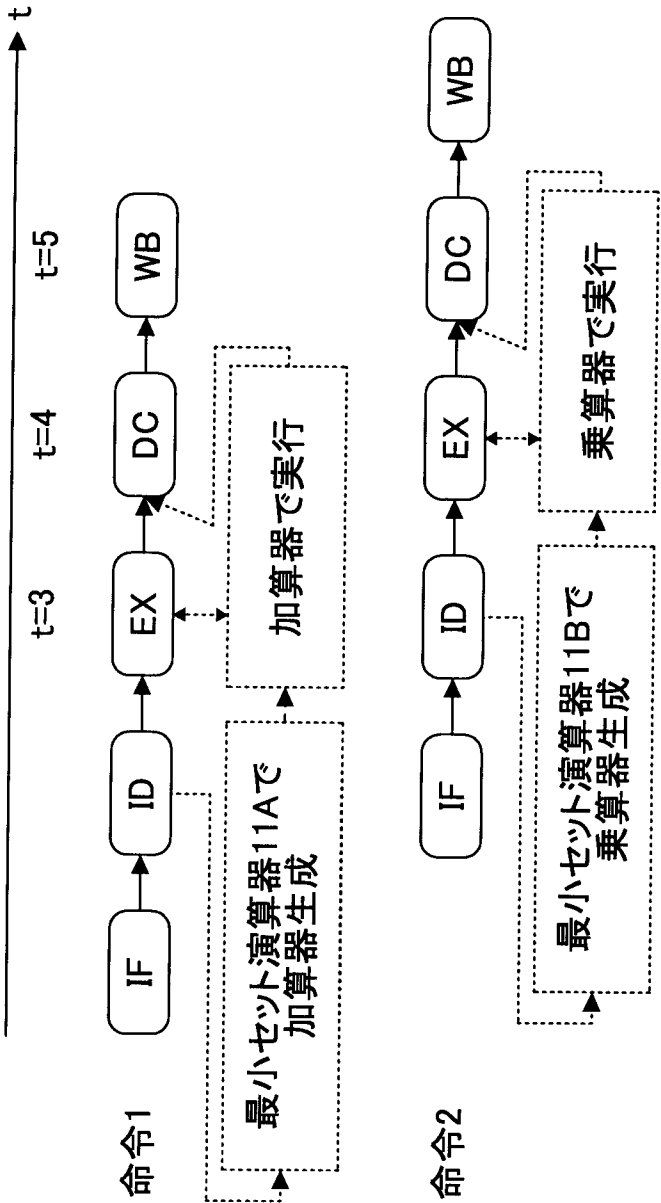
[図5]



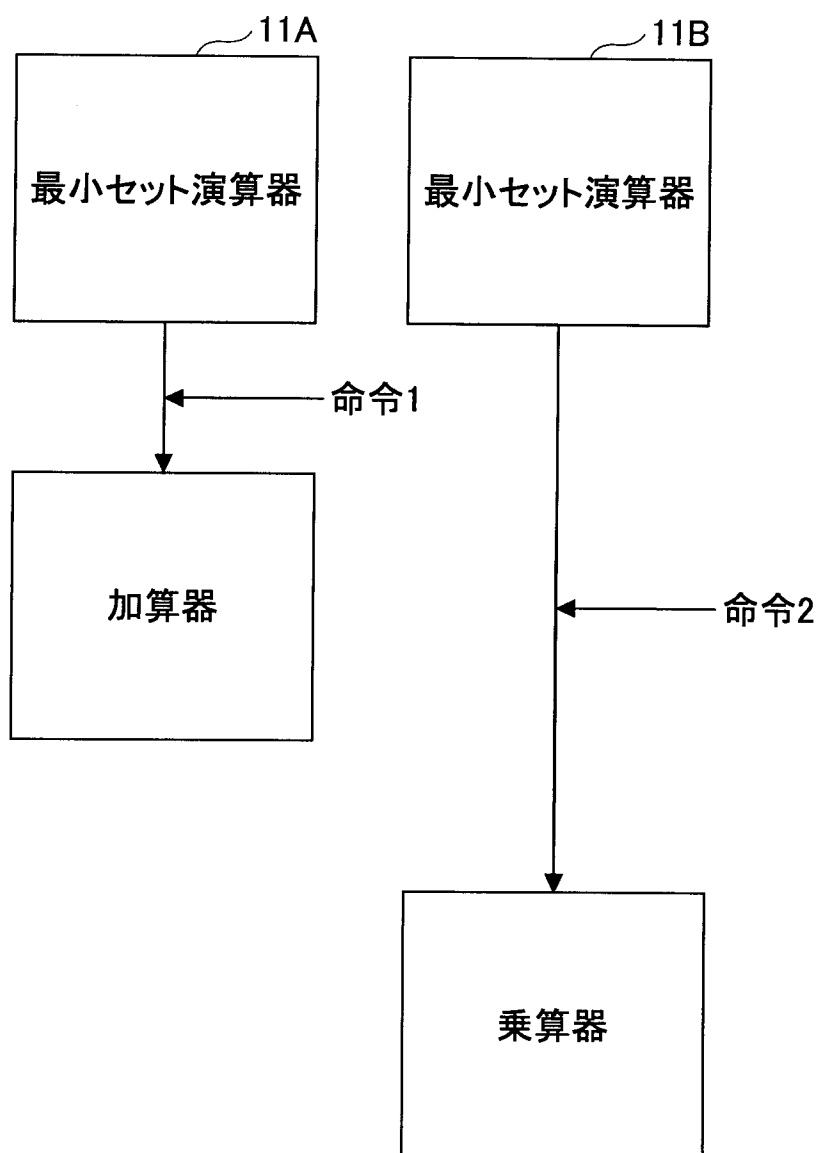
[図6]



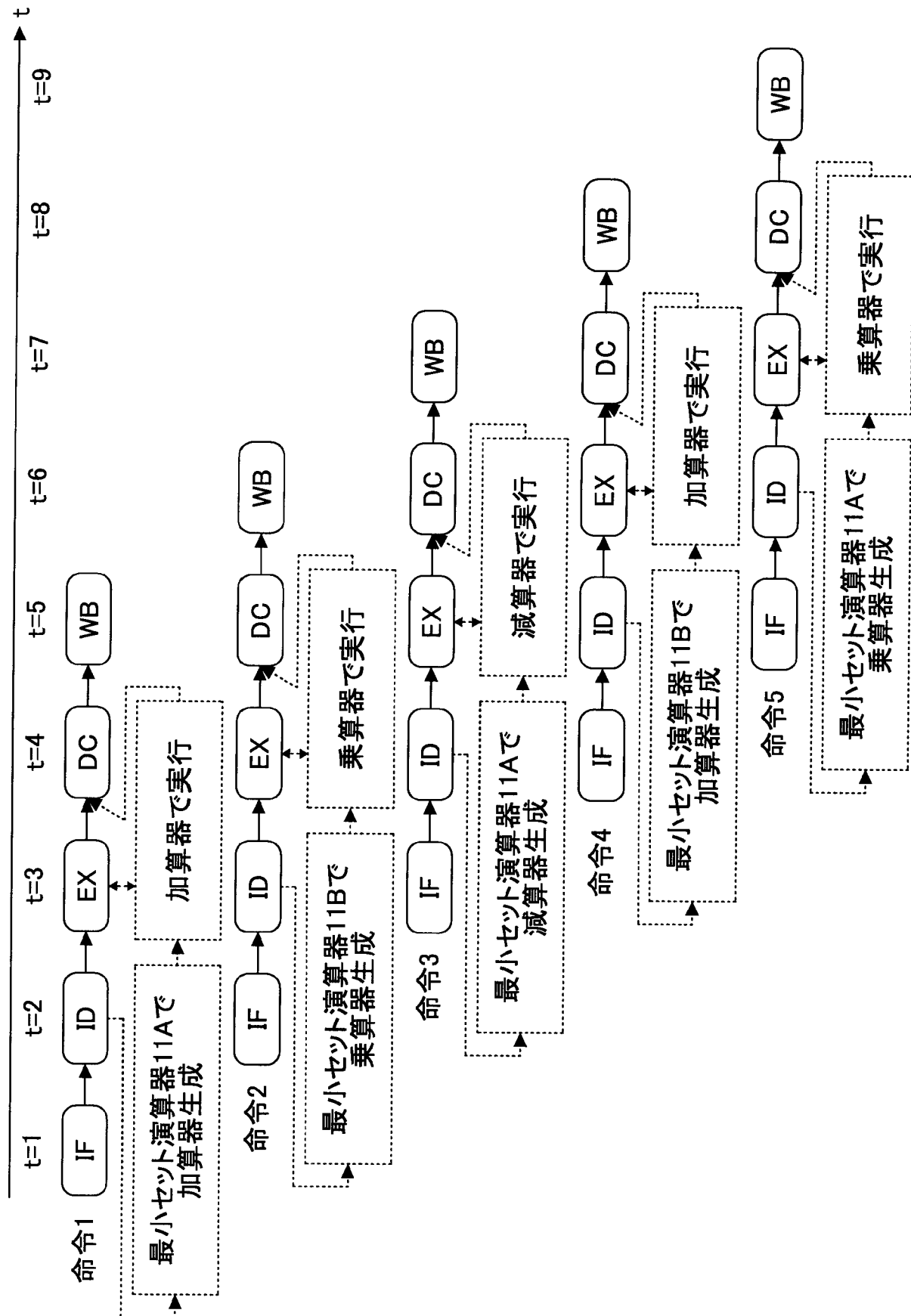
[図7]



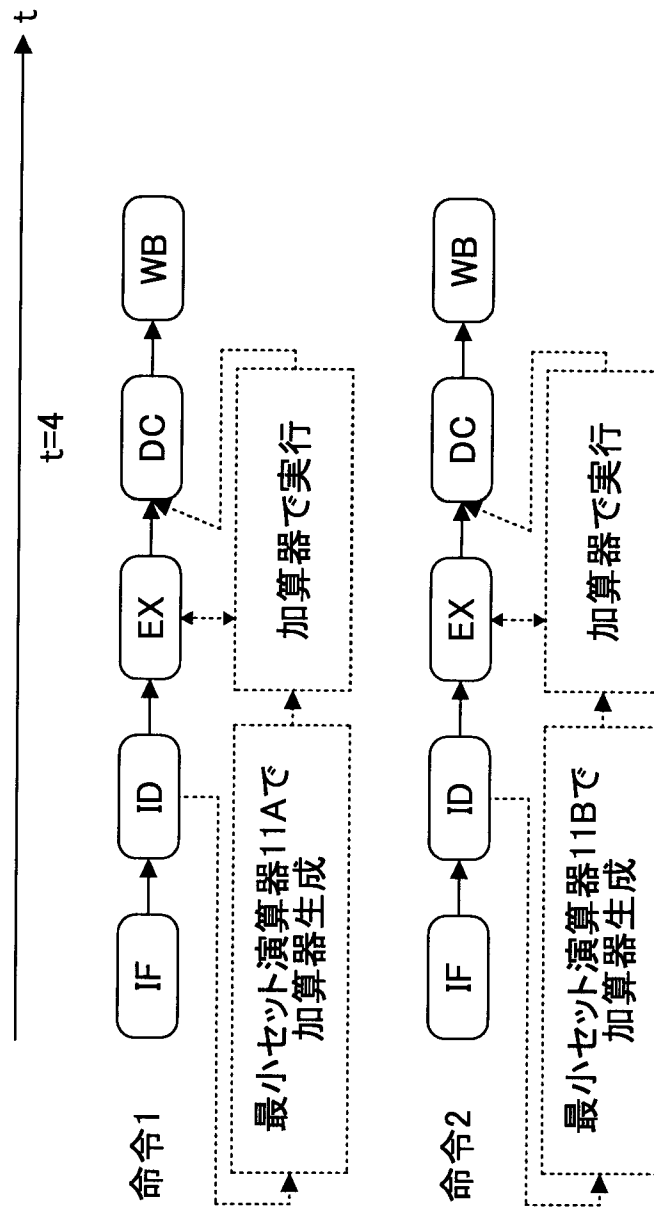
[図8]



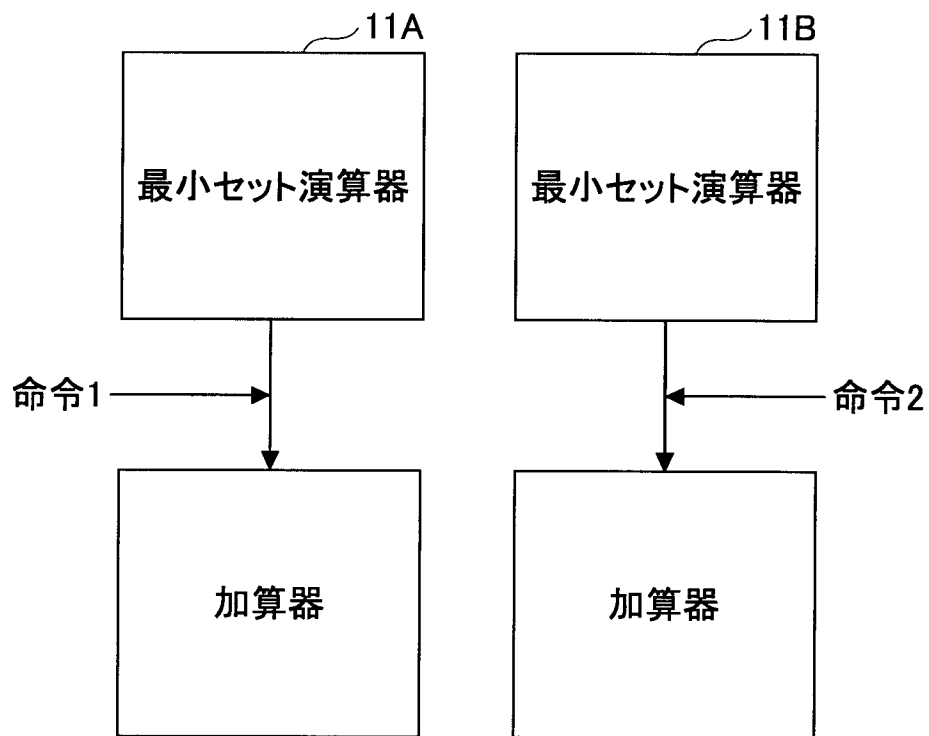
[図9]



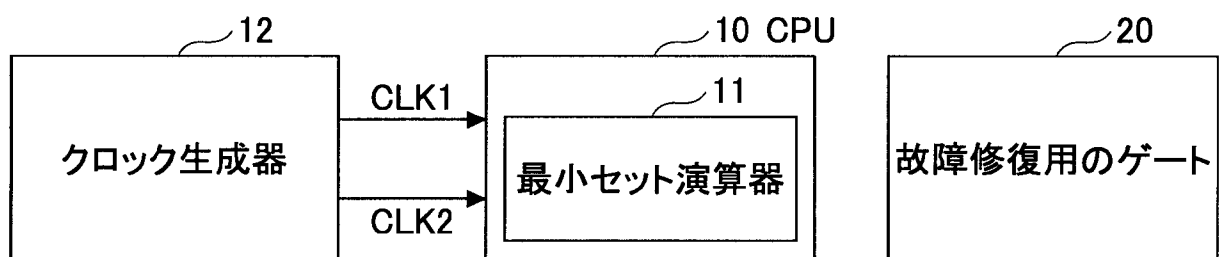
[図10]



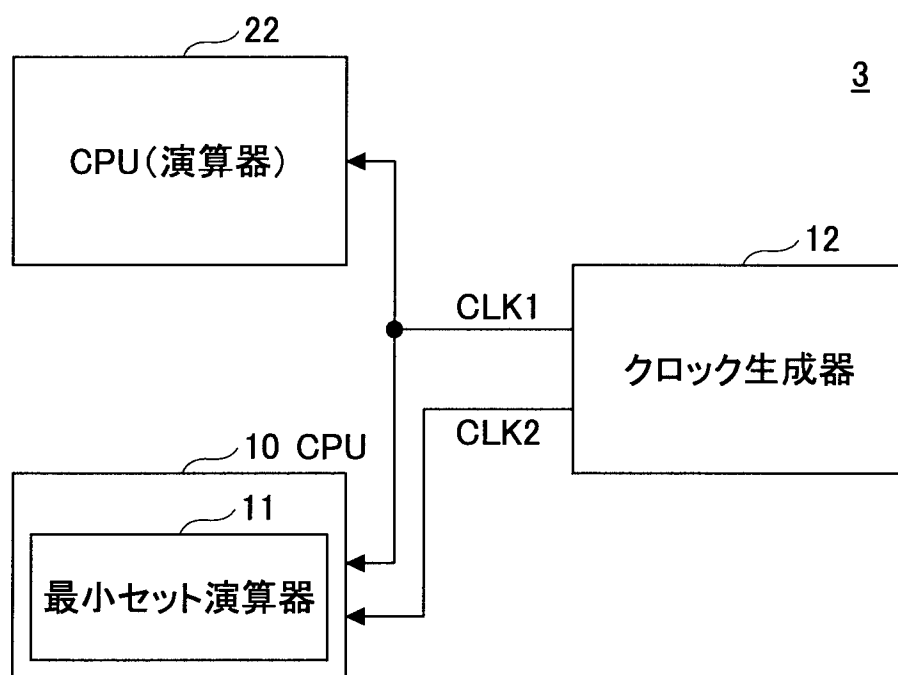
[図11]



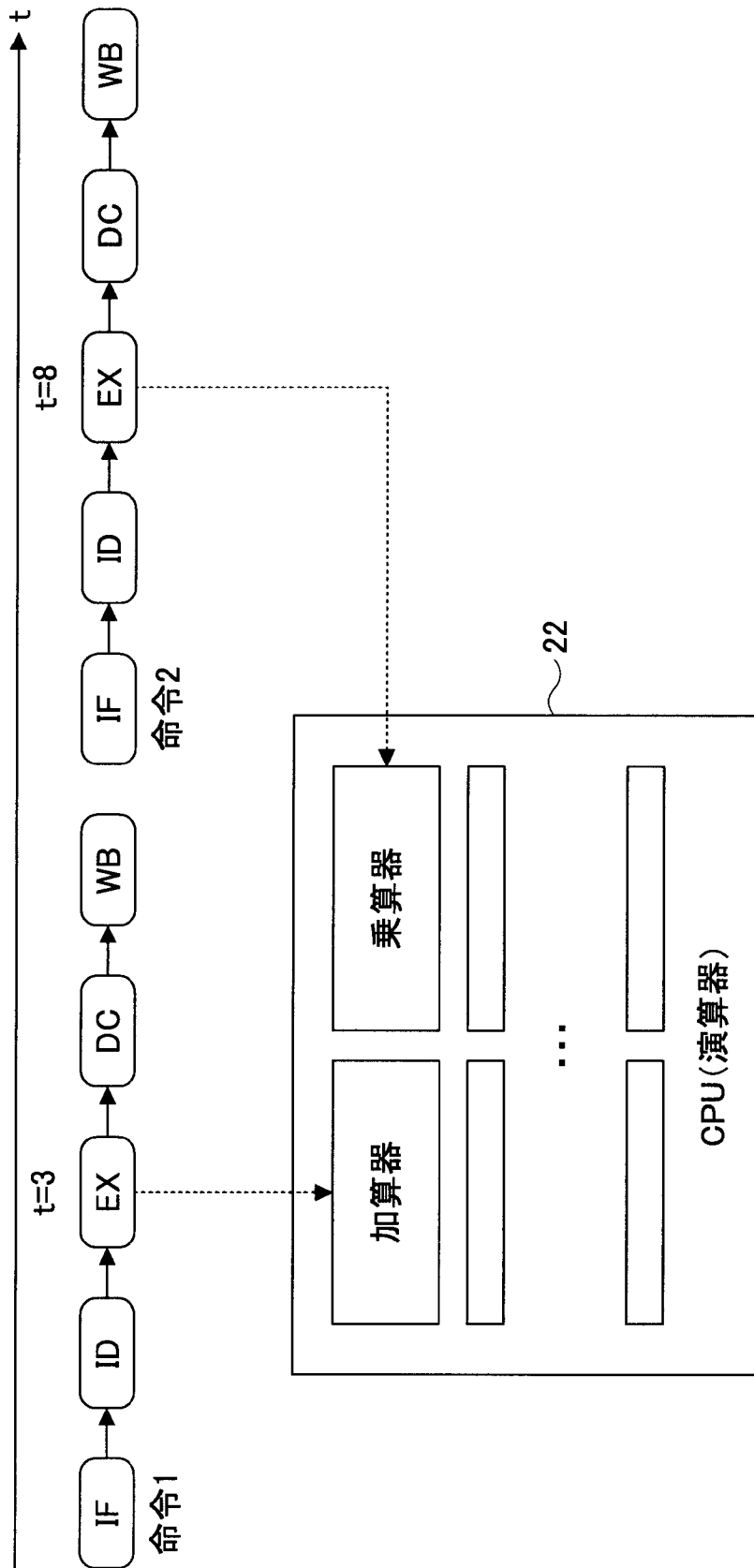
[図12]



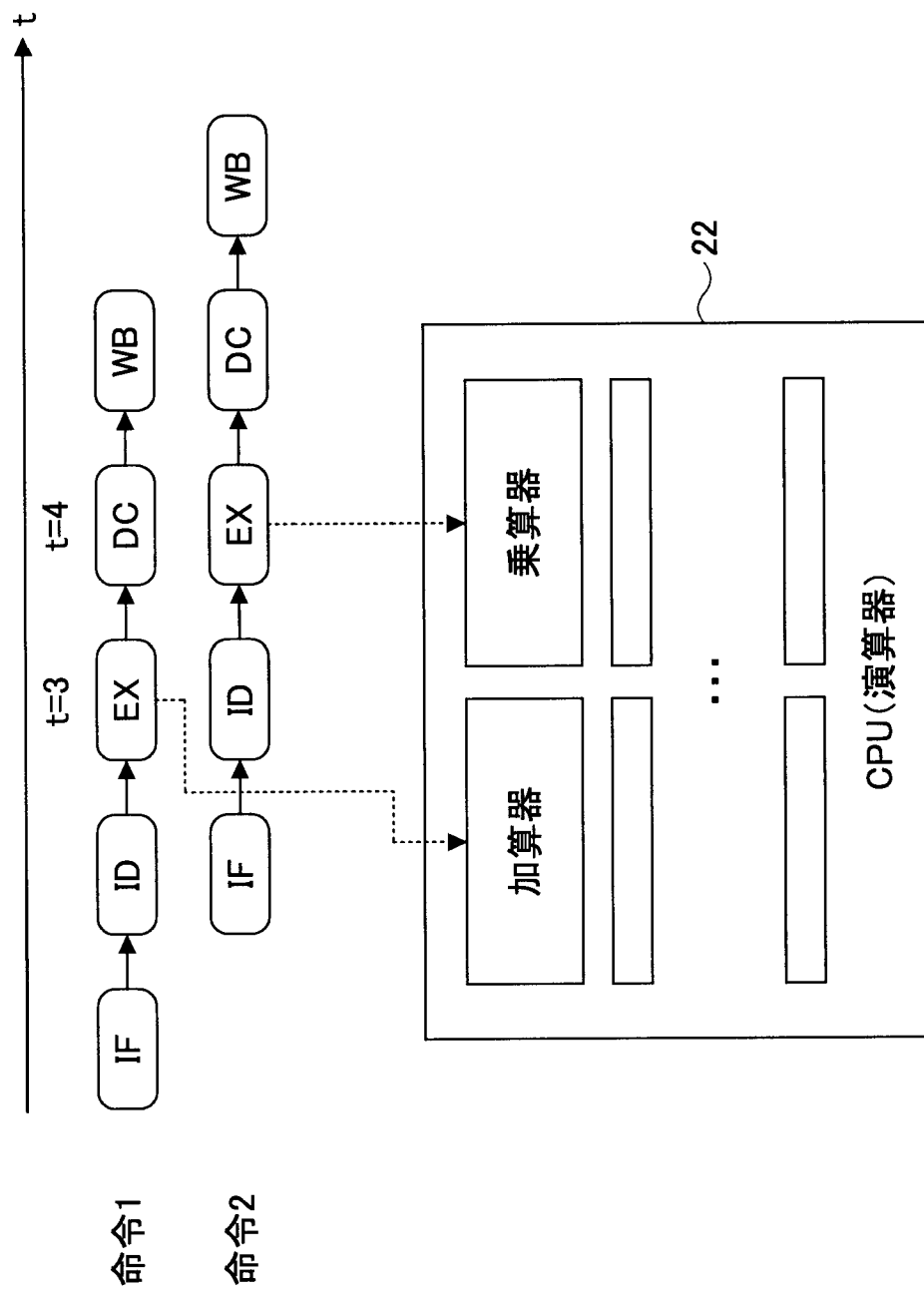
[図13]



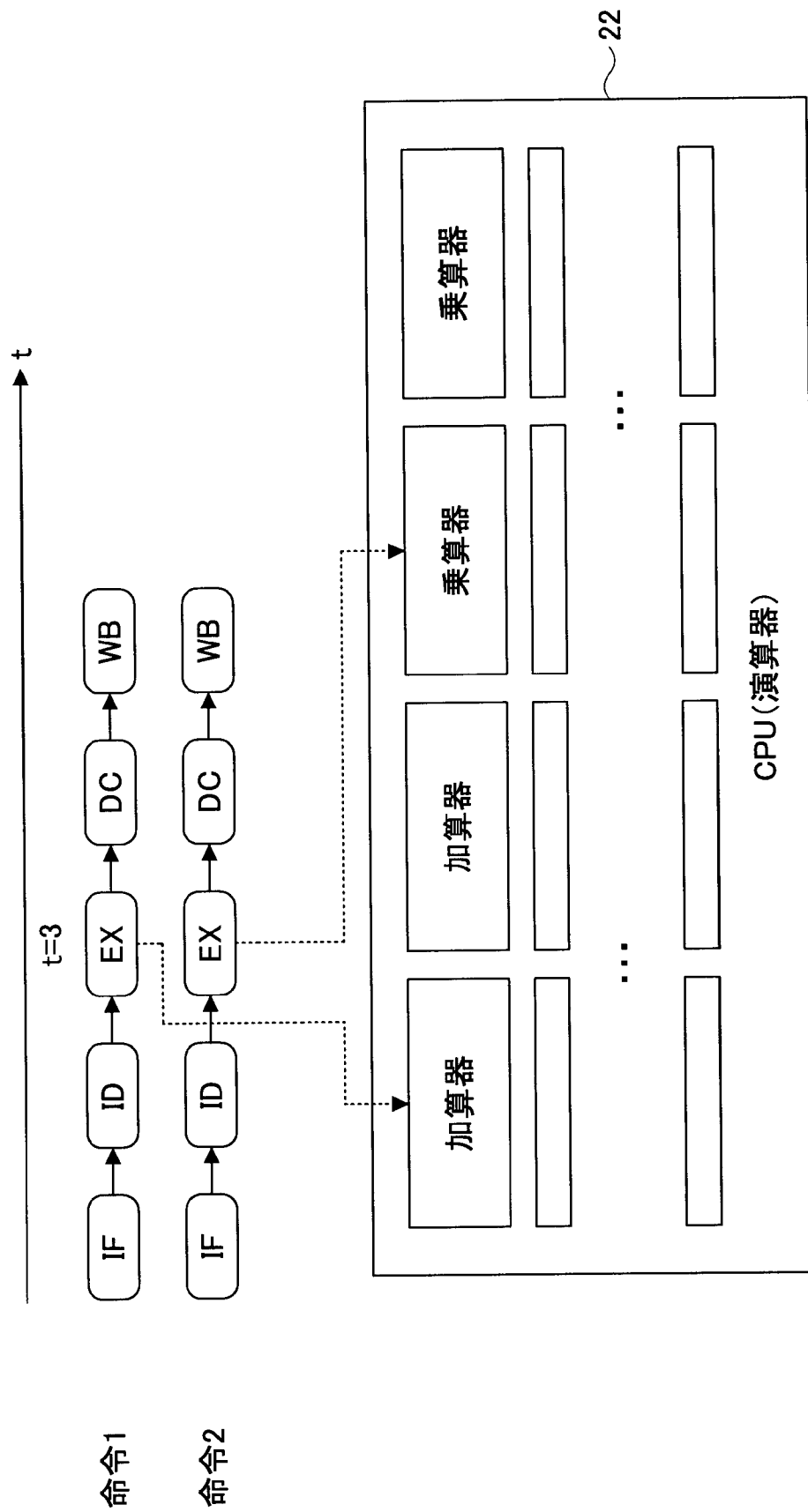
[図14]



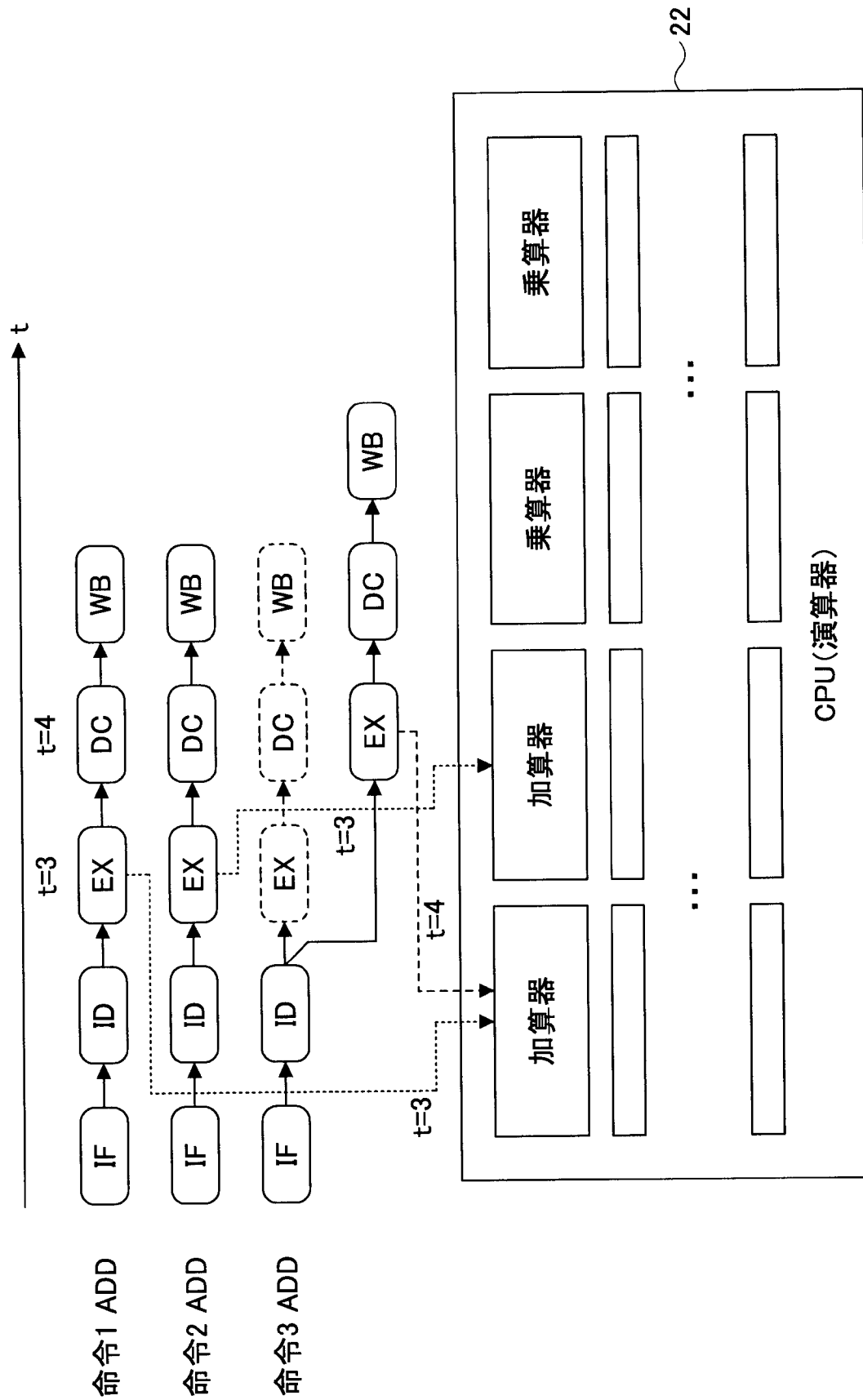
[図15]



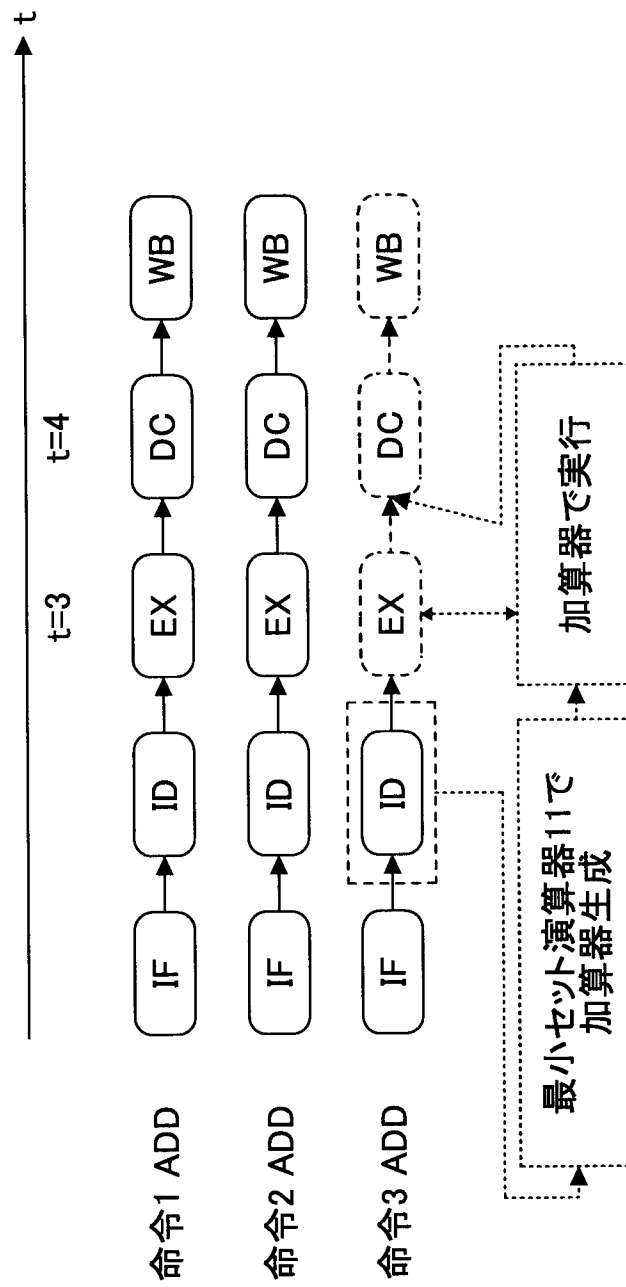
[図16]



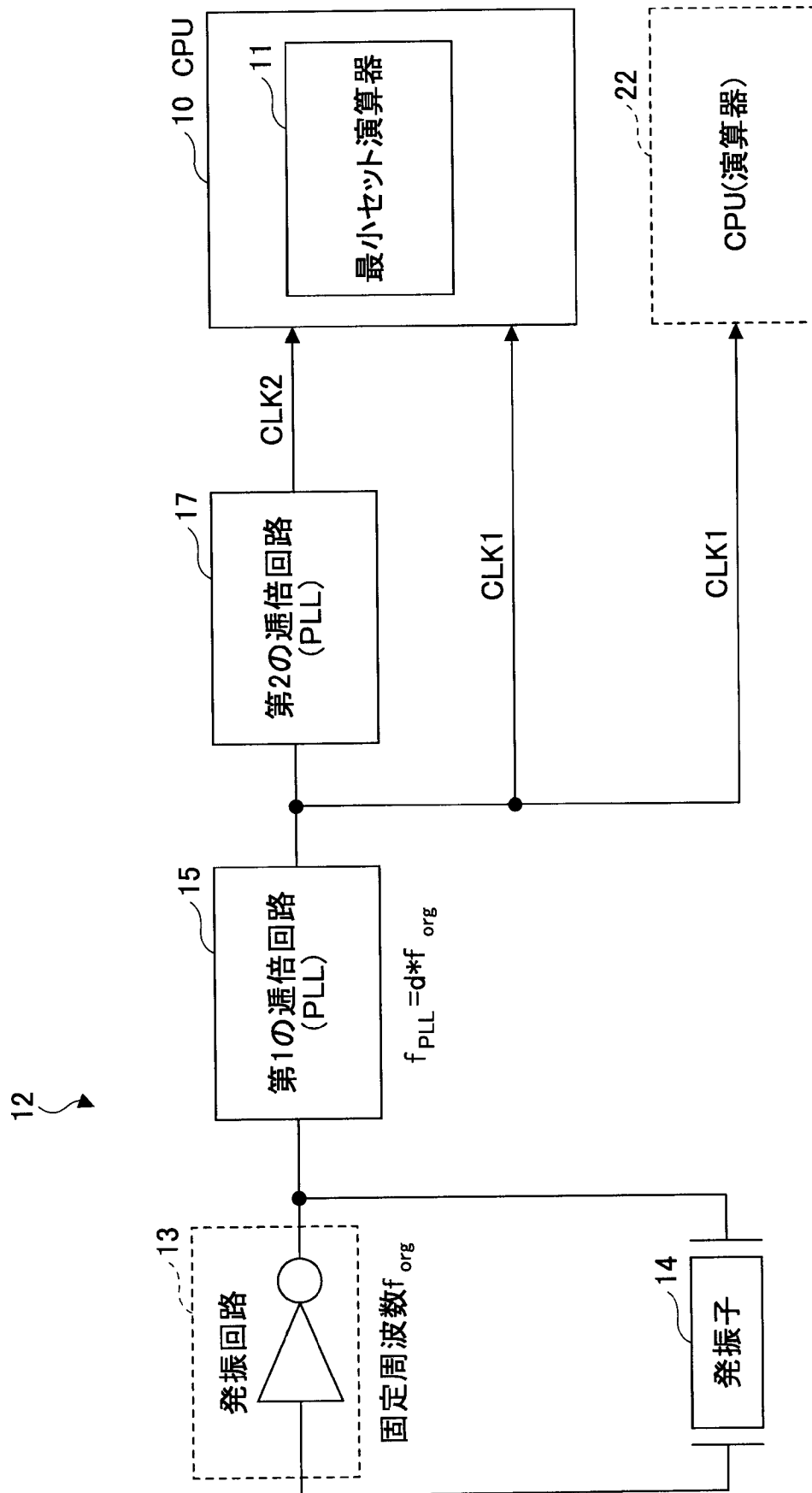
[図17]



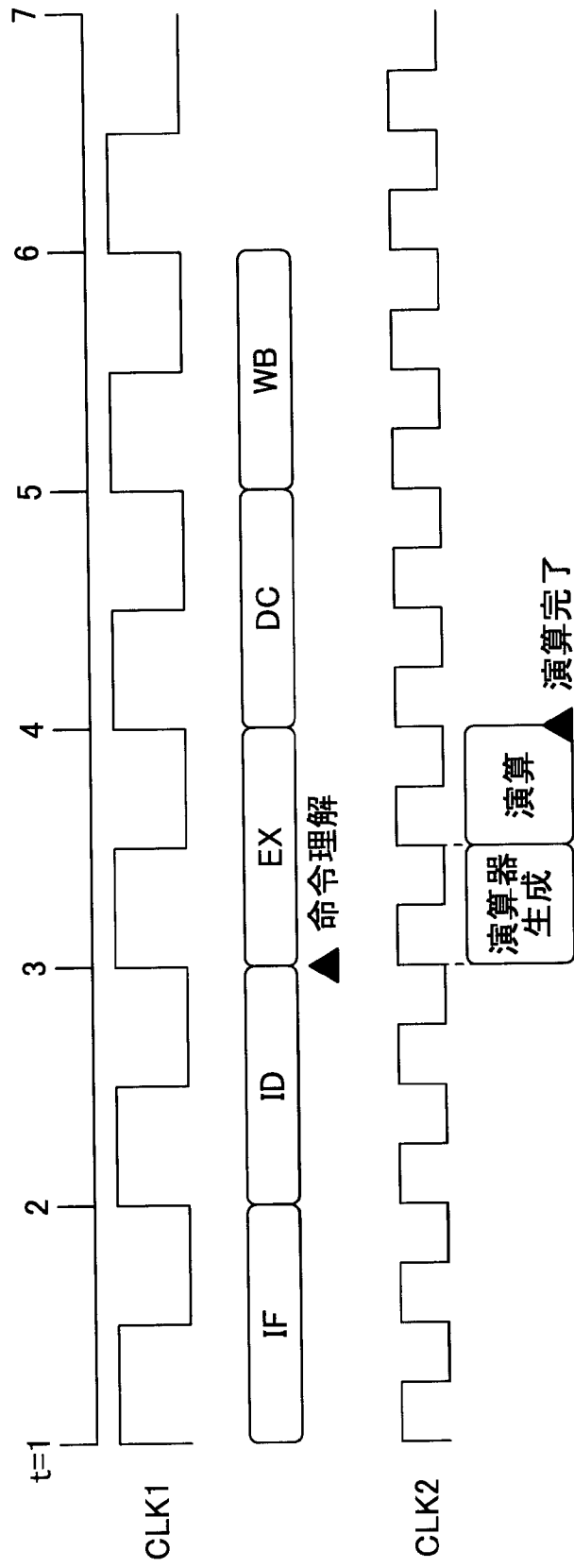
[図18]



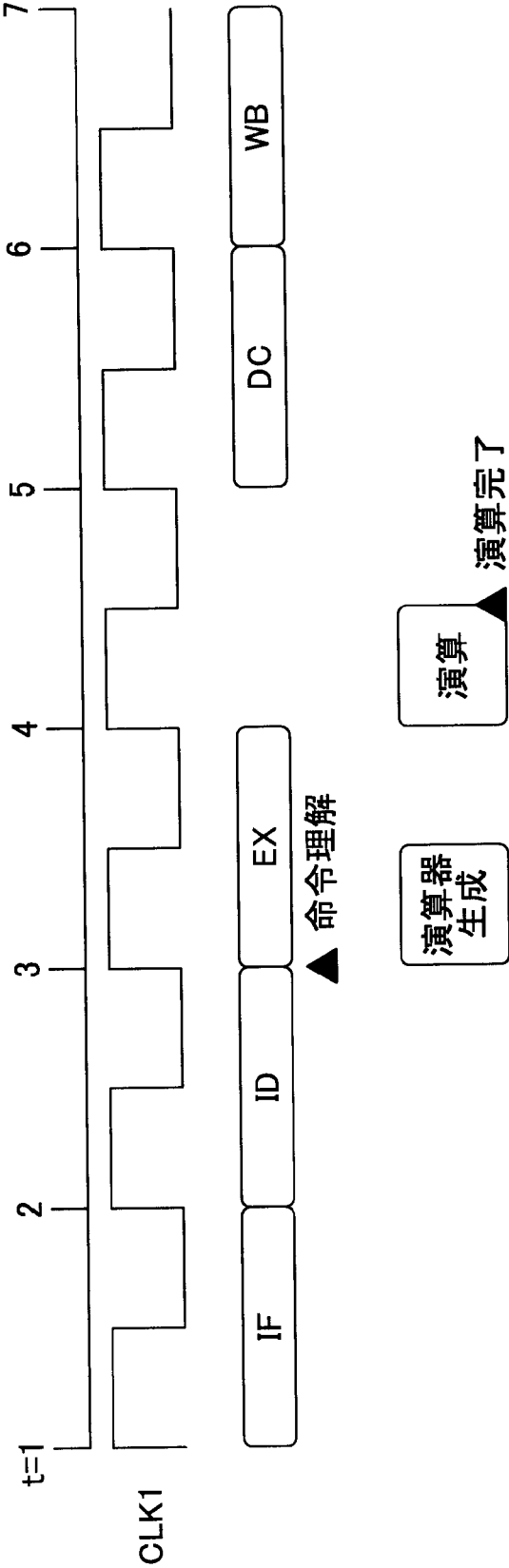
[図19]



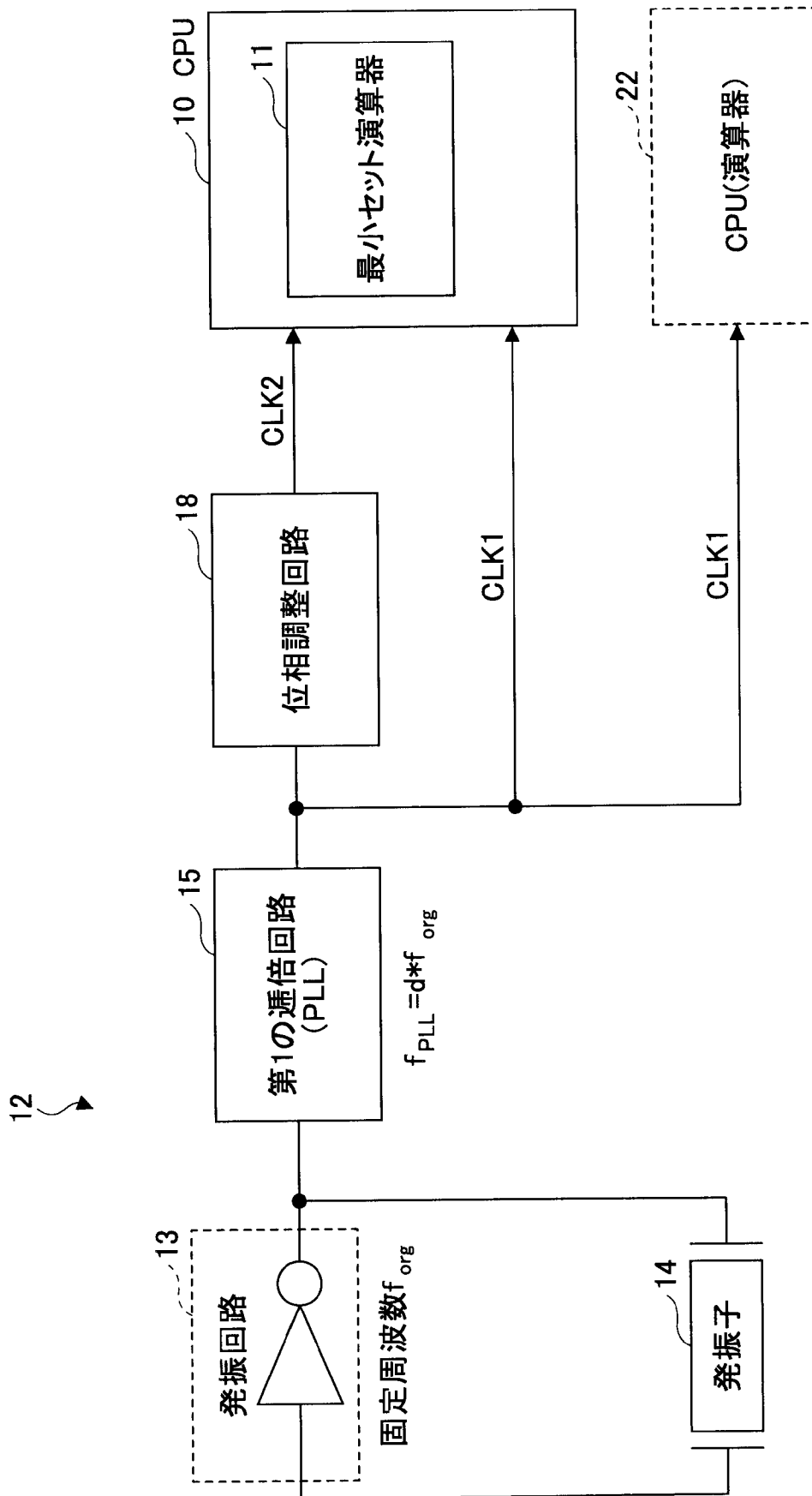
[図20]



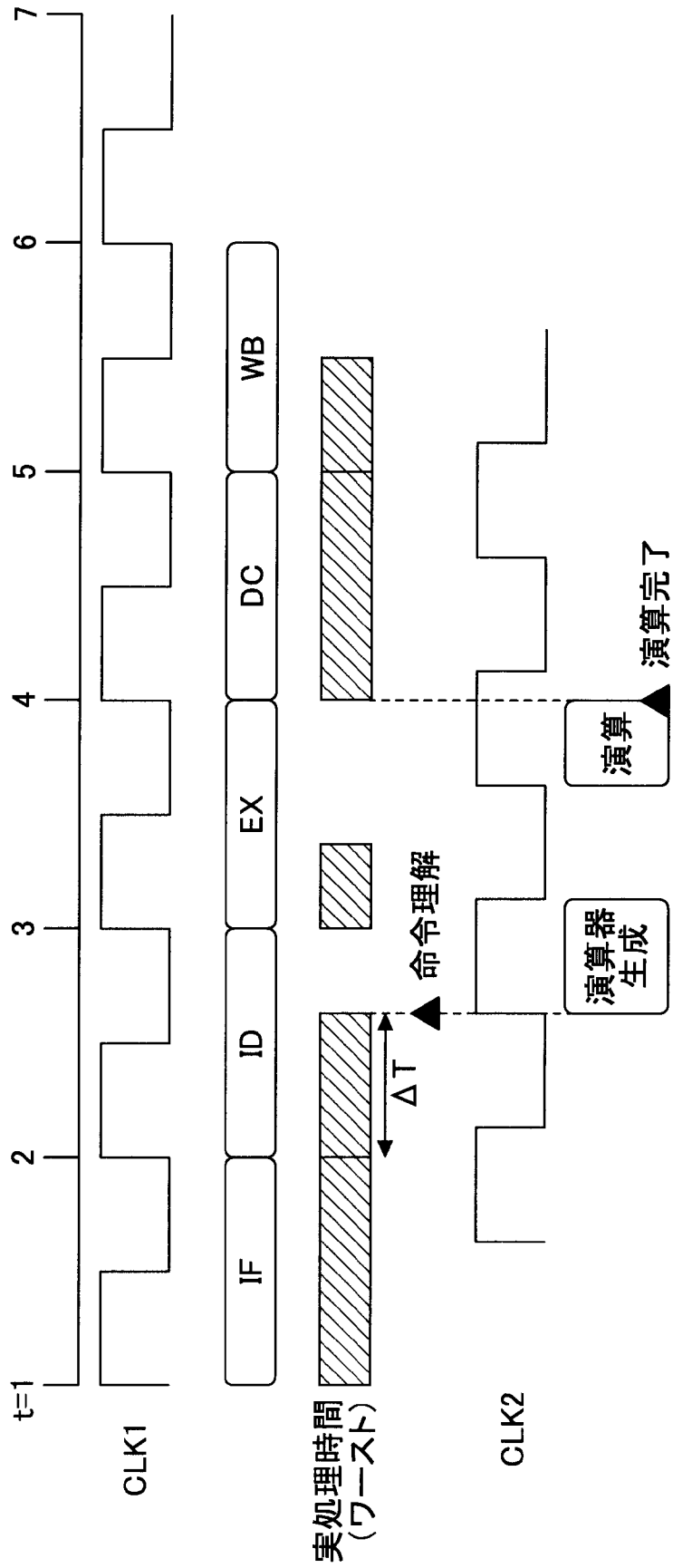
[図21]



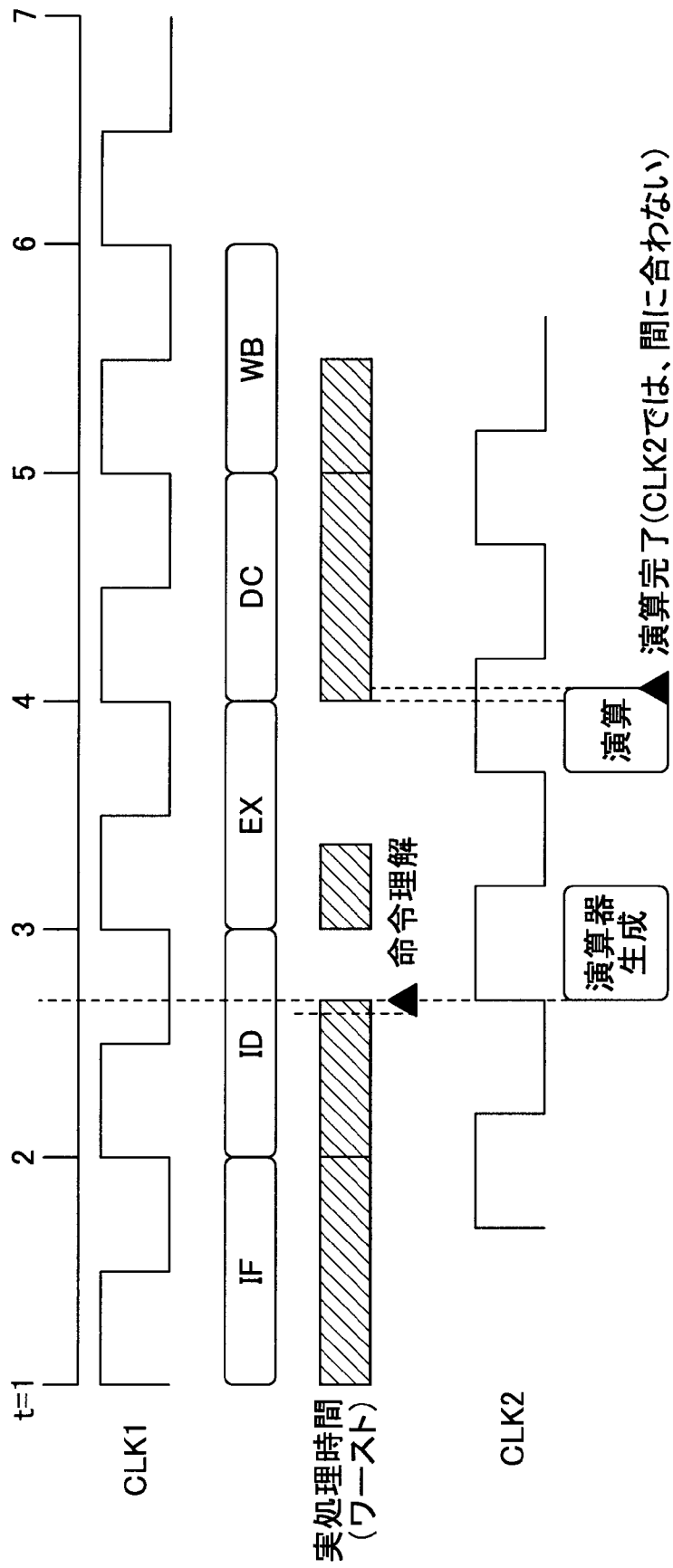
[図22]



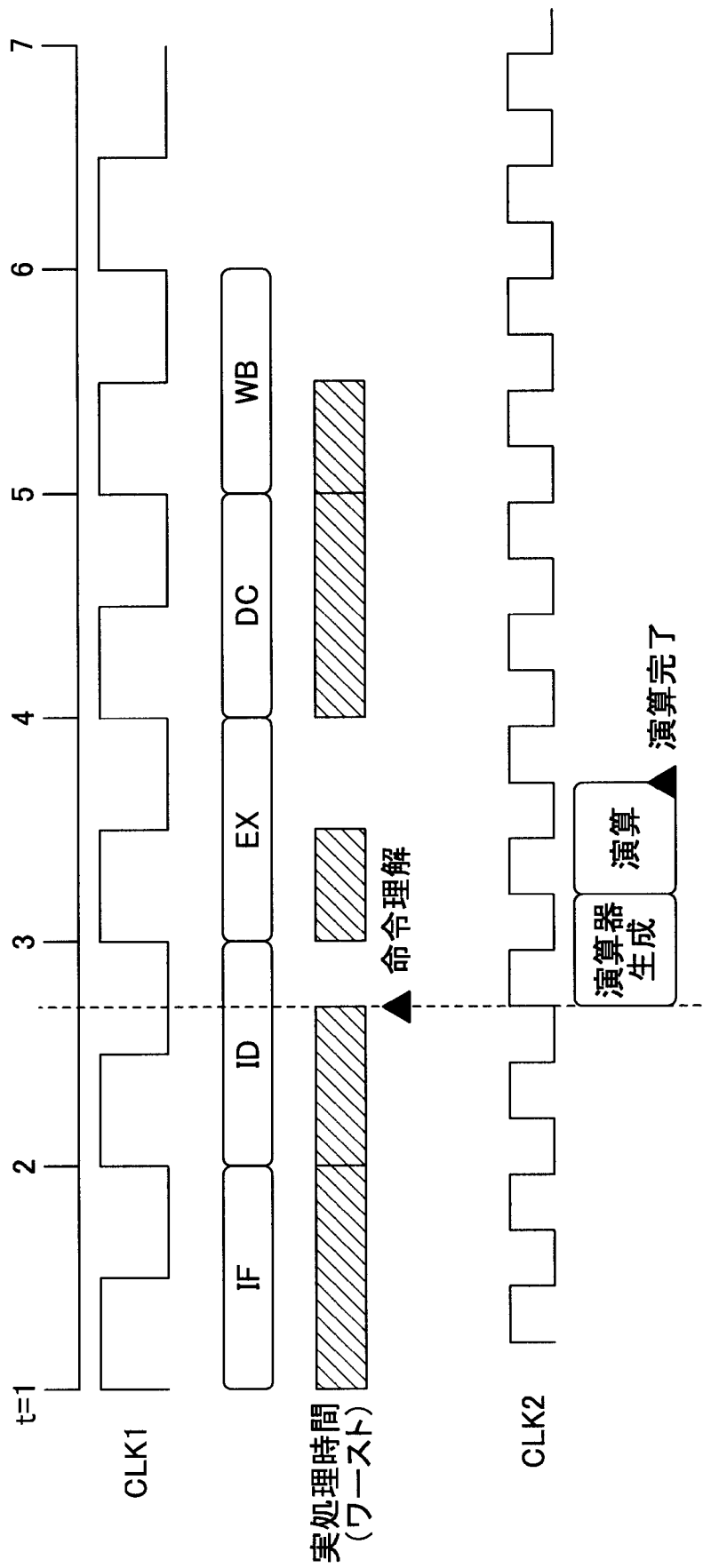
[図23]



[図24]



[図25]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/056227

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/38(2006.01) i, G06F9/30(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)

G06F9/38, G06F9/30

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

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2010
Kokai Jitsuyo Shinan Koho	1971-2010	Toroku Jitsuyo Shinan Koho	1994-2010

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2006-178653 A (Fuji Xerox Co., Ltd.), 06 July 2006 (06.07.2006), paragraphs [0016] to [0022]; fig. 1 to 3 (Family: none)	1-13
Y	JP 8-202549 A (Mitsubishi Electric Corp.), 09 August 1996 (09.08.1996), paragraphs [0040] to [0062]; fig. 3, 4 (Family: none)	1-13
Y	JP 6-195149 A (Panasonic Corp.), 15 July 1994 (15.07.1994), paragraphs [0022] to [0034], [0042] to [0050]; fig. 1, 2, 3, 5, 7 & US 5572714 A	3

☒ 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 April, 2010 (27.04.10)

Date of mailing of the international search report
18 May, 2010 (18.05.10)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/056227

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 11-85507 A (Renesas Technology Corp.), 30 March 1999 (30.03.1999), paragraphs [0024] to [0034], [0042] to [0053]; fig. 1, 3 to 5 & US 6286101 B	7-11
Y	JP 2009-140353 A (Toshiba Corp.), 25 June 2009 (25.06.2009), paragraphs [0022] to [0041]; fig. 1 to 3 (Family: none)	10
A	JP 2008-539485 A (The University Court of the University of Edinburgh), 13 November 2008 (13.11.2008), paragraphs [0084] to [0088]; fig. 12 to 14 & EP 1877927 A	1-13
A	JP 2004-5739 A (Fuji Xerox Co., Ltd.), 08 January 2004 (08.01.2004), paragraphs [0042] to [0070]; fig. 1 to 10 & US 6904514 B	1-13

A. 発明の属する分野の分類 (国際特許分類 (I P C))

Int.Cl. G06F9/38(2006.01)i, G06F9/30(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 (I P C))

Int.Cl. G06F9/38, G06F9/30

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	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 0 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 0 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 0 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)

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
Y	JP 2006-178653 A (富士ゼロックス株式会社) 2006.07.06, 段落 1 6 ~ 2 2、図 1 ~ 3 (ファミリーなし)	1-13
Y	JP 8-202549 A (三菱電機株式会社) 1996.08.09, 段落 4 0 ~ 6 2、図 3、図 4 (ファミリーなし)	1-13
Y	JP 6-195149 A (パナソニック株式会社) 1994.07.15, 段落 2 2 ~ 3 4、段落 4 2 ~ 5 0、図 1、2、3、5、7 & US 5572714 A	3

☒ 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

2 7 . 0 4 . 2 0 1 0

国際調査報告の発送日

1 8 . 0 5 . 2 0 1 0

国際調査機関の名称及びあて先

日本国特許庁 (I S A / J P)

郵便番号 1 0 0 - 8 9 1 5

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)

三坂 敏夫

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 5

5 B

4 1 7 8

C (続き) . 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
Y	JP 11-85507 A (株式会社ルネサステクノロジ) 1999. 03. 30, 段落 24 ～ 34、段落 42 ～ 53、図 1、図 3 ～ 5 & US 6286101 B	7-11
Y	JP 2009-140353 A (株式会社東芝) 2009. 06. 25, 段落 22 ～ 41、図 1 ～ 3 (ファミリーなし)	10
A	JP 2008-539485 A (ザ ユニバーシティ コート オブ ザ ユニ バーシティ オブ エディンバラ) 2008. 11. 13, 段落 84 ～ 88、図 12 ～ 14 & EP 1877927 A	1-13
A	JP 2004-5739 A (富士ゼロックス株式会社) 2004. 01. 08, 段落 42 ～ 70、図 1 ～ 10 & US 6904514 B	1-13