

(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
2021年8月12日(12.08.2021)



(10) 国際公開番号
WO 2021/157448 A1

- (51) 国際特許分類:
G06F 9/48 (2006.01) G06F 15/80 (2006.01)
G06F 15/177 (2006.01)
- (21) 国際出願番号: PCT/JP2021/002880
- (22) 国際出願日: 2021年1月27日(27.01.2021)
- (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 2020-016144 2020年2月3日(03.02.2020) JP
- (71) 出願人: 株式会社ソニー・インタラクティブエンタテインメント (SONY INTERACTIVE ENTERTAINMENT INC.) [JP/

JP]; 〒1080075 東京都港区港南1丁目7番1号 Tokyo (JP).

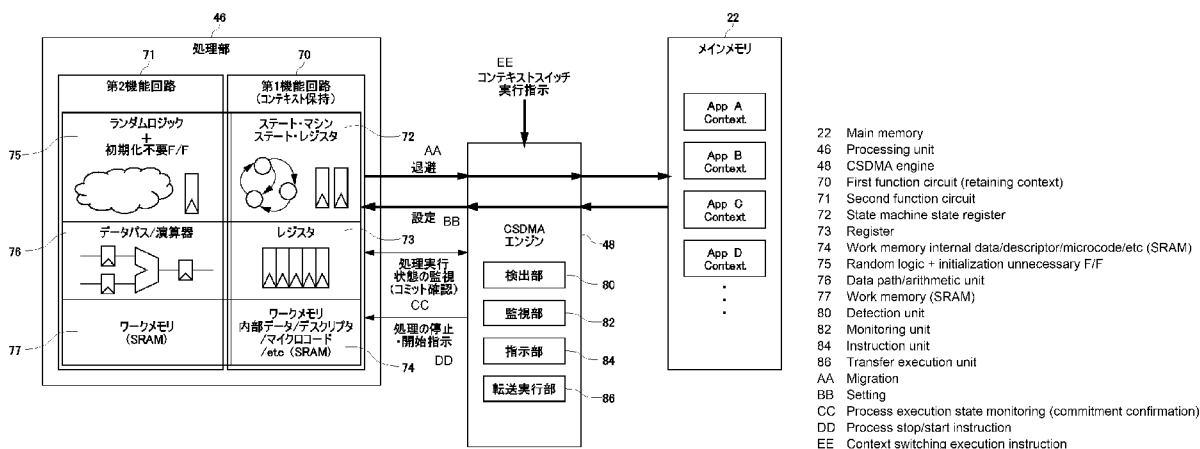
(72) 発明者: 大塚 活志 (OTSUKA Katsushi);
〒1080075 東京都港区港南1丁目7番1号
株式会社ソニー・インタラクティブエンタテインメント内 Tokyo (JP).

(74) 代理人: 森下 賢樹 (MORISHITA Sakaki);
〒1500021 東京都渋谷区恵比寿西2-11-12 Tokyo (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,

(54) Title: DATA PROCESSING SYSTEM, DATA TRANSFER DEVICE, AND CONTEXT SWITCHING METHOD

(54) 発明の名称: データ処理システム、データ転送装置およびコンテキストスイッチ方法



(57) Abstract: A processing unit 46 executes a process for a plurality of applications in time division. A CSDMA engine 48 detects a switching timing of the applications to be executed by the processing unit 46. Upon detection of the switching timing, the CSDMA engine 48 migrates the context of an application having been executed by the processing unit 46 from the processing unit 46 to a main memory 22 without going through a software process for managing the plurality of applications, and sets the context of an application to be processed subsequently by the processing unit 46 from the main memory 22 to the processing unit 46.

(57) 要約: 処理部 46 は、複数のアプリケーションに関する処理を時分割で実行する。CSDMA エンジン 48 は、処理部 46 における実行対象のアプリケーションの切替タイミングを検出する。CSDMA エンジン 48 は、切替タイミングを検出した場合、上記複数のアプリケーションを管理するソフトウェアの処理を介さず、処理部 46 で実行されていたアプリケーションのコンテキストを処理部 46 からメインメモリ 22 へ退避させ、処理部 46 で次に実行されるアプリケーションのコンテキストをメインメモリ 22 から処理部 46 に設定する。

WO 2021/157448 A1

HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, 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,
NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT,
QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, WS, 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 :

- 一 国際調査報告 (条約第21条(3))

明 細 書

発明の名称：

データ処理システム、データ転送装置およびコンテキストスイッチ方法

技術分野

[0001] 本開示はデータ処理技術に関し、特にデータ処理システム、データ転送装置およびコンテキストスイッチ方法に関する。

背景技術

[0002] 複数のアプリケーションに関する処理を時分割で実行する処理部では、実行対象のアプリケーションを切り替える際にコンテキストスイッチを行う必要がある。

発明の概要

発明が解決しようとする課題

[0003] 近年、処理部（例えばGPU等）は、数千以上のレジスタを備えることがあり、また、処理部で実行されるアプリケーションのコンテキストは、数十MB以上のサイズになることがある。この場合、処理部のコンテキストスイッチに長時間を要する可能性がある。

[0004] 本開示はこうした状況に鑑みてなされたものであり、1つの目的は、コンテキストスイッチに伴う処理の遅延を抑制することにある。

課題を解決するための手段

[0005] 上記課題を解決するために、本発明のある態様のデータ処理システムは、複数のアプリケーションに関する処理を時分割で実行する処理部と、処理部とメモリ間でデータを転送するデータ転送部と、を備える。データ転送部は、処理部における実行対象のアプリケーションの切替タイミングを検出する検出部と、切替タイミングが検出された場合、複数のアプリケーションを管理するソフトウェアの処理を介さずに、処理部で実行されていたアプリケーションのコンテキストを処理部からメモリへ退避させ、処理部で次に実行されるアプリケーションのコンテキストをメモリから処理部に設定する転送処

理を実行する転送実行部と、を含む。

[0006] 本発明の別の態様は、データ転送装置である。この装置は、複数のアプリケーションに関する処理を時分割で実行する処理部における実行対象のアプリケーションの切替タイミングを検出する検出部と、切替タイミングが検出された場合、複数のアプリケーションを管理するソフトウェアの処理を介さずに、処理部で実行されていたアプリケーションのコンテキストを処理部からメモリへ退避させ、処理部で次に実行されるアプリケーションのコンテキストをメモリから処理部に設定する転送処理を実行する転送実行部と、を備える。

[0007] 本発明のさらに別の態様は、コンテキストスイッチ方法である。この方法は、複数のアプリケーションに関する処理を時分割で実行する処理部における実行対象のアプリケーションの切替タイミングを検出するステップと、切替タイミングが検出された場合、複数のアプリケーションを管理するソフトウェアの処理を介さずに、処理部で実行されていたアプリケーションのコンテキストを処理部からメモリへ退避させ、処理部で次に実行されるアプリケーションのコンテキストをメモリから処理部に設定する転送処理を実行するステップと、をコンピュータが実行する。

[0008] なお、以上の構成要素の任意の組合せ、本開示の表現を、コンピュータプログラム、コンピュータプログラムを記録した記録媒体などの間で変換したものもまた、本開示の態様として有効である。

発明の効果

[0009] 本開示によれば、コンテキストスイッチに伴う処理の遅延を抑制することができる。

図面の簡単な説明

[0010] [図1]第1実施例のコンピュータの構成を示す図である。

[図2]処理部とCSDMAエンジンの詳細な構成を示すブロック図である。

[図3]処理部におけるビデオ描画処理を時系列に示す図である。

[図4]ビデオタイミングの例を示す図である。

[図5] S O Cの構成を示す図である。

[図6] S O Cの構成を示す図である。

[図7] S O Cの構成を示す図である。

[図8] S O Cの構成を示す図である。

[図9] 図 9 Aは、第 1 実施例におけるコンテキストスイッチ時の内部処理の例を示す図であり、図 9 Bは、変形例におけるコンテキストスイッチ時の内部処理の例を示す図である。

[図10] 第 2 実施例の処理部の構成を示す図である。

[図11] 第 3 実施例におけるコンテキストスイッチ時の内部処理の例を示す図である。

[図12] 図 1 2 A、図 1 2 B、図 1 2 Cは、コンテキストスイッチにおける処理部と C S D M A エンジンの動作を示す図である。

[図13] 第 4 実施例の処理部の構成を示す図である。

[図14] 第 5 実施例の処理部の構成を示す図である。

[図15] 図 1 5 A と 図 1 5 B は、アプリケーションの割当例を示す図である。

発明を実施するための形態

[0011] 実施例では、複数のアプリケーションに関する処理を時分割で実行する処理部であり、言い換えれば、複数のアプリケーションにより時分割で共有される処理部におけるコンテキストスイッチを高速に実行する技術を提案する。実施例の技術は、或るシステムが、サスペンドまたはハイバネーション等、状態を保持したまま一時停止し、その後、保持した状態から実行を再開する動作にも有効であり、その動作を高速化することができる。

[0012] 処理部は、S O C (System-On-a-Chip) 等におけるハードウェア機能ブロックであってもよい。また、処理部は、(1) C P U (Central Processing Unit) ・ D S P (Digital Signal Processor) ・ G P U (Graphics Processing Unit) ・ N P U (Network Processing Unit) 等のプロセッサであってもよく、(2) ビデオの圧縮および伸長を実行するビデオ・コーデック・ブロックであってもよい。ビデオ・コーデック・ブロックのうちビデオの圧縮を

実行するブロックを以下「ビデオエンコーダ」とも呼ぶ。また、処理部は、
(3) ビデオの解像度変換や画質変換、多重化等を実行し、規定のタイミングでピクセルデータを処理するビデオ（ディスプレイ）・パイプライン・ブロック（以下「ビデオパイプライン」とも呼ぶ。）であってもよく、(4) 周辺機器への入出カインタフェースブロックであってもよい。

[0013] <背景>

1つのハードウェア機能ブロック（実施例では「処理部」と呼ぶ。）が、複数のアプリケーション・タスク・スレッド（以下「アプリケーション」と呼ぶ。）を時分割で実行することがある。処理部は、アプリケーションごとに固有の処理を実行するため、実行対象のアプリケーションを指定するデータや、設定・動作を指定するデータ等を保持する。処理部は、処理の開始にあたって、それらのデータをセットアップした上で処理を開始する。処理部は、アプリケーションに関する処理を実行中に、アプリケーション固有のデータや設定、動作状態等を随時生成する。これら個々のアプリケーションに関する処理において必要となるデータや生成されるデータを以下「コンテキスト」と呼ぶ。つまり、コンテキストとは、同じ処理を、再び同じ処理部で実行したときに、同様の結果を得る再現性が確保されるために必要なデータである。

[0014] 1つの処理部が複数のアプリケーションを時分割で実行する場合、当該処理部における切替処理が必要になる。具体的には、それまで処理部が実行していたアプリケーションのコンテキストを外部に退避させ、次に実行するアプリケーションのコンテキストを処理部に設定することで、時分割によるアプリケーション切替がなされる。この切替を以下「コンテキストスイッチ」と呼ぶ。コンテキストの退避と設定を行う間は、当該処理部の処理を停止する必要がある。処理部の利用効率を高め、また、処理部における処理の遅延を低減する（言い換えればリアルタイム性を高める）ために、コンテキストスイッチを高速化し、処理部の処理停止時間を短縮することが求められる。

[0015] 一般にコンテキストスイッチは、ソフトウェアが処理部にアクセスし、ソ

ソフトウェアによる逐次処理にて、コンテキストの退避と設定をレジスタアクセスを介して実行されることが多い。一方、近年の処理部は高機能化が進み、設定用レジスタ数は数千以上、また、処理中に保持されているコンテキストのデータ量は数十MB以上になることがある。そのため、ソフトウェアによる逐次処理にてコンテキストスイッチをおこなうと、数十ミリ秒オーダーの時間を要するなど、リアルタイム性を大きく損ねてしまう可能性がある。

[0016] CPUの中には、コンテキストスイッチの高速化を支援するハードウェア機能を持つものがある。しかし、その機能は、ソフトウェアからアクセス可能な一部のレジスタについてコンテキストの退避と設定をハードウェアが自動的に実行するものに過ぎず、内部に保持される全コンテキストの退避と設定をカバーするものではない。そのため、ソフトウェアによる逐次処理にて他のコンテキストの退避と設定を行っている。これは、一般的なx86またはARMアーキテクチャのCPUにおいて、OS（Windows（登録商標）やLinux（登録商標））と複数のアプリケーションが実行される場合等に言える。

[0017] <実施例の概要>

実施例のデータ処理システムは、処理部のコンテキストスイッチのためのコンテキスト転送を実行する専用機能ブロックを備える。これにより、膨大なコンテキストを利用する処理部とアプリケーションの組み合わせにおいても、コンテキストスイッチを高速に実行する。言い換えれば、コンテキストスイッチに伴う処理の遅延を抑制する。以下、この専用機能ブロックを、「コンテキストスイッチDMA（Direct Memory Access）エンジン」と呼び、「CSDMAエンジン」と表記する。

[0018] CSDMAエンジンは、アプリケーションの切替タイミングを通知されるもしくは自ら検出すると、（1）メモリ上に保持されている各アプリケーションのコンテキストを選択し、ソフトウェアを介さずに処理部へ転送して設定する。もしくは、（2）処理部に保持されているコンテキストを、ソフトウェアを介さずにメモリへ転送・退避させる。

- [0019] コンテキストスイッチにおける転送対象は、レジスタ内容と、処理部内メモリの内容と、ステート・マシンの状態ステート（状態遷移の内部ステート）と、オンザフライで実行中のアウトスタンディングな処理の現在状態とを含む。これらの転送対象は、ソフトウェアからアクセス可能なものに限定されず、それ以外に、処理部が内蔵し、コンテキストを保持しているものも含む。つまり、実施例におけるコンテキストスイッチでの転送対象は、通常はソフトウェアからアクセスできないコンテキストも含む。
- [0020] CSDMAエンジンは、アプリケーションの切替タイミングを通知されるもしくは自ら検出すると、処理部へ動作停止指示を出し、また、オンザフライで実行中の処理、特にアウトスタンディング（out standing）やアウトオブオーダー（out of order）でコミット処理が必要な処理についてその完了を確認次第、コンテキストの退避と設定を開始してもよい。CSDMAエンジンは、コンテキストの退避と設定の完了後に、次のアプリケーションの処理開始を指示してもよい。これら一連の処理は、CSDMAエンジンが実行することに制限されず、最新のコンテキストを保持する処理部とCSDMAエンジンとが連携して実行されてもよい。
- [0021] 処理部には、複数のアプリケーション用の複数のキューを設けてもよい。それぞれのキューには、対応するアプリケーションから、処理部に対する指示を示すデータが随時書き込まれてもよい。処理部は、コンテキストスイッチが発生すると、データを読み出すキューを切り替えてもよい。また、或るアプリケーションに対応するキューにデータが書き込まれるタイミングは、そのアプリケーションが処理部を占有している期間に制限されなくてもよい。この場合、コンテキストスイッチ実行中も、各アプリケーションは、処理の指示をキューに積み続けることができ、コンテキストスイッチに伴う処理の停止時間を短縮することができる。また、キューにデータを書き込むCPU上のソフトウェアが、処理部（例えばGPU）と非同期に動作する場合にも対応できる。
- [0022] CSDMAエンジンは、垂直帰線期間または水平帰線期間の開始をトリガ

として、コンテキストスイッチを実行してもよい。ビデオ処理システム等では、垂直帰線期間および水平帰線期間に、GPUやビデオパイプライン、ビデオエンコーダ等を一時停止することができる。そこで、短時間の垂直帰線期間中または水平帰線期間中にコンテキストスイッチを実行することで、コンテキストスイッチのオーバーヘッドを隠蔽することができる。

[0023] 処理部からコンテキストを退避する際、および、処理部へコンテキストを設定する際に、DFT (Design For Test) 用のスキャンチェーン、または、メモリBIST (Built-In Self-Test) を経路として用いてもよい。これにより、処理部とメモリとをCSDMAエンジンを介して接続する系が、通常データ転送に用いるチップ内バス(ファブリック)を介さないため、通常データ転送を阻害しない。また、コンテキストスイッチのための専用インタフェースを設ける必要がない。また、処理部内においても、コンテキスト情報を保持する回路にDFT回路が接続されているため、通常データ転送用の系や専用系を用いる必要がなくなる。

[0024] 処理部の機能ブロックのうち、コンテキストを保持する機能ブロック(例えば回路)のみ、その一部または全てを多重化してもよい。この場合、以下のステップでコンテキストスイッチが実行されてもよい。(1) 処理部を一時停止させることなく、多重化された他の回路に対して、コンテキストを予め設定する。(2) 処理部の処理を停止させ(直前処理のコミットを待ってもよい)、上記(1)でコンテキストを設定した回路を選択し、次のアプリケーションの実行を開始する。(3) 上記(1)で実行中であったアプリケーションのコンテキストを退避させる。これにより、コンテキストスイッチの処理と、処理部の処理とを時間的にオーバーラップさせ、処理の停止時間を一層削減することができる。

[0025] また、処理部の機能ブロックのうち、コンテキストを保持する回路を多重化し、さらに、処理部は、コミット処理を待つための一時停止を行わずに、次のアプリケーションの処理を開始してもよい。ただし、コンテキストスイッチ前のアプリケーションと、コンテキストスイッチ後のアプリケーション

とで、同一回路を別設定にて動かす等、両アプリケーションの設定が共通ではないことによる処理の破綻がおこるか否かを判定し、必要な場合は、コミット完了まで一時停止してもよい。さらにまた、一部の機能回路を並列化し、処理破綻を回避しながら、複数アプリケーションを同時実行してもよい。これにより、複数アプリケーションの処理（アウトスタンディングまたはアウトオブオーダの処理であってもよい）を並行して実行でき、コンテキストスイッチによる停止時間を短縮することができる。また、回路利用効率、性能スケーラビリティおよびフレキシビリティを向上することができる。

[0026] <第1実施例>

図1は、第1実施例のコンピュータ10の構成を示す。コンピュータ10は、複数のクライアント装置の要求に応じて、複数のアプリケーション（ゲーム等）を並列処理し、各アプリケーションの処理結果（例えば画像データ）を各クライアント装置に提供するサーバであってもよい。また、コンピュータ10は、複数のアプリケーションを時分割で並列実行し、各アプリケーションの処理結果（例えば画像データ）を表示装置に表示させるPCやゲーム機（ビデオゲームコンソール等）であってもよい。

[0027] コンピュータ10は、SOC（System On a Chip）11とメインメモリ22とを備える。SOC11は、1つのチップ上にデータ処理システムの機能が実装された集積回路製品である。SOC11は、CPU12、GPU14、ビデオエンコーダ16、ビデオパイプライン18、ビデオタイミングジェネレータ20を備える。

[0028] CPU12は、並列処理の対象となる複数のアプリケーションを時分割で実行する。また、CPU12は、並列処理の対象となる複数のアプリケーションを管理するソフトウェア（以下「管理ソフトウェア」とも呼ぶ。）を実行する。管理ソフトウェアは、OSやミドルウェア等、アプリケーションより下のレイヤに位置づけられるソフトウェアであってもよく、実施例では、複数のアプリケーションの実行順序やコンテキストスイッチを管理する。なお、管理ソフトウェアの一部の機能は、GPU14やビデオエンコーダ16

等（後述の処理部46）により実行されてもよい。

[0029] GPU14は、CPU12において時分割で実行される複数のアプリケーションの指示に応じて、複数のアプリケーションに関する画像処理や汎用計算処理を時分割で実行する。例えば、GPU14は、複数のアプリケーションそれぞれの画像データを生成し、生成した画像データを、各アプリケーションに対応するフレームバッファに書き込む。

[0030] ビデオエンコーダ16は、並列処理対象の複数のアプリケーションの画像の圧縮処理を時分割で実行する。例えば、ビデオエンコーダ16は、複数のアプリケーションに対応する複数のフレームバッファから各アプリケーションの画像データを順次読み出し、読み出した画像データに対する圧縮符号化処理を実行する。

[0031] ビデオパイプライン18は、画像供給元他ブロック（例えばGPU14または不図示のビデオデコーダ等）から供給された画像であり、言い換えれば、並列処理対象の複数のアプリケーションの画像に対する、解像度変換、画質変換または多重化を時分割で実行する。

[0032] GPU14、ビデオエンコーダ16、ビデオパイプライン18間での画像受け渡しのケースは、以下の3パターンを含む。

(1) GPU14→ビデオパイプライン18→ビデオエンコーダ16。

(2) GPU14→ビデオエンコーダ16。

(3) 不図示他ブロック（ビデオデコーダ等）→ビデオパイプライン18→ビデオエンコーダ16。ただし、ビデオパイプライン18には、GPU14の処理結果も入力される。

すなわち、ビデオエンコーダ16が、(A) CPU12が生成した画像を直接参照するパターンと、(B) ビデオパイプライン18が加工後の画像を参照するパターンのいずれもある。

[0033] このように、GPU14、ビデオエンコーダ16、ビデオパイプライン18は、並列処理の対象となる複数のアプリケーションに関する処理を時分割で実行するものであり、以下、GPU14、ビデオエンコーダ16、ビデオ

パイプライン18を総称して「処理部46」と呼ぶ。

- [0034] ビデオタイミングジェネレータ20は、画像の表示に関する各種タイミング（ビデオタイミングとも言える）を各機器に通知する。実施例では、ビデオタイミングジェネレータ20は、表示装置の垂直帰線期間（もしくは垂直帰線期間に対応する期間）の開始を通知する信号をCPU12、GPU14、ビデオエンコーダ16、ビデオパイプライン18へ送信する。
- [0035] また、SOC11は、CSDMAエンジン40、CSDMAエンジン42、CSDMAエンジン44（総称する場合「CSDMAエンジン48」と呼ぶ。）をさらに備える。CSDMAエンジン48は、コンテキストスイッチのためのコンテキスト転送をDMAにより実行するデータ転送部である。
- [0036] CSDMAエンジン40は、GPU14におけるコンテキストスイッチに伴いGPU14とメインメモリ22間でコンテキストを転送する。CSDMAエンジン42は、ビデオエンコーダ16におけるコンテキストスイッチに伴いビデオエンコーダ16とメインメモリ22間でコンテキストを転送する。CSDMAエンジン44は、ビデオパイプライン18におけるコンテキストスイッチに伴いビデオパイプライン18とメインメモリ22間でコンテキストを転送する。
- [0037] パス50は、GPU14による通常のメモリアクセス用（コマンドの読み込みおよび画像データの格納を含む）のパスである。パス52は、CPU12で実行されるソフトウェア（例えば管理ソフトウェア）からGPU14およびCSDMAエンジン40へのアクセスパス（設定・確認・通知・制御用）である。パス52は、同様の目的で他ブロックにも接続されている。なお、図中の線の交点に黒丸がある箇所は線が接続されている状態を示し、線の交点に黒丸がない箇所は線が接続されていない状態を示す。パス54は、メインメモリ22からGPU14へのコンテキスト設定パスである。パス56は、GPU14からメインメモリ22へのコンテキスト退避パスである。
- [0038] パス58は、CSDMAエンジン40がGPU14の処理状態を監視するパスである。パス60は、CSDMAエンジン40がGPU14へアプリケ

ーション処理の停止と開始（再開）を指示するパスである。パス62は、ビデオタイミングジェネレータ20からGPU14およびCSDMAエンジン40へ垂直帰線期間の開始を通知するパスである。パス62は、同様の目的で他ブロックにも接続されている。ビデオエンコーダ16とCSDMAエンジン42間の各パス、ビデオパイプライン18とCSDMAエンジン44間の各パスも同様である。

[0039] メインメモリ22は、SOC11により参照または更新されるデータを記憶する。メインメモリ22は、GPUコマンドバッファ30、GPUコンテキスト32、ビデオエンコーダコンテキスト34、ビデオパイプラインコンテキスト36を記憶する。

[0040] GPUコンテキスト32は、SOC11において並列実行される複数のアプリケーション（例えばApp A、App B等）に関するGPU14で使用されるコンテキストである。同様に、ビデオエンコーダコンテキスト34は、SOC11において並列実行される複数のアプリケーションに関するビデオエンコーダ16で使用されるコンテキストである。ビデオパイプラインコンテキスト36は、SOC11において並列実行される複数のアプリケーションに関するビデオパイプライン18で使用されるコンテキストである。

[0041] GPUコマンドバッファ30は、SOC11で並列処理される複数のアプリケーションに対応する複数のキュー（例えばApp A用App B用等）を含む。GPUコマンドバッファ30の各キューには、各アプリケーションに関する処理内容を指示するコマンドがCPU12により書き込まれ、蓄積される。言い換えれば、CPU12により実行される各アプリケーションは、自アプリケーションに対応するGPUコマンドバッファ30のキューに、描画指示等のコマンドを格納する。GPU14は、実行対象のアプリケーションに対応するGPUコマンドバッファ30のキューからコマンドを読み出す。GPU14は、実行対象のアプリケーションを切り替える際に、コマンドを読み出すキューを切り替える。

[0042] このように、GPUコマンドバッファ30を設けることで、CPU12は、処理部46（例えばGPU14、ビデオエンコーダ16、ビデオパイプライン18）と非同期に動作することができる。すなわち、CPU12において並列に（時分割等で）実行される各アプリケーションは、処理部46が他アプリケーションの処理をしている間も、自アプリケーションに関するコマンドをGPUコマンドバッファ30に随時格納することができる。言い換えれば、GPUコマンドバッファ30に書き込みを行うCPU12上のソフトウェアが、処理部46と非同期に動作する場合にも対応できる。これにより、処理部46におけるコンテキストスイッチ実行中も、CPU12上の各アプリケーションは、処理の指示をGPUコマンドバッファ30に積み上げることができるため、処理の遅延を抑制することができる。

[0043] なお、実施例のSOC11では、CPU12で実行される管理ソフトウェアが、コンテキストスイッチの初期設定を行う。例えば、管理ソフトウェアは、並列処理の対象となる複数のアプリケーションのそれぞれについて、GPUコンテキスト32、ビデオエンコーダコンテキスト34、ビデオパイプラインコンテキスト36の領域をメインメモリ22に確保する。また、管理ソフトウェアは、各アプリケーションのコンテキスト格納位置（アドレス等）をCSDMAエンジン48（CSDMAエンジン40、CSDMAエンジン42、CSDMAエンジン44）に通知する。

[0044] また、管理ソフトウェアは、並列処理の対象となる複数のアプリケーションの実行順序を、処理部46とCSDMAエンジン48に通知する。処理部46とCSDMAエンジン48は、管理ソフトウェアからの通知や設定に基づいて、今回実行対象となるアプリケーションや次回実行対象となるアプリケーションを把握する。

[0045] 図1には不図示だが、変形例として、CPU12にもコンテキストスイッチのためのCSDMAエンジン48を設けてもよい。この場合、CPU12は、処理部46と同期して動作してもよい。

[0046] 図2は、図1の処理部46（具体的にはGPU14、ビデオエンコーダ1

6、ビデオパイプライン18)とCSDMAエンジン48の詳細な構成を示すブロック図である。図2の処理部46とCSDMAエンジン48の組は、図1のGPU14とCSDMAエンジン40の組、ビデオエンコーダ16とCSDMAエンジン42の組、ビデオパイプライン18とCSDMAエンジン44の組のそれぞれに対応する。すなわち、図2の処理部46の構成は、GPU14、ビデオエンコーダ16、ビデオパイプライン18の少なくとも1つに適用可能である。また、図2のCSDMAエンジン48の構成は、CSDMAエンジン40、CSDMAエンジン42、CSDMAエンジン44の少なくとも1つに適用可能である。

[0047] 本開示のブロック図において示される各ブロックは、ハードウェア的には、コンピュータのCPU・メモリをはじめとする素子や機械装置で実現でき、ソフトウェア的にはコンピュータプログラム等によって実現されるが、ここでは、それらの連携によって実現される機能ブロックを描いている。これらの機能ブロックはハードウェア、ソフトウェアの組合せによっていろいろなかたちで実現できることは、当業者には理解されることである。

[0048] 処理部46は、第1機能回路70、第2機能回路71を備える。第1機能回路70と第2機能回路71は、実行対象のアプリケーションに関するデータ処理（例えば画像生成処理や圧縮伸長処理）を実行する。処理部46は、CPU12上の管理ソフトウェアからの通知や設定等に基づいて、実行対象のアプリケーションを識別し、実行対象のアプリケーションに第1機能回路70および第2機能回路71を割り当てる。

[0049] 処理部46は、実行対象のアプリケーションに関する各種データをメインメモリ22から読み出して、第1機能回路70と第2機能回路71に入力する。例えば、処理部46としてのGPU14は、実行対象のアプリケーションに対応するGPUコマンドバッファ30のキューから、当該アプリケーションに関するコマンドを読み出す。また、GPU14は、画像描画に必要な他のデータもメインメモリ22から読み出す。

[0050] 第1機能回路70は、アプリケーションの実行中に更新されるデータであ

って、処理の再現に必要なデータであるコンテキストを保持する回路である。また、第1機能回路70は、実行対象のアプリケーションに応じて、コンテキストの入れ替えが必要な回路である。第1機能回路70は、ステート・マシン72、レジスタ73、ワークメモリ74を含む。ステート・マシン72は、ステート・レジスタを含み、処理部46で実行中の各処理の現在状態を保持し、言い換えれば、状態遷移の状況を保持する。レジスタ73は、設定および処理中のデータや処理の結果を保持する。ワークメモリ74は、処理部46のメモリの中で、処理に応じて更新される内部データやデスク립タおよびマイクロコードを保持する領域である。

[0051] 第2機能回路71は、アプリケーションの実行中に更新されるデータであって、処理の再現に必要なデータであるコンテキストを保持しない回路である。また、第2機能回路71は、初期化が不要な回路もしくは一括でのリセットが可能な回路であり、例えば演算処理等を行う回路である。第2機能回路71は、ランダムロジック75、演算器76、ワークメモリ77を含む。ランダムロジック75は、ハードワイヤードによる機能を含み、初期化が不要もしくは一定の初期化が可能なフリップフロップ回路（ラッチ）を含む。演算器76は、データパスおよび初期化が不要もしくは一定の初期化が可能なフリップフロップ回路（ラッチ）を含む。ワークメモリ77は、処理部46のメモリの中でワークメモリ74以外の領域である。

[0052] CSDMAエンジン48は、検出部80、監視部82、指示部84、転送実行部86を含む。検出部80は、処理部46における実行対象のアプリケーションの切替タイミング（以下「App切替タイミング」とも呼ぶ。）を検出する。実施例の検出部80は、垂直帰線期間の開始タイミングをApp切替タイミングとして検出し、具体的には、ビデオタイミングジェネレータ20から垂直帰線期間の開始タイミングが通知されたことをもってApp切替タイミングであると判断する。

[0053] 変形例として、検出部80は、水平帰線期間の開始タイミングをApp切替タイミングとして検出してもよい。具体的には、検出部80は、ビデオタ

イミングジェネレータ20から水平帰線期間の開始タイミングが通知されたことをもってApp切替タイミングであると判断してもよい。別の変形例として、CPU12上の管理ソフトウェアが、App切替タイミングを処理部46およびCSDMAエンジン48に通知してもよい。検出部80は、CPU12上の管理ソフトウェアからの通知をもとにApp切替タイミングを検出してもよい。

[0054] 監視部82は、App切替タイミングが検出された場合、処理部46におけるアプリケーションに関する処理の実行状態を監視する。処理部46におけるアプリケーションに関する処理は小さい粒度の複数の内部処理を含み、複数の内部処理が並列して実行される。監視部82は、個々の内部処理の実行状態（例えば個々の内部処理が完了したか否か）を確認する。

[0055] 指示部84は、アプリケーションに関する処理の停止と開始（再開）を処理部46に指示する。指示部84は、App切替タイミングが検出された場合に、アプリケーションに関する処理の停止を処理部46に指示してもよい。処理部46は、CSDMAエンジン48の指示部84からの指示をもとにApp切替タイミングを検出してもよく、CPU12上の管理ソフトウェアまたはビデオタイミングジェネレータ20からの通知をもとにApp切替タイミングを検出してもよい。なお、監視部82と指示部84はオプションな構成であり、CSDMAエンジン48は、監視部82と指示部84の一方または両方を含まない構成であってもよい。

[0056] 転送実行部86は、App切替タイミングが検出された場合、コンテキスト転送処理を実行する。具体的には、転送実行部86は、コンテキスト転送処理として、複数のアプリケーションを管理するソフトウェア（実施例ではCPU12上の管理ソフトウェア）の処理を介さずに、処理部46で実行されていたアプリケーションのコンテキストを処理部46（第1機能回路70）からメインメモリ22（当該アプリケーションのコンテキスト記憶領域）へ退避させる。

[0057] また、転送実行部86は、コンテキスト転送処理として、複数のアプリケ

ーションを管理するソフトウェア（実施例ではCPU 12上の管理ソフトウェア）の処理を介さずに、処理部46で次に実行されるアプリケーションのコンテキストをメインメモリ22（当該アプリケーションのコンテキスト記憶領域）から読み出す。転送実行部86は、読み出したコンテキストを処理部46（第1機能回路70）に設定する。指示部84は、転送実行部86によるコンテキスト設定処理の完了を検出した場合、アプリケーションに関する処理の開始（再開）を処理部46に指示してもよい。

[0058] 転送実行部86による転送の対象となるコンテキストは、処理部46におけるコンテキストスイッチに伴い中断された時点のアプリケーションの処理（上述の個々の内部処理）の再現に必要なデータである。実施例では、転送実行部86による転送の対象となるコンテキストは、処理部46の第1機能回路70に保持されたコンテキストの全てであって、ソフトウェア（例えばCPU 12上のソフトウェアおよび処理部46上のソフトウェア）からアクセスできないコンテキストを含む。ソフトウェアからアクセスできないコンテキストは、例えば、処理部46のステート・マシン72に保持されるコンテキストであってもよく、アプリケーションの処理に含まれる個々の内部処理の状態遷移の状況を示すコンテキストであってもよい。

[0059] 図3は、処理部46（例えばGPU 14）におけるビデオ描画処理を時系列に示す。同図は、4つのアプリケーション（App A、App B等）に関するビデオ描画処理を時分割で実行する例を示している。本実施例において、各アプリケーションは、60fps（frames per second）で画像を生成する必要がある。処理部46は、4つのアプリケーションの画像を各々1/240秒で生成することにより、1つのアプリケーションあたり60fpsのフレームレートを実現する。

[0060] しかし、各アプリケーションの処理時間は描画内容に基づいて常に変動しており、必ずしも1/240秒以内で処理を完了できるとは限らない。このようなとき、既述したように、実施例では、垂直帰線期間の開始タイミング（図3のタイミング90）がApp切替タイミング（言い換えれば画像書き

込み先のフレームバッファを切り替えるフリップタイミング)として検出される。ここで、画像の生成がまだ完了していない場合も、次のアプリケーションへ処理部46を明けわたすことで、各アプリケーションが処理部46を使用できる時間を均等にできる。なお、タイミング90までに画像の生成ができなかった場合、コマ落ちとなり、該当アプリケーションの前画像が繰り返し後段処理に渡される。

[0061] 図3の期間92は、垂直帰線期間に相当し、CSDMAエンジン48を用いることで、処理の再現に必要な全コンテキストの退避・設定をこの期間以内で高速に実行する。このようにして、矢印で示す画像生成期間とコンテキストスイッチを含む期間92（垂直帰線期間）との合計が1/240秒以内になるように制御される。

[0062] なお、アプリケーションが垂直帰線期間の開始タイミングより早く画像の生成を完了できたとき、アプリケーションはフレームバッファを切り替えるフリップの実行が可能であることをCPU12上の管理ソフトウェアへ通知してもよい。そして、CPU12上の管理ソフトウェアは、垂直帰線期間の開始タイミングに代わり、App切替タイミングを自ら処理部46へ通知してもよい。これにより該当アプリケーションの後段処理や、後続のアプリケーションは前倒しで処理を開始し、遅延の削減や、処理部の利用効率を高めることができる。

[0063] なお、変形例では、水平帰線期間の開始タイミングがApp切替タイミングとして検出されてもよく、水平帰線期間内にコンテキストスイッチを行ってもよい。

[0064] 図4は、ビデオタイミングの例を示す。同図は、垂直帰線期間94（Vblank）と水平帰線期間96（Hblank）を示している。なお、サーバがビデオを描画してそのビデオをクライアントへ配信する場合、ビデオタイミングを使用する外部ディスプレイがサーバに接続されないことがある。その場合も、従来のアプリケーションとの互換維持のため、ビデオタイミングが生成されることがあり、実施例ではそのビデオタイミングを利用してA

pp切替タイミングを検出する。

[0065] 垂直帰線期間および水平帰線期間には、GPU14やビデオエンコーダ16、ビデオパイプライン18が処理を一時停止することもできる。実施例では、短時間の垂直帰線期間または水平帰線期間の中でコンテキストスイッチを実行することで、コンテキストスイッチのオーバーヘッドを隠蔽することができる。

[0066] 次に、コンテキストの転送経路について説明する。

図5は、SOC11の構成を示す。SOC11内の各回路は、通常、バスファブリック100およびメモリアンタフェース102を介してメインメモリ22にアクセスする。SOC11（すなわち半導体チップ）には、機能が正しく動作するか否かをテストするためのDFT（Design For Test）機能（DFT制御回路104等）が内蔵されている。また、BIST（Built-In Self Test）は、自己診断機能を持ち、ロジック回路用とメモリ回路用とが設けられる。スキャンチェーン106は、SOC11内の各回路内部へ網羅的に接続され、BISTやチップ外部の指示に基づいて各回路をテストするために使用される。スキャンチェーン106を用いることで、SOC11内部の各状態値の設定および読み出しができる。

[0067] 図6も、SOC11の構成を示す。同図に示すように、CSDMAエンジン48は、通常の実データ転送用のバス（バスファブリック100）と、バス114とを使用して、コンテキストの退避および設定を行うことができる。バス114は、ソフトウェアからはアクセスできないコンテキストを保持する回路にアクセスするためのバスである。しかし、この構成では、コンテキストの転送中に、他の回路（CPU12等）とバスファブリック100を共有するため、通常の実データ転送が阻害されてしまうというデメリットがある。

[0068] 図7も、SOC11の構成を示す。同図に示すように、CSDMAエンジン48は、コンテキスト転送用の専用バス（専用経路108）と、バス114とを使用して、コンテキストの退避および設定を行うこともできる。しか

し、この構成では、専用経路108を新たにSOC11に実装する必要があり、SOC11のコストが増大するというデメリットがある。

[0069] 図8も、SOC11の構成を示す。同図に示すように、実施例のCSDMAエンジン48は、DF T用の回路（スキャンチェーン106を含む）およびBIST用の回路（BIST回路110）を使用して、コンテキストの退避および設定を行う。例えば、CSDMAエンジン48は、スキャンチェーン106を介して第1機能回路70から今回実行されたアプリケーションに関するコンテキストを読み出し、読み出したコンテキストをBIST回路110を介してメインメモリ22に保存する。また、CSDMAエンジン48は、次回実行されるアプリケーションに関するコンテキストをBIST回路110を介してメインメモリ22から読み出し、読み出したコンテキストをスキャンチェーン106を介して第1機能回路70に設定する。

[0070] このようにスキャンチェーン106を使用することで、ソフトウェアがアクセスするパスがない内部回路のコンテキストにもアクセスすることができ、処理の再現に必要なコンテキストの退避と設定が可能になる。また、バスファブリック100、専用経路108、パス114のいずれも新たに設ける必要がなく、低コストでコンテキストの転送を実現できる。さらにまた、コンテキストの転送によりSOC11における通常のリデータ転送を阻害してしまうことを回避できる。

[0071] 次に、コンテキストスイッチの開始タイミングについて説明する。

CSDMAエンジン48の転送実行部86は、処理部46の第1機能回路70（ステート・マシン72等）に保持されたアプリケーションのコンテキストをメインメモリ22に退避させる。処理部46からメインメモリ22に退避されるアプリケーションのコンテキスト、および、メインメモリ22から処理部46に設定されるアプリケーションコンテキストは、ステート・マシン72等に保持されたデータであって、コンテキストスイッチ開始時点で処理部46において未完了の処理（小さい粒度の内部処理）の現在状態を示すデータを含む。

[0072] 処理部46の第1機能回路70および第2機能回路71は、App切替タイミングが検出された場合、それまで実行対象であったアプリケーションの内部処理の状態にかかわらず、言い換えれば、内部処理が未完了であっても、即時に内部処理を停止する。CSDMAエンジン48の転送実行部86は、App切替タイミングが検出された場合、それまで実行対象であったアプリケーションの内部処理の状態にかかわらず、言い換えれば、内部処理が未完了であっても、コンテキストの転送処理を開始する。

[0073] 図9Aは、第1実施例におけるコンテキストスイッチ時の内部処理の例を示す。同図は、横方向が時間経過軸であり、1つの矢印は、1つのアプリケーションに関する処理に含まれる小さな粒度の内部処理を示す。各矢印の左端が内部処理の開始タイミングを示し、右端が内部処理の終了タイミングを示している。

[0074] 図9Aに示すように、実施例の処理部46は、コンテキストスイッチ実行指示を受け付けた時点（すなわちApp切替タイミングを検出した時点）で、即時に全ての内部処理を停止し、コンテキストスイッチを開始する。また、その時点で未完了（すなわちオンザフライ・アウトスタンディング）の内部処理（図中の破線）も即時に中断し、その時点のコンテキストをメインメモリ22に保存する。なお、図中の一点鎖線で示す内部処理は、コンテキストスイッチ実行指示の受付後に開始予定のものであるため、開始されない。CSDMAエンジン48の転送実行部86は、コンテキストスイッチ実行指示を受け付けた時点（すなわちApp切替タイミングを検出した時点）で、即時にコンテキストの転送処理（退避および設定）を開始する。

[0075] 実施例のSOC11では、処理部46における各内部処理の状態であり、すなわち、ソフトウェアからは確認できないデータ（例えばステート・マシン72に保持されるデータ等）を含むコンテキストを退避させる。これにより、コンテキストスイッチ実行指示受付時点で実行途中だった内部処理は、次回の実行時に、その状態を復元して再開することができる。したがって、実施例のSOC11では、コンテキストスイッチ実行指示の受付時点（App

p切替タイミングの検出時点)で即時にコンテキストスイッチを開始でき、処理の遅延を抑制することができる。

[0076] 変形例として、CSDMAエンジン48の転送実行部86は、App切替タイミングが検出された場合、処理部46におけるアプリケーションの内部処理が完了したことが監視部82により確認されるまで待機し、処理部46におけるアプリケーションの内部処理が完了したことが確認された後にコンテキストの転送処理を開始してもよい。

[0077] 図9Bは、変形例におけるコンテキストスイッチ時の内部処理の例を示す。同図でも、コンテキストスイッチ実行指示時に開始済だが未完了(すなわちオンザフライ・アウトスタンディング)の内部処理を破線で示している。本変形例では、コンテキストスイッチ実行指示時に開始済だが未完了の内部処理が終了するまで待ってからコンテキストスイッチを開始する。

[0078] コンテキストスイッチ実行指示時に開始済だが未完了の処理を中断させずに完了させることにより、コンテキストスイッチ対象ブロック外との処理連携が求められる場合や、リアルタイム性が求められる場合において、処理破綻(不整合の発生等)を防止できる。また、転送対象となるコンテキストのデータ量を低減することができる。

[0079] <第2実施例>

本実施例に関して、これまでの実施例と相違する点を中心に以下説明し、共通する点の説明を省略する。本変形例の構成要素のうちこれまでの実施例の構成要素と同一または対応する構成要素には同一の符号を付して説明する。

[0080] 図10は、第2実施例の処理部46の構成を示す。図10の処理部46とCSDMAエンジン48の組は、図1のGPU14とCSDMAエンジン40の組、ビデオエンコーダ16とCSDMAエンジン42の組、ビデオパイプライン18とCSDMAエンジン44の組のそれぞれに対応する。すなわち、図10の処理部46の構成は、GPU14、ビデオエンコーダ16、ビデオパイプライン18の少なくとも1つに適用可能である。また、図10の

CSDMAエンジン48の構成は、CSDMAエンジン40、CSDMAエンジン42、CSDMAエンジン44の少なくとも1つに適用可能である。

[0081] 第2実施例の処理部46は、アプリケーションのコンテキストを保持する第1機能回路70を複数含む。図10の処理部46は、2つの第1機能回路70（第1機能回路70aと第1機能回路70b）を備えるが、3つ以上の第1機能回路70を備えてもよい。

[0082] 第2実施例のCSDMAエンジン48の転送実行部86は、処理部46が或る第1機能回路70に保持されたアプリケーションのコンテキストを用いて当該アプリケーションの処理を実行する間に、別の第1機能回路70に保持された別のアプリケーションのコンテキストを対象としてコンテキスト転送処理を実行する。

[0083] 例えば、処理部46が第1機能回路70aに保持されたApp Aのコンテキストを用いてApp Aの処理を実行する間に、転送実行部86は、（1）第1機能回路70bに保持されたApp D（すなわち前回実行したApp）のコンテキストをメインメモリ22に退避させる。それとともに、転送実行部86は、（2）メインメモリ22に保持されたApp B（すなわち次に実行対象となるApp）のコンテキストを読み出して、第1機能回路70bに設定する。

[0084] 処理部46は、第1機能回路70aに保持されたアプリケーションのコンテキストを用いて当該アプリケーションの処理を実行中にApp切替タイミングを検出すると、即時に当該アプリケーションの処理を中止する。そして、処理部46は、第1機能回路70bに保持された別のアプリケーションのコンテキストを用いて当該別のアプリケーションの処理を開始する。これにより、コンテキストスイッチに伴う待機時間（すなわちコンテキストの退避と設定を待つ時間）をほぼゼロにすることができる。

[0085] <第3実施例>

本実施例に関して、これまでの実施例と相違する点を中心に以下説明し、共通する点の説明を省略する。本変形例の構成要素のうちこれまでの実施例

の構成要素と同一または対応する構成要素には同一の符号を付して説明する。

[0086] 第3実施例の処理部46の構成は、図10に示した第2実施例の処理部46の構成と同じである。処理部46は、第1機能回路70aに保持された第1のアプリケーション（例えばApp A）のコンテキストを用いて第1のアプリケーションに関する処理を実行する。また、処理部46は、第1機能回路70bに保持された第2のアプリケーション（例えばApp B）のコンテキストを用いて第2のアプリケーションに関する処理を実行する。

[0087] 処理部46は、App切替タイミングを検出すると、CPU12上の管理ソフトウェアによる事前設定等をもとに、次に実行すべきアプリケーションを識別する。ここでは、処理部46における実行対象が第1のアプリケーション（例えばApp A）から第2のアプリケーション（例えばApp B）に切り替わることとする。処理部46は、第1のアプリケーションに関する処理のうち第2のアプリケーションに関する処理と混流（言い換えれば並列実行または同時実行）不可能な処理が完了後に、第1のアプリケーションに関する処理のうち第2のアプリケーションに関する処理と混流可能な処理と、第2のアプリケーションに関する処理とを並列実行する。

[0088] 図11は、第3実施例におけるコンテキストスイッチ時の内部処理の例を示す。同図は、上記第1のアプリケーション（切替前のアプリケーション）の内部処理を示す。同図の破線および二点鎖線は、コンテキストスイッチ実行指示時に開始済だが未完了（オンザフライ・アウトスタンディング）の内部処理を示す。このうち破線は、第2のアプリケーションに関する処理と混流不可能な内部処理を示し、二点鎖線は、第2のアプリケーションに関する処理と混流可能な内部処理を示している。

[0089] 混流可能な処理は、例えばデータパスや演算器を使用する内部処理であって、時間的に前後の処理内容に対して依存関係がなく、かつ、空間的に回路として周辺に接続された機能の処理内容に対して依存関係がない内部処理である。混流不可能な処理は、例えば、上記の依存関係を有する内部処理であ

り、また、第2のアプリケーションに関する処理と並列実行すると破綻する可能性がある内部処理である。処理部46は、各内部処理が、他のアプリケーションの内部処理と混流可能か否かを示すデータを予め記憶してもよく、また、CPU12上の管理ソフトウェアがそのデータを処理部46に予め格納してもよい。

[0090] 処理部46は、コンテキストスイッチ実行指示を受け付けても、第1のアプリケーションに関するオンザフライ・アウトスタンディングの内部処理を継続する。処理部46は、オンザフライ・アウトスタンディングの内部処理のうち第2のアプリケーション（言い換えれば他のアプリケーション）と混流不可能な処理（図11の破線）が終了するまで待ってから、コンテキストスイッチを開始する。言い換えれば、処理部46は、オンザフライ・アウトスタンディングの内部処理のうち第2のアプリケーションと混流不可能な処理が全て終了した場合、第2のアプリケーションと混流可能な処理（図11の二点鎖線）が終了したか否かにかかわらず、コンテキストスイッチを開始する。

[0091] コンテキストスイッチ実行後、処理部46（例えば第1機能回路70a）は、第1のアプリケーションに関するオンザフライ・アウトスタンディングの内部処理のうち第2のアプリケーションと混流可能な内部処理を継続して実行する。それと並行して、処理部46（例えば第1機能回路70b）は、第2のアプリケーションに関する内部処理を実行する。混流可能な内部処理が全て終了すると、第1のアプリケーションのコンテキストを保持する回路（例えば第1機能回路70a）には実行結果の状態が記録される。

[0092] CSDMAエンジン48の転送実行部86は、第1のアプリケーションに関するオンザフライ・アウトスタンディングの内部処理が全て完了したことが監視部82により検知されると、第1のアプリケーションのコンテキストを保持する回路（例えば第1機能回路70a）からメインメモリ22に第1のアプリケーションのコンテキストを退避させる。また、転送実行部86は、処理部46において次に実行予定の第3のアプリケーションのコンテキス

トをメインメモリ22から上記回路（例えば第1機能回路70a）に設定する。

[0093] 第3実施例のSOC11によると、第1のアプリケーションに関する処理の破綻を防止しつつ、第2のアプリケーションに関する処理を早期に開始することができ、処理部46における処理の遅延を抑制することができる。

[0094] 図12A、図12B、図12Cは、コンテキストスイッチにおける処理部46とCSDMAエンジン48の動作を示す。図12Aは、第1実施例の処理部46の動作（即時コンテキストスイッチ）を示している。処理部46は、前App処理と次App処理を実行し、CSDMAエンジン48は、前Appのコンテキスト退避処理と次Appのコンテキスト設定処理を実行する。図12Bは、第1実施例の変形例に記載した処理部46の動作（全ての内部処理完了後のコンテキストスイッチ）を示している。前App処理の終了を待つ時間だけ、コンテキストの退避処理の開始が遅れることになる。

[0095] 第2実施例に記載したようにコンテキストを保持する第1機能回路70が多重化される場合、図12Aと図12Bのいずれに示すコンテキストスイッチ方法でも、一方のコンテキスト保持回路を用いてアプリケーションを実行中に、他方のコンテキスト保持回路に次に実行予定のアプリケーションのコンテキストを事前に設定することができる。これにより、コンテキストの退避と設定の完了を待つ時間（図12Aと図12Bの期間112）をほぼゼロにすることができる。

[0096] 図12Cは、第3実施例の処理部46の動作（混流不可能な内部処理完了後のコンテキストスイッチ）を示している。図12Cの「コンテキストスイッチ開始・終了」は、次Appを実行するために、参照先をそれまでとは異なるコンテキスト保持回路に切り替えることを示している。例えば、参照先を第1機能回路70aから第1機能回路70bに切り替えることを示している。既述したように、前Appの混流可能な内部処理が終了すると、CSDMAエンジン48は、前Appのコンテキスト保持回路（例えば第1機能回路70a）からメインメモリ22へ前Appのコンテキストを退避させる。

[0097] <第4実施例>

本実施例に関して、これまでの実施例と相違する点を中心に以下説明し、共通する点の説明を省略する。本変形例の構成要素のうちこれまでの実施例の構成要素と同一または対応する構成要素には同一の符号を付して説明する。

[0098] 図13は、第4実施例の処理部46の構成を示す。第4実施例の処理部46は、図10に示した第3実施例の構成に加えて、複数のアプリケーション間で混流不可能な内部処理を実行する機能ブロックを複数備える。複数のアプリケーション間で混流不可能な内部処理を実行する機能ブロックは、例えば、(1)「時間的に前後の処理内容や、空間的に回路として周辺に接続された機能の処理内容」に対して依存関係がある処理を実行する回路であってもよい。または、(2)各アプリケーションの別設定を前提とした内部処理が投入されると、両アプリケーションの設定が共通でないことによる内処理の破綻が起こる回路であってもよい。

[0099] 図13に示す処理部46は、複数のアプリケーション間で混流不可能な内部処理を実行する機能ブロックの例として、ランダムロジック75を複数備える。図13の処理部46は、2つのランダムロジック75（ランダムロジック75aとランダムロジック75b）を備えるが、3つ以上のランダムロジック75を備えてもよい。

[0100] App切替タイミングが検出され、処理部46における処理対象を第1のアプリケーション（例えばApp A）から第2のアプリケーション（例えばApp B）に切り替えるべき際、処理部46は、第1のアプリケーションに関するオンザフライ・アウトスタンディングの内部処理のうち第2のアプリケーションに関する処理と混流不可能な内部処理を、一方のランダムロジック75（例えばランダムロジック75a）を用いて継続実行する。また、処理部46は、上記混流不可能な内部処理の完了を待たず、即時に、他方のランダムロジック75（例えばランダムロジック75b）を用いて、第2のアプリケーションに関する処理を開始する。

- [0101] 第4実施例のSOC 11によると、第1のアプリケーションに関するオンザフライ・アウトスタンディングの内部処理のうち第2のアプリケーションに関する処理と混流不可能な内部処理の完了を待つ必要がない。これにより、コンテキストスイッチ実行指示時に、第2のアプリケーションに関する処理を早期に開始することができ、コンテキストスイッチを一層高速に実現できる。
- [0102] 第4実施例の変形例を説明する。処理部46は、複数の多重化された機能回路に対して複数のアプリケーションを同時に割り当ててもよく、処理部46は、複数のアプリケーションの内部処理を並列実行してもよい。ここで、機能回路の多重化が2重であり、並列処理対象のアプリケーションが4つ（例えばApp A～D）の場合、選択された2つのアプリケーション（例えばApp AとApp B）を2つの多重化された機能回路に割り当ててもよい。コンテキストスイッチ時には、残りの2つのアプリケーション（例えばApp CとApp D）を新たな実行対象として、それまで実行した2つのアプリケーション（例えばApp AとApp B）と入れ替えてもよい。
- [0103] 例えば、図13に示す構成において、App Aの内部処理とApp Bの内部処理を並列実行する場合、処理部46は、App Aのコンテキストを第1機能回路70aに保持させ、App Bのコンテキストを第1機能回路70bに保持させてもよい。また、処理部46は、App Aに関する内部処理のうちApp Bに関する内部処理と混流不可能な内部処理をランダムロジック75aに割り当て、App Bに関する内部処理のうちApp Aに関する内部処理と混流不可能な内部処理をランダムロジック75bに割り当ててもよい。また、処理部46は、App Aに関する内部処理のうちApp Bに関する内部処理と混流可能な内部処理と、App Bに関する内部処理のうちApp Aに関する内部処理と混流可能な内部処理の両方を共通の演算器76とワークメモリ77に割り当ててもよい。
- [0104] この変形例の構成によると、処理部46が複数のアプリケーションを時分

割に実行するのではなく同時に実行するため、処理の遅延を一層抑制できる。また、コンテキストスイッチの発生回数も抑制できる。また、処理部46が有する回路のアクティブ率（言い換えれば活性化率）が向上し、データ処理の性能を高めることができる。

[0105] <第5実施例>

本実施例に関して、これまでの実施例と相違する点を中心に以下説明し、共通する点の説明を省略する。本変形例の構成要素のうちこれまでの実施例の構成要素と同一または対応する構成要素には同一の符号を付して説明する。

[0106] 図14は、第5実施例の処理部46の構成を示す。第5実施例の処理部46では、コンテキストを保持する第1機能回路70と、コンテキストを保持しない第2機能回路71の両方を多重化する。図14の処理部46は、4つの第1機能回路70と4つの第2機能回路71を備えるが、第1機能回路70と第2機能回路71の多重化は4重に制限されない。

[0107] 本実施例の構成において、各アプリケーションの性能要件が異なる場合、処理部46は、1つのアプリケーションに対して何個の第1機能回路70および第2機能回路71を割り当てるかをコンテキストスイッチのたびに決定する。そして、処理部46は、実行対象となる各アプリケーションに1つ以上の第1機能回路70と1つ以上の第2機能回路71を割り当てる。各アプリケーションの性能要件、または、各アプリケーションが必要とする第1機能回路70と第2機能回路71の個数は、CPU12上の管理ソフトウェアが処理部46に設定してもよい。

[0108] 図15Aと図15Bは、アプリケーションの割当例を示す。図15Aに示すように、処理部46は、4つのアプリケーション（例えばApp A～D）のそれぞれを、1つの第1機能回路70と1つの第2機能回路71の組に割り当ててもよい。また、App Eに関するデータ処理に、App Aに関するデータ処理の4倍の性能が必要である場合、図15Bに示すように、処理部46は、App Eに対して、4つの第2機能回路71と1つの第1

機能回路 70 を割り当ててもよい。App E の実行時、3 つの第 1 機能回路 70 は未使用であってもよい。

[0109] 処理部 46 は、図 15 A に示す割当での処理と、図 15 B に示す割当での処理とを時分割で実行してもよい。この場合、第 1 実施例から第 4 実施例で説明したコンテキストスイッチ方法のいずれかを適用することで、図 15 A の状態と図 15 B の状態との間での遷移を高速に実現できる。

[0110] なお、多重化した第 1 機能回路 70 の間で、現在稼働中か否か（コンテキストを保持する App が実行中か否か）、また、どのようなコンテキストか（App の識別情報等）を共有するインターフェースを設けてもよい。また、処理部 46 に接続された CDMA エンジン 48 が、処理部 46 に代わって、複数の第 1 機能回路 70 を統括し、各第 1 機能回路 70 にどの App のコンテキストが保持されるかを管理してもよい。

[0111] また、図 15 A の状態と図 15 B の状態とを切り替える場合、全ての第 1 機能回路 70 のコンテキストスイッチを同時に実行する必要がある。このとき、第 1 実施例から第 4 実施例で説明したコンテキストスイッチ方法のいずれかを各 App について実行してもよい。そして、全ての App が揃ったタイミング（例えば全ての App の内部処理が完了したタイミング）で、コンテキストスイッチを実行してもよい。なお、図 15 A に示す状態で App A のみを別の App（例えば App F）に切り替える場合で、かつ、App F の性能要件が App A の性能要件以下であれば、App A に対応する第 1 機能回路 70 と第 2 機能回路 71 のみにおいてコンテキストスイッチを実行してもよい。

[0112] なお、第 1 機能回路 70 および第 2 機能回路 71 には、図 15 A、図 15 B に示した態様以外のアプリケーション割当の態様が可能であることはもちろんである。例えば、処理部 46 では、2 つまたは 3 つの第 2 機能回路 71 を占有するアプリケーションが、他のアプリケーションと同時に実行されてもよい。

[0113] 上記の第 2 実施例～第 5 実施例では、コンテキストを保持する第 1 機能回

路70を多重化した。この構成によると、アプリケーションの処理中にオーバーラップして、コンテキストの退避と設定が可能になる。これにより、CSDMAエンジン48およびデータを入出力する系（例えばバスファブリック100、スキャンチェーン106、専用経路108、パス114）の転送能力を簡易化できる。

[0114] 例えば、図3に示したビデオ描画処理において、1/240秒毎（＝期間4.2ミリ秒）に各アプリケーションが動作するとき、垂直帰線期間が0.5ミリ秒、各アプリケーションの処理時間が3.7ミリ秒であったとする。このとき、多重化を行わない構成では、コンテキストスイッチを0.5ミリ秒未満で実行する必要がある。一方、多重化を行った可能性では、3.7ミリ秒未満で実行すればよい。ここで、コンテキストのデータ量が50MBであったとき、垂直帰線期間内に転送するには、100GB/秒以上の転送能力が必要になる。一方、アプリケーション処理時間内の転送であれば、13.5GB/秒以上の転送能力があればよい。

[0115] 以上、本開示を第1実施例～第5実施例をもとに説明した。これらの実施例は例示であり、各構成要素あるいは各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本開示の範囲にあることは当業者に理解されるところである。

[0116] 上記実施例ではコンテキストの退避先をメインメモリ22としたが、コンテキストの退避先となるメモリ（言い換えればコンテキスト保存するメモリ）は、メインメモリ22とは異なるメモリであってもよい。また、コンテキストの退避先となるメモリのデータ保持特性は、揮発性であってもよく、不揮発性であってもよい。

[0117] コンテキストの退避先となるメモリが不揮発性である場合、サスペンド・ハイバネーションなどの、処理部46の一時停止・再開において当該メモリへの電源供給がカットされても、当該メモリに退避されたコンテキストは保持される。これにより、上記実施例に記載の高速なコンテキストスイッチを一層有効に利用できる。

- [0118] SOC 11において、未使用の処理部46に対する電源供給を個別にカットして低電力化を図る場合に、処理完了の区切りを待つ必要があったり、または、コンテキストの退避・設定の時間が長いと実用に耐えない可能性がある。上記実施例に記載の高速コンテキストスイッチを用いると、低電力化を図る場合にも間欠動作が可能となり、電力効率を高めることができる。
- [0119] また、上記実施例では、CPU12、処理部46（GPU14、ビデオエンコーダ16等）、CSDMAエンジン48は、1つのハードウェア（SOC11）上に実装された。変形例として、これらの機能ブロックは、複数のハードウェア上に分散して実装されてもよい。例えば、各実施例のCSDMAエンジン48は、処理部46が実装された装置から独立したデータ転送装置として実現されてもよい。
- [0120] 上述した実施例および変形例の任意の組み合わせもまた本開示の実施の形態として有用である。組み合わせによって生じる新たな実施の形態は、組み合わせられる実施例および変形例それぞれの効果をあわせもつ。また、請求項に記載の各構成要件が果たすべき機能は、実施例および変形例において示された各構成要素の単体もしくはそれらの連携によって実現されることも当業者には理解されるところである。

産業上の利用可能性

- [0121] 本開示の技術は、データを処理するシステムまたは装置に適用することができる。

符号の説明

- [0122] 10 コンピュータ、 11 SOC、 46 処理部、 48 CSDMAエンジン、 80 検出部、 86 転送実行部、 106 スキャンチェーン。

請求の範囲

- [請求項1] 複数のアプリケーションに関する処理を時分割で実行する処理部と、
、
前記処理部とメモリ間でデータを転送するデータ転送部と、
を備え、
前記データ転送部は、
前記処理部における実行対象のアプリケーションの切替タイミングを検出する検出部と、
前記切替タイミングが検出された場合、前記複数のアプリケーションを管理するソフトウェアの処理を介さずに、前記処理部で実行されていたアプリケーションのコンテキストを前記処理部から前記メモリへ退避させ、前記処理部で次に実行されるアプリケーションのコンテキストを前記メモリから前記処理部に設定する転送処理を実行する転送実行部と、を含むことを特徴とするデータ処理システム。
- [請求項2] 前記メモリに退避されるアプリケーションのコンテキストは、前記処理部において未完了の処理の現在状態を含み、
前記転送実行部は、前記切替タイミングが検出された場合、前記処理部におけるアプリケーションに関する処理が未完了であっても前記転送処理を実行することを特徴とする請求項1に記載のデータ処理システム。
- [請求項3] 前記転送実行部は、前記切替タイミングが検出された場合、前記処理部におけるアプリケーションに関する処理が完了後に前記転送処理を実行することを特徴とする請求項1に記載のデータ処理システム。
- [請求項4] 前記処理部におけるアプリケーションに関する処理は、当該アプリケーションにおける小さい粒度の複数の内部処理を含み、
前記転送実行部は、前記処理部における実行対象のアプリケーションの内部処理が完了後に前記転送処理を実行することを特徴とする請求項3に記載のデータ処理システム。

[請求項5] 前記処理部は、前記複数のアプリケーションに対応する複数のキューであって、各アプリケーションに関する処理内容を指示するコマンドが蓄積される複数のキューから、前記コマンドを読み出し、

前記処理部は、実行対象のアプリケーションを切り替える際に、コマンドを読み出すキューを切り替えることを特徴とする請求項1から4のいずれかに記載のデータ処理システム。

[請求項6] 前記検出部は、垂直帰線期間または水平帰線期間の開始を前記切替タイミングとして検出することを特徴とする請求項1から5のいずれかに記載のデータ処理システム。

[請求項7] 前記転送実行部は、DFT (Design For Test) 用のスキャンチェーンとメモリBIST (Built In Self-Test) 用の回路の少なくとも一方を用いて前記転送処理を実行することを特徴とする請求項1から6のいずれかに記載のデータ処理システム。

[請求項8] 前記処理部は、アプリケーションのコンテキストを保持する機能部を複数含み、

前記転送実行部は、前記処理部が第1の機能部に保持されたアプリケーションのコンテキストを用いて当該アプリケーションに関する処理を実行する間に、前記処理部の第2の機能部に保持された別のアプリケーションのコンテキストを対象として前記転送処理を実行することを特徴とする請求項1に記載のデータ処理システム。

[請求項9] 前記処理部は、アプリケーションのコンテキストを保持する機能部を複数含み、

前記処理部は、第1の機能部に保持された第1のアプリケーションのコンテキストを用いて前記第1のアプリケーションに関する処理を実行し、第2の機能部に保持された第2のアプリケーションのコンテキストを用いて前記第2のアプリケーションに関する処理を実行し、

前記処理部における実行対象が前記第1のアプリケーションから前記第2のアプリケーションに切り替わる場合、前記処理部は、前記第

1のアプリケーションに関する処理のうち前記第2のアプリケーションに関する処理と混流不可能な処理が完了後に、前記第1のアプリケーションに関する処理のうち前記第2のアプリケーションに関する処理と混流可能な処理と、前記第2のアプリケーションに関する処理とを並列実行することを特徴とする請求項1に記載のデータ処理システム。

[請求項10] 前記転送実行部による転送処理の対象は、前記処理部のステート・マシンに保持されたコンテキストを含むことを特徴とする請求項1から9のいずれかに記載のデータ処理システム。

[請求項11] 前記転送実行部による転送処理の対象は、前記処理部に保持されるコンテキストであって、ソフトウェアからアクセスできないコンテキストを含むことを特徴とする請求項1から10のいずれかに記載のデータ処理システム。

[請求項12] 前記コンテキストは、中断された時点のアプリケーションの処理の再現に必要なデータであることを特徴とする請求項1から11のいずれかに記載のデータ処理システム。

[請求項13] 複数のアプリケーションに関する処理を時分割で実行する処理部における実行対象のアプリケーションの切替タイミングを検出する検出部と、

前記切替タイミングが検出された場合、前記複数のアプリケーションを管理するソフトウェアの処理を介さずに、前記処理部で実行されていたアプリケーションのコンテキストを前記処理部からメモリへ退避させ、前記処理部で次に実行されるアプリケーションのコンテキストを前記メモリから前記処理部に設定する転送処理を実行する転送実行部と、

を備えることを特徴とするデータ転送装置。

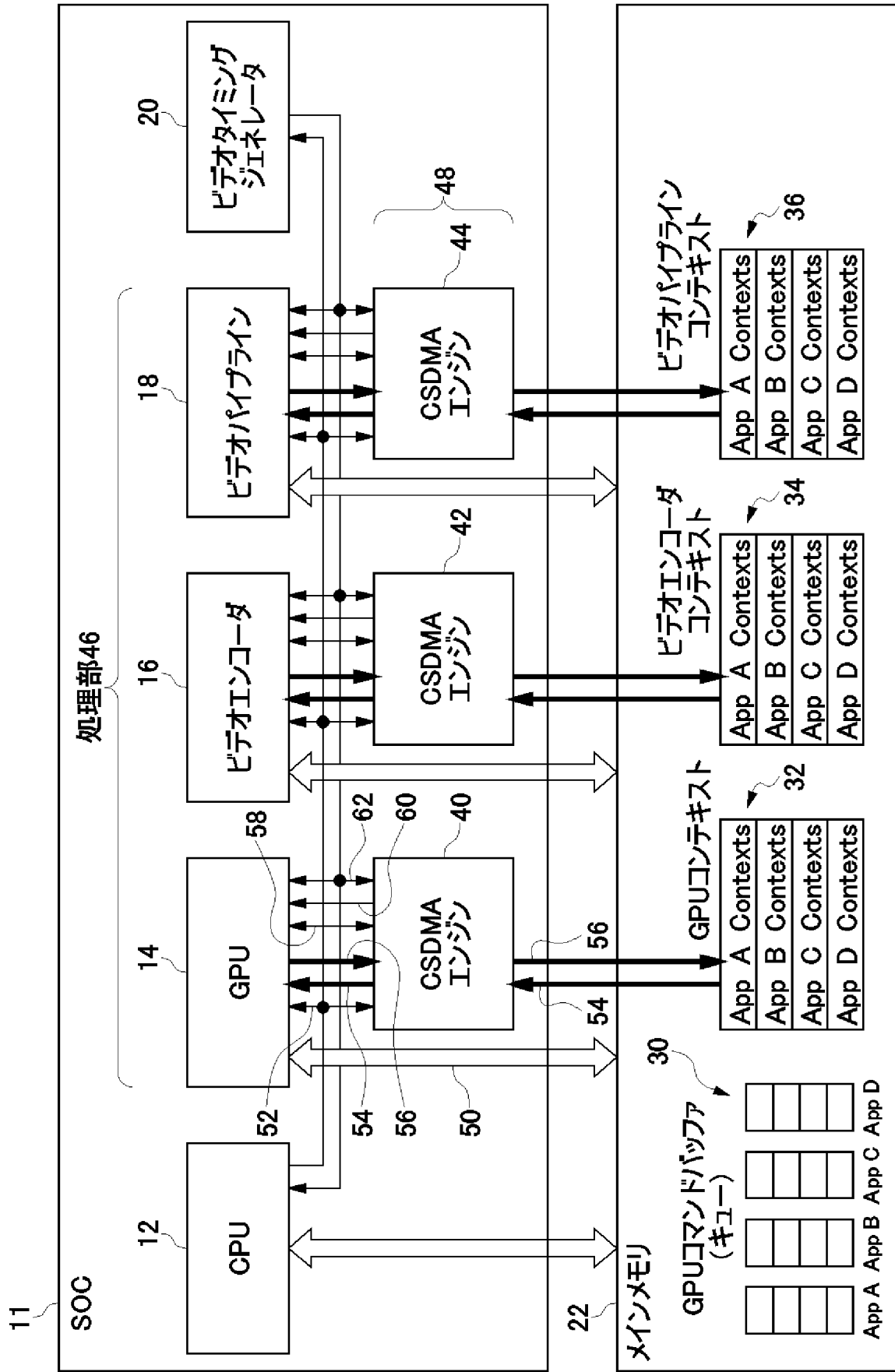
[請求項14] 複数のアプリケーションに関する処理を時分割で実行する処理部における実行対象のアプリケーションの切替タイミングを検出するステ

ップと、

前記切替タイミングが検出された場合、前記複数のアプリケーションを管理するソフトウェアの処理を介さずに、前記処理部で実行されていたアプリケーションのコンテキストを前記処理部からメモリへ退避させ、前記処理部で次に実行されるアプリケーションのコンテキストを前記メモリから前記処理部に設定する転送処理を実行するステップと、

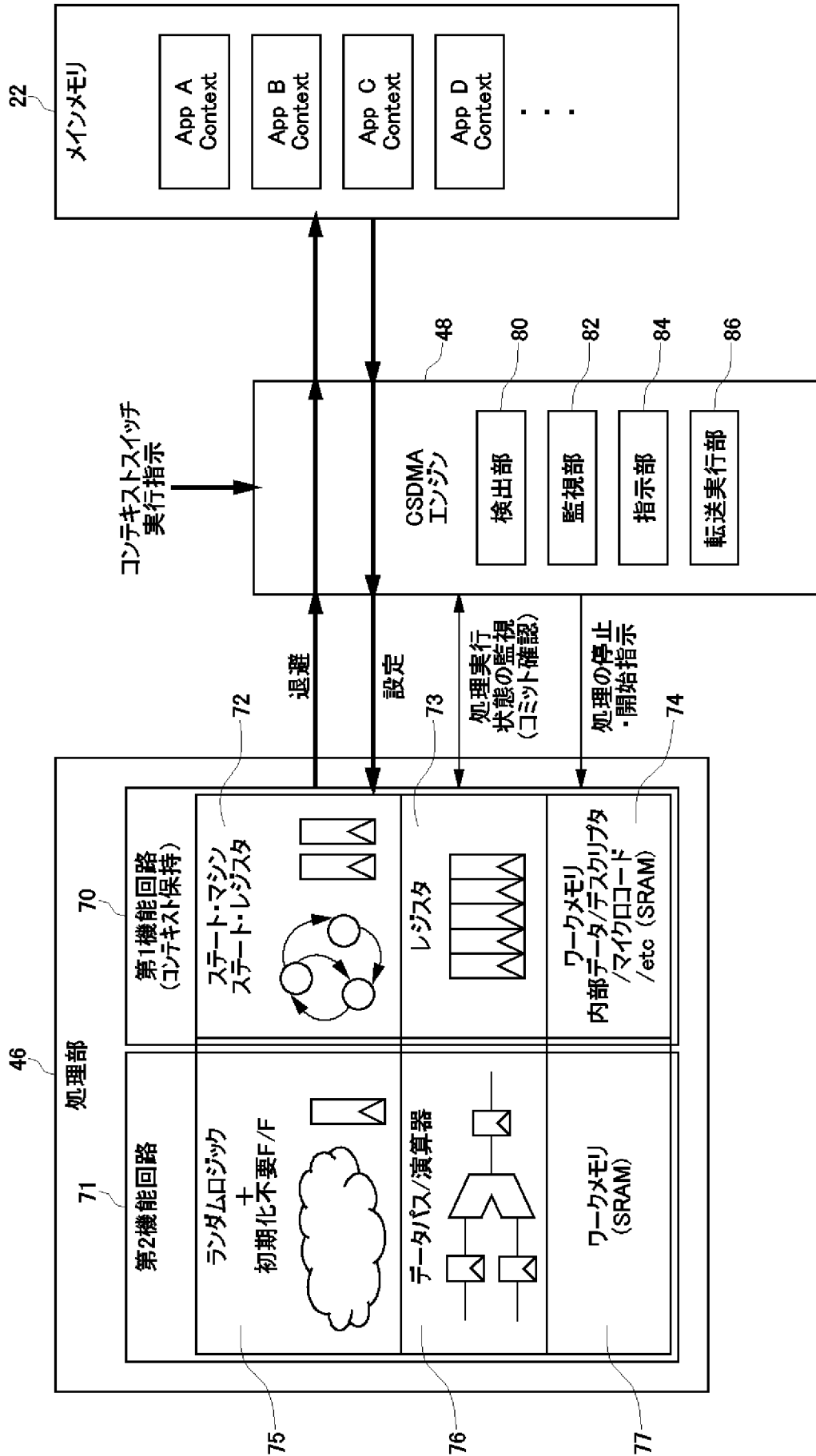
をコンピュータが実行することを特徴とするコンテキストスイッチ方法。

[図1]

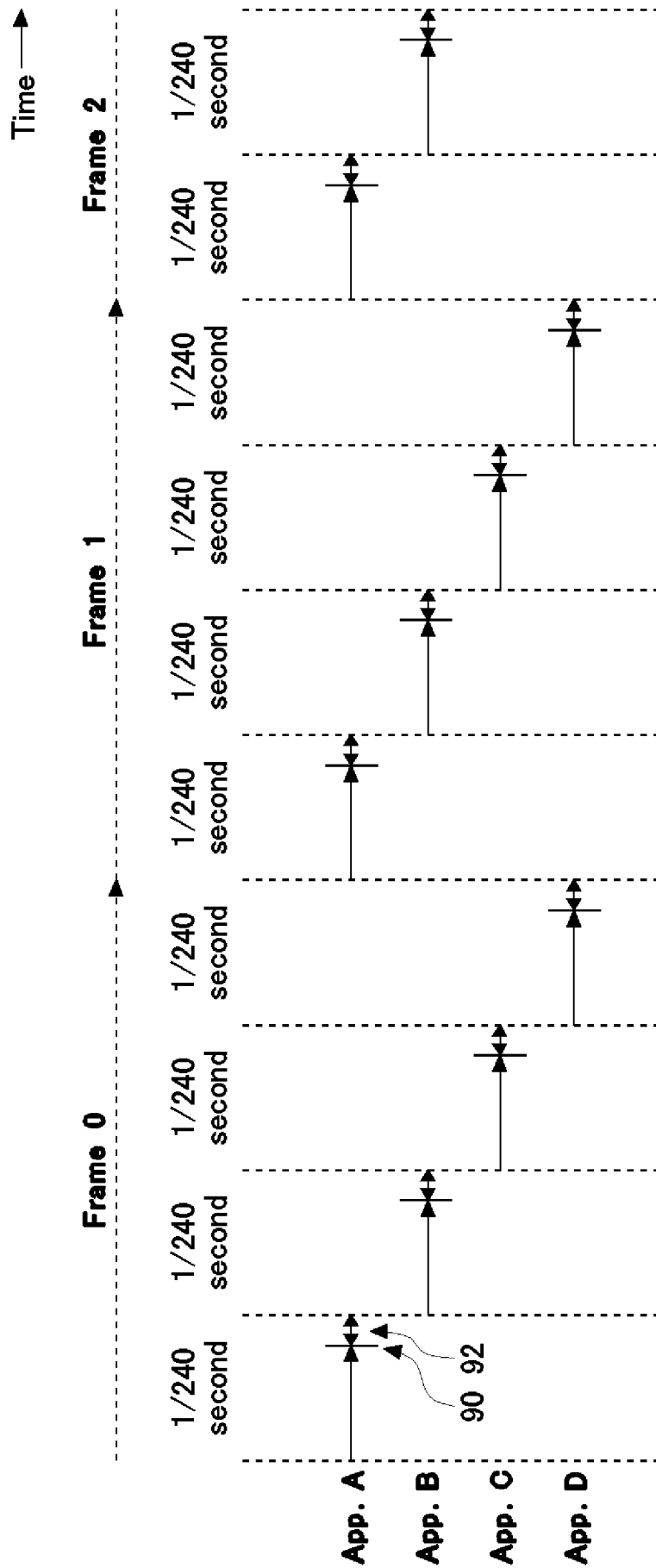


コンピュータ 10

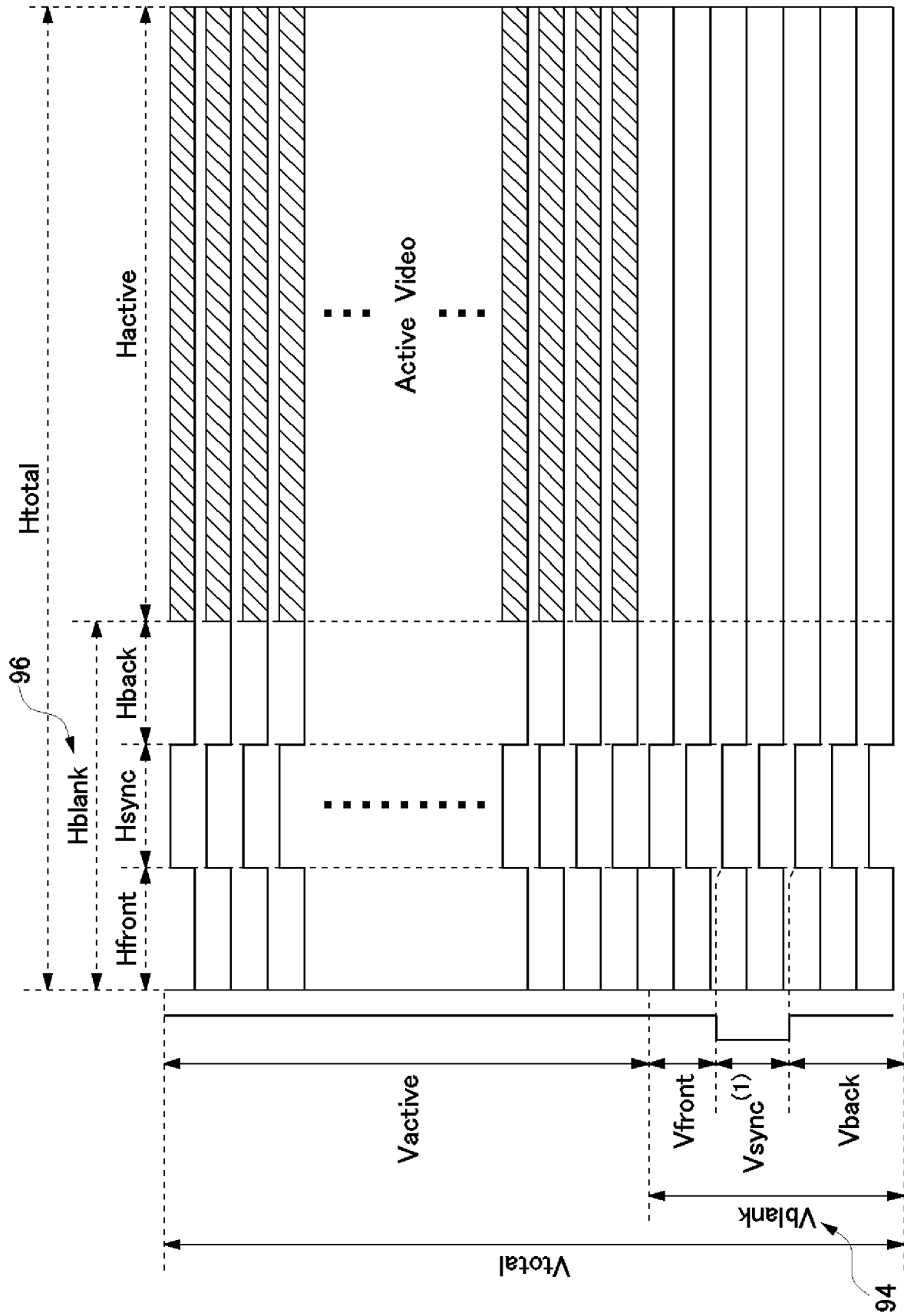
[図2]



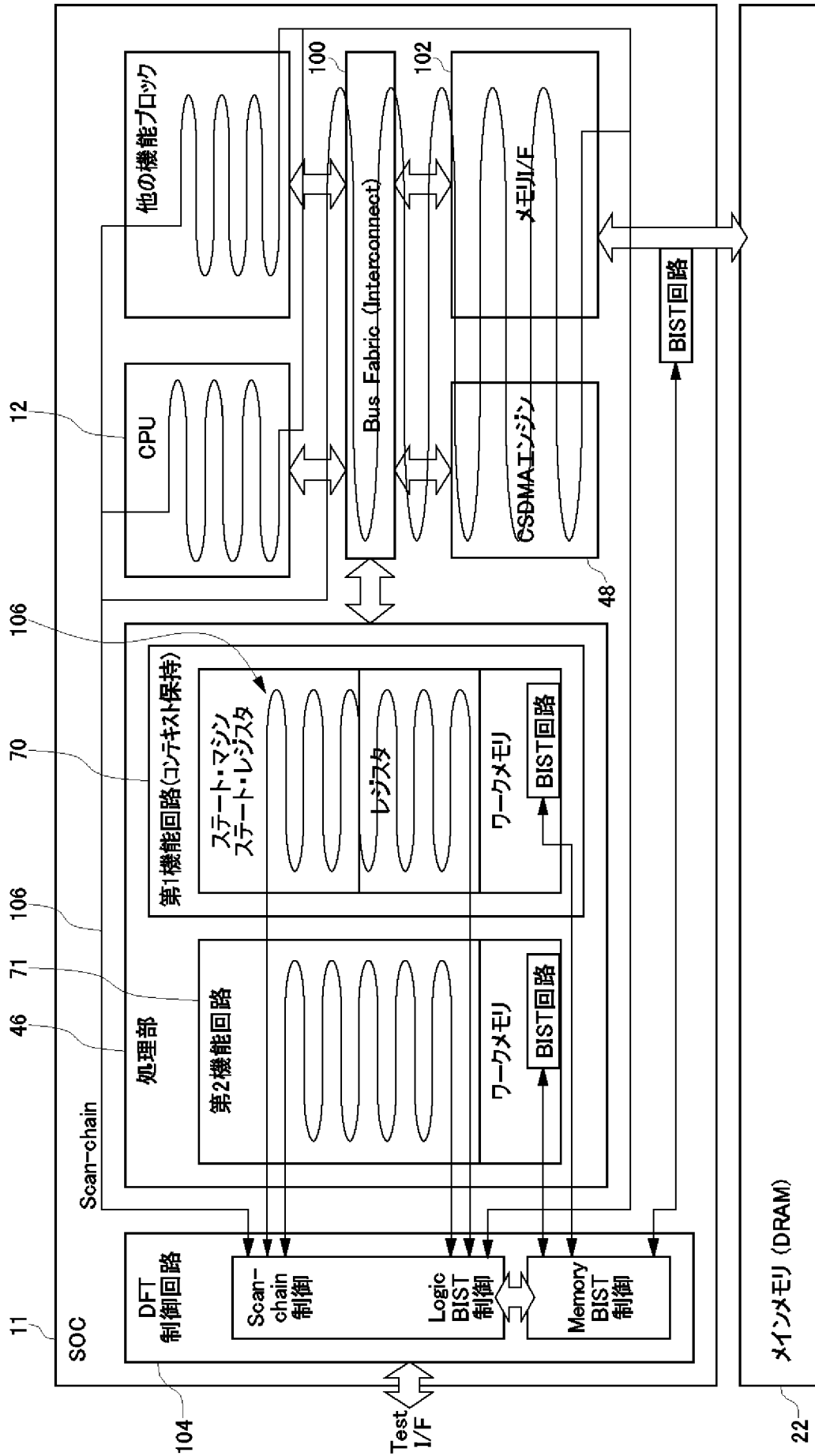
[3]



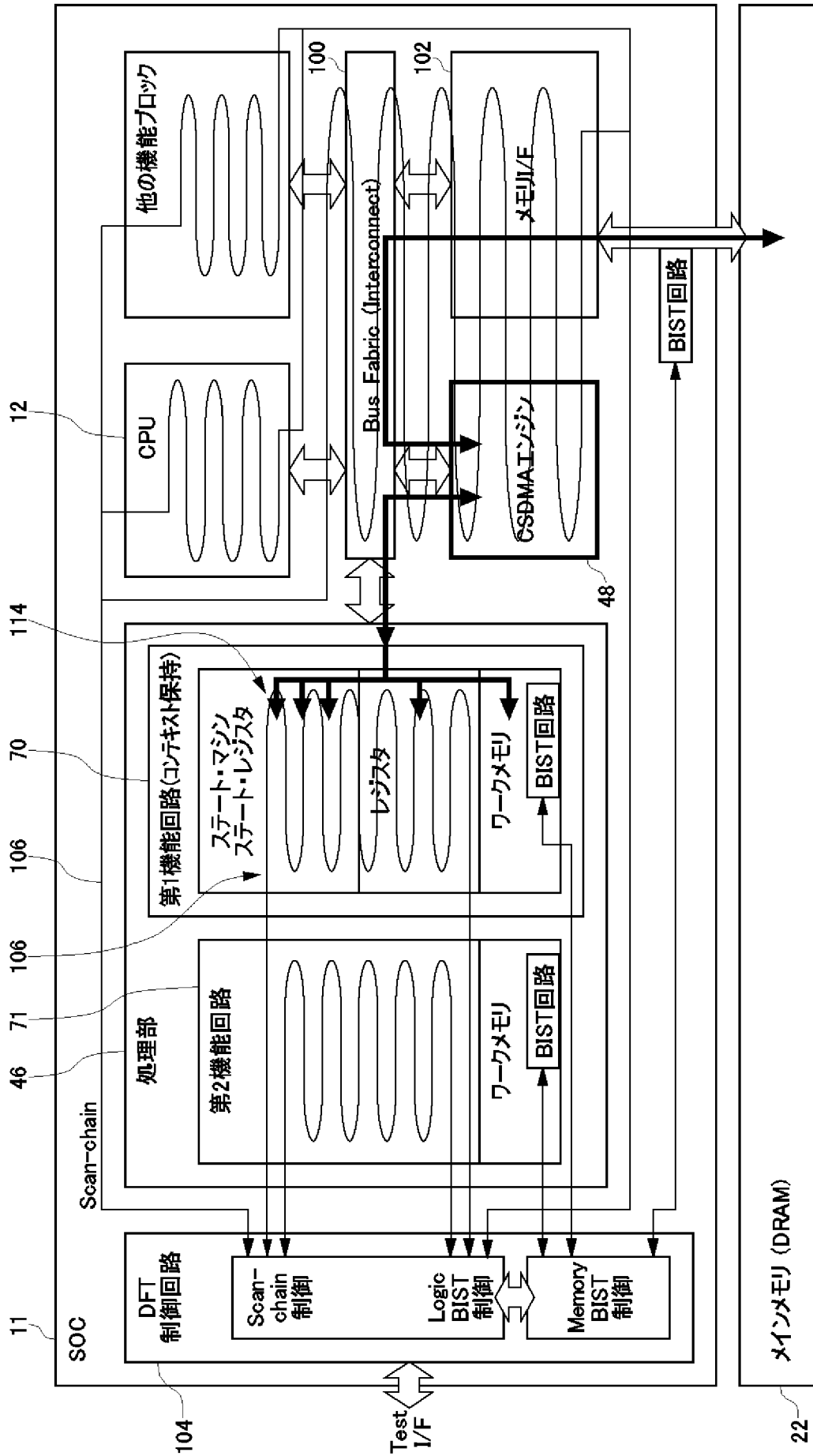
[図4]



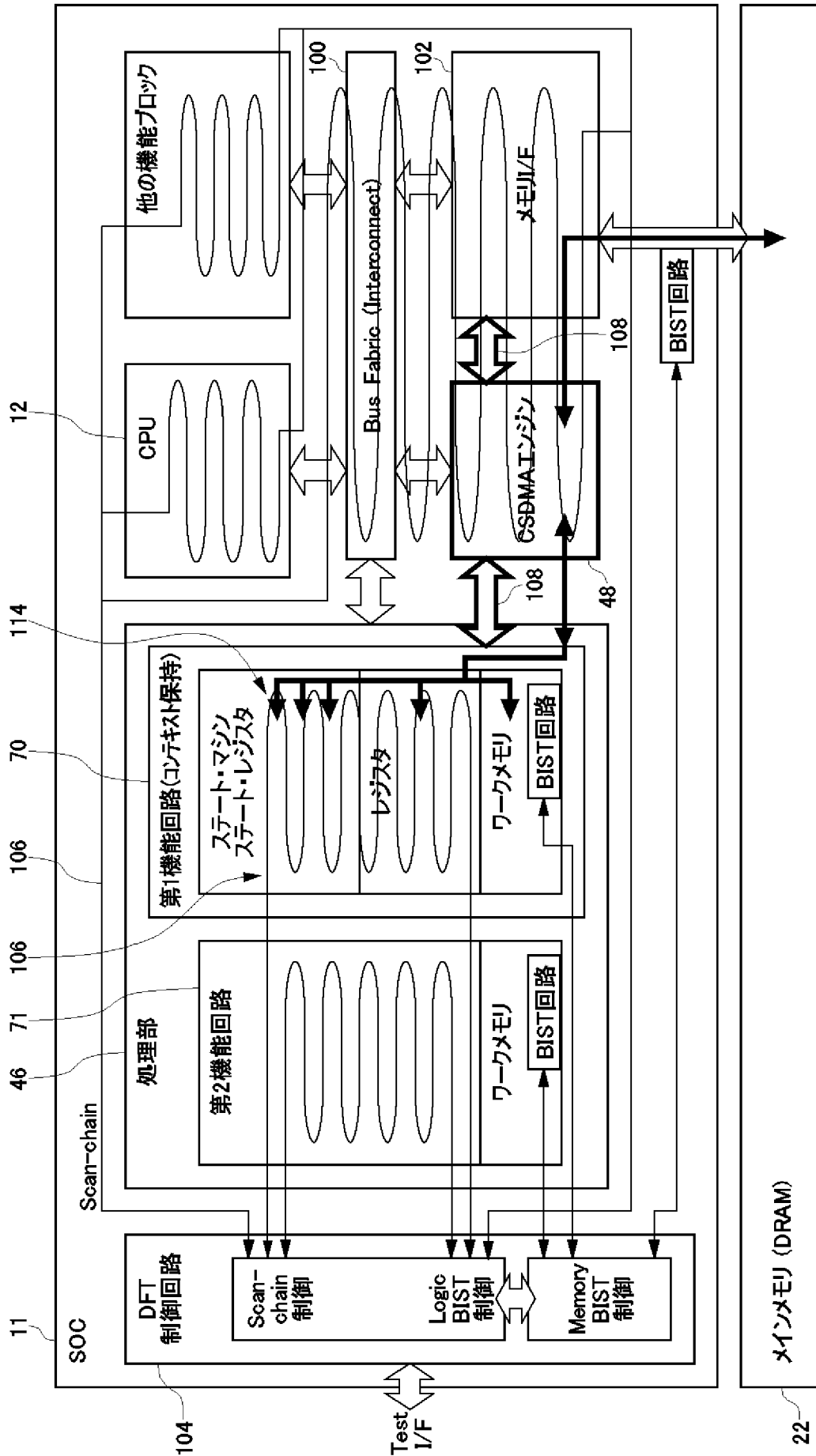
[図5]



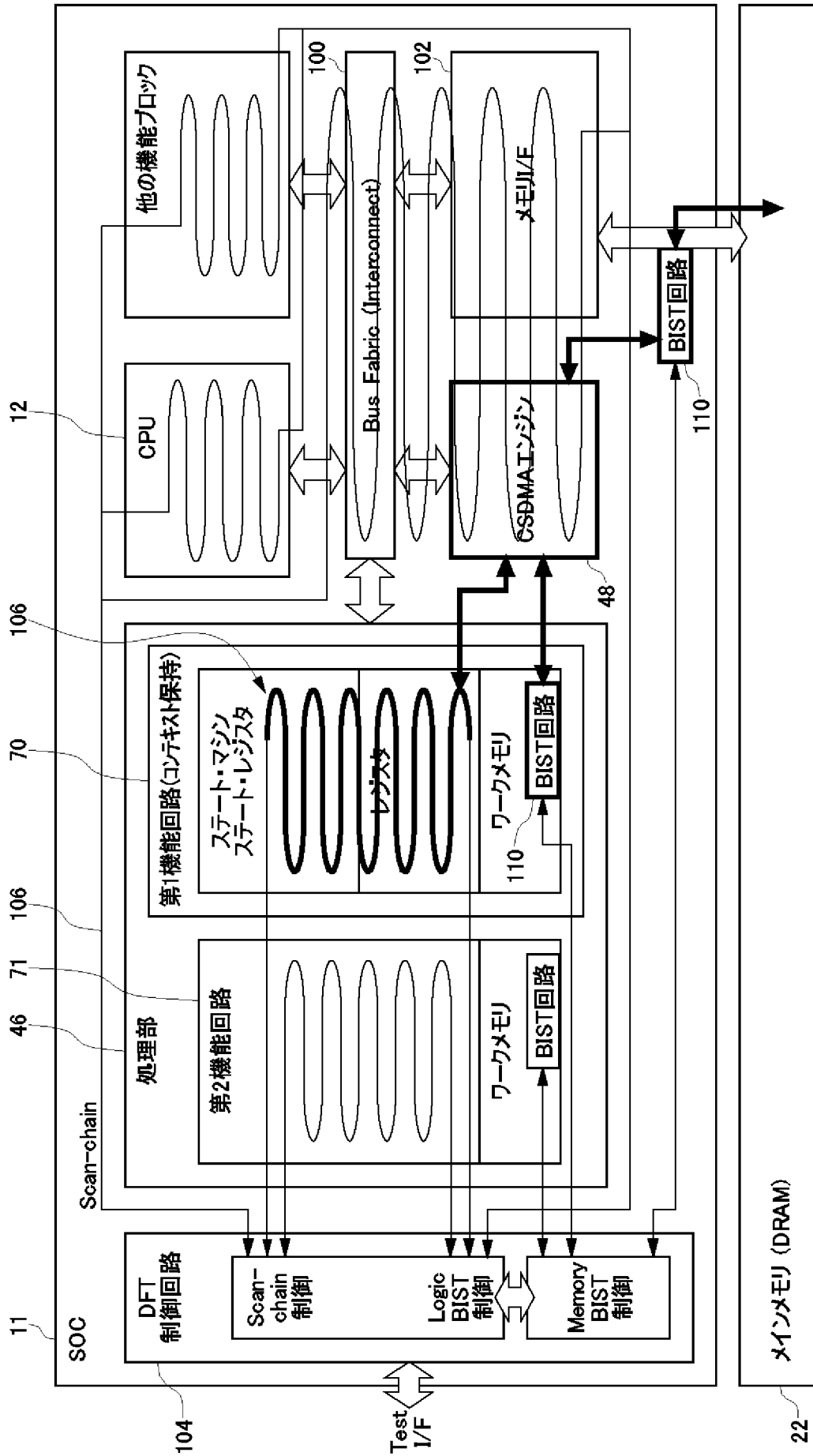
[図6]



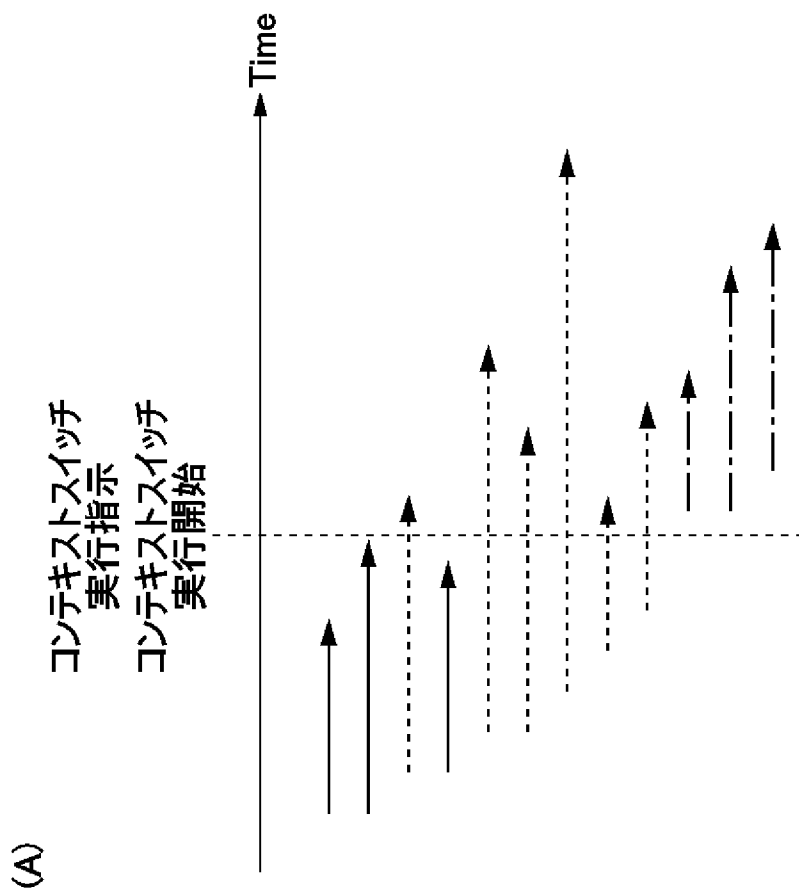
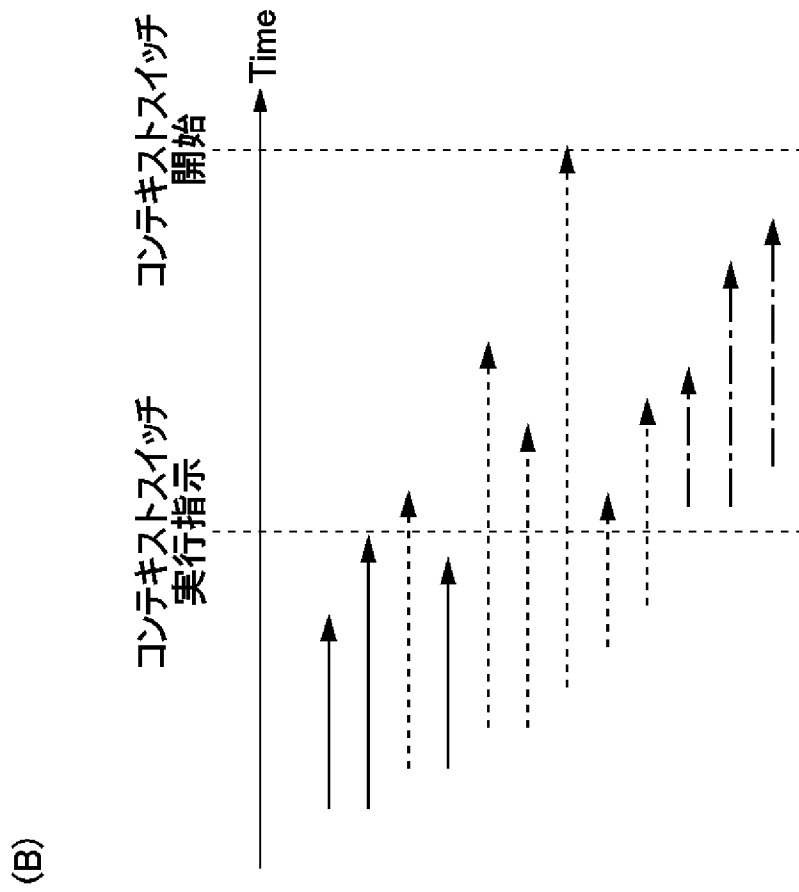
[図7]



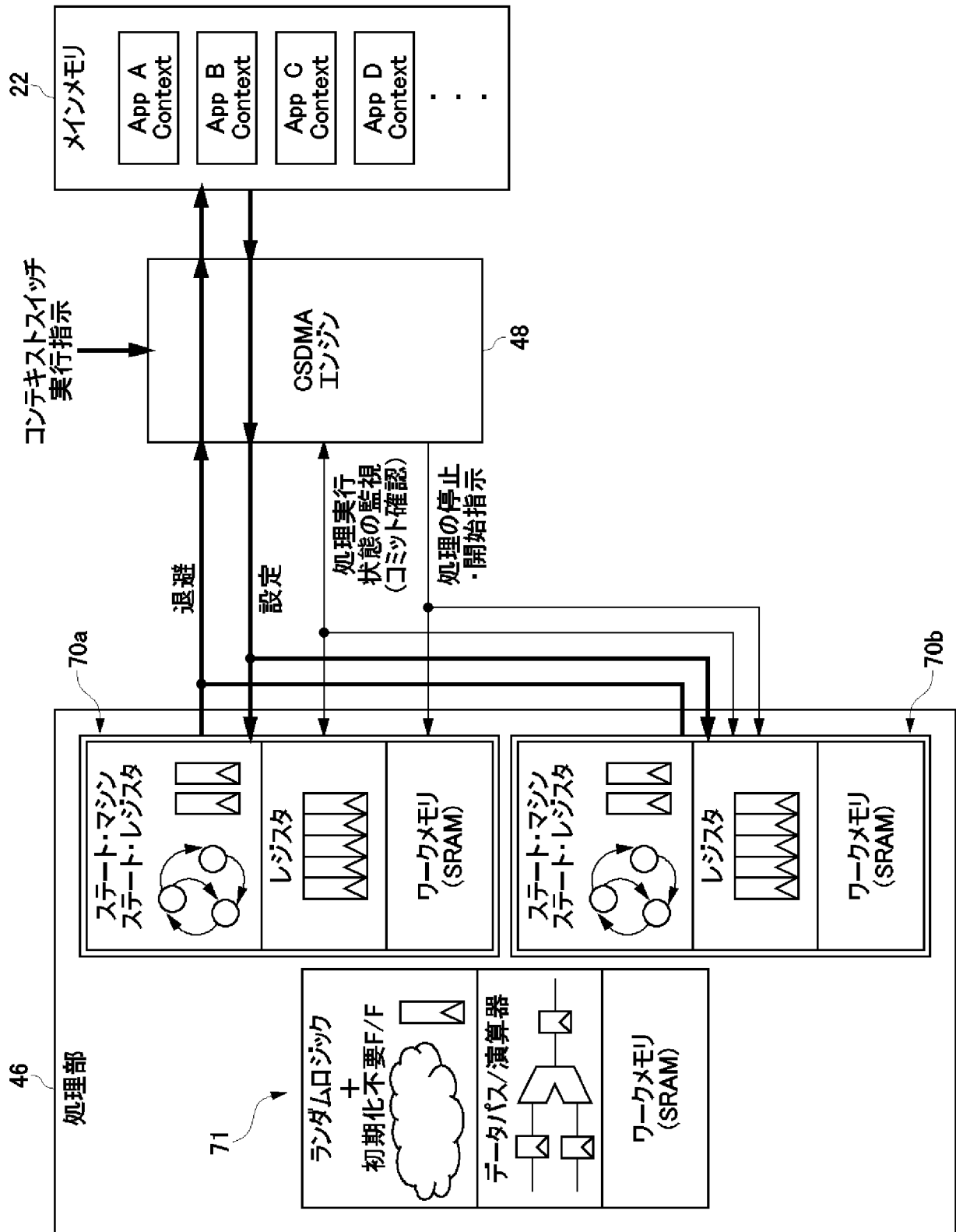
[図8]



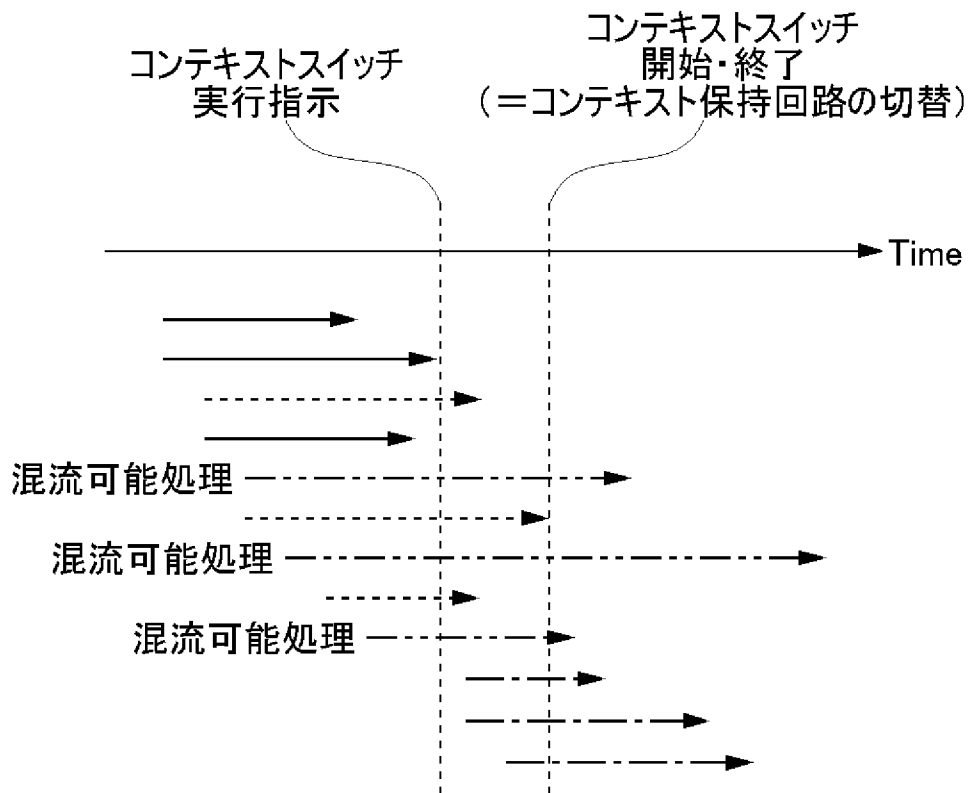
[図9]



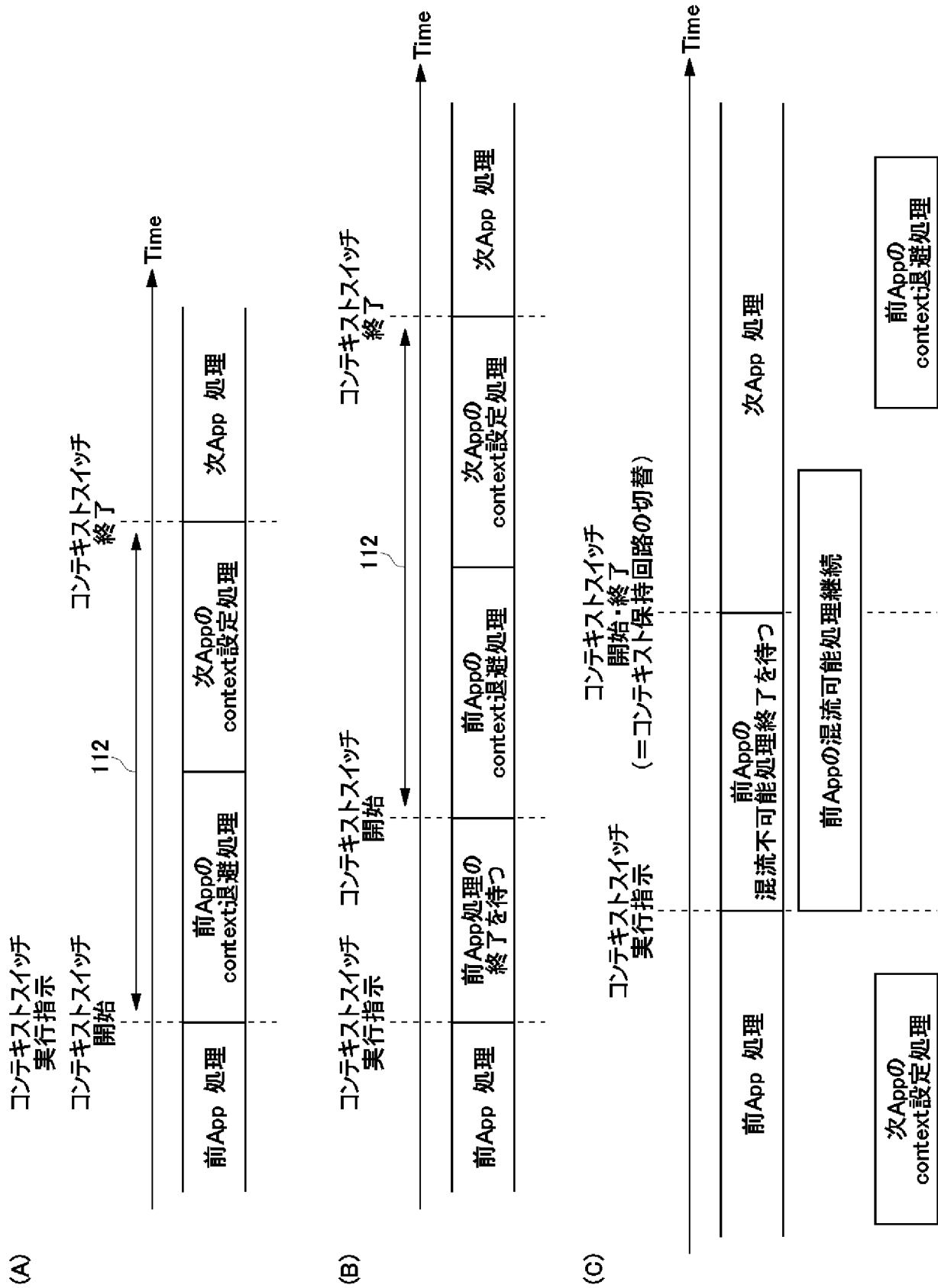
[図10]



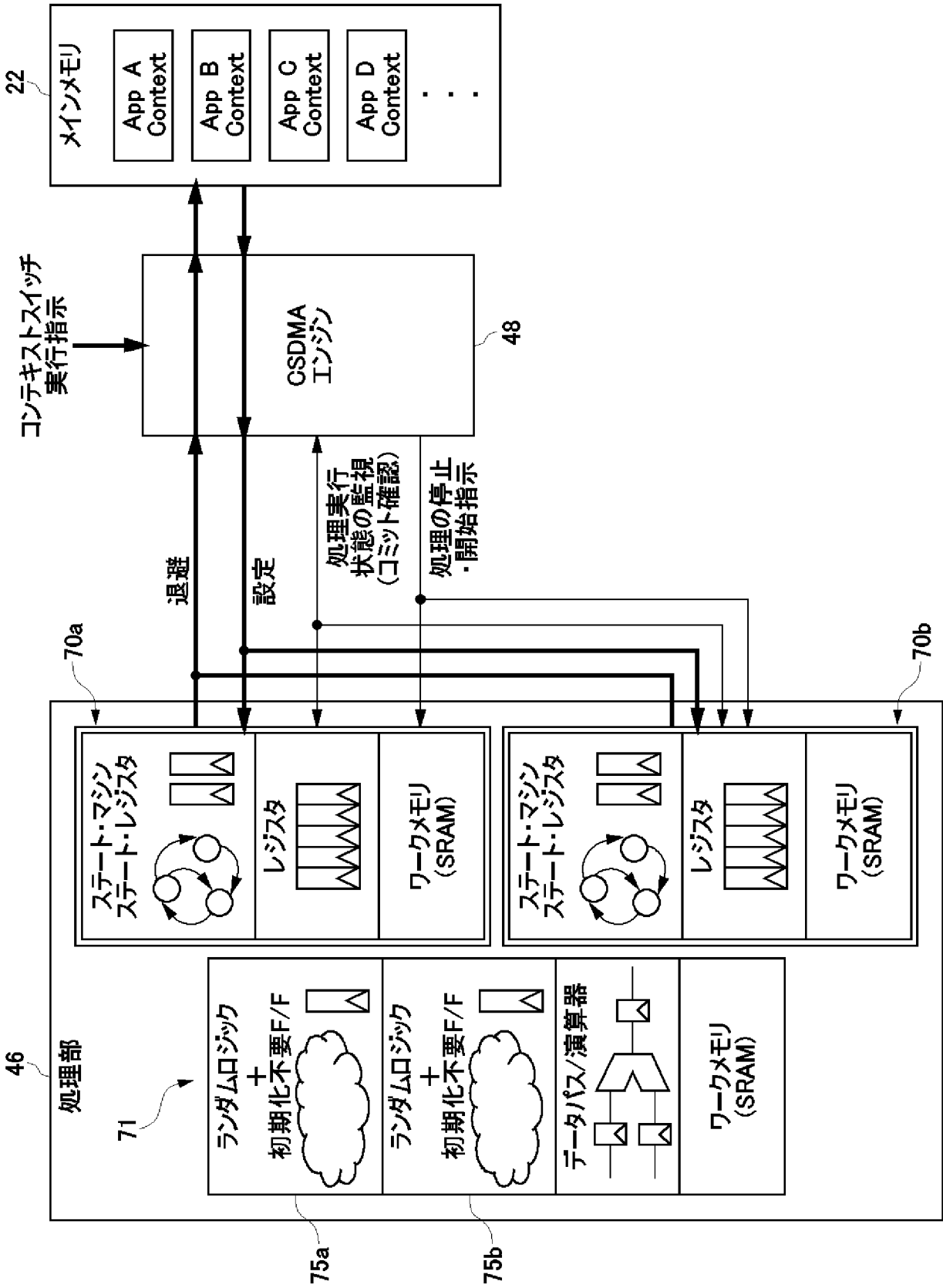
[図11]



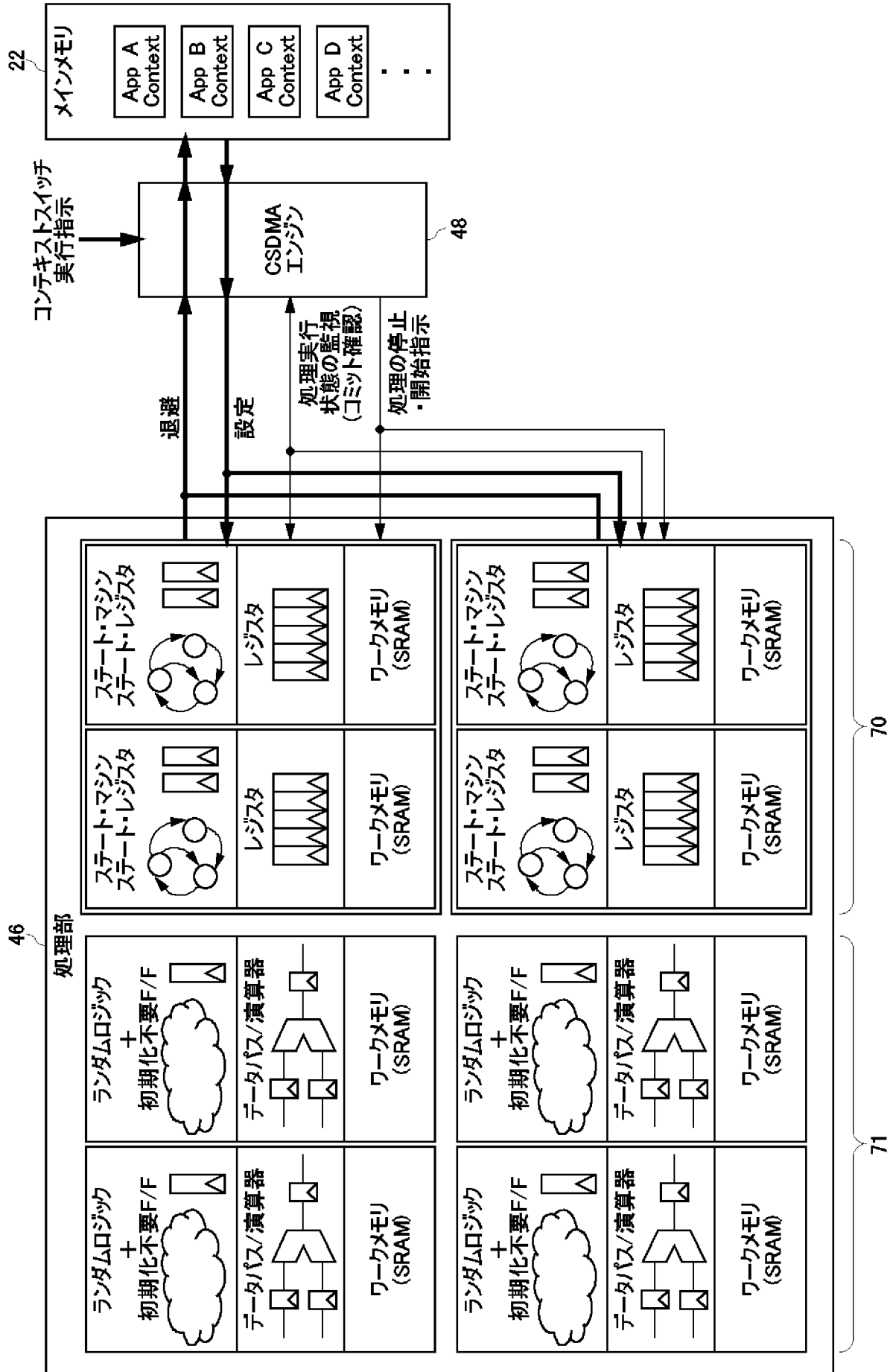
[図12]



[図13]

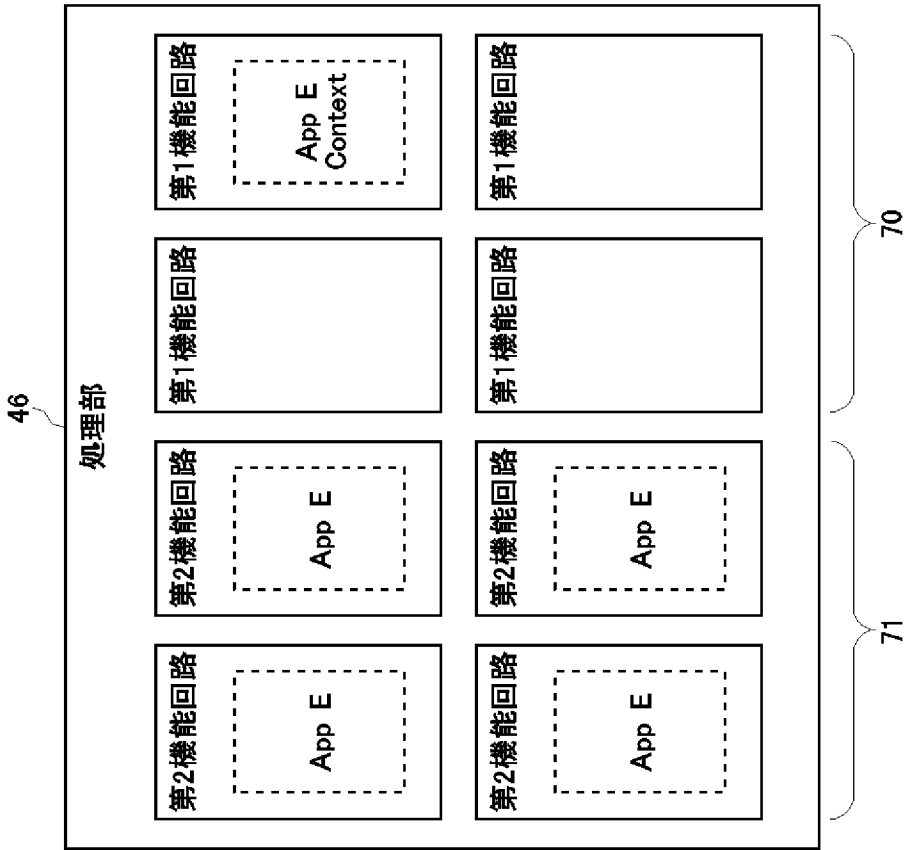


[図14]

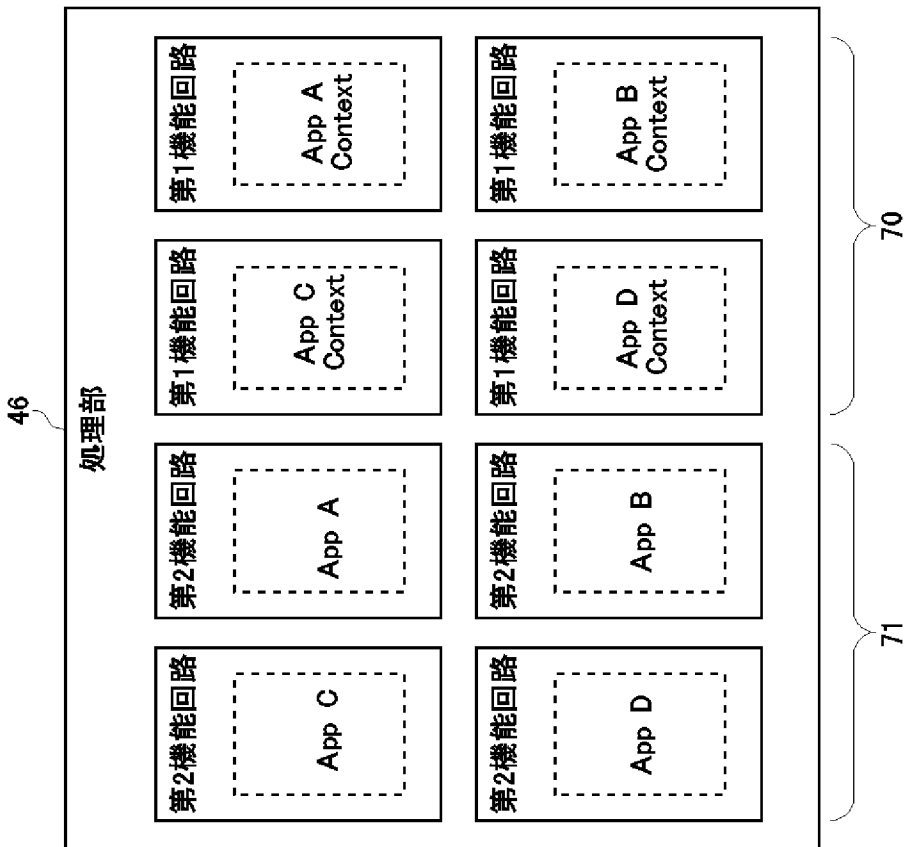


[図15]

(B)



(A)



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2021/002880

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/48 (2006.01) i; G06F 15/177 (2006.01) i; G06F 15/80 (2006.01) i
 FI: G06F9/48 300D; G06F15/177 B; G06F15/80; G06F9/48 350Z

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)

G06F9/48; G06F15/177; G06F15/80

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	1971-2021
Registered utility model specifications of Japan	1996-2021
Published registered utility model applications of Japan	1994-2021

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.
X	JP 2003-271399 A (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.) 26 September 2003 (2003-09-26) paragraphs [0016]-[0042], fig. 1	1-2, 12-14
Y	paragraphs [0016]-[0042], fig. 1	5-7, 10-11
A	paragraphs [0016]-[0042], fig. 1	3-4, 8-9
Y	JP 6-83614 A (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.) 25 March 1994 (1994-03-25) abstract	5
Y	JP 6-180653 A (HUDSON CO., LTD.) 28 June 1994 (1994-06-28) paragraphs [0008], [0037]	6
Y	JP 2004-185602 A (INTERNATIONAL BUSINESS MACHINES CORPORATION) 02 July 2004 (2004-07-02) paragraphs [0064]-[0069]	7
Y	JP 2006-351013 A (SEIKO EPSON CORP.) 28 December 2006 (2006-12-28) claims	10-11

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
 19 April 2021 (19.04.2021)

Date of mailing of the international search report
 27 April 2021 (27.04.2021)

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

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/JP2021/002880

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
JP 2003-271399 A	26 Sep. 2003	US 2003/0149864 A1 paragraphs [0232]- [0285], fig. 1 (Family: none)	
JP 6-83614 A	25 Mar. 1994		
JP 6-180653 A	28 Jun. 1994	US 5530873 A1 column 1, lines 49- 54, column 4, lines 56-61	
JP 2004-185602 A	02 Jul. 2004	US 2009/0247142 A1 paragraphs [0085]- [0091] CN 1504833 A	
JP 2006-351013 A	28 Dec. 2006	US 2006/0288299 A1 claims	

A. 発明の属する分野の分類（国際特許分類（IPC）） G06F 9/48(2006.01)i; G06F 15/177(2006.01)i; G06F 15/80(2006.01)i FI: G06F9/48 300D; G06F15/177 B; G06F15/80; G06F9/48 350Z		
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）） G06F9/48; G06F15/177; G06F15/80 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 1971-2021年 日本国実用新案登録公報 1996-2021年 日本国登録実用新案公報 1994-2021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
X	JP 2003-271399 A (松下電器産業株式会社) 26.09.2003 (2003-09-26) 【0016】～【0042】，【図1】	1-2, 12-14
Y	【0016】～【0042】，【図1】	5-7, 10-11
A	【0016】～【0042】，【図1】	3-4, 8-9
Y	JP 6-83614 A (松下電器産業株式会社) 25.03.1994 (1994-03-25) 【要約】	5
Y	JP 6-180653 A (株式会社ハドソン) 28.06.1994 (1994-06-28) 【0008】，【0037】	6
Y	JP 2004-185602 A (インターナショナル・ビジネス・マシーンス・コーポレーシ ン) 02.07.2004 (2004-07-02) 【0064】～【0069】	7
Y	JP 2006-351013 A (セイコーエプソン株式会社) 28.12.2006 (2006-12-28) 【特許請求の範囲】	10-11
<input type="checkbox"/> C欄の続きにも文献が列挙されている。 <input checked="" type="checkbox"/> 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、一般的な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抵 触するものではなく、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 19.04.2021	国際調査報告の発送日 27.04.2021	
名称及びあて先 日本国特許庁 (ISA/JP) 〒100-8915 日本国 東京都千代田区霞が関三丁目4番3号	権限のある職員（特許庁審査官） 田中 幸雄 5B 9191 電話番号 03-3581-1101 内線 3545	

国際調査報告
 パテントファミリーに関する情報

国際出願番号

PCT/JP2021/002880

引用文献	公表日	パテントファミリー文献	公表日
JP 2003-271399 A	26.09.2003	US 2003/0149864 A1 [0232]-[0285], FIG.1	
JP 6-83614 A	25.03.1994	(ファミリーなし)	
JP 6-180653 A	28.06.1994	US 5530873 A column1, lines49-54, column4, lines56-61	
JP 2004-185602 A	02.07.2004	US 2009/0247142 A1 [0085]-[0091] CN 1504833 A	
JP 2006-351013 A	28.12.2006	US 2006/0288299 A1 claims	