(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
2008 年9 月25 日 (25.09.2008)

PCT

(10) 国際公開番号
WO 2008/114367 A1

(51) 国際特許分類:
G06F 9/50 (2006.01) *H04N 7/26* (2006.01)

(21) 国際出願番号: PCT/JP2007/055439

(22) 国際出願日: 2007 年3 月16 日 (16.03.2007)

(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): 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; および

(75) 発明者/出願人 (米国についてのみ): 今井 賢 (IMAI, Satoshi) [JP/JP]; 〒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). 都市 雅彦 (TOICHI, Masahiko) [JP/JP]; 〒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: 伊東 忠彦 (ITOH, Tadahiko); 〒1506032 東京都渋谷区恵比寿４丁目２０番３号 恵比寿ガーデンプレイスタワー３２階 Tokyo (JP).

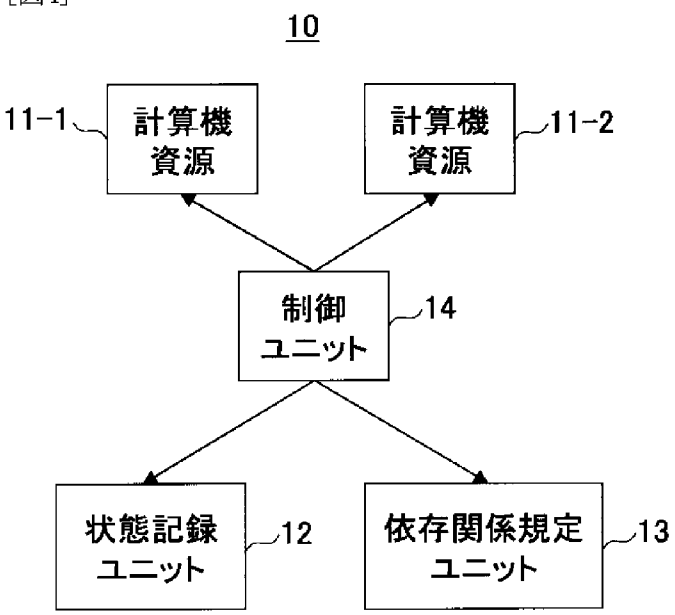
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM,

[続葉有]

(54) **Title:** COMPUTER SYSTEM AND CODING/DECODING METHOD

(54) **発明の名称:** 計算機システム及び符号化／復号化方法

[図4]



11-1, 11-2 COMPUTER RESOURCE
14 CONTROL UNIT
12 STATE RECORDING UNIT
13 DEPENDENT RELATION DEFINING UNIT

(57) **Abstract:** A computer system, which carries out coding/decoding comprised of a plurality of processes for dynamic image data, includes a plurality of computer resources; a state recording unit for recording any one of states of impermissible execution, permissible execution, ongoing execution, and execution completion; and a dependent relation defining unit for showing dependent relations among a plurality of the processes, wherein the state recording unit renews a state from the ongoing execution of one process to its execution completion in response to the completion of one process for a certain image block and, with reference to the dependent relation defining unit, specifies one or a plurality of processes that become the permissible execution by the execution completion of one process, and the state recording unit allocates at least one process of the permissible execution to one of a plurality of the computer resources so as to make it execute, and renews at least one process from the state of the permissible execution to that of the ongoing execution.

(57) 要約: 複数の処理から構成される符号化／復号化を動画像データに対して実行する計算機システムは、複数の計算機資源と、画像ブロックの各々に対して複数の処理の各処理毎に実行不可、実行可、実行中、実行完了の何れかの状態を記録した状態記録ユニットと、複数の処理の間の依存関係を示す依存関係規定ユニットを含み、ある画像ブロックに対する１つの処理の完了に応答して状態記録ユニットにおいて１つの処理を実行中の状態から実行完了の状態に更新するとともに依存関係

[続葉有]

WO 2008/114367 A1

PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, 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, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

規定ユニットを参照して１つの処理の実行完了により実行可となる１つ又は複数の処理を特定して状態記録ユニットにおいて特定された１つ又は複数の処理を実行不可の状態から実行可の状態に更新し、状態記録ユニットにおいて実行可の少なくとも１つの処理を複数の計算機資源のうちの少なくとも１つに割り当てて実行させるとともに実行可の少なくとも１つの処理を実行可の状態から実行中の状態に更新する。

明 細 書

計算機システム及び符号化／復号化方法

技術分野

[0001] 本発明は、一般に計算機システム及び符号化／復号化方法に関し、詳しくは動画像を符号化及び復号化する計算機システム及び符号化／復号化方法に関する。

背景技術

[0002] 動画像の圧縮符号化アルゴリズムとして、従来からMPEG－1、MPEG－2、H．263、MPEG－4等が用いられており、DCT（離散コサイン変換）等の周波数変換、動きベクトルを利用したフレーム間予測、量子化、エントロピー符号化等の各処理によりデータを圧縮符号化している。またこれら圧縮符号化アルゴリズムよりも更に圧縮率及び画質を向上できるアルゴリズムとして、H．264規格が注目されている。

[0003] 図1は、H．264の復号時の各処理の流れを示す図である。まずヘッダを解析し、その結果に基づいて例えばCABAC（Context-based Adaptive Binary Arithmetic Coding：適応算術演算符号化）によりデータを復号する。この符号化方式は、処理対象のデータに応じてシンボル確率分布を変化させるとともにシンボル間の相関を利用することにより、データ内容に応じた最適なモードで算術演算符号化するものである。このCABACによる復号化の後、逆整数変換（IT）及び逆量子化（IQ）により、動き補償後の画像を再構成する。ここで整数変換／逆整数変換は、従来のように浮動小数点演算により周波数変換／逆変換していたDCT／IDCTの代わりに、整数演算のみにより周波数変換／逆変換を行うアルゴリズムであり、丸め誤差を除去することができるという利点がある。

[0004] また動きベクトル復号化（MV）により画像中の動き成分の情報を求め、この情報に基づいて動き補償（MC）のデータを生成する。この動き補償のデータと逆整数変換（IT）及び逆量子化（IQ）により求めた動き補償後の画像とに基づいて、動き補償前の元の画像を再構成する。また画像内復号化（INTRA）により、同一フレーム内の類似する他のマクロブロックとの差分として圧縮されたデータを伸長する。最後に、デブロッキングフィルタ処理を実行する。これは、ブロック間での変換／量子化プロセスや動

きベクトルの違い等によってブロック境界に発生するアーティファクトを除去することを目的とするものであり、ブロックの境界に対して行われるフィルタ処理である。

[0005] 上記は復号化時の処理についての説明であるが、符号化時においても、上記の各処理に対応した複数の処理が行われる。このように、H. 264等の動画像の圧縮符号化アルゴリズムの符号化／復号化処理においては、周波数変換、動きベクトルによる予測、量子化、エントロピー符号化等に対応した複数の処理が実行される。こうした複数の処理からなる符号化／復号化処理をマルチプロセッサシステムやASIC(Application Specific Integrated Circuit)により実行する場合には、各処理を各計算資源に静的に割り当てる。

[0006] 図2は、H. 264の復号化の各処理を2つのプロセッサへ静的に割り当てる場合の割り当ての一例を示す図である。図2において、プロセッサPE0には、ヘッダ解析、CABAC、IT&IQ、及びMVの各処理が割り当てられ、プロセッサPE1には、INTRA、MC、画像結合＋、及びデブロッキングフィルタの各処理が割り当てられている。このような各処理の各計算資源への割り当ては、データ処理量やデータ依存関係等を考慮して決定されるが、データ処理量は処理対象の動画データに依存して変化するので、プロセッサ間での待ち時間が発生することは避けられない。

[0007] 図3は、図2のような処理割り当てにおいて待ち時間が発生する様子を説明するための図である。図3において、まずプロセッサPE0が、ある画像ブロックに対してCABAC、IT、IQ、及びMVを実行する。その間プロセッサPE1は、その画像ブロックに対して引き続き実行されるべき処理であるMC(図3では＋(MCA)と示してある)及びデブロッキングフィルタ(DF)を実行することができずに、プロセッサPE0の処理終了を待っている。

[0008] プロセッサPE0によるCABAC、IT、IQ、及びMV処理が終了すると、リモートプロセス設定フラグRMPset__flgを設定することにより、プロセッサPE1に＋(MCA)及びDFの処理を開始させる。その後、プロセッサPE0は、次の画像ブロックに対してCABAC、IT、IQ、及びMVを実行して処理を終了する。しかしこの時点では、プロセッサPE1の＋(MCA)及びDFの処理が終了していないので、プロセッサPE0にはPE1の処理終了を待つ時間(PE1ウェイト)が発生してしまう。

[0009] 前述のように、符号化／復号化の各処理のデータ処理量は、処理対象の動画データに依存して変化してしまうので、図2のように計算機資源に各処理を静的に割り当てた場合、計算機資源間で待ち時間が発生することは避けられない。即ち、データ処理量やデータ依存関係等をいくら慎重に考慮して処理割り当てを決定しても、図3に示すような待ち時間（PE1ウェイト）を無くすことはできず、計算機資源を有効に活用することができない。

特許文献1:特開平10－145237

発明の開示

発明が解決しようとする課題

[0010] 以上を鑑みて本発明は、動画像の符号化／復号化において、計算機資源を有効に活用できるように複数の処理を複数の計算機資源に割り当てる計算機システムを提供することを目的とする。

課題を解決するための手段

[0011] 本発明は、複数の処理から構成される符号化／復号化を動画像データに対して実行する計算機システムであって、複数の計算機資源と、該動画像データの2つ又はそれ以上の画像ブロックの各々に対して、該複数の処理毎に実行不可、実行可、実行中、実行完了の何れかの状態を記録した状態記録ユニットと、該複数の処理の間の依存関係を示す依存関係規定ユニットを含み、該画像ブロックに対する第1処理の完了に応答して、該状態記録ユニットにおける該第1処理の状態を、実行中の状態から実行完了の状態に更新するとともに、該依存関係規定ユニットを参照して、該第1処理の実行完了に応じて実行可となる第2処理を特定し、該状態記録ユニットにおける該特定された第2処理の状態を、実行不可の状態から実行可の状態に更新し、該第2処理を該複数の計算機資源のうちの少なくとも1つに割り当てて実行させるとともに、該第2処理を実行可の状態から実行中の状態に更新するように該計算機資源及び該状態記録ユニットの制御動作を実行することを特徴する。

[0012] また、複数の処理から構成される符号化／復号化を複数の計算機資源により実行する符号化／復号化方法であって、動画像データの2つ又はそれ以上の画像ブロックの各々に対して該複数の処理の処理毎に実行不可、実行可、実行中、実行完了

の何れかの状態をメモリに記録し、該画像ブロックに対する第1処理の完了に応答して、該メモリにおける該第1処理の状態を、実行中の状態から実行完了の状態に更新し、該複数の処理の間の依存関係を参照して、該第1処理の実行完了に応じて実行可となる第2処理を特定して、該メモリにおいて該特定された該第2処理の状態を、実行不可の状態から実行可の状態に更新し、該メモリにおいて実行可として記録された該第2処理を、実行可の状態から実行中の状態に更新するとともに、該第2処理を該複数の計算機資源のうちの少なくとも1つに割り当てて実行させることを特徴とする。

発明の効果

[0013] 本発明の少なくとも1つの実施例によれば、依存関係規定ユニットの依存関係情報に基づいて実行可能な処理を特定し、特定された実行可能な処理を随時計算機資源に動的に割り当てていくことにより、実行可能な処理があるにも関わらず利用可能な計算機資源がアイドル状態となってしまうことがない。従来のように静的に処理を計算機資源に割り当てた場合、実行可能な処理があっても、その処理が予め(即ち静的に)割り当てられている計算機資源が利用可能でなければ、その実行可能な処理を実行することができない。それに対して本願発明では、実行可能な処理がある場合に、その処理を実行可能な計算機資源は1つに固定されていないので、利用可能な計算機資源に随時その処理を割り当てて実行していくことができる。従って、静的な処理割り当ての場合と比較して、処理対象の動画データ内容に依存してデータ処理量が変動しても、無駄な待ち時間を成るべく避けて効率的な処理を実行することができる。

図面の簡単な説明

[0014] [図1]H.264の復号時の各処理の流れを示す図である。

[図2]H.264の復号化の各処理を2つのプロセッサへ静的に割り当てる場合の割り当ての一例を示す図である。

[図3]図2のような処理割り当てにおいて待ち時間が発生する様子を説明するための図である。

[図4]本発明の原理を説明するための図である。

[図5]本発明の原理を説明するための図である。

[図6]符号化／復号化を構成する処理の依存関係を示す図である。

[図7]状態記録ユニットの記録データ及びその更新状況を示す図である。

[図8]本発明による計算機システムの第1の実施例の構成を示す図である。

[図9]本発明による計算機システムの第2の実施例の構成を示す図である。

[図10]本発明による動的処理割り当て処理の一例を示すフローチャートである。

[図11]図10に示すテーブル更新及びテーブルチェック動作の一例を示すフローチャートである。

[図12]配列Pr[kind][m]を示す図である。

[図13]本発明による動的処理割り当て処理の別の一例を示すフローチャートである。

[図14]図13に示すテーブル更新及びテーブルチェック動作の一例を示すフローチャートである。

[図15]H．264の復号化処理における本発明による処理割り当ての一例を模式的に示す図である。

[図16]H．264の復号化処理における従来技術による処理割り当ての一例を模式的に示す図である。

[図17]H．264の復号化処理における各処理間の依存関係を説明するための図である。

[図18]H．264の復号化処理において1つのマクロブロックに対するCABAC処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。

[図19]H．264の復号化処理において1つのマクロブロックに対するIT&IQ処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。

[図20]H．264の復号化処理において1つのマクロブロックに対するMV処理及びMC処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。

[図21]H．264の復号化処理において1つのマクロブロックに対するMV処理及びM

C処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。

[図22]H. 264の復号化処理において1つのマクロブロックに対するINTRA処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。

[図23]H. 264の復号化処理において1つのマクロブロックに対する加算処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。

[図24]H. 264の復号化処理において1つのマクロブロックに対するデブロッキングフィルタ処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。

符号の説明

[0015] 10 計算機システム

11－1、11－2 計算機資源

12 状態記録ユニット

13 依存関係規定ユニット

14 制御ユニット

発明を実施するための最良の形態

[0016] 以下に、本発明の実施例を添付の図面を用いて詳細に説明する。

[0017] 図4は、本発明の原理を説明するための図である。複数の処理から構成される符号化／復号化を動画像データに対して実行する本発明による計算機システム10は、複数の計算機資源11－1及び11－2と、動画像データの2つ又はそれ以上の画像ブロックの各々に対して複数の処理の各処理毎に実行不可、実行可、実行中、実行完了の何れかの状態を記録した状態記録ユニット12と、複数の処理の間の依存関係を示す依存関係規定ユニット13を含む。図4に示す計算機システム10は、更に制御ユニット14を含む。制御ユニット14は、状態記録ユニット12及び依存関係規定ユニット13を参照して、符号化／復号化処理を構成する複数の処理の各々を計算機資源11－1及び11－2に割り当てて実行させる。この処理割り当て及び処理実行が、状態記録ユニット12及び依存関係規定ユニット13の情報に基づいて動的に行われる。

[0018] 図5に計算機システム10Aとして示すように、計算機資源11－1及び11－2を集中制御する制御ユニット14は設けなくともよい。図5の計算機システム10Aの場合、計算機資源11－1及び11－2のそれぞれが状態記録ユニット12及び依存関係規定ユニット13を参照して、自律的に処理を選択して実行する。この処理選択及び処理実行が、状態記録ユニット12及び依存関係規定ユニット13の情報に基づいて動的に行われる。即ち、複数の計算機資源11－1及び11－2の各々が依存関係規定ユニット13を参照し状態記録ユニット12の内容を更新し処理を実行することにより、複数の計算機資源11－1及び11－2の協調動作として制御動作を実行する。なおこの場合、複数の計算機資源11－1及び11－2による状態記録ユニット12のアクセス及び更新が排他的となるように、状態記録ユニット12に対するロック及びアンロック（ロック解除）の機能を設けてよい。

[0019] なお図4及び図5において、一例として2つの計算機資源11－1及び11－2が示されているが、計算機資源の数は2に限定されるものではない。本発明における計算機資源の数は、2又はそれ以上の任意の数であってよい。

[0020] 図6は、符号化／復号化を構成する処理である処理1乃至5の依存関係を示す図である。図6において、処理1乃至5は、矢印20で示す順番で実行される。即ち、ある画像ブロックに対して処理1乃至5が順番に実行され、その後次の画像ブロックに対して処理1乃至5が順番に実行される。また処理間の依存関係（データ依存、順序依存、制御依存等）として矢印21で示すように、処理2の前に処理1が実行完了し、処理3の前に処理2が実行完了し、処理4の前に処理3が実行完了し、処理5の前に処理4が実行完了している必要がある。これは同一の画像ブロックに対する処理1乃至5についての依存関係である。また画像ブロック間の処理の依存関係として矢印22で示すように、ある処理n（n:1～5）に着目した場合、次の画像ブロックの処理nの前に現在の画像ブロックの処理nが実行完了している必要がある。これらの処理依存関係が依存関係規定ユニット13に規定されている。依存関係規定ユニット13は例えばメモリであり、上記依存関係を表現したデータを格納するよう構成されてよい。

[0021] 図7は、状態記録ユニット12の記録データ及びその更新状況を示す図である。状態記録ユニット12は例えばメモリであり、動画像データの2つ又はそれ以上の画像ブ

ロックの各々に対して各処理毎に実行不可、実行可、実行中、実行完了の何れかの状態を示すデータを格納するよう構成されてよい。

[0022] 図7の状態記録ユニット12はテーブル形式となっており、テーブルの第1行に示す数字1乃至5は図6に示す処理1乃至5を表している。またテーブルの第2行には動画像データ中のある画像ブロック(ここでは第1ブロックと呼ぶ)に対して実行する処理を示し、第3行には第1ブロックの次の画像ブロック(ここでは第2ブロックと呼ぶ)に対して実行する処理を示す。テーブル中のデータエントリは、"0"が実行不可、"1"が実行可、"2"が実行中、"3"が実行完了の状態を示す。

[0023] 図7の状態記録ユニット12の初期状態(左側のテーブル)では、第1ブロックについては、処理1の状態が実行可"1"であり、他の処理2乃至5は実行不可"0"の状態である。また第2ブロックについては、処理1乃至5の全てが実行不可"0"の状態である。この初期状態から、実行可能な第1ブロックの処理1を例えば計算機資源11－1に割り当てて実行させるとともに、状態記録ユニット12において、実行可能な第1ブロックの処理1を実行可の状態"1"から実行中の状態"2"に更新する。

[0024] その後、計算機資源11－1が第1ブロックの処理1を完了すると、その完了に応答して、状態記録ユニット12において第1ブロックの処理1を実行中の状態"2"から実行完了の状態"3"に更新する。この様子が図7の中央のテーブルに示される。

[0025] 更に、依存関係規定ユニット13を参照して第1ブロックの処理1の実行完了により実行可となる1つ又は複数の処理を特定して、状態記録ユニット12において、その特定された1つ又は複数の処理を実行不可の状態"0"から実行可の状態"1"に更新する。この様子が図7の右側のテーブルに示される。この場合、図6に示されるように処理1が完了すると実行可となる処理は、矢印21の依存関係で示される同一画像ブロックの処理2と、矢印22の依存関係で示される次の画像ブロックの処理1である。従って、図7に示す右側のテーブルにおいては、第1ブロックの処理2と第2ブロックの処理1とが実行可の状態"1"に更新される。

[0026] 第1ブロックの処理2と第2ブロックの処理1とが実行可の状態"1"になったので、第1ブロックの処理2を例えば計算機資源11－1に割り当てて実行させ、第2ブロックの処理1を例えば計算機資源11－2に割り当てて実行させることができる。このようにし

て計算機資源11－1及び11－2に処理を割り当てて実行させていくことで、動的な処理の割り当てが実現される。

[0027] このように本発明による計算機システムにおいては、ある画像ブロックに対する1つの処理の完了に応答して状態記録ユニット12においてその1つの処理を実行中の状態から実行完了の状態に更新するとともに依存関係規定ユニット13を参照してその1つの処理の実行完了により実行可となる1つ又は複数の処理を特定して状態記録ユニット12においてその特定された1つ又は複数の処理を実行不可の状態から実行可の状態に更新する。また更に、状態記録ユニット12において実行可の少なくとも1つの処理を複数の計算機資源のうちの少なくとも1つに割り当てて実行させるとともに、その実行可の少なくとも1つの処理を実行可の状態から実行中の状態に更新する。このように計算機資源11－1及び11－2及び状態記録ユニット12を制御することにより、計算機資源に対する動的な処理の割り当てが可能となる。

[0028] 上記のように依存関係規定ユニットの依存関係情報に基づいて実行可能な処理を特定し、特定された実行可能な処理を随時計算機資源に動的に割り当てていくことにより、実行可能な処理があるにも関わらず利用可能な計算機資源がアイドル状態となってしまうことがない。従来のように静的に処理を計算機資源に割り当てた場合、実行可能な処理があっても、その処理が予め(即ち静的に)割り当てられている計算機資源が利用可能でなければ、その実行可能な処理を実行することができない。それに対して本願発明では、実行可能な処理がある場合に、その処理を実行可能な計算機資源は1つに固定されていないので、利用可能な計算機資源に随時その処理を割り当てて実行していくことができる。従って、静的な処理割り当ての場合と比較して、処理対象の動画データ内容に依存してデータ処理量が変動しても、無駄な待ち時間を成るべく避けて効率的な処理を実行することができる。

[0029] 図8は、本発明による計算機システムの第1の実施例の構成を示す図である。図8に示す計算機システムはマルチプロセッサシステムであり、マルチプロセッサチップ30、メインメモリ41、及びROMやASIC等の周辺デバイス42を含む。マルチプロセッサチップ30は、CPUコア31－1乃至31－n、SRAM(Static Random Access Memory)32－1乃至32－n、メモリコントローラ33、DMAC(Direct Memory Access Contro

ller)34、高速クロスバー35、DMAC36、ローカルバスIF(Interface)37、バスブリッジ38、周辺バス39、及びクロック・リセット・パワーコントロール40を含む。マルチプロセッサチップ30は、メモリコントローラ33によりメインメモリ41へのデータ読み書きを行う。またマルチプロセッサチップ30は、DMAC36及びローカルバスIF37を介して周辺デバイス42とデータのやり取りをする。

[0030] 図8において、CPUコア31－1乃至31－n(及びSRAM32－1乃至32－n)が本発明の複数の計算機資源に対応する。図4に対応する構成の場合には、CPUコア31－1乃至31－nのうちの何れか1つが制御ユニット14として機能する構成であってよい。即ち、制御ユニット14として機能するCPUコアがメモリ上に設けられた状態記録ユニット12及び依存関係規定ユニット13をアクセスしながら、他のCPUコアに処理を動的に割り当て実行させるように構成すればよい。また図5に対応する構成の場合には、全てのCPUコア31－1乃至31－nが複数の計算機資源として動作して、メモリ上に設けられた状態記録ユニット12及び依存関係規定ユニット13をアクセスしながら、処理を動的に選択実行する構成としてよい。

[0031] 図9は、本発明による計算機システムの第2の実施例の構成を示す図である。図9に示す計算機システムはASIC利用のシステムであり、ASIC50とSDRAM(Synchronous Dynamic Random Access Memory)60とを含む。ASIC50は、各々がCABACを実行する複数のCABACユニット51－1及び51－2、各々がMVを実行する複数のMVユニット52－1及び52－2、IT／IQを実行するIT／IQユニット53、INTRA(フレーム内符号化／復号化)を実行するINTRAユニット54、各々がINTER(フレーム間符号化／復号化)を実行する複数のINTERユニット55－1乃至55－3、各々がデブロッキングフィルタ処理を実行する複数のDFユニット56－1乃至56－3、参照画像を一時的に格納する参照画像キャッシュ57、DMA(Direct Memory Access)ユニット58、及びバス59を含む。

[0032] 図8において、CABACユニット51－1及び51－2、MVユニット52－1及び52－2、IT／IQユニット53、INTRAユニット54、INTERユニット55－1乃至55－3、及びDFユニット56－1乃至56－3の各処理ユニットが、本発明の複数の計算機資源に対応する。またASIC50の上記各処理ユニットを制御して各処理を実行させるプログ

ラム及び制御機構が、図4の制御ユニット14として機能する構成であってよい。即ちこの制御機構が、メモリ上に設けられた状態記録ユニット12及び依存関係規定ユニット13をアクセスしながら、ASIC50の各処理ユニットに処理を動的に割り当て実行させるように構成すればよい。なおASICの場合、各処理ユニットの処理内容は固定であるが、例えばCABACについては複数のCABACユニット51－1及び51－2が設けられている。従って、あるCABAC処理が実行可能となったときに、利用可能なCABACユニットに処理を随時割り当てて実行させればよい。

[0033] 図10は、本発明による動的処理割り当て処理の一例を示すフローチャートである。図10に示す例では、復号化処理を第1の処理（CABAC, IT, IQ, MV）と第2の処理（＋（MCA）, DF）との2つの処理に分割し、これら2つの処理を2つのプロセッサユニット（計算機資源）PE0及びPE1に割り当てて実行している。

[0034] ステップS1で、ある画像ブロックに対して第1の処理（CABAC, IT, IQ, MV）をプロセッサユニットPE0が実行する。この処理の実行が完了すると、ステップS2乃至S4において、プロセッサユニットPE0が状態記録ユニットのテーブルロック、テーブル書込み、及びテーブル更新を実行する。即ち、まずステップS2で状態記録ユニットのテーブルをロックして、他のプロセッサユニットPE1がデータ書込みできない状態とする。その後ステップS3で、当該画像ブロックに対して実行完了した処理のテーブルエントリを、実行中の状態を示す値から実行完了の状態を示す値に更新する。更にステップS4で、依存関係規定ユニットに規定される処理間の依存関係の情報に基づいて、当該画像ブロックの当該処理が完了することにより実行可能となる処理について、そのテーブルエントリを実行不可の状態を示す値から実行可の状態を示す値に更新する。

[0035] 次にステップS5で、上記処理が一番最初の処理であるか否かを判断する。一番最初の処理である場合には、他のプロセッサユニットPE1がまだ動作していない状態であるので、プロセッサユニットPE1の動作を開始させる必要がある。ステップS6で、プロセッサユニットPE0は、リモートプロセス設定フラグRMPset__flgを設定する。このときプロセッサユニットPE1は、ステップS7でリモートプロセス設定フラグRMPset__flgの設定待ちをしている。プロセッサユニットPE1は、ステップS6でのフラグ設定に応

答して、ステップS7からステップS8に移り、次の画像ブロックに対する第1の処理(CABAC, IT, IQ, MV)又は同一画像ブロックに対する第2の処理(+(MCA), DF)を実行する。

[0036] プロセッサユニットPE0は、ステップS9で、テーブルチェックして実行可能な処理があるか否かを判断する。実行可能な処理がない場合には、ステップS10でテーブルロック解除する。これにより他のプロセッサユニットPE1がテーブルに書き込める状態となる。その後プロセッサユニットPE0は、ステップS11でテーブルをロックし、ステップS12でテーブルチェックし、ステップS9で実行可能な処理があるか否かを再度判断する。実行可能な処理が現れるまで、このロック解除、ロック、テーブルチェック動作を繰り返す。

[0037] ステップS9で実行可能な処理が見つかった場合、プロセッサユニットPE0は、ステップS13で、テーブル中の実行可能な処理についてそのテーブルエントリを実行可能の状態を示す値から実行中の状態を示す値に更新する。その後ステップS14でテーブルロック解除する。更にステップS15で、プロセッサユニットPE0は、ステップS13で実行中の状態に更新した処理を自ら実行するために、所定の情報を示すフラグset_flgを設定する。更にステップS16で、プロセッサユニットPE0は、次の画像ブロックに対する第1の処理(CABAC, IT, IQ, MV)又は同一画像ブロックに対する第2の処理(+(MCA), DF)の何れかを実行する。その後、ステップS2以降の処理を繰り返す。

[0038] プロセッサユニットPE1側では、前述のように、ステップS8で第1の処理(CABAC, IT, IQ, MV)又は第2の処理(+(MCA), DF)を実行する。その処理が完了すると、ステップS17乃至S19において、プロセッサユニットPE1が状態記録ユニットのテーブルロック、テーブル書込み、及びテーブル更新を実行する。即ち、まずステップS17で状態記録ユニットのテーブルをロックして、他のプロセッサユニットPE0がデータ書込みできない状態とする。その後ステップS18で、当該画像ブロックに対して実行完了した処理のテーブルエントリを、実行中の状態を示す値から実行完了の状態を示す値に更新する。更にステップS19で、依存関係規定ユニットに規定される処理間の依存関係の情報に基づいて、当該画像ブロックの当該処理が完了することによ

り実行可能となる処理について、そのテーブルエントリを実行不可の状態を示す値から実行可の状態を示す値に更新する。

[0039] 次にステップS20で、テーブルチェックして実行可能な処理があるか否かを判断する。実行可能な処理がない場合には、ステップS21でテーブルロック解除する。これにより他のプロセッサユニットPE0がテーブルに書き込める状態となる。その後プロセッサユニットPE1は、ステップS22でテーブルをロックし、ステップS23でテーブルチェックし、ステップS20で実行可能な処理があるか否かを再度判断する。実行可能な処理が現れるまで、このロック解除、ロック、テーブルチェック動作を繰り返す。

[0040] ステップS20で実行可能な処理が見つかった場合、プロセッサユニットPE1は、ステップS24で、テーブル中の実行可能な処理についてそのテーブルエントリを実行可能の状態を示す値から実行中の状態を示す値に更新する。その後ステップS25でテーブルロック解除する。更にステップS26で、プロセッサユニットPE1は、ステップS24で実行中の状態に更新した処理を自ら実行するために、所定の情報を示すフラグset__flgを設定する。その後ステップS8に戻り、プロセッサユニットPE1は、第1の処理(CABAC, IT, IQ, MV)又は第2の処理(+(MCA), DF)の何れかを実行する。その後、ステップS17以降の処理を繰り返す。

[0041] 図11は、図10に示すテーブル更新及びテーブルチェック動作の一例を示すフローチャートである。図11のステップS1において、まずプロセッサユニットPE0又はPE1の何れかがテーブルロックを実行する。これは図10のステップS2又はステップS17のテーブルロック動作に対応する。

[0042] ステップS2において、実行中であった処理が第1の処理(CABAC, IT, IQ, MV)又は第2の処理(+(MCA), DF)の何れであるかを示す変数kindが、0であるか否かを判定する。変数kindが0であれば第1の処理(CABAC, IT, IQ, MV)が実行完了したことを示し、変数kindが1であれば第2の処理(+(MCA), DF)が実行完了したことを示している。

[0043] ステップS2において変数kindが0であることが判明した場合、ステップS3において、ピクチャ番号n(画像ブロック番号)を2で割った乗余が0であるか否かを判定する。1つの画像フレームは縦横に並べられた複数の画像ブロックに分割されており、例え

ば画像フレームの左上から主走査方向(横方向)及び副走査方向(縦方向)に順に走査したときの走査順に、各画像ブロックにピクチャ番号nが割り当てられている。このピクチャ番号n(画像ブロック番号)を2で割った乗余は、画像ブロックが偶数番目の画像ブロックであるか奇数番目の画像ブロックであるかを示している。

[0044] 乗余が0の場合、ステップS4で、配列値Pr[0][0]に実行完了を示す値"3"を設定する(即ちテーブルを更新する)。図12は、配列Pr[kind][m]を示す図である。初期状態において、配列Pr[kind][m]は、変数kindが0で乗余mが0である位置の配列値Pr[0][0]が実行可"1"であり、その他の位置の配列値が実行不可"0"であるような配列である。この初期状態から、変数kindの値0が示す第1の処理(CABAC, IT, IQ, MV)を乗余mの値0が示す偶数番目の画像ブロックに対して実行し、配列値Pr[0][0]が実行中"2"に更新される。その後、上記のようにステップS4で、配列値Pr[0][0]に実行完了を示す値"3"が設定される。

[0045] 更にステップS5で、乗余mの値0が示す偶数番目の画像ブロック(kindの値が0の処理が直前に完了した画像ブロック)に対する変数kindの値1が示す第2の処理(+(MCA), DF)を実行可に更新するために、配列値Pr[0][1]に実行可を示す値"1"を設定する。またステップS6で、乗余mの値1が示す奇数番目の画像ブロック(次の画像ブロック)に対する変数kindの値0が示す第1の処理(CABAC, IT, IQ, MV)を実行可に更新するために、配列値Pr[1][0]に実行可を示す値"1"を設定する。このステップS5とS6との処理は、依存関係規定ユニットの情報を参照して実行される。即ち、依存関係規定ユニットに規定される依存関係に従い、変数kindの値が0の処理(第1の処理)が完了したことにより実行可能となる処理について、実行可を示す値に更新する。

[0046] ステップS3での乗余が1の場合、ステップS7で、配列値Pr[1][0]に実行完了を示す値"3"を設定する(即ちテーブルを更新する)。更にステップS8で、乗余mの値1が示す奇数番目の画像ブロック(kindの値が0の処理が直前に完了した画像ブロック)に対する変数kindの値1が示す第2の処理(+(MCA), DF)を実行可に更新するために、配列値Pr[1][1]に実行可を示す値"1"を設定する。またステップS9で、乗余mの値0が示す偶数番目の画像ブロック(次の画像ブロック)に対する変数kindの

値0が示す第1の処理(CABAC, IT, IQ, MV)を実行可に更新するために、配列値Pr[0][0]に実行可を示す値"1"を設定する。このステップS7とS8との処理は、依存関係規定ユニットの情報を参照して実行される。即ち、依存関係規定ユニットに規定される依存関係に従い、変数kindの値が0の処理(第1の処理)が完了したことにより実行可能となる処理について、実行可を示す値に更新する。

[0047] またステップS2において変数kindが1であることが判明した場合、ステップS10で、配列値Pr[n%2][1]に実行完了を示す値"3"を設定する。即ち、変数kindの値が1の処理(第2の処理)が直前に完了した画像ブロックの第2の処理について、そのテーブルエントリを、実行完了を示す値"3"に更新する。この場合、テーブルの他のエントリについて、実行可能を示す値"1"に更新する動作は実行しない。即ちこの例においては、依存関係規定ユニットに規定される依存関係に従うと、変数kindの値が1の処理(第2の処理)が完了したことにより実行可能となる処理は存在しない。

[0048] ステップS6、S9、及びS10の何れかを実行した後に、ステップS11以降において実行可能な処理を実行する。

[0049] まずステップS11において、配列値Pr[(n+1)%2][0]=1であり且つ配列値Pr[(n+1)%2][1]=3(又は0)であるか否かが判定される。即ち、現在のピクチャ番号nの画像ブロックの次の画像ブロック(ピクチャ番号n+1)について、変数kindの値が0の処理(第1の処理)が実行可であり且つ変数kindの値が1の処理(第2の処理)が実行済み(又は実行不可)であるか否かを判定する。この条件が満たされる場合には、ステップS12で、配列値Pr[(n+1)%2][0]を実行可能を示す値"2"に設定し、ステップS13でテーブルロック解除する。更にステップS14で、ピクチャ番号nを1増加するとともに、変数kindを値0に設定する。その後ステップS25で、変数kindの値が0の処理(第1の処理(CABAC, IT, IQ, MV))を実行する。即ちこの場合には、元のピクチャ番号nの画像ブロックの次の画像ブロックに対して第1の処理を実行することになる。

[0050] ステップS11の判定結果がNoの場合には、ステップS15において、配列値Pr[n%2][1]=1であるか否かが判定される。即ち、現在のピクチャ番号nの画像ブロックについて、変数kindの値が1の処理(第2の処理)が実行可であるか否かを判定する。

この条件が満たされる場合には、ステップS16で、配列値Pr[n%2][1]を実行可能を示す値"2"に設定し、ステップS17でテーブルロック解除する。更にステップS18で、変数kindを値1に設定する。その後ステップS25で、変数kindの値が1の処理(第2の処理(+(MCA), DF))を実行する。即ちこの場合には、現在のピクチャ番号nの画像ブロックに対して第2の処理を実行することになる。

[0051] ステップS15の判定結果がNoの場合には、ステップS19において、配列値Pr[(n-1)%2][1]=1であるか否かが判定される。即ち、現在のピクチャ番号nの画像ブロックの直前の画像ブロックについて、変数kindの値が1の処理(第2の処理)が実行可であるか否かを判定する。この条件が満たされる場合には、ステップS20で、配列値Pr[(n-1)%2][1]を実行可能を示す値"2"に設定し、ステップS21でテーブルロック解除する。更にステップS22で、ピクチャ番号nを1減少させるとともに、変数kindを値1に設定する。その後ステップS25で、変数kindの値が1の処理(第2の処理(+(MCA), DF))を実行する。即ちこの場合には、元のピクチャ番号nの画像ブロックの直前の画像ブロックに対して第2の処理を実行することになる。

[0052] ステップS19の判定結果がNoの場合には、ステップS23でテーブルロック解除する。これにより他のプロセッサユニットがテーブルに書き込める状態となる。その後ステップS24で再度テーブルをロックし、ステップS11に戻り、以降の処理を繰り返す。

[0053] 図13は、本発明による動的処理割り当て処理の別の一例を示すフローチャートである。図13に示す例では、符号化処理を第1の処理(MV, MC, T, Q, CABAC)と第2の処理(IQ, IT, DF, MV, +(MCA), -)との2つの処理に分割し、これら2つの処理を2つのプロセッサユニット(計算機資源)PE0及びPE1に割り当てて実行している。ここで"T"は整数変換、"Q"は量子化、"-"は画像差分処理を示す。図10のフローチャートと比較して、2つのプロセッサユニット(計算機資源)PE0及びPE1に割り当てられる処理の種類が異なるだけであり、処理手順は同一であるので説明を省略する。

[0054] 図14は、図13に示すテーブル更新及びテーブルチェック動作の一例を示すフローチャートである。図14に示すフローチャートは、図11に示すフローチャート殆ど同一であり、各ステップの処理についての説明は省略する。但し、最後のステップS25

において、符号化のための第1の処理(MV, MC, T, Q, CABAC)と第2の処理(IQ, IT, DF, MV, +(MCA), −)との何れかを実行する点が、図11のステップS25において復号化のための第1の処理又は第2の処理の何れかを実行する点と異なる。

[0055] 図15は、H. 264の復号化処理における本発明による処理割り当ての一例を模式的に示す図である。図15の例において、復号化は5つの処理に分割され、5つのプロセッサユニットPE0乃至PE4により実行される。ここで5つの処理とは、CABAC、IT&IQ、MV&MV又はINTRA、+、及びDFである。図15において横軸は時間を表し、各プロセッサユニットは、まず状態記録ユニットのテーブルをチェック(図中の「チェック」)し、実行可能な処理を実行し、テーブルへの実行完了処理の書込み(図中の「書込み」)及び実行可となる処理の更新(図中の「更新」)を行う。図15に示される本発明による動的処理割り当てにおいて、各プロセッサユニットは、特定の処理の実行に固定的に割り当てられているのではなく、実行可の状態の処理を随時実行するように制御される。

[0056] 図16は、H. 264の復号化処理における従来技術による処理割り当ての一例を模式的に示す図である。図16の例において、復号化は5つの処理に分割され、5つのプロセッサユニットPE0乃至PE4により実行される。ここで5つの処理とは、CABAC、IT&IQ、MV&MV又はINTRA、+、及びDFである。図16において横軸は時間を表す。各プロセッサユニットは、特定の処理の実行に固定的に割り当てられており、実行すべき処理が実行可となるとその処理を実行する。データ処理量は処理対象の動画データに依存して変化するので、図16に示すような固定的な処理の割り当てに基づく制御では、プロセッサ間での待ち時間が発生することは避けられない。

[0057] それに対して図15に示すような動的な処理の割り当てに基づく制御では、処理を実行可能な計算機資源は1つに固定されていないので、利用可能な計算機資源に随時その処理を割り当てて実行していくことができる。従って、静的な処理割り当ての場合と比較して、処理対象の動画データ内容に依存してデータ処理量が変動しても、無駄な待ち時間を成るべく避けて効率的な処理を実行することができる。

[0058] 図17は、H. 264の復号化処理における各処理間の依存関係を説明するための図

である。依存関係規定ユニットに規定される複数の処理間の依存関係について説明するために、図17に示されるようなH．264の復号化処理を例として説明する。

[0059] H．264では16画素×16画素のマクロブロックMBを単位として復号化の各処理が実行される。複数の処理間の依存関係は、1つのマクロブロックMB内でのデータ依存関係と、1つのピクチャの異なるマクロブロックMB間でのデータ依存関係と、異なるピクチャ間でのデータ依存関係とを含む。

[0060] 図17のMB内データ依存関係70に示されるように、CABAC処理81、MV処理82、MC処理83、IT&IQ処理84、INTRA処理85、及びデブロッキングフィルタ処理86は、同一のマクロブロックMB内において図示の矢印で示されるようなデータ依存関係がある。即ち、CABAC処理81が実行完了しなければ、MV処理82、IT&IQ処理84、INTRA処理85は実行できない。またMV処理82が実行完了しなければ、MC処理83は実行できない。またMC処理83、IT&IQ処理84、及びINTRA処理85が実行完了しなければ、デブロッキングフィルタ処理86は実行できない。

[0061] またMB／ピクチャ間データ依存関係71に示すように、着目マクロブロックMBのMV処理82は、着目マクロブロックMBが含まれる処理対象ピクチャ90よりも時間的に過去のピクチャ91（アンカーピクチャ）の動きベクトルMVのデータに依存する場合がある。また着目マクロブロックMBのMV処理82は、着目マクロブロックMBの上、左、右上のマクロブロック92の動きベクトルMVに依存する場合がある。

[0062] またMB／ピクチャ間データ依存関係71に更に示すように、着目マクロブロックMBのMC処理83は、着目マクロブロックMBが含まれる処理対象ピクチャ90よりも時間的に過去のピクチャ93（既にデコードされた画像）の対応マクロブロックMBの画素データに依存する。

[0063] またMB／ピクチャ間データ依存関係72に示すように、着目マクロブロックMBのCABAC処理81は、着目マクロブロックMBの上及び左のマクロブロック94のデータに依存する。このデータはコンテキストに依存するが、コンテキストは先頭で初期化され、各マクロブロックMBのCABAC処理81毎に更新される。従って、着目マクロブロックMBのCABAC処理81は、着目マクロブロックMBの上及び左のマクロブロック94のCABAC処理81に依存する。

[0064] またMB／ピクチャ間データ依存関係72に示すように、着目マクロブロックMBのINTRA処理85は、着目マクロブロックMBの左上、上、左、及び右上のマクロブロック95のデブロッキングフィルタ処理前の画素データに依存する。

[0065] またMB／ピクチャ間データ依存関係72に示すように、着目マクロブロックMBのデブロッキングフィルタ処理86は、着目マクロブロックMBの上及び左のマクロブロック96のデブロッキングフィルタ処理後の画素データに依存する。

[0066] 図18は、H. 264の復号化処理において1つのマクロブロックに対するCABAC処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。図18に示す状態記録ユニットのテーブルには、3つのフレーム(ピクチャ)Frame1乃至Frame3の各マクロブロックM1乃至Mnについての状態データが記録されている。状態データ"0"は実行不可、"1"は実行可、"2"は実行中、"3"は実行完了を示す。

[0067] 図18の上段には、フレームFrame1のマクロブロックM1のCABAC処理について、状態データを実行可"1"から実行中"2"に更新した様子が示される。図18の下段には、フレームFrame1のマクロブロックM1のCABAC処理について、状態データを実行中"2"から実行完了"3"に更新した様子が示される。この例では更に、CABAC処理の実行完了に応答して、同一フレームFrame1の同一のマクロブロックM1のIT&IQ処理について実行不可"0"から実行可"1"に更新するとともに、同一フレームFrame1の着目マクロブロックM1の右隣のマクロブロックM2のCABAC処理について実行不可"0"から実行可"1"に更新している。

[0068] 図19は、H. 264の復号化処理において1つのマクロブロックに対するIT&IQ処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。図19に示す状態記録ユニットのテーブルには、3つのフレーム(ピクチャ)Frame1乃至Frame3の各マクロブロックM1乃至Mnについての状態データが記録されている。状態データ"0"は実行不可、"1"は実行可、"2"は実行中、"3"は実行完了を示す。

[0069] 図19の上段には、フレームFrame1のマクロブロックM1のIT&IQ処理について、状態データを実行可"1"から実行中"2"に更新した様子が示される。図19の下段に

は、フレームFrame1のマクロブロックM1のIT&IQ処理について、状態データを実行中"2"から実行完了"3"に更新した様子が示される。この例では更に、IT&IQ処理の実行完了に応答して、同一フレームFrame1の同一のマクロブロックM1のMV処理、MC処理、又はINTRA処理について実行不可"0"から実行可"1"に更新してある。

[0070] 図20は、H. 264の復号化処理において1つのマクロブロックに対するMV処理及びMC処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。図20に示す状態記録ユニットのテーブルには、2つのフレーム(ピクチャ)Frame1及びFrame2の各マクロブロックM1乃至Mnについての状態データが記録されている。状態データ"0"は実行不可、"1"は実行可、"2"は実行中、"3"は実行完了を示す。なお各フレームの一行にはm－1個のマクロブロックが設けられており、マクロブロックMmはマクロブロックM1の直下の位置にあるとする。

[0071] 図20の上段には、フレームFrame1のマクロブロックM1のMV処理及びMC処理について、状態データを実行可"1"から実行中"2"に更新した様子が示される。図20の下段には、フレームFrame1のマクロブロックM1のMV処理及びMC処理について、状態データを実行中"2"から実行完了"3"に更新した様子が示される。この例では更に、MV処理及びMC処理の実行完了に応答して、同一フレームFrame1の同一のマクロブロックM1の加算処理"＋"について実行不可"0"から実行可"1"に更新するとともに、次のフレームFrame2の対応マクロブロックM1のMV処理、MC処理、又はINTRA処理について実行不可"0"から実行可"1"に更新してある。

[0072] 図21は、H. 264の復号化処理において1つのマクロブロックに対するMV処理及びMC処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。図21は、図20に示すMV処理及びMC処理とは依存関係が異なる場合を示している。図21に示す状態記録ユニットのテーブルには、2つのフレーム(ピクチャ)Frame1及びFrame2の各マクロブロックM1乃至Mnについての状態データが記録されている。状態データ"0"は実行不可、"1"は実行可、"2"は実行中、"3"は実行完了を示す。なお各フレームの一行にはm－1個のマクロブロックが設けられており、マクロブロックMmはマクロブロックM1の直下の位置にあるとする。

[0073] 図21の上段には、フレームFrame2のマクロブロックM2のMV処理及びMC処理について、状態データを実行可"1"から実行中"2"に更新した様子が示される。図21の下段には、フレームFrame2のマクロブロックM2のMV処理及びMC処理について、状態データを実行中"2"から実行完了"3"に更新した様子が示される。この例では更に、MV処理及びMC処理の実行完了に応答して、同一フレームFrame2の同一のマクロブロックM2の加算処理"＋"について実行不可"0"から実行可"1"に更新するとともに、同一フレームFrame2の着目マクロブロックM2の左下のマクロブロックMmのMV処理、MC処理、又はINTRA処理について実行不可"0"から実行可"1"に更新してある。

[0074] 図22は、H. 264の復号化処理において1つのマクロブロックに対するINTRA処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。図22に示す状態記録ユニットのテーブルには、2つのフレーム(ピクチャ)Frame1及びFrame2の各マクロブロックM1乃至Mnについての状態データが記録されている。状態データ"0"は実行不可、"1"は実行可、"2"は実行中、"3"は実行完了を示す。なお各フレームの一行にはm－1個のマクロブロックが設けられており、マクロブロックMmはマクロブロックM1の直下の位置にあるとする。

[0075] 図22の上段には、フレームFrame2のマクロブロックM2のINTRA処理について、状態データを実行可"1"から実行中"2"に更新した様子が示される。図22の下段には、フレームFrame2のマクロブロックM2のINTRA処理について、状態データを実行中"2"から実行完了"3"に更新した様子が示される。この例では更に、INTRA処理の実行完了に応答して、同一フレームFrame2の同一のマクロブロックM2の加算処理"＋"について実行不可"0"から実行可"1"に更新するとともに、同一フレームFrame2の着目マクロブロックM2の左下のマクロブロックMmのMV処理、MC処理、又はINTRA処理について実行不可"0"から実行可"1"に更新してある。

[0076] 図23は、H. 264の復号化処理において1つのマクロブロックに対する加算処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。ここで加算処理とは、図1又は図17に示されるようにMC処理後の動き補償データをIT&IQ処理後の画像データに加算する処理、又はINTRA処理後のフレーム

内予測データをIT&IQ処理後の画像データに加算する処理のことである。図23に示す状態記録ユニットのテーブルには、3つのフレーム(ピクチャ)Frame1乃至Frame3の各マクロブロックM1乃至Mnについての状態データが記録されている。状態データ"0"は実行不可、"1"は実行可、"2"は実行中、"3"は実行完了を示す。

[0077] 図23の上段には、フレームFrame1のマクロブロックM1の加算処理"+"について、状態データを実行可"1"から実行中"2"に更新した様子が示される。図23の下段には、フレームFrame1のマクロブロックM1の加算処理"+"について、状態データを実行中"2"から実行完了"3"に更新した様子が示される。この例では更に、加算処理"+"の実行完了に応答して、同一フレームFrame1の同一のマクロブロックM1のデブロッキングフィルタ処理DFについて実行不可"0"から実行可"1"に更新している。

[0078] 図24は、H. 264の復号化処理において1つのマクロブロックに対するデブロッキングフィルタ処理が実行完了したときの状態記録ユニットのテーブル更新について説明するための図である。図24に示す状態記録ユニットのテーブルには、2つのフレーム(ピクチャ)Frame1及びFrame2の各マクロブロックM1乃至Mnについての状態データが記録されている。状態データ"0"は実行不可、"1"は実行可、"2"は実行中、"3"は実行完了を示す。なお各フレームの一行にはm－1個のマクロブロックが設けられており、マクロブロックMmはマクロブロックM1の直下の位置にあるとする。

[0079] 図24の上段には、フレームFrame2のマクロブロックM1のデブロッキングフィルタ処理DFについて、状態データを実行可"1"から実行中"2"に更新した様子が示される。図24の下段には、フレームFrame2のマクロブロックM1のデブロッキングフィルタ処理DFについて、状態データを実行中"2"から実行完了"3"に更新した様子が示される。この例では更に、デブロッキングフィルタ処理DFの実行完了に応答して、同一フレームFrame2の着目マクロブロックM1の次のマクロブロックM2のデブロッキングフィルタ処理DFについて実行不可"0"から実行可"1"に更新するとともに、同一フレームFrame2の着目マクロブロックM1の直下のマクロブロックMmのデブロッキングフィルタ処理DFについて実行不可"0"から実行可"1"に更新してある。

[0080] 以上、本発明を実施例に基づいて説明したが、本発明は上記実施例に限定される

ものではなく、特許請求の範囲に記載の範囲内で様々な変形が可能である。

請求の範囲

[1] 複数の処理から構成される符号化／復号化を動画像データに対して実行する計算機システムであって、

複数の計算機資源と、

該動画像データの2つ又はそれ以上の画像ブロックの各々に対して、該複数の処理毎に実行不可、実行可、実行中、実行完了の何れかの状態を記録した状態記録ユニットと、

該複数の処理の間の依存関係を示す依存関係規定ユニット

を含み、

該画像ブロックに対する第1処理の完了に応答して、該状態記録ユニットにおける該第1処理の状態を、実行中の状態から実行完了の状態に更新するとともに、

該依存関係規定ユニットを参照して、該第1処理の実行完了に応じて実行可となる第2処理を特定し、該状態記録ユニットにおける該特定された第2処理の状態を、実行不可の状態から実行可の状態に更新し、

該第2処理を該複数の計算機資源のうちの少なくとも1つに割り当てて実行させるとともに、

該第2処理を実行可の状態から実行中の状態に更新する

ように該計算機資源及び該状態記録ユニットの制御動作を実行することを特徴とする計算機システム。

[2] 該計算機資源及び該状態記録ユニットの該制御動作を実行する制御ユニットを更に含むことを特徴とする請求項1記載の計算機システム。

[3] 該複数の計算機資源の各々が該依存関係規定ユニットを参照し、該複数の計算機資源の各々が該状態記録ユニットの内容を更新して処理を実行することを特徴とする請求項1又は2記載の計算機システム。

[4] 該複数の計算機資源による該状態記録ユニットのアクセス及び更新が排他的となるように該状態記録ユニットに対するロック及びロック解除の機能が設けられていることを特徴とする請求項3記載の計算機システム。

[5] 該依存関係規定ユニットは、該複数の処理の間の依存関係を表現したデータを格

納するよう構成されたメモリであることを特徴とする請求項1～4のいずれか一つに記載の計算機システム。

[6] 該状態記録ユニットは、該動画像データの2つ又はそれ以上の該画像ブロックの各々に対して該複数の処理の各処理毎に実行不可、実行可、実行中、実行完了の何れかの状態を示すデータを格納するよう構成されたメモリであることを特徴とする請求項1～5のいずれか一つに記載の計算機システム。

[7] 該状態記録ユニットの該メモリに格納されるデータは、該2つ又はそれ以上の該画像ブロックを行又は列の一方とし、該複数の処理を行又は列の他方としたテーブル形式のデータであることを特徴とする請求項6記載の計算機システム。

[8] 該複数の計算機資源はマルチプロセッサシステムの複数のプロセッサであることを特徴とする請求項1～7のいずれか一つに記載の計算機システム。

[9] 該複数の計算機資源はASICの複数の処理ユニットであることを特徴とする請求項1～7のいずれか一つに記載の計算機システム。

[10] 複数の処理から構成される符号化／復号化を複数の計算機資源により実行する符号化／復号化方法であって、

動画像データの2つ又はそれ以上の画像ブロックの各々に対して該複数の処理の処理毎に実行不可、実行可、実行中、実行完了の何れかの状態をメモリに記録し、

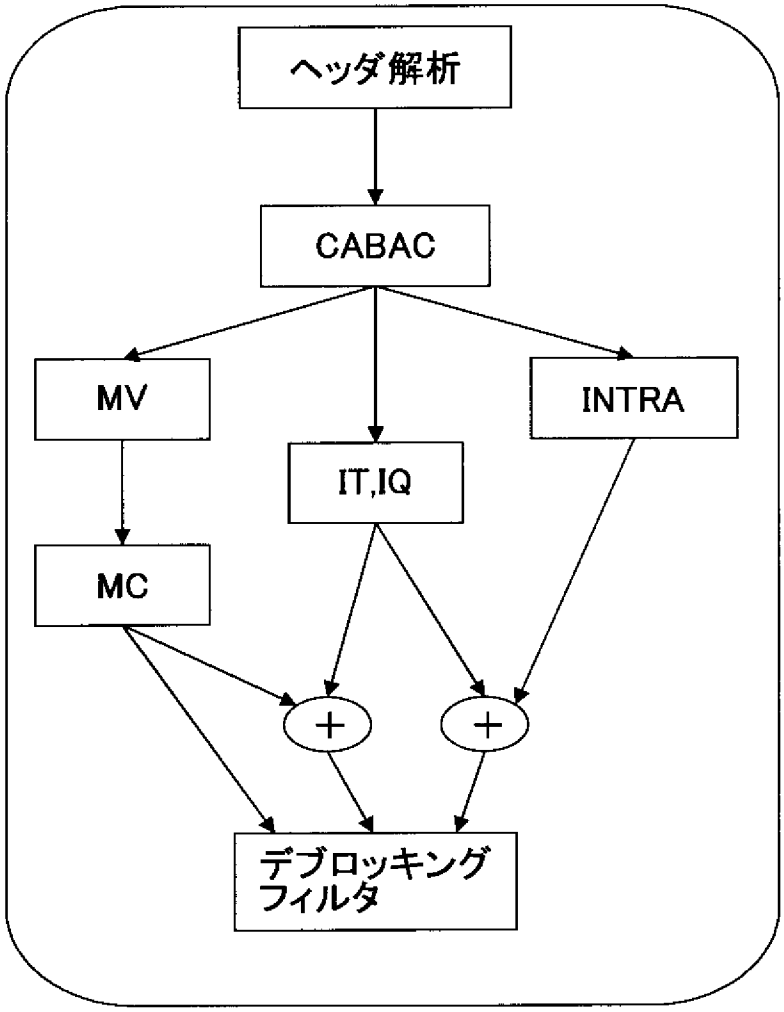
該画像ブロックに対する第1処理の完了に応答して、該メモリにおける該第1処理の状態を、実行中の状態から実行完了の状態に更新し、

該複数の処理の間の依存関係を参照して、該第1処理の実行完了に応じて実行可となる第2処理を特定して、該メモリにおいて該特定された該第2処理の状態を、実行不可の状態から実行可の状態に更新し、

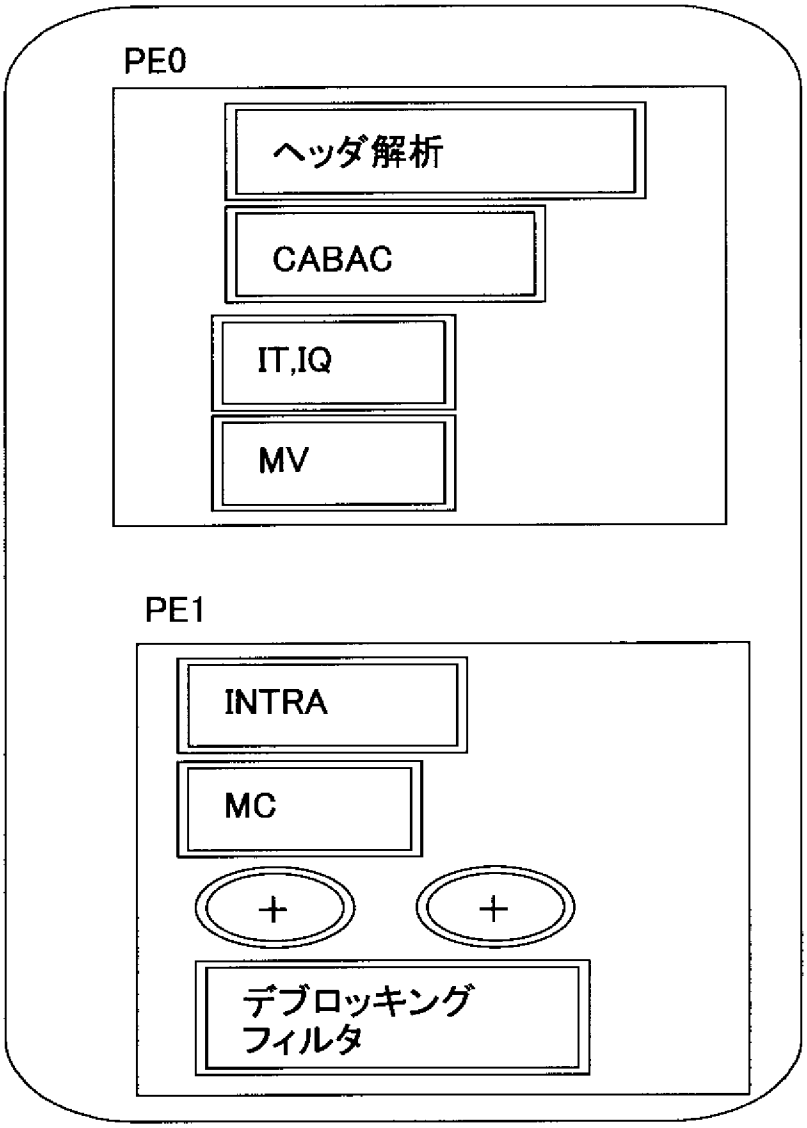
該メモリにおいて実行可として記録された該第2処理を、実行可の状態から実行中の状態に更新するとともに、該第2処理を該複数の計算機資源のうちの少なくとも1つに割り当てて実行させる

ことを特徴とする符号化／復号化方法。

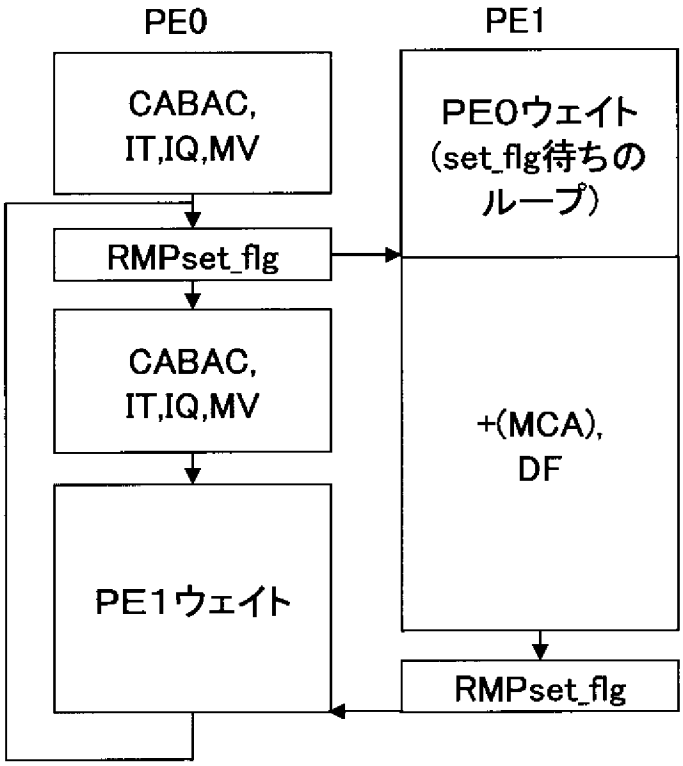
[図1]



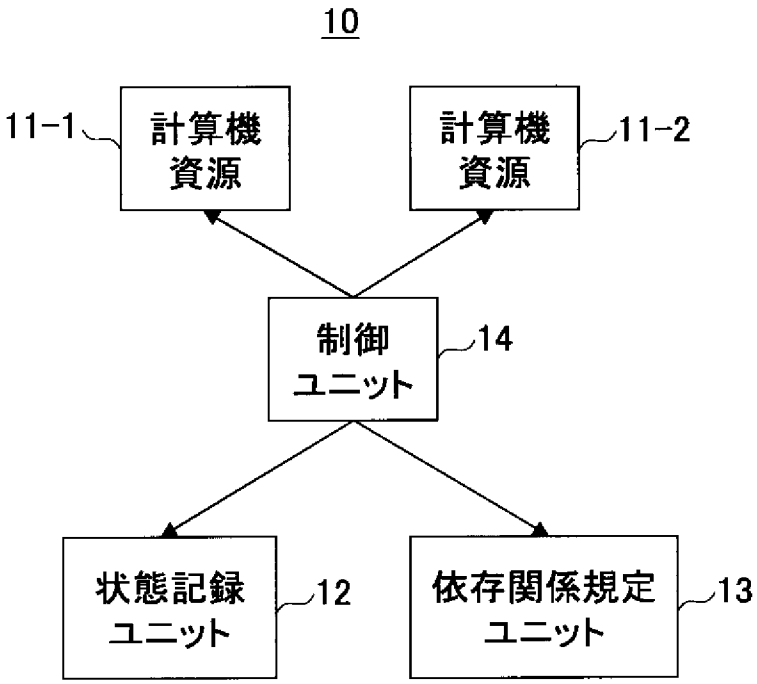
[図2]



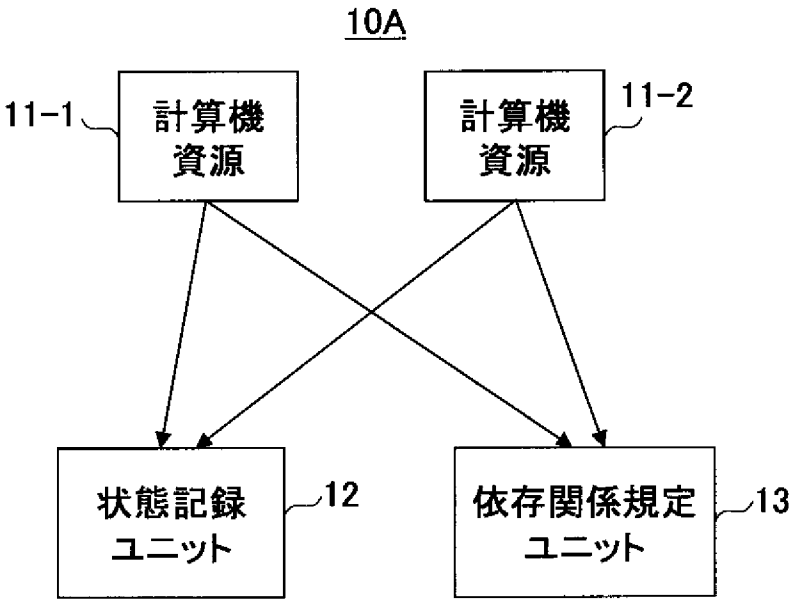
[図3]


PE0
PE1
CABAC,
IT,IQ,MV
PE0ウェイト
(set_flg待ちの
ループ)
RMPset_flg
CABAC,
IT,IQ,MV
+(MCA),
DF
PE1ウェイト
RMPset_flg

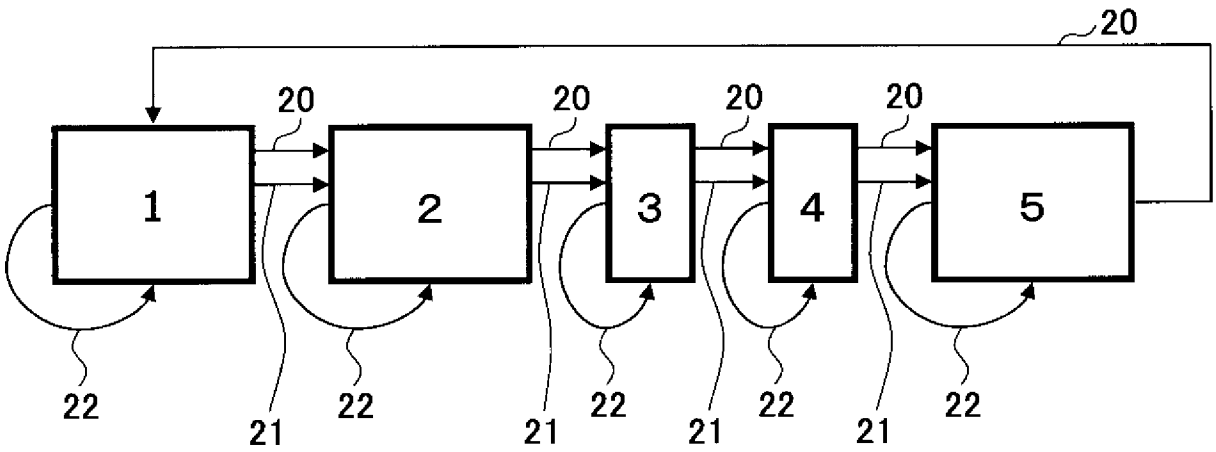
[図4]


10
11-1
計算機
資源
計算機
資源
11-2
制御
ユニット
14
状態記録
ユニット
12
依存関係規定
ユニット
13

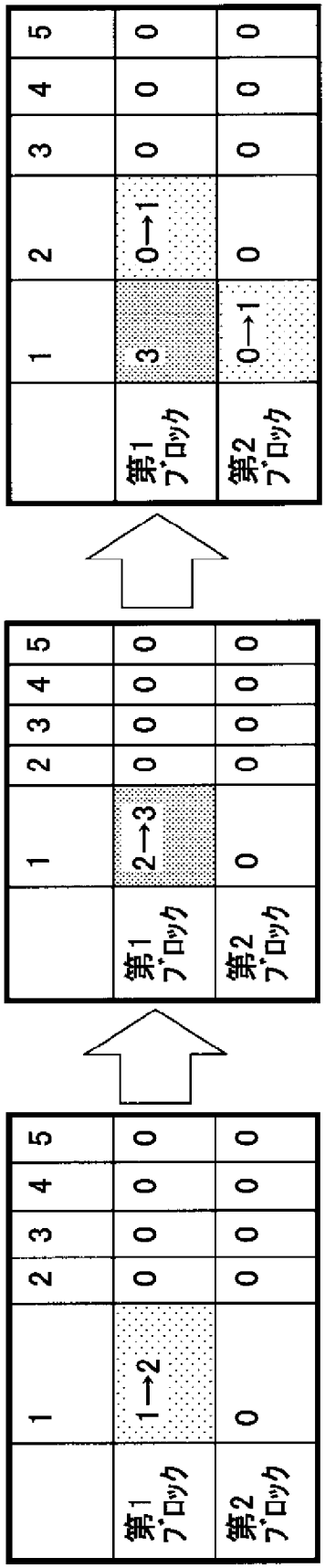
[図5]


10A
11-1
計算機資源
11-2
計算機資源
状態記録ユニット
12
依存関係規定ユニット
13

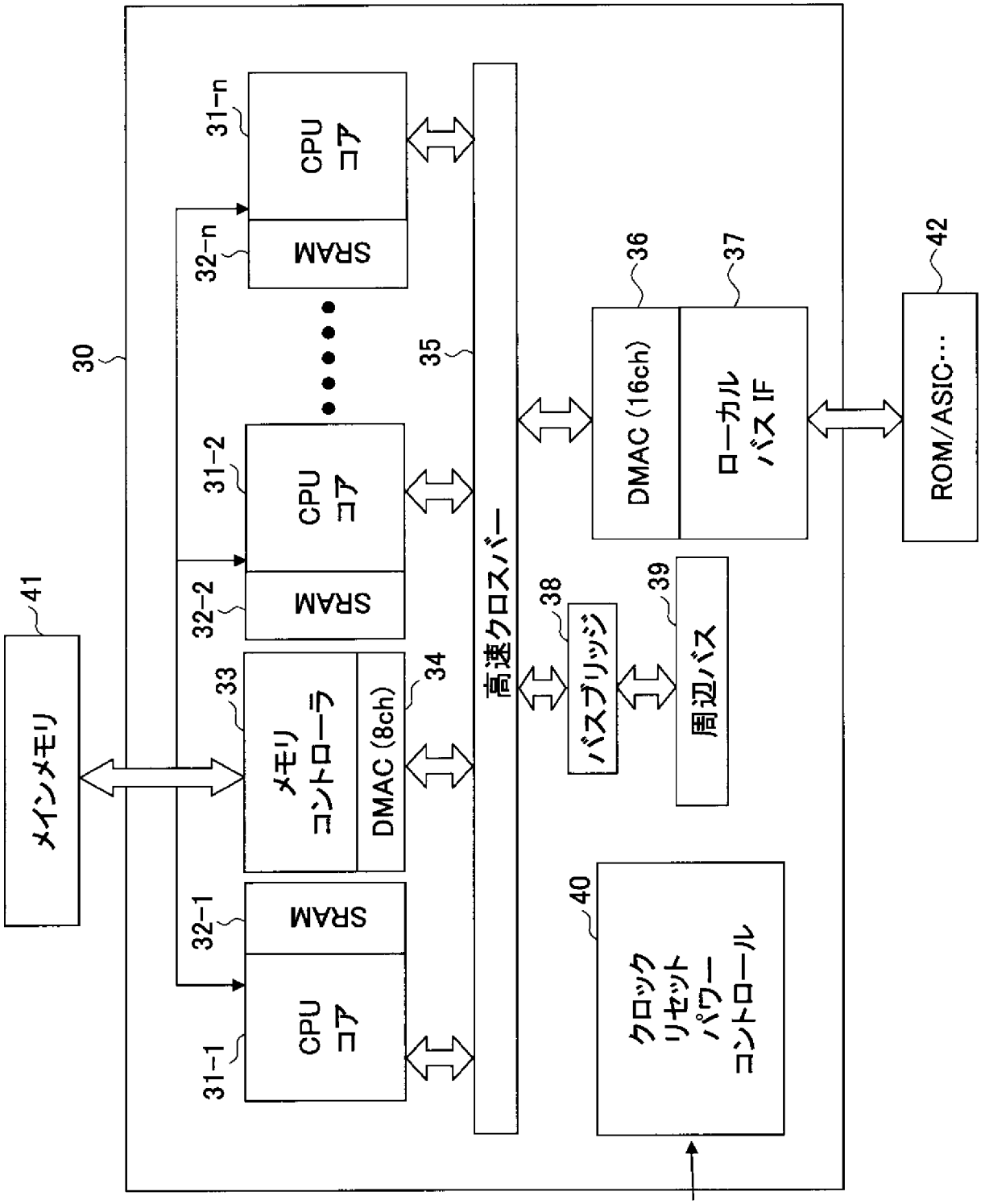
[図6]


20
1
2
3
4
5
20
20
20
20
22
21
22
21
22
21
22
21
22

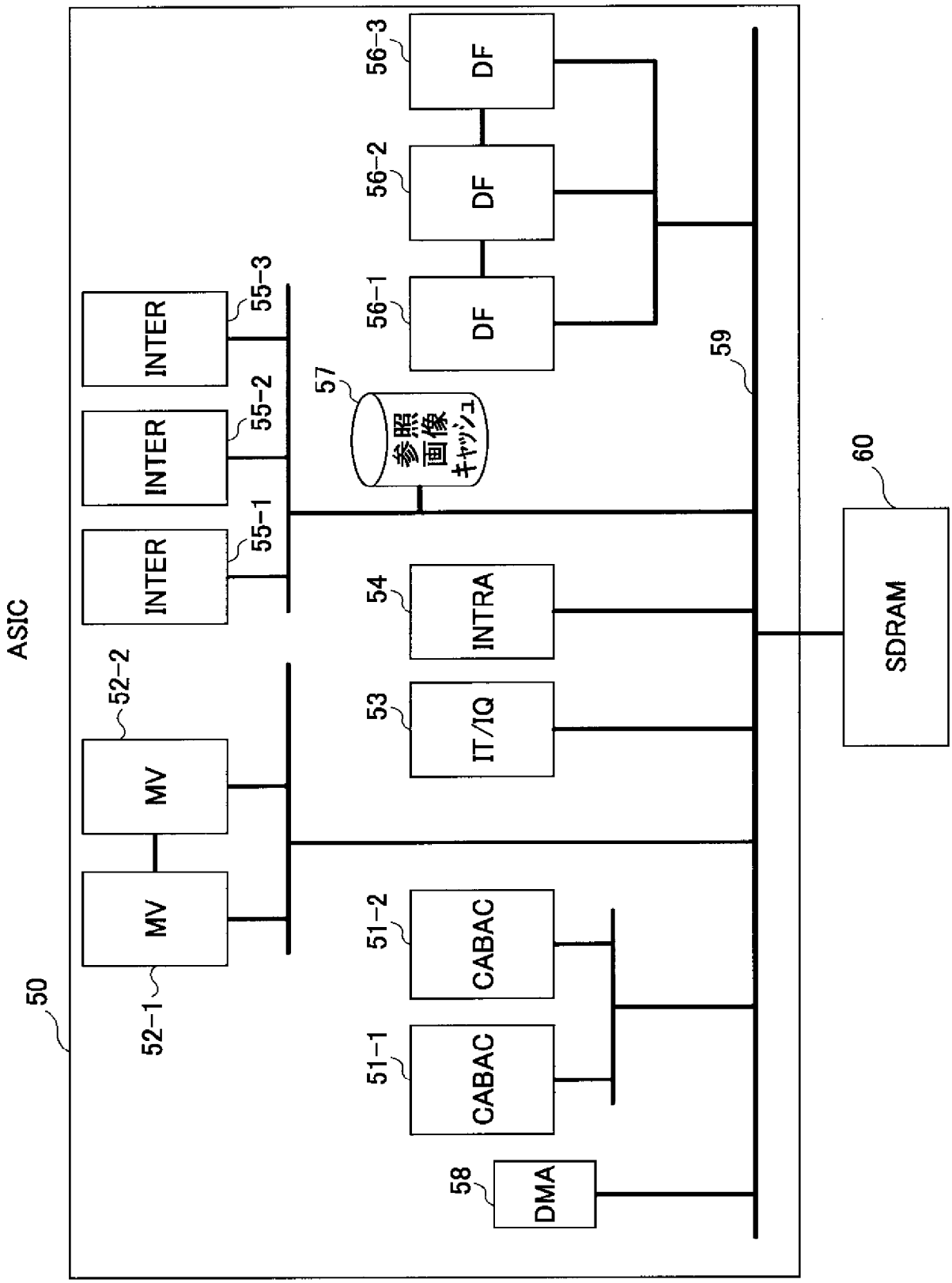
[図7]



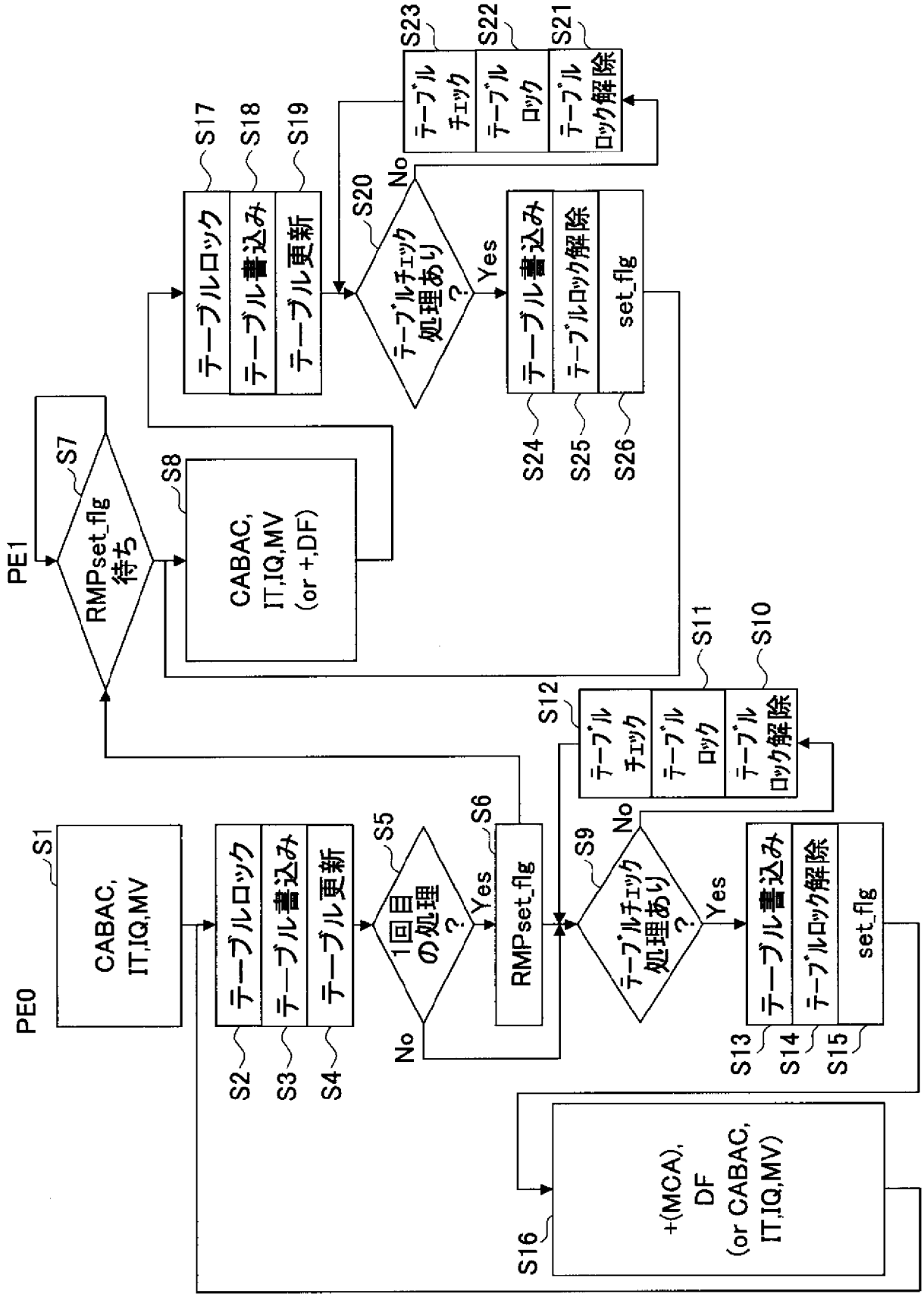
[図8]


30
31-n
CPU コア
32-n
SRAM
31-2
CPU コア
32-2
SRAM
33
メモリ コントローラ
34
DMAC (8ch)
32-1
SRAM
31-1
CPU コア
35
高速クロスバー
36
DMAC (16ch)
37
ローカル バス IF
42
ROM/ASIC…
41
メインメモリ
38
バスブリッジ
39
周辺バス
40
クロック リセット パワー コントロール

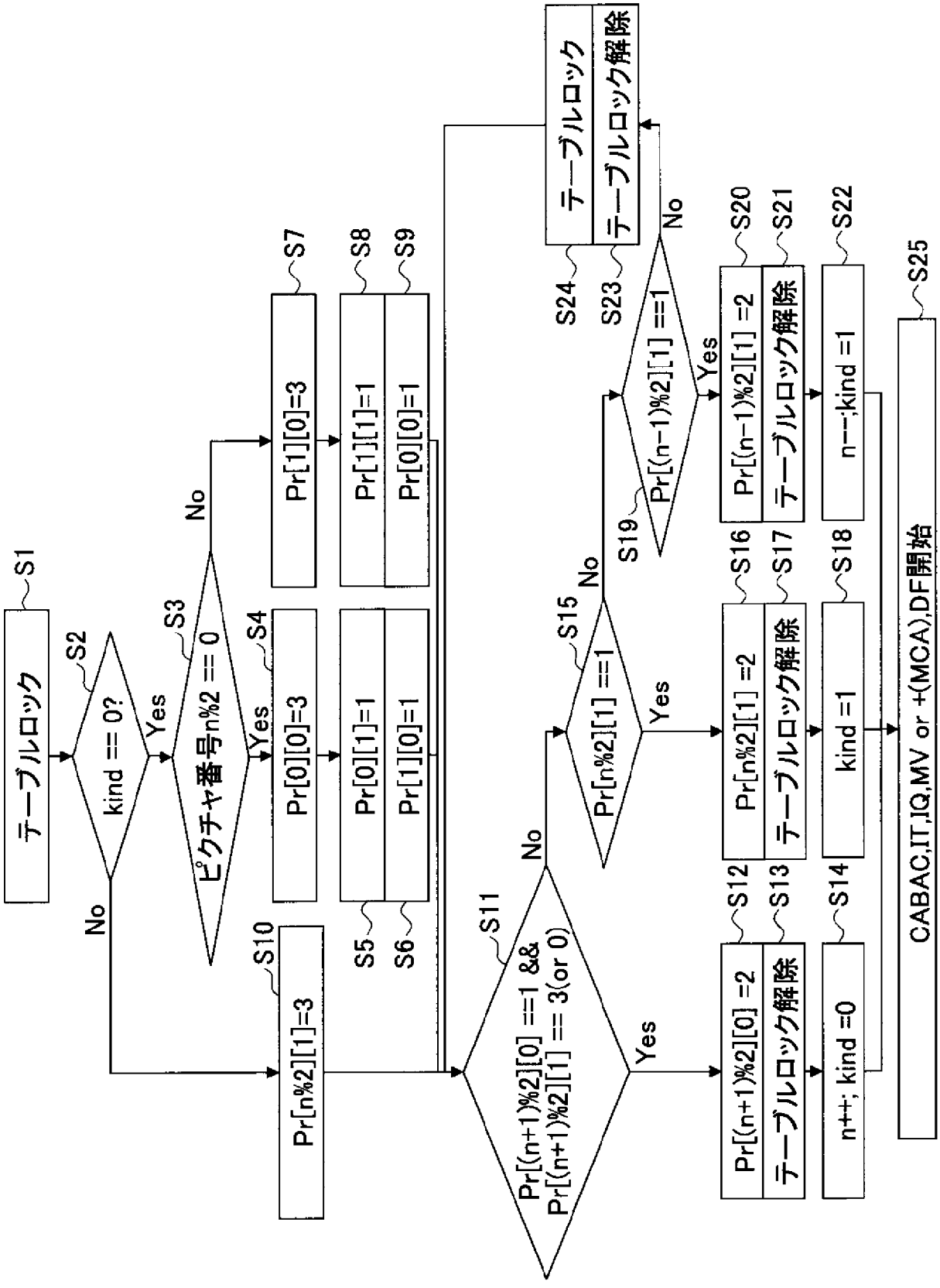
[図9]


ASIC
50
52-1
MV
52-2
MV
INTER
55-1
INTER
55-2
INTER
55-3
51-1
CABAC
51-2
CABAC
53
IT/IQ
54
INTRA
58
DMA
57
参照画像キャッシュ
56-1
DF
56-2
DF
56-3
DF
59
60
SDRAM

[図10]


PE0
S1
CABAC, IT,IQ,MV
S2
テーブルロック
S3
テーブル書込み
S4
テーブル更新
S5
1回目の処理?
Yes
No
S6
RMPset_flg
PE1
S7
RMPset_flg待ち
S8
CABAC, IT,IQ,MV (or +,DF)
S9
テーブルチェック処理あり?
Yes
No
S10
テーブルロック解除
S11
テーブルロック
S12
テーブルチェック
S13
テーブル書込み
S14
テーブルロック解除
S15
set_flg
S16
+(MCA), DF (or CABAC, IT,IQ,MV)
S17
テーブルロック
S18
テーブル書込み
S19
テーブル更新
S20
テーブルチェック処理あり?
Yes
No
S21
テーブルロック解除
S22
テーブルロック
S23
テーブルチェック
S24
テーブル書込み
S25
テーブルロック解除
S26
set_flg

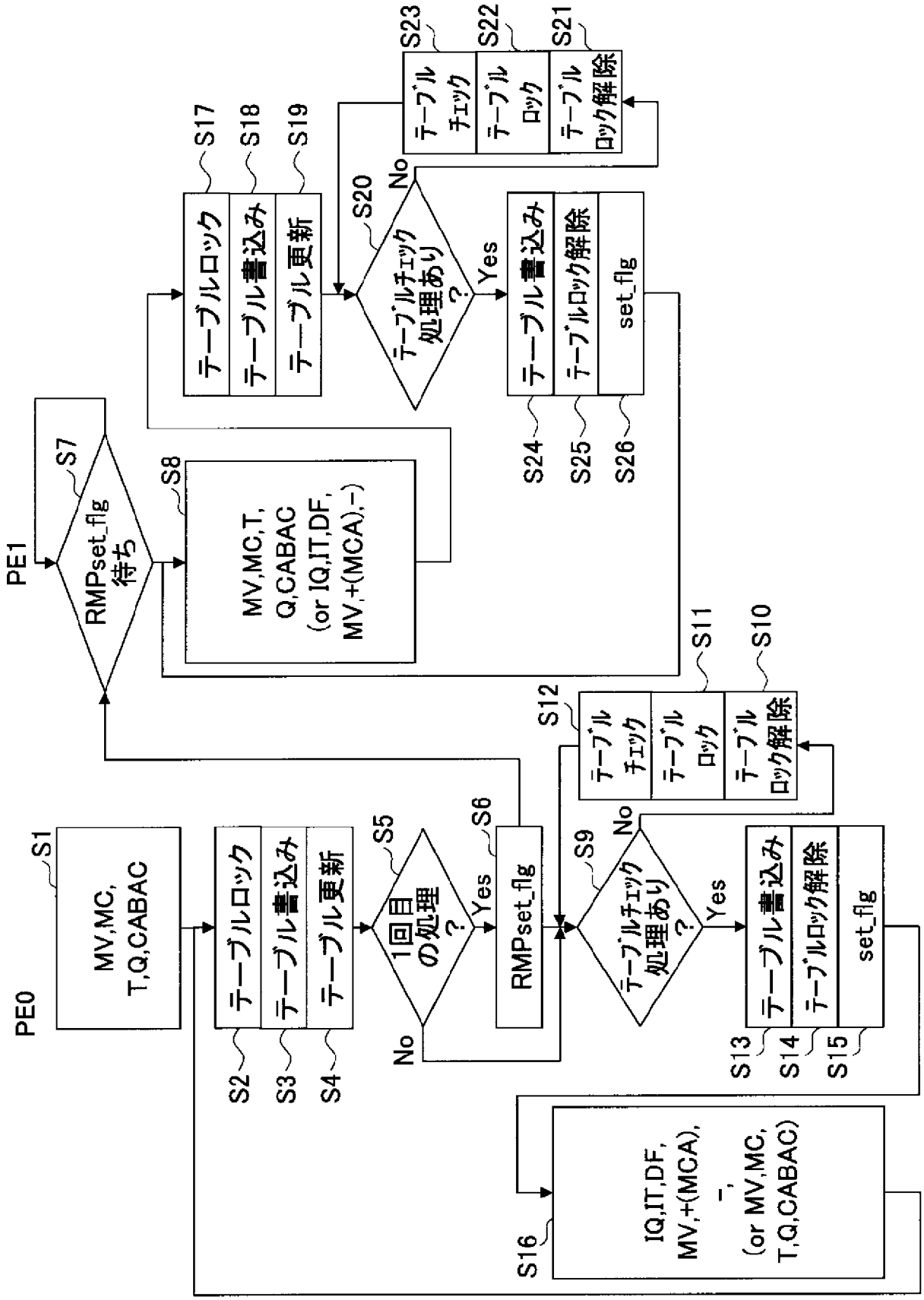
[図11]


テーブルロック
S1
kind == 0?
S2
Yes
No
ピクチャ番号n%2 == 0
S3
Yes
No
Pr[0][0]=3
S4
Pr[1][0]=3
S7
Pr[0][1]=1
Pr[1][0]=1
S5
S6
Pr[1][1]=1
Pr[0][0]=1
S8
S9
Pr[n%2][1]=3
S10
Pr[(n+1)%2][0] ==1 &&
Pr[(n+1)%2][1] == 3(or 0)
S11
Yes
No
Pr[n%2][1] ==1
S15
Yes
No
Pr[(n-1)%2][1] ==1
S19
Yes
No
テーブルロック
テーブルロック解除
S24
S23
Pr[(n+1)%2][0] =2
テーブルロック解除
S12
S13
Pr[n%2][1] =2
テーブルロック解除
S16
S17
Pr[(n-1)%2][1] =2
テーブルロック解除
S20
S21
n++; kind =0
S14
kind =1
S18
n--;kind =1
S22
CABAC,IT,IQ,MV or +(MCA),DF開始
S25

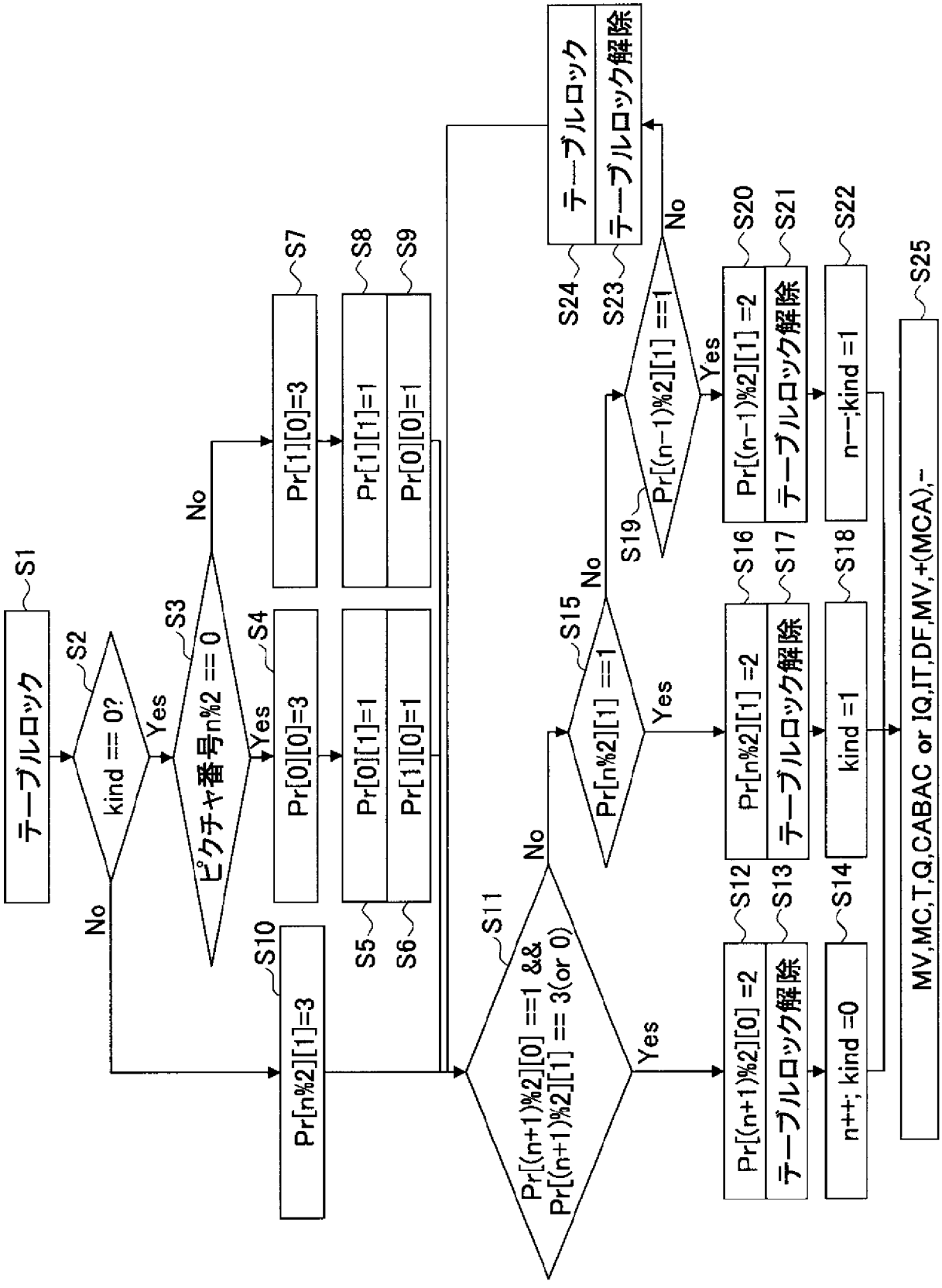
[図12]

kind \ m	0	1
CABAC,IT,IQ,MV	1	0
+(MCA,DF)	0	0

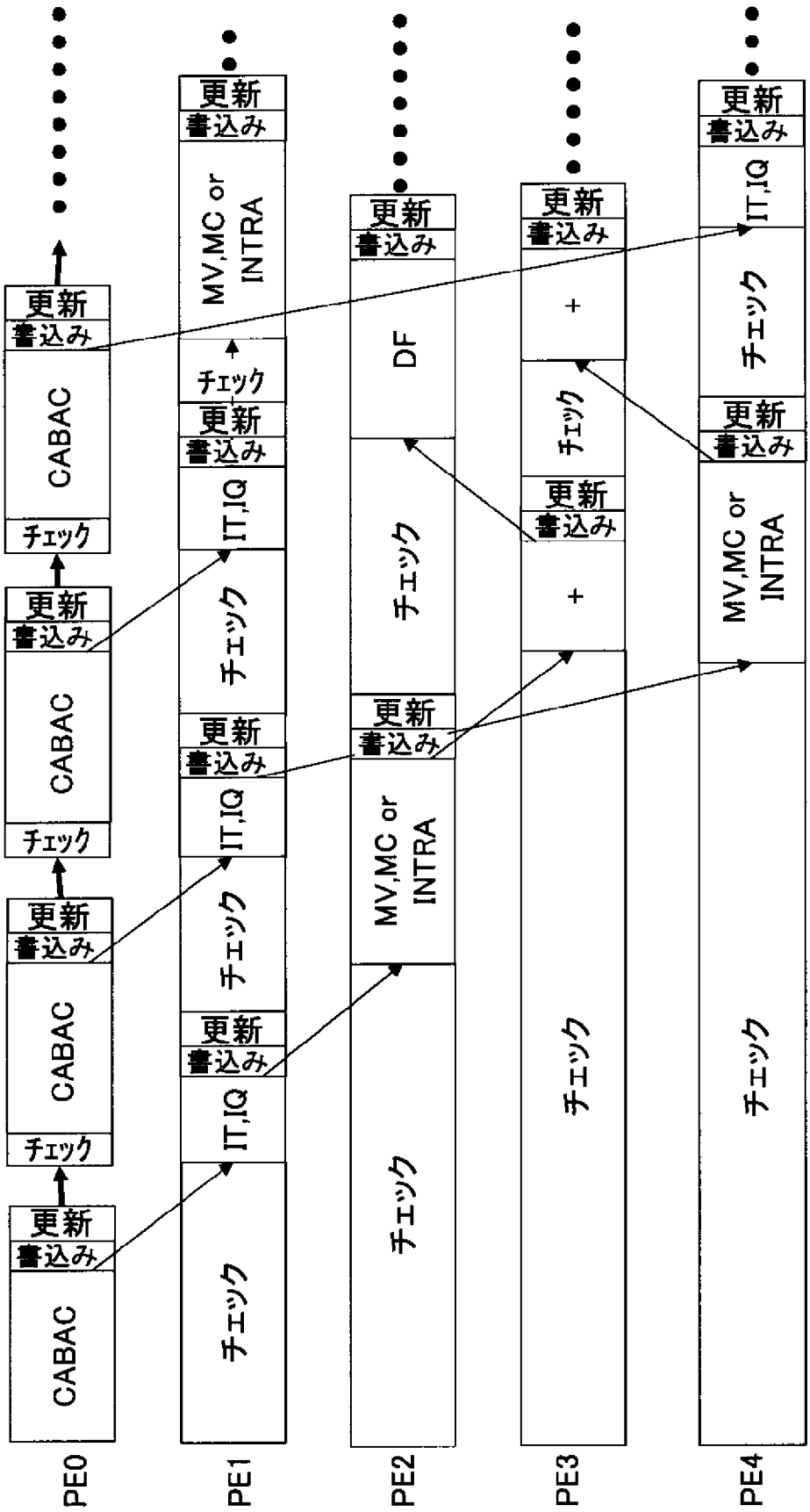
[図13]


PE0
S1
MV,MC,
T,Q,CABAC
テーブルロック
テーブル書込み
テーブル更新
S2
S3
S4
S5
1回目の処理?
No
Yes
S6
RMPset_flg
PE1
S7
RMPset_flg待ち
S8
MV,MC,T,
Q,CABAC
(or IQ,IT,DF,
MV,+(MCA),-)
S9
テーブルチェック処理あり?
No
Yes
S10
テーブルロック解除
S11
テーブルロック
S12
テーブルチェック
S13
テーブル書込み
S14
テーブルロック解除
S15
set_flg
S16
IQ,IT,DF,
MV,+(MCA),
-,
(or MV,MC,
T,Q,CABAC)
S17
テーブルロック
S18
テーブル書込み
S19
テーブル更新
S20
テーブルチェック処理あり?
No
Yes
S21
テーブルロック解除
S22
テーブルロック
S23
テーブルチェック
S24
テーブル書込み
S25
テーブルロック解除
S26
set_flg

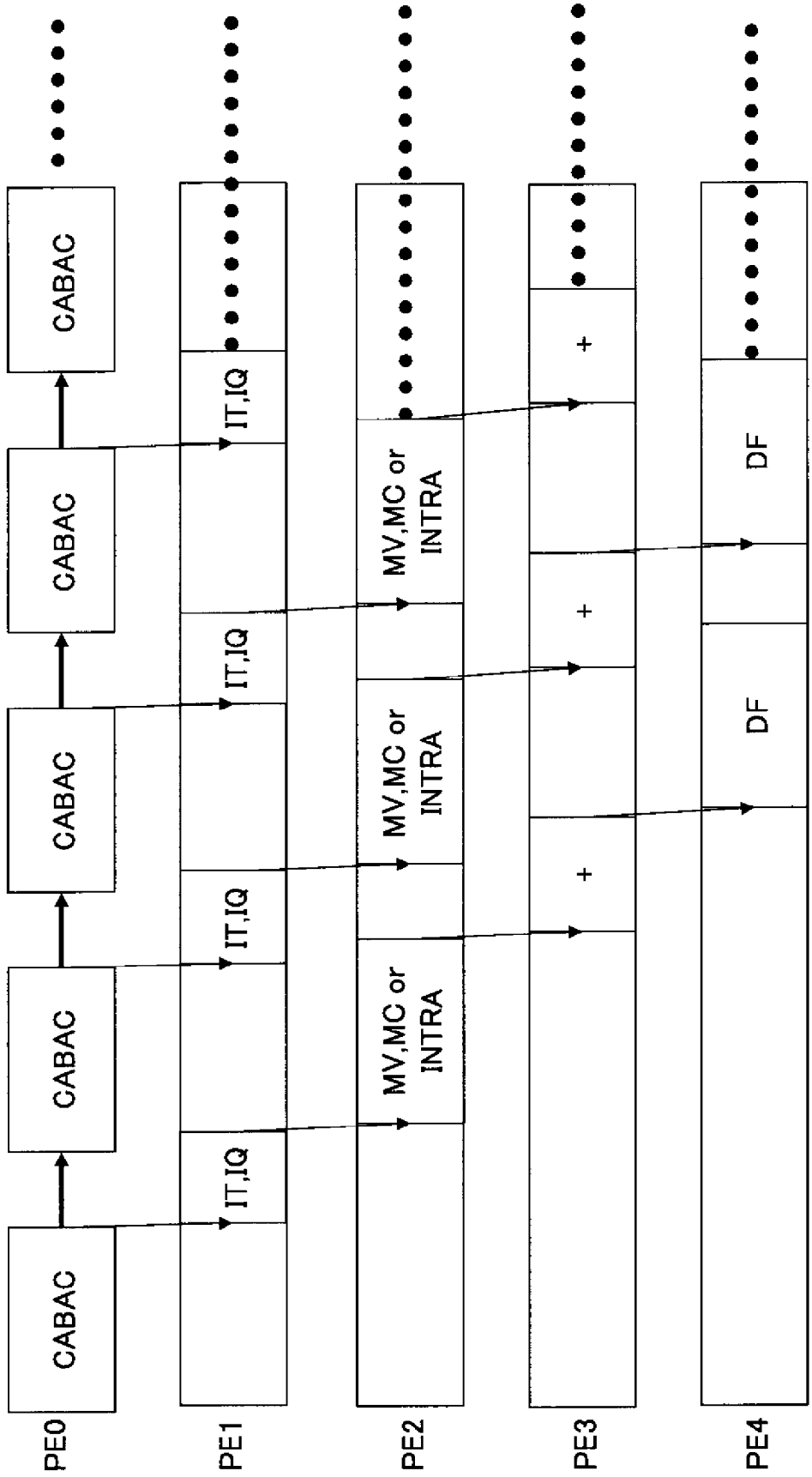
[図14]


テーブルロック
S1
kind == 0?
S2
Yes
No
ピクチャ番号n%2 == 0
S3
Yes
No
Pr[n%2][1]=3
S10
Pr[0][0]=3
S4
Pr[0][1]=1
Pr[1][0]=1
S5
S6
Pr[1][0]=3
S7
Pr[1][1]=1
Pr[0][0]=1
S8
S9
Pr[(n+1)%2][0] ==1 &&
Pr[(n+1)%2][1] == 3(or 0)
S11
Yes
No
Pr[(n+1)%2][0] =2
テーブルロック解除
S12
S13
n++; kind =0
S14
Pr[n%2][1] ==1
S15
Yes
No
Pr[n%2][1] =2
テーブルロック解除
S16
S17
kind =1
S18
Pr[(n-1)%2][1] ==1
S19
Yes
No
テーブルロック
テーブルロック解除
S24
S23
Pr[(n-1)%2][1] =2
テーブルロック解除
S20
S21
n--;kind =1
S22
MV,MC,T,Q,CABAC or IQ,IT,DF,MV,+(MCA),-
S25

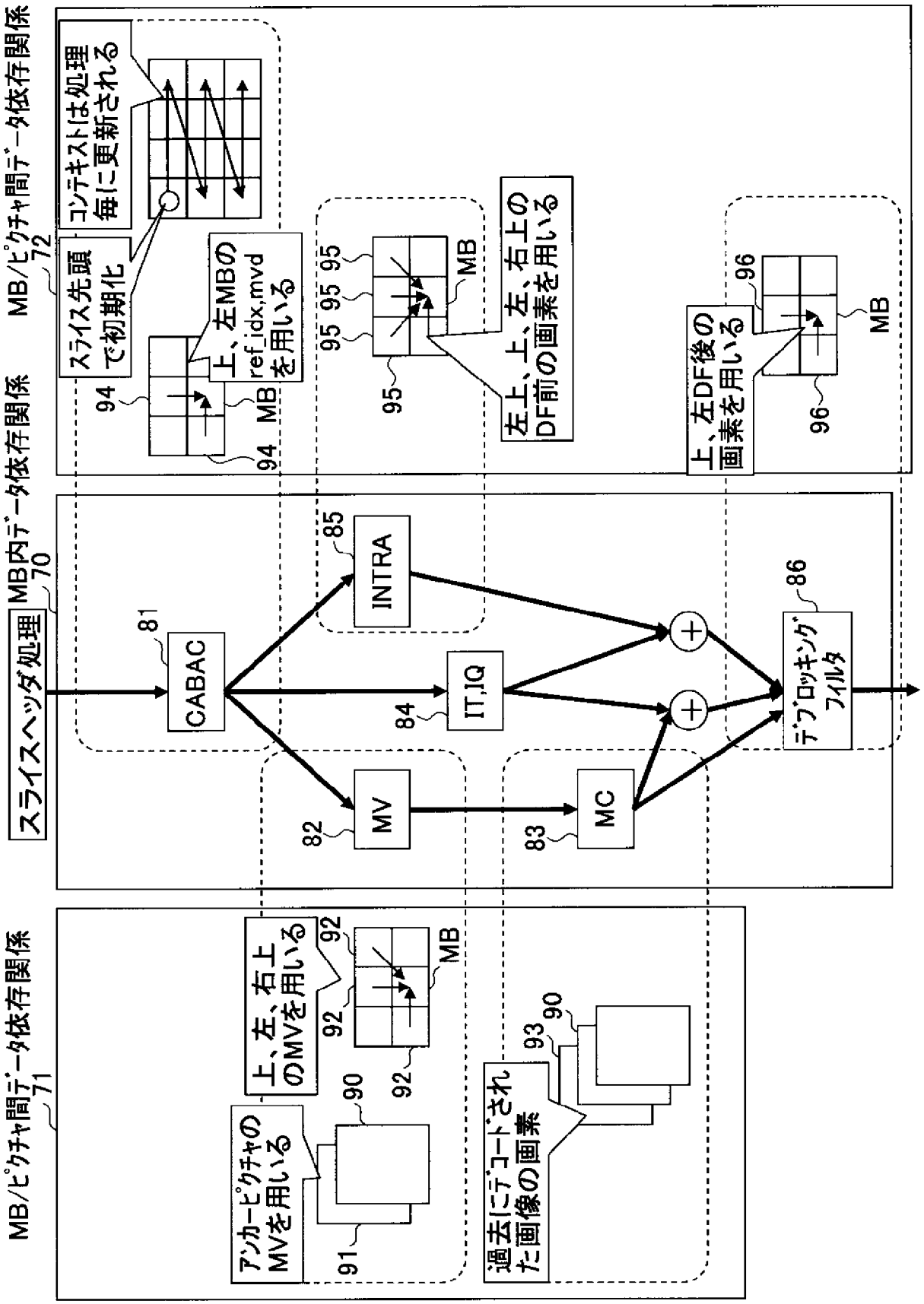
[図15]


更新
書込み
CABAC
チェック
MV,MC or INTRA
IT,IQ
チェック
DF
+
PE0
PE1
PE2
PE3
PE4

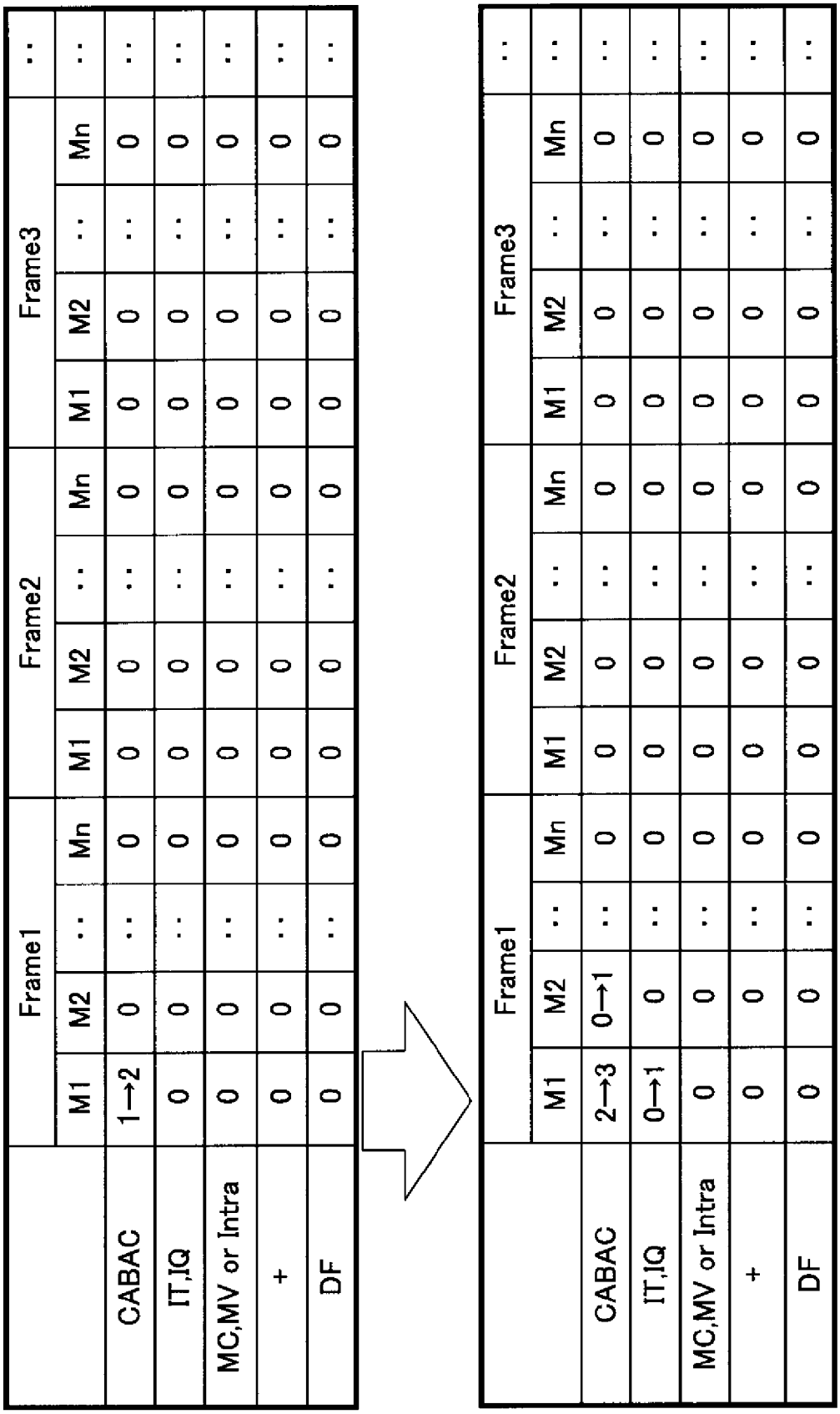
[図16]


CABAC
CABAC
CABAC
CABAC
IT,IQ
IT,IQ
IT,IQ
IT,IQ
MV,MC or INTRA
MV,MC or INTRA
MV,MC or INTRA
+
+
+
DF
DF
PE0
PE1
PE2
PE3
PE4

[図17]


MB/ピクチャ間データ依存関係
71
アンカーピクチャのMVを用いる
上、左、右上のMVを用いる
過去にデコードされた画像の画素
MB/ピクチャ間データ依存関係
72
スライス先頭で初期化
コンテキストは処理毎に更新される
上、左MBのref_idx,mvdを用いる
左上、上、左、右上のDF前の画素を用いる
上、左DF後の画素を用いる
MB内データ依存関係
70
スライスヘッダ処理
CABAC
INTRA
IT,IQ
MV
MC
デブロッキングフィルタ
MB
90
91
92
93
94
95
96

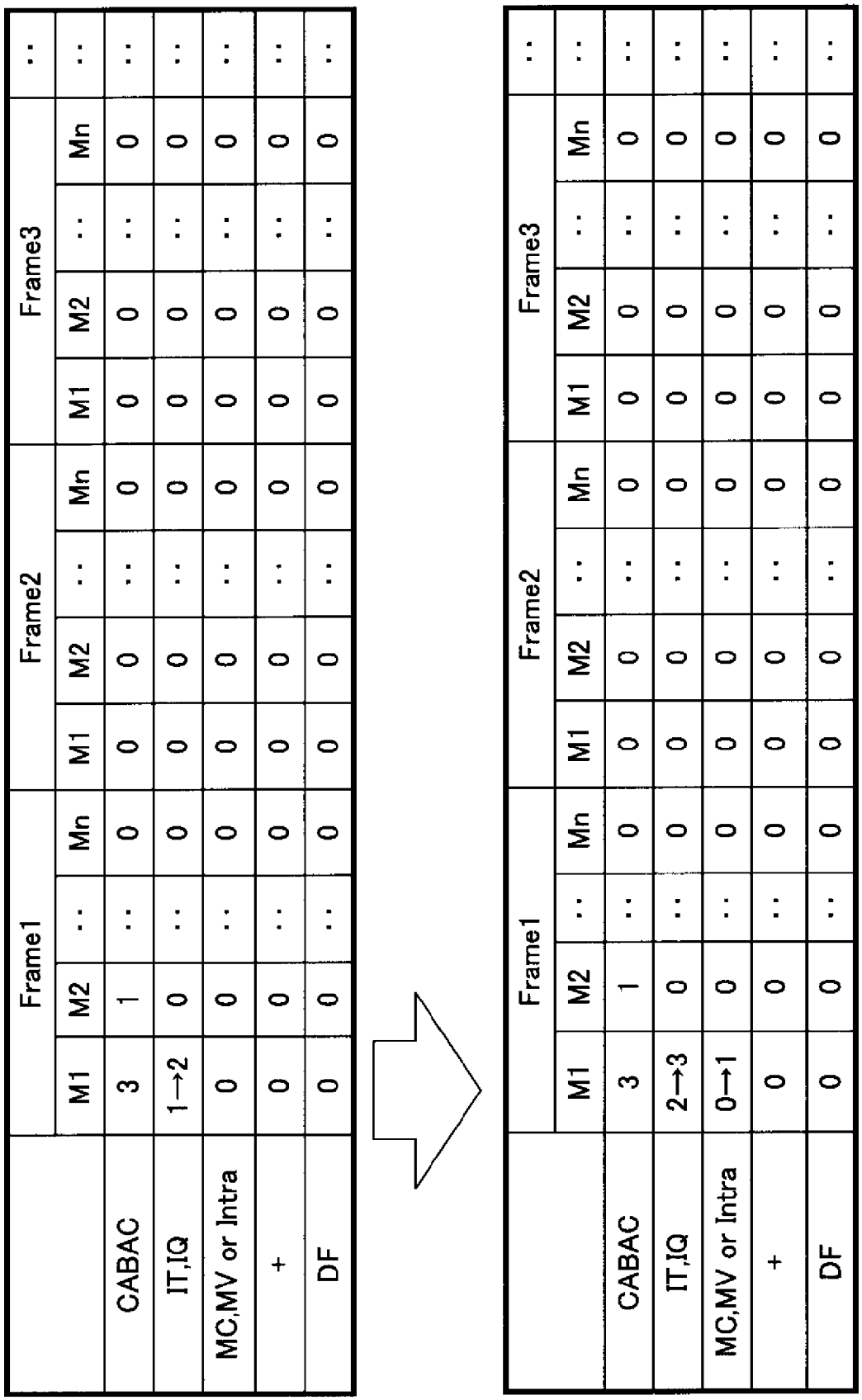
[図18]



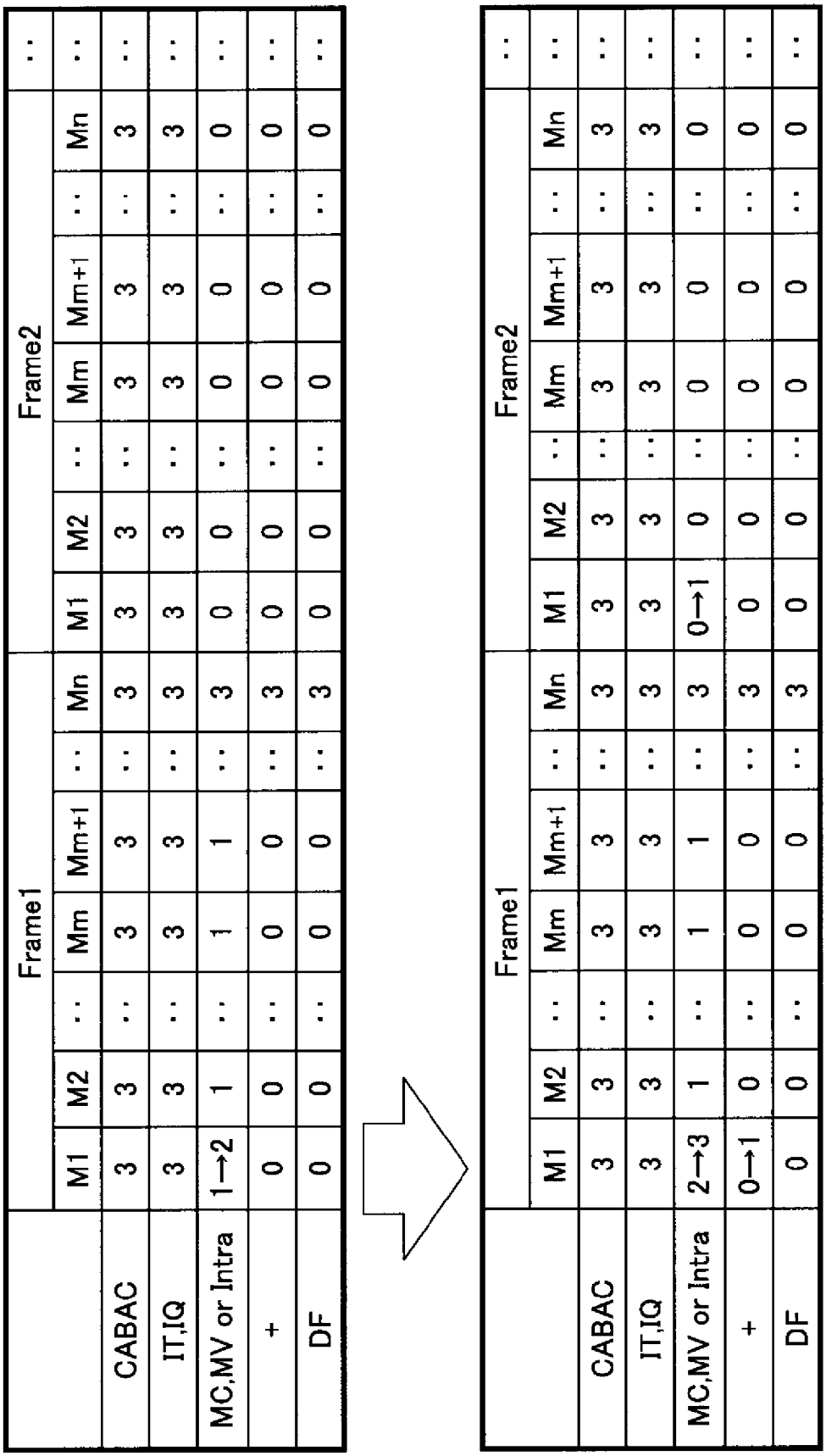
	Frame1				Frame2				Frame3			
	M1	M2	‥	Mn	M1	M2	‥	Mn	M1	M2	‥	Mn
CABAC	1→2	0	‥	0	0	0	‥	0	0	0	‥	0
IT,IQ	0	0	‥	0	0	0	‥	0	0	0	‥	0
MC,MV or Intra	0	0	‥	0	0	0	‥	0	0	0	‥	0
+	0	0	‥	0	0	0	‥	0	0	0	‥	0
DF	0	0	‥	0	0	0	‥	0	0	0	‥	0

	Frame1				Frame2				Frame3			
	M1	M2	‥	Mn	M1	M2	‥	Mn	M1	M2	‥	Mn
CABAC	2→3	0→1	‥	0	0	0	‥	0	0	0	‥	0
IT,IQ	0→1	0	‥	0	0	0	‥	0	0	0	‥	0
MC,MV or Intra	0	0	‥	0	0	0	‥	0	0	0	‥	0
+	0	0	‥	0	0	0	‥	0	0	0	‥	0
DF	0	0	‥	0	0	0	‥	0	0	0	‥	0

[図19]


Frame1
Frame2
Frame3
M1
M2
Mn
CABAC
IT,IQ
MC,MV or Intra
+
DF
3
1→2
0
2→3
0→1

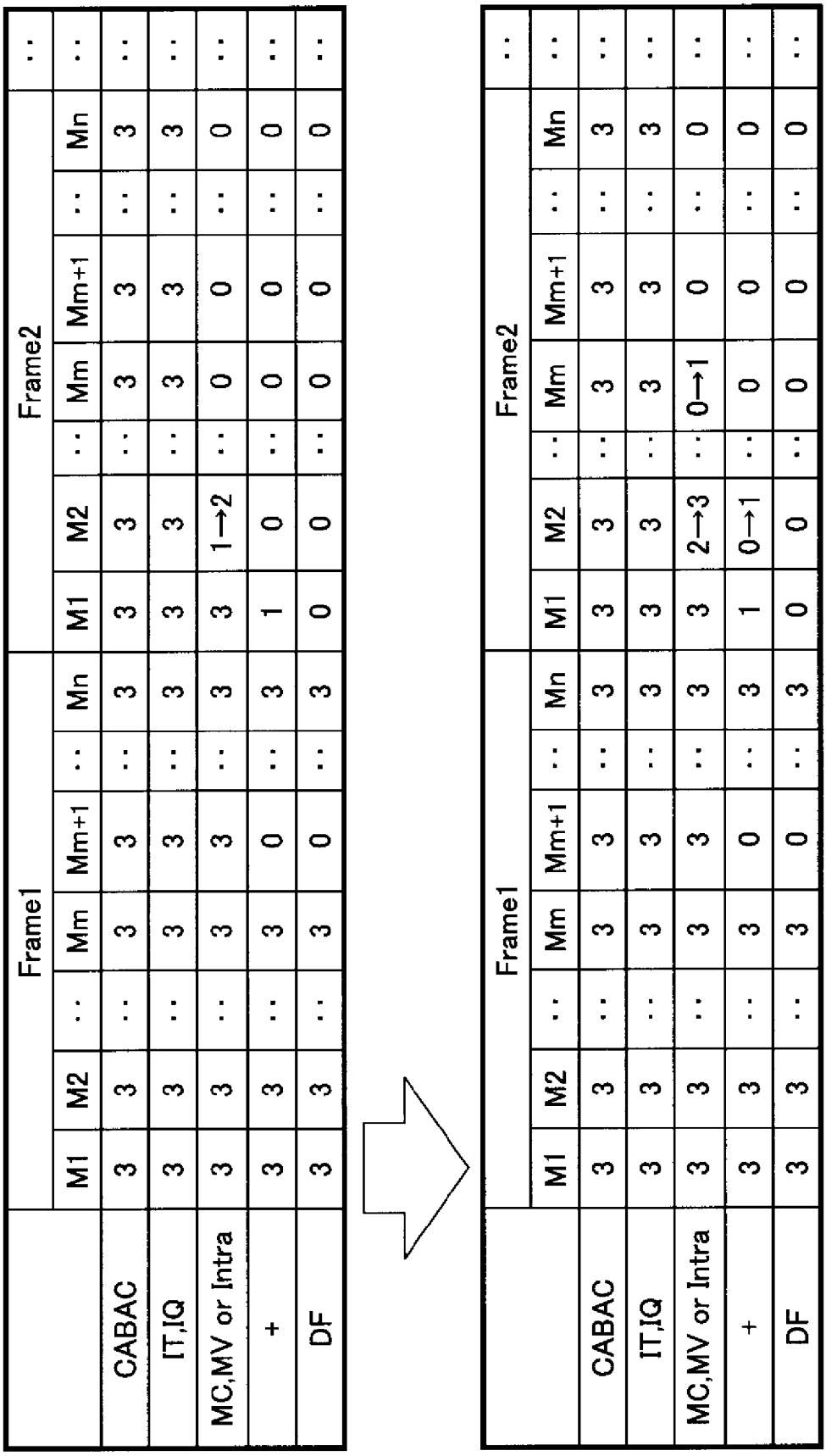
[図20]



	Frame1							Frame2							
	M1	M2	..	Mm	Mm+1	..	Mn	M1	M2	..	Mm	Mm+1	..	Mn	..
CABAC	3	3	..	3	3	..	3	3	3	..	3	3	..	3	..
IT,IQ	3	3	..	3	3	..	3	3	3	..	3	3	..	3	..
MC,MV or Intra	1→2	1	..	1	1	..	3	0	0	..	0	0	..	0	..
+	0	0	..	0	0	..	3	0	0	..	0	0	..	0	..
DF	0	0	..	0	0	..	3	0	0	..	0	0	..	0	..

	Frame1							Frame2							
	M1	M2	..	Mm	Mm+1	..	Mn	M1	M2	..	Mm	Mm+1	..	Mn	..
CABAC	3	3	..	3	3	..	3	3	3	..	3	3	..	3	..
IT,IQ	3	3	..	3	3	..	3	3	3	..	3	3	..	3	..
MC,MV or Intra	2→3	1	..	1	1	..	3	0→1	0	..	0	0	..	0	..
+	0→1	0	..	0	0	..	3	0	0	..	0	0	..	0	..
DF	0	0	..	0	0	..	3	0	0	..	0	0	..	0	..

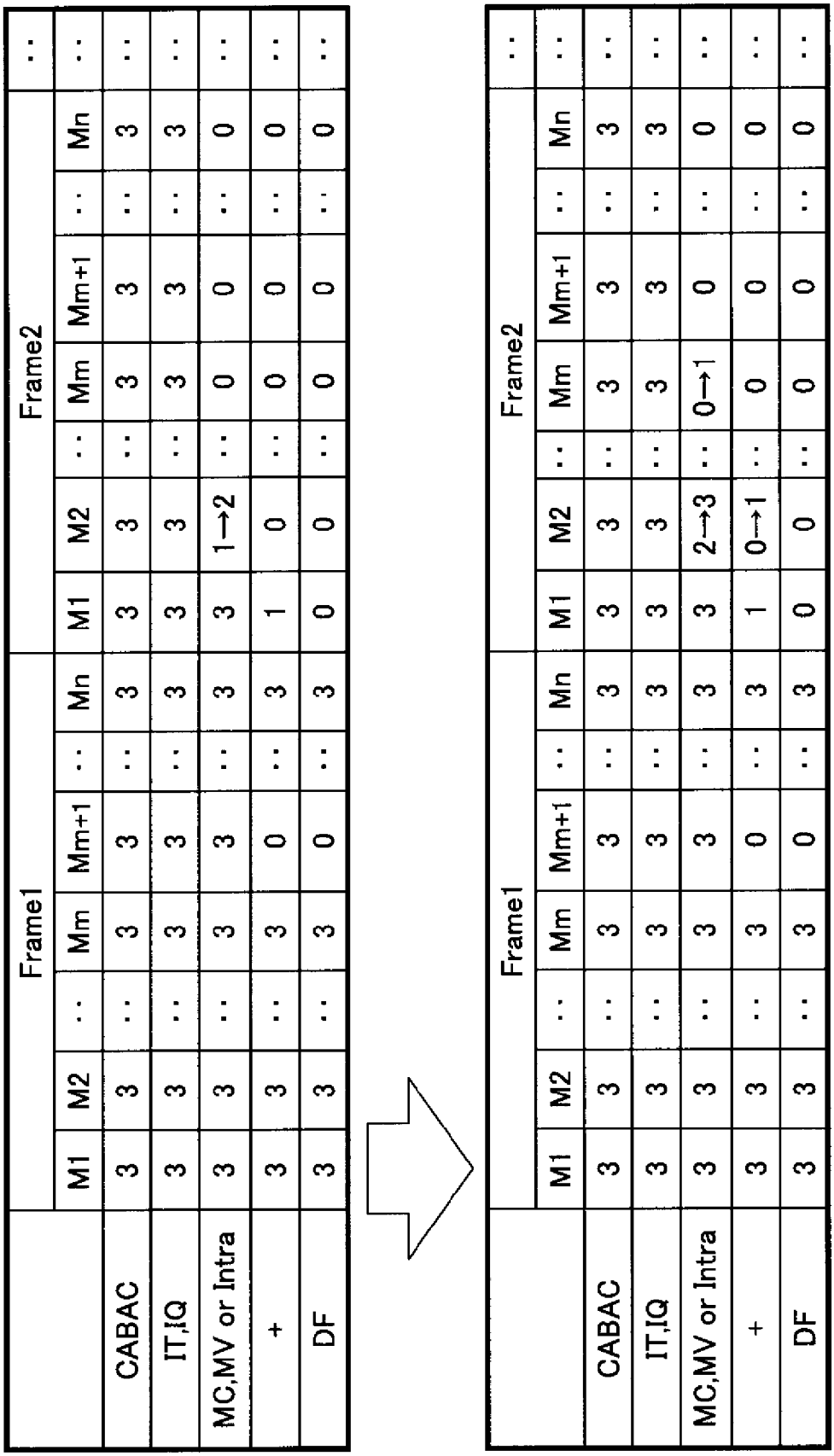
[図21]



	Frame1						Frame2						
	M1	M2	・・	Mm	Mm+1	・・	Mn	M1	M2	・・	Mm	Mm+1	・・	Mn	・・
CABAC	3	3	・・	3	3	・・	3	3	3	・・	3	3	・・	3	・・
IT,IQ	3	3	・・	3	3	・・	3	3	3	・・	3	3	・・	3	・・
MC,MV or Intra	3	3	・・	3	3	・・	3	3	1→2	・・	0	0	・・	0	・・
+	3	3	・・	3	0	・・	3	1	0	・・	0	0	・・	0	・・
DF	3	3	・・	3	0	・・	3	0	0	・・	0	0	・・	0	・・

	Frame1						Frame2						
	M1	M2	・・	Mm	Mm+1	・・	Mn	M1	M2	・・	Mm	Mm+1	・・	Mn	・・
CABAC	3	3	・・	3	3	・・	3	3	3	・・	3	3	・・	3	・・
IT,IQ	3	3	・・	3	3	・・	3	3	3	・・	3	3	・・	3	・・
MC,MV or Intra	3	3	・・	3	3	・・	3	3	2→3	・・	0→1	0	・・	0	・・
+	3	3	・・	3	0	・・	3	1	0→1	・・	0	0	・・	0	・・
DF	3	3	・・	3	0	・・	3	0	0	・・	0	0	・・	0	・・

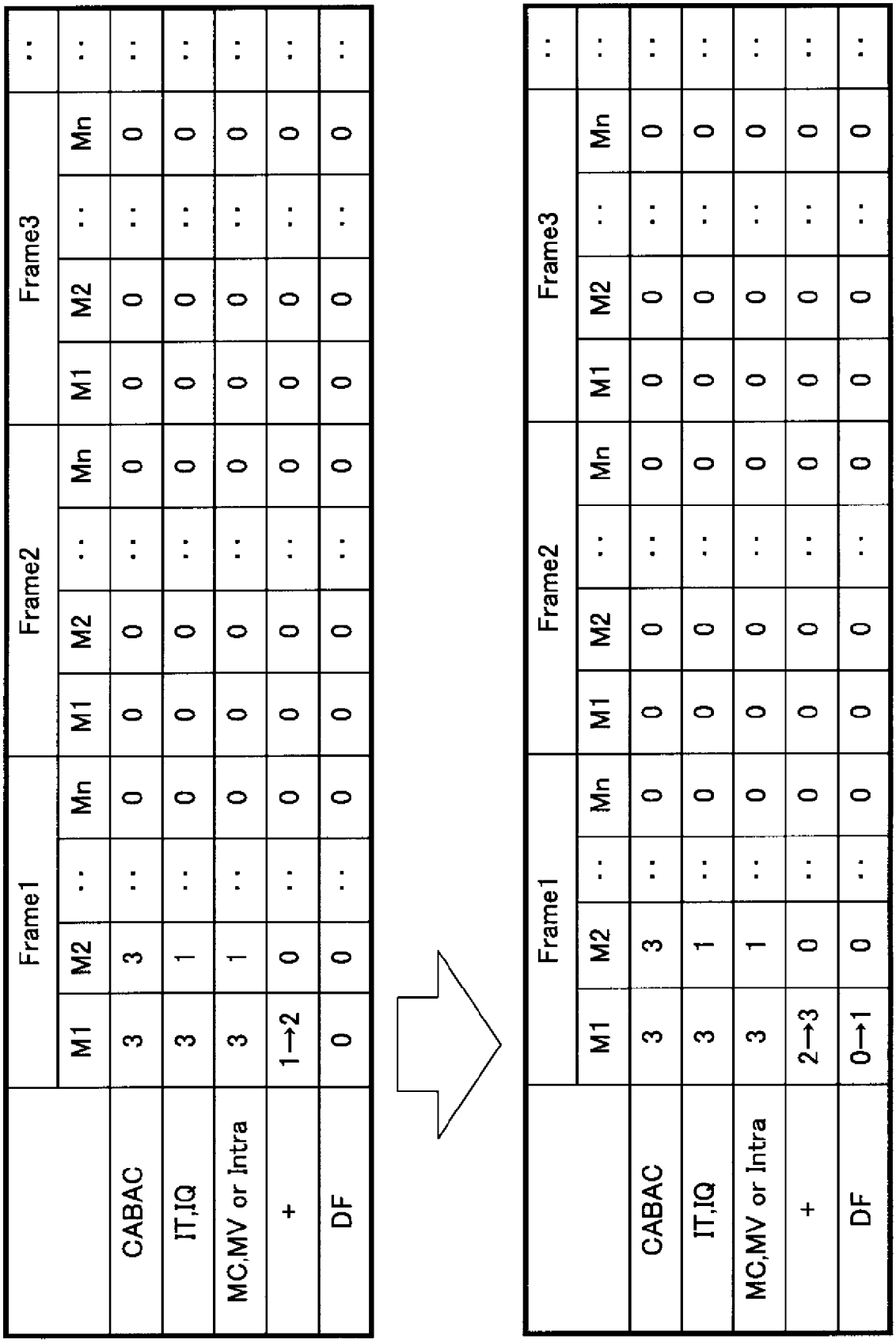
[図22]



	Frame1						Frame2						
	M1	M2	..	Mm	Mm+1	Mn	M1	M2	..	Mm	Mm+1	..	Mn
CABAC	3	3	..	3	3	3	3	3	..	3	3	..	3
IT,IQ	3	3	..	3	3	3	3	3	..	3	3	..	3
MC,MV or Intra	3	3	..	3	3	3	3	1→2	..	0	0	..	0
+	3	3	..	3	0	3	1	0	..	0	0	..	0
DF	3	3	..	3	0	3	0	0	..	0	0	..	0

	Frame1						Frame2						
	M1	M2	..	Mm	Mm+1	Mn	M1	M2	..	Mm	Mm+1	..	Mn
CABAC	3	3	..	3	3	3	3	3	..	3	3	..	3
IT,IQ	3	3	..	3	3	3	3	3	..	3	3	..	3
MC,MV or Intra	3	3	..	3	3	3	3	2→3	..	0→1	0	..	0
+	3	3	..	3	0	3	1	0→1	..	0	0	..	0
DF	3	3	..	3	0	3	0	0	..	0	0	..	0

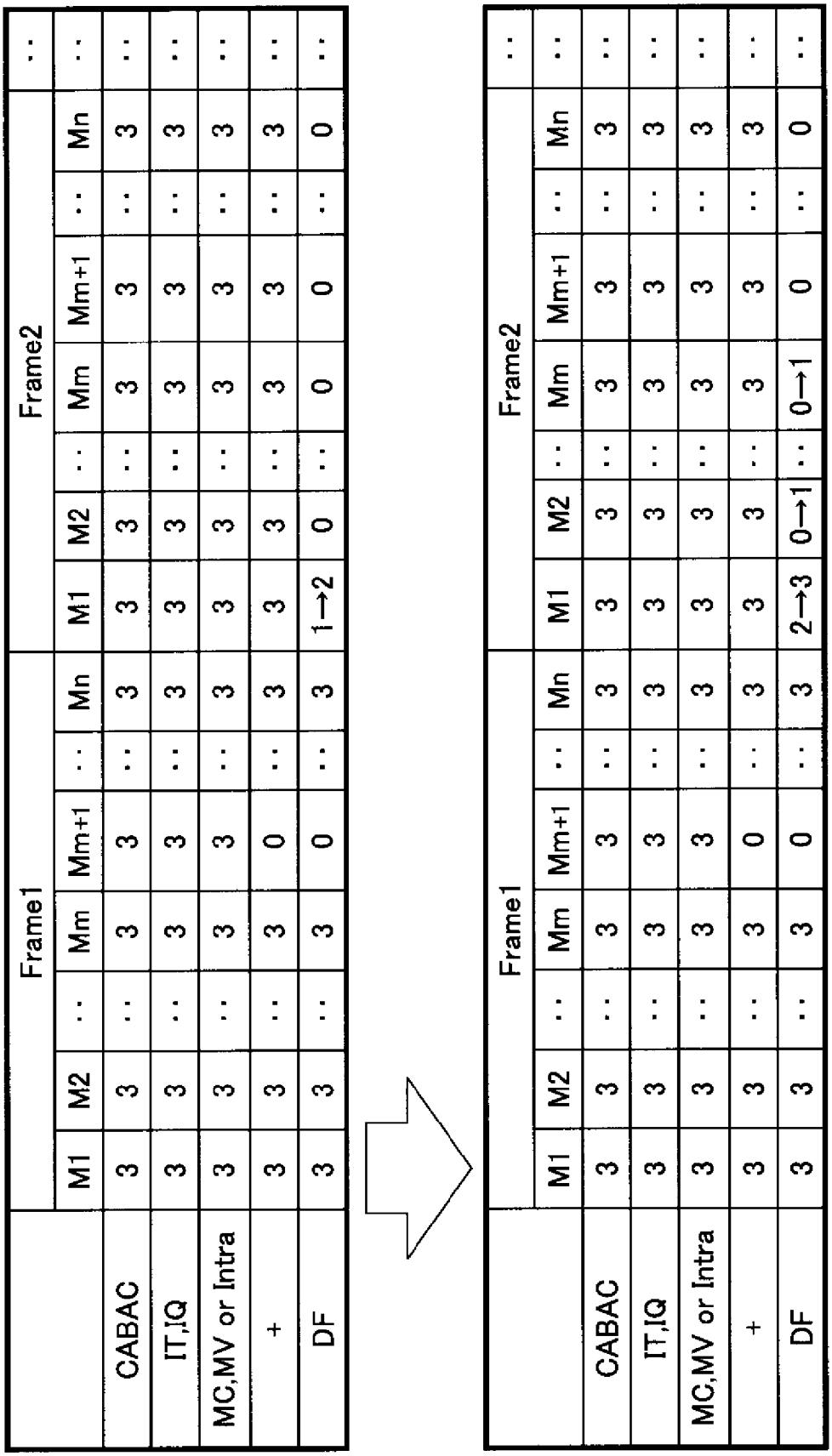
[図23]



	Frame1				Frame2				Frame3			
	M1	M2	‥	Mn	M1	M2	‥	Mn	M1	M2	‥	Mn
CABAC	3	3	‥	0	0	0	‥	0	0	0	‥	0
IT,IQ	3	1	‥	0	0	0	‥	0	0	0	‥	0
MC,MV or Intra	3	1	‥	0	0	0	‥	0	0	0	‥	0
+	1→2	0	‥	0	0	0	‥	0	0	0	‥	0
DF	0	0	‥	0	0	0	‥	0	0	0	‥	0

	Frame1				Frame2				Frame3			
	M1	M2	‥	Mn	M1	M2	‥	Mn	M1	M2	‥	Mn
CABAC	3	3	‥	0	0	0	‥	0	0	0	‥	0
IT,IQ	3	1	‥	0	0	0	‥	0	0	0	‥	0
MC,MV or Intra	3	1	‥	0	0	0	‥	0	0	0	‥	0
+	2→3	0	‥	0	0	0	‥	0	0	0	‥	0
DF	0→1	0	‥	0	0	0	‥	0	0	0	‥	0

[図24]


Frame1
Frame2
M1
M2
Mm
Mm+1
Mn
CABAC
IT,IQ
MC,MV or Intra
+
DF
1→2
2→3
0→1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2007/055439

A. CLASSIFICATION OF SUBJECT MATTER
G06F9/50(2006.01)i, *H04N7/26*(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)
G06F9/50, H04N7/26

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 1996-2007
Kokai Jitsuyo Shinan Koho 1971-2007 Toroku Jitsuyo Shinan Koho 1994-2007

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 2006-129285 A (Sony Corp.), 18 May, 2006 (18.05.06), Par. Nos. [0057] to [0077], [0126] to [0132], [0151] to [0172] & US 2006/0093043 A1	1-10
X	JP 2006-129284 A (Sony Corp.), 18 May, 2006 (18.05.06), Par. Nos. [0061] to [0075], [0130] to [0136], [0160] & US 2006/0093042 A1	1-10
A	JP 2000-295616 A (Matsushita Electric Industrial Co., Ltd.), 20 October, 2000 (20.10.00), Full text; all drawings (Family: none)	1-10

☐ 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	Date of mailing of the international search report
28 May, 2007 (28.05.07)	05 June, 2007 (05.06.07)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００７／０５５４３９

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G06F9/50(2006.01)i, H04N7/26(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.　G06F9/50, H04N7/26

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	１９７１－２００７年
日本国実用新案登録公報	１９９６－２００７年
日本国登録実用新案公報	１９９４－２００７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	関連する請求の範囲の番号
X	JP 2006-129285 A（ソニー株式会社）2006.05.18, 段落【0057】－【0077】,【0126】－【0132】,【0151】－【0172】 & US 2006/0093043 A1	1 - 10
X	JP 2006-129284 A（ソニー株式会社）2006.05.18, 段落【0061】－【0075】,【0130】－【0136】,【0160】 & US 2006/0093042 A1	1 - 10

☑　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	国際調査報告の発送日
２８．０５．２００７	０５．０６．２００７

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
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）　５Ｂ　９６４６
殿川　雅也
電話番号　０３－３５８１－１１０１　内線　３５４４

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	関連する請求の範囲の番号
A	JP 2000-295616 A（松下電器産業株式会社）2000.10.20, 全文，全図（ファミリーなし）	1 - 10