

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4219647号
(P4219647)

(45) 発行日 平成21年2月4日(2009.2.4)

(24) 登録日 平成20年11月21日(2008.11.21)

(51) Int.Cl.

F 1

G06F 12/12 (2006.01)
G06F 12/08 (2006.01)G06F 12/12 503
G06F 12/08 565
G06F 12/08 511E
G06F 12/08 543B

請求項の数 10 (全 10 頁)

(21) 出願番号 特願2002-295543 (P2002-295543)
 (22) 出願日 平成14年10月9日 (2002.10.9)
 (65) 公開番号 特開2003-186743 (P2003-186743A)
 (43) 公開日 平成15年7月4日 (2003.7.4)
 審査請求日 平成17年10月7日 (2005.10.7)
 (31) 優先権主張番号 09/975764
 (32) 優先日 平成13年10月9日 (2001.10.9)
 (33) 優先権主張国 米国(US)

(73) 特許権者 500587067
 アギア システムズ インコーポレーテッド
 アメリカ合衆国, 18109 ペンシルヴァニア, アレンタウン, アメリカン・パークウェイ エヌイー 1110
 (74) 代理人 100064447
 弁理士 岡部 正夫
 (74) 代理人 100085176
 弁理士 加藤 伸晃
 (74) 代理人 100106703
 弁理士 産形 和央
 (74) 代理人 100096943
 弁理士 白井 伸一

最終頁に続く

(54) 【発明の名称】適応的キャッシュフレームのロックング及びアンロックングのための方法及び装置

(57) 【特許請求の範囲】

【請求項1】

主メモリからの情報を格納するための複数のキャッシュフレーム；及び所定の数の前記最近最も使用された、タスクと関連するフレームをロックングするための適応的フレームロックング機構を備えることを特徴とするキャッシュメモリ。

【請求項2】

さらに、前記n個の最近最も使用されたフレームの識別子を記録するためのメモリを備えることを特徴とする請求項1記載のキャッシュメモリ。

【請求項3】

前記識別子がフレームアドレス、または前記最近最も使用されたフレームと関連するフラグであることを特徴とする請求項2記載のキャッシュメモリ。 10

【請求項4】

前記適応的フレームロックング機構がセット内の全てのフレームを同時にロックしないことを特徴とする請求項1記載のキャッシュメモリ。

【請求項5】

前記所定の数の最近最も使用されたフレームとして平均で3n/2個の最近最もアクセスされたフレームが識別されることを特徴とする請求項1記載のキャッシュメモリ。

【請求項6】

前記適応的フレームロックング機構が前記キャッシュの各フレームに対して3つのラッチ(a、b及びlock)を備えることを特徴とする請求項1記載のキャッシュメモリ。 20

【請求項 7】

さらにあるタスクに対する性能の劣化の原因となるフレームを自動的にアンロックする適応的フレームアンロッキング機構を備えることを特徴とする請求項1記載のキャッシュメモリ。

【請求項 8】

キャッシュメモリ内のフレームをロッキングするための方法であって、
主メモリからの情報を前記キャッシュメモリのフレームに格納するステップ；
所定の数の最近最も使用されたフレームを監視するステップ；及び
あるタスクが別のタスクによって割り込まれたとき、前記所定の数の最近最も使用されたフレームをロッキングするステップを含むことを特徴とする方法。 10

【請求項 9】

前記監視ステップが前記最近最も使用されたフレームのフレームアドレス、または前記最近最も使用されたフレームと関連するフラグを維持することを特徴とする請求項8記載の方法。

【請求項 10】

主メモリからの情報を格納するための複数のキャッシュフレームを持つキャッシュメモリ；及び

所定の数の前記最近最も使用された、タスクと関連するフレームをロッキングするための適応的フレームロッキング機構を備えることを特徴とする集積回路。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明は一般的にはキャッシュメモリデバイス、より詳細にはキャッシュメモリ内のフレームを適応的にロッキング及びアンロッキングするための方法及び装置に関する。

【0002】**【従来の技術】**

プロセッサは典型的にはシステムの性能の改善及びコストの削減のためにキャッシュを用いる。キャッシュは、最近アクセスされた情報（インストラクション或いはデータ）を、典型的にはより大きな主メモリよりも小さく、速くアクセスすることができるメモリ内に一時的に格納する。キャッシュは、一度アクセスされたことのあるブロックは近い将来再びアクセスされる可能性が高く、しばしば最近アクセスされたブロックの付近に存在するため有効である。このため、タスクの実行に合わせて、タスクのワーキングセット（そのタスクに対して現在要求されているインストラクション及びデータ）がその情報が再びアクセスされたときに備えキャッシュ内に格納される。キャッシュは典型的には主メモリからの複数の情報のブロックをキャッシュ内の、通常「セット（set）」とよばれる一箇所にマッピングする。「ブロック（block）」とはキャッシュ内に存在することができる最小単位の情報を表し、「フレーム（frame）」とはキャッシュ内の单一のブロックを格納することができる場所を表す。セット連想キャッシュ（set associative cache）においては、複数のフレームがセットにグループ化される。例えば、二方向セット連想キャッシュ（two-way set associative cache）は、各セット内に2つのフレームを含む。 30

【0003】

多くの埋め込みアプリケーションにおいては、一次タスクは一つ或いは複数の二次タスクによって割り込まれる。こうして、割り込みに続いて、二次の割り込んだタスクが実行している最は、割り込んだタスクのワーキングセットがキャッシュ内に格納され、潜在的に割り込まれた一次タスクのワーキングセットは追い出され、このため一次タスクが実行を再開した際の性能が低減される。つまり、一次タスクが実行を再開した時点で、キャッシュから追い出されていたワーキングセットの部分を主メモリから得ることが必要となり、「キャッシュミス（cache miss）」の要因となる。このため、一次タスクの実行時間は、二次タスクをランするために要した時間に、以前キャッシュに格納されていた情報の追い出された部分を主メモリから得るために発生したミスペナルティ（miss penalty）を加え 40

た時間となる。

【0004】

埋め込みプロセッサに用いることができるキャッシュの性能を改善するための様々な技法が提唱或いは示唆されている。例えば、キャッシュフレームロックングは、指定されたキャッシュフレームの内容が追い出されることを阻止するための技法である。この技法においては、ある指定された一群のフレームが「ロック（locked）」され、これらが含むブロックはこれらがアンロックされるまで追い出されることはない。ロックされたフレーム内のブロックは任意のタスクによって読み出したり、書き込んだりできる。フレームの無効化（invalidation）によってフレームはアンロックされる。ロックングとアンロックングは、特別なインストラクションの実行によって或いは一群のフレーム或いは個々のフレームを制御するメモリマップド制御レジスタ（memory-mapped control register）への書き込みによって実行される。実行中のタスクの幾つかのフレームを別のタスクの実行の前にロックングすることで、これらフレーム内のブロックが追い出されないことが保証される。こうすることで、割り込まれたタスクが実行を再開した際の性能が改善される。10

【0005】

【発明が解決しようとする課題】

従来のキャッシュフレームロックング技法は埋め込みプロセッサシステムにおけるキャッシュ性能を大幅に向上させるが、これらは幾つかの制約を持ち、克服することができれば、キャッシュメモリの性能をさらに改善することができる。より具体的には、従来のキャッシュフレームロックング技法は、あるタスクによって使用されるフレームが事前に知られていない限り役に立たず、さもなければ、不要なフレームがロックされたり、タスクの効率的な実行に不可欠なフレームがロックされなかつたりする。加えて、不要なフレームのロックングは、割り込んだタスクによって利用可能なキャッシュスペースを低減させ、割り込んだタスクの性能を不必要に低減させる。ただし、あるタスクによって使用されるフレームの知識を得ることは、あるタスクが実際にどのフレームにアクセスするかは、時間とともに広範に、しかも、しばしば予測不能に変化するために、困難である。あるタスクのフレームの使用がある時間に渡って知られている場合でも、割り込んだタスクが、しばしば行われるように、その期間の途中で実行を開始したときはこの知識は価値のないものとなる。20

【0006】

従って、任意の与えられた時間においてロックすべき適当なフレームを自動的に識別するキャッシュフレームロックング技法に対する必要性が存在する。より具体的には、あるタスクによって近い将来アクセスされる可能性のあるフレームを識別するキャッシュフレームロックング機構に対する必要性が存在する。さらに、ある特定のタスクによるフレームの使用に適応するキャッシュフレームロックング技法に対する必要性も存在する。加えて、ある実行中のタスクに対して著しい劣化の要因となるフレームを自動的にアンロックする機構に対する必要性も存在する。30

【0007】

【課題を解決するための手段】

一般的には、キャッシュメモリ内の最近最もアクセスされたフレームをロックングするための方法及び装置が開示される。キャッシュメモリ内のこれら最近最もアクセスされたフレームは、近い将来タスクによって再びアクセスされる可能性がある。このため、これら最近最も使用されたフレームが、キャッシュの性能を改善するために、本発明に従って、タスクのスイッチ或いは割り込みの開始において、特別なインストラクション或いは自動手段を用いて、ロックされる。最近最も使用されたフレームのリストがタスクの実行とともに更新される。この最近最も使用されたフレームのリストは、例えば、フレームアドレスのリストとして或いは各フレームと関連するフラッグとして実現される。この最近最も使用されたフレームのリストは、複数のタスクが互いに割り込み合う場合は、各タスクに対して別個に維持することもできる。40

【0008】

あるタスクに対して著しい性能の劣化の要因となり得るフレームを自動的にアンロッキングする適応的フレームアンロッキング機構も開示される。この適応的フレームアンロッキング機構は、あるタスクがフレームミスに遭遇した回数を監視し、フレームミスの回数が所定の閾値を超えたときはある与えられたフレームをアンロックする。

本発明並びに本発明のさらなる特徴及び長所のより完全な理解が以下の詳細な図面及び図面から得られるものである。

【0009】

【発明の実施の形態】

図1は本発明による適応的フレームロッキング機構 (adaptive frame locking mechanism) 100を示す。図1に示すように、適応的フレームロッキング機構100はキャッシュ150のフレーム、例えば、セット0内のフレーム1、セット1とn-1内のフレーム2、及びセットN内のフレームNを選択的にロックする。本発明の一面によると、任意の特定の時間において、あるタスクによって近い将来アクセスされる可能性が高いフレームがロッキングのために識別される。いったんロックされると、フレーム内に含まれる情報は割り込んだタスクによって撤退させる（立ち退かせる）ことはできない。フレーム本発明のもう一面によると、フレームロッキング機構はある特定のタスクのフレームの使用に適応する。加えて、後に詳細に説明するように、本発明は、実行中のタスクに対して著しい性能の劣化の要因となり得るフレームを自動的にアンロックする。

10

【0010】

本発明は、実行の最中、あるタスクによって近い将来アクセスされる可能性のあるキャッシュフレームを、必要であれば任意の時間にこれら「貴重なフレーム (valuable frames)」をロックできるように絶えず識別する。本発明はあらゆるキャッシュ編成（データかインストラクションかに関係なく）に適用することができるが、以下では、本発明は各セットアドレスに2つのフレームを持つ二方向セット連想インストラクションキャッシュ (two-way set associative instruction cache) を用いて説明される。テーブルエントリは、ロックのアドレスによってアドレスされるが、そのロックがキャッシュ内に存在するか否かを識別し、その位置を周知のやり方にて与える。適応的フレームロッキング機構100及びキャッシュ150は、デジタル信号プロセッサ (DSP)、マイクロコントローラ、マイクロプロセッサ、特定用途向け集積回路 (ASIC)、或いは他の集積回路の一部から構成される。

20

【0011】

前述のように、キャッシュは、主として、現在アクセスされている情報が、最近アクセスされた可能性がある、或いはアドレス空間内においてそのような情報の付近に存在する可能性があるために有効である。本発明は、アプリケーションによってアクセスされた最後のn個のフレームのキャッシュ位置を記録する。ここでnは事前に定義するか、或いは特別なインストラクションによって指定される。こうすることで、これら最近最も使用されたn個のフレームを、タスクスイッチ或いは割込みの開始時において、本発明に従って容易に指定及びロックすることが可能となる。これら最近アクセスされたn個のフレームは、再びアクセスされる可能性が高く、中断されたタスクが実行を再開したとき貴重なものとなる可能性が高い。加えて、このフレームの集まりはタスクが実行に合わせて更新されるために、任意の時点においてあるタスクにとって貴重なフレームを含む可能性が高く、タスクの実行が中断された時点で最新なものとなる。

30

【0012】

例えば、nが16とされる場合は、キャッシュ内の最後にアクセスされた16個のフレームの位置が絶えず記録される。この情報を記録及びこれにアクセスするために様々な方法を用いることができる。図2はn個の最近最も使用されたフレームのアドレスを示す一例としての最近最も使用されたフレーム用のデータベース (most recently used frame database) 200のサンプルテーブルである。この最近最も使用されたフレーム用のテーブル200は、例えば、n個の要素を持つ先入れ先出し (FIFO) バッファとして実現される。こうして、最近最も使用されたフレーム用のテーブル200内の該当するフレームの位

40

50

置が絶えず更新される。タスクスイッチが発生すると、この最近最も使用されたフレーム用のテーブル 200 内に識別された全てのフレームが適応的フレームロッキング機構 100 にて、特別なインストラクション或いは用意されている場合は自動手段を用いてロックされる。この最近最も使用されたフレーム用のテーブル 200 は、複数のタスクが互いに他に割り込む場合は、複数の列（カラム）を設け、各タスクに一つの列（カラム）を割り当てるこどもできる。

【0013】

代替として、「リースントリイアクセスドフレーム (recently accessed frame, (raf))」ビットをキャッシュ内の各フレームに割り当て、この raf ビットを、例えは、それと関連するフレームが最後にアクセスされた 16 個のフレームの一つである場合は 1 なる二進値にセットし、それと関連するフレームが最後にアクセスされた 16 個のフレームの一つでない場合は 0 なる二進値にセットすることもできる。これらビットは、キャッシュ内に記録することも、別の所（例えは、プロセッサ内）に記録することもできる。タスクスイッチが発生時点で、「真（true）」のビットと関連する全てのフレームが特別なインストラクション、或いは用意されている場合は自動手段を用いてロックされる。複数のタスクが互いに割り込む場合は、複数のセットの raf ビットを維持することもでき、この場合は、各タスクに対して、一セットの raf ビットが維持される。

【0014】

一つのバリエーションにおいては、あるセット内の全てのフレームを同時にロックすることを避けることで、実行中のタスクに対する最小のキャッシュサポートが維持される。フレームのロッキングはブロックの追立て (block evictions) を阻止するのみで、アクセスは阻止せず、あるセット内の全てのフレームをロッキングすること自体がデッドロックをもたらすことはないことに注意する。あるタスクが読み出し動作の際に、全てのフレームがロックされているセットにアクセスし、このタスクがミス (miss) に遭遇した場合は、このタスクはメモリからデータを受信するが、このデータはキャッシュ内には格納されない。あるタスクがロックされているセットへの書き込みの際にミス (miss) に遭遇した場合は、プロセッサからの新たなデータはメモリから読み出されたデータと併合され、メモリに書き戻される。あるタスクがロックされているセットへの書き込みの際にヒット (hit) に遭遇した場合は、キャッシュはこの新たなデータにて更新される。

【0015】

書戻しキャッシュ (write back cache) は、それに書き込まれたが、まだ主メモリ内に書き込まれていないデータを保持する。キャッシュ内の主メモリ内のそのイメージより最近のブロックは「汚れている (dirty)」といわれる。この汚れたブロックは、それが追立てられたとき、或いはコヒーレントメモリスキームにおいてもう一つのプロセッサまたは別のプロセスがそのブロックにアクセスしたとき、主メモリに書き込まれる。別のプロセスがある汚れたブロックの所有権を得ようと試みた場合は、その汚れたブロックはメモリに書き戻され、そのフレームはキャッシュ内で無効にされる。別のプロセスがある汚れたブロック内のデータを読み出すことを試みた場合は、その汚れたブロックはメモリ内に書き戻され、こうして「きれい (clean)」になるが、ただし、そのブロックを含むフレームは無効にされない。これら動作は、フレームがロックされているか否かに關係なく行われる。従って、ロックされているフレーム内に汚れたブロックが存在する場合でも、書戻しデータキャッシュ (write back data cache) 内での特別な動作は必要とされない。フレームロッキングスキームを使用するキャッシュにおいて、フレームロッキングスキームを使用しないそれとの対比で、要求される唯一の追加の動作は、推測されるように、無効にされたロックされているフレームのアンロックし、これを任意のタスクに使用できるように解放することのみである。

【0016】

適応的フレームロッキング機構

図 3 は図 1 の適応的フレームロッキング機構 100 とともに用いるための最近最も使用されたフレーム用の回路 300 の一つの実施例を示す。最近最も使用されたフレーム用の回

10

20

30

40

50

路 3 0 0 は最近最もアクセスされたフレームを記録する。図 3 に示す特定の実施例においては、(平均して) $3n/2$ 個の最近最もアクセスされたフレームが記録される。ここで、 n は実行の際に設定される偶数であり、タスクによって異なる。記録される最近使用されたフレームの数は、時間によって n と $2n$ の間で異なり、平均して $3n/2$ 個となる。別のやり方として、ロックされるフレームの数は事前に決定し、一定とすることもできる。

【0017】

図 3 に示すように、この一例としての最近最も使用されたフレーム用の回路 3 0 0 は、各フレーム i と関連する(以降ラッチ「a」、「b」と及び「ロックド」と呼ばれる)3個の 1 ビットラッチ $310-i$ 、 $320-i$ 、 $330-i$ を備える。ラッチ「a」はそれと関連するフレームがアクセスされたときセットされる。 n 回のアクセスの度に、ラッチ「a」内のフレームの値がラッチ「b」に移され、ラッチ「a」はリセットされる。従って、最後の n 回のアクセスの間にフレームがアクセスされた場合、ラッチ「b」がセットされる。転送の後、ラッチ「a」は、そのフレームがアクセスされると再びセットされる。この動作がプログラムの実行を通じて継続される。

【0018】

割込み或いはタスクのスイッチが発生し、フレームロックキングが必要となった場合、特別のインストラクションを実行するか、或いは他の手段によって「ロック(lock)信号が生成される。ロック信号は、ロック信号が発動(asserted)されたときラッチ「a」とラッチ「b」のいずれかがセットされている場合、与えられたフレームのロックラッチ(lock latch)をセットし、そのフレームをロックする。従って、ロックされるフレームの数は、シフト信号とロック信号との間の時間関係に依存する。この効果が図 4 に示される。(全ての a ラッチをリセットする) シフトの直後には、最後にアクセスされた n 個のフレームの位置が記録され、シフトの直前には $2n$ 個が記録される。

【0019】

こうして、2 個のラッチ a と b を用いた場合のロックされるフレームの数は平均で $3n/2$ 個となるが、論理内に用いられるラッチの数を様々に変えることで、ロックされる平均での(或いは所定の)フレームの数を他の値に変えることができる。例えば、3 個のラッチを用いた場合は、ロックされるフレームの平均数は $5n/2$ となる。加えて、ロック信号とシフト信号が同期される場合、つまり、これらが一定の時間関係を持つ場合は、ロックされるフレームの数は平均ではなく一定となる。ロック信号がシフトの直前に発動された場合は、常に $2n$ 個の最近最もアクセスされたフレームがロックされる。同様に、ロック信号が常にシフトの直後に発動される場合は、 n 個の最近最もアクセスされたフレームがロックされる。こうして、ロックされるフレームの数を時間に渡っての平均ではなく、所定の固定された値とすることもできる。

【0020】

適応的無効化機構

あるタスクが(例えば、一つ或いは複数の前に実行されていたタスクによって)ロックされたフレームを含むキャッシュを持つシステム内で実行している場合は、これらフレームは情報の格納のためにそのタスクによって利用できない。タスクの性質及びロックされたフレームの数及び位置によっては、このようなタスクは許容できないほどのミス率(miss rates)に遭遇し、このため性能が劣化することがある。本質的に、フレームのロックのために、タスクに対してキャッシュが小さくなりすぎる。このような事態が発生することを回避するために、以下に図 5 との関連で説明する適応的アンロックキング機構 5 0 0 が、必要に応じて $r a f$ ビットが無効にするために用いられる。

【0021】

図 5 に示すように、適応的アンロックキング機構 5 0 0 は各フレームと関連して(例えば、2 ビットの)「フレームミスピット(frame miss bits)」を維持する。これらビットはフレームがロックされているときは零にセットされる。ロックされたフレームを持つセットの所でキャッシュミス(cache miss)が発生すると、下位のミスピットが 1 にセットさ

10

20

30

40

50

れ、そのセットの所で次のミスが発生すると、上位のミスピットが1にセットされる。次のミス（両方のミスピットがセットされている状態でのミス）が発生すると、そのフレームはアンロックされる。こうして、この一例としての機構500は、（2つのミスピットが用いられている場合）そのフレームが属するセット上で3つのミスが発生すると、ロックされているフレームをアンロックする。こうすることで、タスクが重大な性能の劣化を引き起こしているフレームをアンロックすることが可能になる。

【0022】

図5に示す一例としての機構500は、指定される条件のもとで（例えば、所定の回数のキャッシュアクセスの後に、或いはロックされたフレームを含むセット内で所定の数のビットが発生した後に）、これらミスピットをリセットすることで改善することができる。
10 このようなスキームにおいては、アンロッキング基準を、総数ではなく、例えば、中断無しに発生したミスの数とすることもできる。使用されるビット数を増減することもでき、その場合でも、この機構は、この数とは無関係に、本質的に同一にとどまる。適応的無効化機構500は、ロックされたフレームを持つキャッシュ内で動作するタスクの性能を著しく制約するフレームを適応的かつ漸進的にアンロッキングすることを可能にする。

【0023】

セット連想キャッシュは、通常、新たなブロックがあるセット内に書き込むことが必要になったとき、最近最も使用されてない（least recently used）ブロックを追い出す。この目的のために、各セットに対してリーストリースントリイユースド（least recently used, lru）ビットが割り当てられる。これらリーストリースントリイユースド（least recently used, lru）ビットは、セット内の、そのセット内の任意の他のブロックの最後のアクセスの前にそれに対する最後のアクセスが発生した情報のブロックを保持するフレームを指定するように管理される。二方向セット連想キャッシュは、例えば、セット内に2つのフレームを持ち、セットごとに、次に追立てられるべきブロックを含むフレームを指定する一つのlruビットを維持する。例えば、二進の零（0）によってフレーム-0が最近最も使用されなかつたことが指定され、二進の1によってフレーム-1が最近最も使用されなかつたことが指定される。
20

【0024】

ロックされるべきフレームの数が常にキャッシュ内のフレームの総数の半分であり、同一セット内の2つのフレームが同時にはロックされない実現においては、極めて単純な適応的フレームロッキング機構100を実現することができる。この場合は、現存のリーストリースントリイユースド（lru）ビットをキャッシュ内の最近最も使用されたフレームをロックするために用いることができる。lruビットはセット内の最近最も使用されたフレームを指定するために、（二方向セット連想キャッシュにおける）そのインバース（逆数）は、セット内の最近最も使用されたフレームを指定する。一体となってこれらビットは、二方向セット連想キャッシュ内の最近最も使用されたフレームを指定する。従って、割込み或いはタスクスイッチが発生したとき唯一必要とされるのは、各セット内のそのセットのlruビットのインバースによって指定されるフレームをロックすることのみである。
30

【0025】

上で説明の実施例及びバリエーションは単に本発明の原理を解説するためのものであり、当業者においては、本発明の精神及び範囲から逸脱することなく、様々な修正ができる。
40

【図面の簡単な説明】

【図1】本発明による適応的フレームロッキング機構を示す図である。

【図2】n個の最近最も使用されたフレームのアドレスを示す一例としての最近最も使用されたフレーム用のデータベースからのサンプルテーブルを示す図である。

【図3】図1の適応的フレームロッキング機構とともに用いるための最近最も使用されたフレーム用の回路の一つの実施例を示す図である。

【図4】図3の最近最も使用されたフレーム用の回路内のラッチリセットの相対タイミン
50

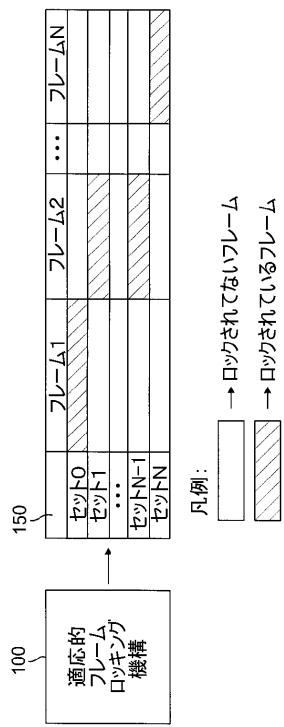
グを示す図である。

【図5】本発明による適応的フレームアンロッキング機構を示す図である。

【符号の説明】

- 100 適応的フレームロッキング機構
- 150 キャッシュ
- 300 最近最も使用されたフレーム用の回路
- 500 適応的アンロッキング機構

【図1】

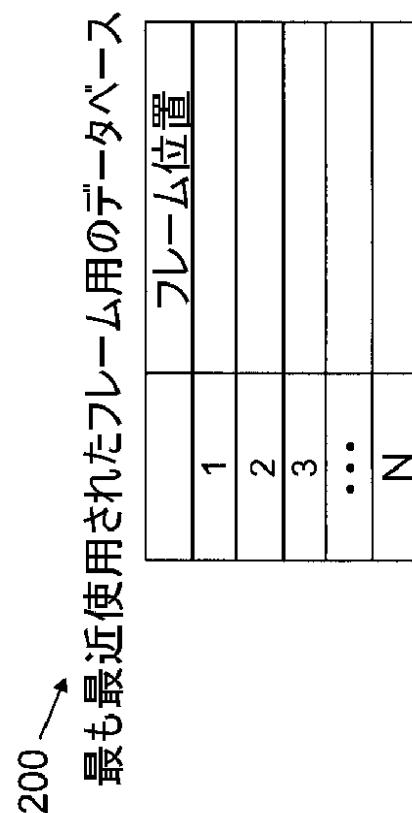


凡例:

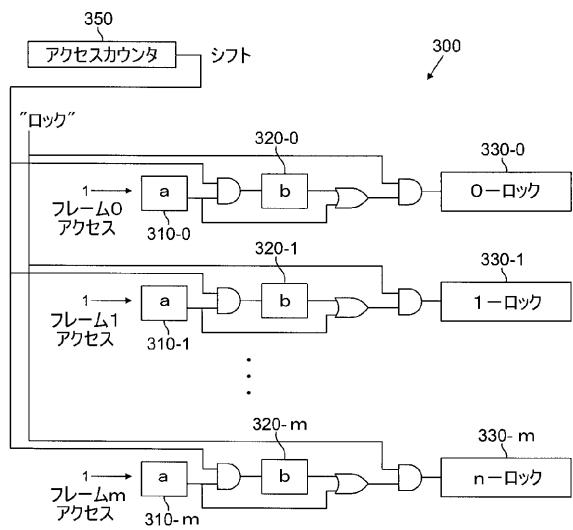
→ ロックされてないフレーム

→ ロックされているフレーム

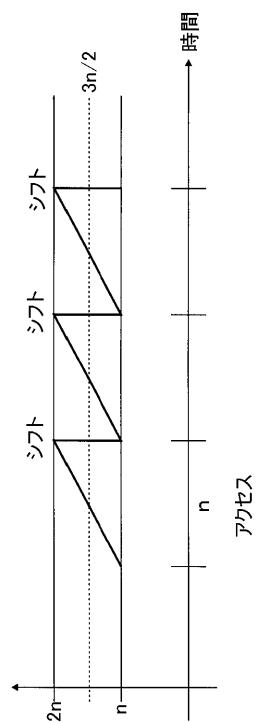
【図2】



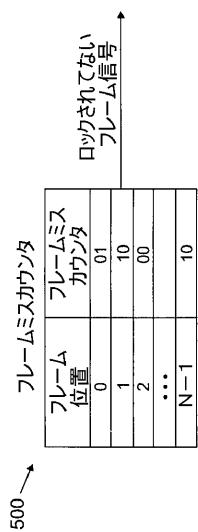
【図3】



【図4】



【図5】



フロントページの続き

(74)代理人 100091889
弁理士 藤野 育男
(74)代理人 100101498
弁理士 越智 隆夫
(74)代理人 100096688
弁理士 本宮 照久
(74)代理人 100102808
弁理士 高梨 憲通
(74)代理人 100104352
弁理士 朝日 伸光
(74)代理人 100107401
弁理士 高橋 誠一郎
(74)代理人 100106183
弁理士 吉澤 弘司
(72)発明者 ハリー デワイヤー
アメリカ合衆国 08801 ニューヨーク, アナンデール, ブリストル コート 4
(72)発明者 ジョン スザンザ ファーナンド
アメリカ合衆国 18036 ペンシルヴァニア, クーパースバーグ, リバティ ロード 457
9

審査官 清木 泰

(56)参考文献 特開平02-066654 (JP, A)
特開平07-334428 (JP, A)
特開平06-019797 (JP, A)
特開昭57-015274 (JP, A)
特開2000-339222 (JP, A)
特表2001-507846 (JP, A)
特開2000-148590 (JP, A)
特開平04-003241 (JP, A)
特開平03-232036 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F12/08-12/12