

# 第4章 記憶装置の構成

## 4.1 記憶階層方式

### 4.1.1 記憶装置への要求事項

(1)速度: アクセスタイムとサイクルタイム

(2)容量

(3)不揮発性 (Non-Volatile) : 電源切っても記憶は残る

(4)書換え可能性

(5)ランダムアクセス性

(6)可搬性

## 4.1.2 各種の記憶デバイスの速度と容量

### (1) 半導体メモリ

SRAMとDRAM

S:Static

電気を入れておけば安定に情報を記憶

D:Dynamic

電気を入れておいても、ときどき読み出さないと記憶がなくなる リフレッシュが必要

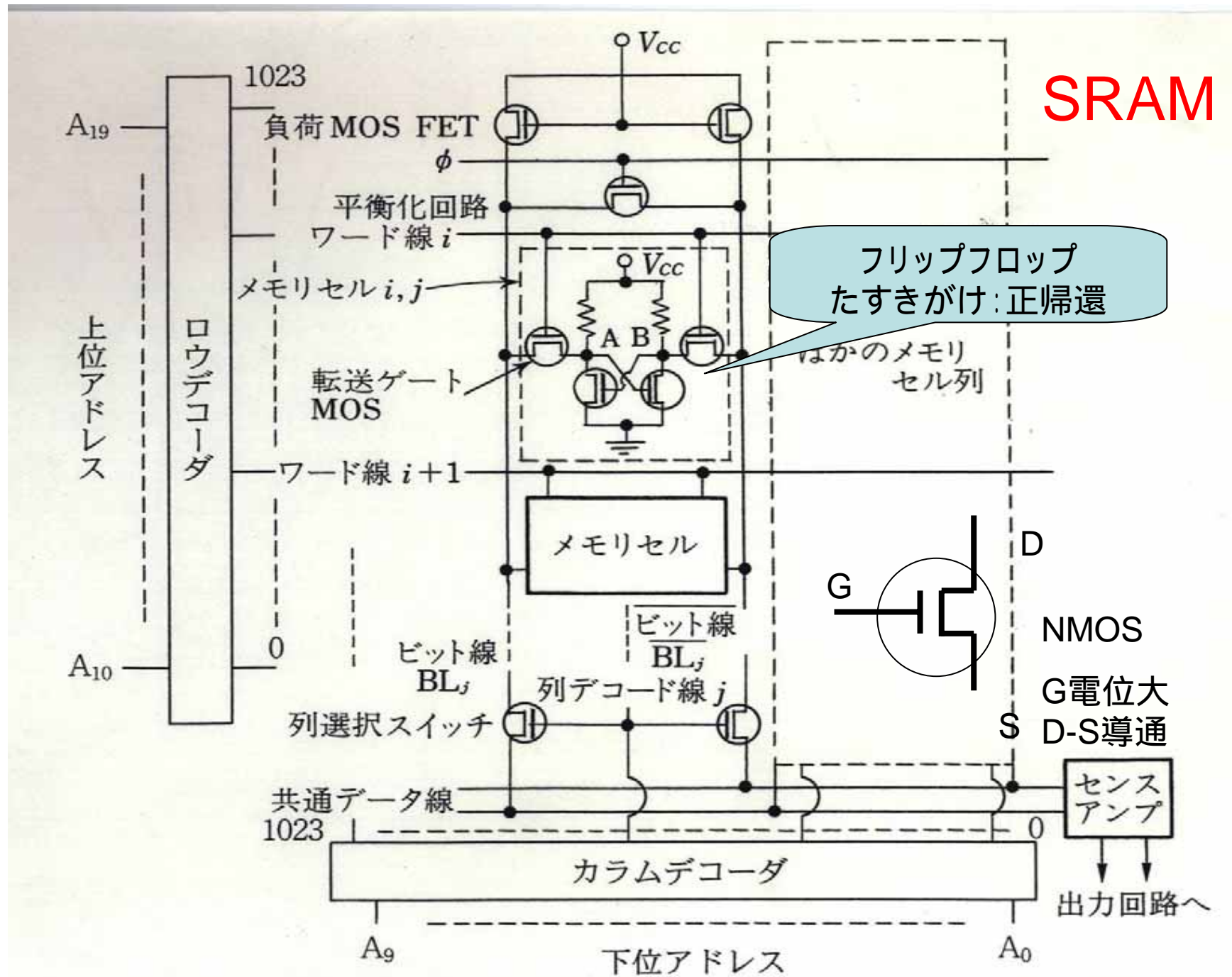


図 4.1 SRAM の構成 [電子情報通信ハンドブック, オーム社 (1988)]

# DRAM

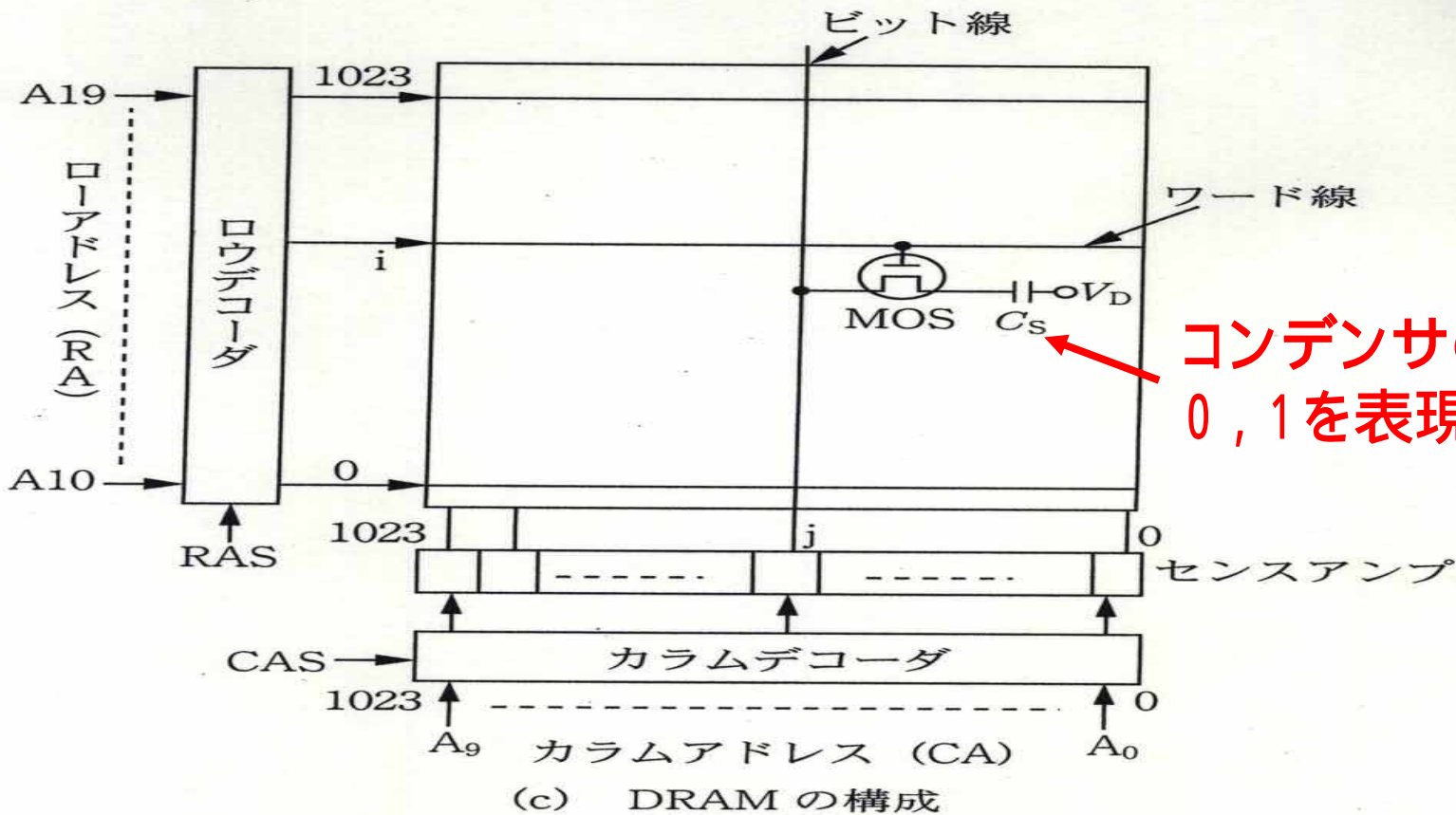
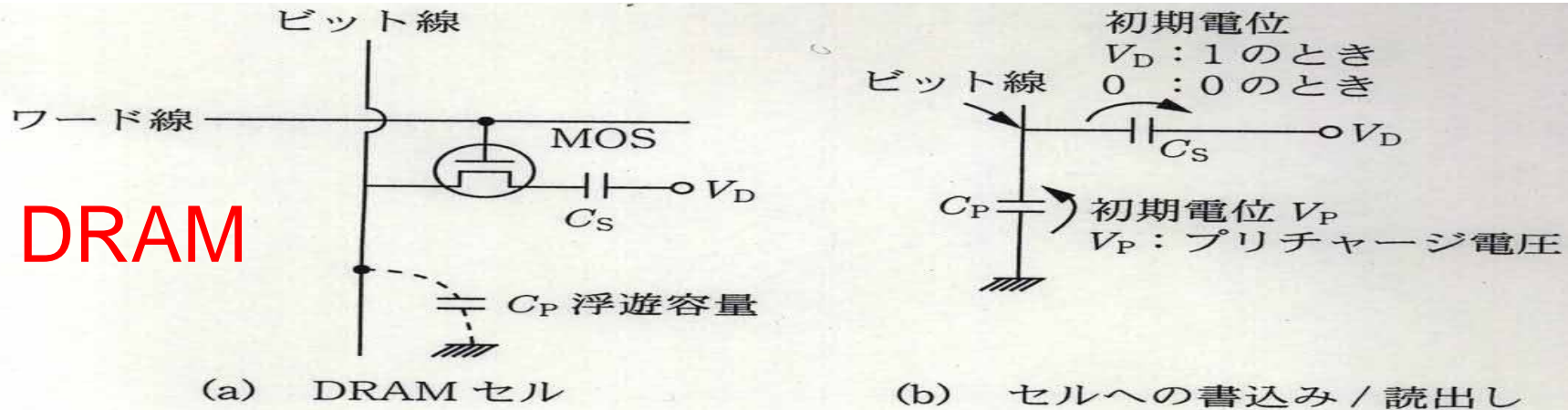
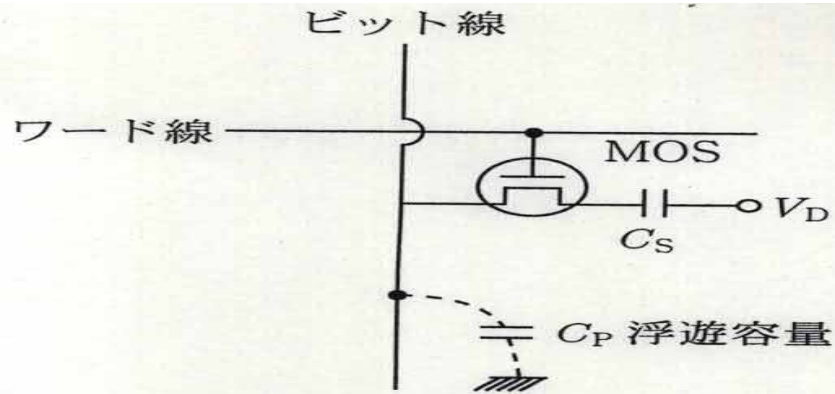
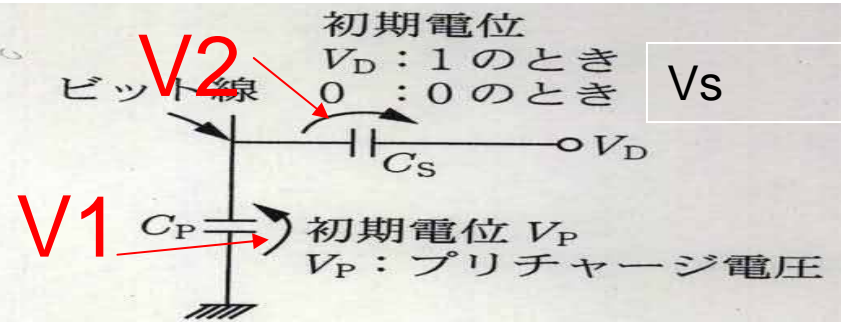


図 4.2 DRAM の構成の動作



(a) DRAM セル



(b) セルへの書き込み / 読出し

$$V_1 + V_2 = V_D$$

$$V_1 C_p - V_2 C_s = V_p C_p - V_s C_s$$

$$V_1 = (V_p C_p - V_s C_s + V_D C_s) / (C_p + C_s)$$

電荷保存則

## DRAM

1の読み出し時:  $V_p C_p / (C_p + C_s)$

0の読み出し時:  $(V_p C_p + C_s V_D) / (C_p + C_s)$

$V_p = V_D / 2$ とすると,

1の読み出し時:  $V_D (1 - C_s / (C_p + C_s)) / 2$

0の読み出し時:  $V_D (1 + C_s / (C_p + C_s)) / 2$

高速ページモード, SDRAM, RDRAM

リフレッシュ

セル当たり96msecごと

1行単位で同時に読み出して書き込む

# DRAMについての私の経験

## ・ 1 Kb DRAM

1974に購入(東光株式会社)

アクセスタイム 350nsec

容量 256KB

価格 1000万円

## ・ 256 Mb DRAM

2001年にパソコンのアドオンメモリとして購入

アクセスタイム 70nsec

容量 128MB

価格 5000円

## (2)固定型補助記憶装置

### ハードディスク

年率:60%で容量増加

浮上隙間は10~20nm程度

トラック数:5,000~30,000

セクタ数:100~500

セクタバイト数:512B

シーク(seek)時間:5~12msec

回転待ち(rotation latency)

3,600~15,000RPM (Rotations Per Minute)

3,600RPMで $1/(2 * 60)=8\text{msec}$

7,200RPMで4msec



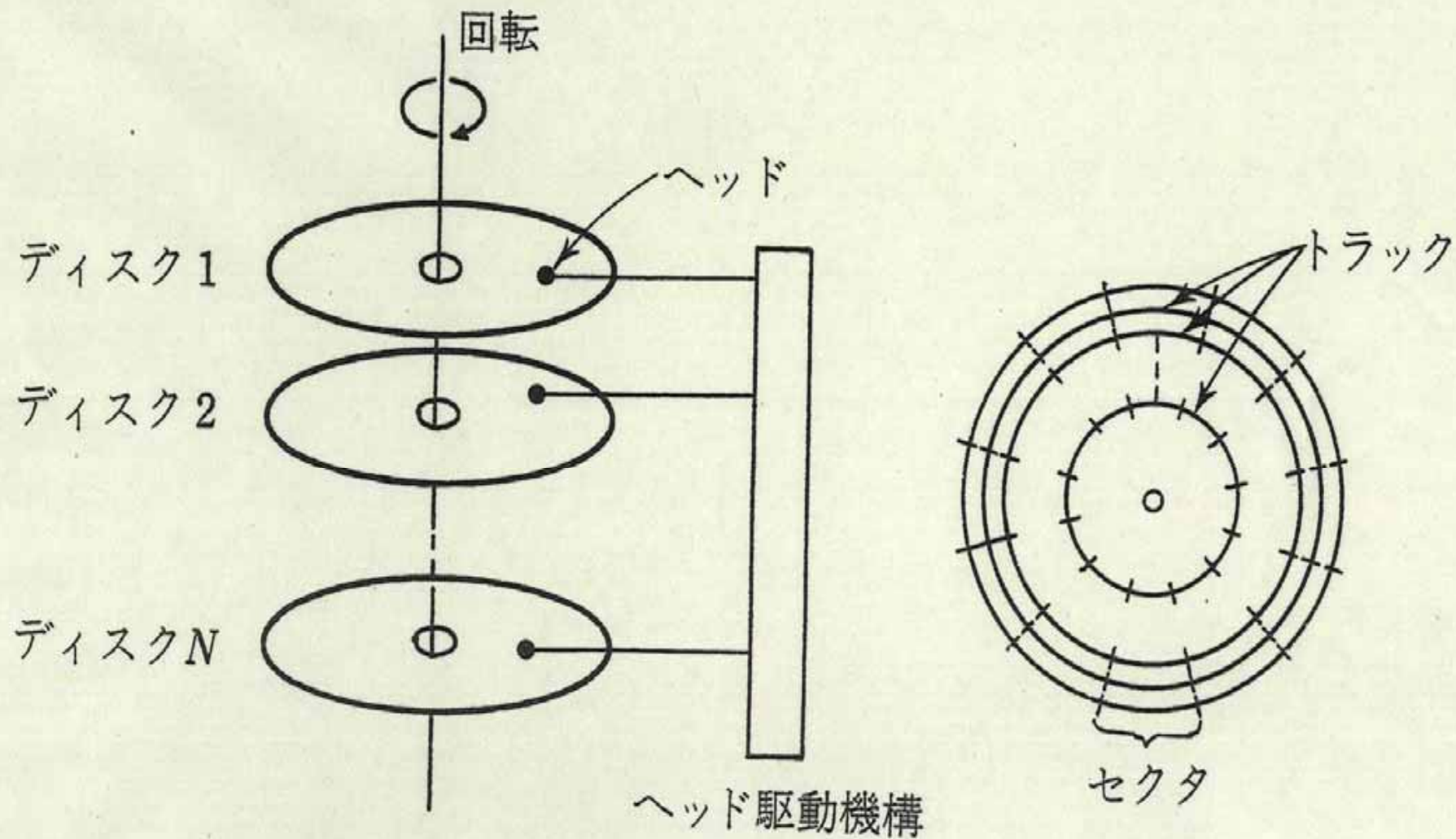


図 4.4 磁気ディスク装置 (HDA) の構成

### (3)可搬(リムーバブル)型補助記憶装置

フロッピディスク(FD)

フラッシュ(flash memory)

SuperDisk,

光および光磁気ディスク

CD-R(追記型, 一回の書込みのみ可能)

CD-RW(複数回の読出し書込みが可能)

MO(Magneto - Optical)

DVD(Digital Versatile Disk) - RAM

カセット型磁気テープ

2倍速, 4倍速のドライブ装置:

音楽用CDの基準データ転送速度

150KB/sの何倍

## (4)リードオンリメモリ (ROM)

マスクROM

PROM(Programmable ROM)

EPROM(Erasable PROM)

EEPROM(Electrically Erasable PROM)

OSなどの基本プログラム部分の格納

例えばローダ

制御記憶などのデコーダ

文字パターン

関数表

表 4.1 記憶素子/装置の諸元 (1999 年現在)

素子・装置	容量	アクセス時間	転送速度
SRAM	4~16 Mb	10~30 ns	—
DRAM	16~256 Mb	30~100 ns	—
SDRAM	16~128 Mb		100 Mb/s
	RAS より 2~3 クロック, CAS より 2~3 クロック, 以後 1 クロックごとにバースト転送		
補助記憶装置：固定型 磁気ディスク	4~20 GB (記憶密度 1~5 Gb/平方インチ)	10~20 ms	10~50 MB/s
補助記憶装置：リムーバブル型 半導体記憶 フラッシュメモリ	10~240 MB		W 0.8 MB/s R 8 MB/s
磁気記憶 フロッピディスク	1.44 MB	100 ms	100 KB/s
光磁気ディスク(MO)	0.64~1.3 GB	30 ms	3~6 MB/s
光コンパクトディスク CD-ROM (書込み不可)	640 MB	125 ms	3.6 MB/s
CD-R (書込み一度だけ可)	640 MB	170 ms	W 1.2 MB/s R 3 MB/s
CD-RW (書込み複数回可)	640 MB	150 ms	W 0.3 MB/s R 3.6 MB/s
DVD	5.2 GB	120 ms	W 1.4 MB/s R 2.8 MB/s
Zip	250 MB	30 ms	3 MB/s
SuperDisk	120 MB	100 ms	680 KB/s
磁気テープカートリッジ	10~50 GB	数 秒	0.5~6 MB/s

### 4.1.3 記憶階層方式

参照の局所性 (locality of references)

時間局所性 (temporal locality)

空間局所性 (spatial locality)

アナロジ:

頭の中、メモ帳、机上的本、引出しのファイル、  
部屋の本箱、地下倉庫

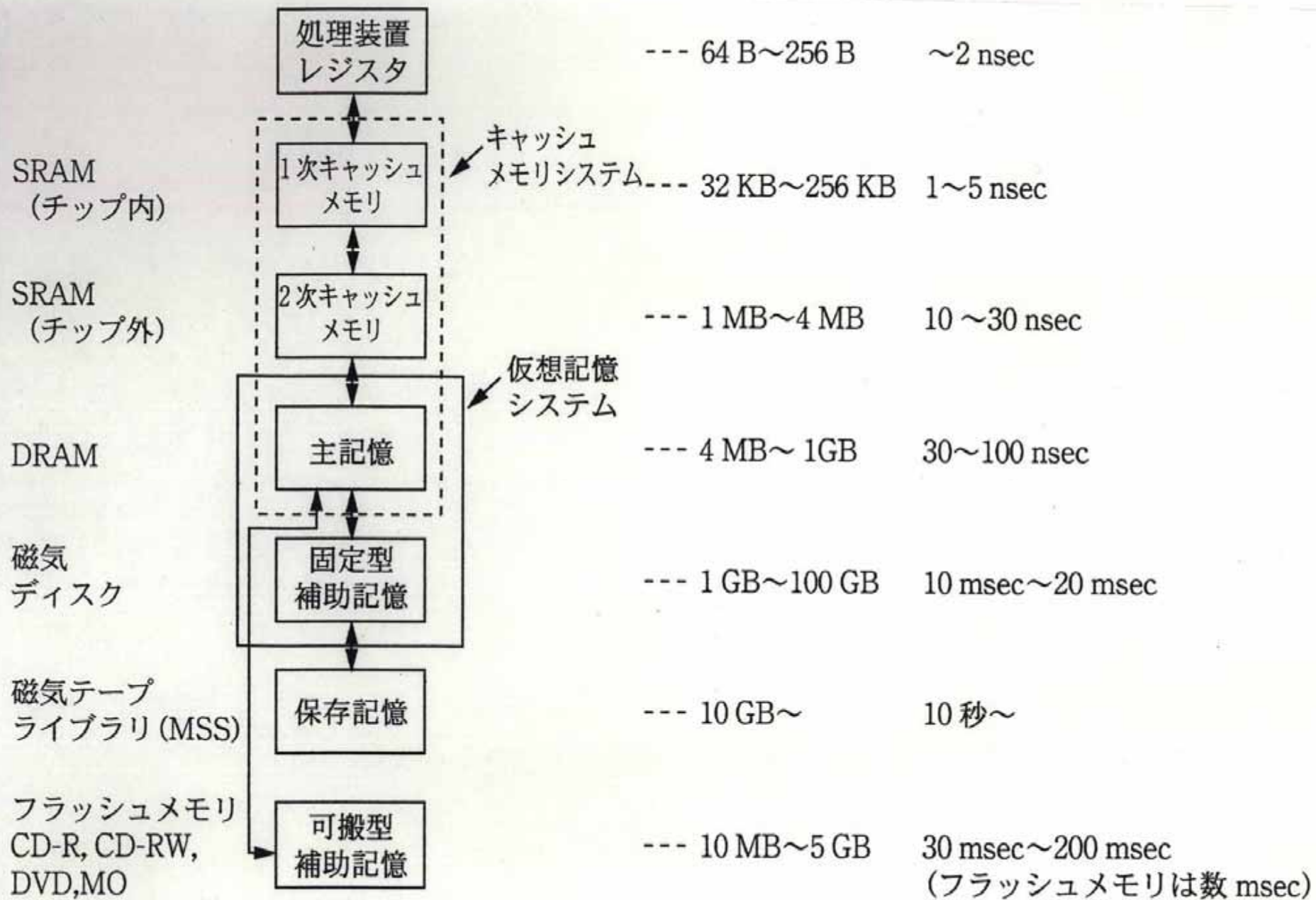


図 4.5 記憶階層方式

## 4.2 仮想記憶

ユーザのアドレス空間: 4 GB

実記憶容量: 4 MB

### 4.2.1 基本方式

(1) ページング方式

(2) セグメンテーション方式

### 4.2.2 写像方式

(1) 直接写像

ページテーブル

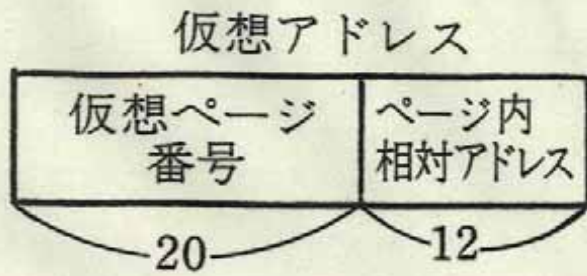
多重レベルページング

セグメントテーブル

セグメンテーション + ページング

Pentiumの方式

ページ化セグメンテーション



# 1次元アドレス

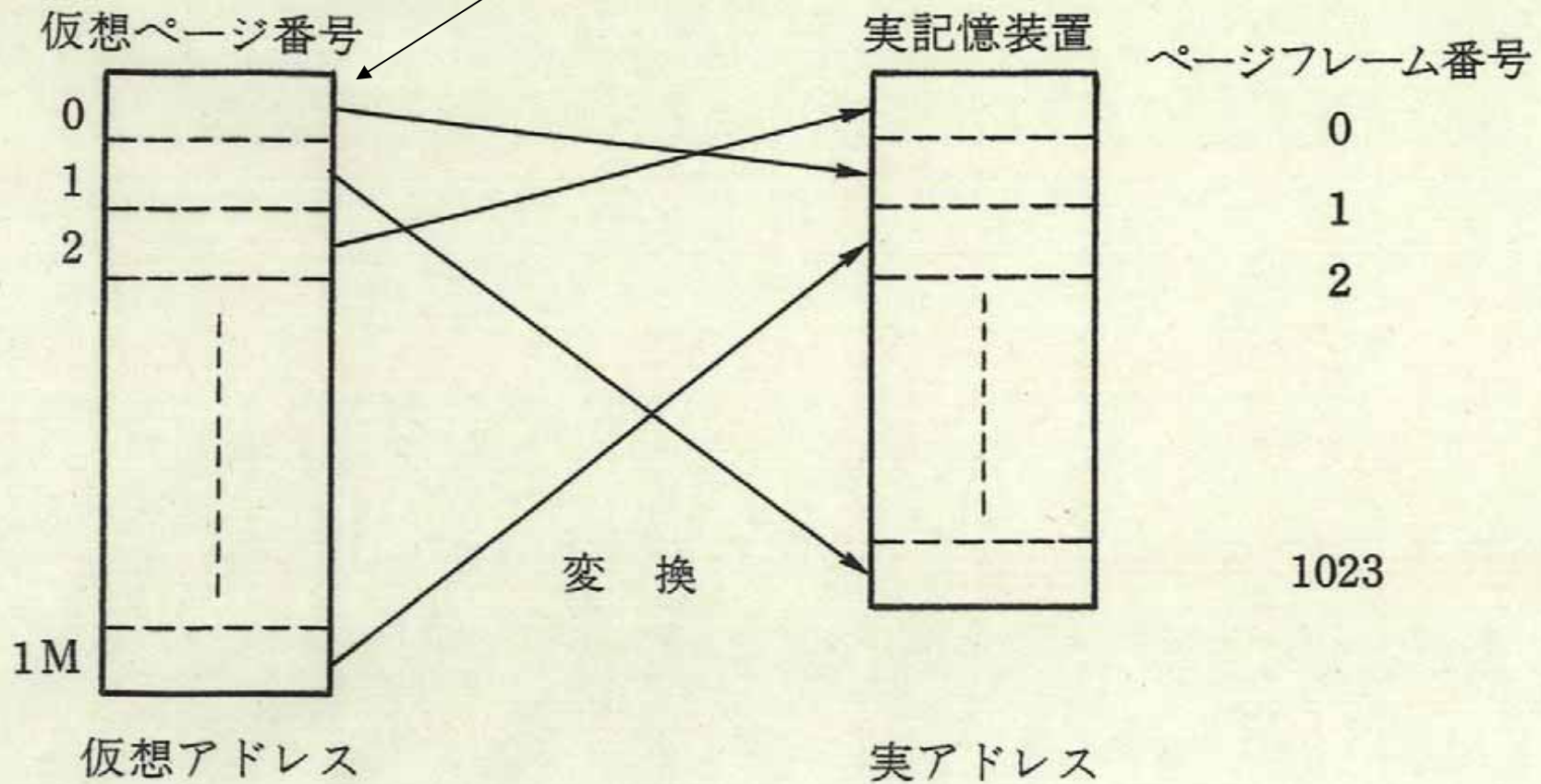


図 4.6 ページング方式



## ページングでの仮想アドレスの生成

IBMメインフレームのアドレッシングモード

例インデックスモード

$R_b + R_x + \text{変位}$

$R_b$ :ベースレジスタ(汎用レジスタ使用)

$R_x$ :インデックスレジスタ(同上)

OP	$R_d$	$R_b$	$R_x$	D
----	-------	-------	-------	---

8	4	4	4	12
---	---	---	---	----

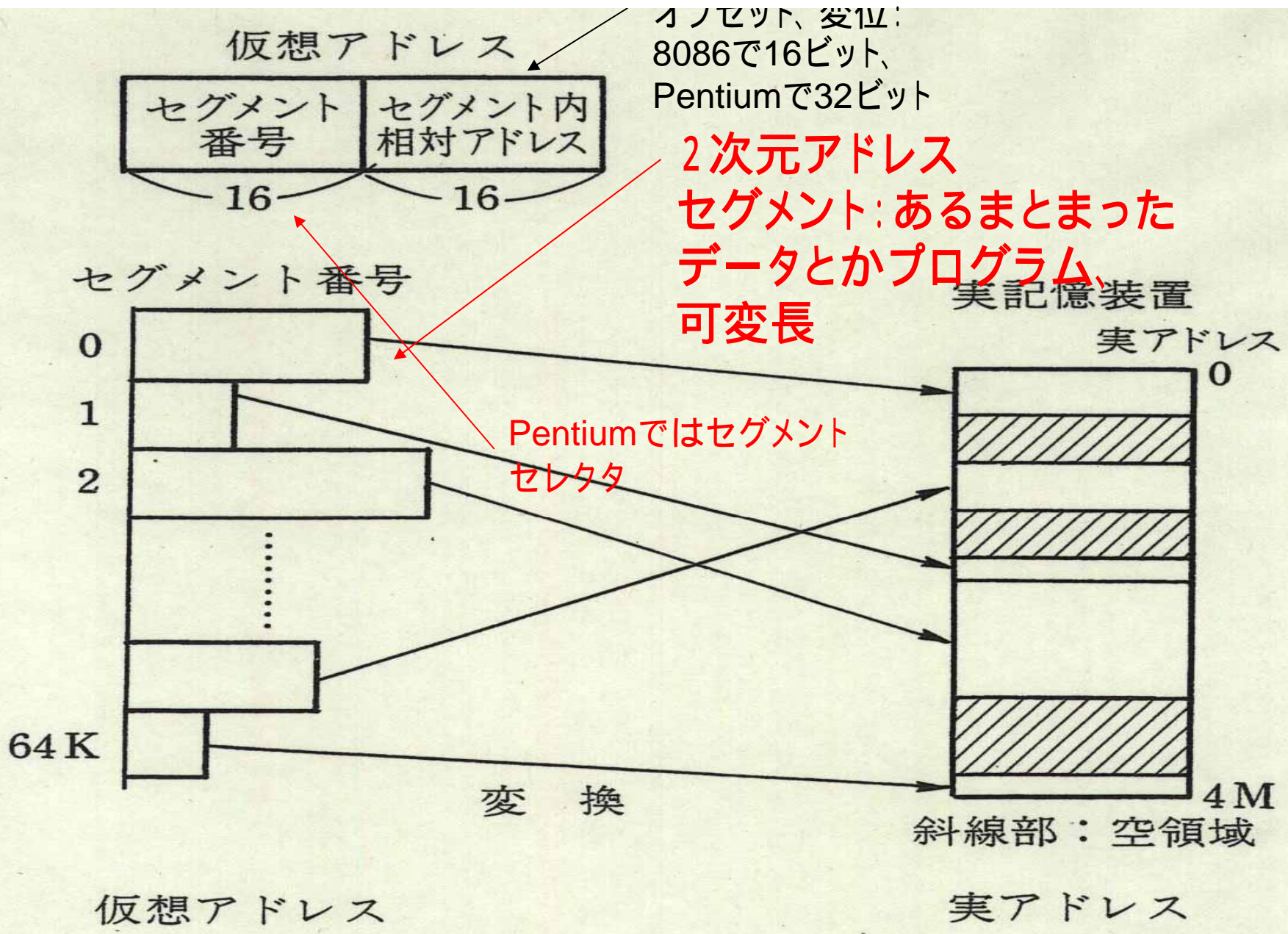


図 4.7 セグメンテーション方式

# Pentiumでのセグメンテーション

## セグメントの種類

コード、データ、スタック、...

セグメントレジスタ:セグメントセレクタを格納

CS:コード

SS:スタック

DS:データ

ES,FS,GS:予備

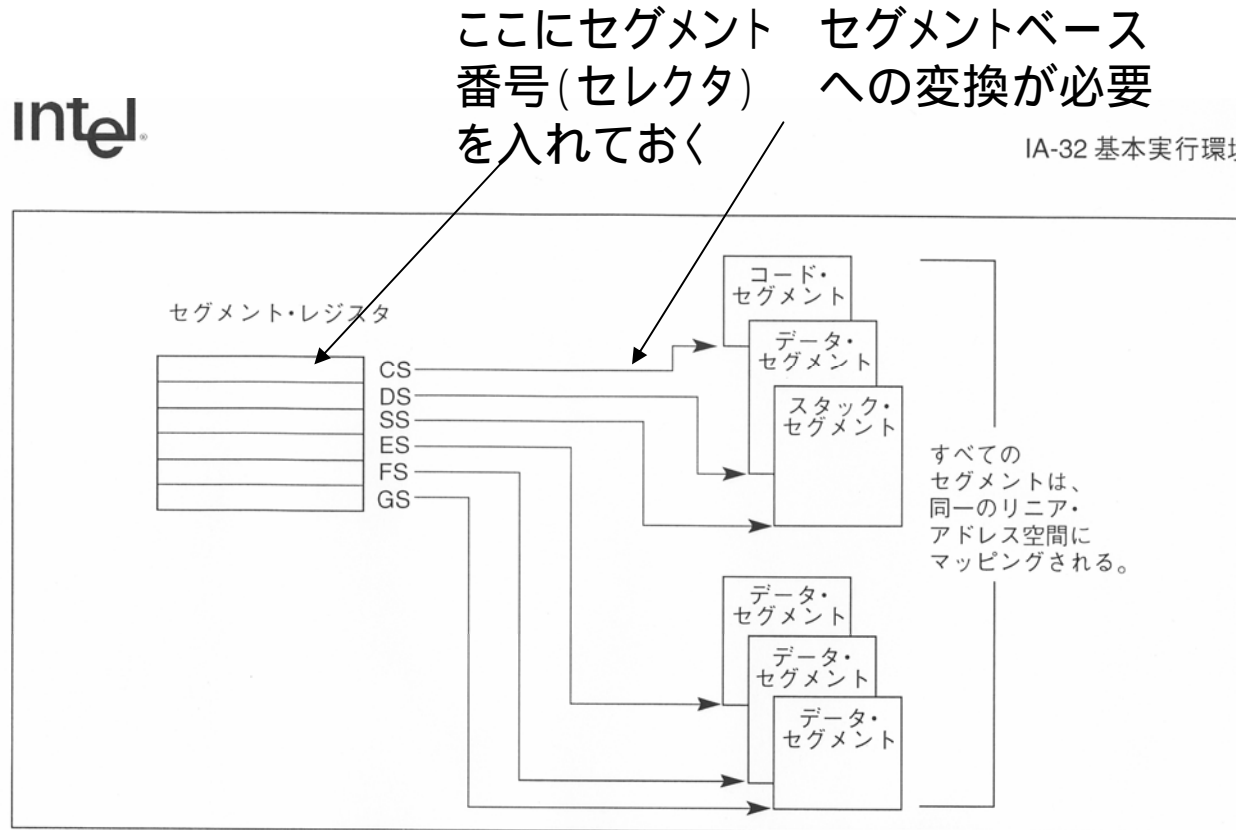
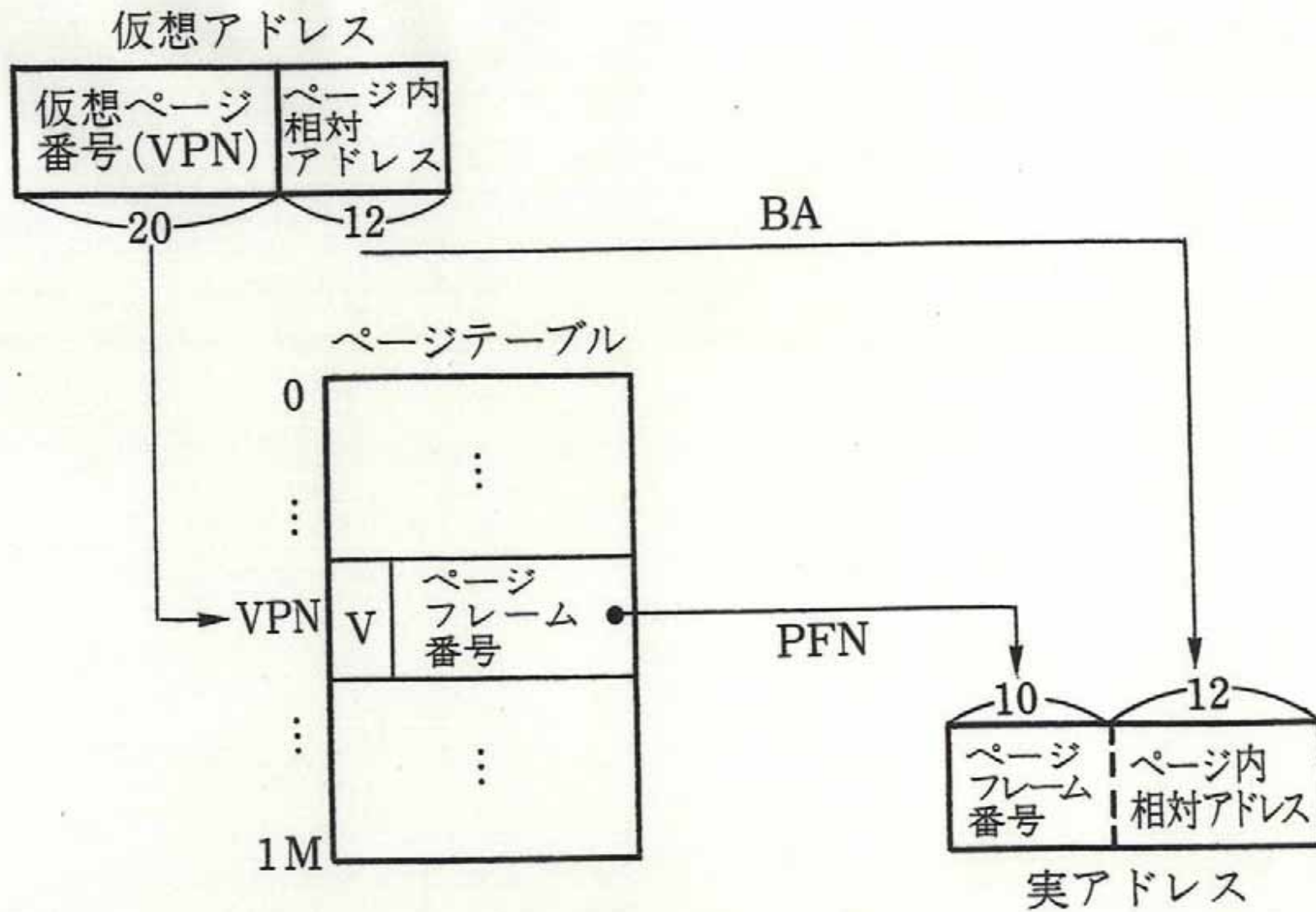


図 3-6. セグメント化メモリ・モデルでのセグメント・レジスタの使用法

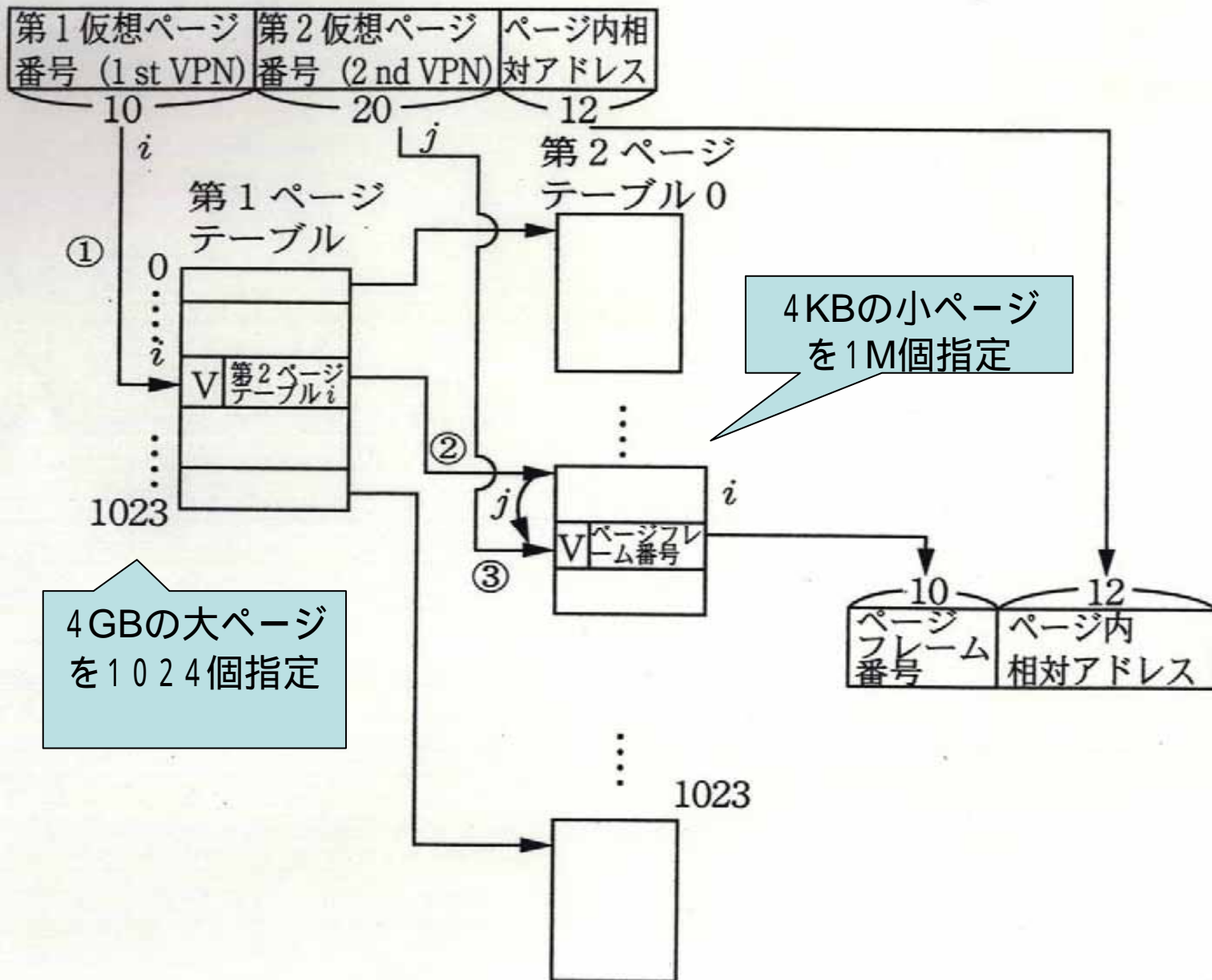
各命令のオペランドごとに対象となるセグメントレジスタが暗黙に決まっている。

そのセグメント内でアドレッシングモードを適用

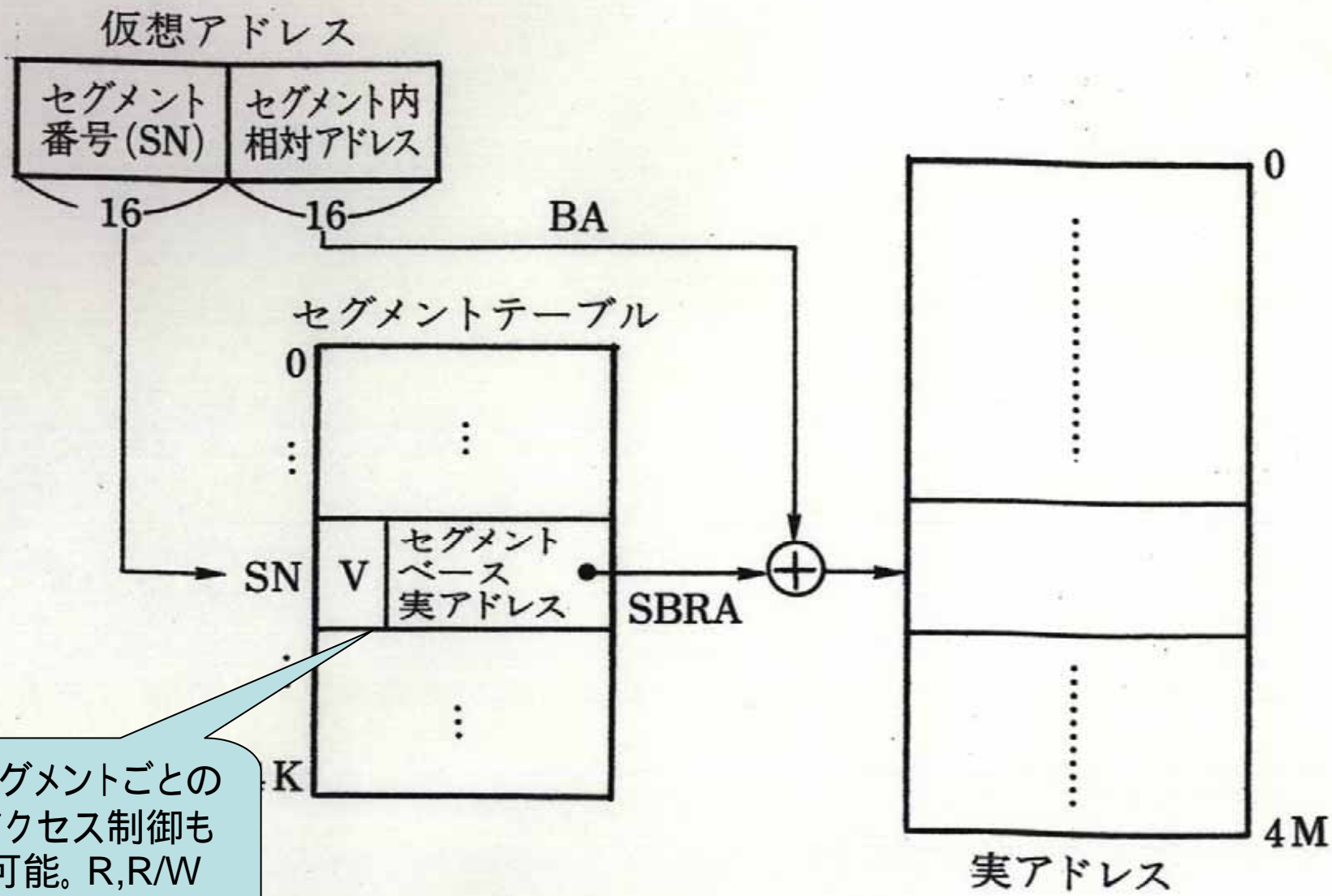


(a) ページテーブル

図 4.8 直接写像方式



(b) 多重レベル (階層) ページング



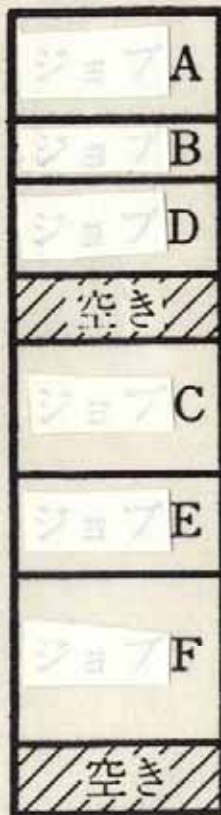
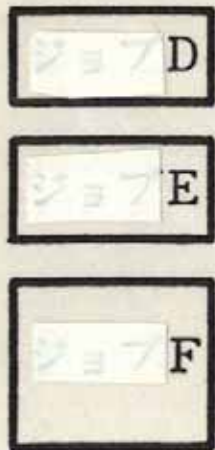
(c) セグメンテーション方式

図 4.8 直接写像方式 (つづき)



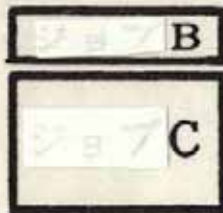
(i)

ジョブ D.E.F 割付け

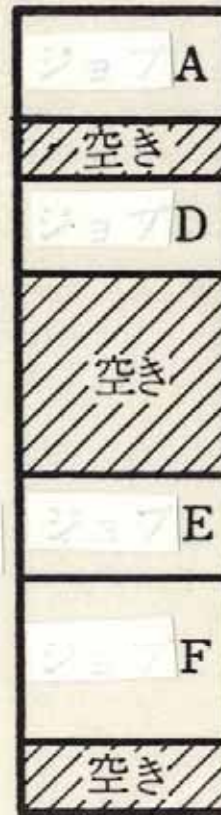
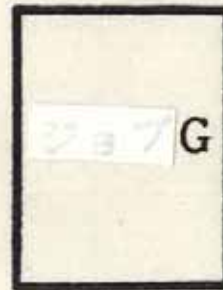


(ii)

B.C 解放



\*G 割付け不可能



(iii)

ジョブ G のサイズ

Eの追い出し  
詰め(コンパクション)

フラグメンテーション、断片化



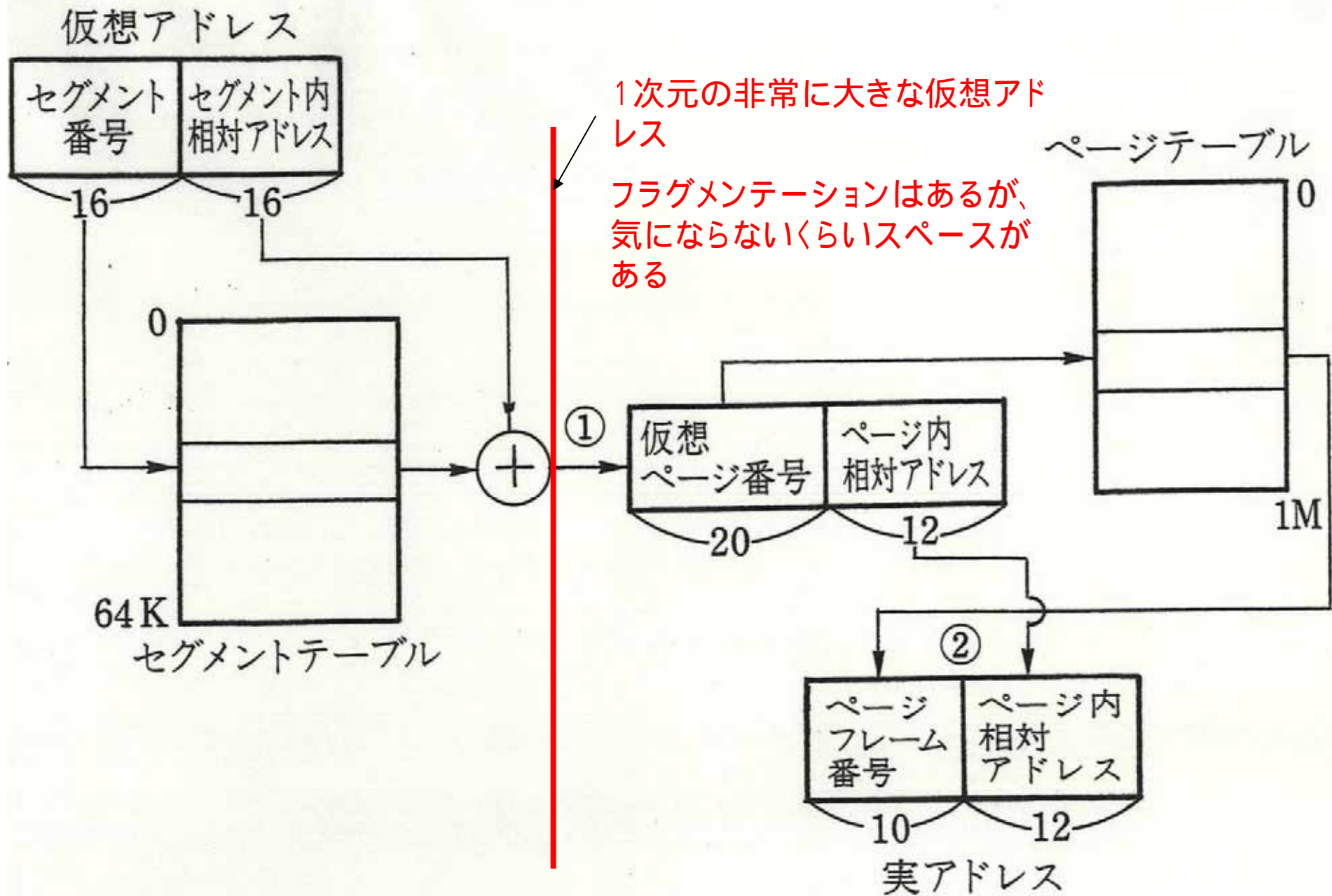


図 4.9 セグメンテーション方式へのページング機構の導入

## (2)連想写像

ページフレームテーブル

ハッシュ法

TLB法

TLBミスの対処

ハードまたはソフト

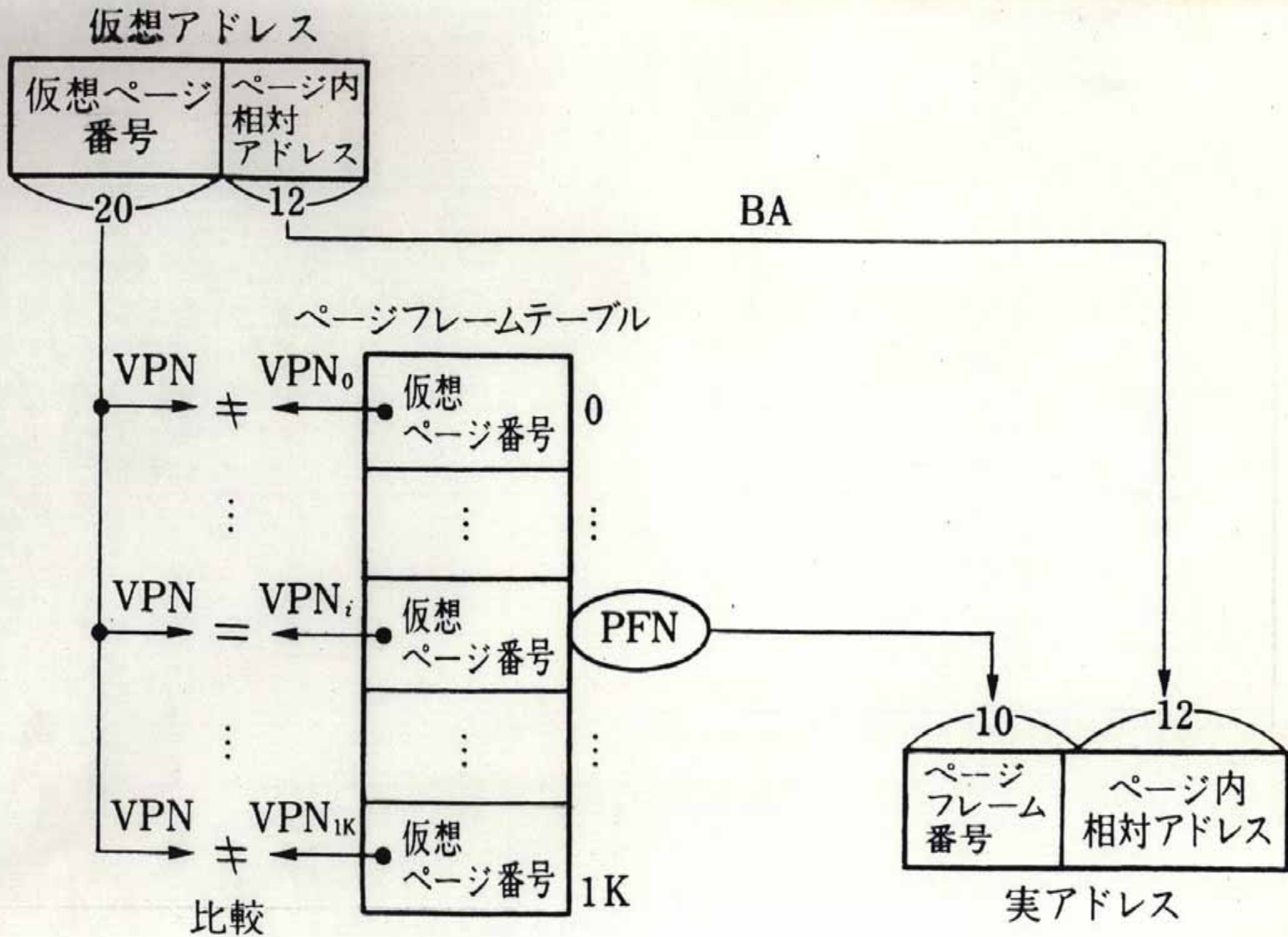
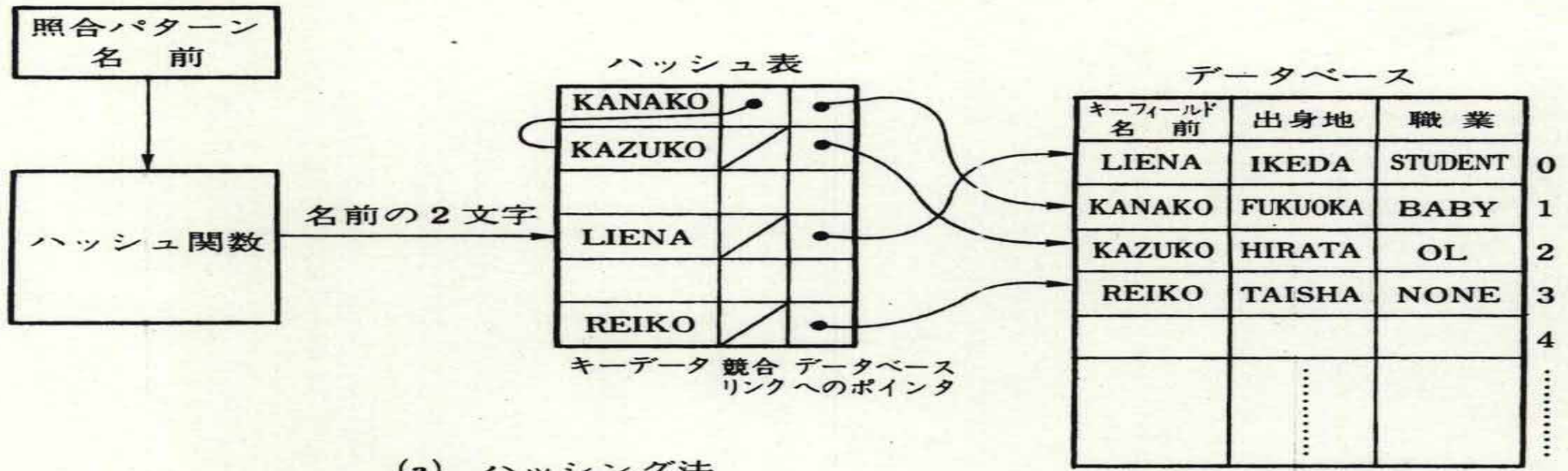
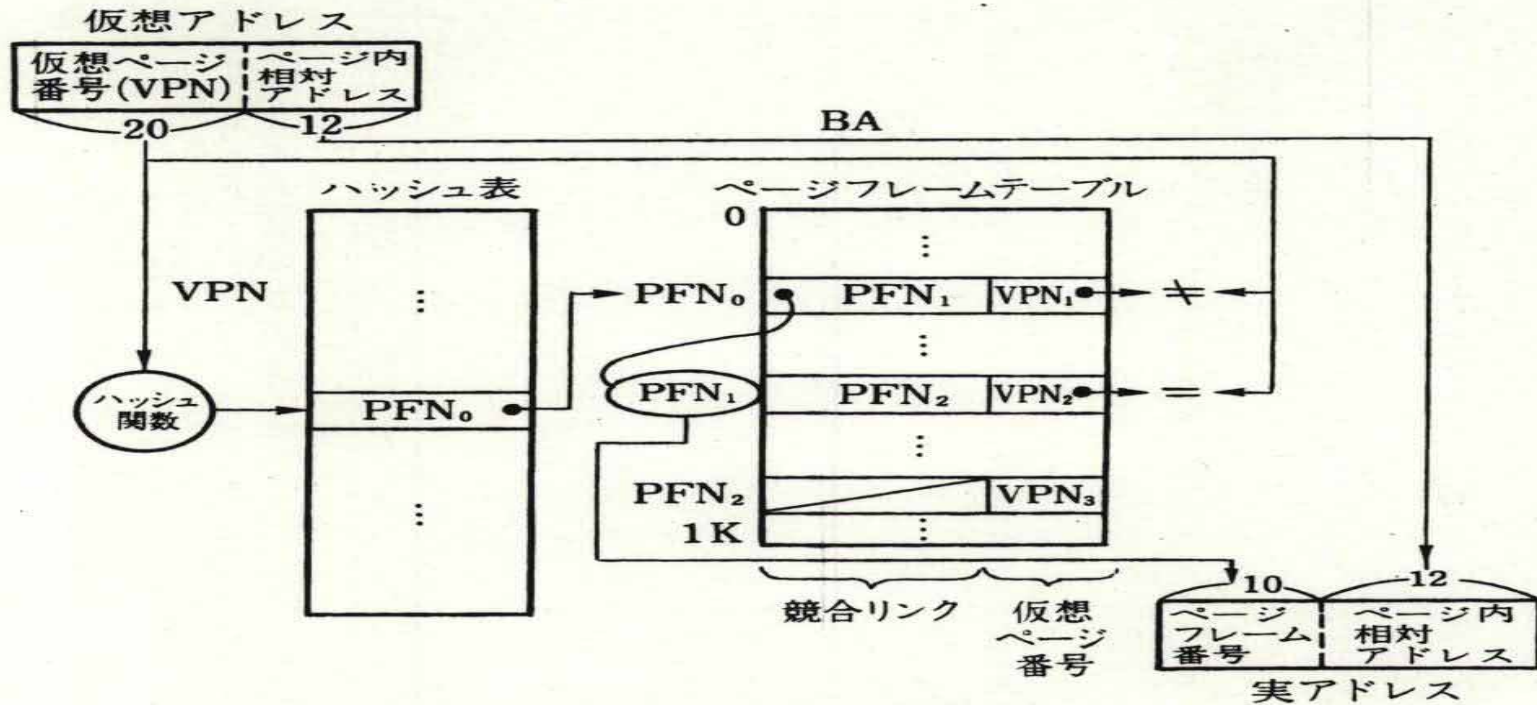


図 4.9 連想写像方式



(a) ハッシング法



(b) ハッシングを用いた連想写像

図 4.10 ハッシングを用いた連想写像

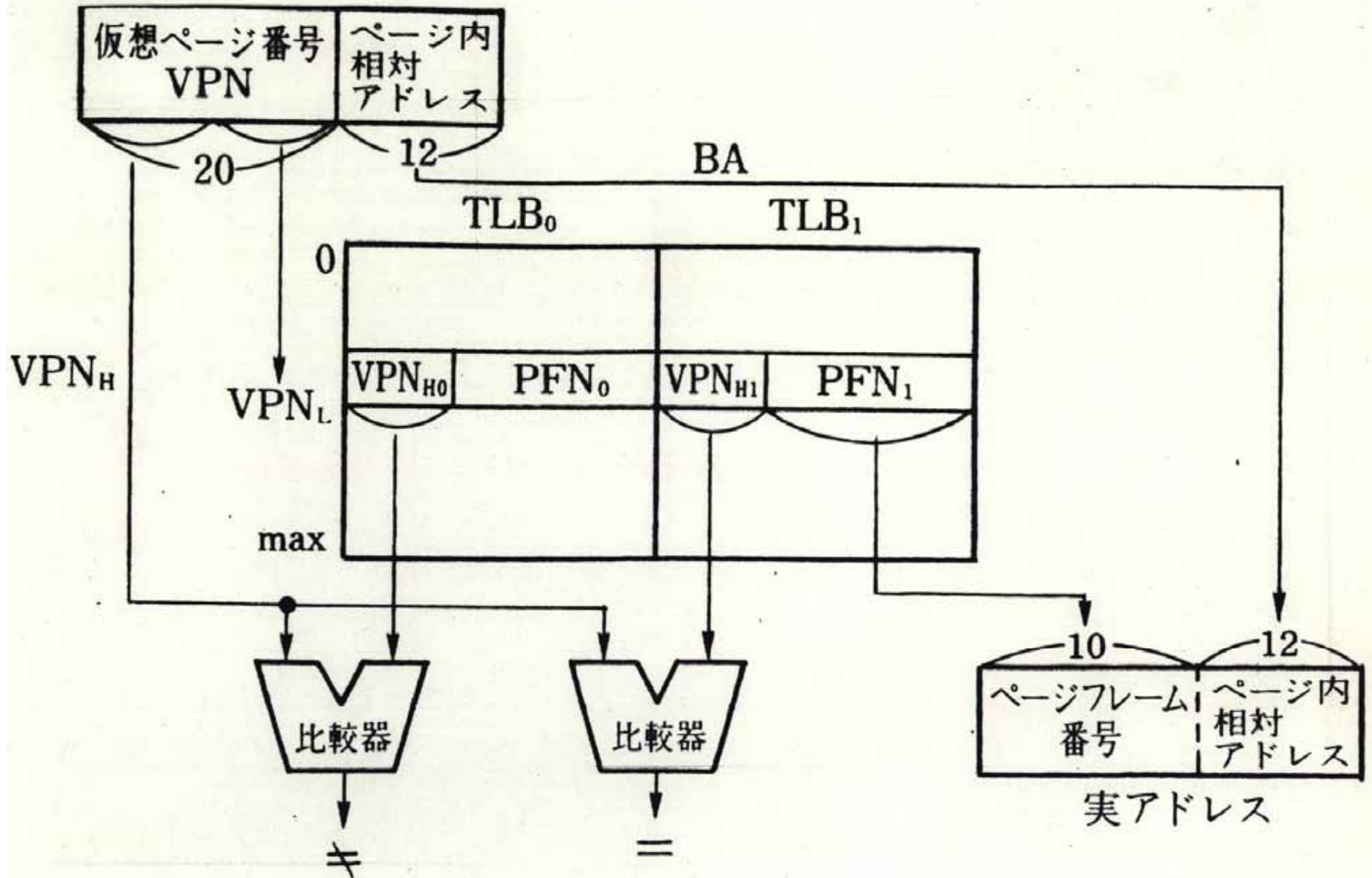


図 4.11 TLB の構成

# 実際のアドレス空間の大きさ

## IBMメインフレーム

1964年 S360 24ビット

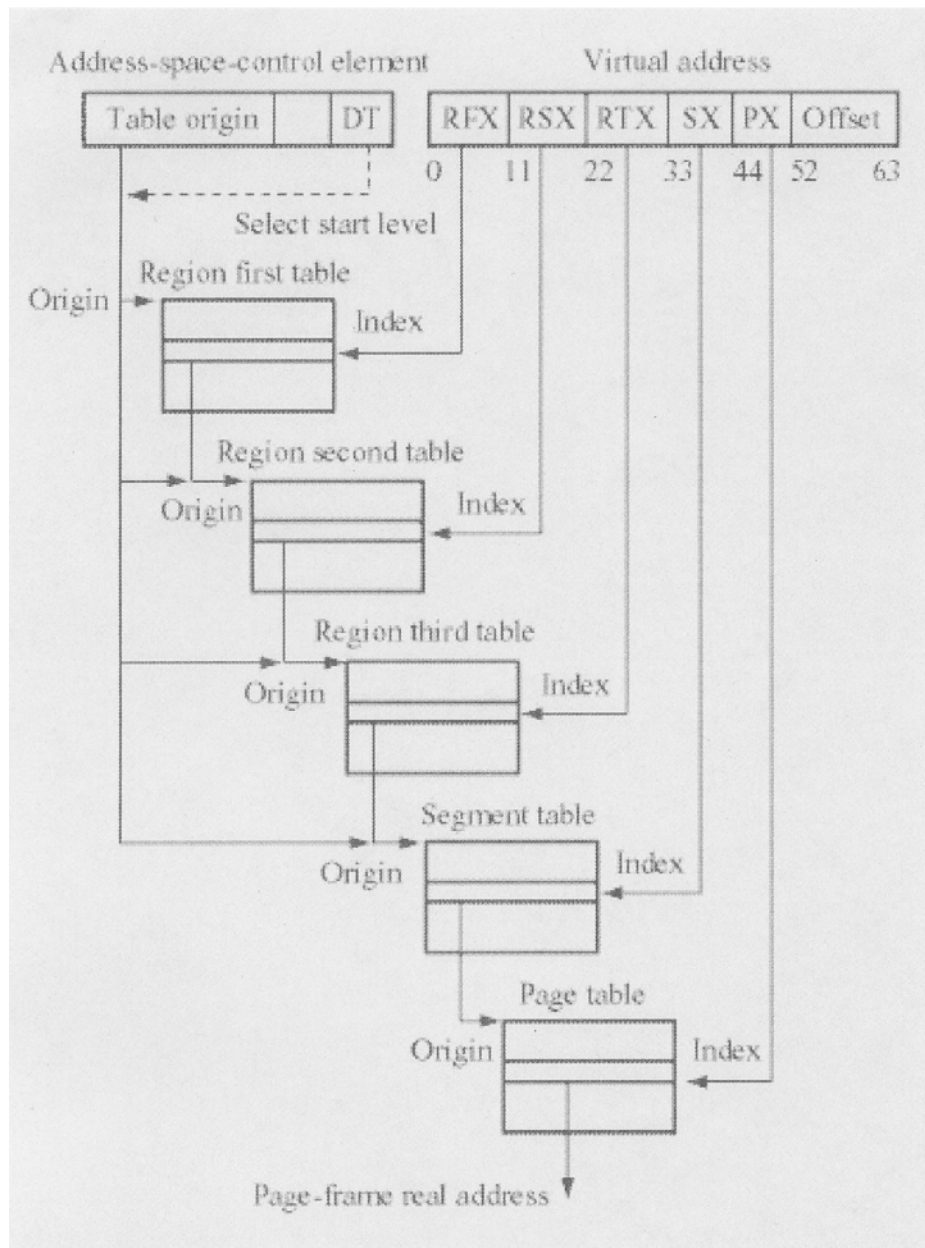
1970年 S370 24ビット、仮想記憶の導入

1983年 S370-XA 31ビット

1988年 ESA/370 (Enterprise Systems  
Architecture)

1990年 ESA/390

2000年 z/Architecture 64ビット



# z/Architectureでの ページテーブル ページテーブル ウォーク

K.E.Plambeck : Development  
and Attributes of  
z/Architecture, IBM J.Res  
Dev,46,4/5,2002

Figure 6

z/Architecture dynamic address translation.

Intel マイクロプロセッサ		最大実アドレス容量	
1978	8086	1MB	16ビット セグメント長最大64kB
1982	286	16MB	
1985	386	4GB	32ビット セグメント長最大4GB
1989	486	4GB	セグメント数:16383
1993	Pentium	4GB	
1995	Pentium Pro	64GB	仮想アドレス空間
1997	Pentium II	64GB	14ビット:セグメント番号指定
1999	Pentium III	64GB	32ビット:セグメント内アドレス指定
2000	Pentium 4	64GB	計46ビット



### 4.2.3 ページフレームの管理

#### (1) 各種管理テーブル

#### (2) ページ置き換えアルゴリズム

FIFO: First In First Out

FINUFO: First In Not Used First Out

LRU: Least Recently Used

ワーキングセット: Working Set

#### (3) 多重プログラミング制御と置き換えアルゴリズム

グローバルLRU法

ワーキングセット法

### 4.2.4 仮想空間の共有と保護

多重仮想記憶方式

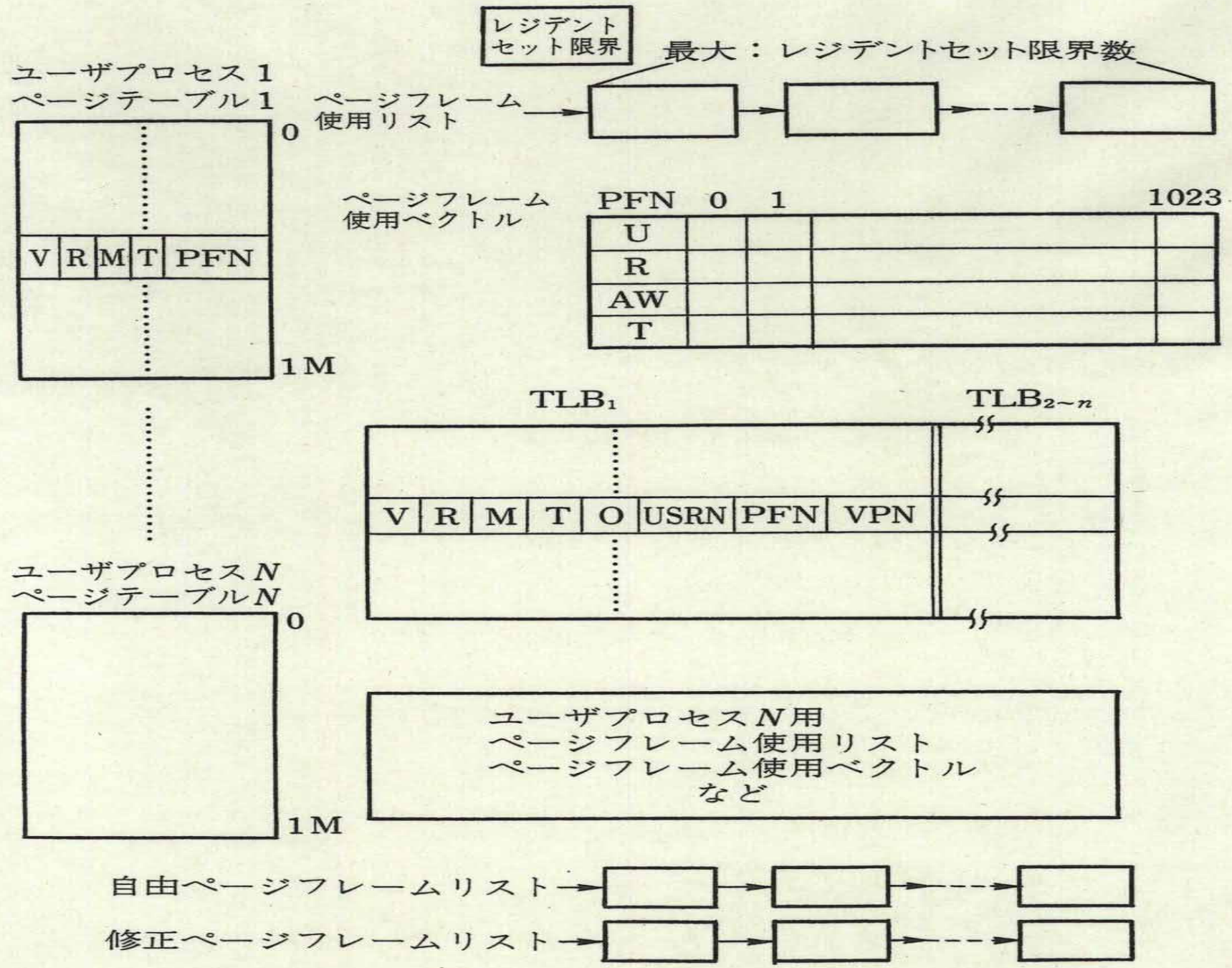


図 4.13 ページフレームの管理

# ページ要求

- 時刻-1、-0 プロセス0, 1 仮想ページ0、仮想ページ10要求  
ページフォールト
- 時刻1 プロセス1 仮想ページ0 ページ枠0  
実行中へ
- 時刻3 プロセス2 仮想ページ10 ページ枠1  
プロセススイッチで実行中
- 時刻4 プロセススイッチでプロセス1へ  
仮想ページ256 ページ枠2
- 時刻5 プロセス1 仮想ページ512 ページ枠3  
TLBフル 仮想ページ0 TLB追い出し
- 時刻7 プロセス1 仮想ページ7 レジデントセット限界  
仮想ページ0に対応したページ枠0を置き換えて  
仮想ページ7に割り付け

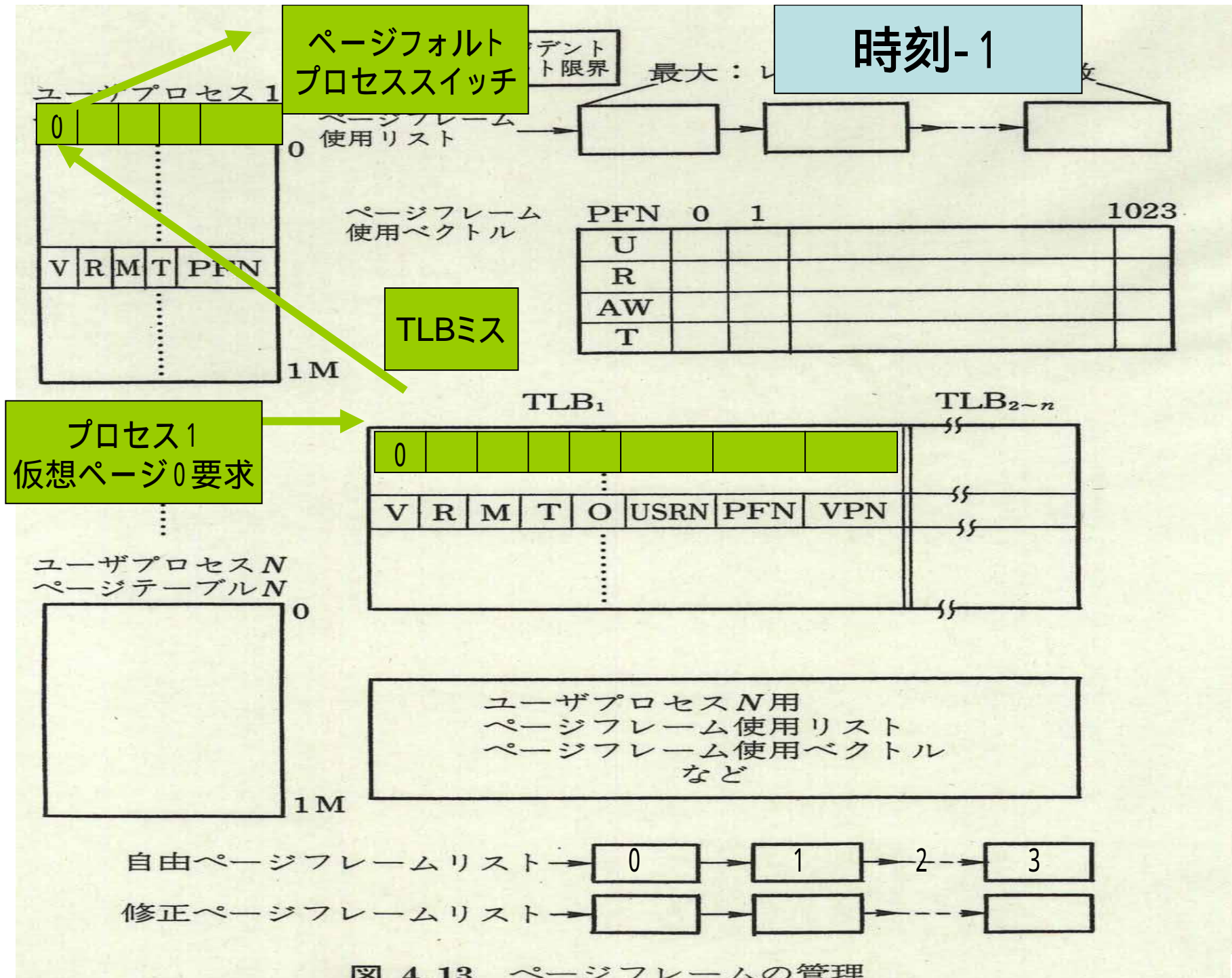


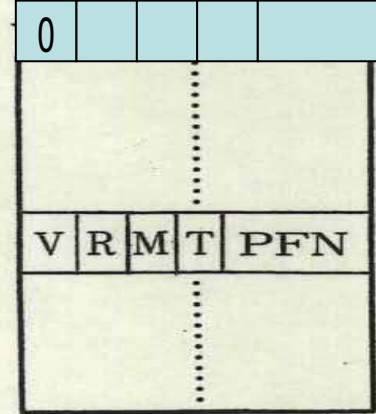
図 4.13 ページフレームの管理

時刻-0

レジデント  
セット限界

最大：レ

ユーザプロセス 1



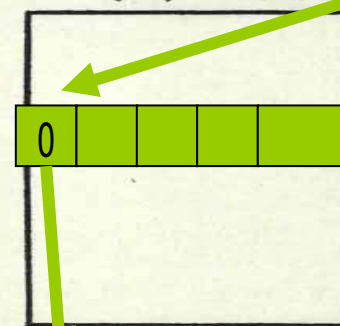
ページフレーム  
使用リスト

ページフレーム  
使用ベクトル

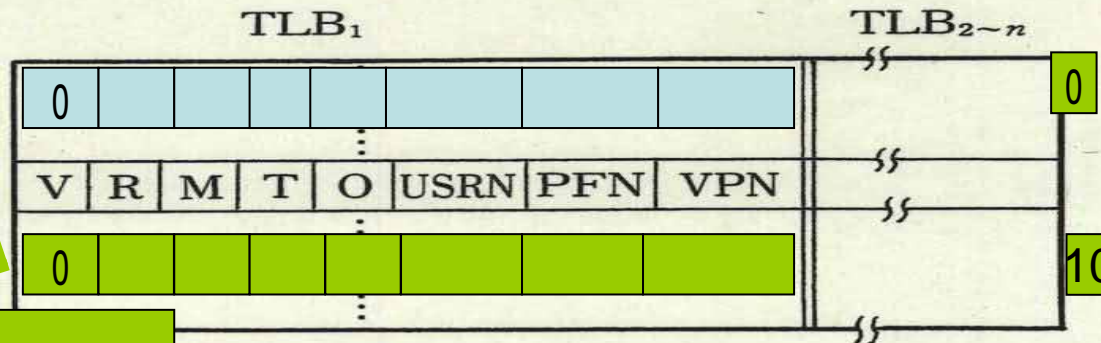
PFN	0	1	...	1023
U				
R				
AW				
T				

プロセス2  
仮想ページ10要求

ユーザプロセス  
ページテーブル



TLBミス



ユーザプロセスN用  
ページフレーム使用リスト  
ページフレーム使用ベクトル  
など

ページフォルト  
プロセススイッチ

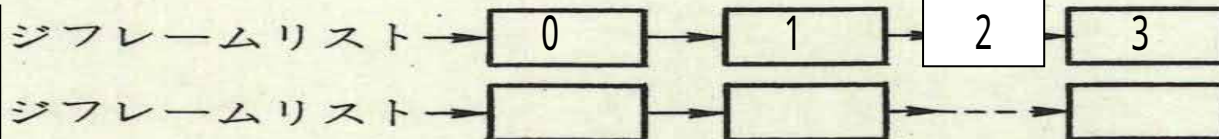


図 4.13 ページフレームの管理

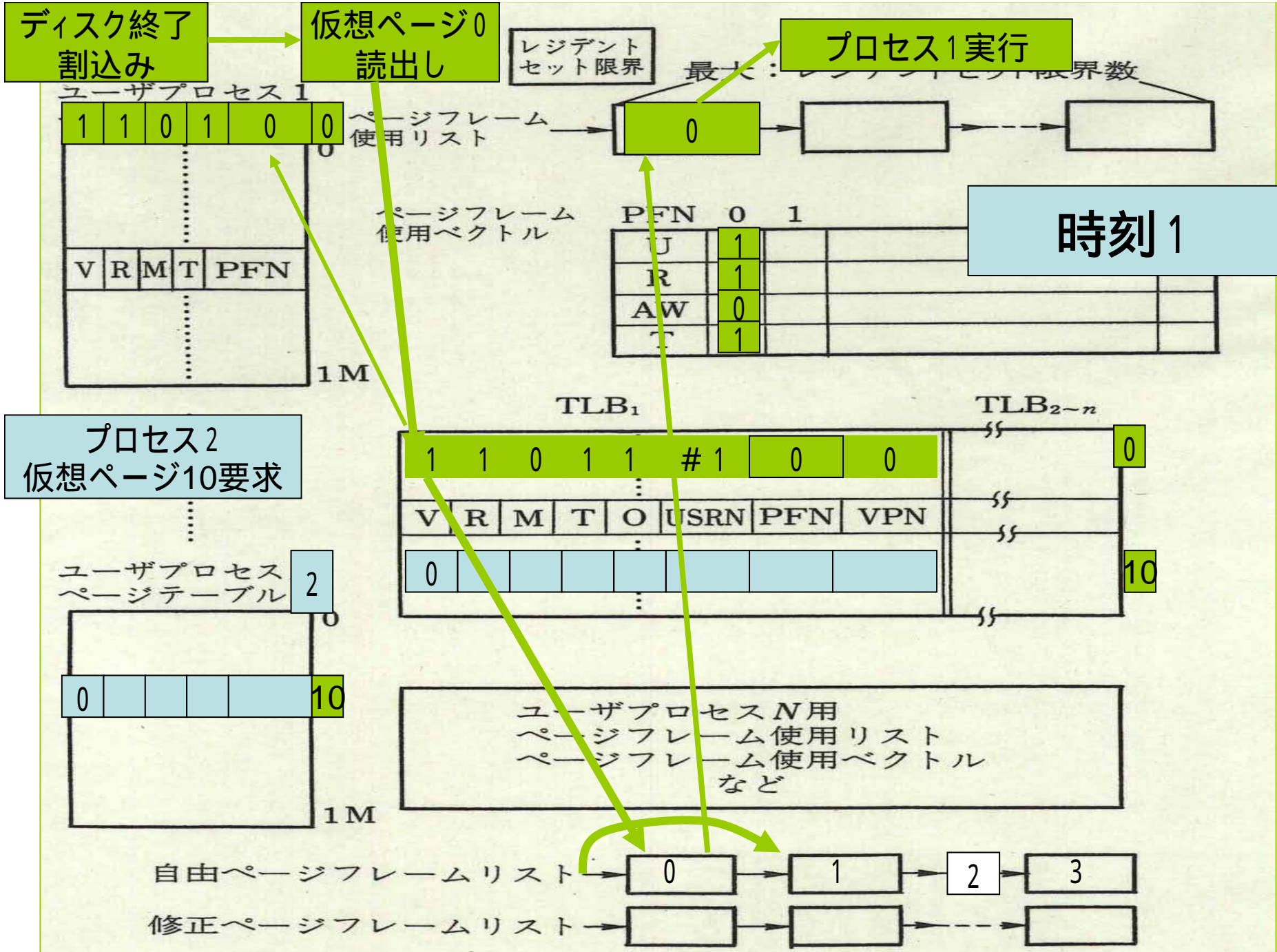


図 4.13 ページフレームの管理

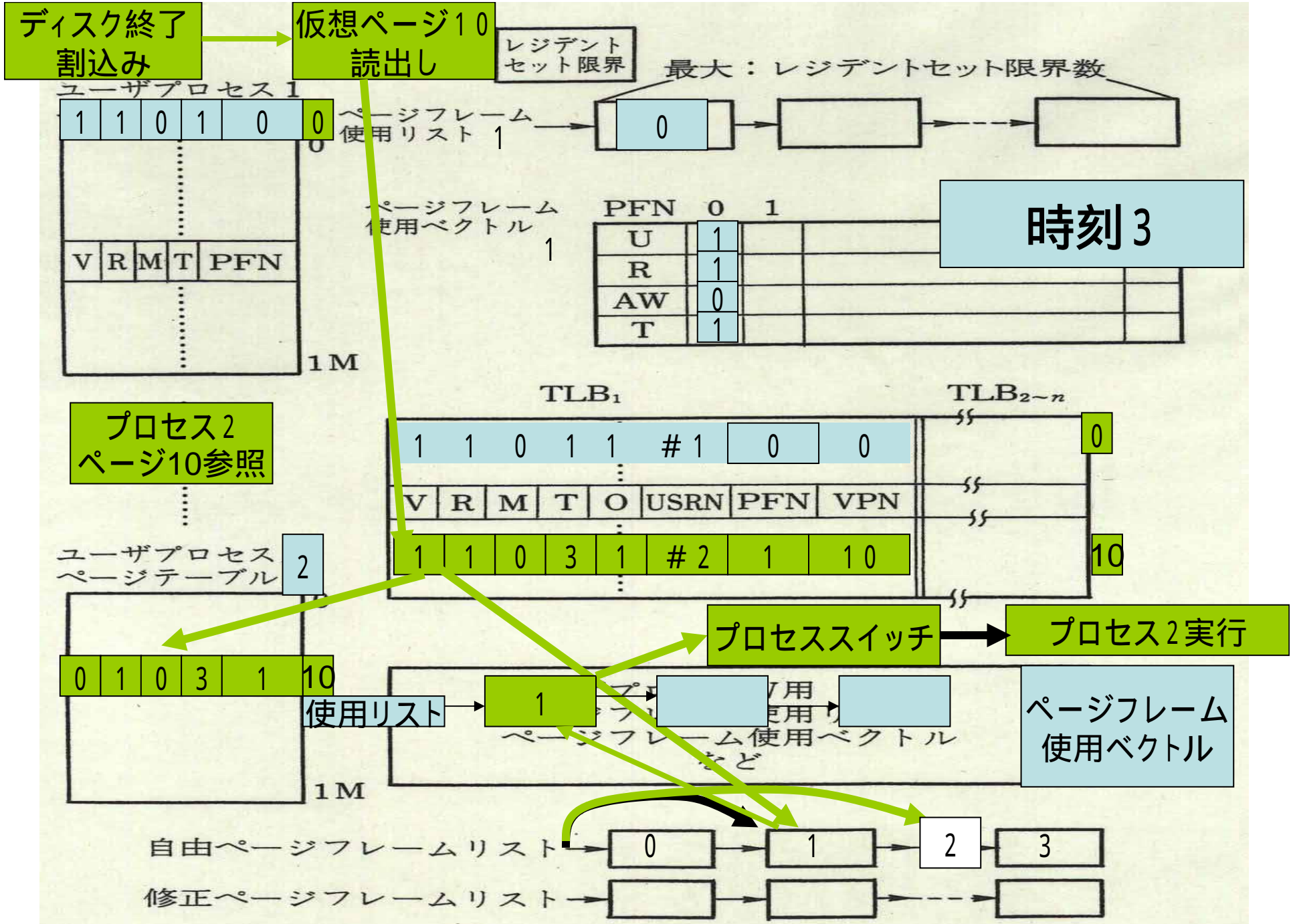


図 4.13 ページフレームの管理

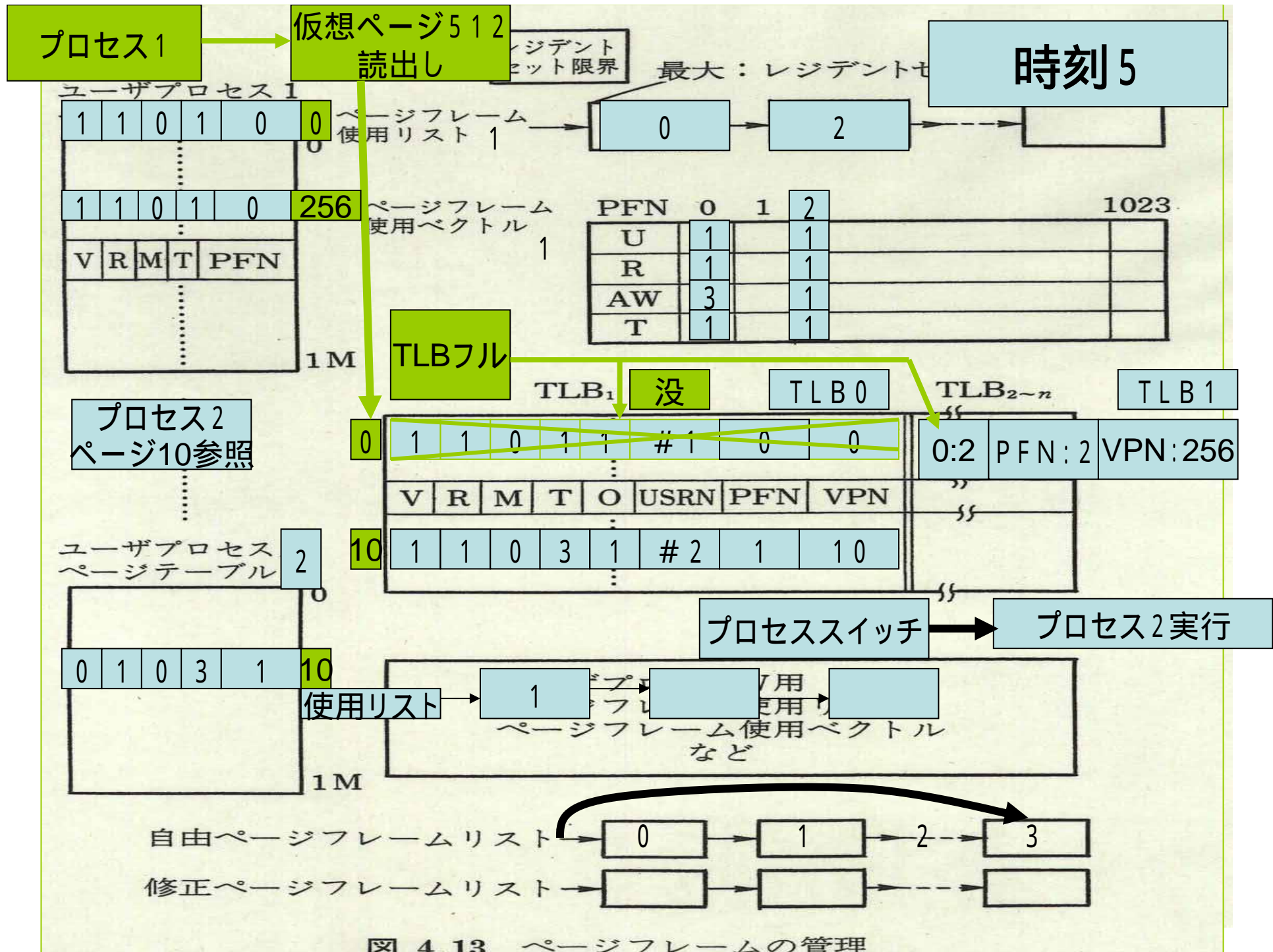


図 4.13 ページフレームの管理



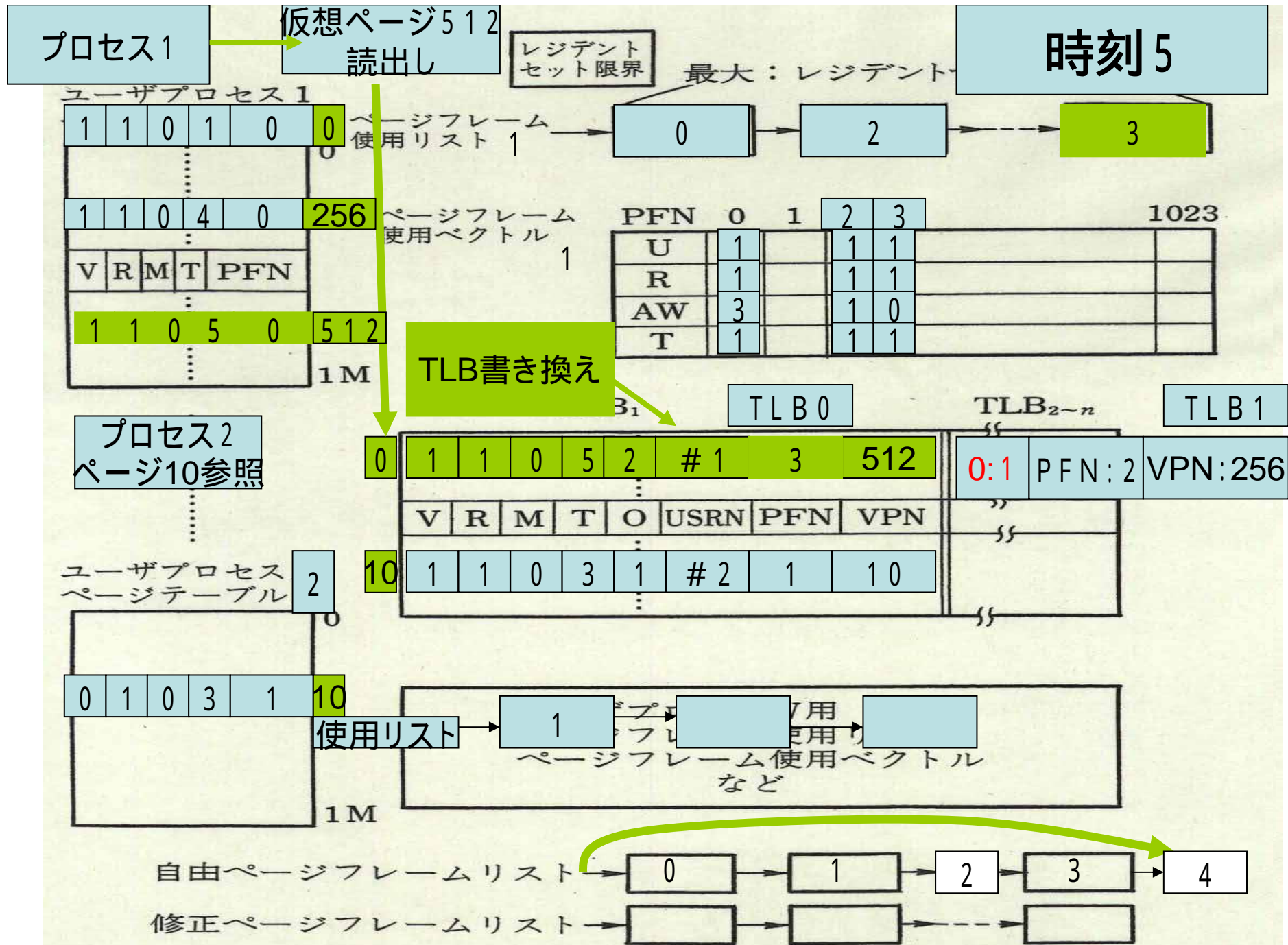


図 4.13 ページフレームの管理

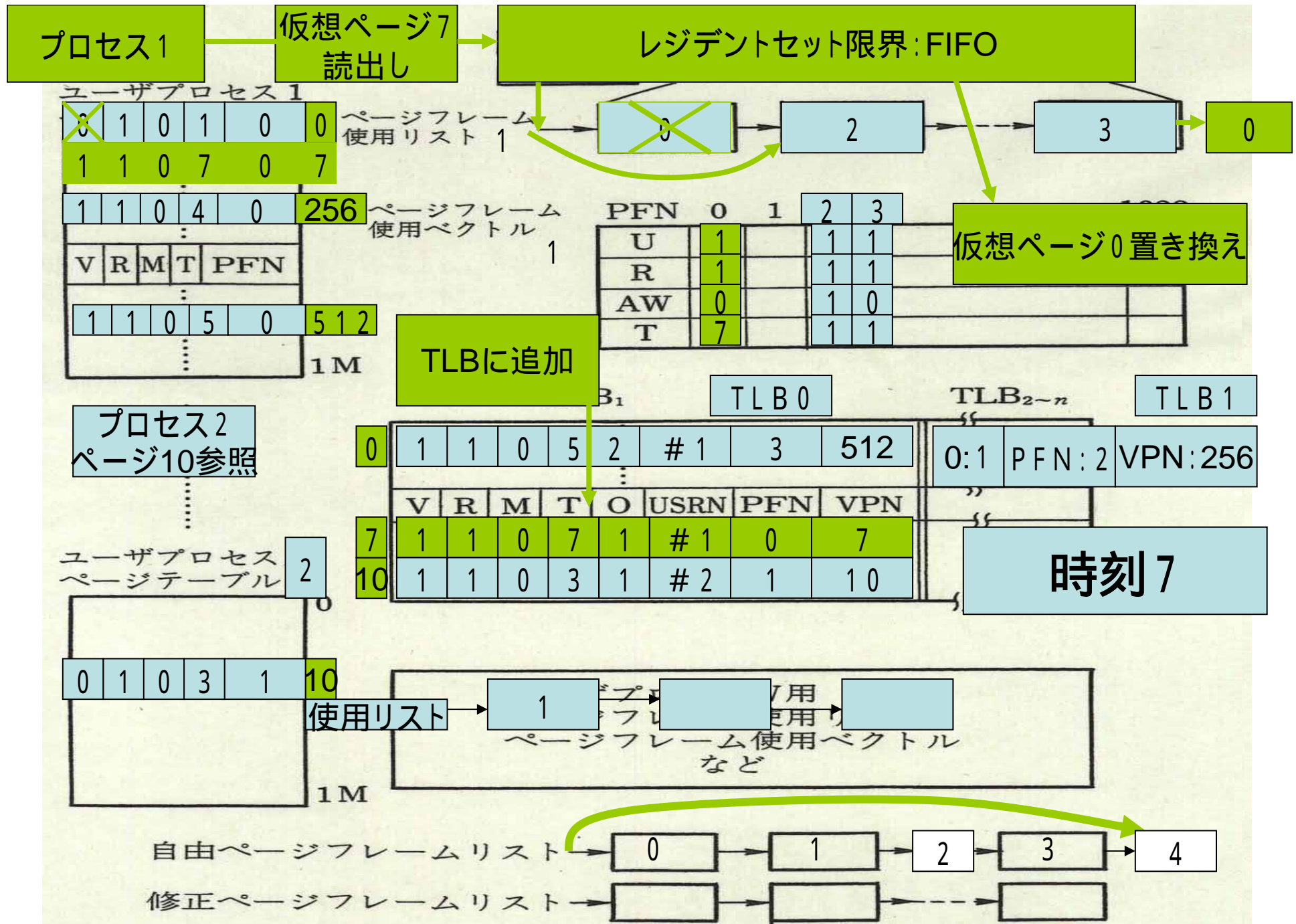


図 4.13 ページフレームの管理

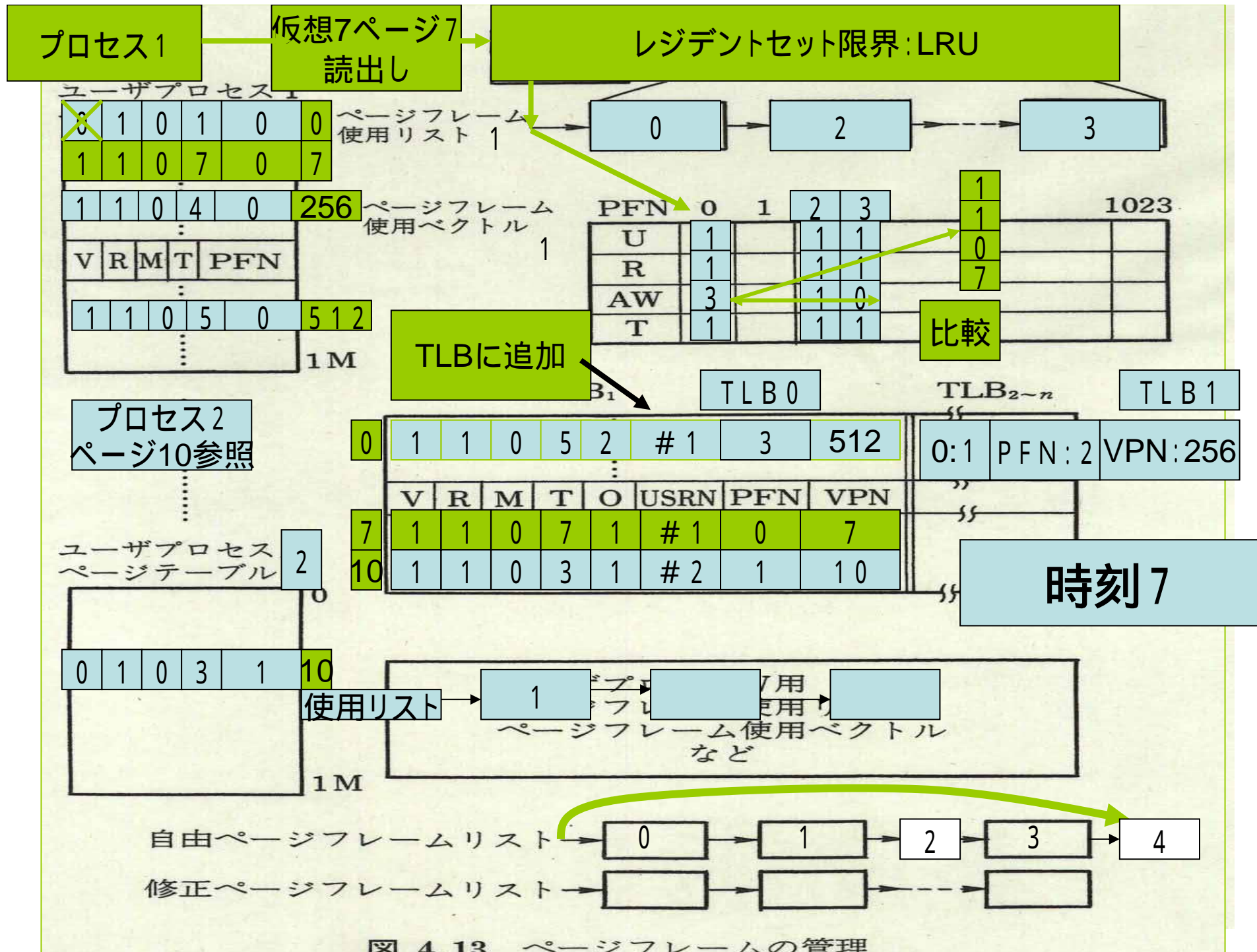


図 4.13 ページフレームの管理

# ページ置換えアルゴリズム:FIFO

		時刻	1	2	3	4	5	6	7	8	9	10	11	12
参照仮想ページ番号			1	2	3	4	1	2	5	1	2	3	4	5
主記憶の内容	レジデントセット 限界=3の場合 ページフォールト:9回		1†	2†	3†	4†	1†	2†	5†	5	5	3†	4†	4
				1	2	3	4	1	2	2	2	5	3	3
					1	2	3	4	1	1	1	2	5	5
	レジデントセット 限界=4の場合 ページフォールト:10回		1†	2†	3†	4†	4	4	5†	1†	2†	3†	4†	5†
				1	2	3	3	3	4	5	1	2	3	4
					1	2	2	2	3	4	5	1	2	3
						1	1	1	2	3	4	5	1	2

† : ページフォールト

図 4.13 FIFO 方式における異常現象

[マドニック, ドノバン (池田克夫訳): オペレーティングシステム, 日本コンピュータ協会 (1976)]

# ページ置換えアルゴリズム: FINUFO

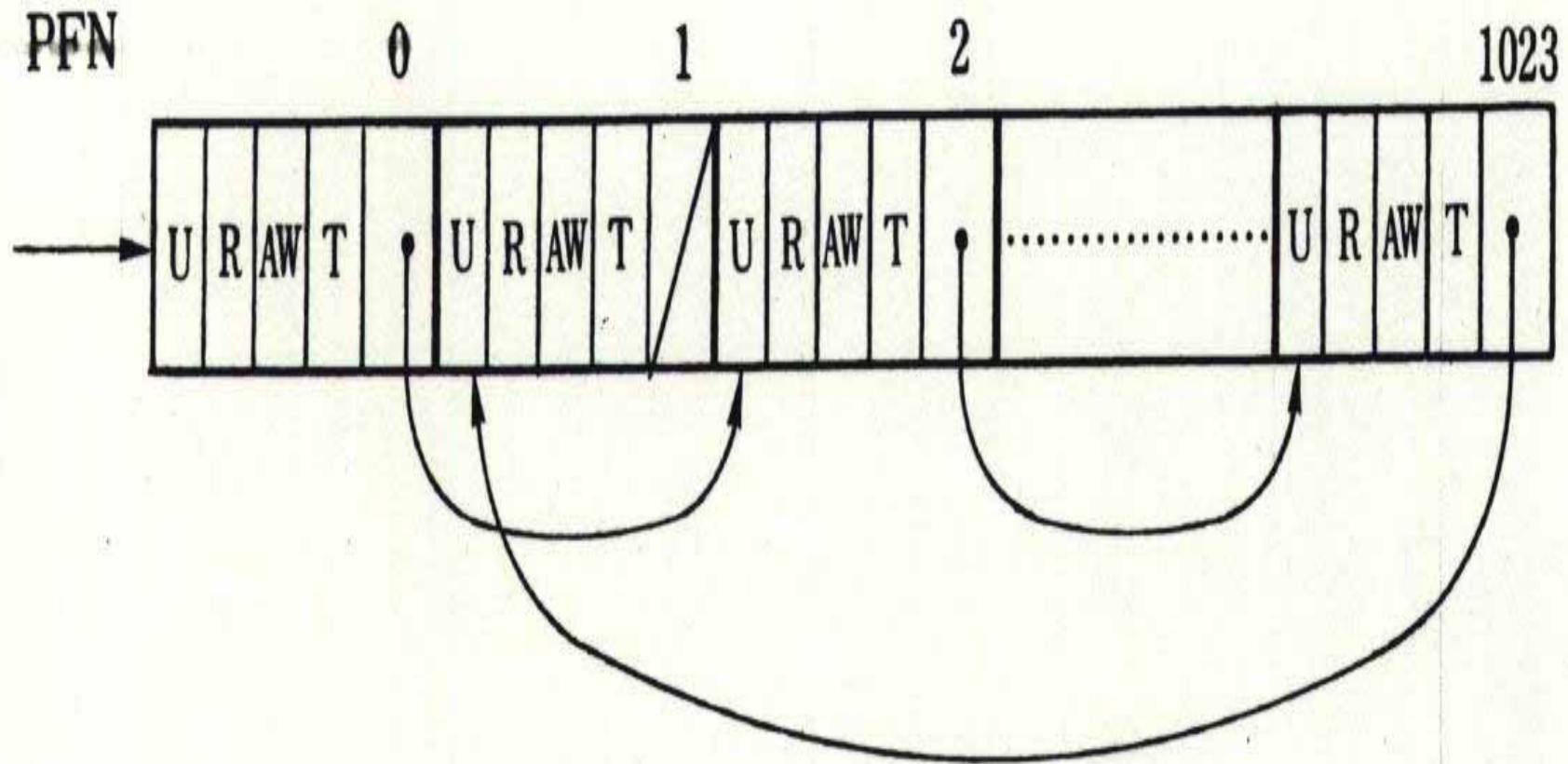


図 4.14 FINUFO のページフレーム使用ベクトル

	1	2	3	4	5	6	7	8	9	10	11	12	
P =	4	3	2	1	4	3	5	4	3	2	1	5	
M = 3		4+	3+	2+	1+	4+	3+	5+	4	3	2+	1+	5+
			4	3	2	1	4	3	5	4	3	2	1
			(4)	(3)	(2)	(1)	4	3	(5)	(4)	(3)	2	
F =		+	+	+	+	+	+			+	+	+	

包含関係

(a) M = 3

$$F = 10$$

$$f = \frac{10}{12} \approx 83\%$$

	1	2	3	4	5	6	7	8	9	10	11	12	
P =	4	3	2	1	4	3	5	4	3	2	1	5	
M = 4		4+	3+	2+	1+	4	3	5+	4	3	2+	1+	5+
			4	3	2	1	4	3	5	4	3	2	1
			4	3	2	1	4	3	5	4	3	2	
				4	3	(2)	1	1	(1)	(5)	(4)	3	
F =		+	+	+	+		+			+	+	+	

(b) M = 4

$$F = 8$$

$$f = \frac{8}{12} \approx 67\%$$

図 3.29 LRU 置換えアルゴリズムの例

ドノバン

### 3.6 デマンド・ページングによる記憶管理

包含関係の正式な証明は多少長たらしいので本書では示さない。証明の手がかりは、 $M(P, c, t)$  の値をどのようにして決定するかを考えることにより得られる。たとえば、 $M(P, 3, 7)$  を決定するには7番目のページ参照——ページ5——から開始して、ページの軌跡を逆にたどり、最初に出てきたページ番号を3つ求めればよい。 $M(P, 3, 7)$  は5, 3, 4である。 $M(P, 4, 7)$  はページ番号を4つ求めればよい。このとき同じページ番号が現われても2回含めないことにする。その例は  $M(P, 4, 9)$  で説明される：

$$\begin{array}{r} t = 9 \ 8 \ 7 \ 6 \ 5 \ 4 \\ P(t) = 3 \ 4 \ 5 \ 3 \ 4 \ 1 \end{array}$$

したがって、 $M(P, 4, 9) = 3, 4, 5, 1$  である。時刻6および5のページ番号3および4はすでに求められているので捨てられる。 $M(P, c, t)$  を求めるには、 $t$  番目のページ参照から始めページ軌跡を逆にたどり、最効に出てくる  $c$  個のページ番号を求める。 $M(P, c+1, t)$  を求めるには、最初に出てくる  $c+1$  個のページ番号を求める。この定義より、 $M(P, c, t) \subset M(P, c+1, t)$  であることは容易に理解されるであろう。すなわち、 $M(P, c, t)$  に含まれるページ番号は、 $M(P, c+1, t)$  に含まれている。この包含関係は、記憶容量を増加してもページ割込みの発生回数を増加させないことを保証する。

# TLB Miss と ページウォーク

ペナルティ: 15 - 30 サイクル

For S/370 and 4<sup>KB</sup> Pages:

Machine	TLB Organization	Entries	Not-in-TLB rate
Amdahl V/6	128 × 2	256	0.3-0.4%
Amdahl V/8	256 × 2	512	(not available)
IBM 3081	128 × 1	128	1%
IBM 3033	64 × 2	128	(not available)

For VAX and 512<sup>B</sup> Pages:

VAX 11/780	64 × 2	128	4.5%
------------	--------	-----	------

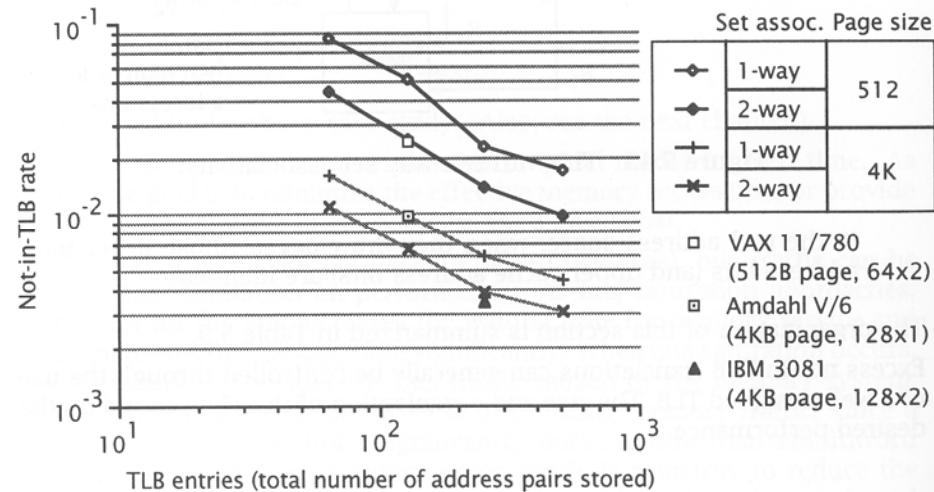


Figure 5.46 Not-in-TLB rate.

M.J.Flynn:Computer Architecture, Jones & Bartlett, 1995



# 仮想アドレス空間の共有と保護

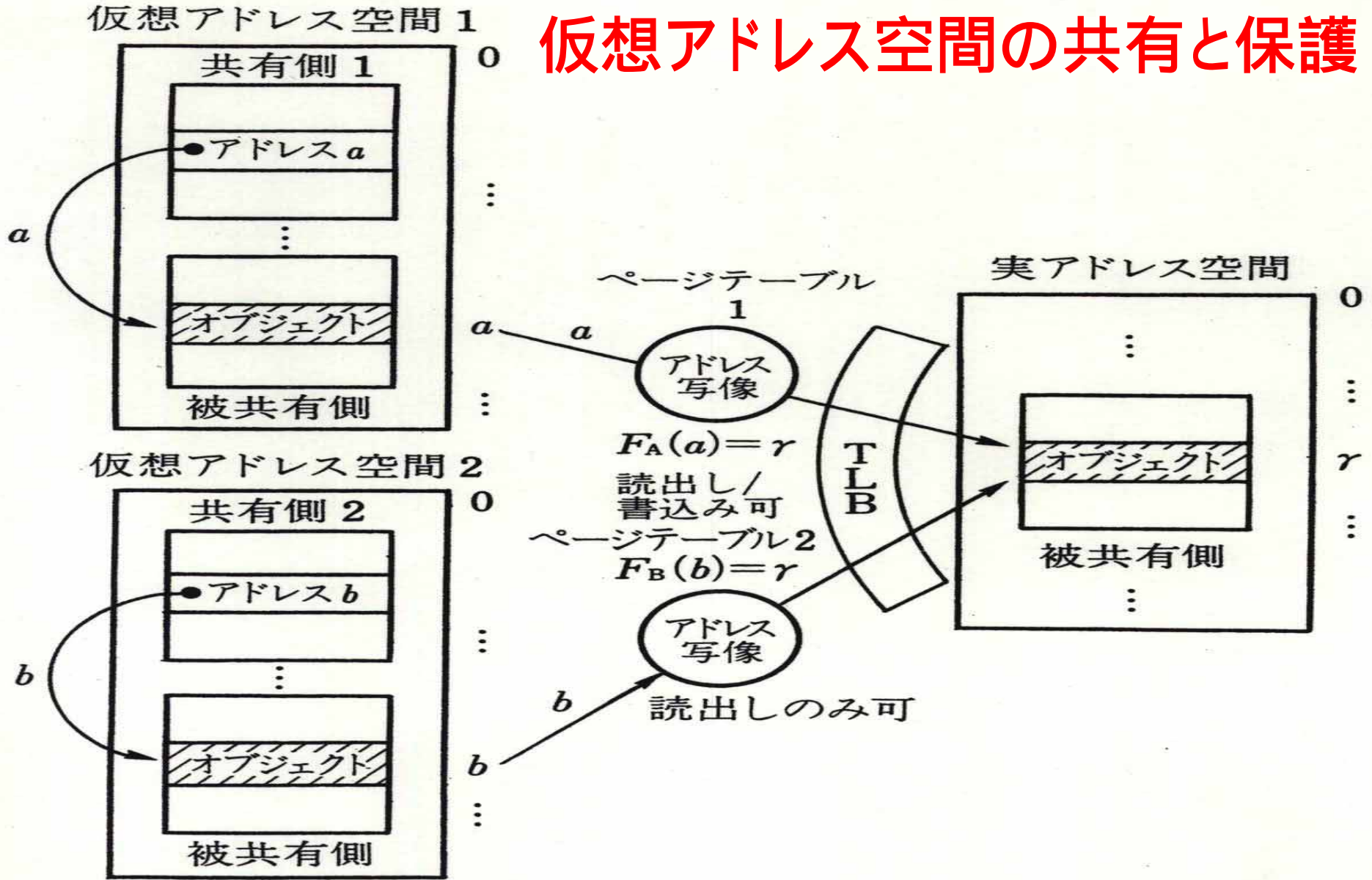


図 4.16 仮想アドレス空間の共有と保護

## 4.3 キャッシュ・メモリ

### 4.3.1 基本原理

参照の局所性を利用

on demand

メモリとの写像

ブロック単位で写像：空間局所性、64B程度

セットアソシアティブ方式

セット分割

セット数1：フルアソシアティブ方式

ロー数

ロー数1：ダイレクトマッピング(旧プロセッサ)

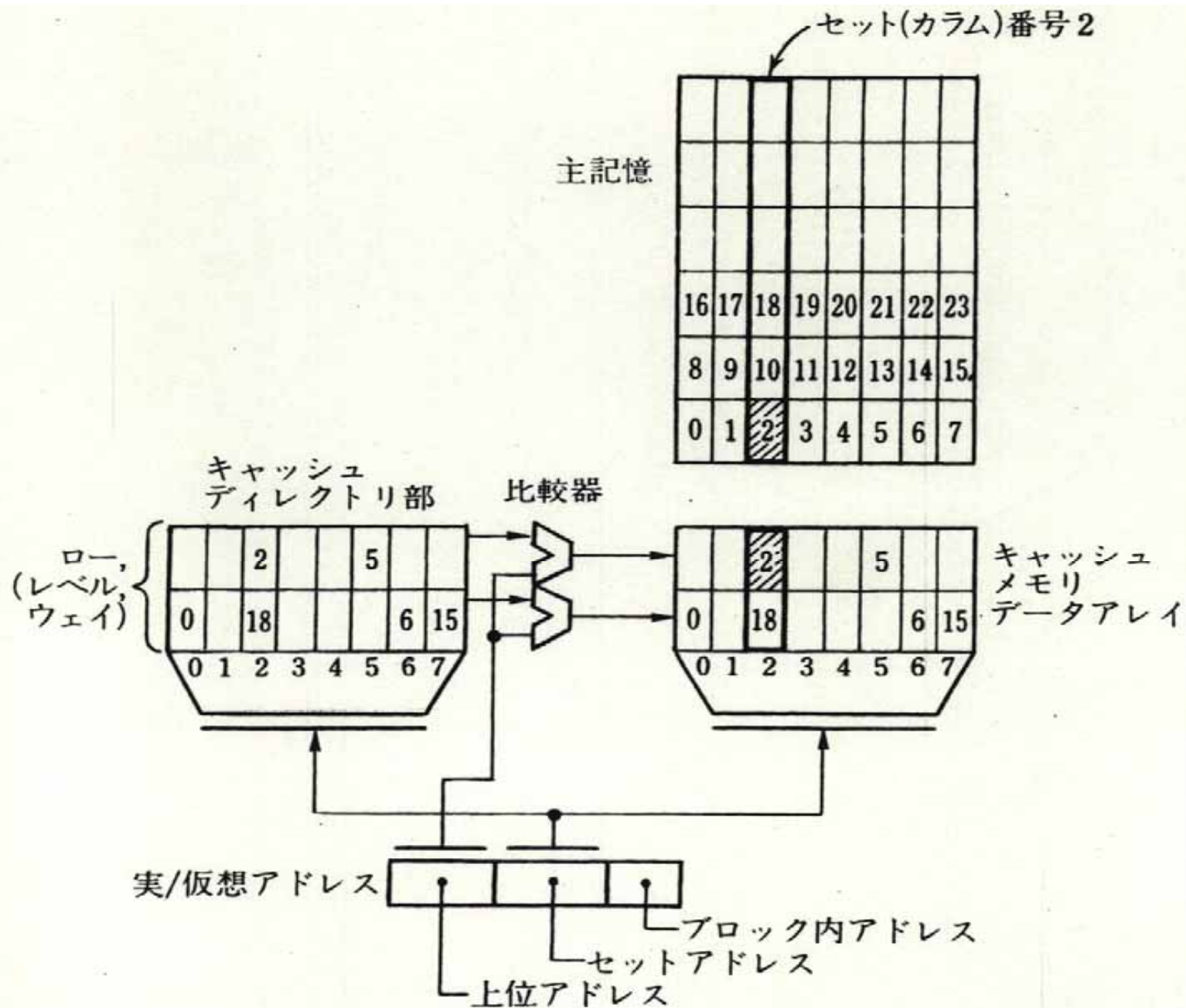


図 4.16 セットアソシアティブ方式

キャッシュメモリの容量

ブロックサイズ \* セット数 \* ロー数

キャッシュメモリの実効アクセス時間

$$T_C = T_H (1 - \text{ミス率}) + \text{ミス率} (T_H + T_{L1}) = T_H + \text{ミス率} T_{L1}$$

:ミス率、 $T_{L1}$ :メモリからの転送時間

容量大 小、 $T_H$  増大

容量: 64 kB 程度

ブロックサイズ: 空間局所性

32 ~ 64 B 程度

表 4.2 キャッシュミスヒット率の例<sup>9)</sup>(整数系, 浮動小数点系を単純合計)

(a) ロー数 2, ブロックサイズ 64B の場合のミスヒット率

キャッシュ容量	命令キャッシュ	データキャッシュ	統合キャッシュ
2 KB	0.0115	0.1708	0.0697
4 KB	0.0082	0.1400	0.0517
8 KB	0.0054	0.0942	0.0342
16 KB	0.0032	0.0454	0.0173
32 KB	0.0022	0.0289	0.0101
64 KB	0.0009	0.0215	0.0067
128 KB	0.0001	0.0164	0.0045

(b) キャッシュ容量 64 KB, 統合キャッシュの場合のミスヒット率

ブロックサイズ	ロー数 1	ロー数 2	ロー数 4	ロー数 8
16 B	0.0204	0.0160	0.0150	0.0145
32 B	0.0135	0.0096	0.0089	0.0086
64 B	0.0106	0.0067	0.0059	0.0057
128 B	0.0102	0.0054	0.0047	0.0046
256 B	0.0115	0.0050	0.0042	0.0040

命令実行数 ( $N_I$ ) : 132, 594, 094, 021, データ参照回数 : 45, 165, 001, 895

## 4.3.2 置換えアルゴリズム

各セットでLRU (Least Recently Used)

## 4.3.3 実記憶への書込み

(1) ストアスルー (store through、write through)

書込み時: メモリにも同時に書込み

(2) ストアイン (store in または write back)

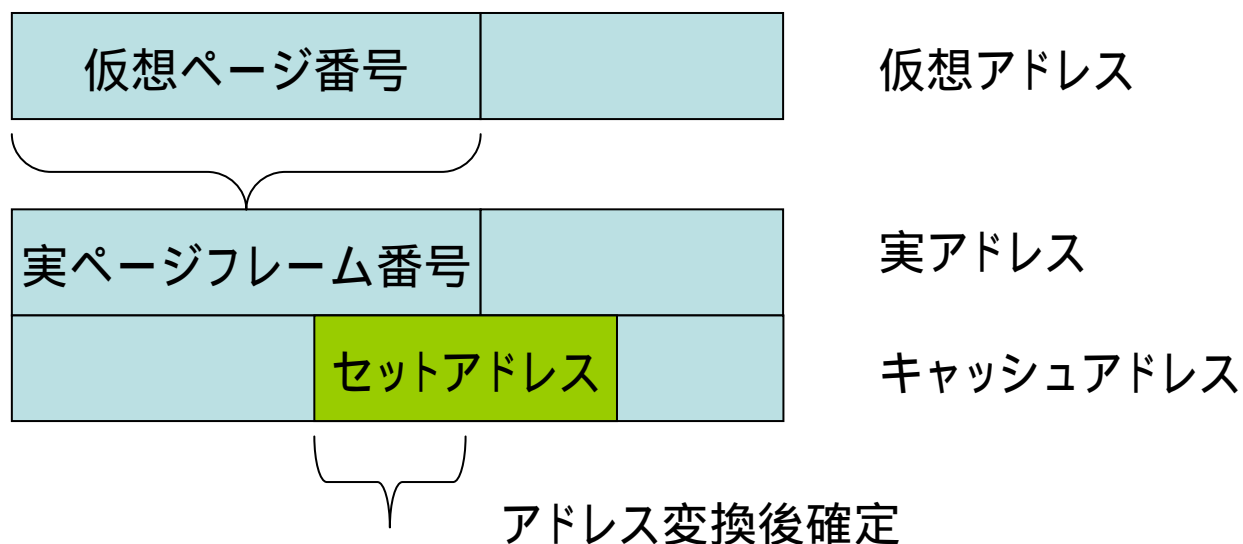
書込み時: メモリにはすぐには書かず、  
置換え時に格納

## 4.3.4 仮想アドレスキャッシュと 実アドレスキャッシュ

- ・キャッシュディレクトリへのアドレスの与え方  
仮想アドレス: Virtually Indexed  
実アドレス : Physically Indexed
- ・キャッシュディレクトリ内の情報  
仮想アドレス(ページ番号): Virtually Tagged  
実アドレス(ページフレーム番号): Physically Tagged
- ・実アドレスキャッシュ: P/P
- ・仮想アドレスキャッシュ: V/V, V/P

# (1)実アドレスキャッシュ

仮想アドレス TLB 実アドレス  
キャッシュディレクトリ データアレイ



高速化: ページ境界とセット境界同一化

シノニム問題なし

キャッシュコヒーレンス容易



# 実アドレスキャッシュの高速化

1024個のページフレーム

P/P方式 64セット × 16로우

Physically Indexed  
Physically Tagged

仮想アドレス



境界を揃える  
セットアドレス:  
仮想・実変換に  
無関係

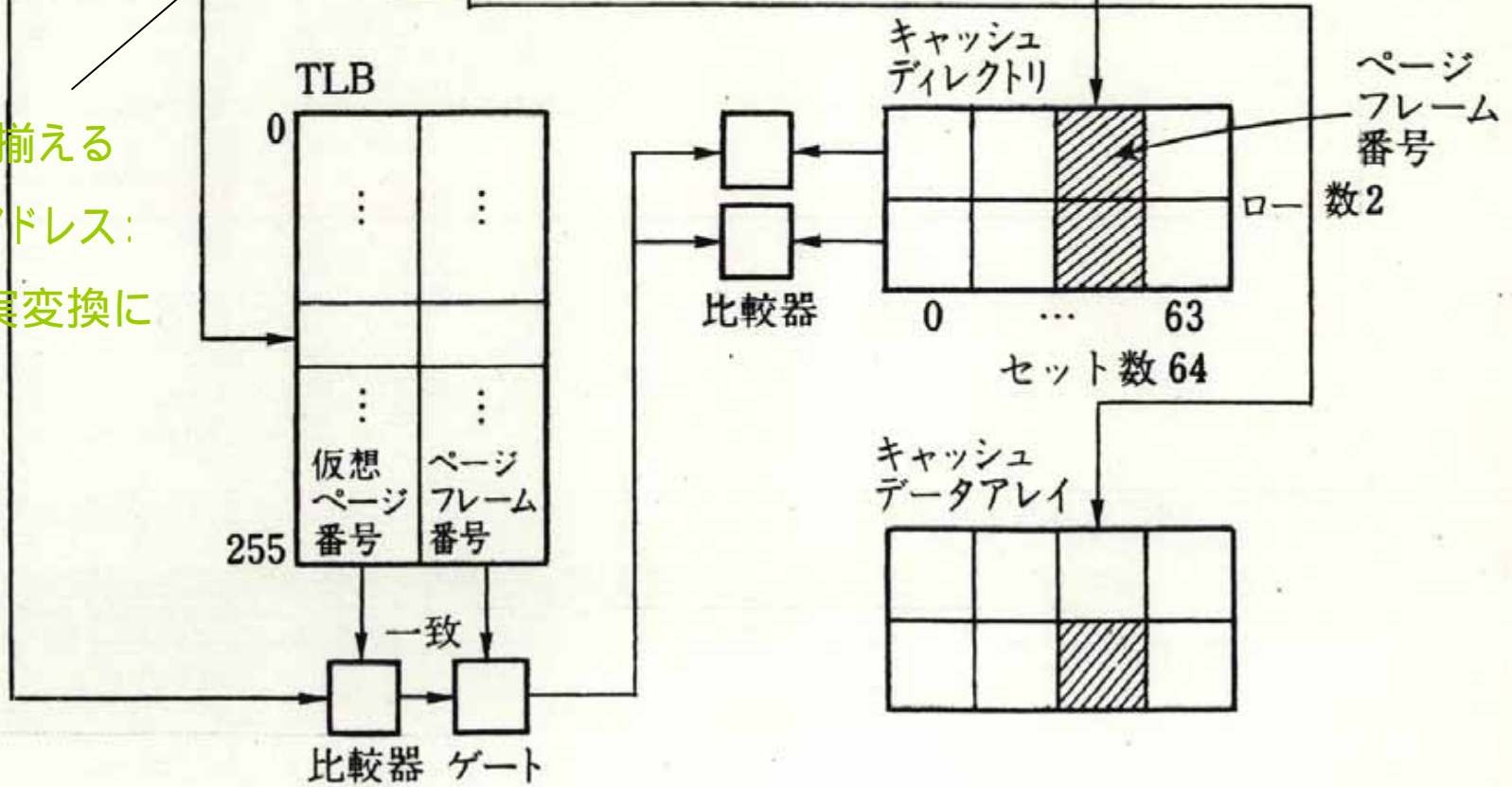


図 4.18 実アドレスキャッシュの高速化

## (2) 仮想アドレスキャッシュ (V/V方式)

高速で、

TLBを通常引く必要がない

しかし

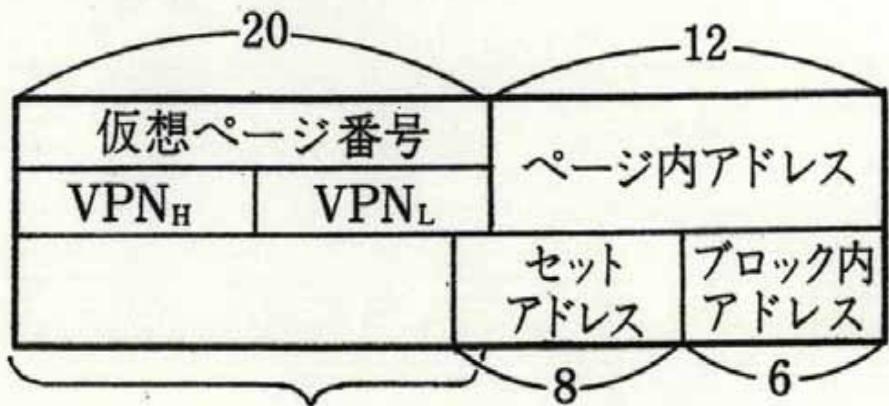
シノニム問題が生じる

仮想アドレス

# 仮想アドレスキャッシュ

## V/V方式

Virtually Indexed  
Virtually Tagged方式



キャッシュ  
ヒット:  
TLBは引  
かない

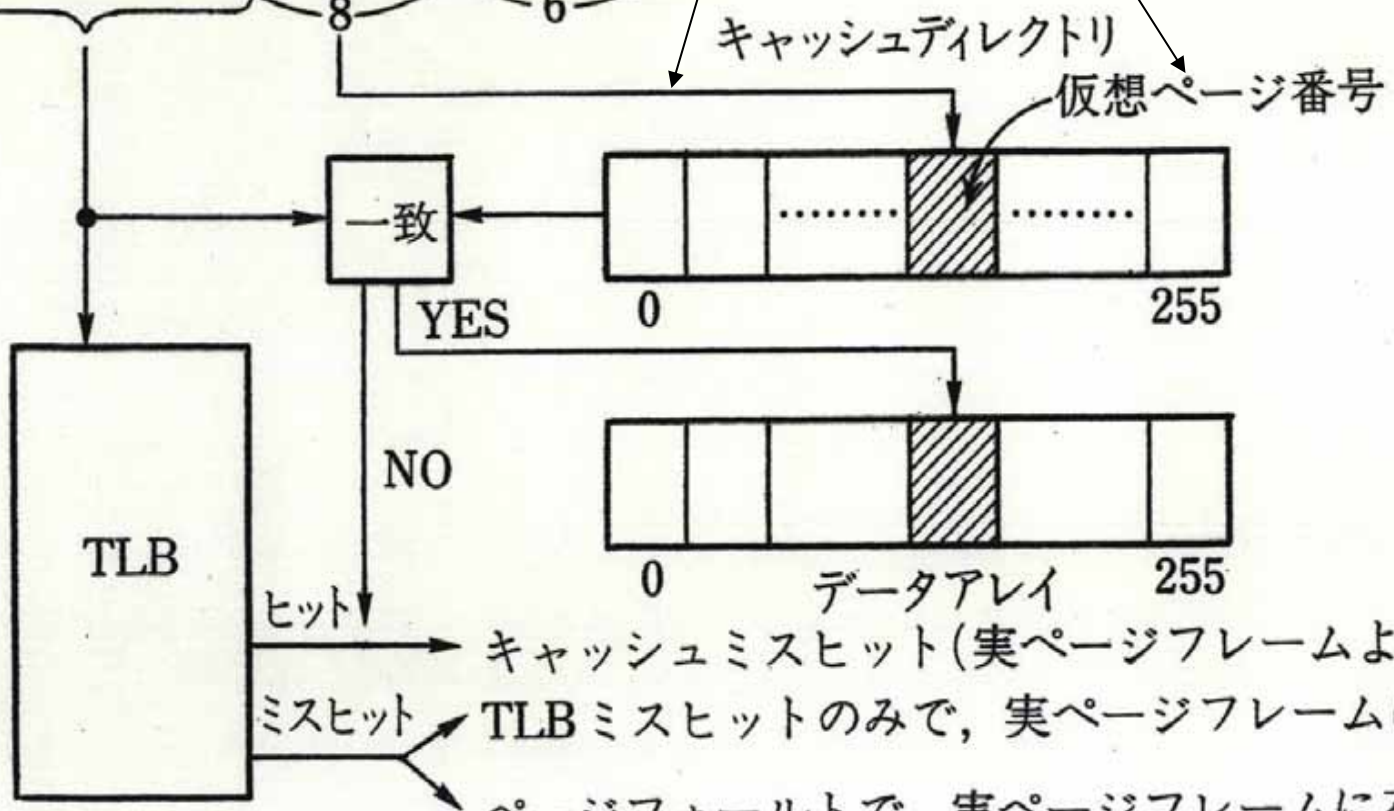
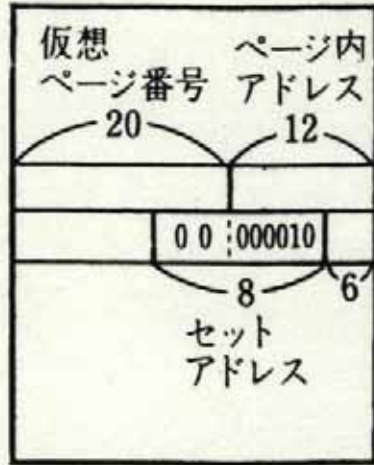


図 4.19 仮想アドレスキャッシュの構成

# シノニム問題

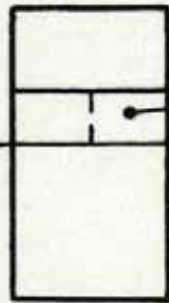
仮想アドレス空間A



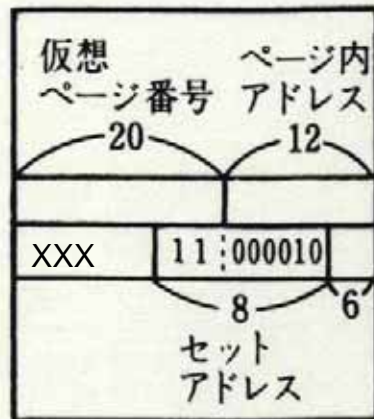
ページテーブル



ページテーブル



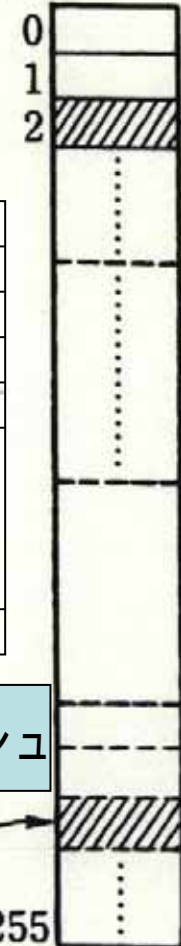
仮想アドレス空間B



仮想アドレスキャッシュ

キャッシュメモリ

セット  
番号



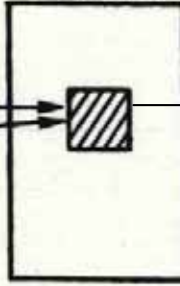
←セットアドレ  
0000000

同一

セットアドレス  
11000010

同一

実記憶



TLB

実アドレスキャッシュ

図 4.20 シノニムの発生

## シノニム問題の回避

### V/V方式

(a)共有空間を同一仮想アドレスに

設定(ソフト的):セットアドレスを含むビット

(b)全キャッシュパーージ法

プロセス切り替え時にキャッシュを無効化

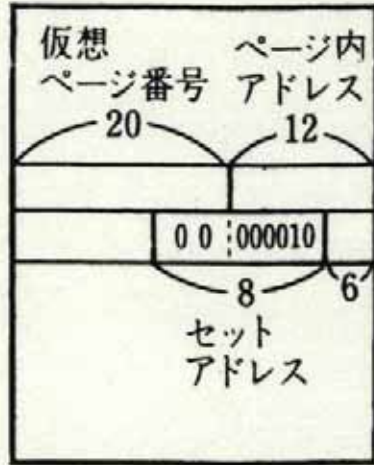
(c)逆変換バッファ法(RTB)

### V/P方式

(d)仮想実アドレスの混合型

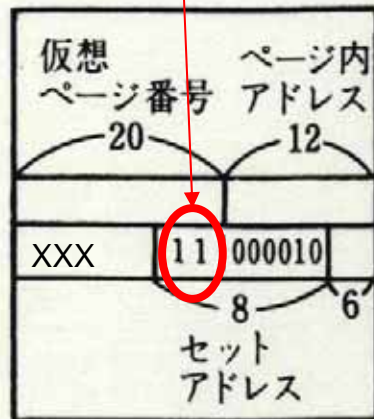
# シノニムの回避 (a), (b)

仮想アドレス空間A



回避(a): 00とする

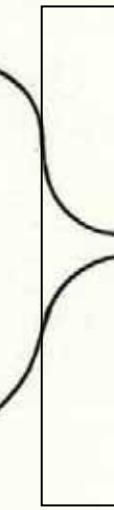
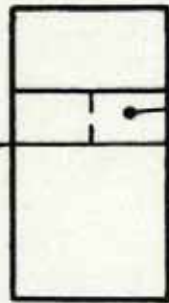
仮想アドレス空間B



ページテーブル

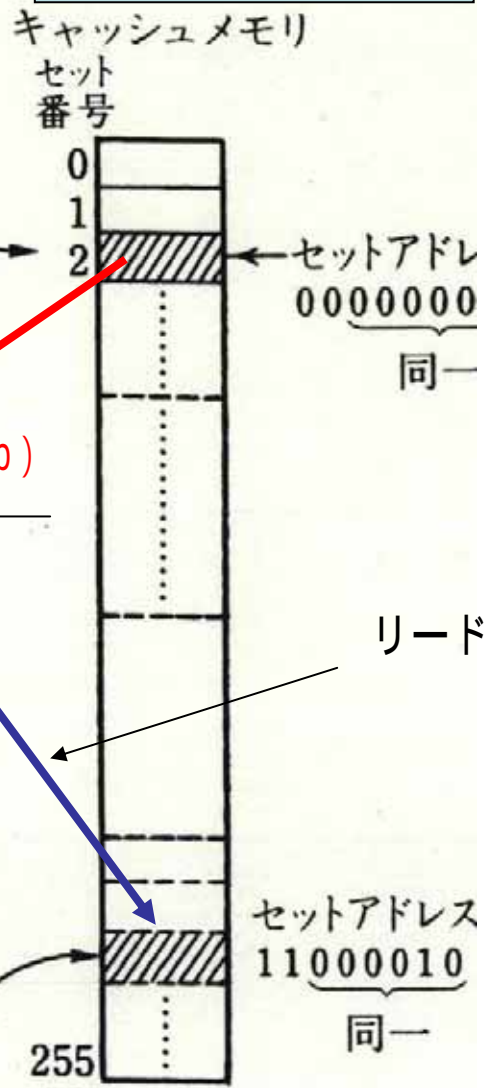


ページテーブル



無効化と書き  
戻し  
実記憶

仮想アドレスキャッシュ



リードミス

図 4.20 シノニムの発生

シノニムの回避 (c)

(4)

仮想ページA

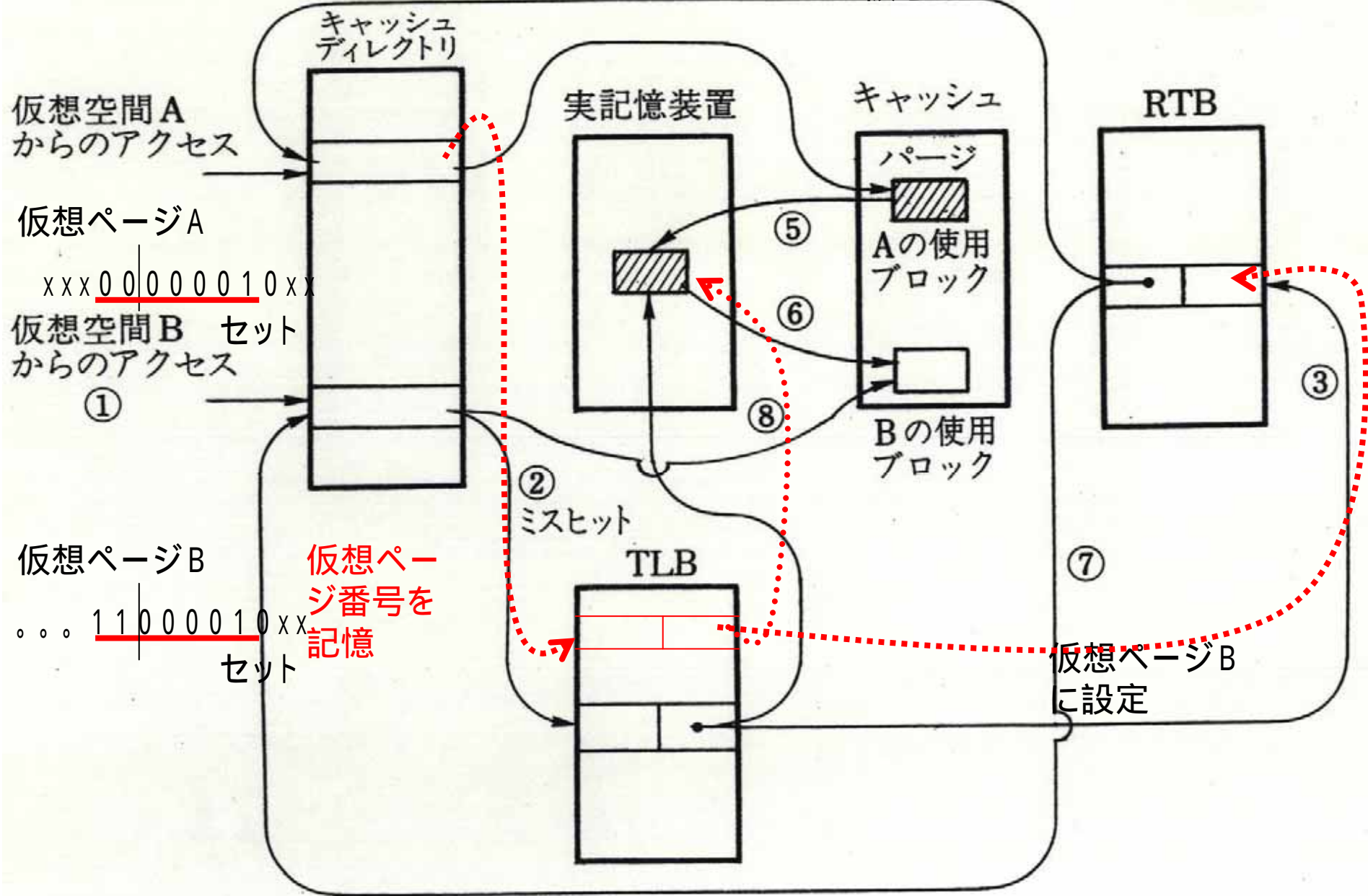
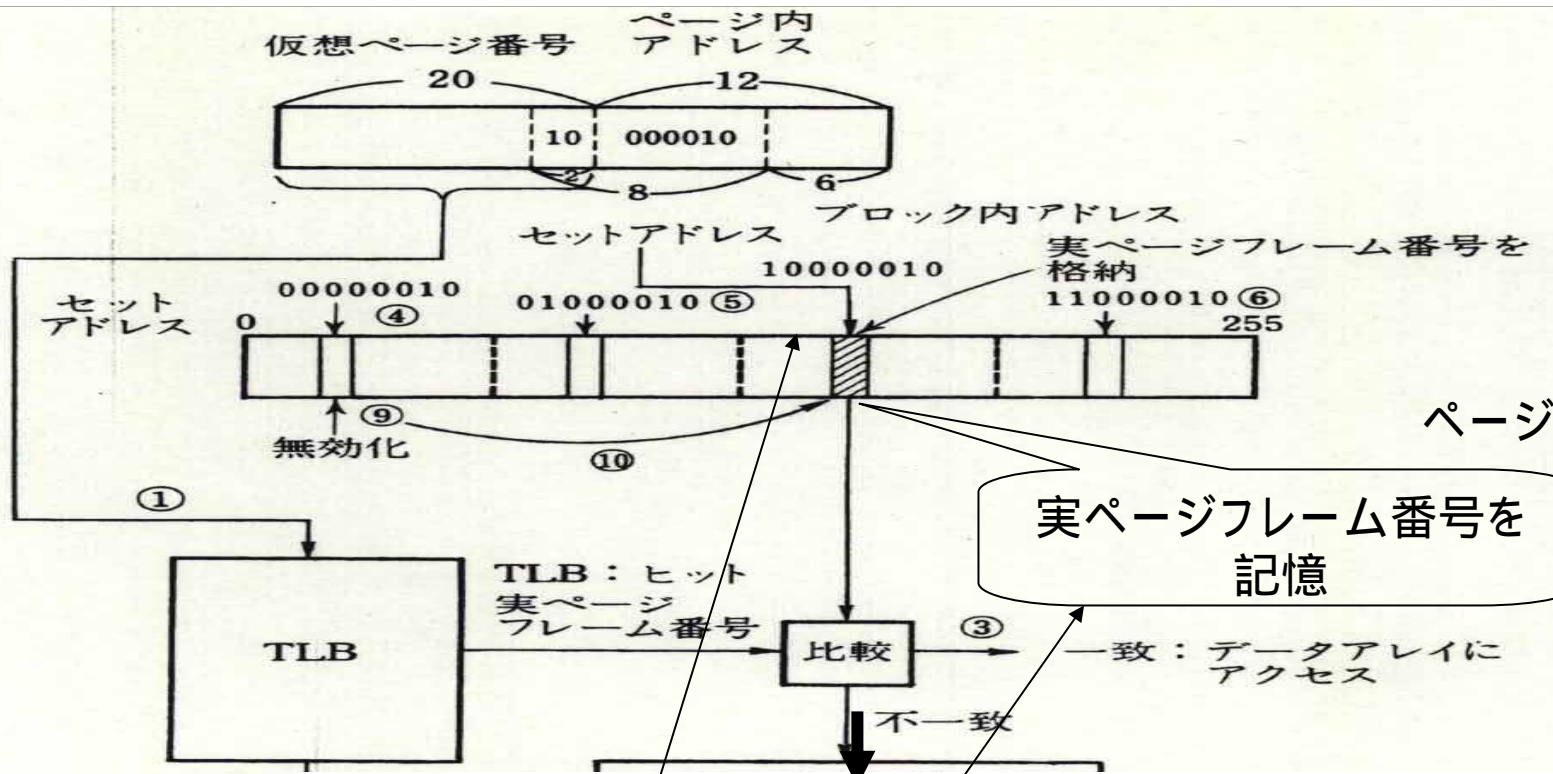


図 4.21 RTB 法によるシノニムの防止



TLBとキャッシュ  
ディレクトリを同時  
にアクセス

TLBヒット: メモリからキャッシュへ転送

TLBミスヒット: ページテーブル参照

**V/P方式**

**仮想アドレスキャッシュ**

Virtually Indexed

Physically Tagged

図 4.22 仮想アドレスキャッシュの仕組み



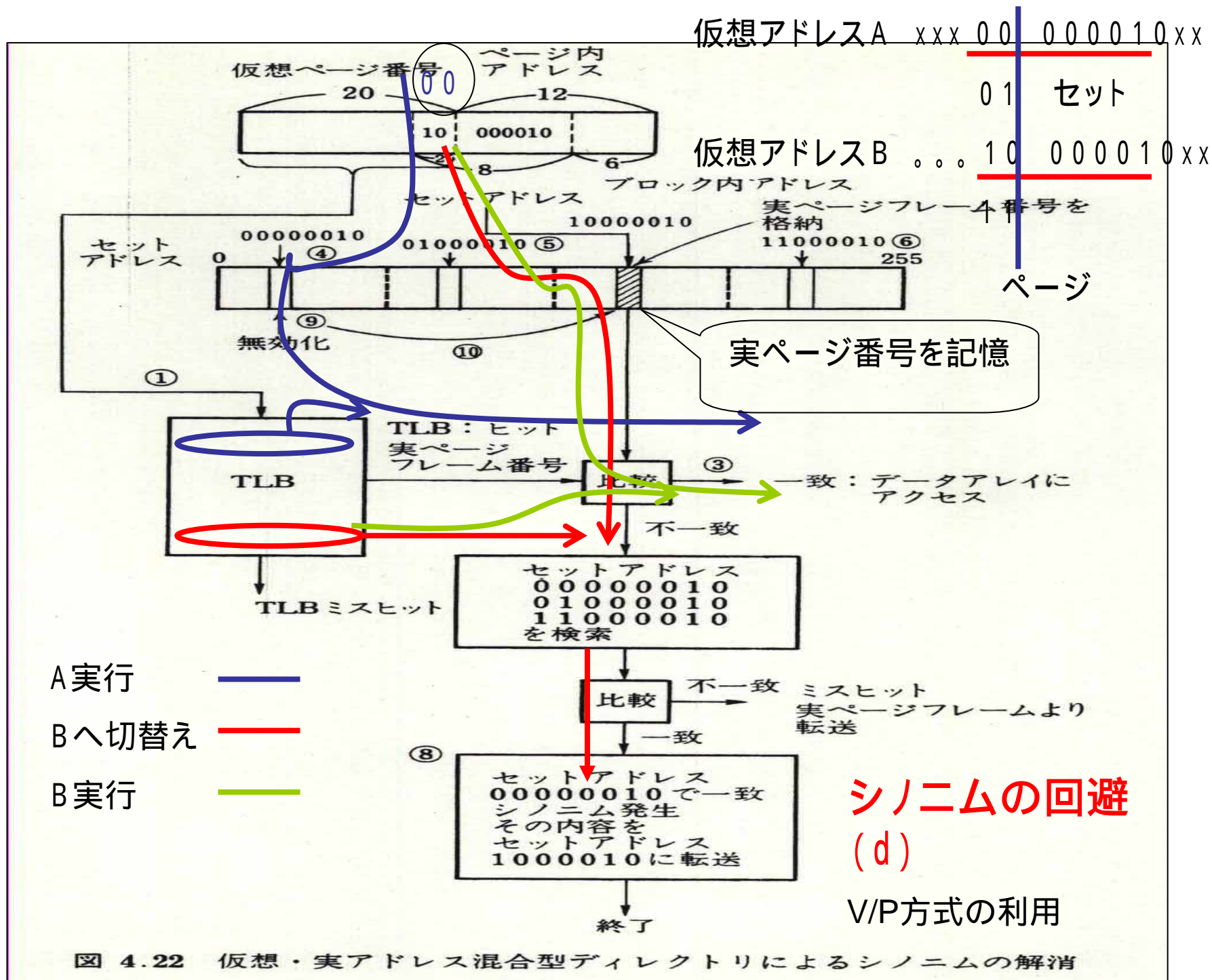


図 4.22 仮想・実アドレス混合型ディレクトリによるシノニムの解消

# ソフトウェア管理のアドレス変換 TLBは必要ない？

V/V型仮想アドレスキャッシュ

キャッシュヒット時: TLB参照の必要なし

セカンドキャッシュ: 数MB

ほとんどセカンドキャッシュでヒット

セカンドキャッシュミスヒットのとき

ページテーブルウォークをする

OS起動あるいは

ハードウェア支援

TLB: ARMで17%電力消費

TABLE 1  
Qualitative Comparison of Cache-Access/Address-Translation Mechanisms

Event	Frequency of Occurrence		Actions Performed by Hardware and Operating System per Occurrence of Event	
	I-side	D-side	TLB + Virtual cache	Software-Mgd Addr Translation
L1 hit, TLB hit	96.7%	95.8%	L1 access (w/ TLB access in parallel)	L1 access
L1 hit, TLB miss	0.01%	0.06%	L1 access + page table access + TLB reload	L1 access
L1 miss, L2 hit, TLB hit	3.2%	3.9%	L1 access + L2 access	L1 access + L2 access
L1 miss, L2 hit, TLB miss	0.03%	0.09%	L1 access + page table access + TLB reload + L2 access	L1 access + L2 access
L1 miss, L2 miss, TLB hit	0.008%	0.12%	L1 access + L2 access + memory access	L1 access + L2 access + page table access + memory access
L1 miss, L2 miss, TLB miss	0.0001%	0.0009%	L1 access + page table access + TLB reload + L2 access + memory access	L1 access + L2 access + page table access + memory access

B.Jacob,T.Mudge: Uniprocessor Virtual Memory without TLBs,  
IEEE Trans Computers,50,5,pp.482-499,2001

TABLE 5  
Level-2 Cache Misses per 1,000 Instructions

Benchmark	L2 I/I cache size	L2 icache misses per 1000 instrs	L2 dcache misses per 1000 instrs
GCC/Alpha	512K/512K	2.10	1.94
	1024K/1024K	1.37	1.88
	2048K/2048K	1.18	1.85
VORTEX/PowerPC	512K/512K	1.10	8.80
	1024K/1024K	0.83	7.96
	2048K/2048K	0.71	7.54
WINWORD/x86	512K/512K	1.01	6.61
	1024K/1024K	0.41	5.84
	2048K/2048K	0.19	5.78

L2キャッシュミス: 1000命令で5回(0.5%)

L2キャッシュミスのときのペナルティ: 10 - 40サイクル

1000命令で50 - 200サイクル

オーバヘッドCPI: 0.05 - 0.2

TLBを用いた場合と遜色がない

## 4.3.5 高速化技法

### (1) 命令キャッシュとオペランドキャッシュの分離

物理的に使用する場所が異なる

キャッシュミスの際の性能への影響

命令キャッシュには、書込み操作がない。

### 分離型キャッシュの場合

命令キャッシュ, データキャッシュ

ミスヒット率:  $I^S$ ,  $D^S$

実行命令数  $N_I$ , 内ミスヒット回数  $N_{IM}$

データ参照回数  $N_D$ , 内ミスヒット回数  $N_{DM}$

$k = N_D / N_I$ : 1命令での平均データアクセス回数

ロード命令とストア命令の出現頻度

0.34

1 命令を実行するのに必要とされる

キャッシュに関係した実行時間  $T_s$

$$N_I T_s = T_H * \text{Max}(N_I, N_D) + (N_{IM} + N_{DM}) T_L$$

より,

$$T_s = T_H + (I^S + k D^S) T_L$$

## 統合型キャッシュの場合

キャッシュメモリヒット時

命令とデータアクセスで競合

$N_I$ の命令の実行で $N_I + N_D$ 個の  
キャッシュアクセス,

優先度が低い $N_D$ 個は待たされ,

$2T_H$ 必要

ミスヒット率:

1 命令を実行するのに必要とされる

キャッシュに関係した実行時間  $T_u$ は

$$N_I T_u = T_H * N_I + T_H * N_D + (N_{IM} + N_{DM}) T_L \text{ より,}$$

$$T_u = T_H + k T_H + (N_I + N_D) / N_I$$

$$* (N_{IM} + N_{DM}) T_L / (N_I + N_D)$$

$$= T_H + k T_H + (1 + k) u T_L$$

表 4.2 キャッシュミスヒット率の例<sup>9)</sup>(整数系, 浮動小数点系を単純合計)

(a) ロー数 2, ブロックサイズ 64B の場合のミスヒット率

キャッシュ容量	命令キャッシュ	データキャッシュ	統合キャッシュ
2 KB	0.0115	0.1708	0.0697
4 KB	0.0082	0.1400	0.0517
8 KB	0.0054	0.0942	0.0342
16 KB	0.0032	0.0454	0.0173
32 KB	0.0022	0.0289	0.0101
64 KB	0.0009	0.0215	0.0067
128 KB	0.0001	0.0164	0.0045

(b) キャッシュ容量 64 KB, 統合キャッシュの場合のミスヒット率

ブロックサイズ	ロー数 1	ロー数 2	ロー数 4	ロー数 8
16 B	0.0204	0.0160	0.0150	0.0145
32 B	0.0135	0.0096	0.0089	0.0086
64 B	0.0106	0.0067	0.0059	0.0057
128 B	0.0102	0.0054	0.0047	0.0046
256 B	0.0115	0.0050	0.0042	0.0040

命令実行数 ( $N_I$ ) : 132, 594, 094, 021, データ参照回数 : 45, 165, 001, 895



# 分離型と統合型キャッシュの性能比較

TH = 1 サイクル, TL = 30 サイクル

32 KB, 32 KB 分離型

64 KB 統合型

命令パイプライン

分離型キャッシュ:

$$(0.0022 + 0.34 \times 0.0289) \times 30 = 0.36$$

統合型キャッシュ:

$$\underline{0.34} + (1 + 0.34) \times 0.0067 \times 30 = 0.61$$

大きい: 数命令同時読み出しでカバー可能

## (2) 2階層キャッシュメモリ

### 基本原理

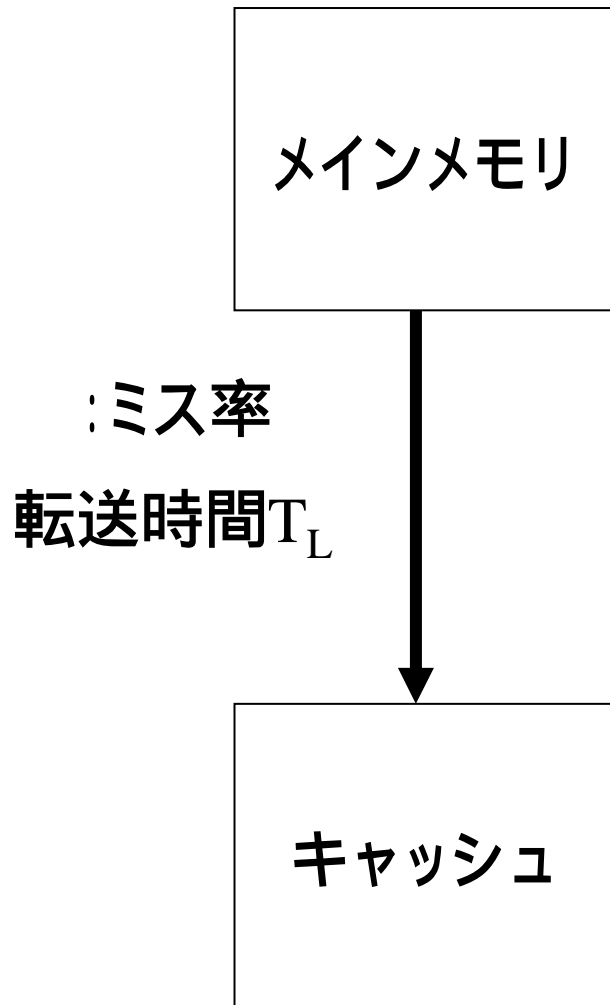
$$T_C = T_H + T_{L1} + T_{L2}$$
$$= 0.02, \quad = 0.2,$$

$T_{L1} / T_H = 5, T_{L2} / T_H = 50$  の場合

$$T_C = 1.3 T_H$$

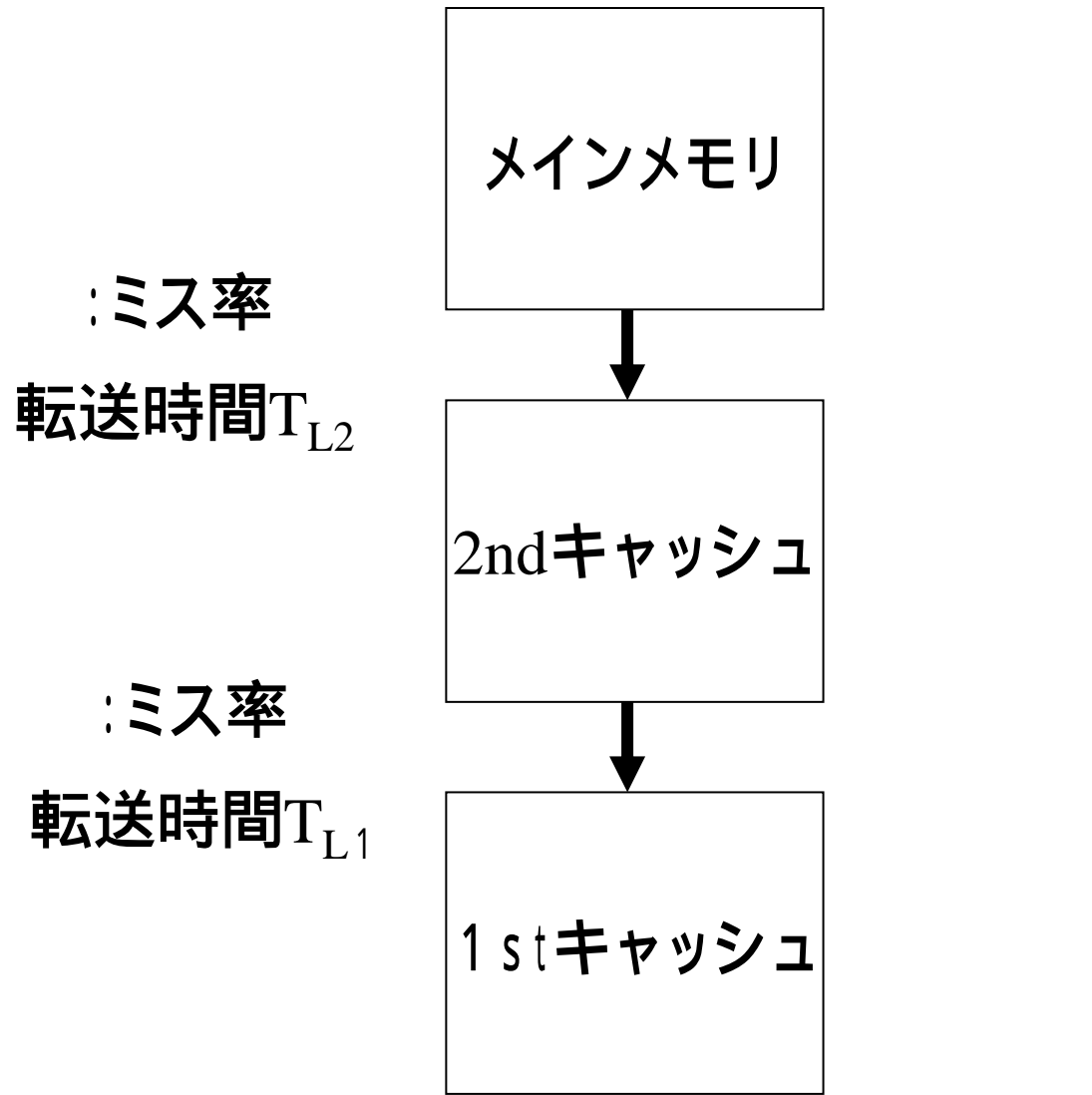
## (3) キャッシュバイパスバッファ

キャッシュ内のアクセスバイトから転送



$$T = T_H + T_L = 2T_H$$

$$= 0.02, T_H = 1 \text{ ns}, T_L = 50T_H$$



$$T = T_H + T_{L1} + T_{L2} = 1.3T_H$$

$$= 0.2, T_{L1} = 5T_H$$

# Power4の記憶階層

Table 3: Storage hierarchy organization and size

Component	Organization	Capacity per Chip	latency
L1 Instruction Cache	Direct map, 128-byte line managed as 4 32-byte sectors	128 KB (64 KB per processor)	
L1 Data Cache	2-way, 128-byte line	64 KB (32 KB per processor)	4
L2	8-way, 128-byte line	~ 1.5 MB	12
L3	8-way, 512-byte line managed as 4 128-byte sectors	32 MB 16 MB eDRAM外付け	?
Memory	---	0-16 GB	340

# Pentium4の記憶階層

L1 Data 8KB レイテンシ 2

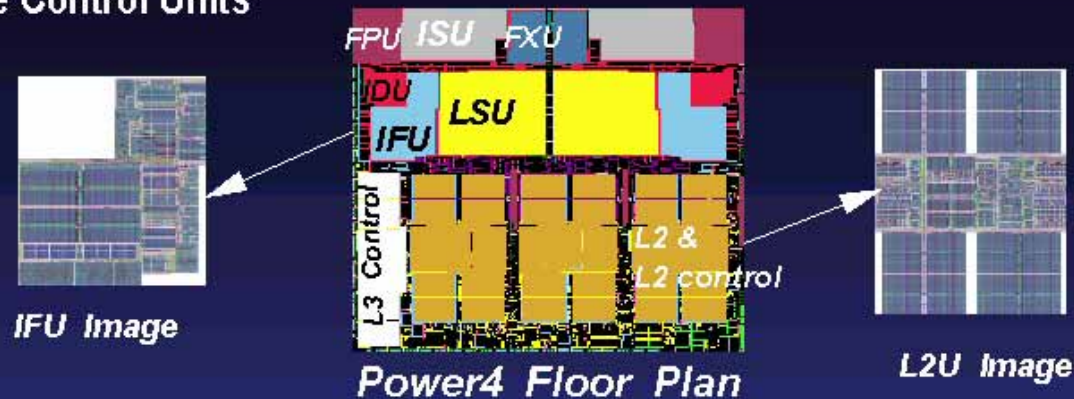
L2 256KB レイテンシ 18

**Table 1. Features of the Itanium 2 processor.**

<b>Design</b>		
Frequency	1 GHz	
Pipe stages	8 in-order	
Issue/retire	6 instructions	
Execution units	2 integer, 4 memory, 3 branch, 2 floating-point	
<b>Silicon</b>		
Technology	180 nm	
Core	40 million transistors	
L3 cache	180 million transistors	
Size	421 mm <sup>2</sup>	
<b>Caches</b>		
L1 instruction	Size	16 Kbytes
	Latency	1 cycle
	Protection	Parity
L1 data	Size	16 Kbytes
	Latency	1 cycle
	Protection	Parity
L2	Size	256 Kbytes
	Latency	5, 7, or 9+ cycles
	Protection	Parity or ECC*
L3	Size	3 Mbytes
	Latency	12+ cycles
	Protection	ECC
<b>Benchmark results</b>		
Spec CPU2000 score	810	IEEE Micro, March,2003
Spec FP2000 score	1,431	

# Power4 Processor Design

- ▶ Power4 Floor Plan and Images of the Instruction Fetch (IFU) & L2 Cache Control Units



- ▶ **Power4 Processor Chip Features**

- ▶ Two microprocessors, 3-level cache hierarchy
- ▶ >30 GB/s (>500 MHz) interconnection fabric
- ▶ 2200 signal I/O; 170M transistors
- ▶ Configured for 8-way system on MCM
- ▶ CMOS 8S2-SOI technology & copper wiring

- ▶ **Microprocessor core**

- ▶ > 1GHz clock frequency; 64KB L1 instruction cache
- ▶ 64-bit PowerPC for RS6000 & AS400 systems
- ▶ Out-of-order, speculative, 8-issue superscalar design

- ▶ **Memory sub-system**

- ▶ 1.5 MB shared L2 caches
- ▶ 32 MB off-chip L3 (on-chip control); Bandwidth > 10GB/s

Figure 2. Power5 chip (FXU = fixed-point execution unit, ISU = instruction sequencing unit, IDU = instruction decode unit, LSU = load/store unit, IFU = instruction fetch unit, FPU = floating-point unit, and MC = memory controller).

#### (4) ノンブロッキングキャッシュ

先行命令でキャッシュミスでも

後続命令どんどん実行

パイプラインバブル減少

データの先読みによる遅延時間

(レイテンシ)減少: **コンパイラによる**

**キャッシュ制御VS オンデマンド制御**

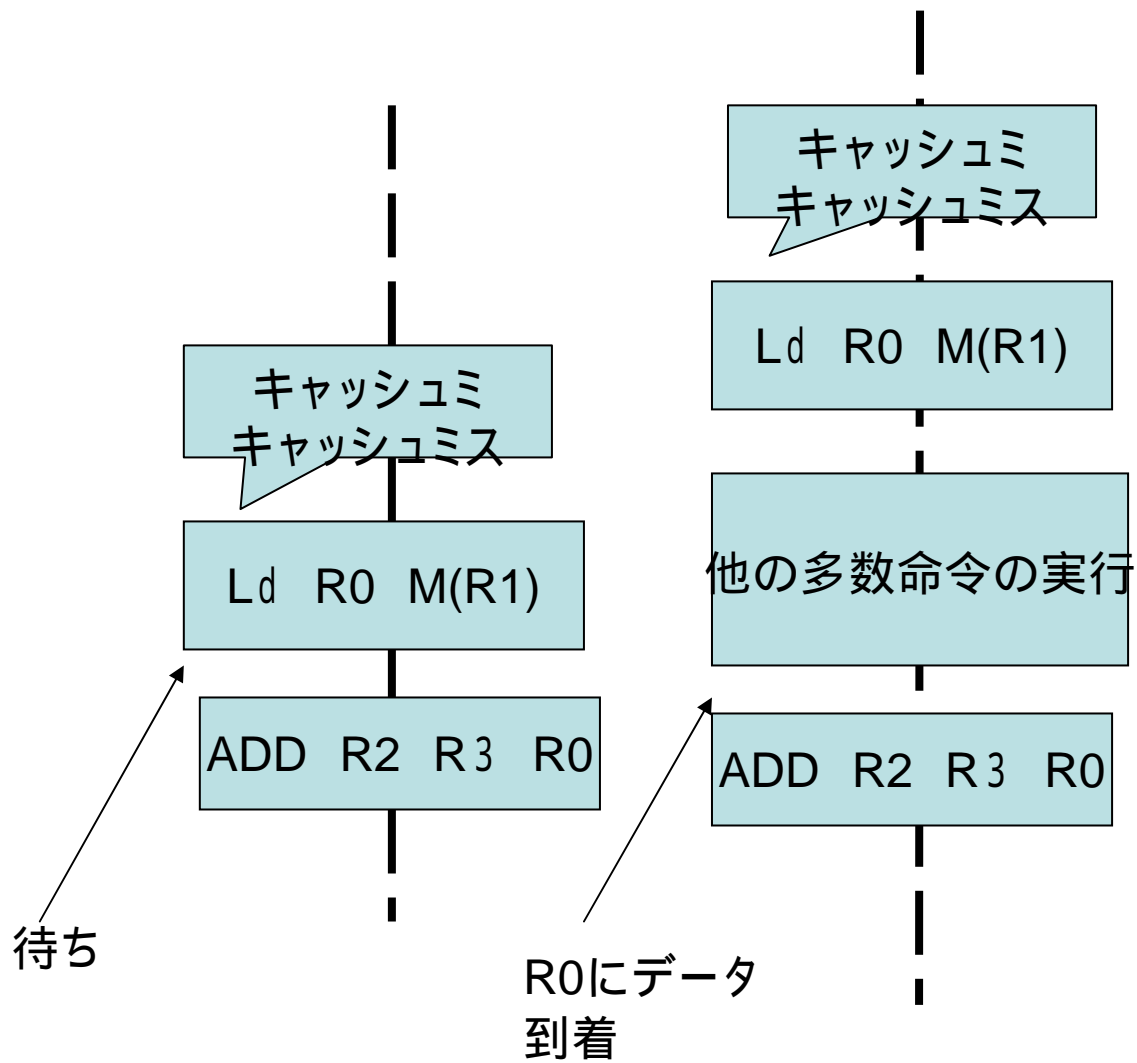
#### (5) ストアバッファ

ストアスルーでメモリ書込みを待たないで

先実行: 置いてきぼり制御

#### (6) 命令バッファ

命令キャッシュから数命令同時フェッチ





## 4.3.6 キャッシュメモリの有効性

1024×1024の2次元配列AとベクトルXの積B

$$B_i = \sum_j A_{ij} X_j$$

各要素データは8B

キャッシュブロックのサイズは64B

(すなわち8要素の格納が可能)

キャッシュ容量: 18KB

## 単純なプログラム

```
DO 10 I=1,1024  
DO 10 J=1,1024  
B(I) = A(I, J) * X(J)
```

```
10 CONTINUE
```

## 改良プログラム

```
DO 10 I=1,1017,8  
DO 10 K=1,4  
DO 10 J=256(K-1)+1,256K  
B(I) = B(I) + A(I, J) * X(J)  
B(I+1) = B(I+1) +  
A(I+1, J) * X(J)  
.....  
B(I+7) = B(I+7) +  
A(I+7, J) * X(J)
```

```
10 CONTINUE
```

タイリング法

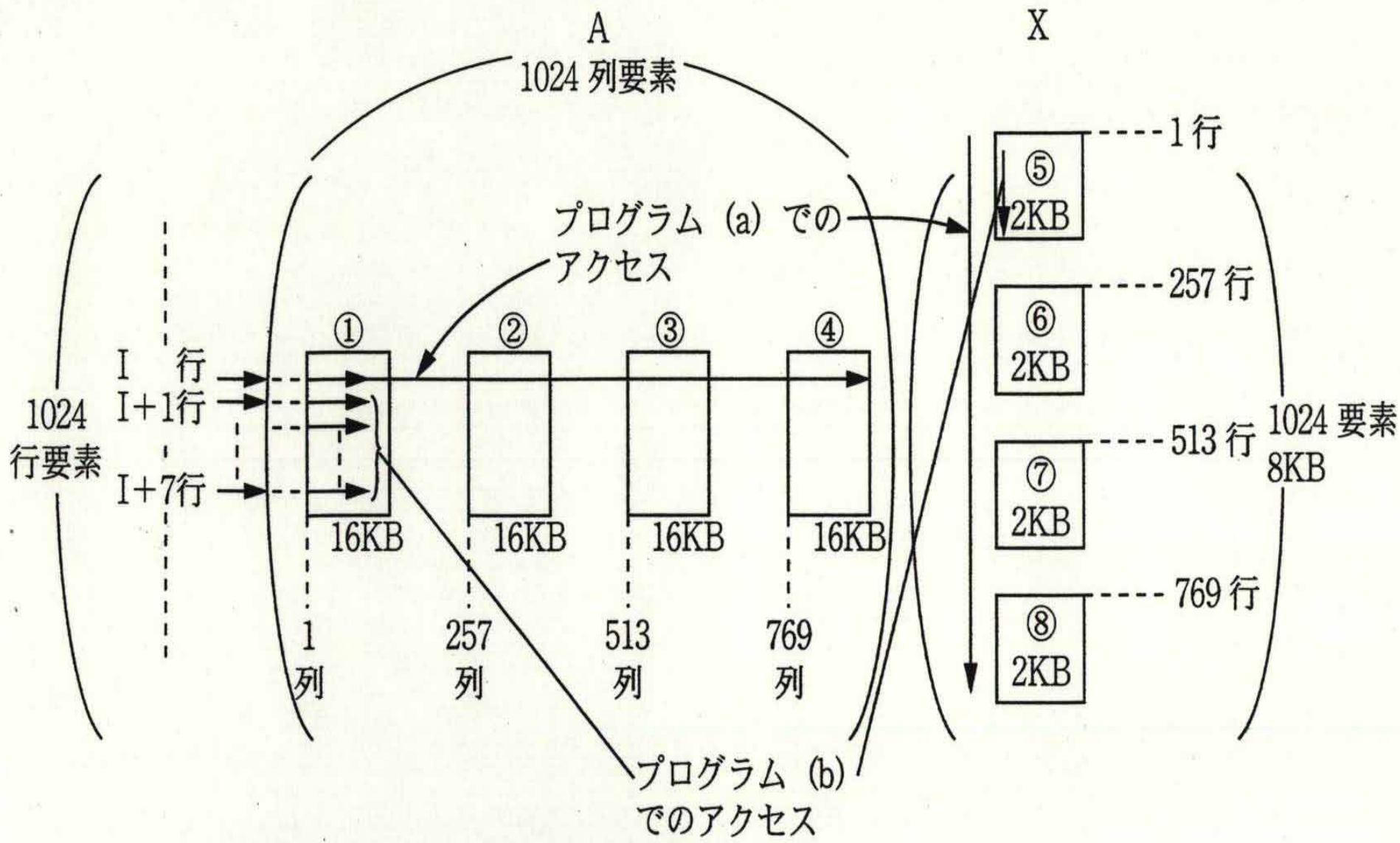


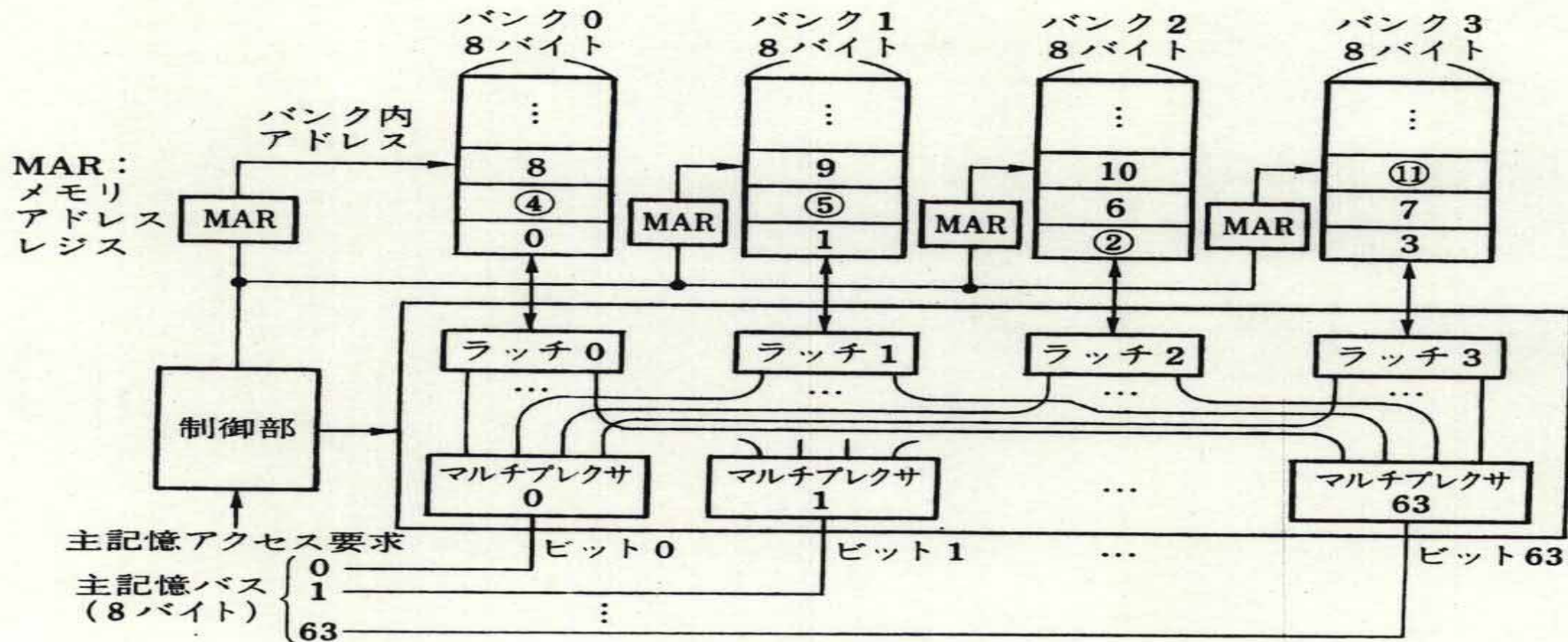
図 4.23 行列ベクトル積

## 4.6 主記憶装置

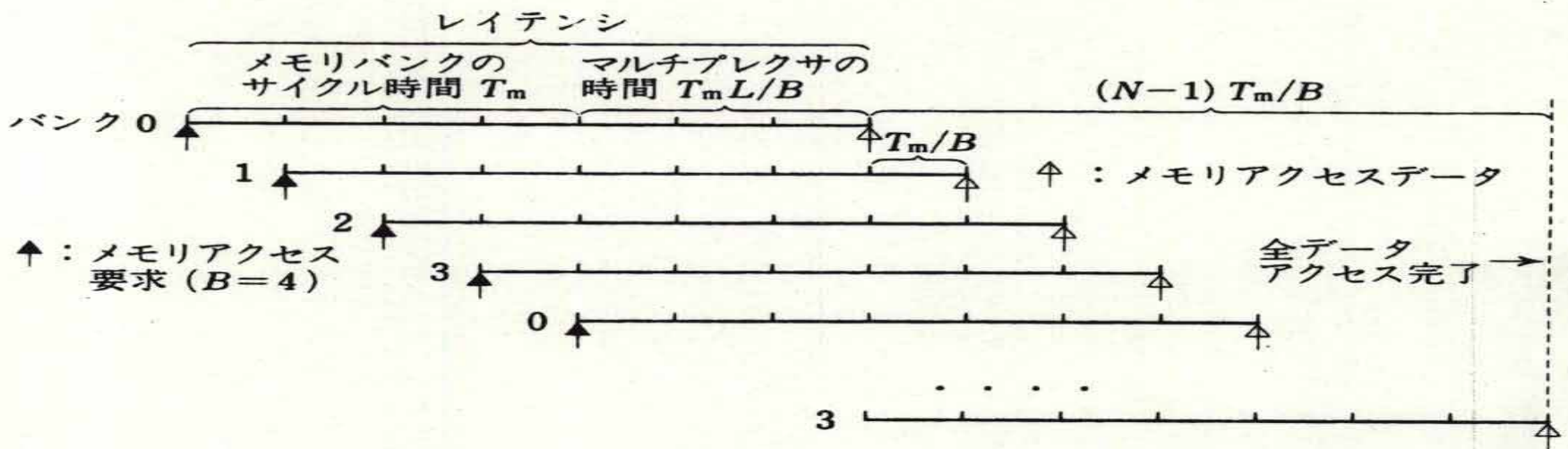
### インターリーブ

ストアスルーの時: 同時書込み可能で  
高速化

ストアインの時: メモリバス幅が  
太ければよい  
ブロック単位の転送



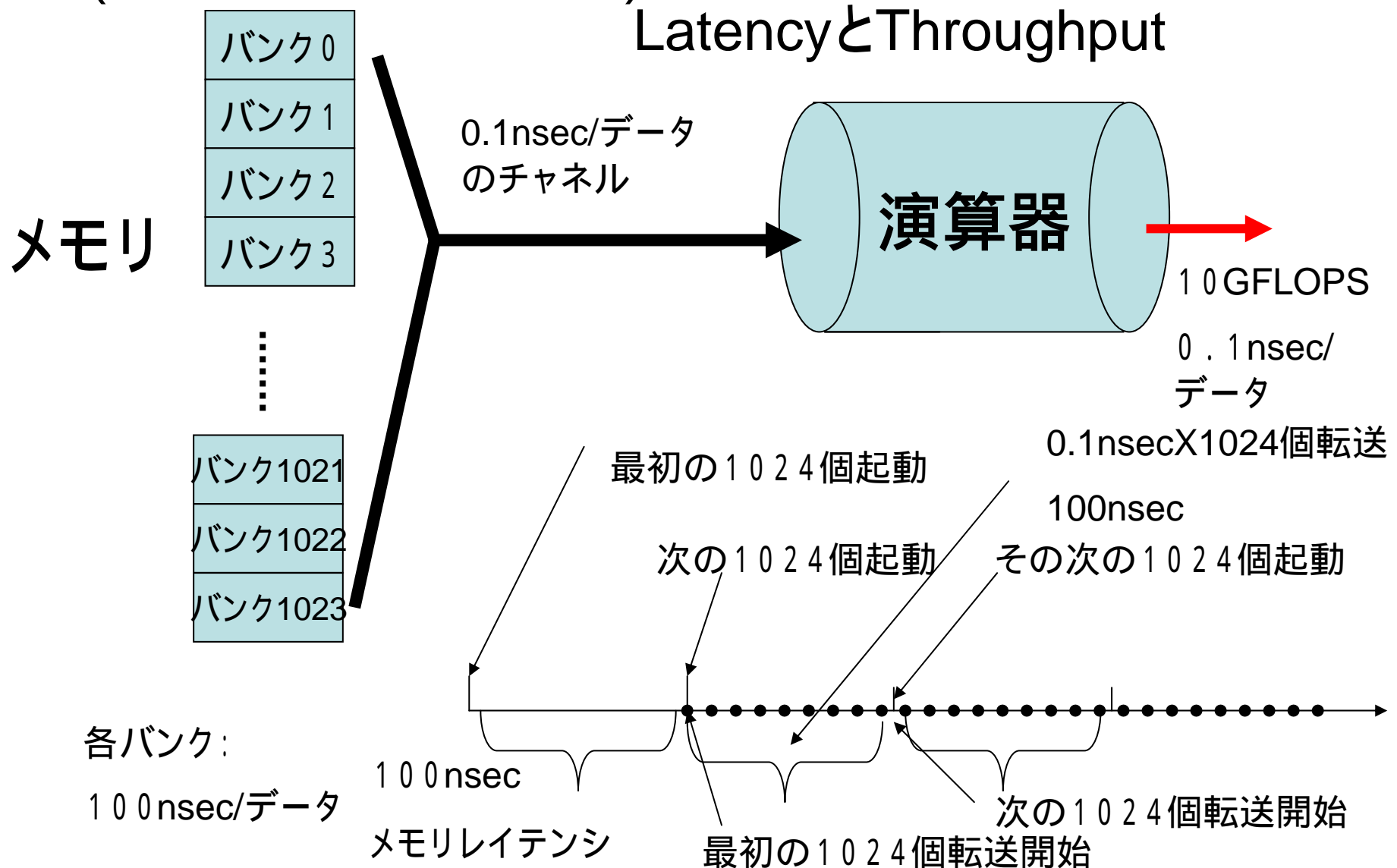
(a) メモリインタリーブ (4ウェイ)



(b) レイテンシとメモリバンド幅

図 4.30 主記憶インタリーブ方式

# スーパーコンピュータ: 1024バンク構成 (ベクトルプロセッサ)



# スキュードメモリ(直交メモリ化)

バンク内アドレス

3	a <sub>41</sub>	a <sub>42</sub>	a <sub>43</sub>	a <sub>44</sub>
2	a <sub>31</sub>	a <sub>32</sub>	a <sub>33</sub>	a <sub>34</sub>
1	a <sub>21</sub>	a <sub>22</sub>	a <sub>23</sub>	a <sub>24</sub>
0	a <sub>11</sub>	a <sub>12</sub>	a <sub>13</sub>	a <sub>14</sub>

メモリバンク

(a)

バンク内アドレス

4	a <sub>42</sub>	a <sub>43</sub>	a <sub>44</sub>	a <sub>45</sub>
3	a <sub>33</sub>	a <sub>34</sub>	a <sub>35</sub>	a <sub>41</sub>
2	a <sub>24</sub>	a <sub>25</sub>	a <sub>31</sub>	a <sub>32</sub>
1	a <sub>15</sub>	a <sub>21</sub>	a <sub>22</sub>	a <sub>23</sub>
0	a <sub>11</sub>	a <sub>12</sub>	a <sub>13</sub>	a <sub>14</sub>

メモリバンク

(b)

図 4.33 スキュードメモリ方式