

コンピュータアーキテクチャ (6)

坂井 修一

東京大学大学院 情報理工学系研究科 電子情報学専攻
東京大学 工学部 電子情報工学科 / 電気電子工学科

- ・ はじめに
- ・ 仮想記憶

はじめに

- 本講義の目的
 - コンピュータアーキテクチャの基本を学ぶ
- 時間・場所
 - 火曜日 8:40 - 10:10 工学部2号館241
- ホームページ（ダウンロード可能）
 - url: <http://www.mtl.t.u-tokyo.ac.jp/~sakai/hard/>
- 教科書
 - 坂井修一『コンピュータアーキテクチャ』（コロナ社、電子情報レクチャーシリーズC-9）
 - 坂井修一『実践 コンピュータアーキテクチャ』（コロナ社）
 - 教科書通りやります
- 参考書
 - D. Patterson and J. Hennessy, Computer Organization & Design, 3rd Ed. (邦訳『コンピュータの構成と設計』(第3版)上下(日系BP))
 - 馬場敬信『コンピュータアーキテクチャ』(改訂2版)、オーム社
 - 富田眞治『コンピュータアーキテクチャ I』、丸善
- 予備知識： 論理回路
 - 坂井修一『論理回路入門』、培風館
- 成績
 - 試験＋レポート＋出席

講義の概要と予定 (1 / 2)

1. コンピュータアーキテクチャ入門

デジタルな表現、負の数、実数、加算器、ALU, フリップフロップ、レジスタ、計算のサイクル

2. データの流れと制御の流れ

主記憶装置、メモリの構成と分類、レジスタファイル、命令、命令実行の仕組み、実行サイクル、算術論理演算命令、シーケンサ、条件分岐命令

3. 命令セットアーキテクチャ

操作とオペランド、命令の表現形式、アセンブリ言語、命令セット、算術論理演算命令、データ移動命令、分岐命令、アドレッシング、サブルーチン、RISCとCISC

4. パイプライン処理 (1)

パイプラインの原理、命令パイプライン、オーバヘッド、構造ハザード、データハザード、制御ハザード

5. パイプライン処理 (2)

フォワーディング、遅延分岐、分岐予測、命令スケジューリング

6. キャッシュ

記憶階層と局所性、透過性、キャッシュ、ライトスルーとライトバック、ダイレクトマップ型、フルアソシアティブ型、セットアソシアティブ型、キャッシュミス

7. 仮想記憶

仮想記憶、ページフォールト、TLB、物理アドレスキャッシュ、仮想アドレスキャッシュ、メモリアクセス機構

講義の概要と予定 (2 / 2)

8. 基本CPUの設計

デジタル回路の入力、Verilog HDL、シミュレーションによる動作検証、アセンブラ、基本プロセッサの設計、基本プロセッサのシミュレーションによる検証

9. 命令レベル並列処理(1)

並列処理、並列処理パイプライン、VLIW、スーパースカラ、並列処理とハザード

10. 命令レベル並列処理(2)

静的最適化、ループアンローリング、ソフトウェアパイプラインニング、トレーススケジューリング

11. アウトオブオーダー処理

インオーダーとアウトオブオーダー、フロー依存、逆依存、出力依存、命令ウィンドウ、リザーベーションステーション、レジスタリネーミング、マッピングテーブル、リオーダーバッファ、プロセッサの性能

12. 入出力と周辺装置

周辺装置、ディスプレイ、二次記憶装置、ハードウェアインタフェース、割り込みとポーリング、アービタ、DMA、例外処理

試験: 7月

8. 仮想記憶

■ 内容

– 仮想記憶

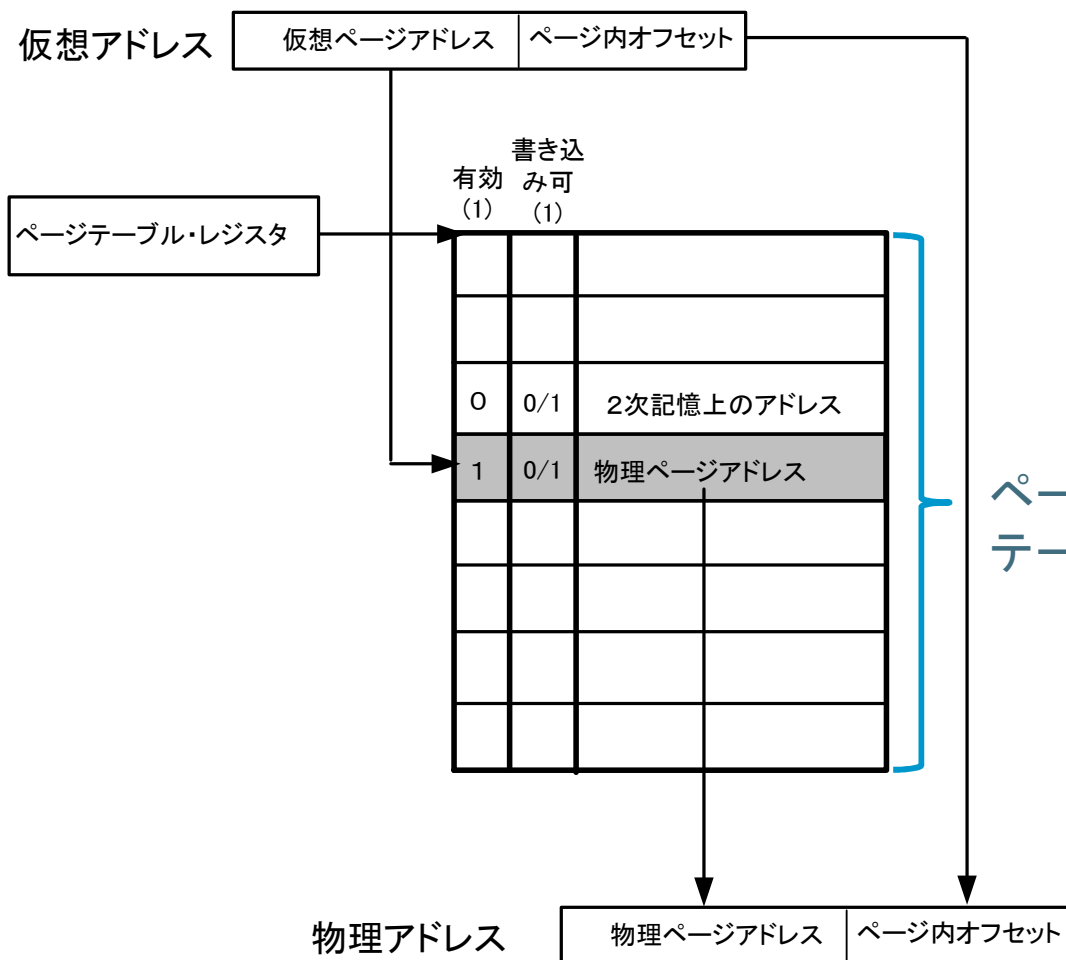
- ・ 仮想記憶とはなにか
- ・ 仮想記憶の構成

– 透過性と互換性

仮想記憶とはなにか

- 仮想記憶 = 主記憶以下の階層化
 - 主記憶と二次記憶のメモリ階層を「**巨大な主記憶**」として使えるように**透過性**をもたせたもの
 - ・ 主記憶よりも大きな容量のメモリがあるものとしてプログラムが書けるようになる
 - ・ 複数のプログラムが一つの物理記憶を安全に分ち合って使えるようになる
- 仮想記憶の動作原理
 - アドレス変換
 - ・ 仮想アドレス(virtual address) ⇒ 物理アドレス(physical address)
 - スワップ
 - ・ 2次記憶(HDD)のデータ ⇔ 主記憶のデータ

仮想記憶の構成



- ページ
 - データ移動の単位。
通常数KB =>最近は数MBのものも(ラージページ)
- アドレス変換
 - ページテーブル
- スワップ
 - フルアソシタティブ、ライトバック

ページフォールト

■ ページフォールト

ページが主記憶に入っていない状態

= ページテーブルで有効ビットが0のとき

■ ページフォールトが起こったとき

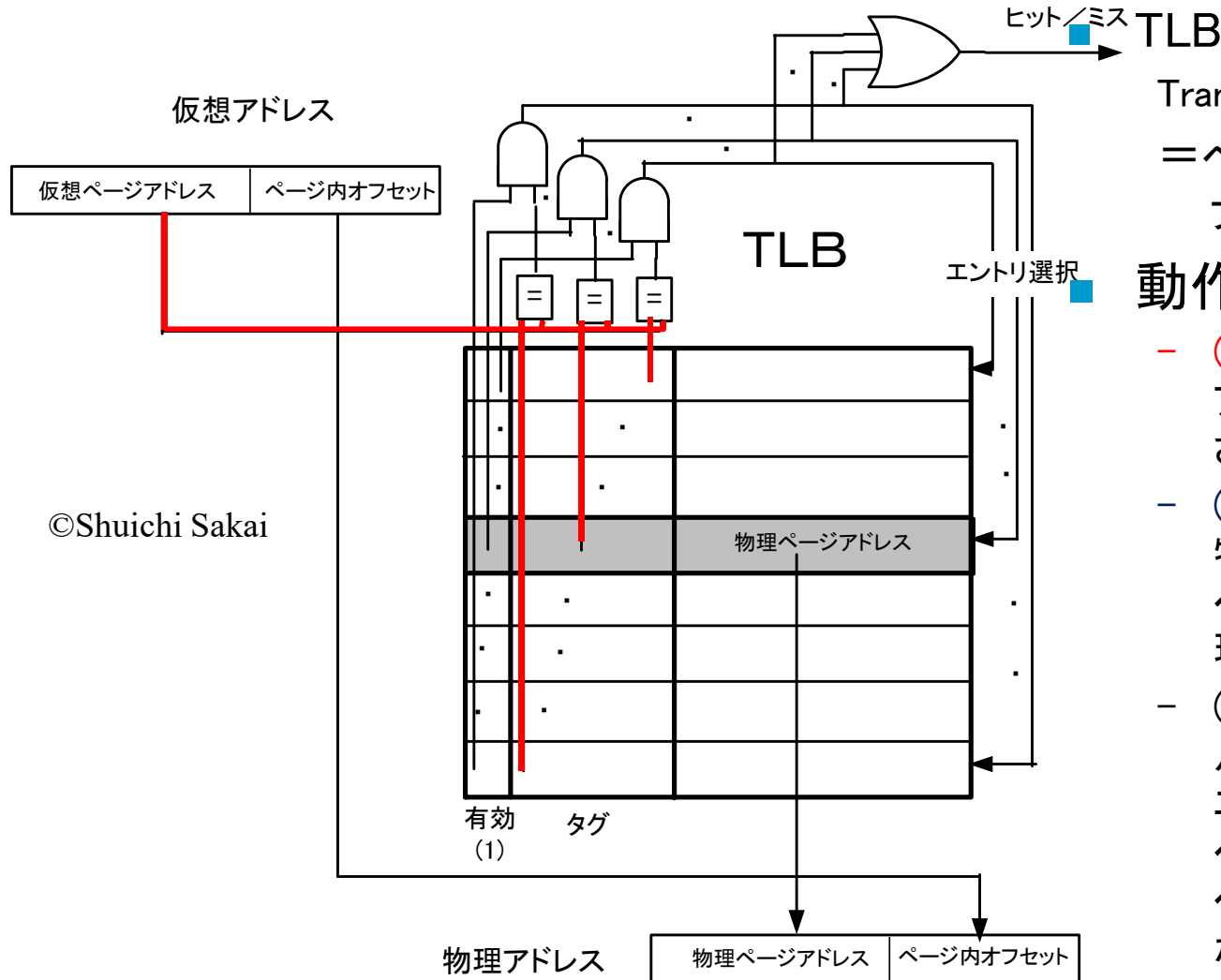
(1) CPUの処理を中断する ⇒ 例外処理

(2) テーブルのエントリに入っている二次記憶上のアドレスから主記憶の空いている場所にページをコピー

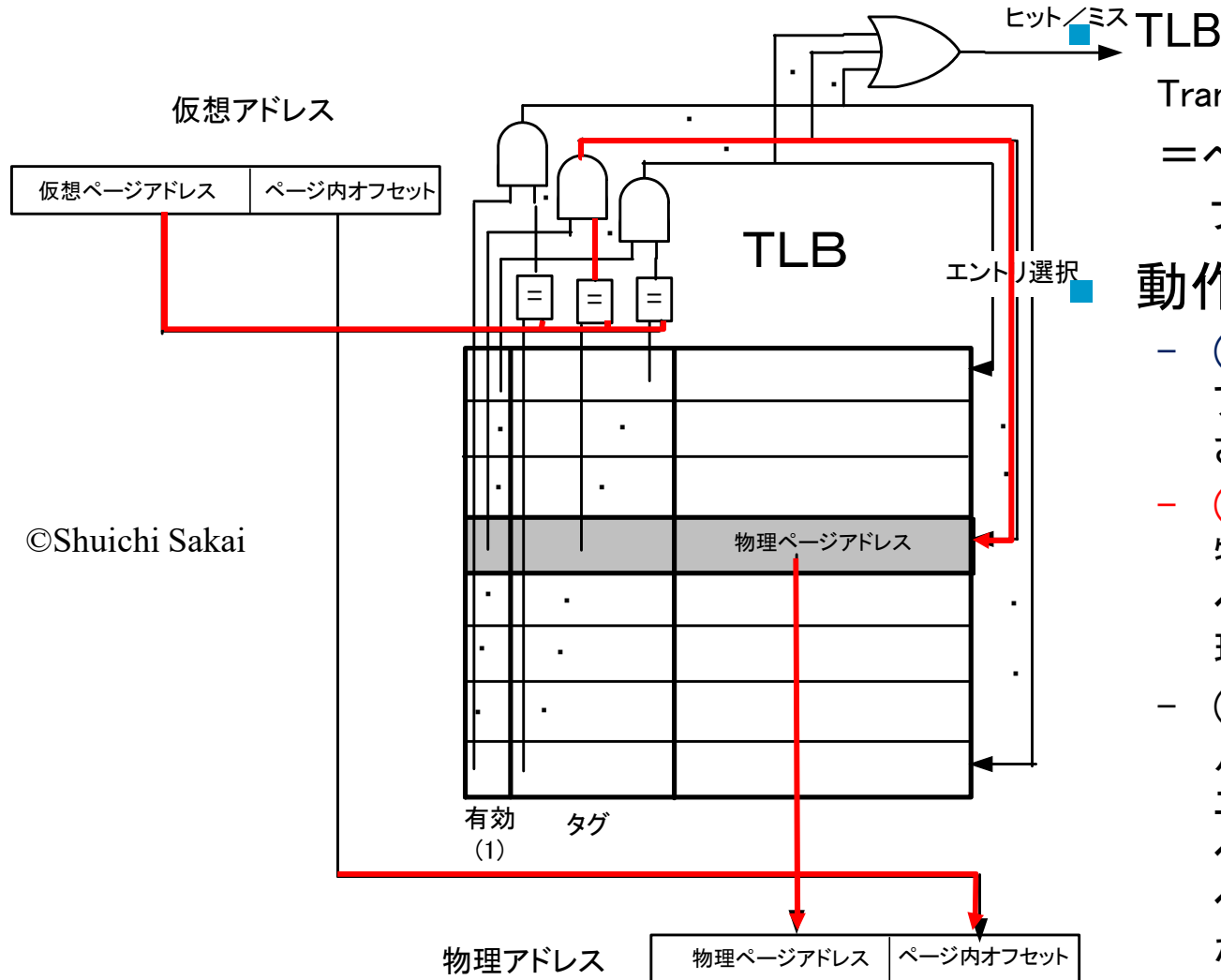
(空いている場所がなければ追い出す)

(3) ページテーブルのエントリに物理ページアドレスを書き込み、有効ビットを1にする

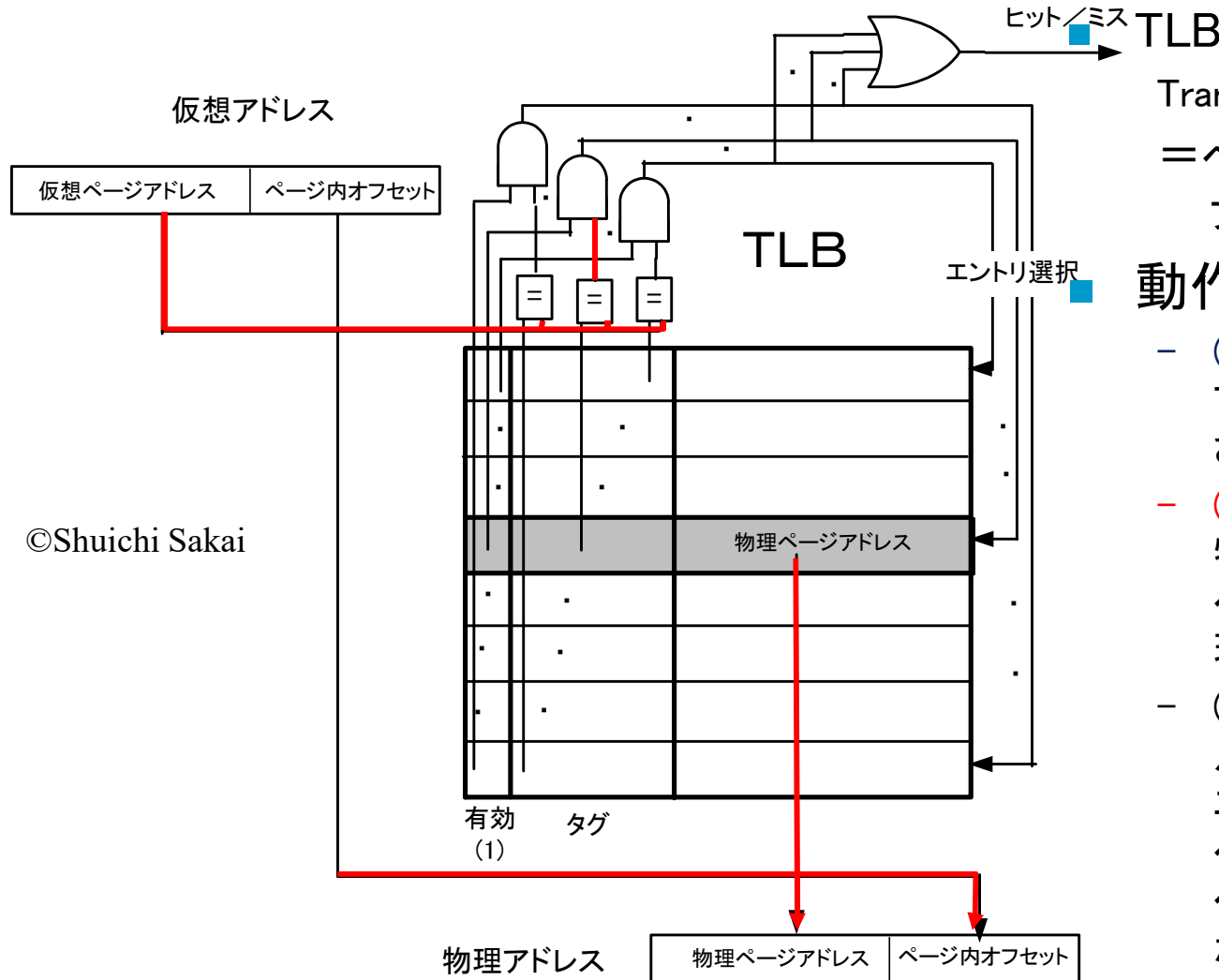
TLB



TLB



TLB



Translation Lookaside Buffer
= ページテーブルのキャッシュ
フルアソシアティブ方式

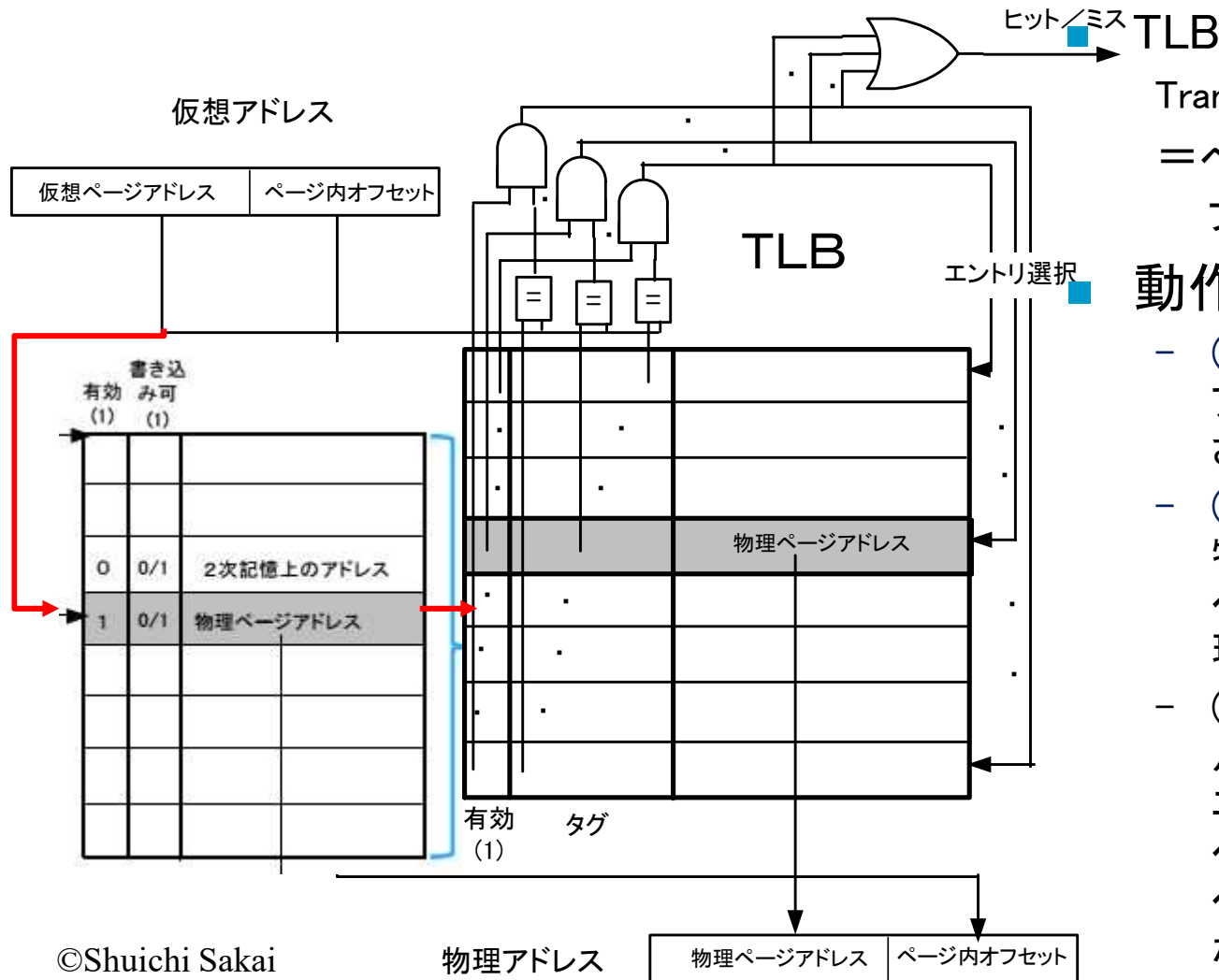
動作

- (1) メモリアクセスが起こると、仮想アドレスをタグとして、TLBが参照される。
- (2) TLBがヒットすると、該当する物理ページアドレスが取り出され、ページ内オフセットとあわせて物理アドレスが作られる。
- (3) TLBがミスすると、ページテーブルが参照され、TLBの空いているエントリに、現在参照している仮想ページアドレスに対応する物理ページアドレスが入れられる。TLBが空いていない場合は、LRUなどのやりかたでエントリがひとつあけられる。

東大・坂井

©Shuichi Sakai

TLB



Translation Lookaside Buffer
= ページテーブルのキャッシュ
フルアソシアティブ方式

動作

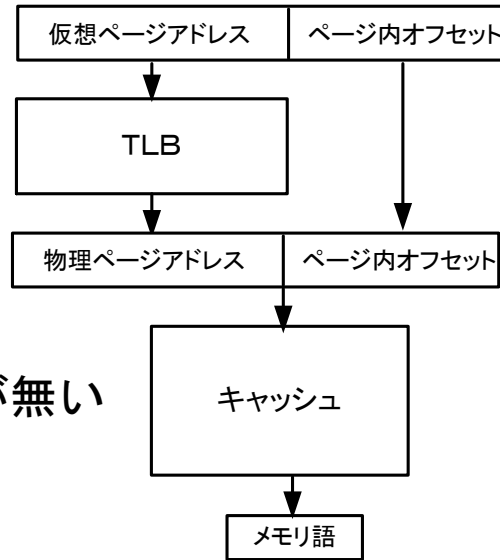
- (1) メモリアクセスが起こると、仮想アドレスをタグとして、TLBが参照される。
- (2) TLBがヒットすると、該当する物理ページアドレスが取り出され、ページ内オフセットとあわせて物理アドレスが作られる。
- (3) TLBがミスすると、ページテーブルが参照され、TLBの空いているエントリに、現在参照している仮想ページアドレスに対応する物理ページアドレスが入れられる。TLBが空いていない場合は、LRUなどのやりかたでエントリがひとつあけられる。

©Shuichi Sakai

コンピュータアーキテクチャ

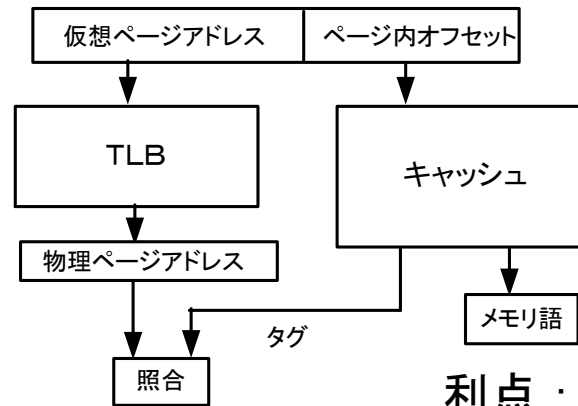
東大・坂井

キャッシュと仮想記憶



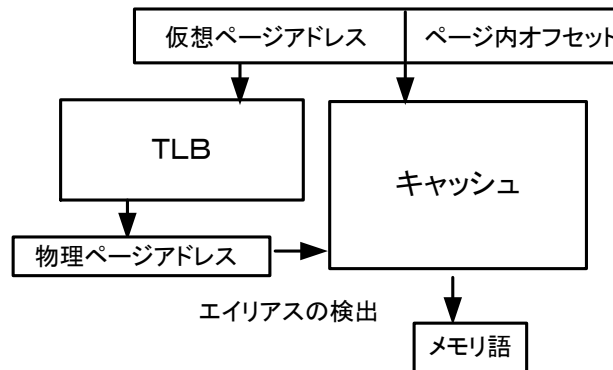
利点：制限が無い
欠点：遅い

(1) 直列型物理アドレスキャッシュ



(2) 並列型物理アドレスキャッシュ

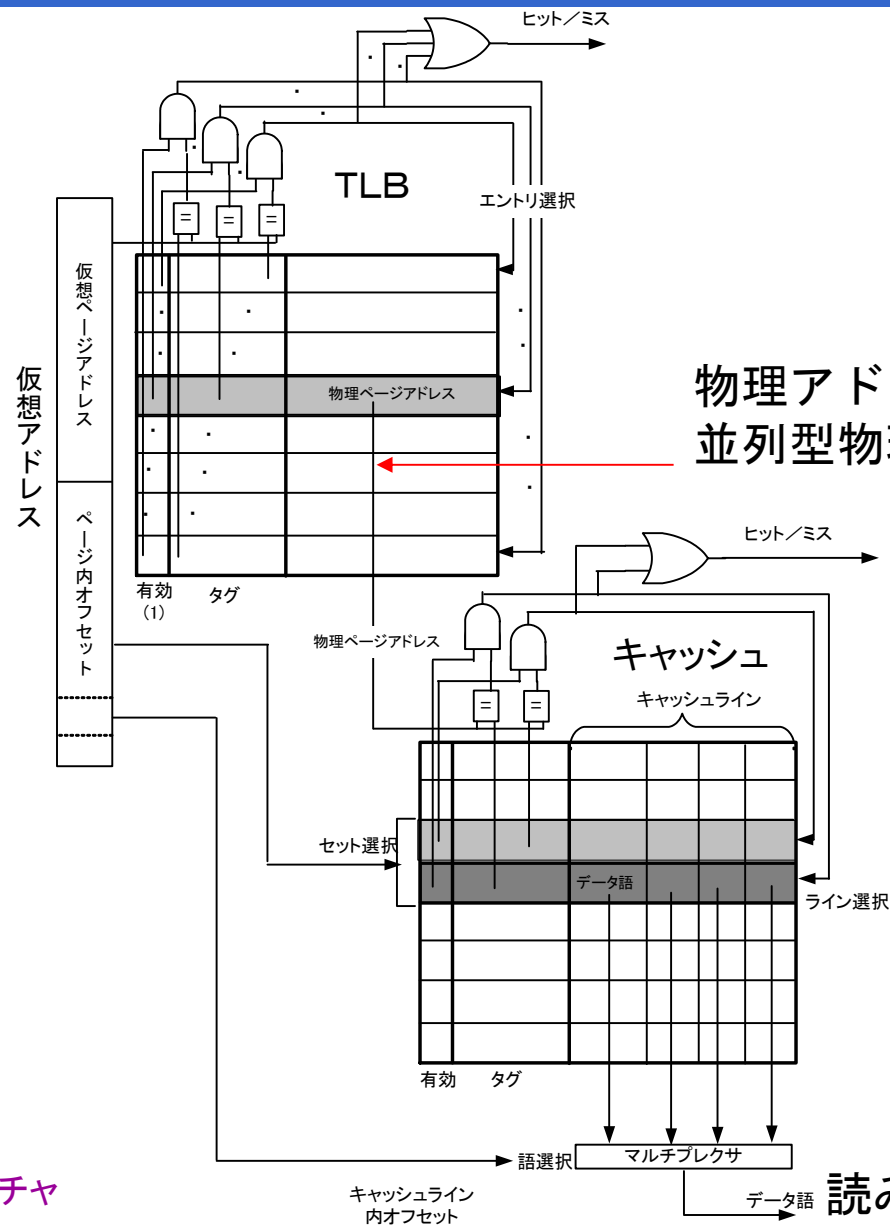
利点：速い
欠点：インデックスが
ページ内
オフセットに
制限される



(3) 仮想アドレスキャッシュ

利点：速いし、容量
制限なし
欠点：Aliasingが起きる

メモリアクセス機構



物理アドレスで照合しているのが
並列型物理アドレスキャッシュ

2way Set associative

©Shuichi Sakai

メモ：書き込みへの対応

- ページにデータが書かれた場合：
 - それを2次記憶に反映させるために、ライトバック方式と、ライトスルー方式がキャッシュ同様考えられる。
 - しかし、2次記憶に書き込むには数百万クロックサイクルもかかるので、ライトスルー方式は現実的では無い。
 - そこで仮想記憶システムではライトバック方式をとらざるを得ない。
- ダーティビット
 - ライトバック方式を効率よく行うには、ページにデータが書き込まれたか否かを示すダーティビットというビットをTLBやページテーブルに追加する必要がある。
 - ダーティビットはページにデータが書き込まれたときにOnになり、ページが主記憶から追い出される時ダーティビットがOnであればページは2次記憶に書き込まれる。

透過性と互換性

■ 透過性 (transparency)

- 機械語プログラムが実装の詳細に影響されない性質
- キャッシュ、仮想記憶、スーパスカラなどの導入で保持される
 - ・ ハードウェア量の増大・速度低下を招いても実現する！

■ 互換性 (compatibility)

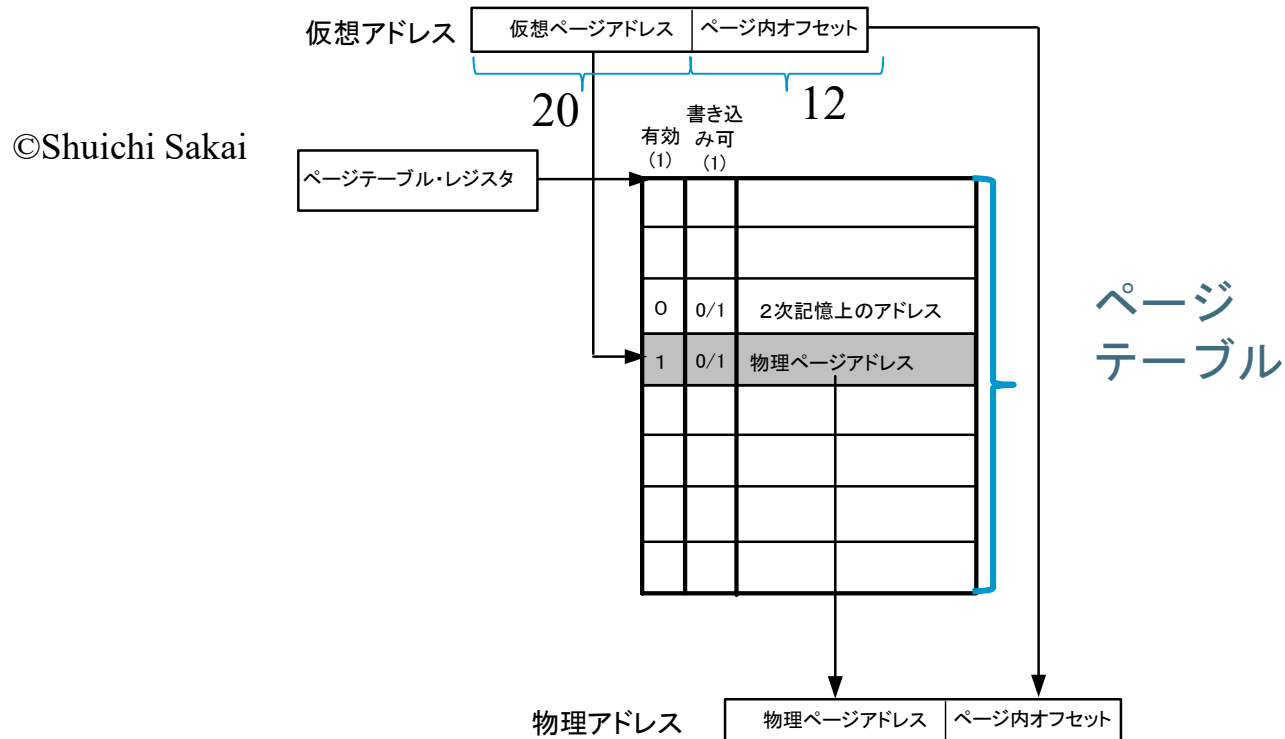
- 異なるコンピュータ間で同じ機械語プログラムが同じ動作をする性質
- コンピュータの世代間や上位・下位機種の間で保持されるのが望ましい
 - ・ システム発展の障害にもなりえる

問題

1. あるコンピュータが32ビットの仮想アドレス空間を持ちページサイズが4KBで有ったとする。
 1. ページテーブルの構成が図5.10で有ったとき、ページテーブルのエントリー数を求めよ。
 2. 物理アドレスが30ビットであったときにページテーブルに必要な容量を求めよ。
2. あるコンピュータが32ビットの仮想空間を持ち、ページサイズが4KB、物理アドレスが31ビット、キャッシュラインが4語(1語32ビット)であったとする。2-Wayセットアソシアティブ型の並列型物理アドレスキャッシュを使った場合、キャッシュに必要なメモリ量を求めよ。但し、インデックスはページ内オフセットを最大限使うものとする。

回答

- 1 ページサイズ4KB⇒ページ内オフセット12ビット
 - 仮想ページアドレス⇒ $32 - 12 = 20$ ⇒ 1Mエントリ
 - 物理アドレスは30ビットなので物理ページアドレスは18ビット
 - 1エントリは $1 + 1 + 18 = 20$ ビット
 - だけど、これは主記憶上なのでバイト単位にしたい ⇒ 32ビット
 - $4B * 1M$ エントリ = 4MB
 - なお前提として2次記憶上のアドレスが30ビットに収まっているとしている。



回答2

- ページ内オフセット=12ビット
 - 2ビットは語内アドレス、2ビットは語選択=>
 - 8ビットがセット選択に=>2Wayなので、 $256 * 2 = 512$ エントリ
- 物理アドレスが31ビット
 - タグが19ビット+有効ビット1+32ビット*4語
 - => $20 + 128 = 148$ ビット
 - $512 \text{ エントリ} * 148 \text{ ビット} = 74 \text{ Kbits} = 9.25 \text{ KB}$

