

## コンピュータハードウェア (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 / 電気工学科

- はじめに
- キャッシュ

コンピュータハードウェア

東大・坂井

## 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 10:15 - 11:45、I3 - 31
- 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)  
教科書通りやります
- 参考書
  - D. Patterson and J. Hennessy, Computer Organization & Design, 2nd Ed. (邦訳 『コンピュータの構成と設計』 (第2版) 上下 (日経 B P))
  - 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
  - 富田真治 『コンピュータアーキテクチャ』 a、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 (+出席)

コンピュータハードウェア

東大・坂井

## 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)  
フォワードリング、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、フルアソシティブ型、セットアソシティブ型、キャッシュミス

コンピュータハードウェア

東大・坂井

## 講義の概要と予定 (2 / 2)

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
8. 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、スーバスカラ、並列処理とハザード
9. 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、トレーススケジューリング
10. 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
11. 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月後半

コンピュータハードウェア

東大・坂井

## 6. キャッシュ

### ■ 内容

#### － 記憶階層

- 命令パイプラインとメモリ
- 記憶階層と局所性
- 透過性

#### － キャッシュ

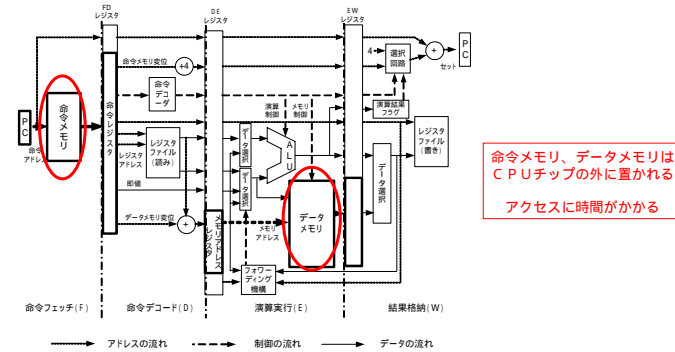
- キャッシュとはなにか
- ライトスルーとライトバック
- ダイレクトマップ型キャッシュ
- キャッシュミス
- フルアソシアティブ型キャッシュ
- セットアソシアティブ型キャッシュ
- キャッシュの入ったCPU
- キャッシュの性能

コンピュータハードウェア

東大・坂井

## 命令パイプラインとメモリ

- 問題: はたしてメモリは1クロックで読み書きできるか?



コンピュータハードウェア

東大・坂井

## メモリの理想と現実

### ■ 理想

- － 無限大の容量
- － 無限小のアクセス時間
- － 単純なアドレッシング

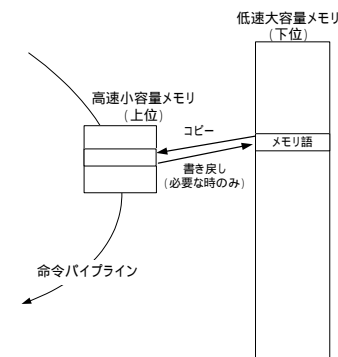
### ■ 現実

- － 大容量と高速アクセスは両立しない
  - 容量が大きくなるとアクセス時間も大きくなる

コンピュータハードウェア

東大・坂井

## 記憶階層と局所性



- なぜ記憶階層が有効か?

命令やデータに局所性があるから

#### － 空間局所性

- あるメモリ語が参照されたときに、その語の近くの語が引き続き参照される性質

#### － 時間局所性

- あるメモリ語が参照されたとき、その語が時間をおかず再び参照される性質

コンピュータハードウェア

東大・坂井

## 記憶階層と機械語プログラム

### 記憶階層を陽に見せる方式

- 各階層のメモリのそれぞれにアドレスを付け、コピーや書き戻しもロード命令・ストア命令で実現する
- ×アセンブリ言語のプログラマが記憶階層を意識しなくてはならず、いつも最良のメモリの利用法を考えてプログラムをしなくてはならない。
- ×命令セットが同じでもメモリの階層構成が変化したり、各階層のメモリの大きさが変化したりするたびにプログラムを作り直さなくてはならない。

### 記憶階層を見せない方式

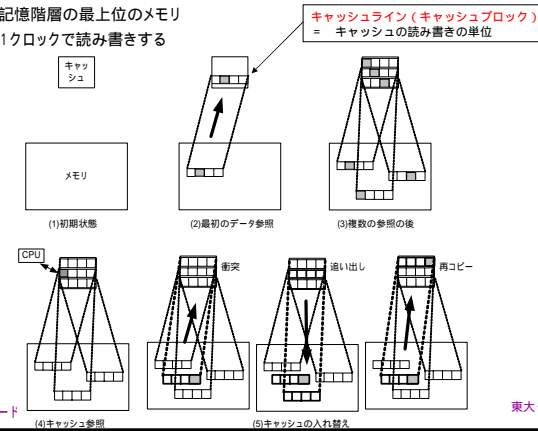
- 見かけ上は、高速で大容量のメモリが1つだけあるものとして機械語のプログラムを書き、ハードウェアの機構でどの階層のメモリをどう使って局所性を活かすかを決める
- 単純に命令セットだけを意識して機械語プログラムを書いておけば、効率や完全性はハードウェアが勝手に面倒を見てくれる。この性質を**透過性 (transparency)**と呼ぶ。

コンピュータハードウェア

東大・坂井

## キャッシュとは何か

- キャッシュ
  - 記憶階層の最上位のメモリ
  - 1クロックで読み書きする



コンピュータハード

東大・坂井

## ライトスルー方式とライトバック方式

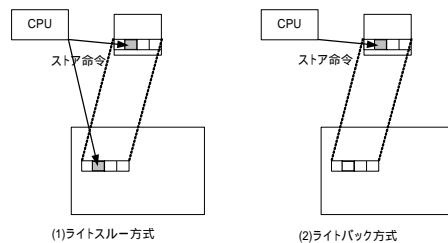


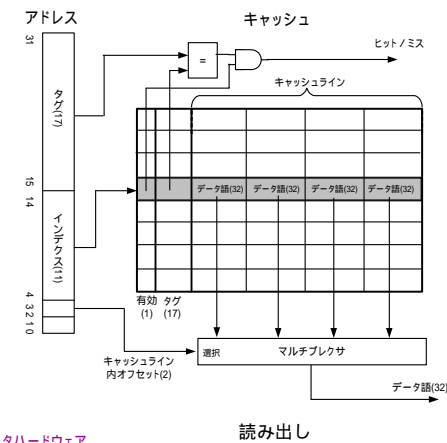
表 5.1 ライトスルー方式とライトバック方式

	ライトスルー方式	ライトバック方式
メモリアクセス	ストア命令の実行時	キャッシュライン追い出しの時
ライト命令の実行速度	ライトバッファの速度	キャッシュの速度
キャッシュライン書き戻し	不要	キャッシュライン追い出しの時
実装	単純	複雑

コンピュータハードウェア

東大・坂井

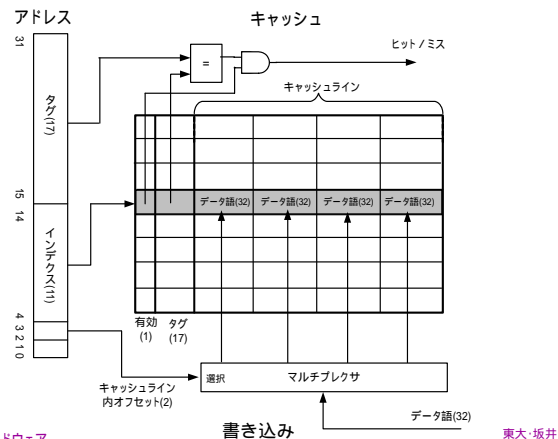
## ダイレクトマップ型キャッシュ(1)



コンピュータハードウェア

東大・坂井

## ダイレクトマップ型キャッシュ(2)



## キャッシュミス

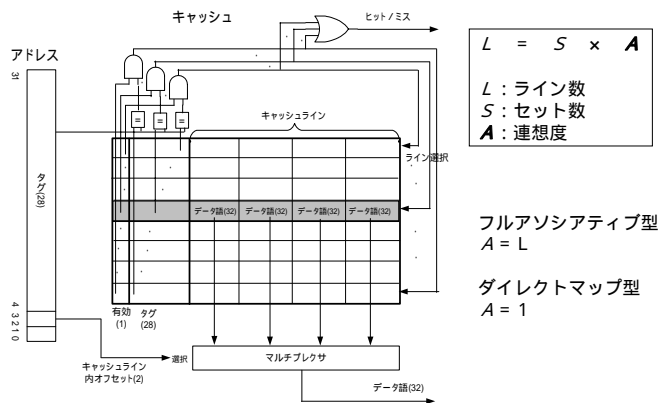
キャッシュミス = 3つのC

- 初期参照ミス(compulsory miss, cold start miss)  
キャッシュラインを最初にアクセスするときにかかるミス
- 競合性ミス(conflict miss, collision miss)  
同じインデックスをもつ異なるキャッシュラインにアクセスすることで起こるミス
- 容量性ミス(capacity miss)  
キャッシュしたいライン数がキャッシュ容量を上回ることで起こるミス

コンピュータハードウェア

東大・坂井

## フルアソシアティブ型キャッシュ



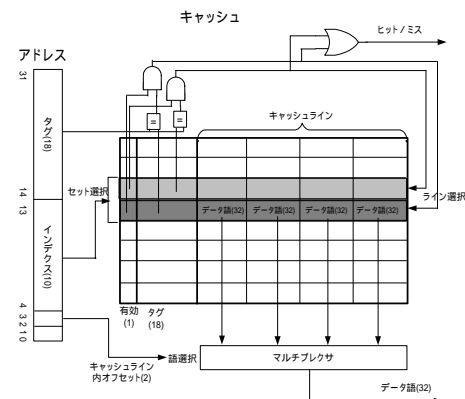
$$L = S \times A$$

$L$ : ライン数  
 $S$ : セット数  
 $A$ : 連想度

フルアソシアティブ型  
 $A = L$

ダイレクトマップ型  
 $A = 1$

## セットアソシアティブ型



## キャッシュ方式の比較

表 5.2 キャッシュの3つの型

	ダイレクトマップ	セットアソシアティブ	フルアソシアティブ
連想度	1	$A(2, 4 \text{ など})$	=ライン数
セット数	=ライン数	$S$	1
ハードウェア			×
ゲート遅延			×
競合性ミス	×		

### ■ キャッシュラインの交換

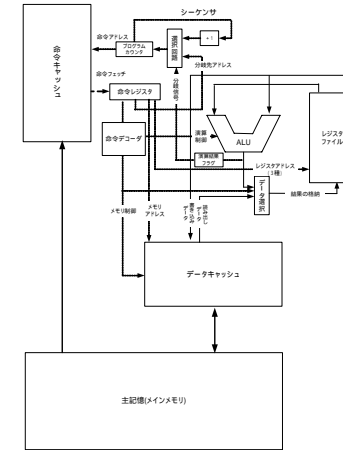
追い出すラインをどう決めるか？

- ランダム
- LRU (Least Recently Used)  
使われていない時間が最も長い(Least Recent Used)ラインを  
追い出しの対象とする

コンピュータハードウェア

東大・坂井

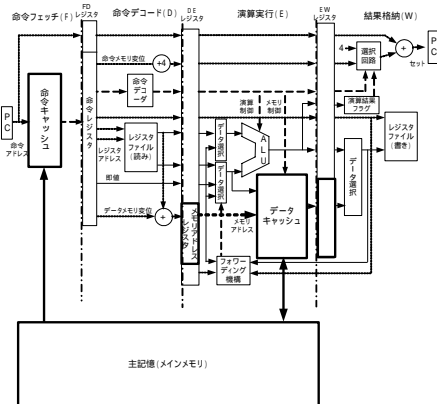
## キャッシュの入ったCPU



コンピュータハードウェア

東大・坂井

## キャッシュの入った命令パイプライン



コンピュータハードウェア

→ アドレスの流れ    - - - - - 制御の流れ    → データの流れ

東大・坂井

## キャッシュの性能

- $T_p = N \cdot (1 + r_{ls} \cdot r_{miss} \cdot t_{mstall}) / C$ 
  - $T_p$ : プログラム実行時間
  - $C$  [Hz]: CPUのクロック速度
  - $N$ : プログラムで実行される命令の数
  - $r_{ls}$ : ロード・ストア命令の割合
  - $r_{miss}$ : ロード・ストア命令ごとのキャッシュミス率
  - $t_{mstall}$ : 1回のミスによるストール時間(ミスペナルティ, miss penalty)

表 5.4 キャッシュミス率とミスペナルティの例(解答)

	ミス率	ミスペナルティ	実行時間相対値
事例 1	0	-	1
事例 2	0.05	10	1.15
事例 3	0.05	50	1.75
事例 4	0.5	10	2.5
事例 5	0.5	50	8.5

$R_{ls} = 0.3$ の場合

コンピュータハードウェア

東大・坂井