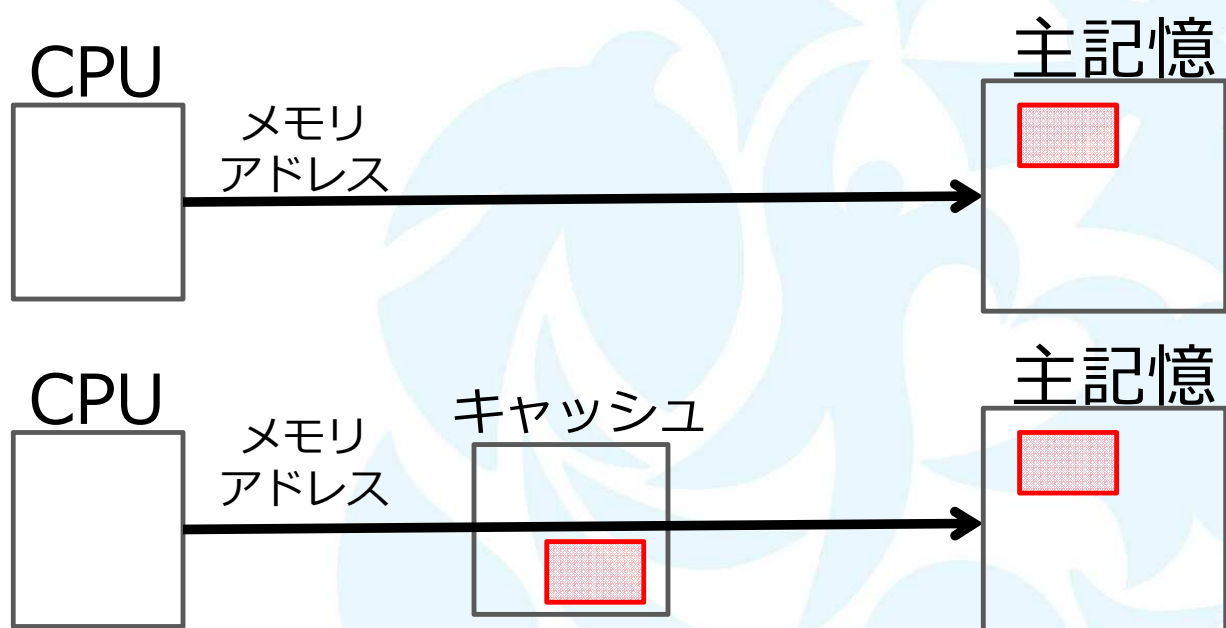


キャッシュの アドレスマッピング

アドレスマッピングとは

- キャッシュ内のアドレスの決め方



マッピング問題の筋立て

- 要求：
 - すぐ見つかる
 - キャッシュ容量が無駄にならない
- 2つの考え方
 - アドレスから計算できる場所
 - ⇒ すぐ見つかる、無駄が出る
 - 先頭から詰めて置く
 - ⇒ 無駄がない、探す必要あり



駐車場のイメージ

空いている所どこでも

場所の効率がいい (いっぱいまで使える)

決まった所 (番号割当)



駐車場のイメージ

空いている所どこでも

場所の効率がいい（いっぱいまで使える）

車を探すのが大変（場所が分からない）

決まった所（番号割当）

駐車場のイメージ

空いている所どこでも

場所の効率がいい（いっぱいまで使える）

探すのが大変（場所番号が分からない）

決まった所（ナンバー下2桁＝場所番号）

駐車場のイメージ

空いている所どこでも

場所の効率がいい（いっぱいまで使える）

入れるとき簡単（空いている所どこでも）

探すのが大変（場所番号が分からない）

決まった所（ナンバー下2桁＝場所番号）

探すのは簡単（ナンバーからすぐ分かる）

6



東邦大学

駐車場のイメージ

空いている所どこでも

場所の効率がいい（いっぱいまで使える）

入れるとき簡単（空いている所どこでも）

探すのが大変（場所番号が分からない）

決まった所（ナンバー下2桁＝場所番号）

探すのは簡単（ナンバーからすぐ分かる）

場所の効率悪い（他に空いていてもダメ）

7



東邦大学

キャッシュの置き方も同じ

空いている所どこでも（フルアソシアティブ方式）
場所の効率がいい（いっぱいまで使える）
探すのが大変（場所番号が分からない）

探す時

メインメモリ
アドレス
→
7250

アドレス情報

ブロックデータ

5320	XXXXXX
1760	XXXXXX
一致 7250	XXXXXX
⋮	
⋮	
空き	



東邦大学

8

キャッシュの置き方も同じ

決まった所(アドレス一部)(ダイレクトマップ方式)
探すのは簡単（アドレスからすぐ分かる）
場所の効率悪い（他に空いていてもダメ）

探す時

メインメモリ
アドレス
→
7250

7 2 5 0

アドレスの一部
を切り出す

ブロックデータ

00	XXXXXX
01	XXXXXX
⋮	
25	XXXXXX
⋮	
99	



東邦大学

9

というわけで折衷案

10



東邦大学

というわけで折衷案 セットアソシアティブ方式

行内の場所を選ぶのは
フル・アソシアティブ方

どの行を
選ぶかは
ダイレクト
マップ方式

11



東邦大学

キャッシュの置き方(マッピング)のまとめ

方式

場所が決まる

方式

置ける

方式

両者の折衷案



東邦大学

キャッシュの置き方(マッピング)のまとめ

ダイレクトマップ方式

アドレスによって場所が決まる

方式

置ける

方式

両者の折衷案



東邦大学

キャッシュの置き方(マッピング)のまとめ

ダイレクトマップ方式

アドレスによって場所が決まる

フルアソシアティブ方式

任意の場所に置く

方式

両者の折衷案

キャッシュの置き方(マッピング)のまとめ

ダイレクトマップ方式

アドレスによって場所が決まる

フルアソシアティブ方式

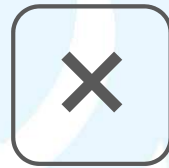
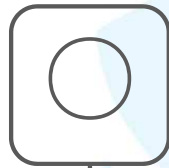
任意の場所に置く

セットアソシアティブ方式

両者の折衷案

ダイレクトマップで決めたブロックの中でフルアソシアティブ

キャッシュのマッピング方式について理解できましたか？



↓
次へ



東邦大学