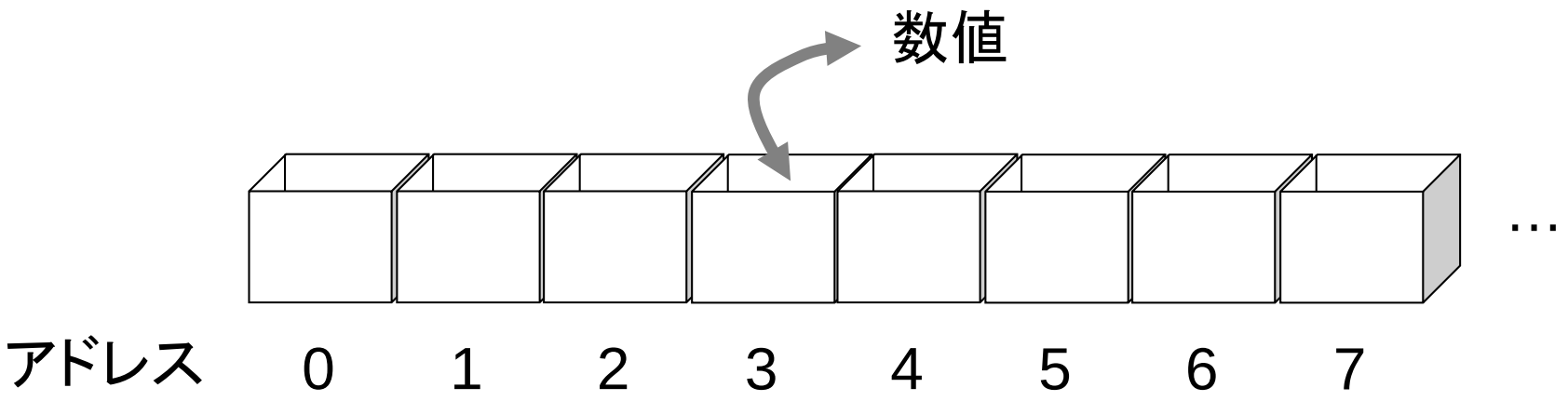
東北大学 工学部 機械知能・航空工学科
2020年度 クラス C D

情報科学基礎 I

13. メモリシステム
(教科書8章)

大学院情報科学研究科
鏡 慎吾

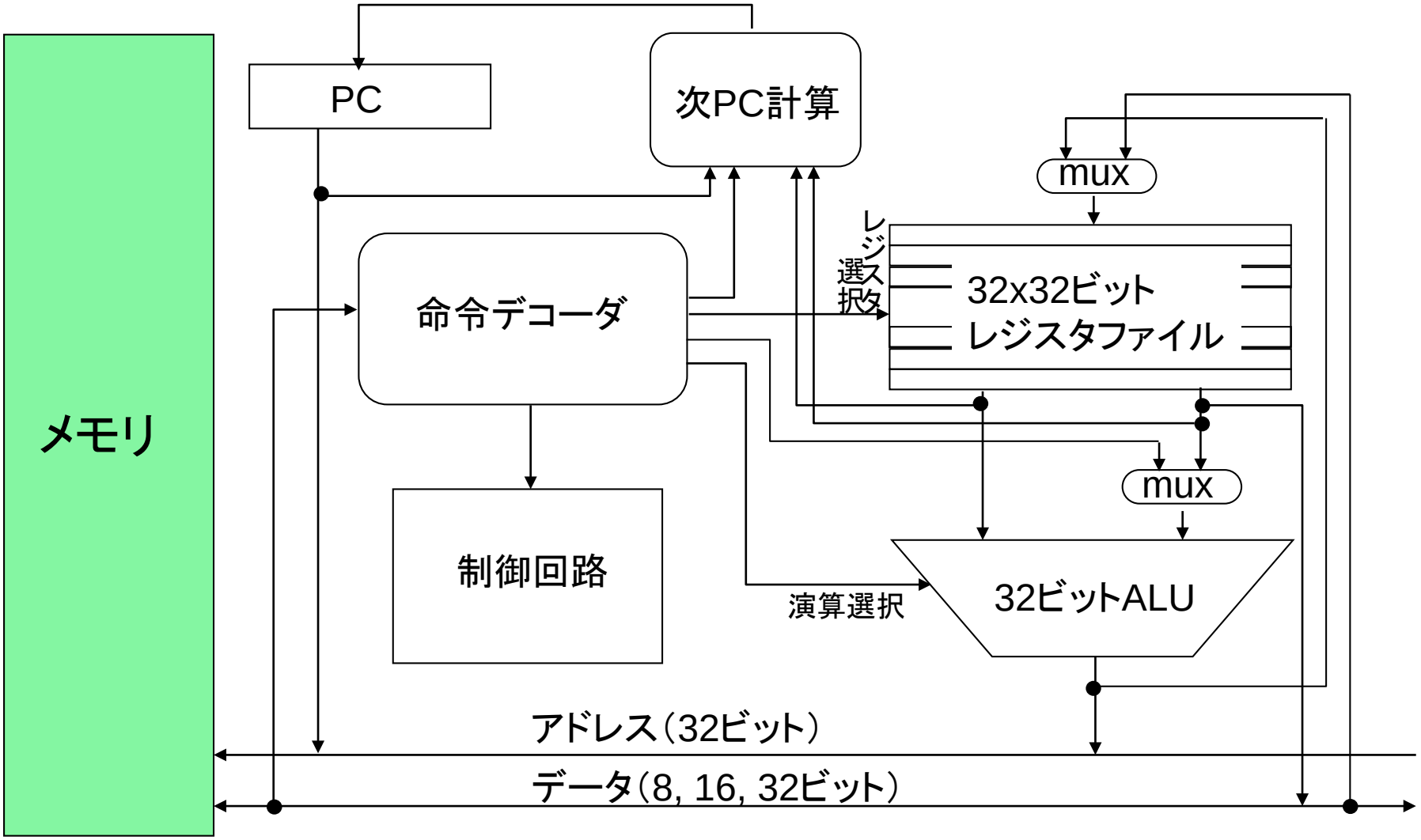
復習: メモリの概念



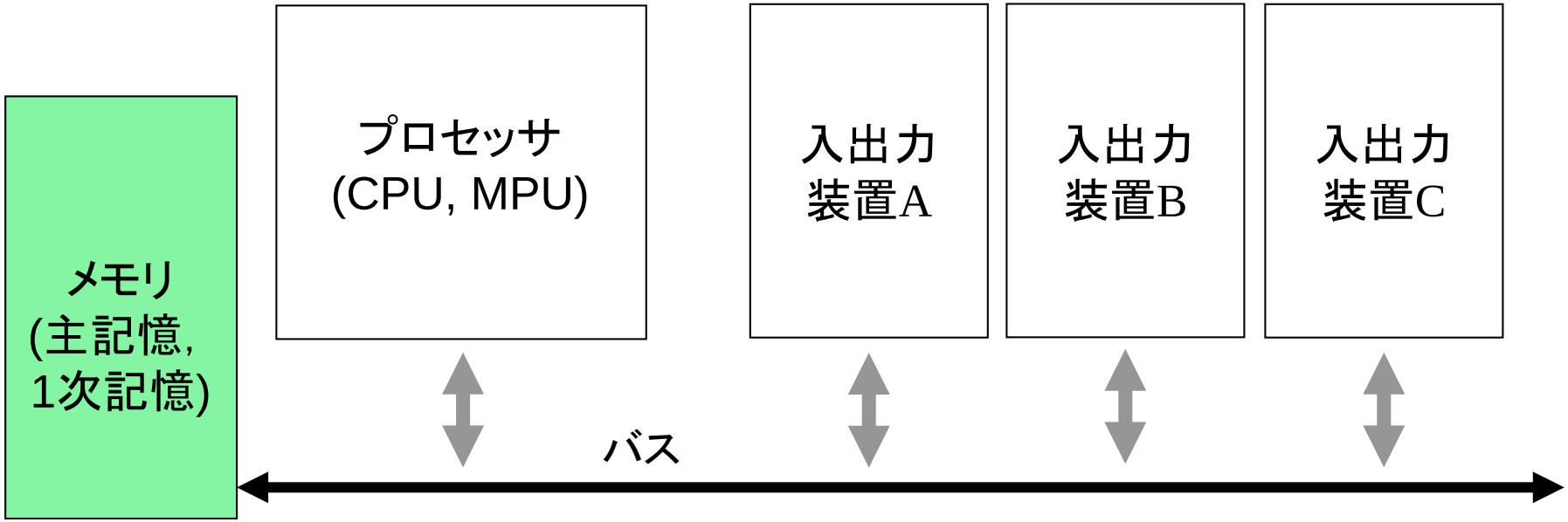
- 数値を格納する区画が並んだもの
- 番号（**アドレス**）でインデックスづけられている
- アドレスを指定して，数値を書き込んだり，読み出したりする
- 書き込むことと読み出すことをあわせて「アクセスする」という

- **同じ大きさ**の区画に，**順番に**アドレスがついていることが重要

(復習) MIPSの構造


メモリ
PC
次PC計算
mux
レジスタ選択
32x32ビット
レジスタファイル
命令デコーダ
mux
制御回路
演算選択
32ビットALU
アドレス（32ビット）
データ（8, 16, 32ビット）

(復習) 計算機の基本構成



別物!

入出力装置 (Input/Output, I/O) の例

- 二次記憶（外部記憶，ストレージ）: ハードディスク, CD, DVD
- キーボード，マウス
- グラフィックス，ディスプレイ
- ネットワーク

「メモリ」という用語の混乱

Q: 以下の会話は，2017年頃のウェブ上で実際に見られたやり取りの例である（一部改変）．何がおかしいのかを指摘せよ．

「当社では社員が使う PC のメモリはすべて 32 GB です．快適に作業ができます」
「えっ? 32 GB って少なくないですか? 私の iPhone のメモリは 128 GB なんですけど」

A: 前者は主記憶の話をしている．後者は二次記憶の話をしている

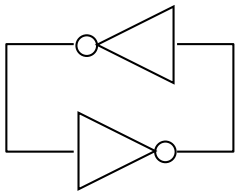
- コンピュータアーキテクチャの観点では，主記憶をメモリ，二次記憶をストレージと呼ぶことが多い
- デバイスの観点では，半導体記憶素子をすべてメモリと呼んでしまうことがある (e.g.:「フラッシュメモリ」は半導体記憶素子だが，主な用途は二次記憶)
- そこにマーケティング上の思惑が絡むのでさらにややこしい，というか迷惑

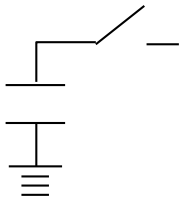
Cプログラムの場合

```
int main() {
    FILE *fp;
    char str[1024];

    fp = fopen("file.txt");
    fgets(str, 1024, fp);
    ...
}
```

- プログラム上の変数は主記憶（またはレジスタ）上にある
- 二次記憶上のデータは，入出力関数を使って読み書きする

記憶装置の原理





- フリップフロップ構造
  - レジスタ
  - SRAM → キャッシュメモリ（後述）
- キャパシタへの電荷蓄積
  - DRAM → 主記憶
- 磁気
  - 磁気記憶装置 (ハードディスク)
    → 二次記憶

（ただし最近はフラッシュメモリによる置き換えが進んでいる）

揮発性
（電源を切ると内容は消える）

不揮発性

速

遅

https://commons.wikimedia.org/w/index.php?title=File:HardDisk1.ogv

半導体メモリアレイ

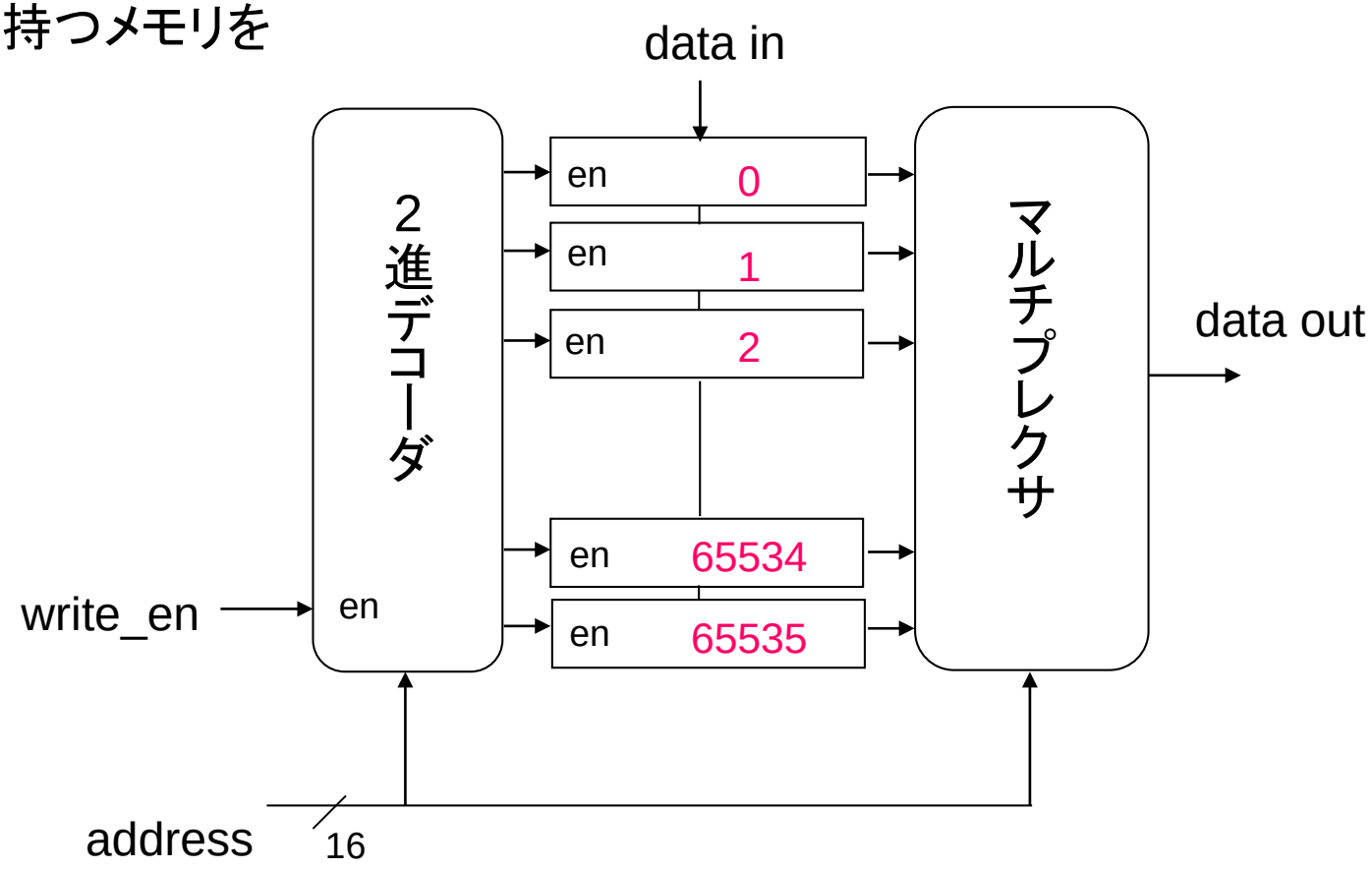
半導体メモリの分類

- シーケンシャルアクセス
- ランダムアクセス (RAM: Random Access Memory)
  - 揮発性
    - SRAM
    - DRAM
  - 不揮発性
    - 読み出し専用 (ROM: Read-Only Memory)
      - マスクROM
    - 読み書き可能
      - PROM (Programmable ROM)
        - ワンタイムPROM
        - EPROM (Erasable PROM)
          - UV-EPROM (紫外線で消去)
          - EEPROM (電気的に消去)
            - フラッシュメモリ

※ RAM と ROM は対義語ではない(ほとんどの ROM はランダムアクセス可能)
※ ROM といいつつ実は書き込めるものも市場には多い
※「メモリ」という名前でも実は「二次記憶装置」の場合がある (e.g. USBメモリ)

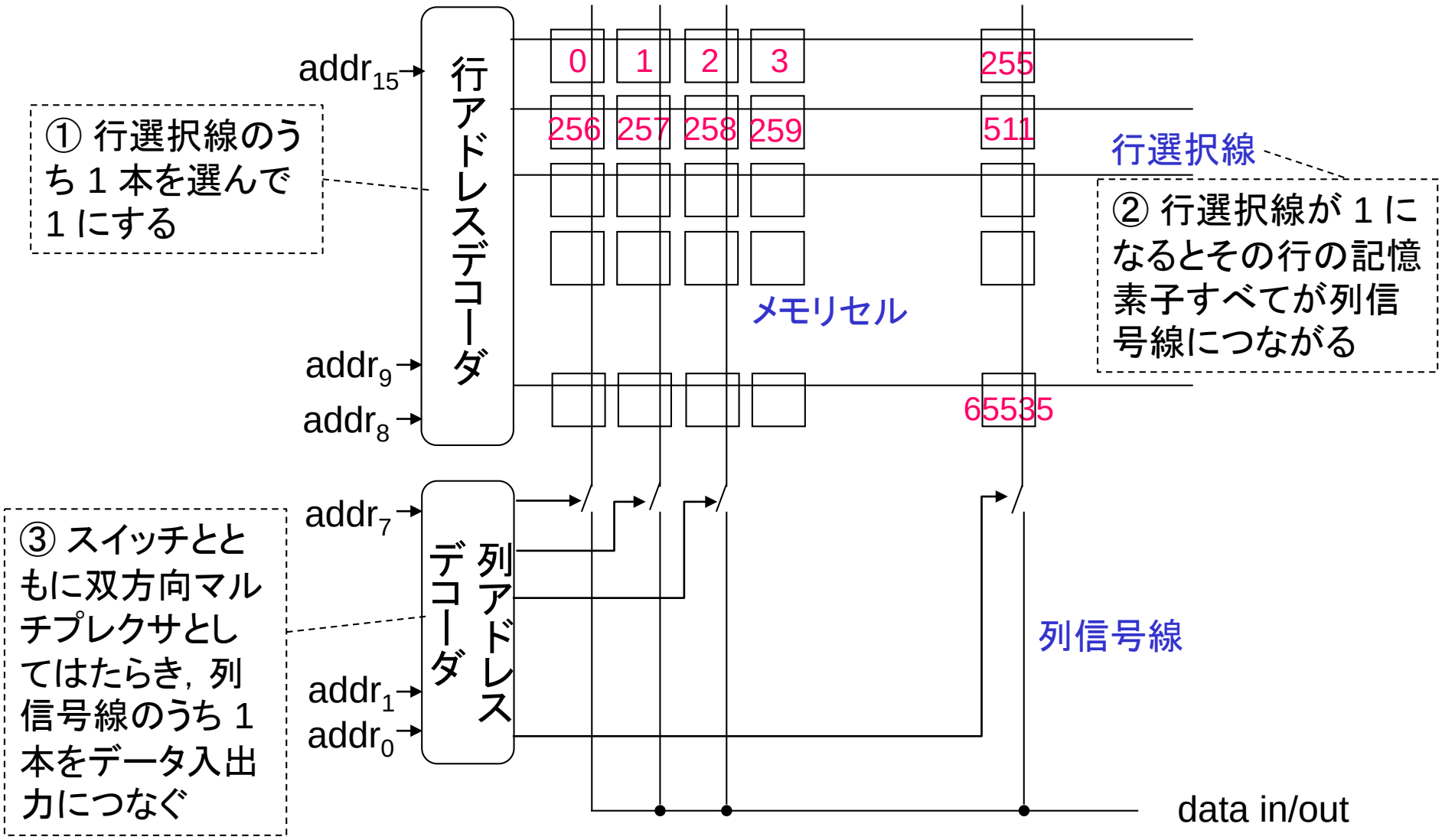
レジスタファイルと同じように作るとどうなるか

N 個のアドレスを持つメモリを
作ろうとすると…

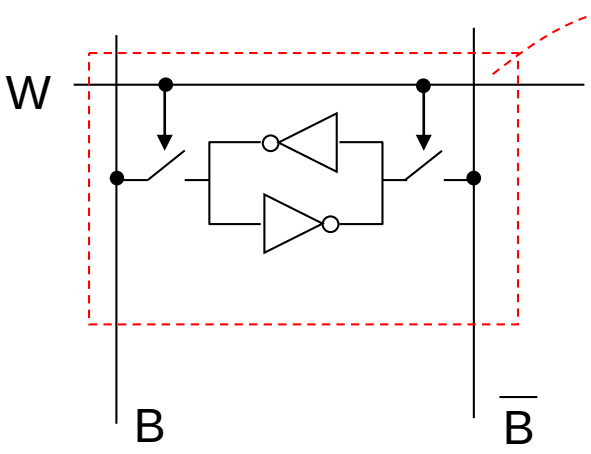


- 書き込み選択に N 出力デコーダが必要
- 読み出し選択に N 入力マルチプレクサが必要

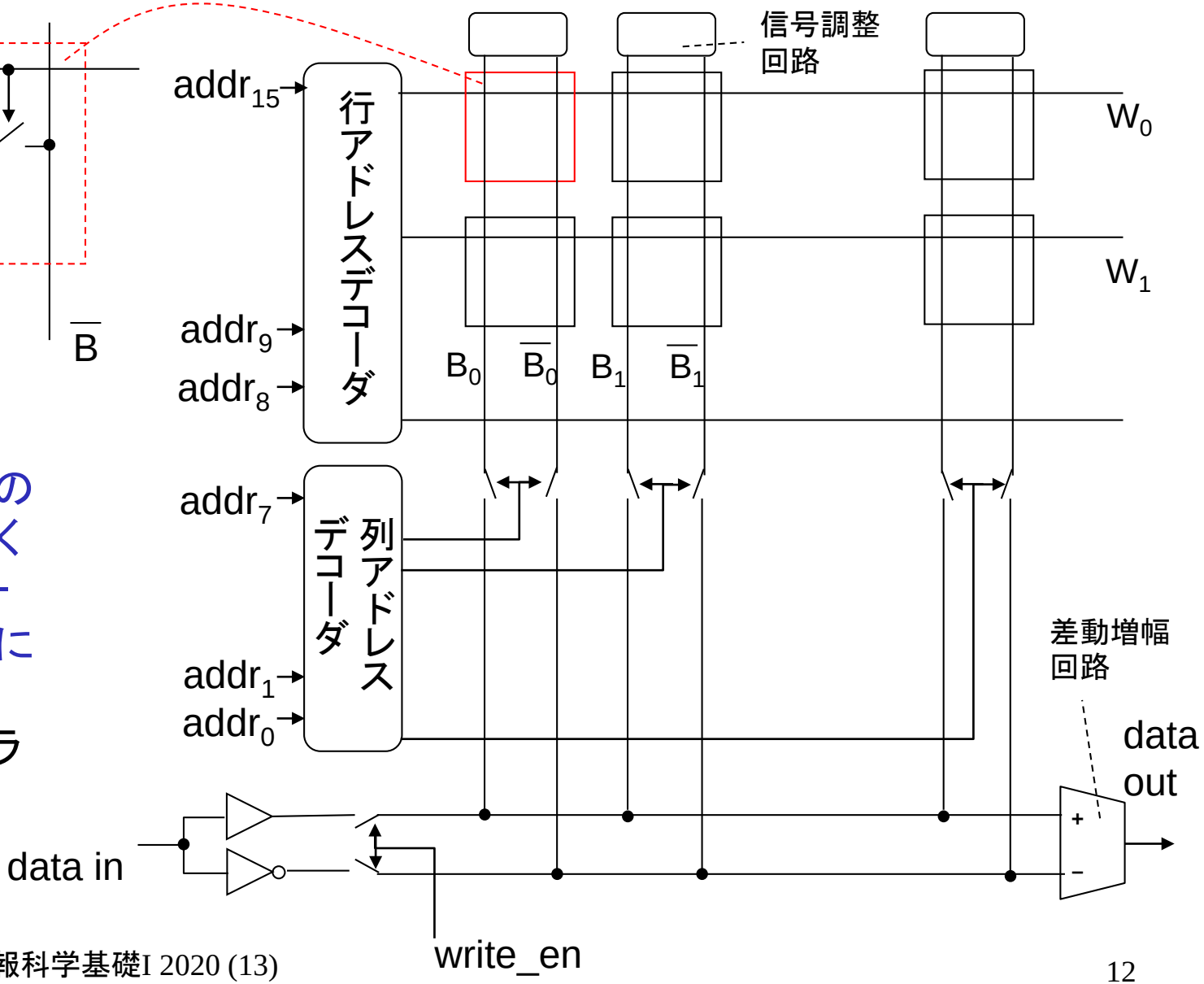
2次元アレイ構造の導入



Static RAM (SRAM)



- フリップフロップの基本回路と同じく2 個のNOTゲートのループ構成により記憶
- 1 ビットあたりトランジスタ 6 個



SRAM の動作

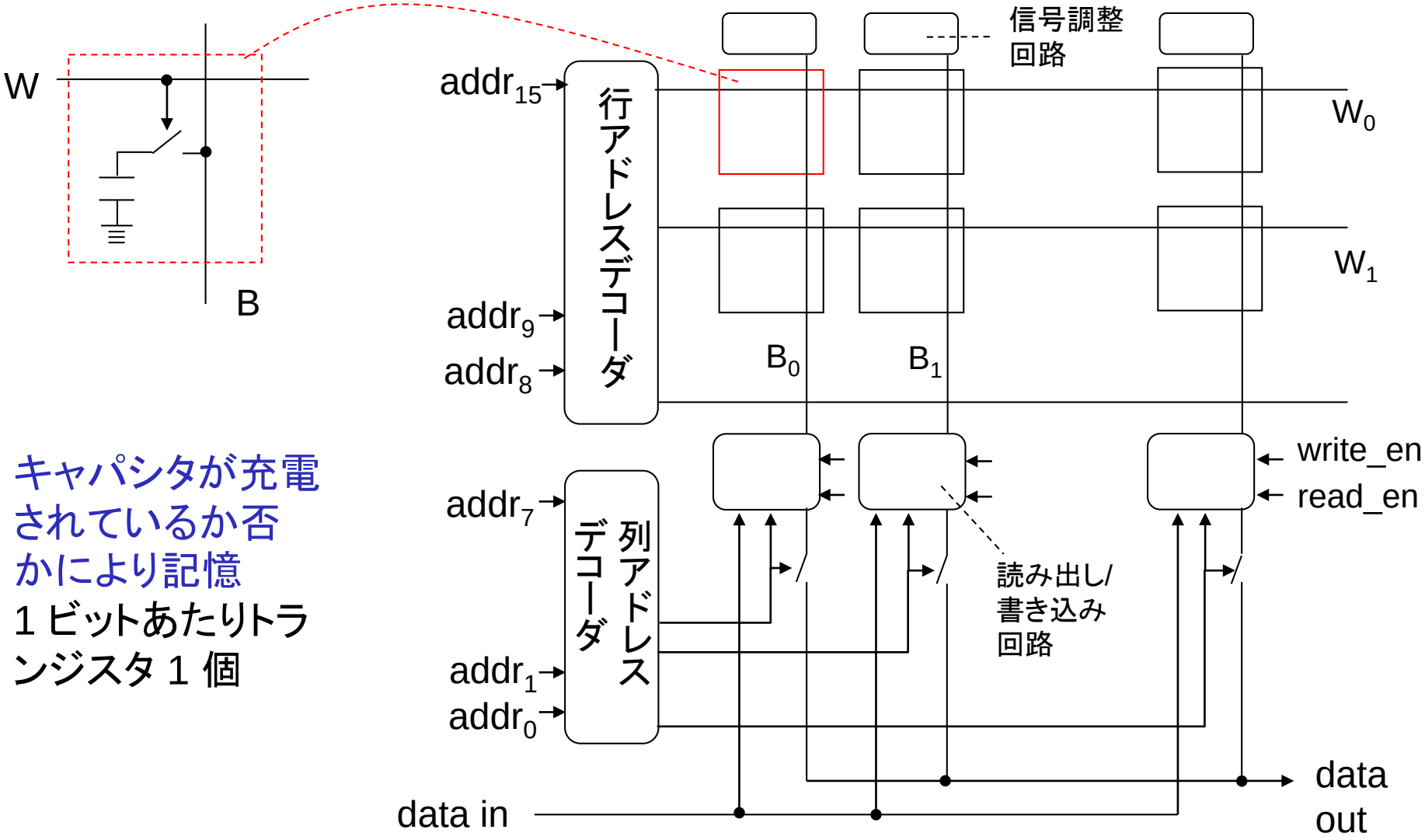
書き込み

- 行・列アドレスデコーダによりセルを選択する
- write_en を 1 にし，data in 部から，セル内のフリップフロップよりも強い駆動力でデータ信号を送る
- 一定時間経ってフリップフロップの状態が安定したら write_en を 0 にする

読み出し

- (write_en は常に 0 にしておく)
- 行・列アドレスデコーダによりセルを選択する
- セル内のフリップフロップからの電流供給により，列信号線対に電圧差が現れ始める
- data out 部の差動増幅器によってその電圧差を検出し，0 または 1 として出力する

Dynamic RAM (DRAM)



- キャパシタが充電されているか否かにより記憶
- 1 ビットあたりトランジスタ 1 個

DRAM の動作

書き込み

1. 行アドレスデコーダにより行を選択する
2. 選択された行の各セル内のキャパシタと列信号線の間で電荷の分配が起きる（注: この行に記憶されていたデータは失われる）
3. 各列の読み出し/書き込み回路の列信号線の電圧変化（+ または –）を検出し，値 1 または 0 としていったん記憶する（行まるごと）
4. 列アドレスデコーダによって選択された列の読み出し/書き込み回路が，さっき記憶した値を data in からのデータで置き換える
5. 各列の読み出し/書き込み回路は，記憶されている値に応じて，列信号線およびその先のキャパシタに 1 または 0 の電圧を出力する（行まるごと）
6. 行アドレスデコーダによる行の選択を解除する

読み出し

1〜3. 書き込み手順と同じ

4. 列アドレスデコーダによって選択された列の読み出し/書き込み回路が，さっき記憶した値を data out に出力する

5〜6. 書き込み手順と同じ

リフレッシュ

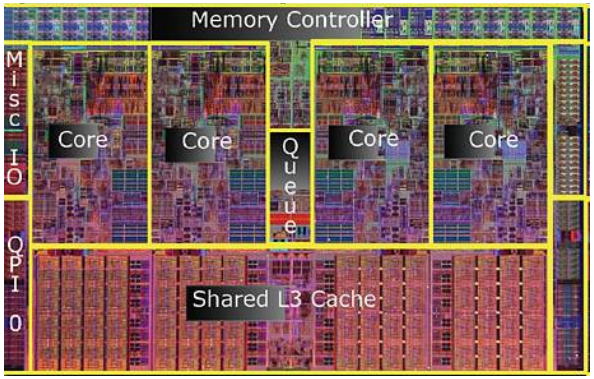
- キャパシタ内の電荷は自然放電するので，上記 1〜3, 5〜6 の手順を定期的に実行する必要がある

SRAM vs DRAM

SRAM

- 1セルの回路が大きい
- 制御が比較的簡単

よって，速いが小容量
→ 主記憶を補助するメモリとして使う
（後述するキャッシュメモリ）

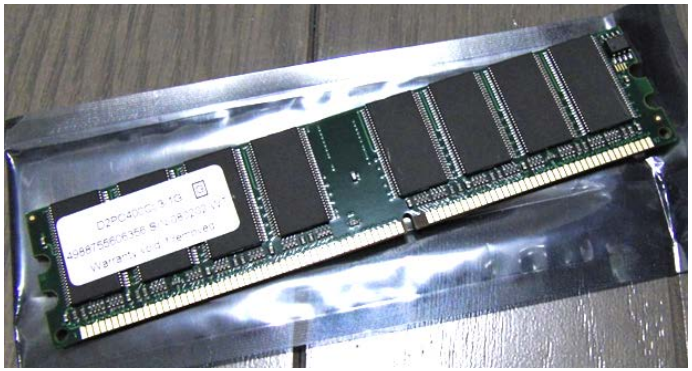


http://ja.wikipedia.org/wiki/Intel_Core_i7
Intel Developer Forum 2008, Shanghai

DRAM

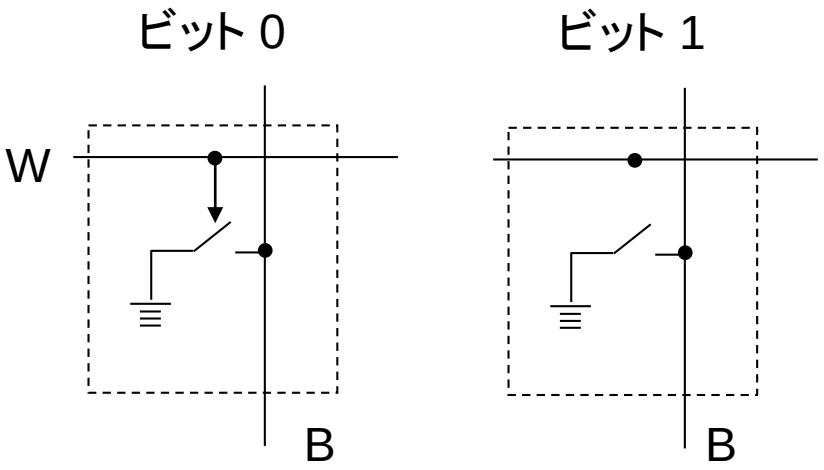
- 1セルの回路が小さい
- 制御が比較的複雑

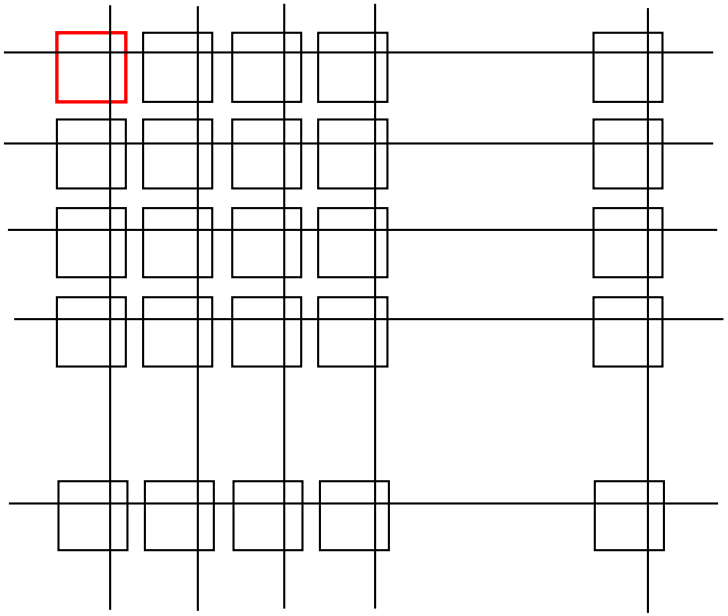
よって，遅いが大容量
（ただし行まるごとの連続アクセスはまあまあ速い）
→ 主記憶として使う



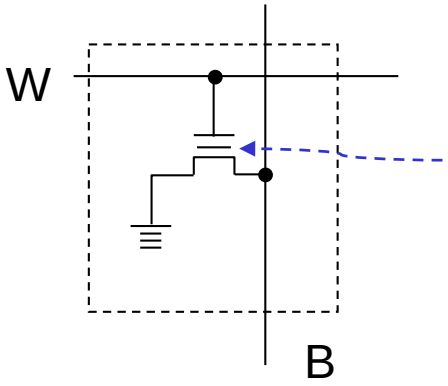
マスクROM と EEPROM

マスクROMの構成例

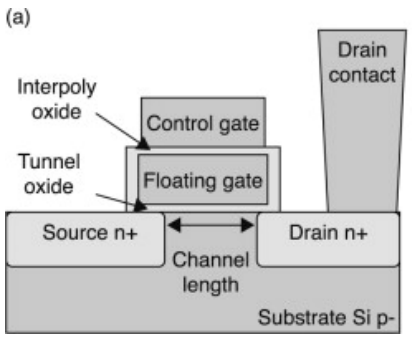


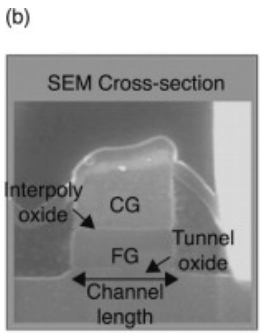


EEPROMの構成例



フローティングゲートと呼ばれる部分の電荷の有無によって，ゲートに電圧をかけてもスイッチオンできなくすることができる





R. Bez, A. Pirovano, in Advances in Non-Volatile Memory and Storage Technology, 2014

ファミリーコンピュータ用ROMカートリッジ（ロムカセット）

http://ja.wikipedia.org/wiki/
ファイル:Famicom_ROM_cassette.jpg

http://blog.livedoor.jp/game_retro/archives/1403347.html

キャッシュメモリ

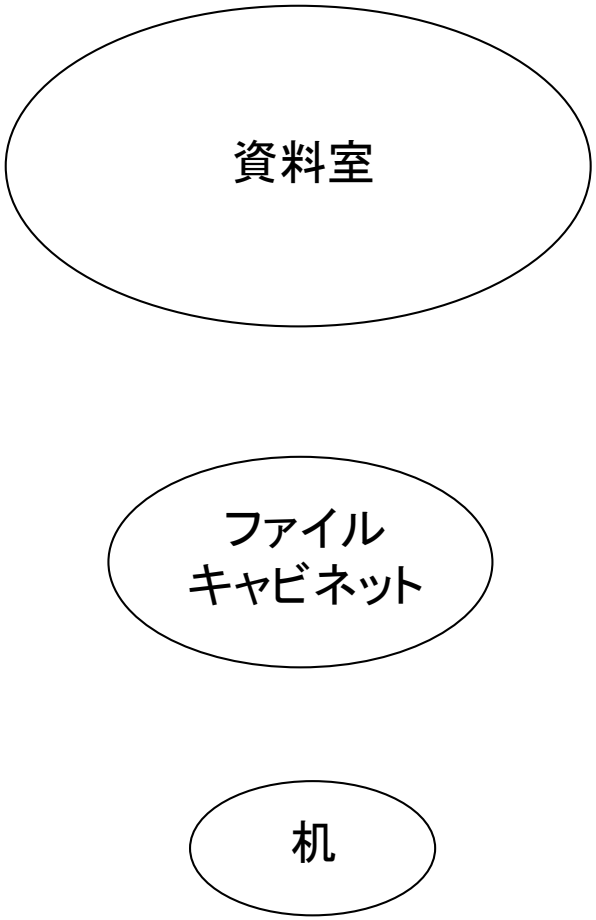
記憶階層

一般論として
- 記憶装置は小容量だと速く，大容量だと遅い
- アクセス開始には時間がかかり，連続データのアクセスは速い

	レイテンシ（遅延時間）	容量
ネットワーク上の記憶	~ ∞	~ ∞
ハードディスクドライブ	~ 10 ms	~ TBytes
DRAM	~ 100 ns	~ GBytes
SRAM	~ 10 ns (1 ~ 10クロック)	K ~ MBytes
レジスタ	~ 1 ns (1 クロック)	32 ~ 128 Bytes

よく使うものは速い記憶装置に置きたい．しかしサイズは限られている

デスクワークからの類推

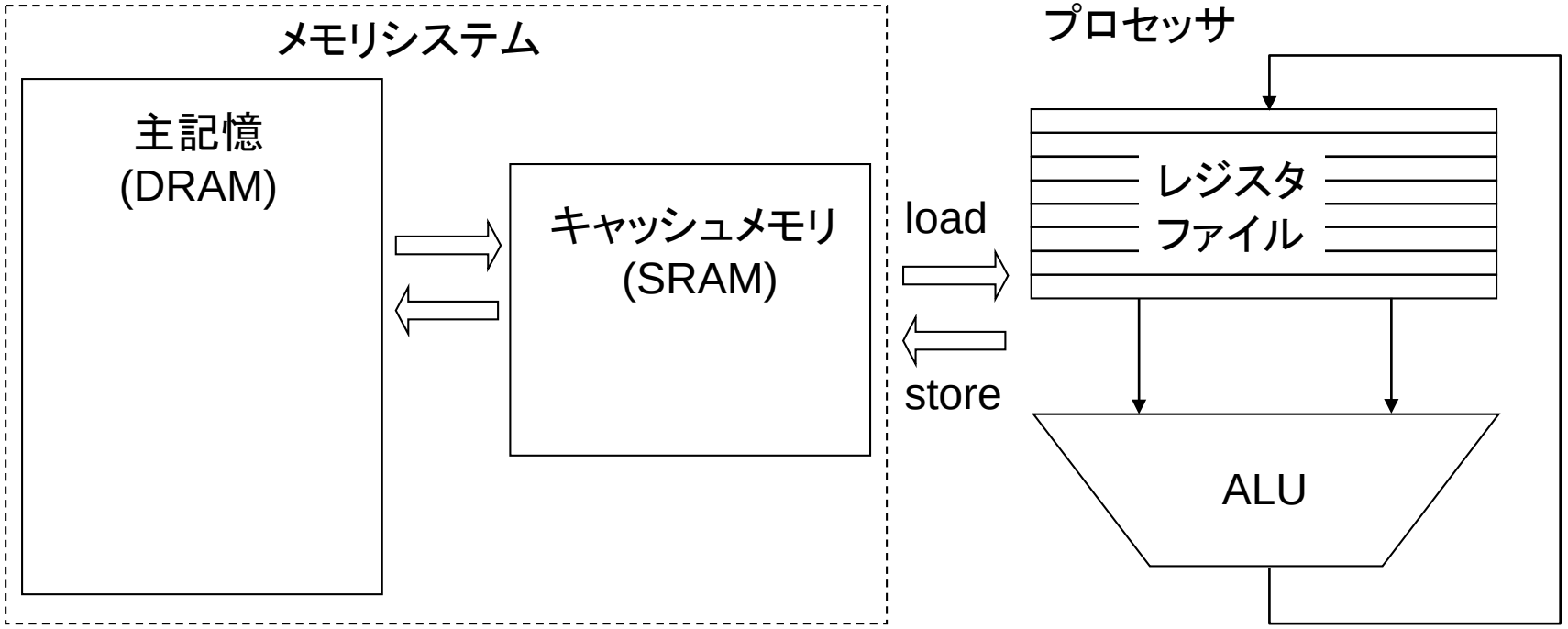


- 机のサイズは限られているので，適宜，室内のファイルキャビネットや，社内の資料室に書類を取りにいかなくてはならない
- 新しい書類が必要になったら，当面不要なものをキャビネットまたは資料室に仕舞わなくてはならない．
- さてどうするか?

自然な戦略:

- **一度使った資料はまたすぐ使う可能性が高い**ので，すぐにしまわずに机に置いておく(あるいは資料室まで戻さずにキャビネットに置いておく)
- **関連する資料がすぐ必要になる可能性が高い**ので，ある資料が必要なときには，それを綴じてあるファイルブックごと机に持ってくる

キャッシュメモリ



キャッシュメモリの制御は，以下の経験則を利用して自動的に行われる

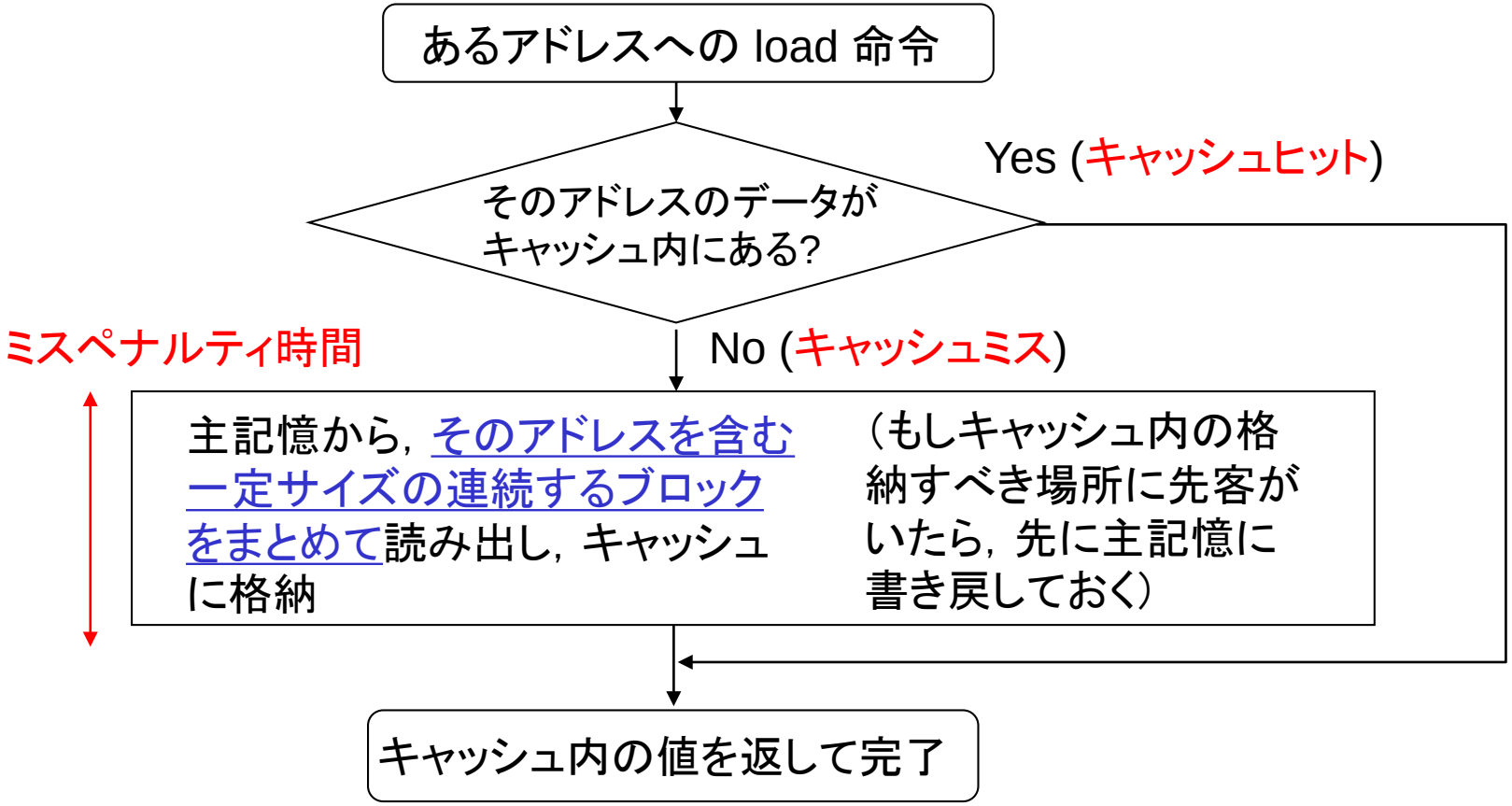
- **時間的局所性**

  あるデータがアクセスされる場合，近いうちにその同じデータが再度アクセスされる可能性が高い

- **空間的局所性**

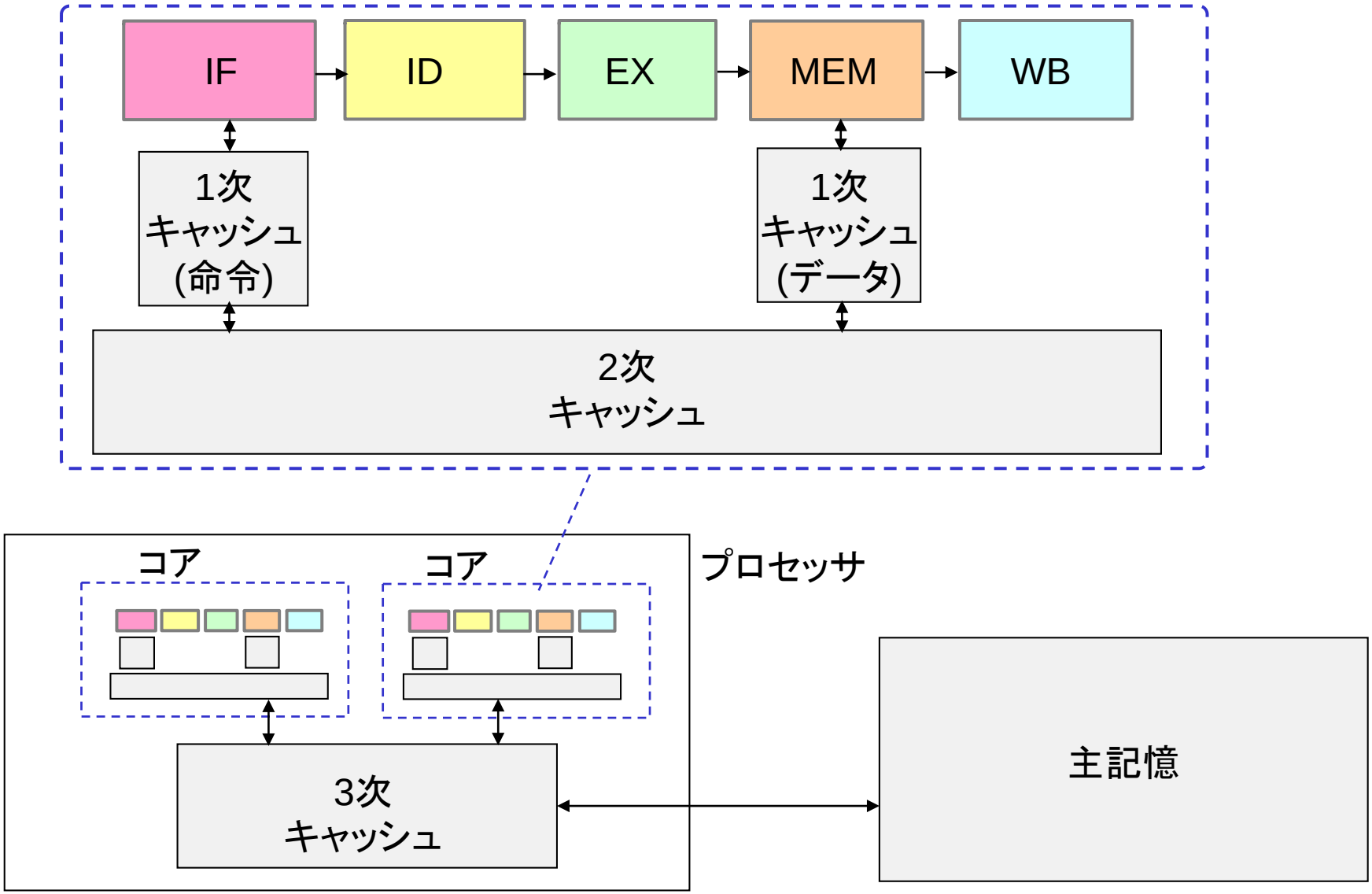
  あるデータがアクセスされた場合，その周囲の値もアクセスされる可能性が高い

キャッシュメモリの動作例



平均メモリアクセス時間
＝ ヒット時間 ＋ キャッシュミス率 × ミスペナルティ時間

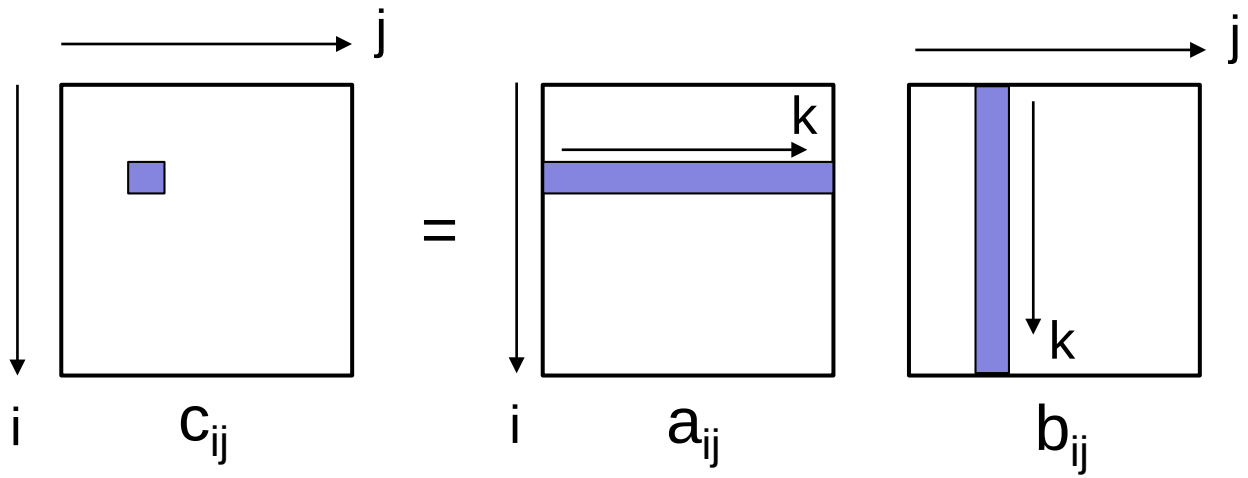
キャッシュメモリの階層構成例



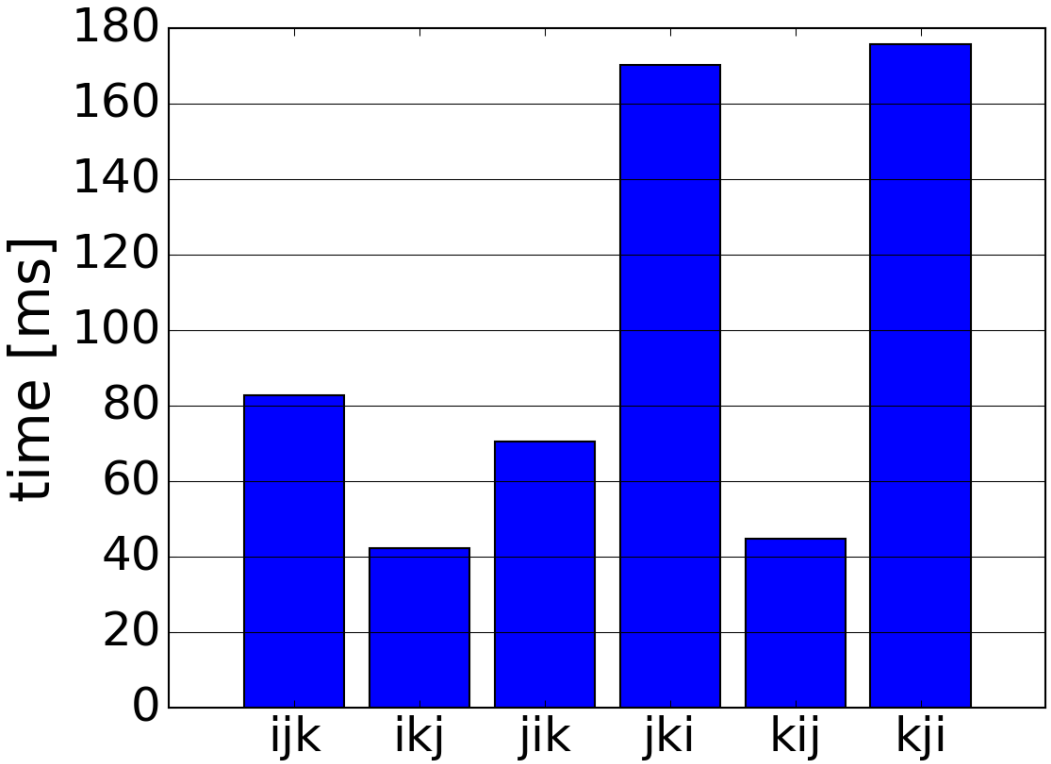
キャッシュメモリの効果: 行列積の例

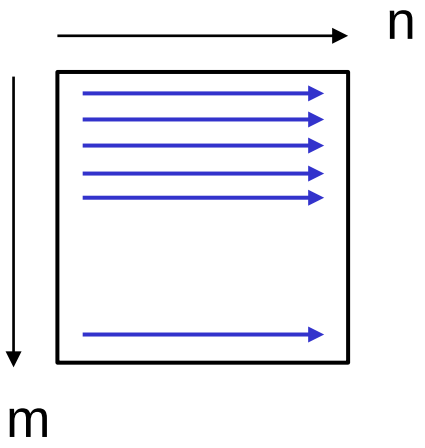
```
#define N 500
double a[N][N], b[N][N], c[N][N];

for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        for (int k = 0; k < N; k++) {
            c[i][j] += a[i][k] * b[k][j];
        }
    }
}
```

この3行の並び順によって実行時間が大きく変わる



Core i7-7600U 並列処理なしの場合





x[m][n] の要素の
メモリ上での並び順

```
c[i][j] += a[i][k] * b[k][j];
```

最も内側のループで i や k が変化すると不連続なメモリアクセスが発生する

練習問題 (1)

1. ヒット時間が 1 ns, ミスペナルティ時間が 20 ns のメモリシステムを考える. キャッシュミス率が 5 % のときの平均メモリアクセス時間を求めよ.

2. 1のシステムにおいて, 平均メモリアクセス時間を 1.5 [ns] にするために必要なキャッシュミス率を求めよ.

3. 一般にキャッシュメモリのサイズを大きくするとキャッシュミス率は下がるが, ヒット時間は増大する傾向にある. ある計算機の設計において, キャッシュサイズを 2 倍にすることによってキャッシュミス率が 5 % から 4 % に改善することがわかった. これによって平均メモリアクセス時間を短縮できるためには, ヒット時間の増大はどの程度に抑えられている必要があるか述べよ. ただしミスペナルティ時間は変更前のヒット時間の 20 倍で, キャッシュサイズに依存しないとする.

練習問題 (2)

以下の各項の説明が正しいかどうか判別せよ.

A) DRAM は，キャパシタの自然放電によって時間が経つとデータが失われるので，一時的な記憶であるキャッシュメモリへの利用に適している.

B) DRAM は時間的局所性を持つため，キャッシュメモリへの利用に適している.

C) SRAM は不揮発性メモリであり，DRAM は揮発性メモリである.

D) DRAM は制御が複雑であるため回路規模が大きく，SRAM に比べて大容量化が難しい.

E) キャッシュメモリを備える計算機の命令セットには，主記憶用の load/store 命令とは別にキャッシュメモリ用の load/store 命令が必要である.

F) キャッシュメモリの制御は自動で行われるため，プログラマはその存在を意識しなくても常に最高の性能を得ることができる.

解答例 (1)

平均メモリアクセス時間 = ヒット時間 + キャッシュミス率 × ミスペナルティ時間

1. $1 + 5 \times 10^{-2} \times 20 = 2$ [ns]

2. $1 + p \times 10^{-2} \times 20 = 1.5$ を p について解いて，$p = 2.5$ [%]

3. 変更前，変更後の平均メモリアクセス時間を t_{ma1}，t_{ma2}，同じくヒット時間を t_{hit1}，t_{hit2} と書くと，

$$t_{ma1} = t_{hit1} + 5 \times 10^{-2} \times 20\, t_{hit1}$$
$$t_{ma2} = t_{hit2} + 4 \times 10^{-2} \times 20\, t_{hit1}$$

$$t_{ma2} - t_{ma1} = t_{hit2} - t_{hit1} - 1 \times 10^{-2} \times 20\, t_{hit1}$$
$$= t_{hit2} - 1.2\, t_{hit1}$$

よって 1.2 倍までの増大は許容できる．(逆に言うと，ヒット時間が それ以上増大してしまうなら，ミス率改善の努力は無駄になる)

解答例 (2)

すべて誤り