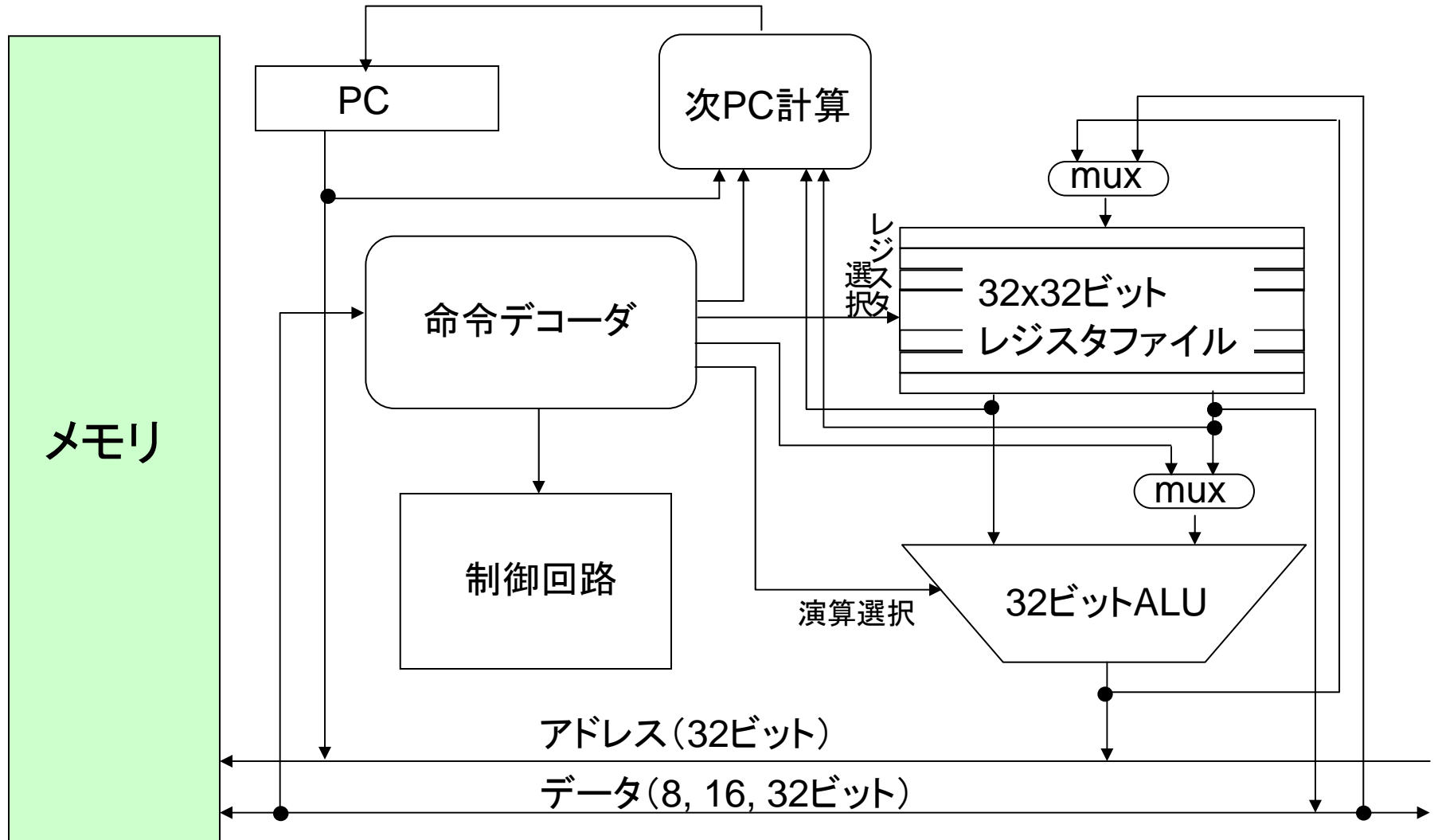
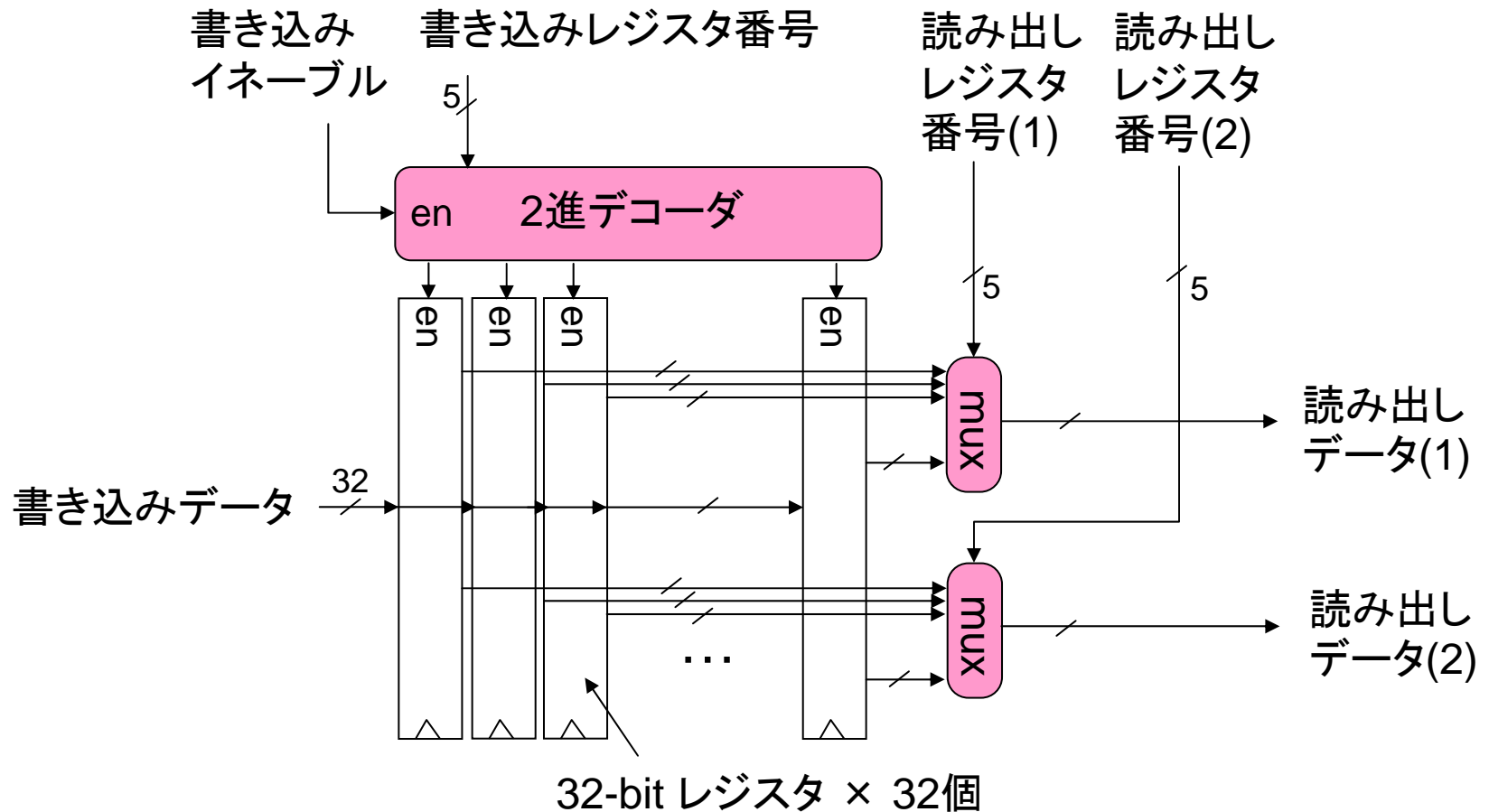

メモリシステム

(復習) MIPSの構造



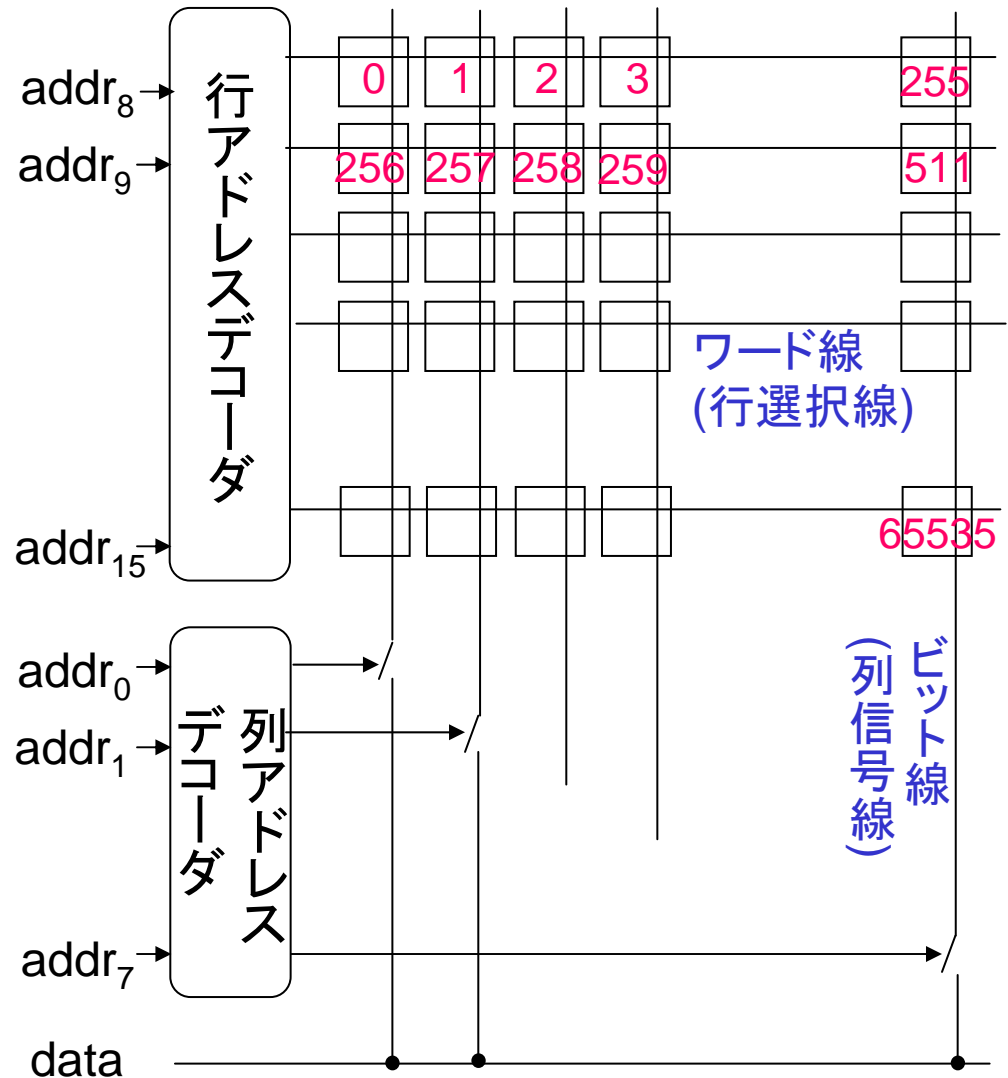
(復習) レジスタファイル (32 × 32ビット, 1入力2出力)

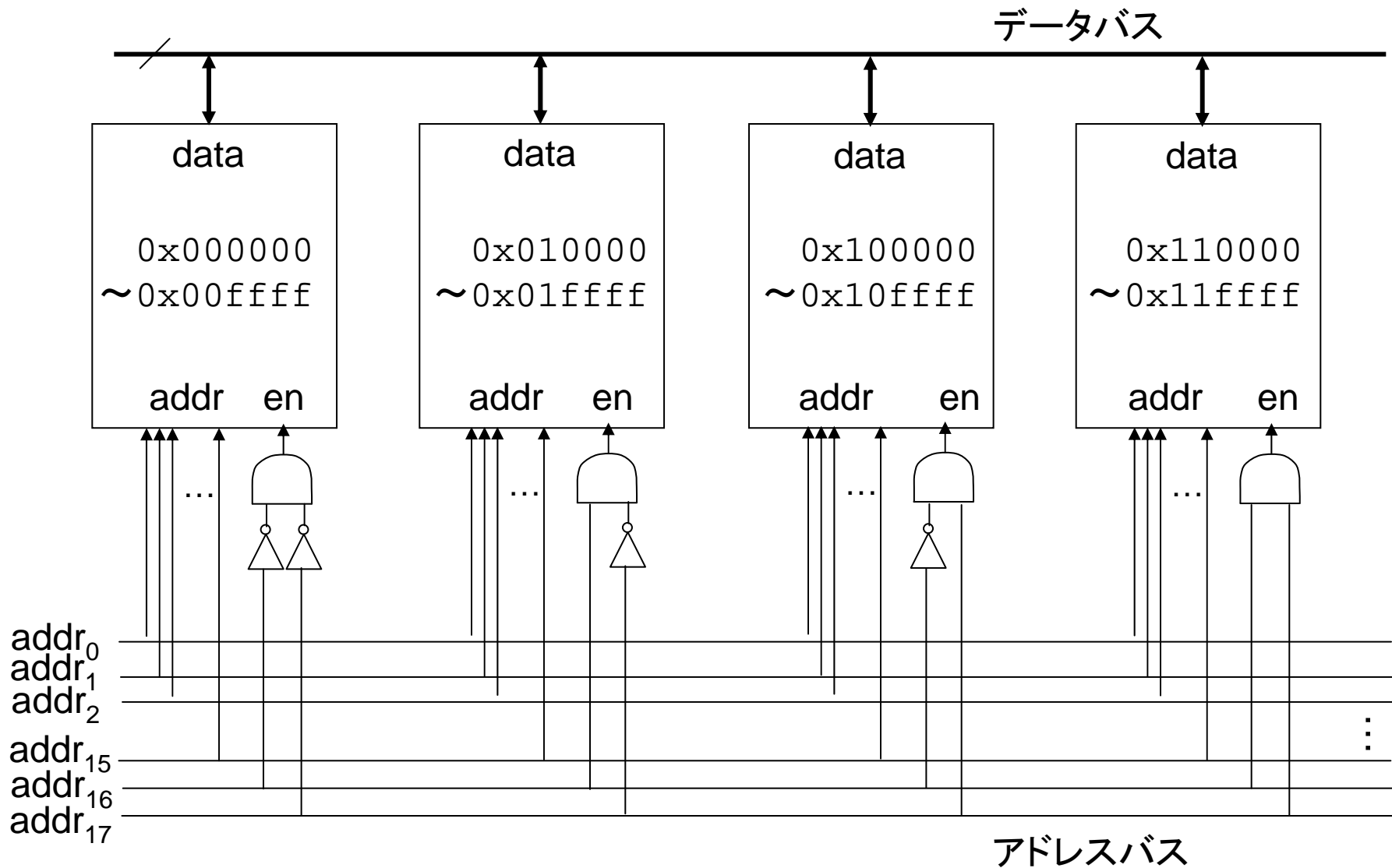


この構造のまま, 単純に容量を拡大するのは困難
(デコーダ・マルチプレクサが肥大化するため)

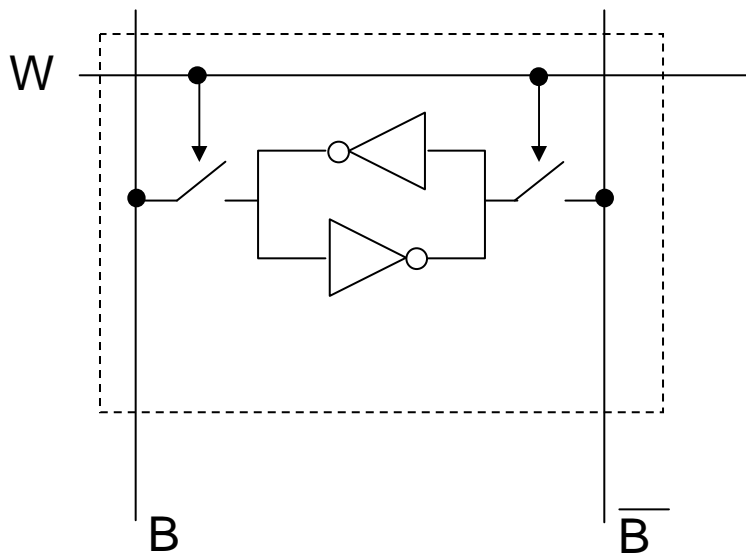
メモリの構造

- セル(記憶素子)を2次元マトリクス状に配置して, アクセスを縦・横に分解
- 読み出し・書き込み回路を共通化



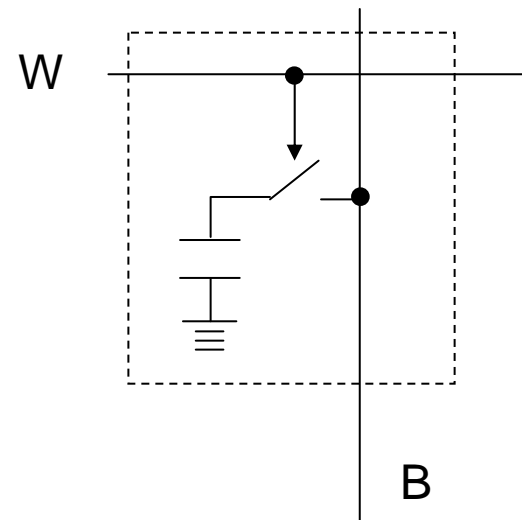


メモリセルの構造



Static Random Access Memory (SRAM)

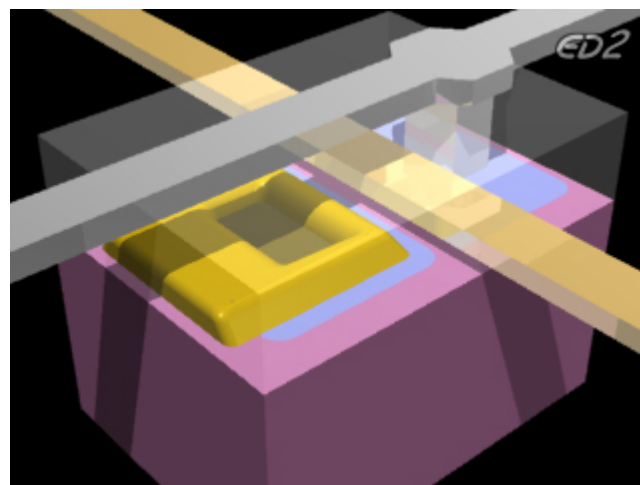
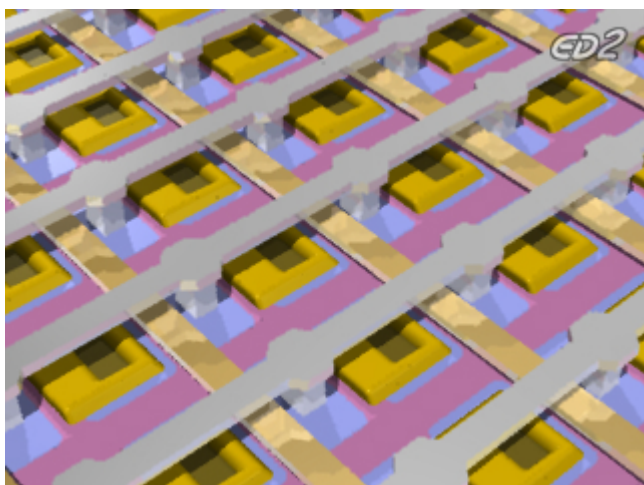
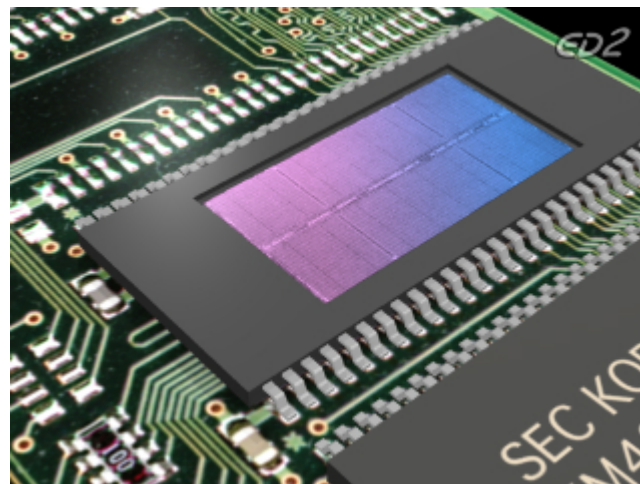
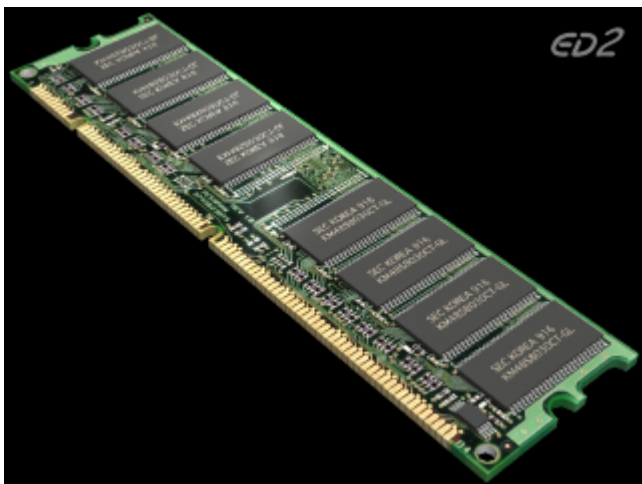
- 原理的にはフリップフロップと同じ
- NOTゲートが能動的に電流を供給してビット線を駆動する
- 1ビットあたりトランジスタ 6 個



Dynamic Random Access Memory (DRAM)

- キャパシタが充電されていれば 1, 放電されていれば 0
- ビット線の電位は, キャパシタからのわずかな電荷で微小に変化する
- 時間とともに電荷が漏れる
- 1ビットあたりトランジスタ 1 個

メモリの動作 (DRAMの場合)

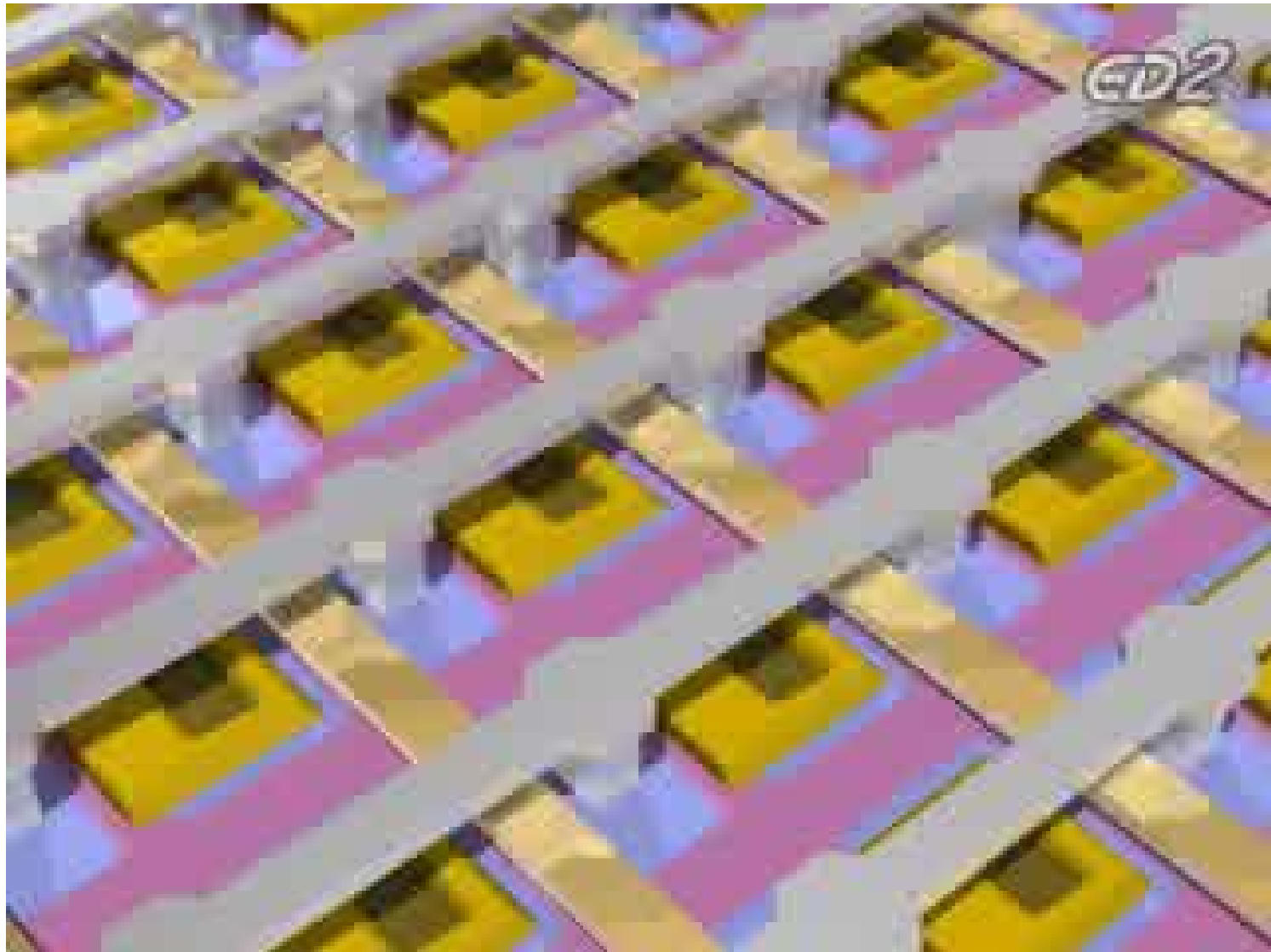


<http://www.sugilab.net/jk/joho-kiki/>
(1400 処理装置 → 1404 メモリのしくみ)

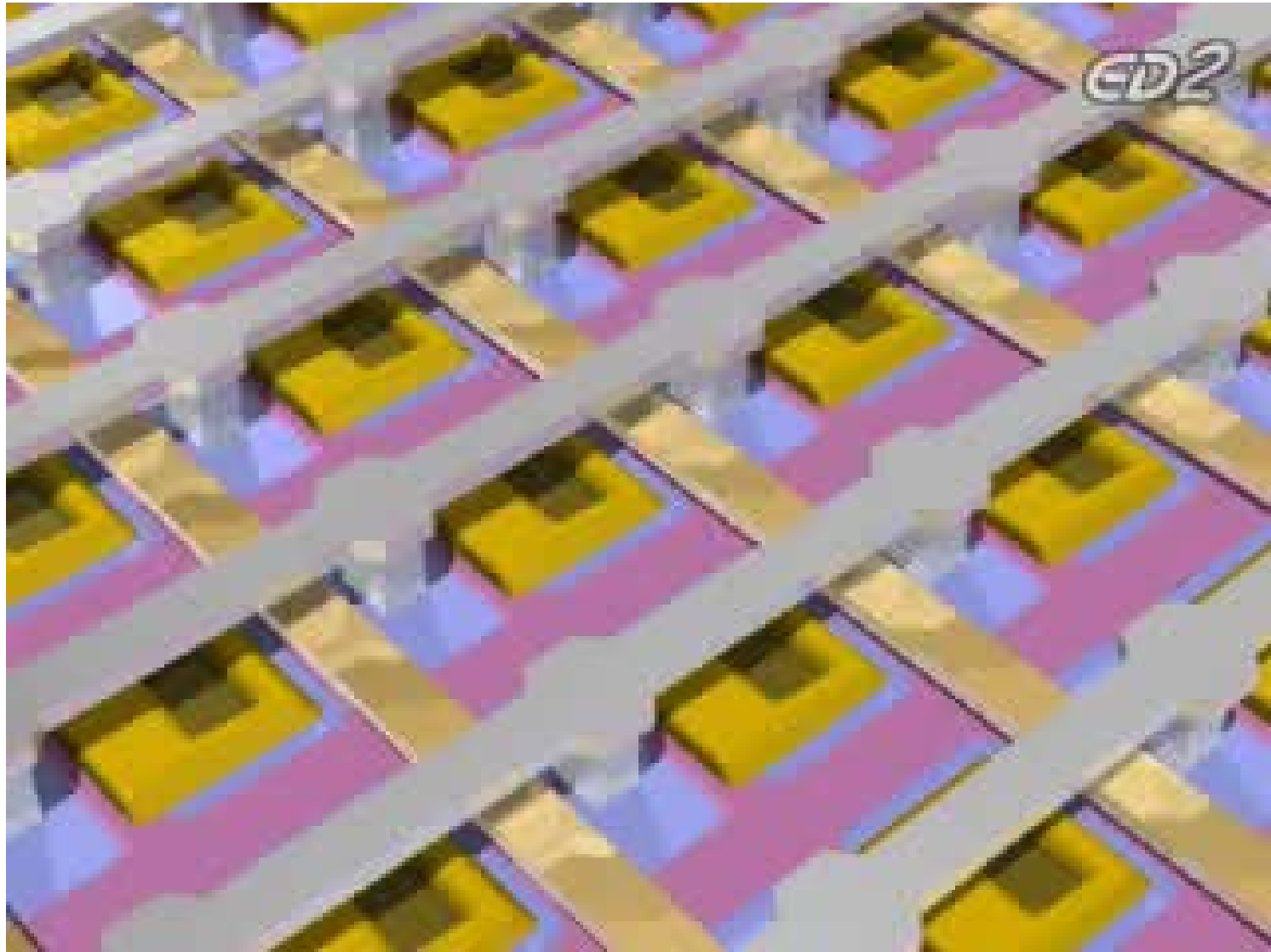
構造



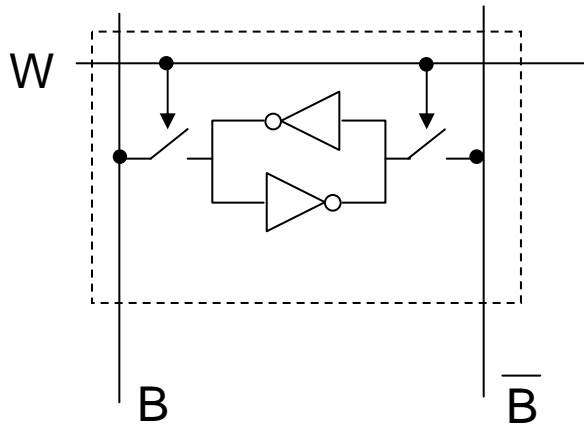
書き込み



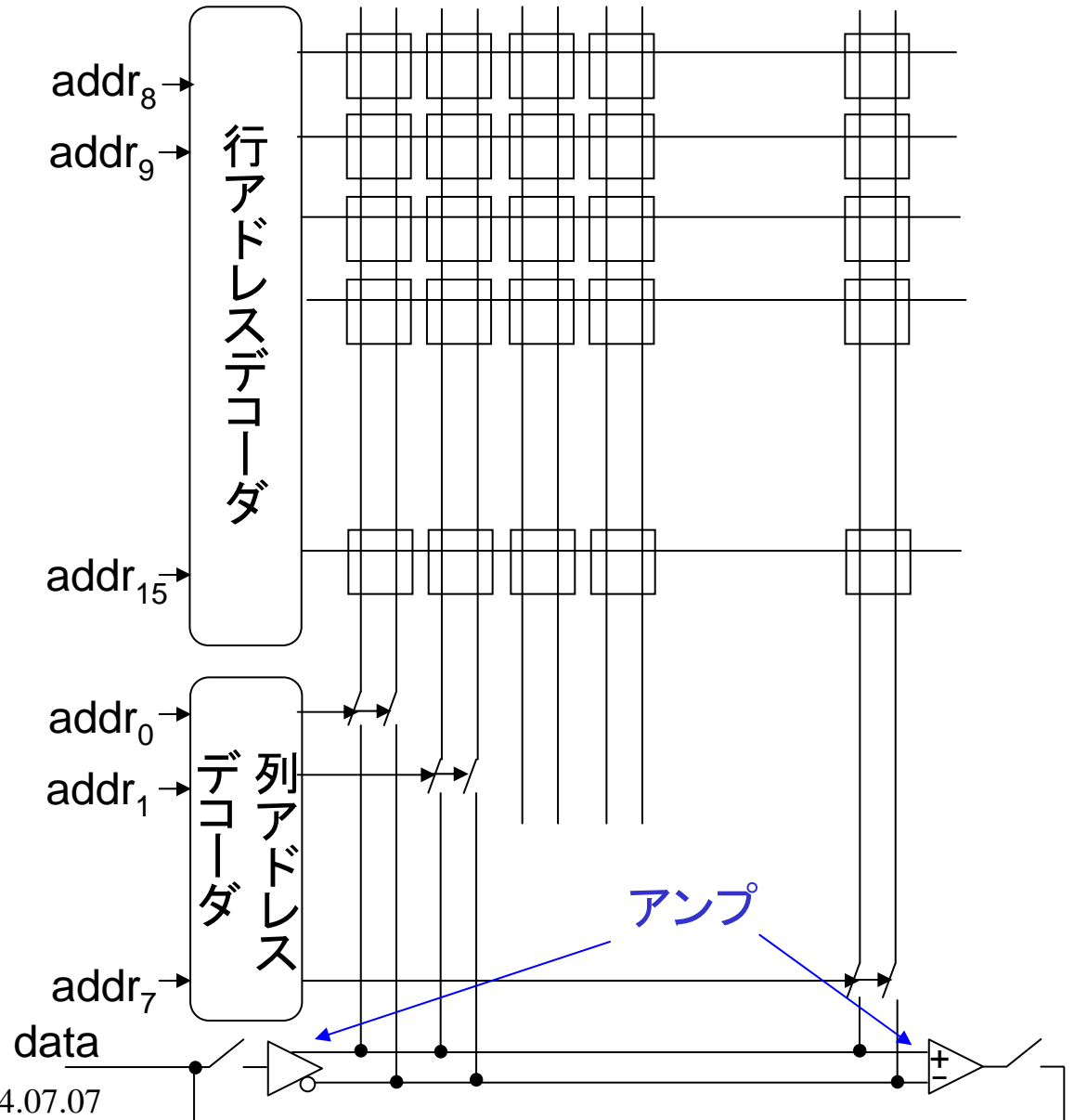
読み出し



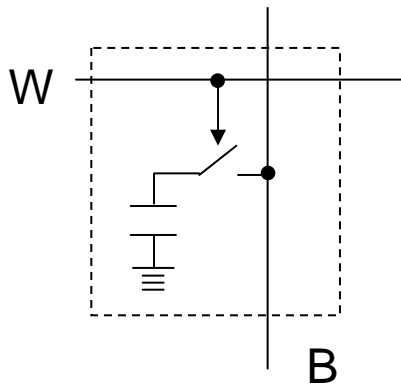
SRAM



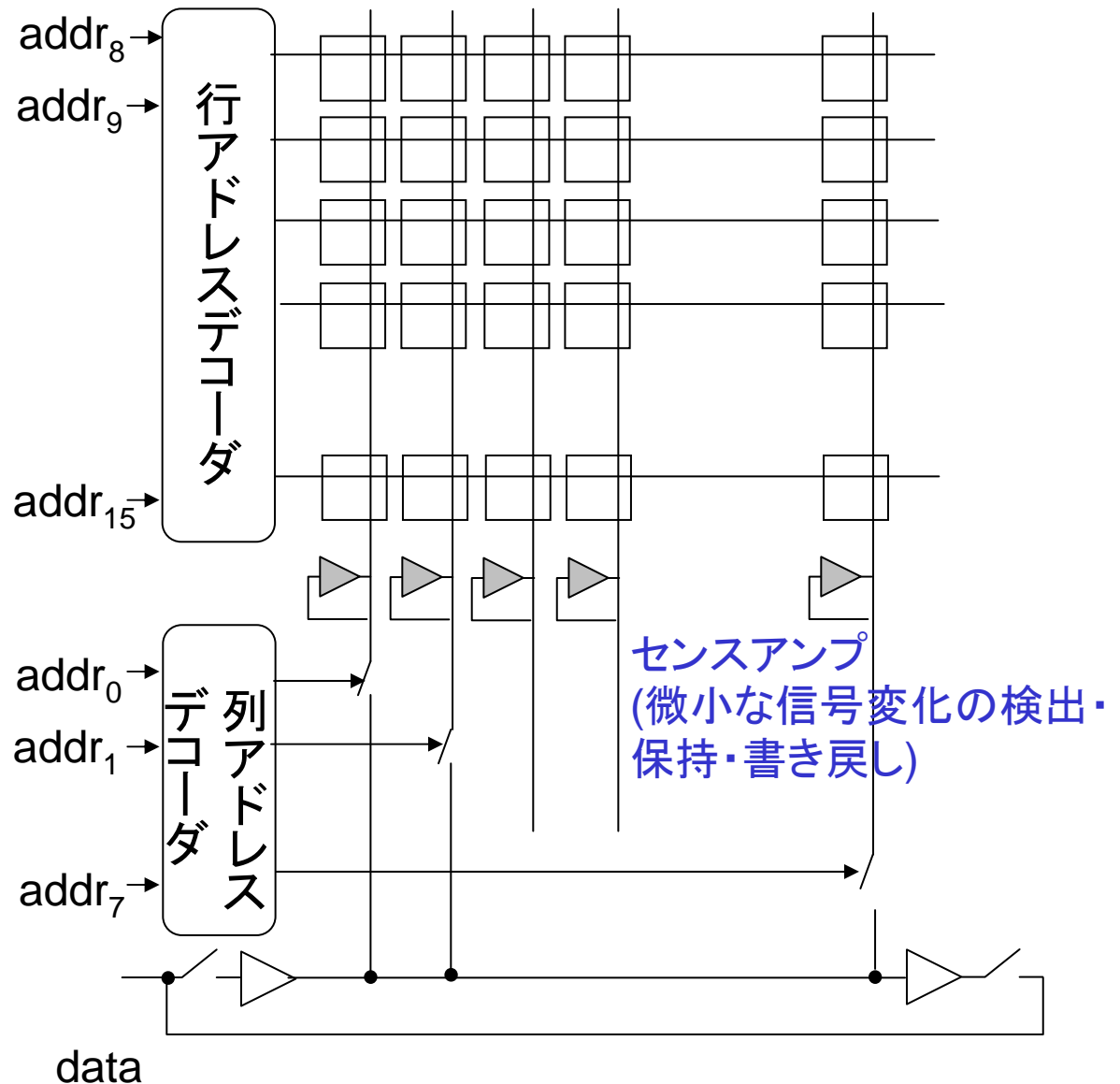
- 読み出し時には、ビット対線の電位差を増幅して値を得る
- 書き込み時には、セルのNOTゲートよりも強くビット線を駆動して記憶内容を上書きする



DRAM



- 読み出し時は、セルから流れ込む電荷によるビット線の電位の微小変化をセンスアンプが検知して増幅・保持する。選択列の値が読み出される。
- 書き込み時は、読み出し時と同じ動作の後、選択列のビット線のみ入力電圧で上書きする。



SRAM vs DRAM

SRAM

- 1セルの回路が大きい
- 制御が比較的簡単
- 記憶内容は、電源が入っている限り安定
- よって、速いが小容量

DRAM

- 1セルの回路が小さい
- 制御が比較的複雑
- 時間が経つと記憶が消える (リフレッシュと呼ばれる再書き込み動作を数ミリ秒に1回行う必要がある)
- よって、遅いが大容量

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



資料室

ファイル
キャビネット

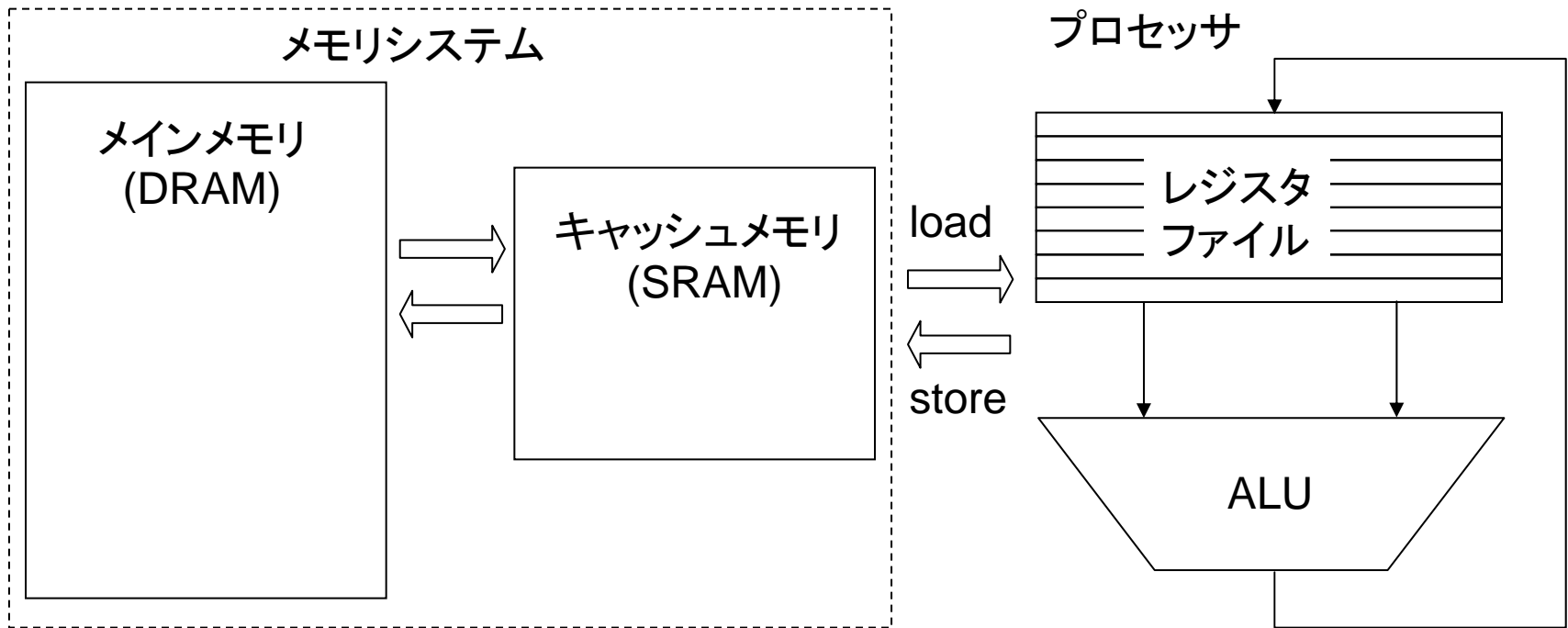
机

- 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

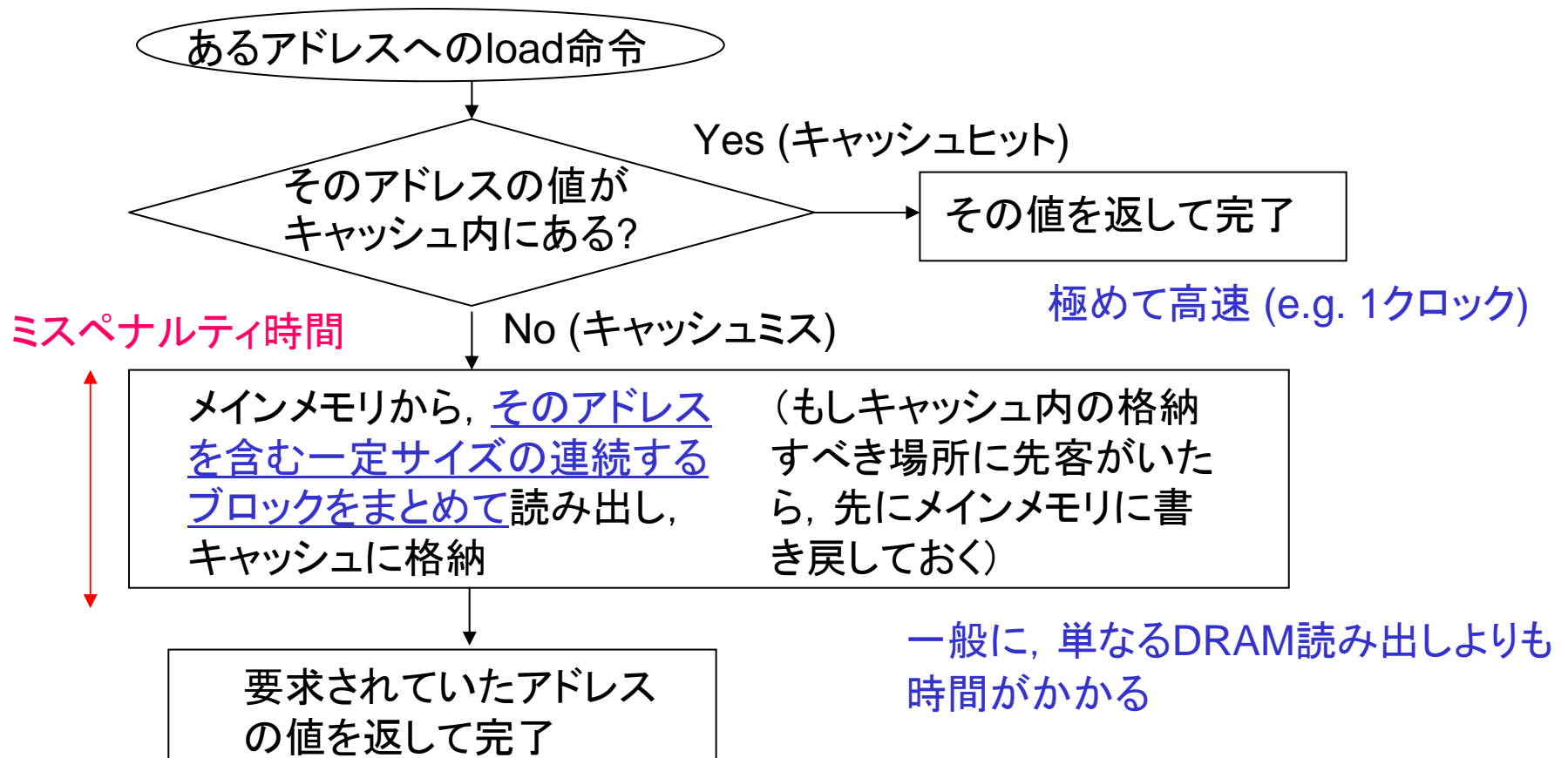
- **時間的局所性**

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

メモリの分類

ランダムアクセスメモリ vs シーケンシャルアクセスメモリ

- 任意の順序で読み書きできるものを RAM (Random Access Memory) と呼ぶ
- 最近の「メモリ」はほぼ例外なくランダムアクセス可能

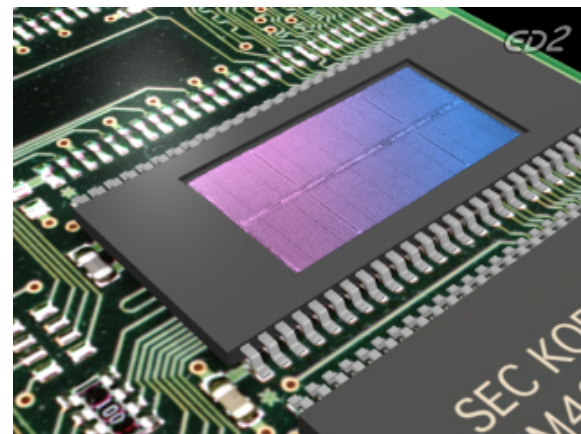
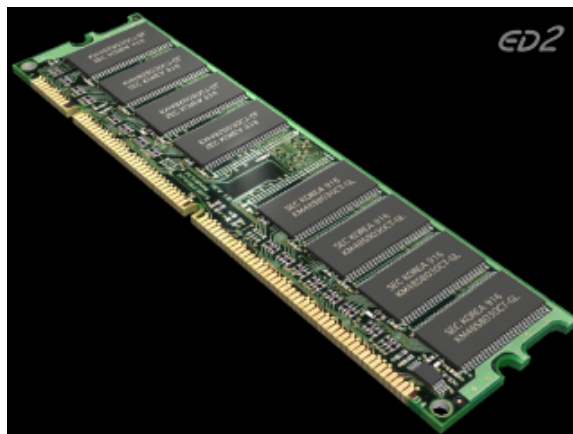
揮発性メモリ vs 不揮発性メモリ

- 電源を切るとデータが消えるのが揮発性メモリ
- 不揮発性メモリのうち、主に読み出しに用いるものを ROM (Read Only Memory) と呼ぶ
 - マスクROM (半導体製造時に内容を決めてしまう)
 - PROM (Programmable ROM): 書き込み可能
 - EPROM (Erasable PROM): 消去も可能
 - UV-EPROM: 紫外線で消去
 - EEPROM: 電氣的に消去 (e.g. 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)

※ 「メモリ」という名前でも実は「補助記憶装置」の場合がある (e.g. USBメモリ)

DRAMモジュール

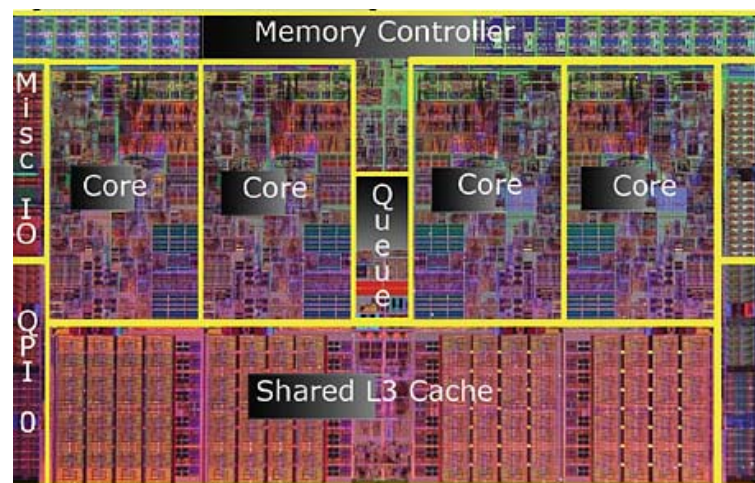


<http://www.sugilab.net/jk/joho-kiki/>

SRAM (キャッシュメモリ) 内蔵プロセッサ



http://ja.wikipedia.org/wiki/Intel_Core_i7



<http://www.atmarkit.co.jp/fsys/zunouhoudan/102zunou/corei7.html>

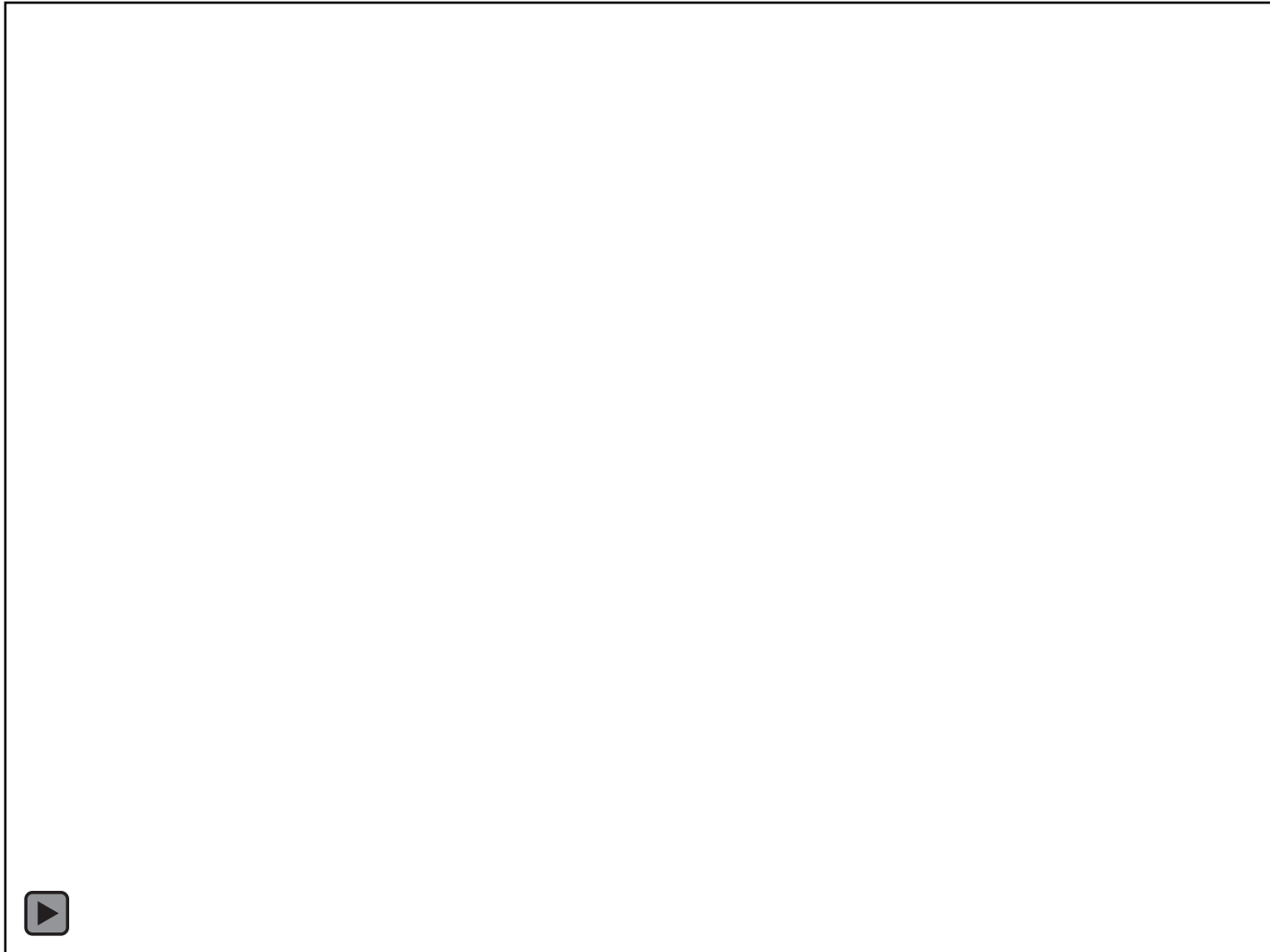
ファミリーコンピュータ用ROMカートリッジ(ロムカセット)



[http://ja.wikipedia.org/wiki/
ファイル:Famicom_ROM_cassette.jpg](http://ja.wikipedia.org/wiki/ファイル:Famicom_ROM_cassette.jpg)

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

おきのどくですが ぼうけんのしょ1ばんは きえてしまいました



http://www.youtube.com/watch?v=gQf_QTsvXCU

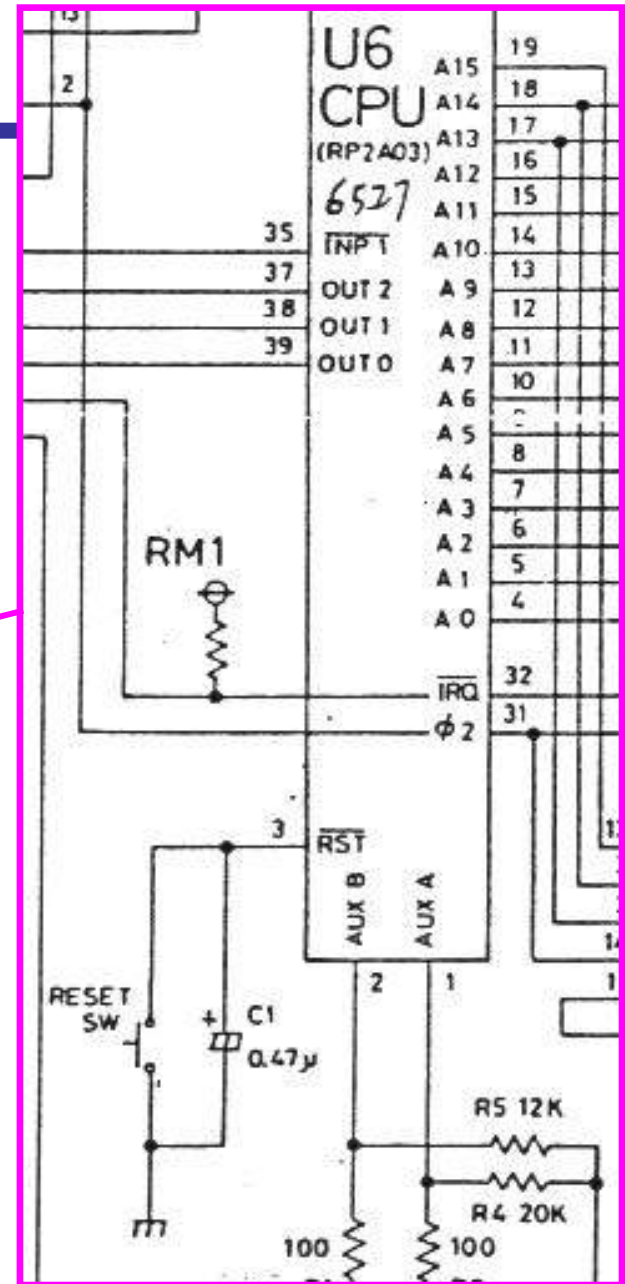
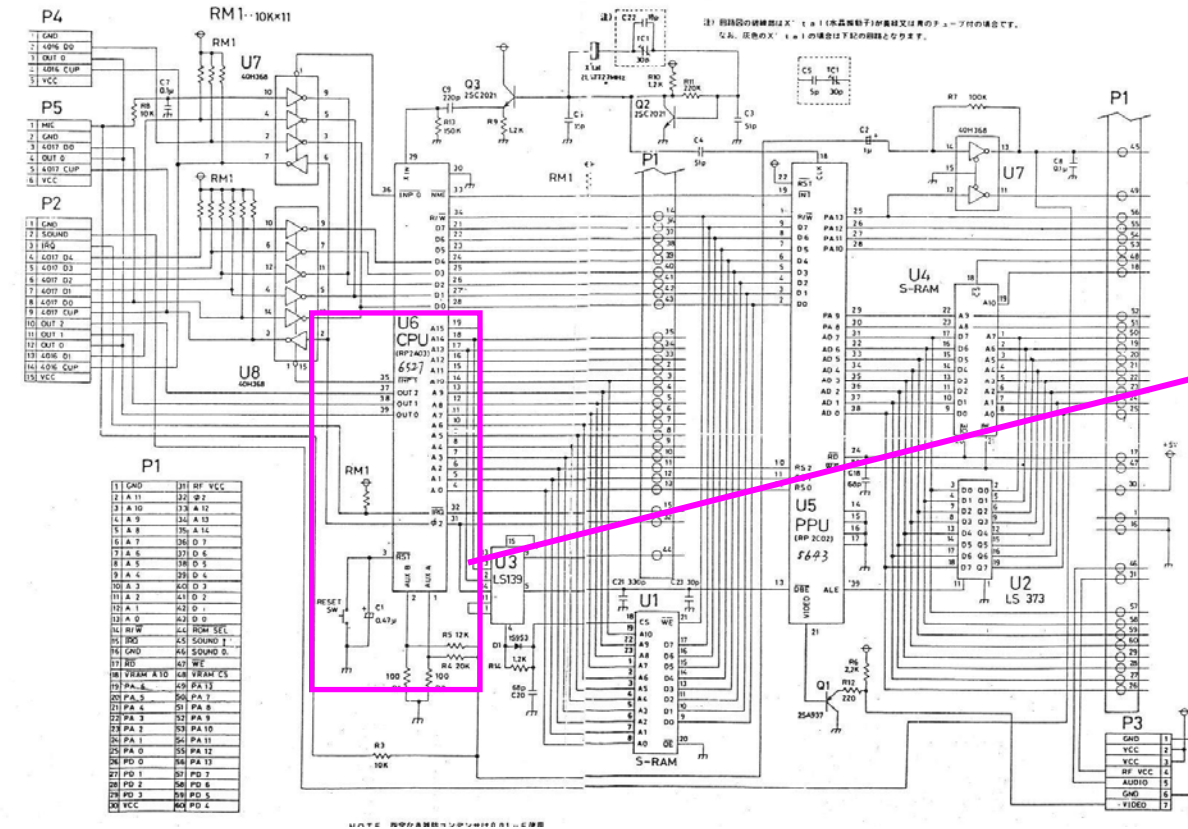
例題(おまけ, あまり本題と関係ない)

任天堂のゲーム機 ファミリーコンピュータ (1983年発売) では, ソフトウェアは ROMカートリッジの形で提供されていた. 基本的にはマスク ROM を本体に直結する構造であるが, 「森田将棋」(セタ, 1987年) 以降, カートリッジ内に SRAM と電池も内蔵し, ゲームの進行状況を記録できる機能(バッテリーバックアップ)を持つものがしばしば発売された.

それらのゲームでは, ゲーム終了時に本体のリセットボタンを押しながら電源スイッチを切るように指示されることが多かった. その理由として正しいと思われるものを選び, 理由とともに答えよ. (次ページの回路図を参考にせよ)

1. SRAM を保護する機能がリセットボタンに割り当てられていたから.
2. リセットボタンを押している間はカートリッジへの電源供給が遮断される構造になっていたので, 電源スイッチからのノイズが SRAM に伝わるのを防ぐことができたため.
3. リセットボタンを押している間はプロセッサが初期状態に固定されるので, SRAM へのアクセスが発生しなくなるため.

CPU基板回路図



<http://nesdev.icequake.net/>
http://nesdev.icequake.net/Ntd_8bit.jpg

解答例

1983年のゲーム機本体発売当時にバッテリーバックアップの仕組みが考慮されていたと考えるのは合理的ではないため、1は除外できる。

回路図を参照すると、リセットボタンに電源供給を遮断する機能は無いことがわかる。したがって2も除外できる。(この説はウェブ上でまことしやかに流れているものだが、デマであると結論できる)

回路図によると、リセットボタンを押している間はプロセッサのRST信号が0になる。その間、有限状態機械が初期状態に留まり続け、メモリアクセスが発生しないようにしていると考えるのが妥当である。

練習問題

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 0.5% から 0.3% 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.

解答例

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

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,

$$t_{ma1} = t_{hit1} + 0.5 \times 10^{-2} \times 20 \times t_{hit1}$$

$$t_{ma2} = t_{hit2} + 0.3 \times 10^{-2} \times 20 \times t_{hit1}$$

$$t_{ma2} - t_{ma1} = t_{hit2} - t_{hit1} - 0.2 \times 10^{-2} \times 20 \times t_{hit1}$$

$$= t_{hit2} - t_{hit1} \times 1.04$$

よって 1.04 倍までの増大は許容できる. (逆に言うと, ヒット時間が 4 % 増大してしまうなら, ミス率改善の努力は無駄になる)