

コンピュータ基礎
記憶階層とキャッシュ
テキスト第10章

天野英晴

hunga@am.ics.keio.ac.jp

記憶システム

- 膨大な容量を持ち、アクセス時間(読み出し、書き込み)が短いメモリが欲しい！

しかし

- 容量の大きい(ビット単価が安い)メモリは遅い
- 高速なメモリは容量が小さい

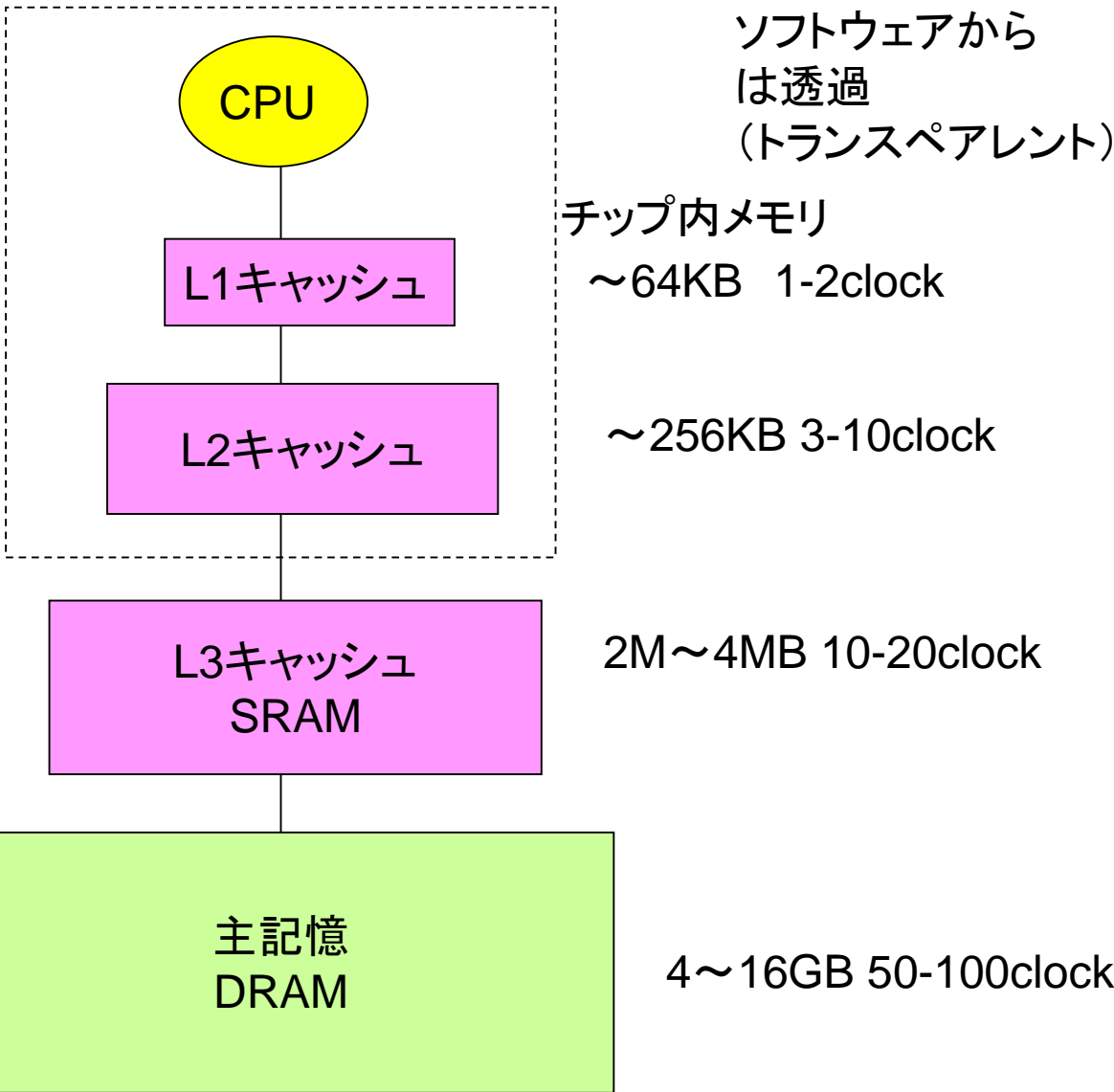
お金にモノを言わせて高速なメモリをたくさん揃えても大容量化の段階で遅くなってしまう

- そこでアクセスの局所性(Locality)を利用
 - 時間的局所性(Temporal Locality)
 - 一度アクセスされたアドレスは近いうちにまたアクセスされる
 - 空間的局所性(Spatial Locality)
 - 一度アクセスされたアドレスに近い場所がまたアクセスされる

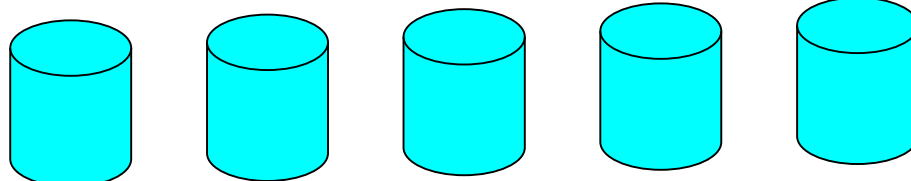
記憶の階層

高速小容量の
CPUの近くに置き
よく使うデータを入れておく

そこになければより遅い
大容量メモリに取りに行く



OSが管理



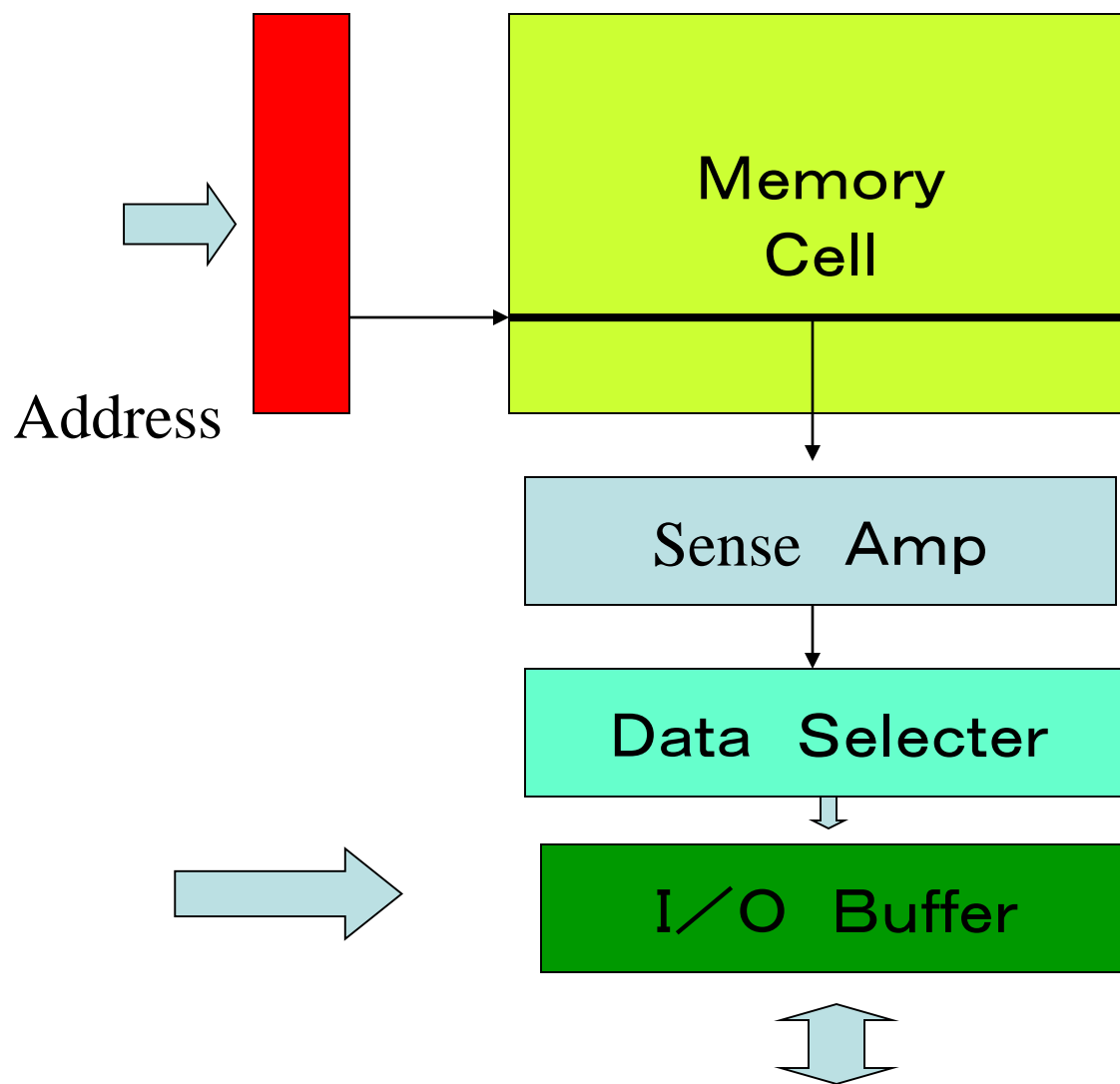
補助記憶（2次記憶）
μ-msecオーダー
数百GB

半導体メモリの分類

- RAM (RWM): 揮発性メモリ
 - 電源を切ると内容が消滅
 - SRAM(Static RAM)
 - DRAM(Dynamic RAM)
- ROM(Read Only Memory): 不揮発性メモリ
 - 電源を切っても内容が保持
 - Mask ROM 書き換え不能
 - PROM(Programmable ROM) プログラム可
 - One Time PROM 一回のみ書き込める
 - Erasable PROM 消去、再書き込み可能
 - UV EPROM (紫外線消去型)
 - EEPROM (電氣的消去可能型) **FLASH Memory**

メモリの基本構造

記憶要素により
性質が定まる

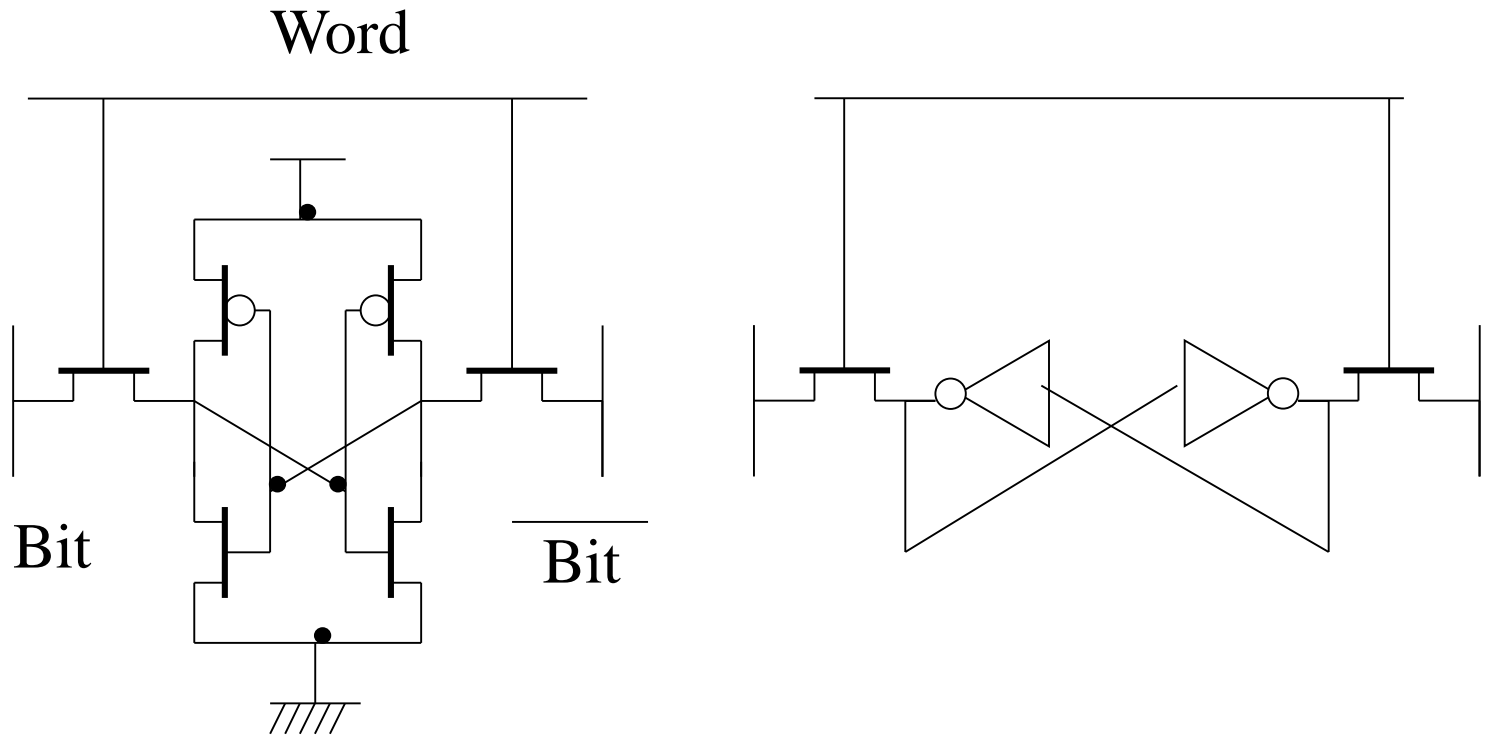


I/Oバッファは
3ステート

SRAM (Static RAM)

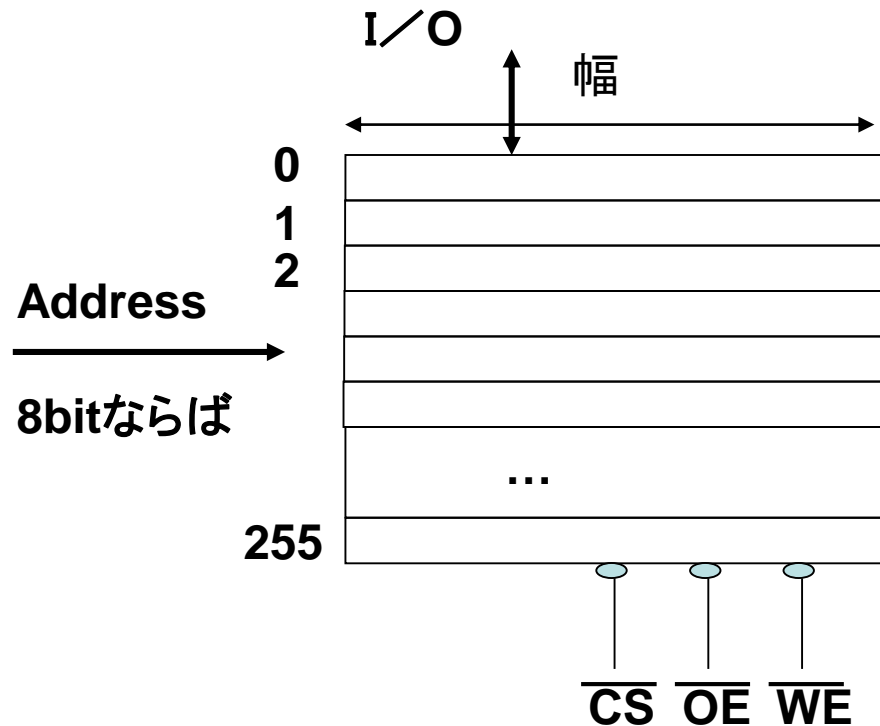
- 非同期式SRAM
 - 古典的なSRAM
 - クロックを用いない
 - 現在も低電力SRAMシリーズなどで用いられる
- 連続転送機能を強化したSSRAM (Synchronous SRAM)が登場、高速大容量転送に用いられる
 - 8Mbit/Chip-64Mbit/Chip程度
 - TSOP (Thin Small Outline Package)やBGA(Ball Grid Array)を利用

SRAM型のメモリセル構造



最も基本に忠実な6トランジスタ方式:安定なのでよく用いられる

非同期SRAM



メモリは幅wbit, 深さ 2^n
この例はw=16, n=8
深さ $2^8=256$
(本当はもっとずっと多数のデータを格納する)

制御端子は3本
CS: チップ全体を制御
OE: 読み出し
WE: 書き込み
いずれもアクティブL

非同期SRAMのモデル

RAMの容量

- 深さ×幅
- 右の表に幅を掛ければ全体の容量が出る
- 省略した言い方でも十分(端数を覚えている人は少ない)

アドレス 本数	容量	省略した言い方
8	256	256
10	1024	1K
12	4096	4K
16	65536	64K
18	262144	256K
20	1048576	1M
24	16777216	16M
28	26835456	256M
30	1073741824	1G
32	4204067296	4G

DRAM (Dynamic RAM)

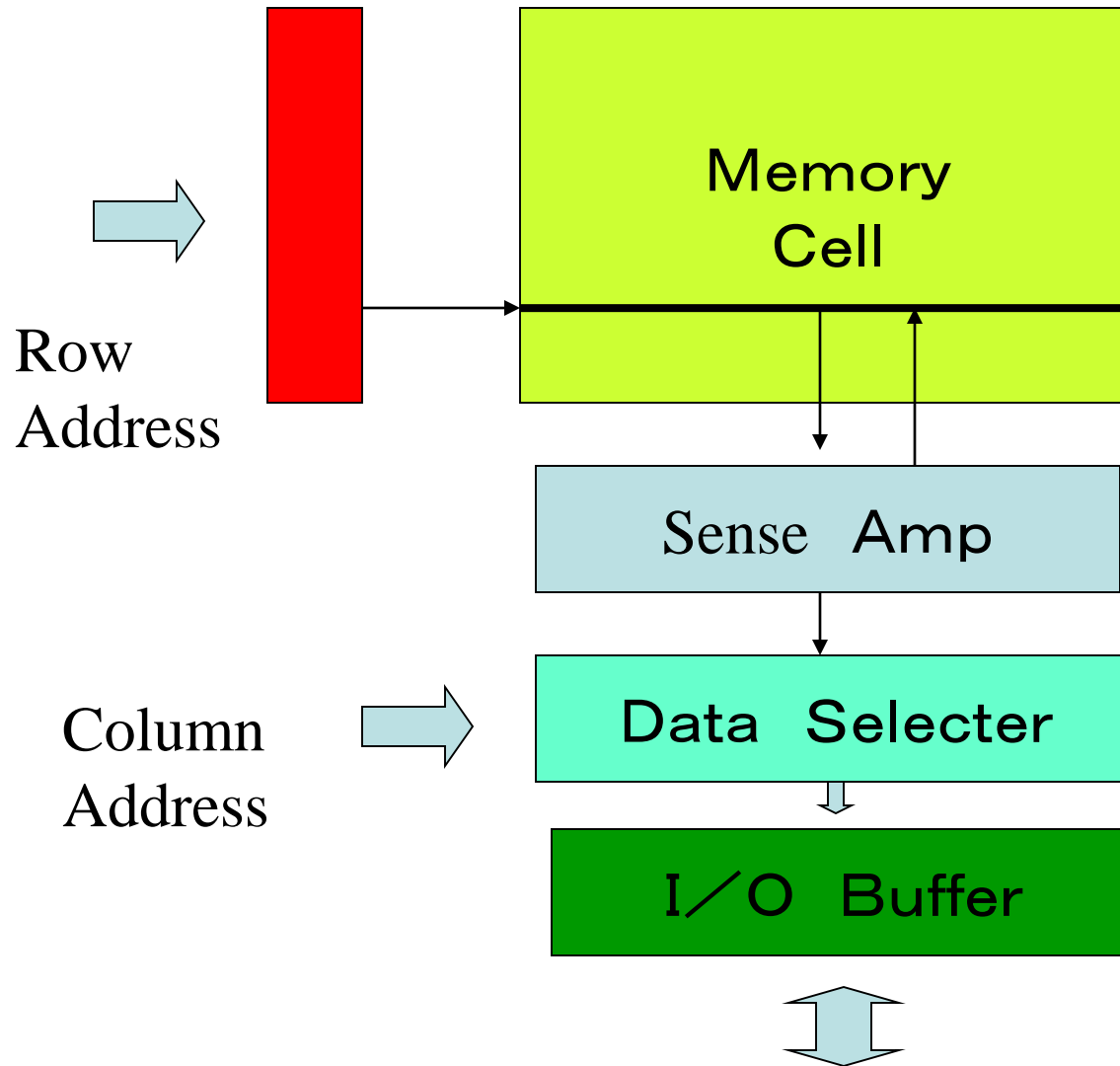
- 記憶はコンデンサ内の電荷によって行う
- リフレッシュ、プリチャージが必要
- 256Mbit/Chipの大容量
- 連続転送は高速
- SDRAM (Synchronous DRAM) の普及
- DDR-SDRAMの登場
 - DDR2 → DDR3

DDR-SDRAMカードの例

- 下は1GBでやや小さい。今は4GB－8GBのカードが良く使われる



DRAMの構造



古典的なDRAMの利用法

アドレス

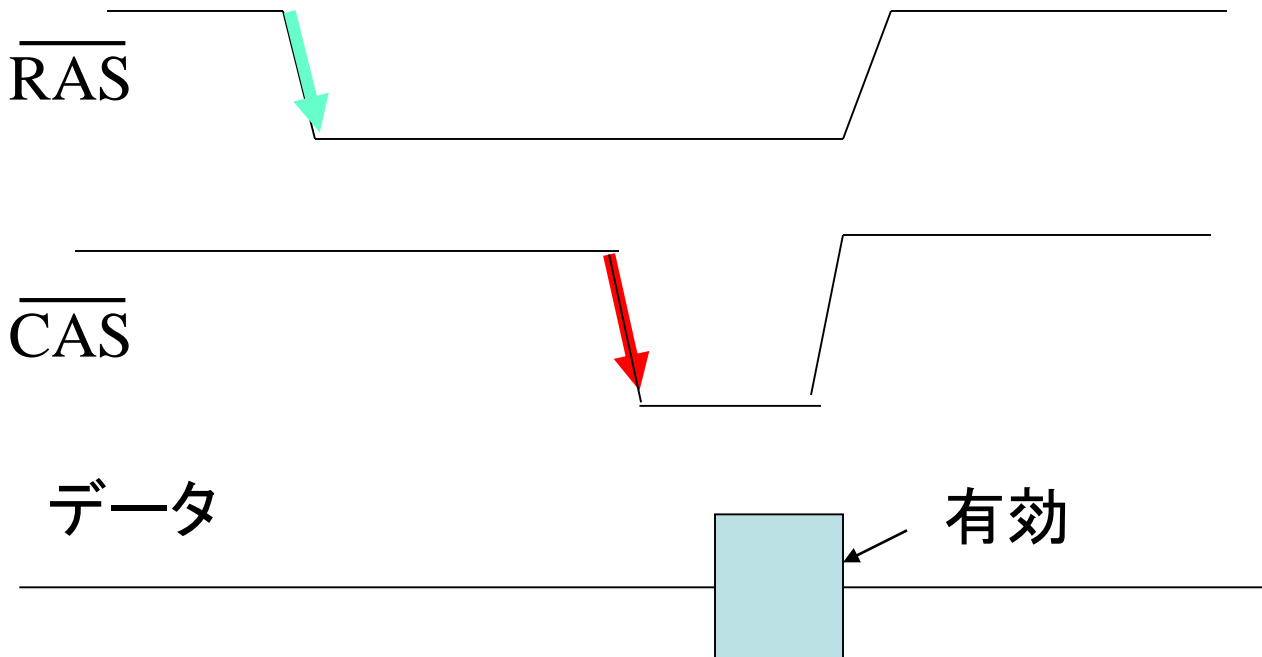


$\overline{\text{RAS}}$

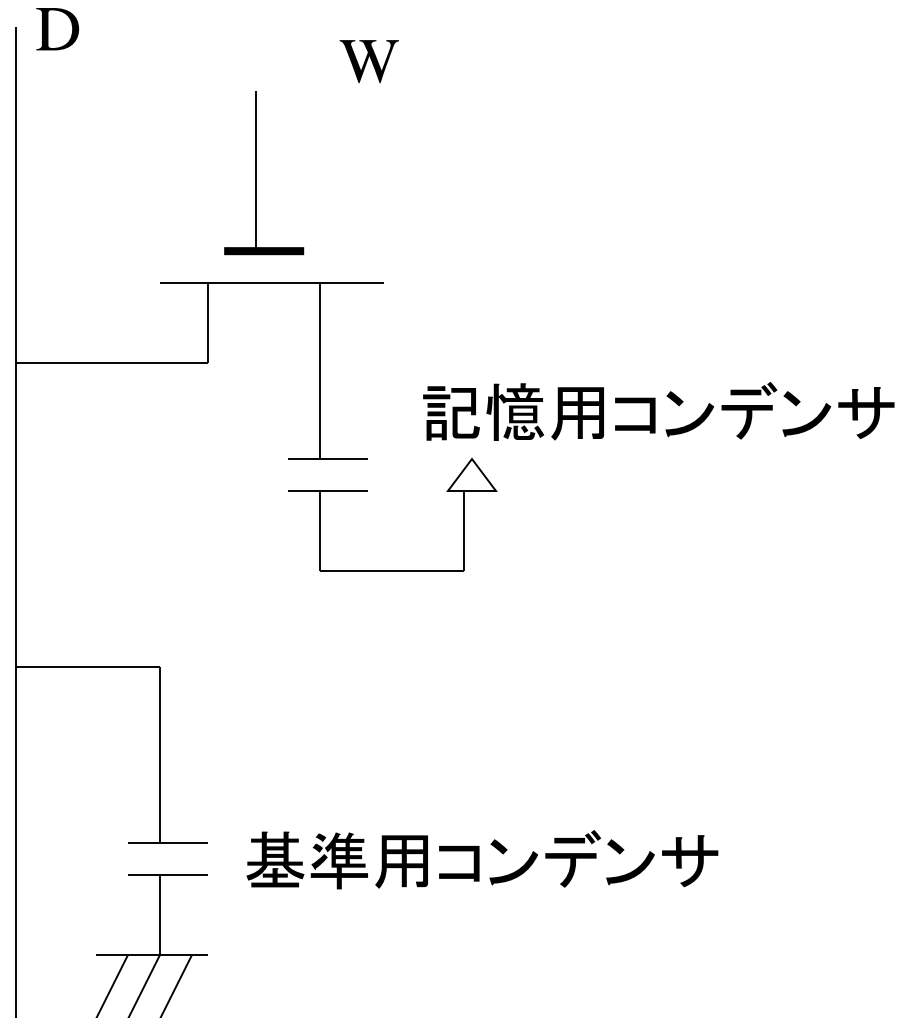
$\overline{\text{CAS}}$

データ

有効

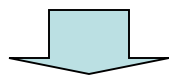


DRAMの記憶部分



DRAMアクセスの特徴

- 破壊読出しなので、書き戻しが必要
- 微小電位を検出するセンスアンプが必要
- 基準コンデンサを充電するためのプリチャージ時間が必要
- ほっておくと電荷が放電してしまうので、リフレッシュが必要



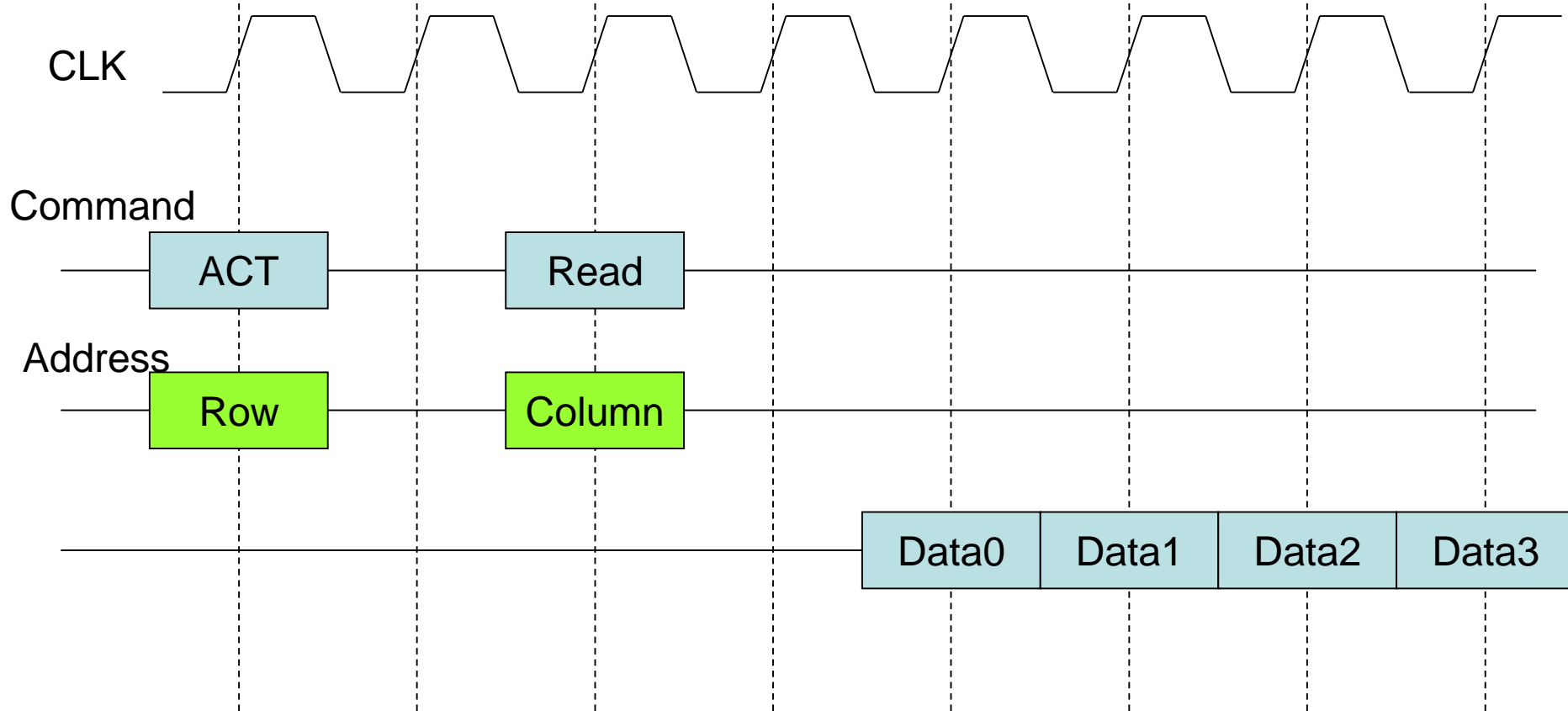
SRAMに比べて使い難い

SDR (Single Data Rate)

SDRAM: 同期式DRAM

- 100MHzー133MHzの高速クロックに同期した読み・書きを行う
- CS,RAS,CAS,WEなどの制御線の組み合わせでコマンドを構成
- コマンドにより、同期式に読み、書き、リフレッシュ等を制御
- バンクの切り替えにより連続読み・書きが高速に可能

SDR-SDRAMの 読み出しタイミング

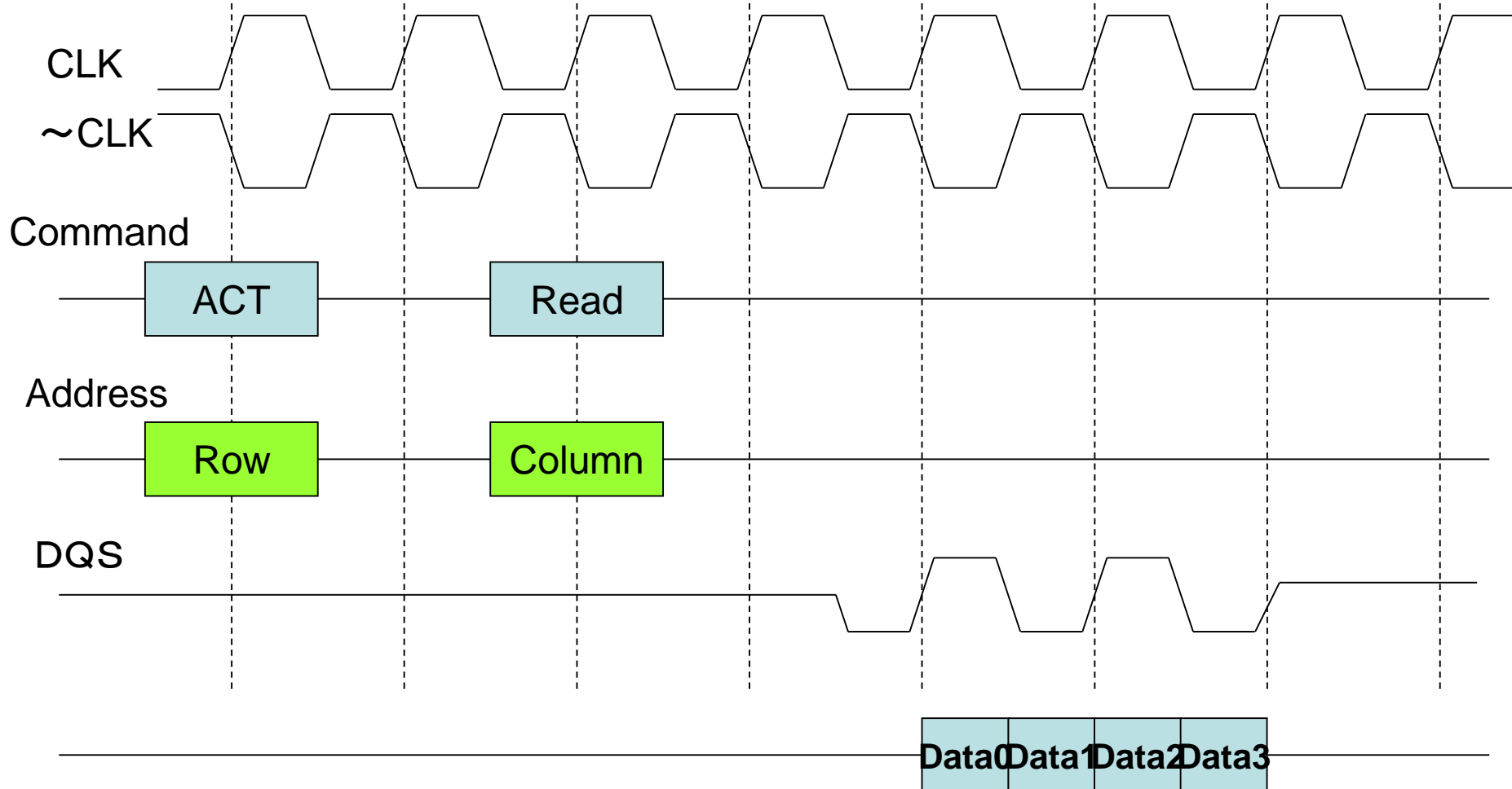


DDR (Double Data Rate)

SDRAM: 同期式DRAM

- SDR SDRAM同様の高速周波数(100MHzー133MHz)のクロックの両エッジで転送を行うことにより、倍のデータ転送レートを実現
- 差動クロックを利用
- データストロージブ信号によりタイミング調整
- より豊富なコマンド

DDR-SDRAMの 読み出しタイミング



DRAMのまとめ

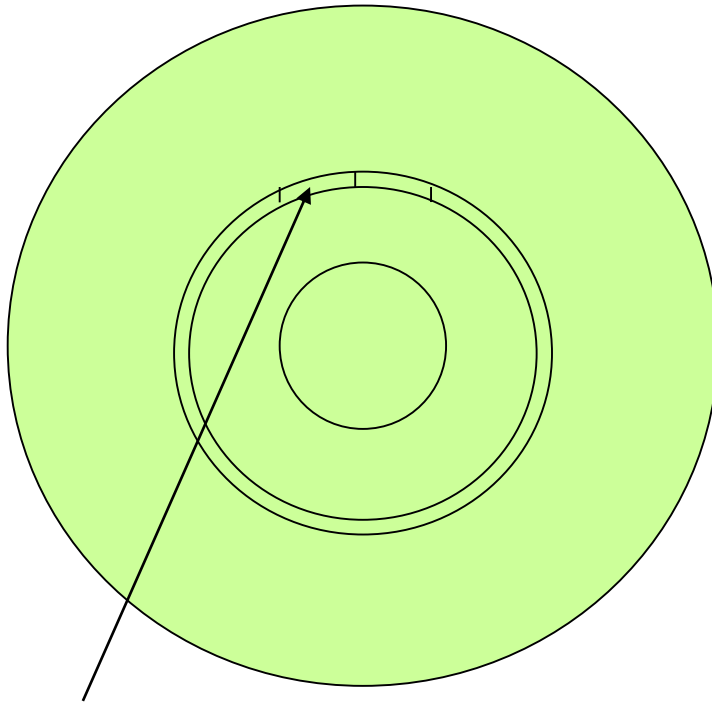
- SRAMの4倍程度集積度が大
- 使い難いが、連続アクセスは高速
- 転送はますますパケット化する傾向にある
 - SDR-SDRAM→ DDR-SDRAM→DDR2-SDRAM
 - DDR2: 800Mbps (400MHz両エッジ) 2Gbit /Chip
 - DDR3: 1600Mbps (800MHz両エッジ) 4Gbit /Chip
 - パッケージ: FBGA(Fine pitch Ball Grid Array)の利用
 - SO-DIMM(Small outline Dual dual in-line memory module)の形で供給される: 8GByte/DIMM
 - 現在PC用にはDDR3が標準となる
 - プリフェッチ機能→ 連続転送可能
 - 1.5V電源、電気的特性の改善
 - DDR-4が準備中
 - Hybrid Memory Cubeという3次元積層の新しい規格が登場
- 制御は複雑、高速なため取り扱いもたいへん
 - IP(Intellectual Property)の利用が進む

フラッシュメモリ

- EEPROM型の発展: 小型化のために選択ゲートを用いず、ブロック単位で消去を行う.
- NOR型、NAND型、DINOR型、AND型等様々な構成法がある.
 - オンチップ用: 高速消去可能NOR型 1Gbit程度まで
 - 単独読み出しが可能、消去が高速
 - ファイルストレージ用: 大容量のNAND型 1Gbit- 128Gbit/チップ
 - 連続読み出し、消去はミリ秒オーダー掛かる
 - SDメモ리카ード・SDHCメモ리카ードなど、8GB-32GBが使われる
 - 書き換え回数に制限がある
- みんなが最も身近に使うメモリ

ストレージシステム：ディスク装置

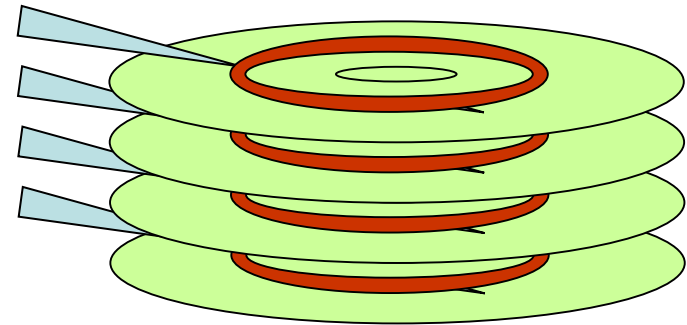
トラック：同心円状のアクセスの単位
1万-5万ある



セクタ：512B程度に分割したアクセスの単位
100-500 セクタ番号、誤り訂正符号付きのデータを含む

シリンダ：ヘッドの下にある
すべてのトラックのこと

ヘッド

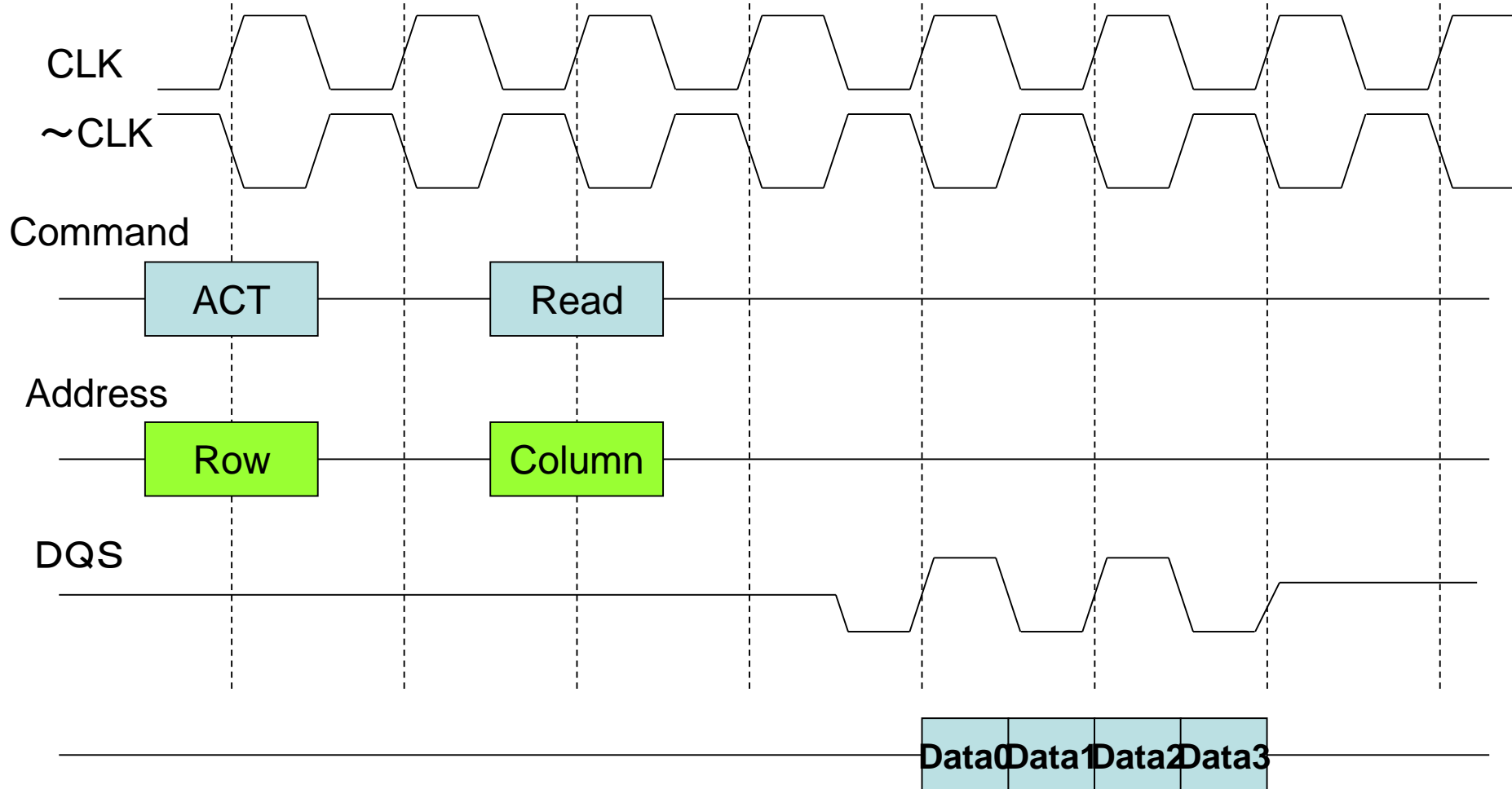


磁性体の塗布された円板に
データを格納
可動式のヘッドを使って読み書き
不揮発性

容量と動作速度

- 2.5インチー3.5インチ
- ヘッド数: 2-4
- 容量: 100GB-1TB
- 平均ディスクアクセス時間 =
平均シーク時間(ヘッドを動かす時間) +
平均回転待ち時間 + 転送時間 → 数msec
- インタフェース
 - ATA(Advanced Technology Attachment)
 - SCSI(Small Computer Systems Interface)
- ディスク内にマイクロプロセッサを装備し、アクセス時間を最適化
- ディスクキャッシュの利用

演習1

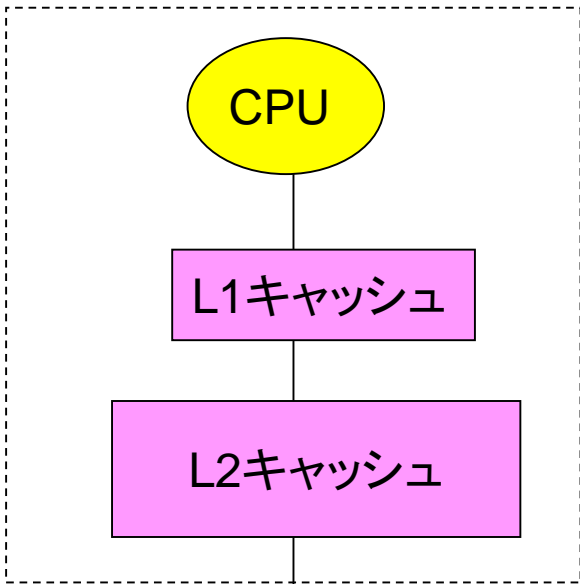


100MHzで動作する時、データの幅が32ビットだとすると、転送容量は何Mbyte/secか？

記憶の階層

高速小容量の
CPUの近くに置き
よく使うデータを入れておく

そこになければより遅い
大容量メモリに取りに行く

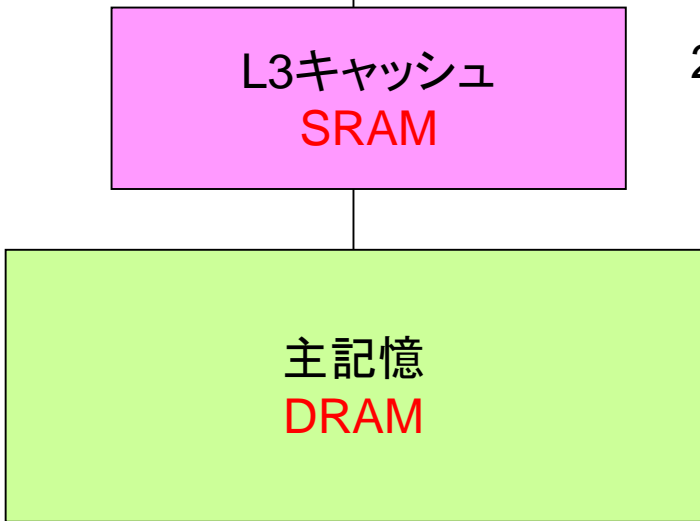


ソフトウェアから
は透過
(トランスペアレント)

チップ内メモリ=SRAM
~64KB 1-2clock

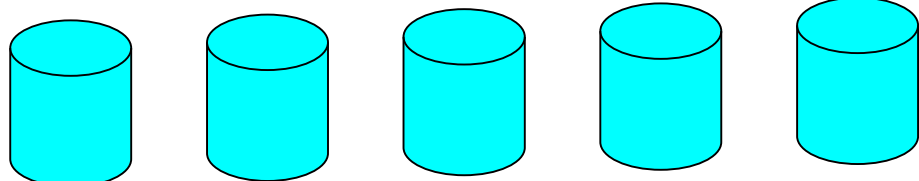
~256KB 3-10clock
SRAM

2M~4MB 10-20clock



4~16GB 50-100clock

OSが管理



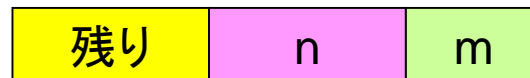
補助記憶 (2次記憶)
μ-msecオーダー
数百GB ディスク
フラッシュROM

キャッシュ

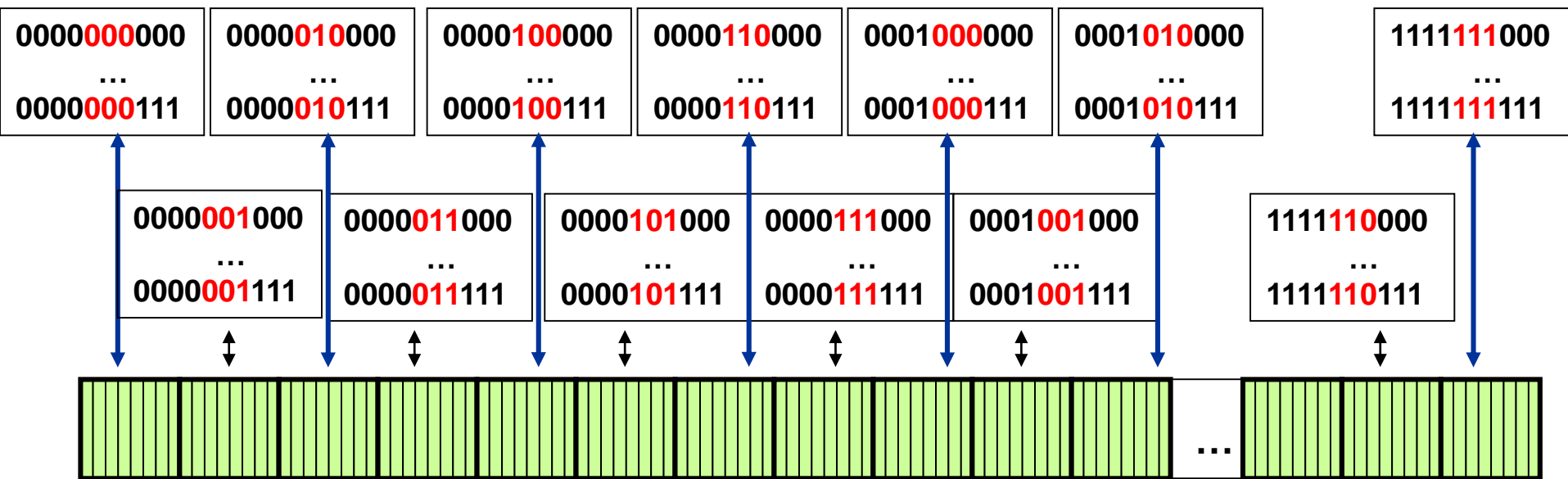
- 頻繁にアクセスされるデータを入れておく小規模高速なメモリ
 - CacheであってCashではないので注意
 - 元々はコンピュータの主記憶に対するものだが、IT装置の色々なところに使われるようになった
 - ディスクキャッシュ、ページキャッシュ..etc..
- 当たる(ヒット)、はずれる(ミスヒット)
 - ミスヒットしたら、下のメモリ階層から取ってきて入れ替える(リプレイス)
- マッピング(割り付け)
 - 主記憶とキャッシュのアドレスを高速に対応付ける
 - Direct map ⇔ Full associative cache
- 書き込みポリシー
 - ライトスルー、ライトバック
- リプレイス(追い出し)ポリシー
 - LRU (Least Recently Used)

アドレスマッピング(割り付け)

- ワード単位に割り付けるのは効率が悪い
 - 一定の連続アドレスのブロック(ライン)を管理単位とする
 - ブロックサイズは8byte-128byte程度
 - ここでは8word(16byte)を使う
 - やや小さい
- 順番に割り付けていって1周したら、元に戻る
 - キャッシュのブロック数(セット数)が2の n 乗、ブロックサイズが2の m 乗とすると、、、

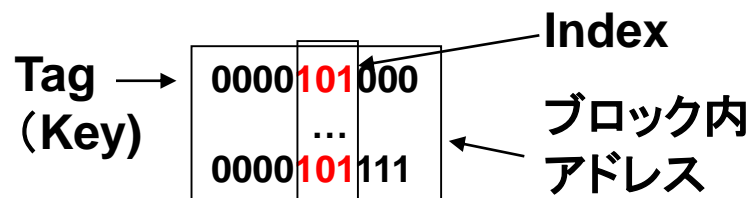
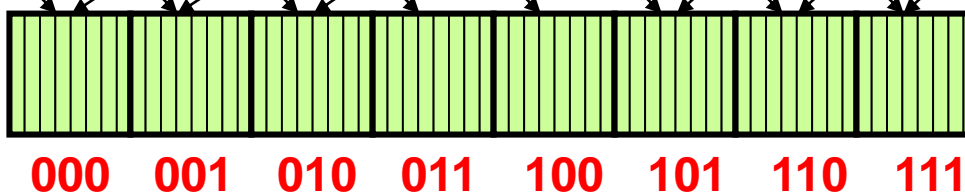


タグ (キー) インデックス ブロック内アドレス

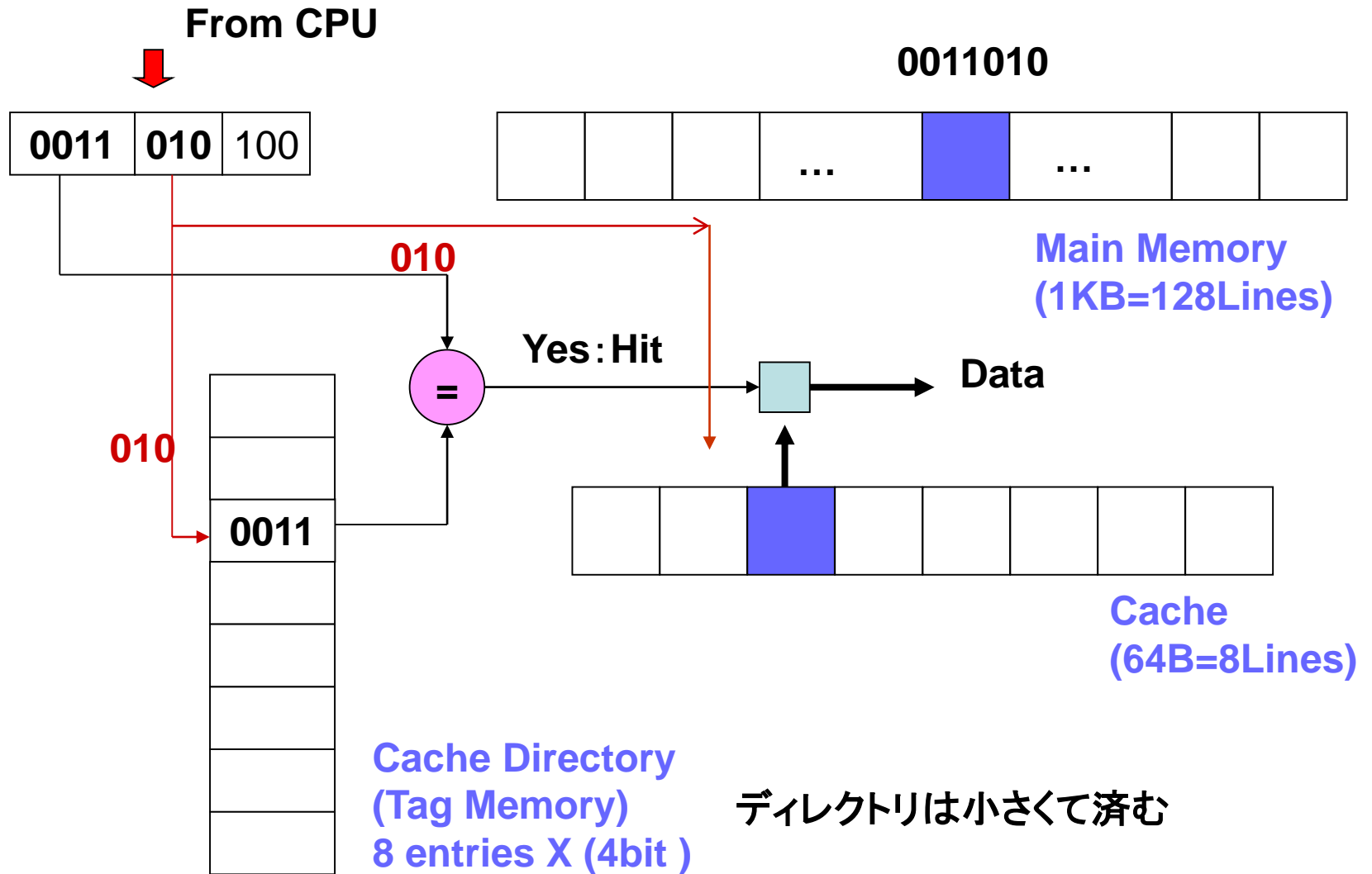


**Direct Map
のアドレス
割り付け**

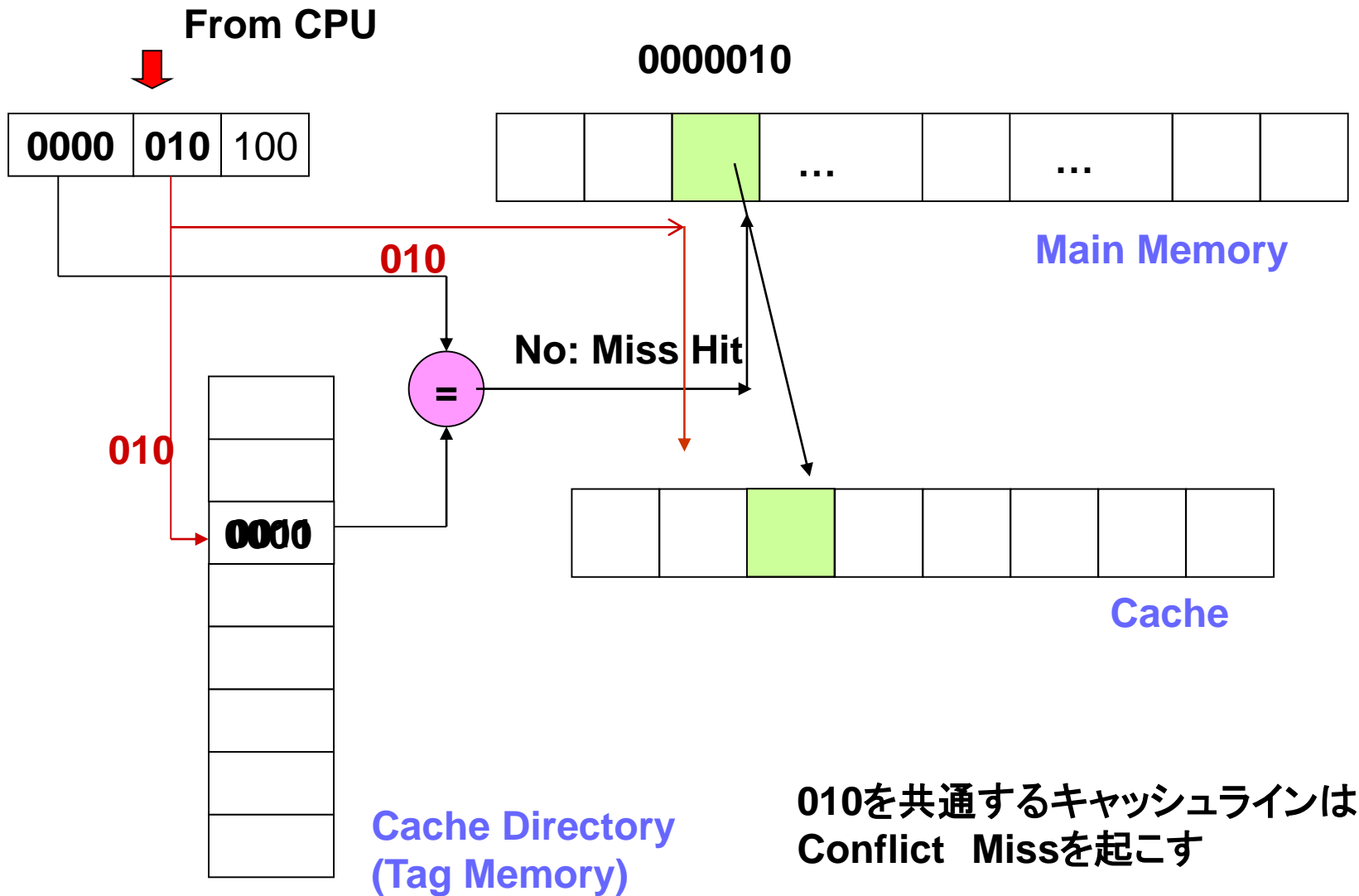
主記憶: 1024ワード
ブロックサイズ: 8ワード
キャッシュ: 64ワード
= 8ブロック

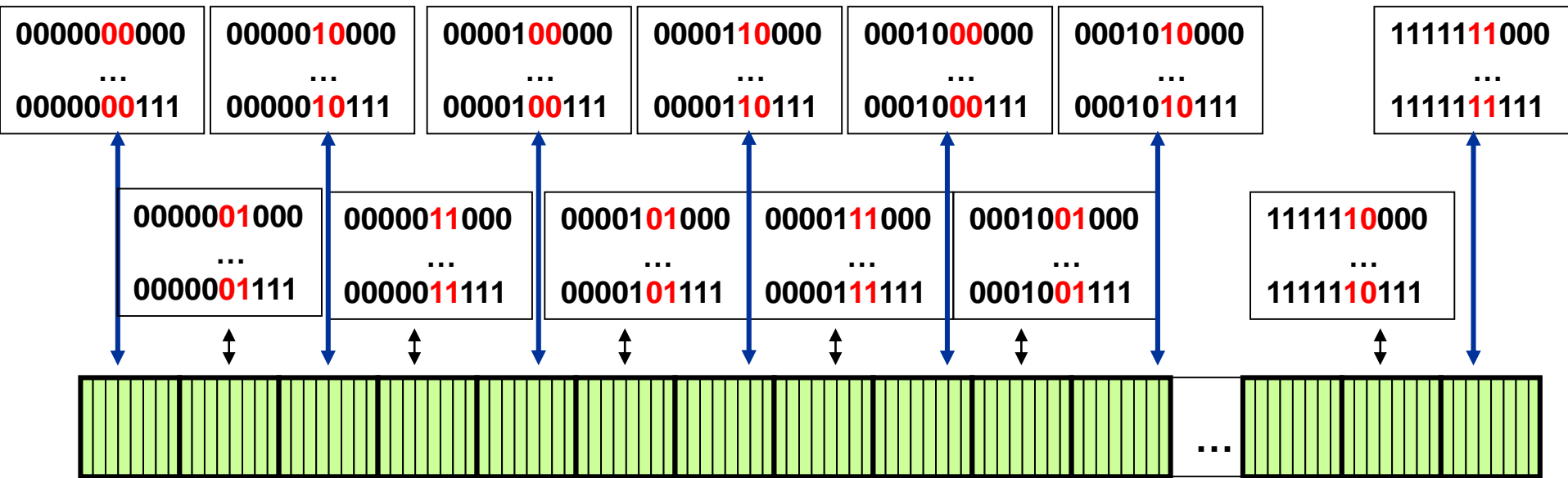


ダイレクトマッピング方式

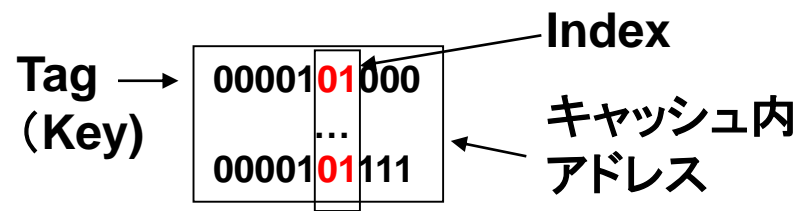
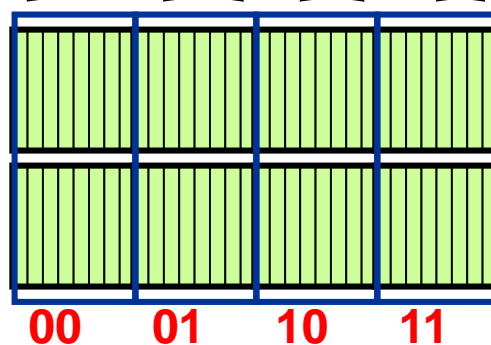


ダイレクトマッピング方式 (Conflict Miss)

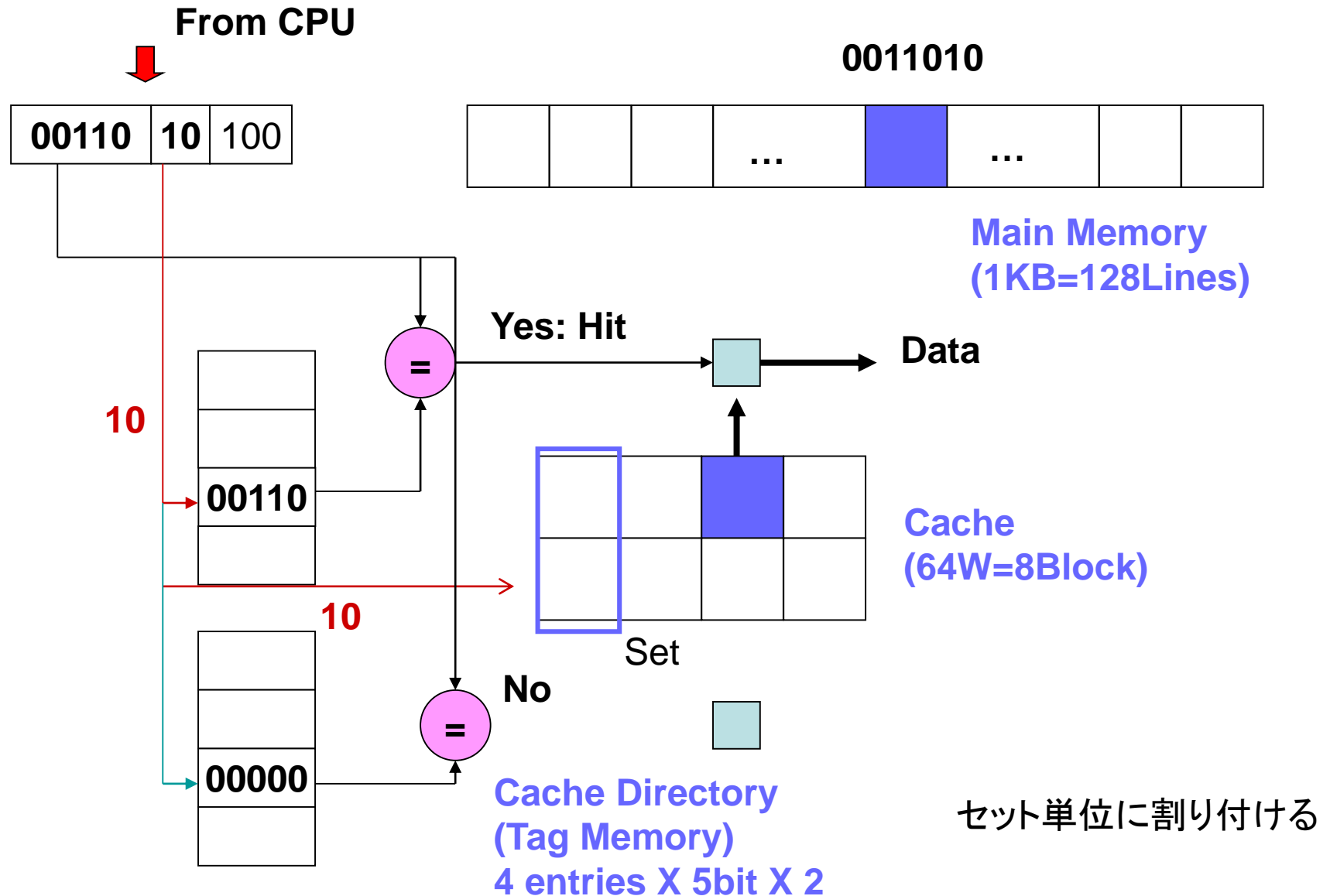




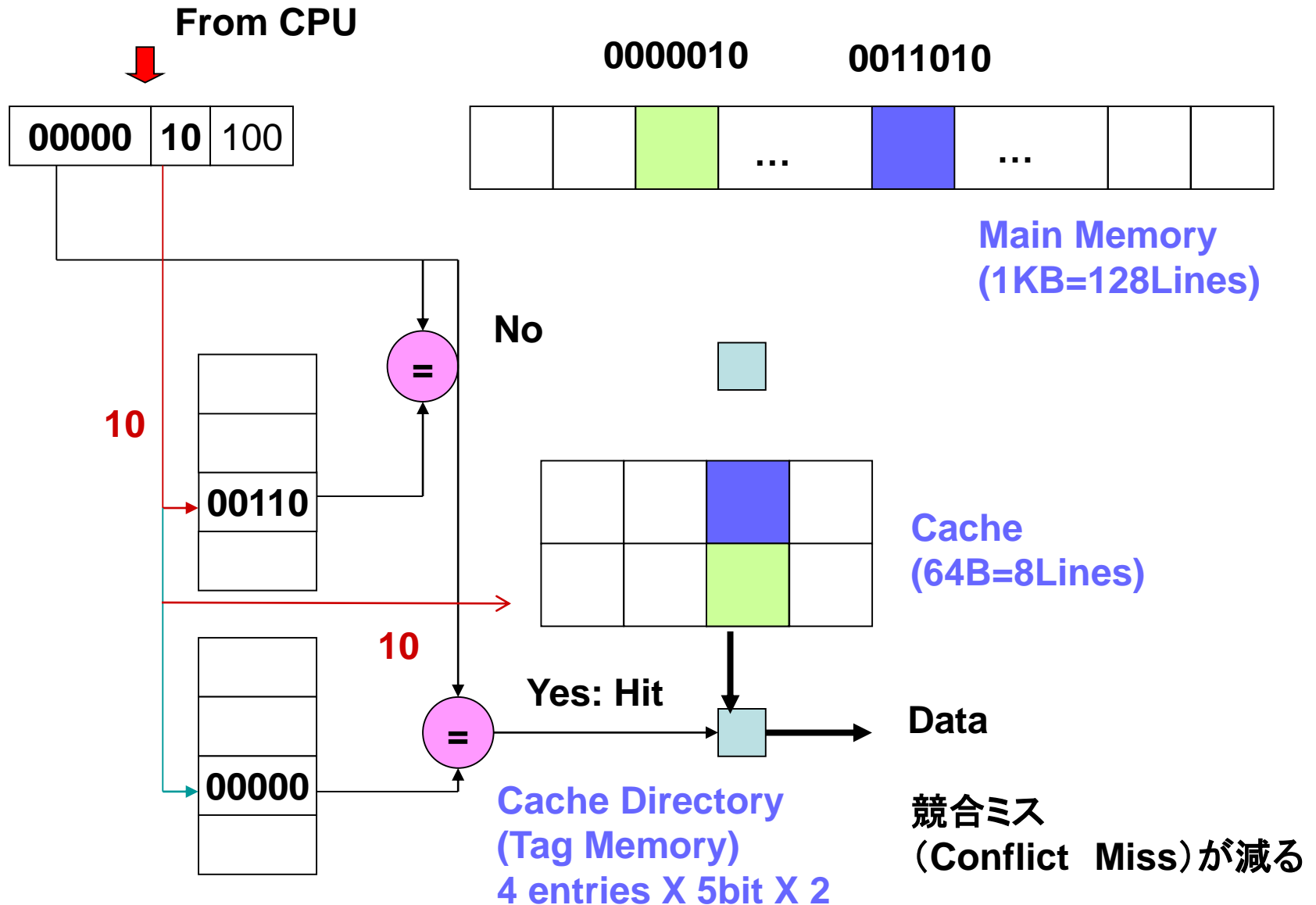
2-way set associative
のアドレス
割り付け



2-wayセットアソシアティブマップ



2-wayセットアソシアティブマップ



4-wayセットアソシアティブマップ

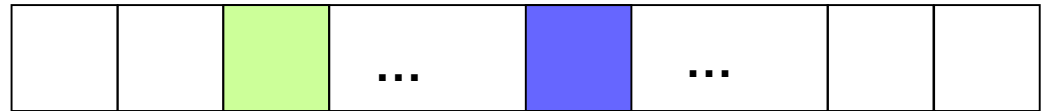
From CPU



001101 0 100

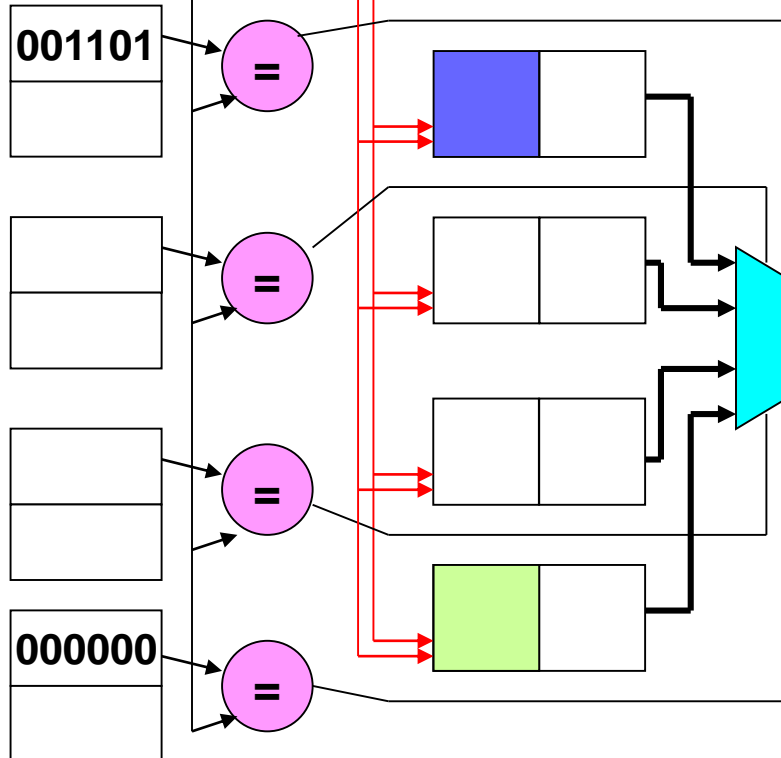
0000010

0011010



Main Memory
(1KB=128Lines)

0



Cache Directory
(Tag Memory)
2 entries X 6bit X 4

Cache
(64W=8Blocks)

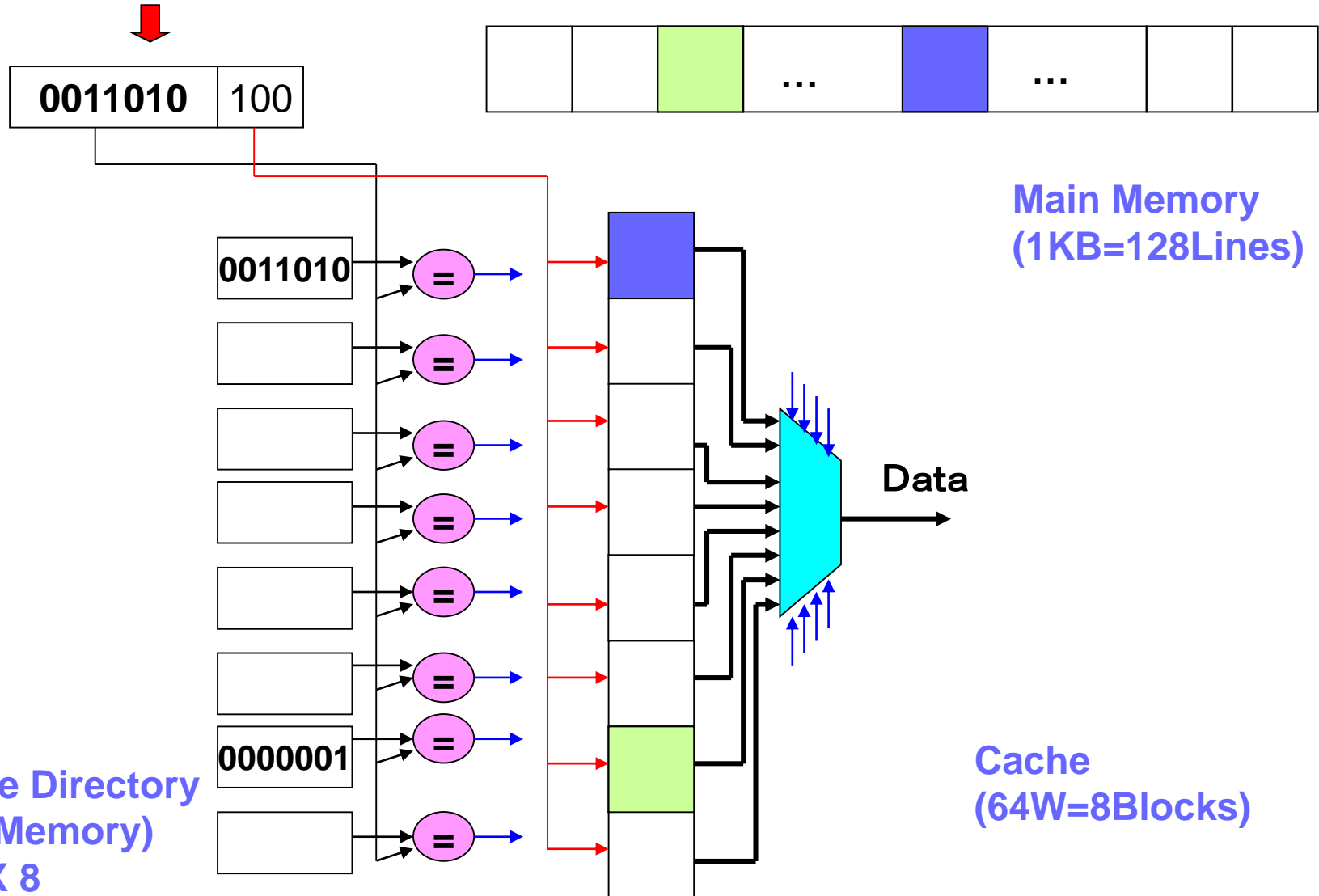
Data

8-wayセットアソシアティブ→ここではフルマップ

From CPU

0000010

0011010



タグメモリの設計法

- キャッシュ内に何ブロック入るかを計算する。
 - 2^n 乗である時
 - インデックスは n bit となる
- メモリ内に何ブロック入るかを計算する。
 - 2^h 乗である時
 - タグは $h - n = m$ bit となる
- ダイレクトマップでは幅 m , 深さ 2^n のタグメモリが必要
- 2-way set associative は、インデックスが 1 bit 減り深さが半分となり、タグが 1 bit を増える。しかしこれがダブルで必要
 - way 数が倍になる度にインデックスが 1 bit 減り、深さが半分になり、タグが 1 bit 増え、タグ自体が倍になる。

Way数のトレードオフ

大きくすると、、、

－ヒット率が改善

- Direct Map→2way set associative

32人で1つの椅子を争う VS. 64人で2つの椅子を争う

偶然同じ時間に椅子を狙うライバルが居る場合は効果的

サイズを倍にするのと同じ程度の効果が見込まれる

- それ以上はどんどん効果が減る
- 4以上はあまり効果が上がらない

－遅延時間が大きくなる(マルチプレクサの遅延)

－8くらいまでが多い

演習2

- 64kワードの主記憶に対して8kワードのキャッシュを設ける
- ブロックサイズは16ワードとする
- ダイレクトマップ、2way set associative、4way set associativeキャッシュのタグメモリ構成をそれぞれ示せ