

Kiến trúc máy tính

Bộ nhớ và các thiết bị lưu trữ

NGUYỄN Ngọc Hoá

Bộ môn Hệ thống thông tin, Khoa CNTT
Trường Đại học Công nghệ,
Đại học Quốc gia Hà Nội

Nội dung

1. Khái niệm chung
2. Bộ nhớ chính
3. Bộ nhớ cache
4. Bộ nhớ ngoài (các thiết bị lưu trữ)
5. Tổng kết và bài tập

Tham khảo chương 4,5,6 của [1]

1. Khái niệm

- Bộ nhớ: thiết bị có thể *bảo quản* và *khôi phục* một thông tin
- Từ nhớ: tập bits có thể được đọc hay ghi đồng thời
- Các kiểu vật liệu nhớ:
 - Bán dẫn – semiconductor (register, cache, bộ nhớ chính, ...)
 - Từ - magnetic (đĩa mềm, đĩa cứng, ...)
 - Optic (CD-ROM, DVD-ROM)
 - ...

Đặc điểm

1. Vị trí
2. Dung lượng
3. Đơn vị truyền
4. Kiểu truy cập
5. Hiệu năng
6. Kiểu vật liệu
7. Đặc trưng vật liệu
8. Tổ chức

Đặc điểm...

- Vị trí
 - CPU
 - Internal
 - External
- Dung lượng
 - Phụ thuộc vào kích thước từ nhớ, và
 - Số lượng từ nhớ
- Đơn vị truyền
 - Bên trong: phụ thuộc vào độ rộng bus dữ liệu
 - Bên ngoài: block(>từ nhớ)
 - Đơn vị có thể đánh địa chỉ được

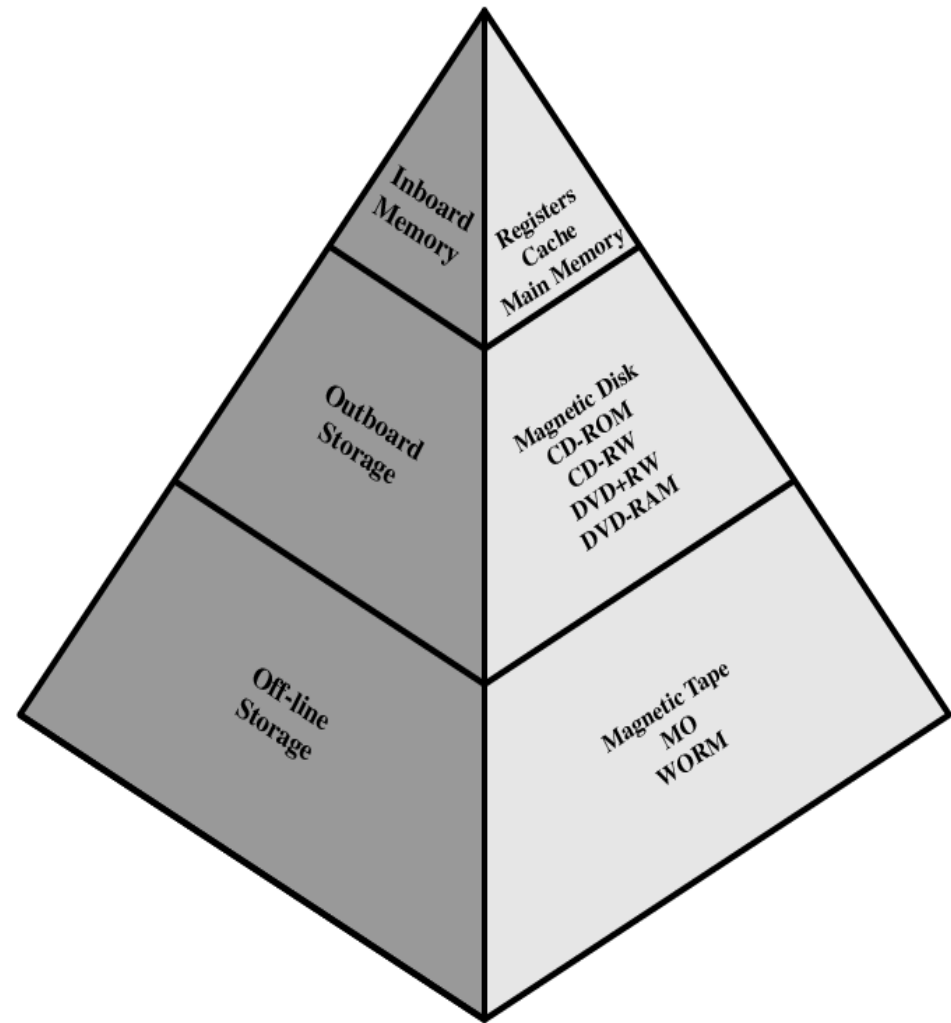
Đặc điểm...

■ Kiểu truy cập

- Tuần tự: VD băng từ
- Trực tiếp:
 - Mỗi 1 block có 1 địa chỉ duy nhất
 - Truy cập = cách nhảy đến vùng lân cận và tìm tuần tự
 - Thời gian truy cập vào vị trí hiện tại hiện tại và trước đó
 - VD: HardDisk, Floppy Disk,...
- Ngẫu nhiên:
 - Mỗi địa chỉ xác định chính xác một vị trí
 - Thời gian truy cập không phụ thuộc vào vị trí cũng như lần truy cập trước
 - VD: RAM, ...
- Kết hợp:
 - Một từ được định vị thông qua việc sử dụng một phần nội dung của từ đó
 - Thời gian truy cập không phụ thuộc vào vị trí cũng như lần truy cập trước
 - VD: cache, ...

Đặc điểm...

- Chiến thuật phân cấp bộ nhớ: How much? How fast? How expensive?
 - Registers
 - L1 Cache
 - L2 Cache
 - Main memory
 - Disk cache
 - Disk
 - Optical
 - Tape



Đặc điểm...

■ Hiệu năng:

- Thời gian truy cập: khoảng thời gian từ khi gửi địa chỉ cho đến khi thu được dữ liệu trọn vẹn
- Thời gian chu trình nhớ - Memory Cycle Time:
 - Thời gian bộ nhớ đòi hỏi để “hồi phục” trước lần truy cập kế tiếp
 - = access + recovery
- Tốc độ chuyển dữ liệu

■ Kiểu vật liệu:

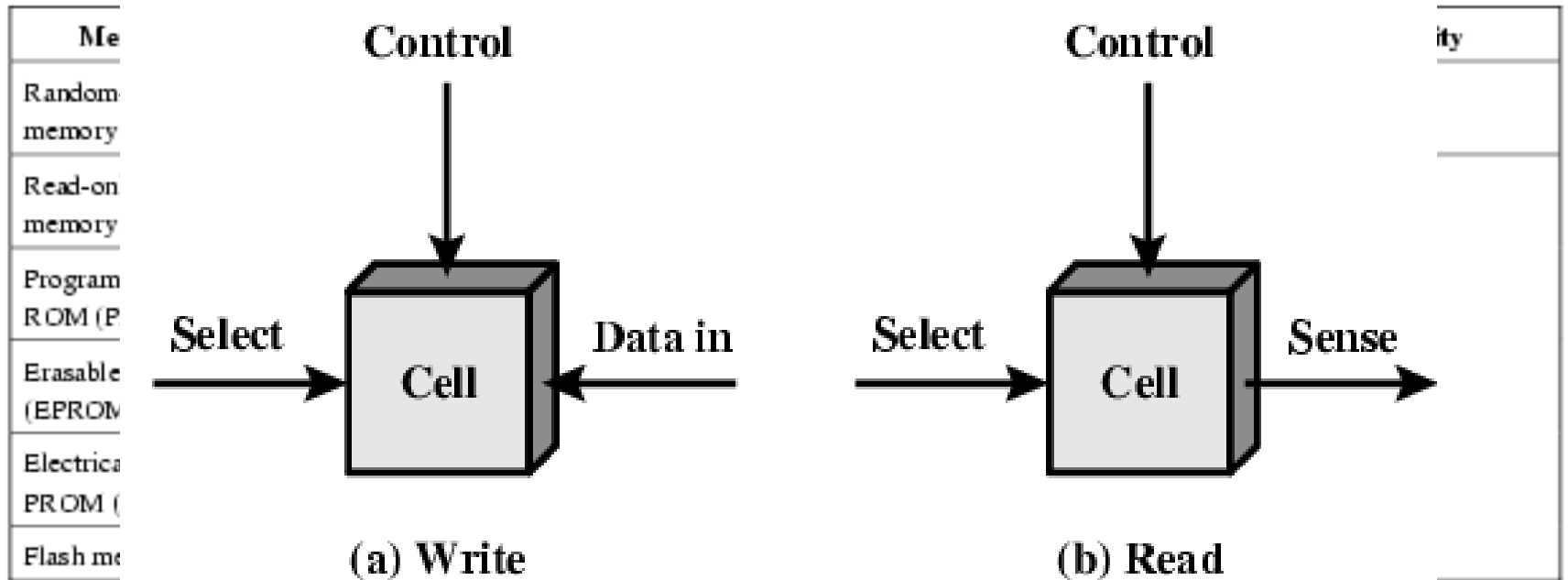
- Semiconductor :RAM
- Magnetic: Disk & Tape
- Optical: CD & DVD
- Others: Bubble, Hologram

Đặc điểm...

- Đặc trưng vật liệu:
 - Phân rã - Decay
 - Dễ thay đổi - Volatility
 - Có thể xoá được - Erasable
 - Năng lượng tiêu thụ
- Tổ chức:
 - Cách thức sắp xếp các bits trong một từ
 - Thường không rõ ràng
 - VD: interleaved

2. Bộ nhớ chính

- Bộ nhớ bên trong máy tính
 - Semi-conductor
 - Truy cập ngẫu nhiên
- Kiểu:
 - RAM- Random Access Memory: lưu giữ những dữ liệu tạm thời
 - ROM – Read Only Memory: lưu giữ thông tin cố định



Read Only Memory (ROM)

- Lưu giữ thông tin cố định - permanent storage, nonvolatile
 - Microprogramming
 - Library subroutines
 - Systems programs (BIOS)
 - Function tables
- Kiểu:
 - Written during manufacture
 - Very expensive for small runs
 - Programmable (once)
 - PROM
 - Needs special equipment to program
 - Read “mostly”
 - Erasable Programmable (EPROM)
 - Erased by UV
 - Electrically Erasable (EEPROM)
 - Takes much longer to write than read
 - Flash memory
 - Erase whole memory electrically

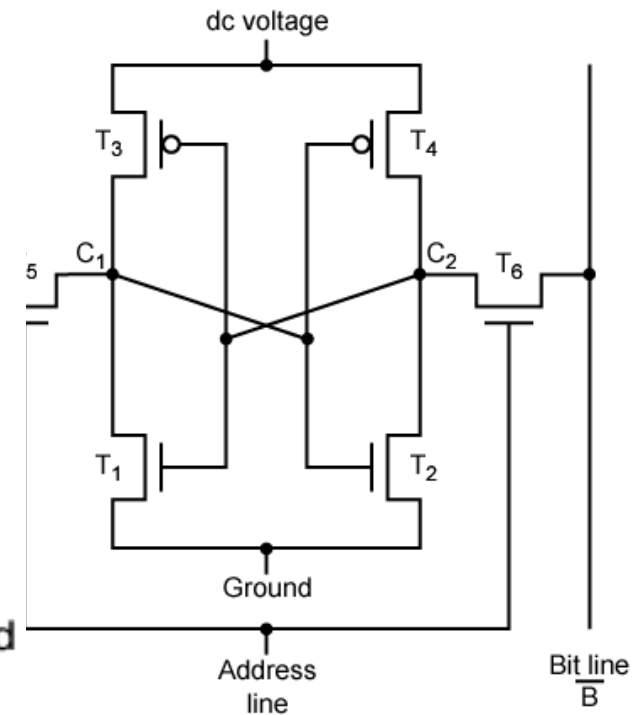
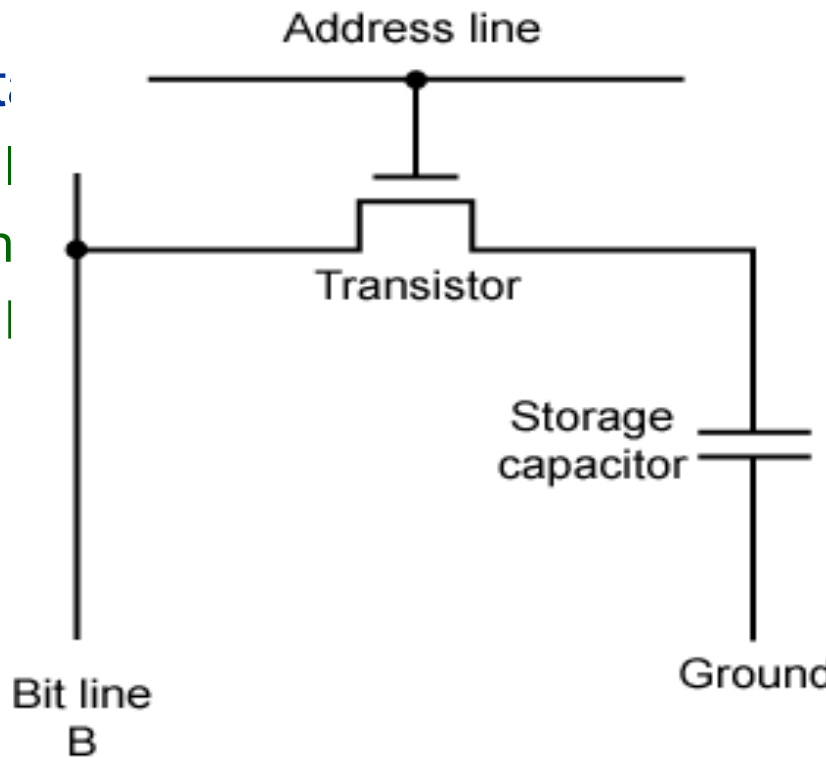
RAM

■ DRAM – Dynamic RAM

- Bits được lưu trữ trong các tụ điện
- Đơn giản, kích thước bé, giá thành rẻ
- Chậm, cần 1 chu trình làm tươi ngay cả khi đã được cung cấp nguồn

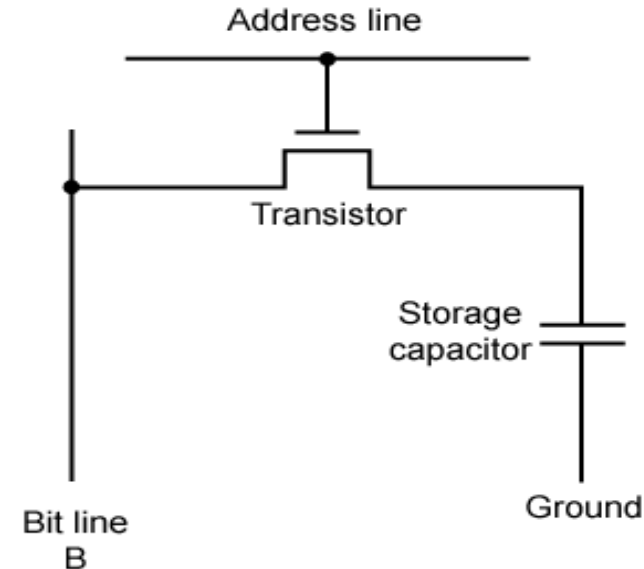
■ SRAM – Static RAM

- Bits được lưu trữ trong các transistor
- Không cần làm tươi
- Phức tạp, kích thước lớn

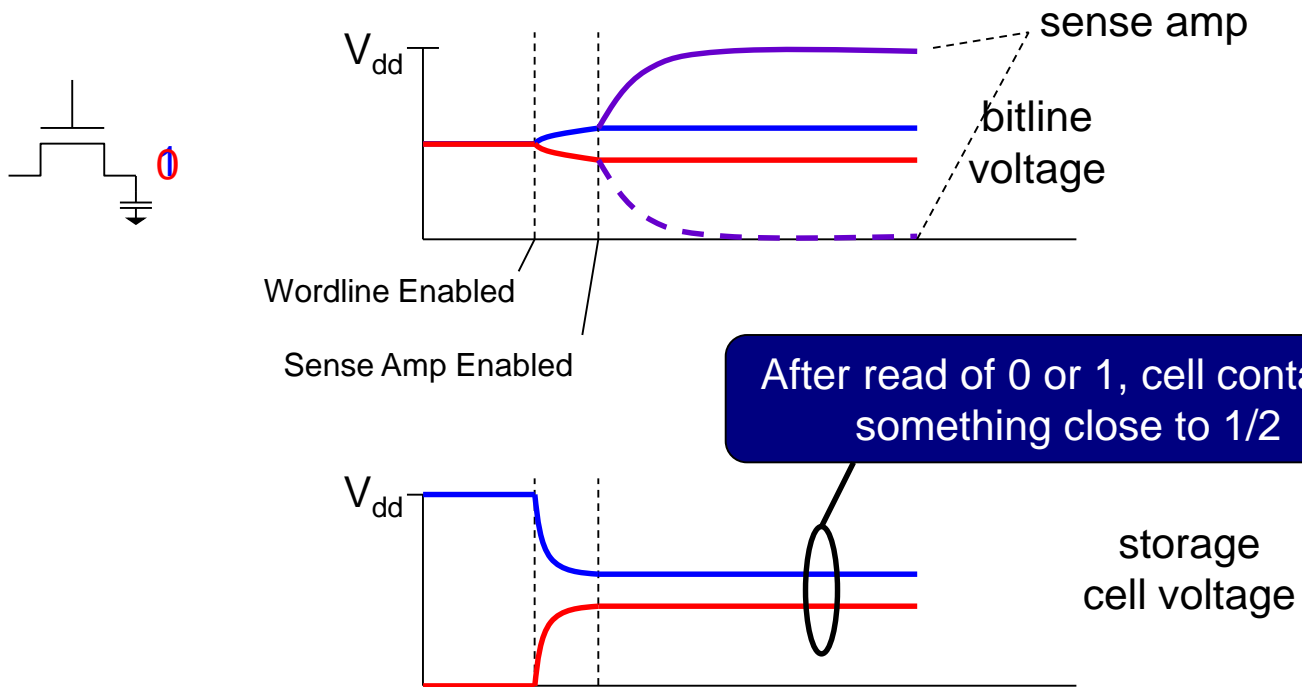


Dynamic RAM

- Đường địa chỉ được kích hoạt khi đọc/ghi bit
 - Transistor switch closed (current flows)
- Ghi
 - Voltage to bit line
 - High for 1 low for 0
 - Then signal address line
 - Transfers charge to capacitor
- Đọc
 - Address line selected
 - transistor turns on
 - Charge from capacitor fed via bit line to sense amplifier
 - Compares with reference value to determine 0 or 1
 - Capacitor charge must be restored

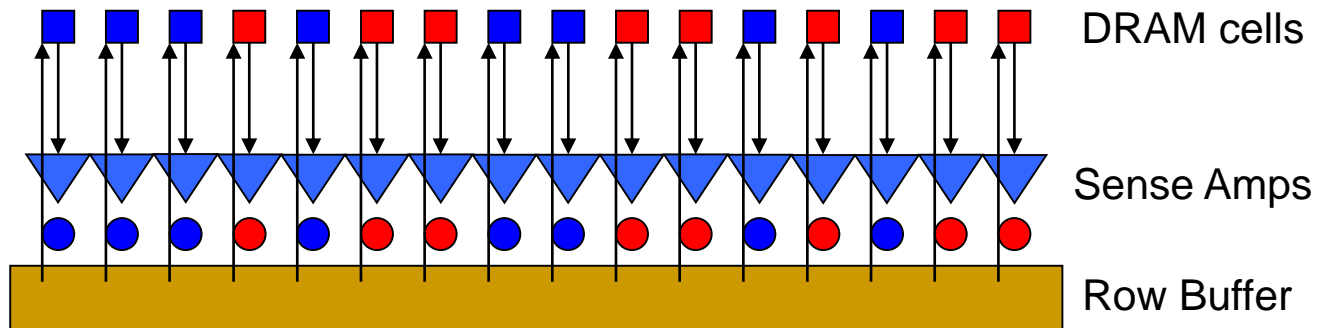


Destructive Read

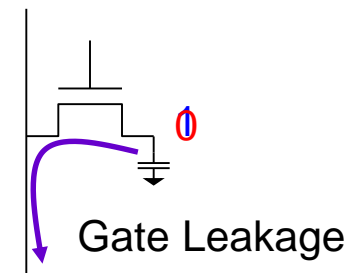


Cơ chế làm tươi - refresh

- Sau khi đọc, nội dung của DRAM cell đã bị thay đổi
- Lưu các giá trị cells trong bộ đệm hàng *row buffer*
- Ghi lại các giá trị đó cho các cells trong lần đọc kế tiếp



- Thực tế, DRAM cell sẽ mất nội dung ngay cả khi không có tác vụ đọc
 - ➔ lý do được gọi là “dynamic”
- Vì thế các cells trong DRAM cần được định kỳ đọc và ghi lại nội dung



Static RAM

- Transistor arrangement gives stable logic state

- State 1

- C_1 high, C_2 low
- T_1 T_4 off, T_2 T_3 on

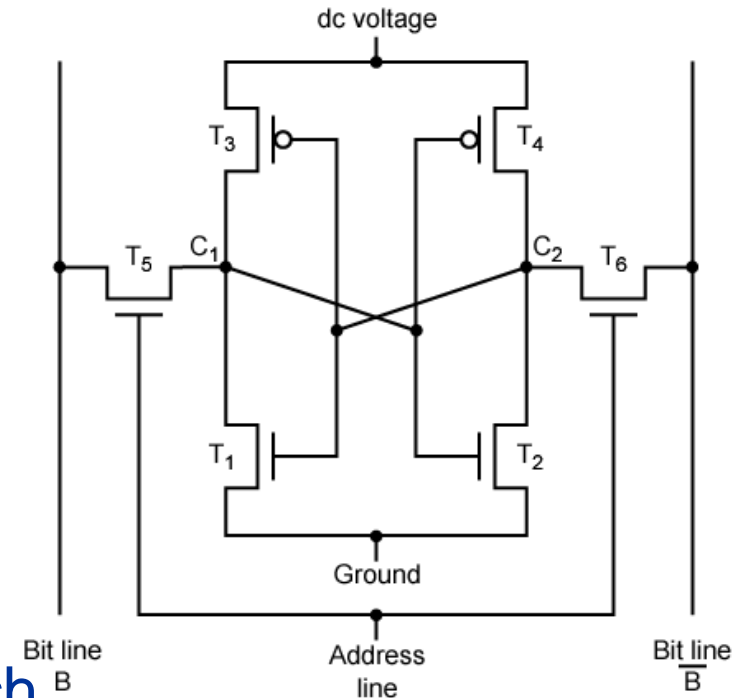
- State 0

- C_2 high, C_1 low
- T_2 T_3 off, T_1 T_4 on

- Address line transistors T_5 T_6 is switch

- Write – apply value to B & compliment to B

- Read – value is on line B



SRAM vs. DRAM

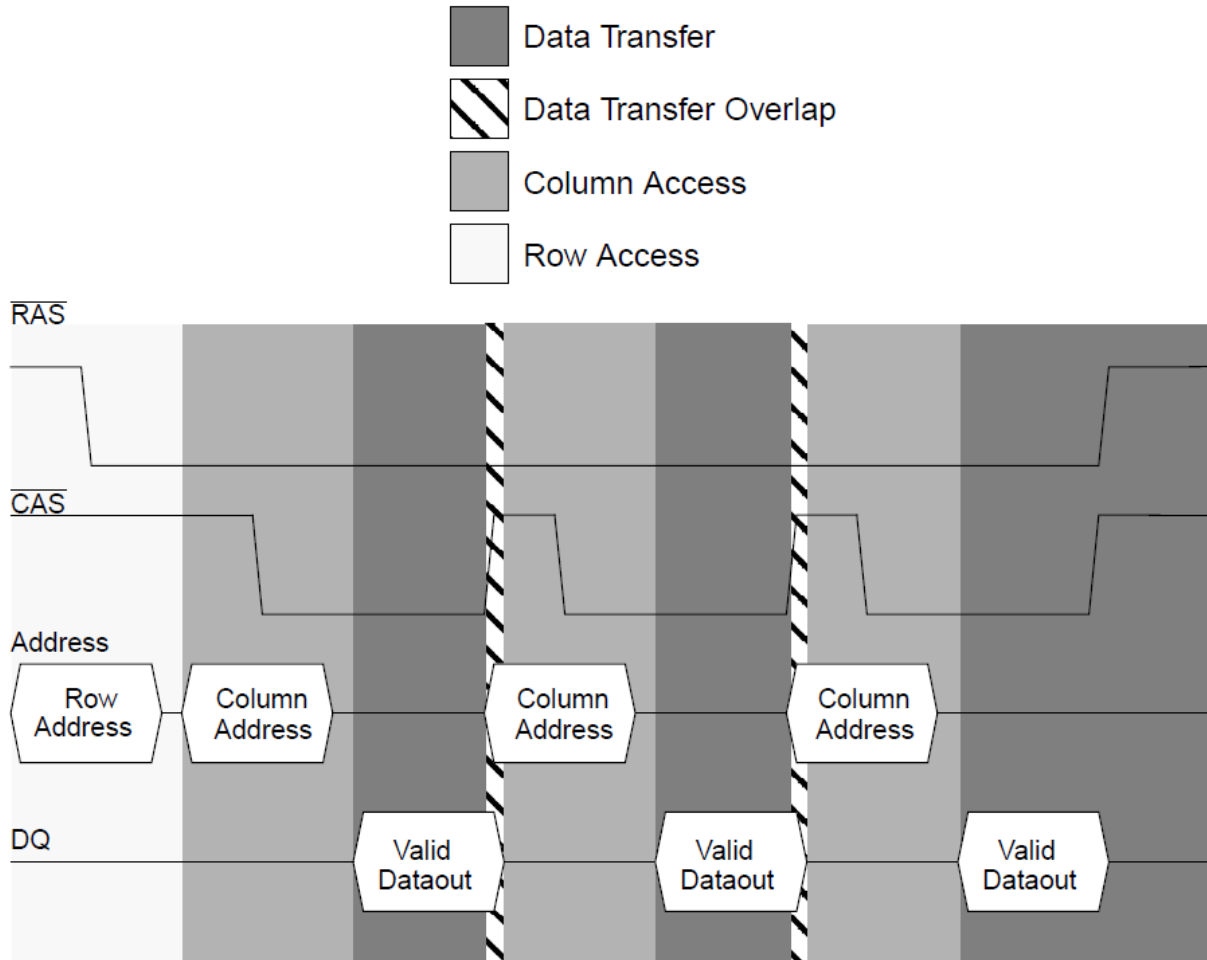
- Cả hai đều có tính chất volatile
 - Cần cung cấp năng lượng để bảo quản dữ liệu
- Dynamic cell
 - Đơn giản, kích thước nhỏ gọn
 - Mật độ cell cao
 - Chi phí thấp
 - Cần chu kỳ làm tươi
 - Cho phép kết hợp thành các đơn vi nhớ lớn
- Static
 - Nhanh hơn, công kênh hơn
 - Cho phép xây dựng các bộ nhớ Cache

Synchronous DRAM (SDRAM)

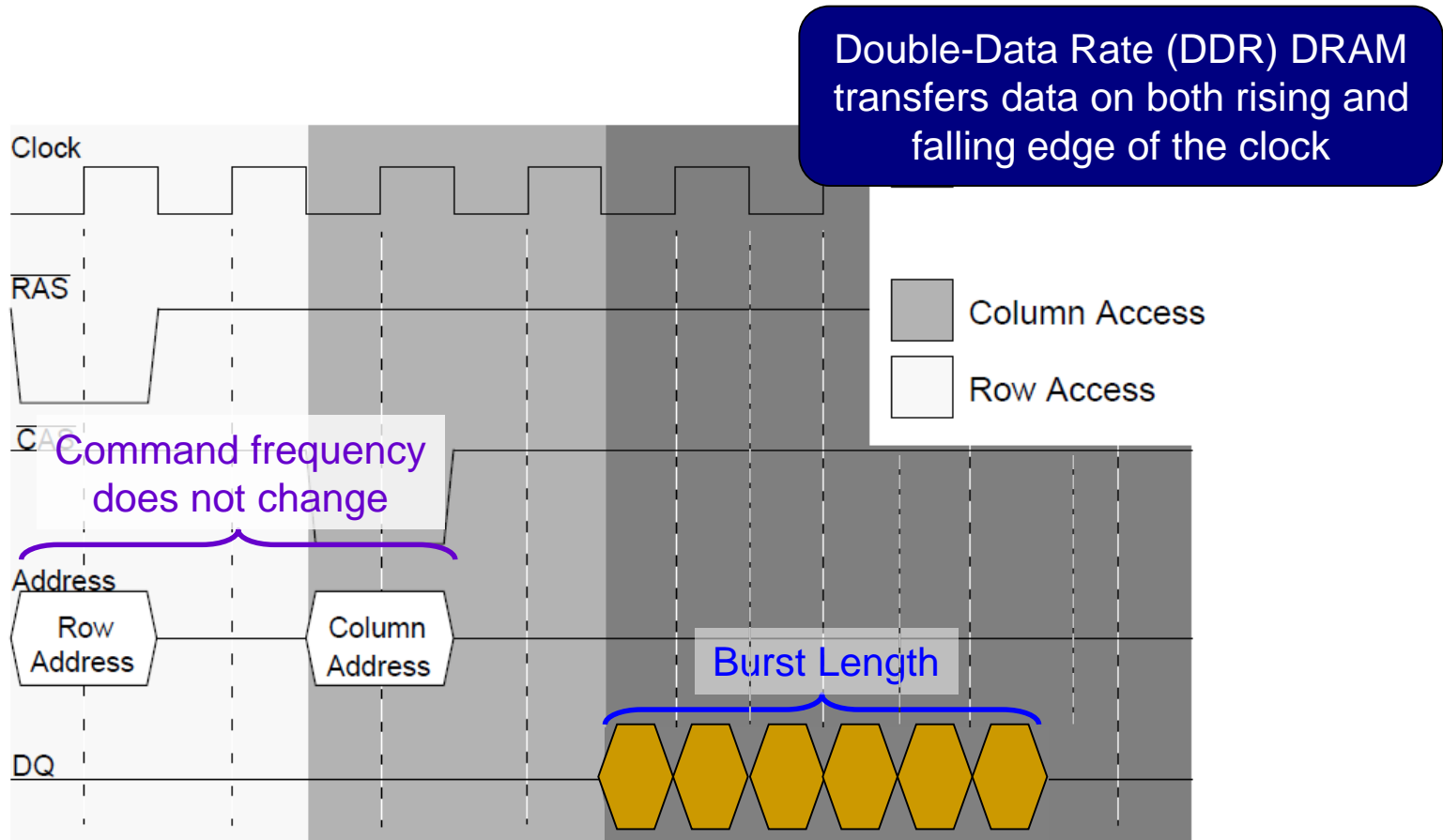
- Truy cập được đồng bộ hoá với một đồng hồ bên ngoài
 - Địa chỉ được truyền đến RAM
 - RAM tìm dữ liệu (CPU đợi như DRAM thông thường)
 - Khi SDRAM chuyển dữ liệu theo thời gian đồng bộ với system clock, CPU biết được khi nào dữ liệu sẵn sàng
- ⇒ **CPU không cần phải chờ và có thể làm việc khác**
 - Burst mode: cho phép SDRAM thiết lập dòng dữ liệu theo từng block
 - Chỉ chuyển dữ liệu 1 lần trong 1 chu kỳ đồng hồ
- **DDR-SDRAM - Double-data-rate 1 SDRAM**
 - Gửi dữ liệu 2 lần trong một chu kỳ đồng hồ (leading & trailing edge)
- **DDR2-SDRAM - Double-data-rate 2 SDRAM**
- **DDR3-SDRAM - Double-data-rate 3 SDRAM**
- **Cache DRAM: (misubishi)**
 - Tích hợp SRAM cache (16k) vào trong DRAM chip

DRAM Read Timing

Việc truy cập là không đồng bộ: được kiểm soát bởi các tín hiệu RAS & CAS, các tín hiệu này có thể được sinh ra ngẫu nhiên



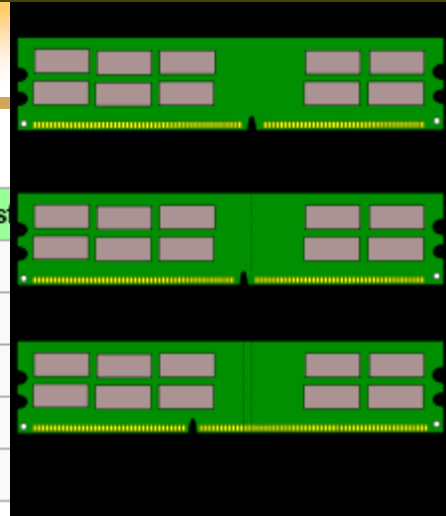
SDRAM Read Timing



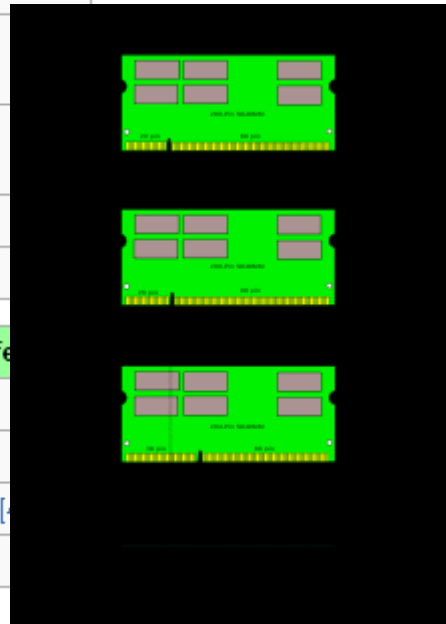
Timing figures taken from "A Performance Comparison of Contemporary DRAM Architectures" by Cuppu, Jacob, Davis and Mudge

DDR SDRAM

Standard name	Memory clock	Cycle time	I/O Bus clock	Data transfers per second	Module name	Peak transfer rate
DDR-200	100 MHz	10 ns ^[1]	100 MHz	200 Million	PC-1600	1600 MB/s
DDR-266	133 MHz	7.5 ns	133 MHz	266 Million	PC-2100	2100 MB/s
DDR-300	150 MHz	6.67 ns	150 MHz	300 Million	PC-2400	2400 MB/s
DDR-333	166 MHz	6 ns	166 MHz	333 Million	PC-2700	2700 MB/s
DDR-400	200 MHz	5 ns	200 MHz	400 Million	PC-3200	3200 MB/s



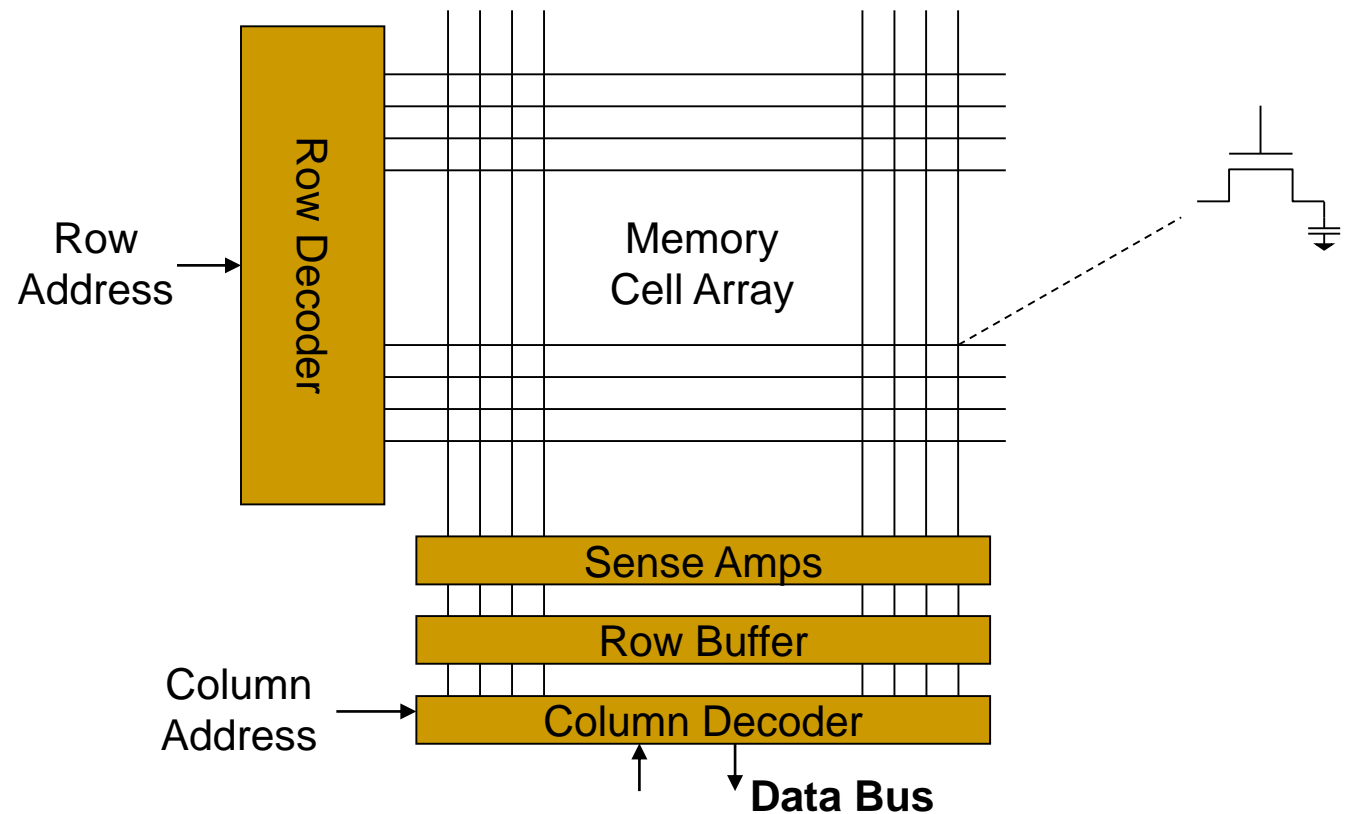
Standard name	Memory clock	Cycle time	I/O Bus clock	Data transfers per second	Module name	Peak transfer rate
DDR2-400	100 MHz	10 ns	200 MHz	400 Million	PC2-3200	3200 MB/s
DDR2-533	133 MHz	7.5 ns	266 MHz	533 Million	PC2-4200 PC2-4300 ¹	4266 MB/s
DDR2-667	166 MHz	6 ns	333 MHz	667 Million	PC2-5300 PC2-5400 ¹	5333 MB/s
DDR2-800	200 MHz	5 ns	400 MHz	800 Million	PC2-6400	6400 MB/s
DDR2-1066	266 MHz	3.75 ns	533 MHz	1066 Million	PC2-8500	8533 MB/s



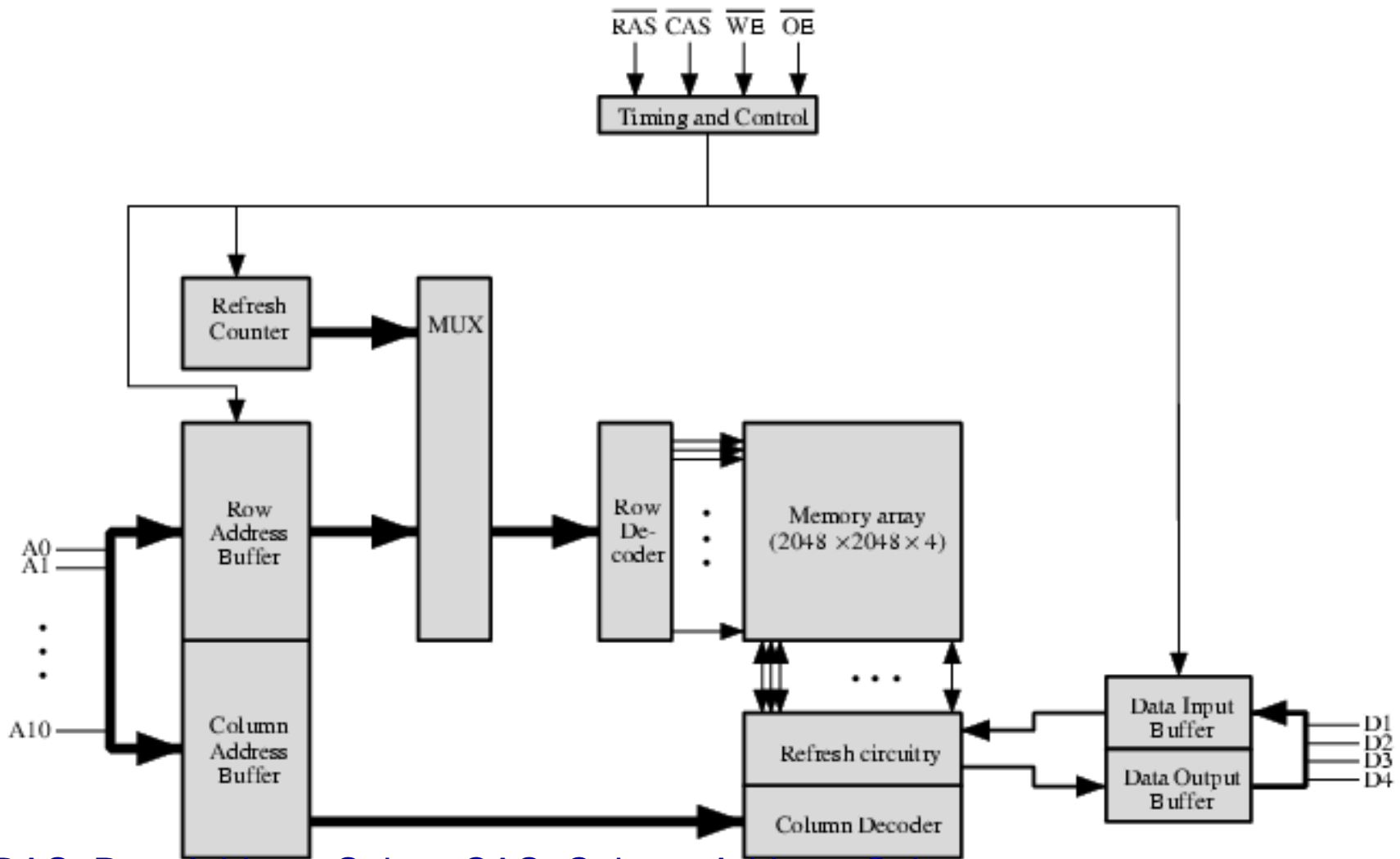
Standard name	Memory clock	Cycle time	I/O Bus clock	Data transfers per second	Module name	Peak transfer rate
DDR3-800	100 MHz	10 ns	400 MHz	800 Million	PC3-6400	6400 MB/s
DDR3-1066	133 MHz	7.5 ns	533 MHz	1066 Million	PC3-8500	8533 MB/s
DDR3-1333	166 MHz	6 ns	667 MHz	1333 Million	PC3-10600	10667 MB/s
DDR3-1600	200 MHz	5 ns	800 MHz	1600 Million	PC3-12800	12800 MB/s

Tổ chức bộ nhớ

- Mạch nhớ W từ B bits được tổ chức dưới dạng 1 ma trận n hàng và m cột từ nhớ B bits
 - Cần $n*m = W$ words, kích thước bus địa chỉ = $\lceil \log_2 W \rceil$
 - B đường dữ liệu



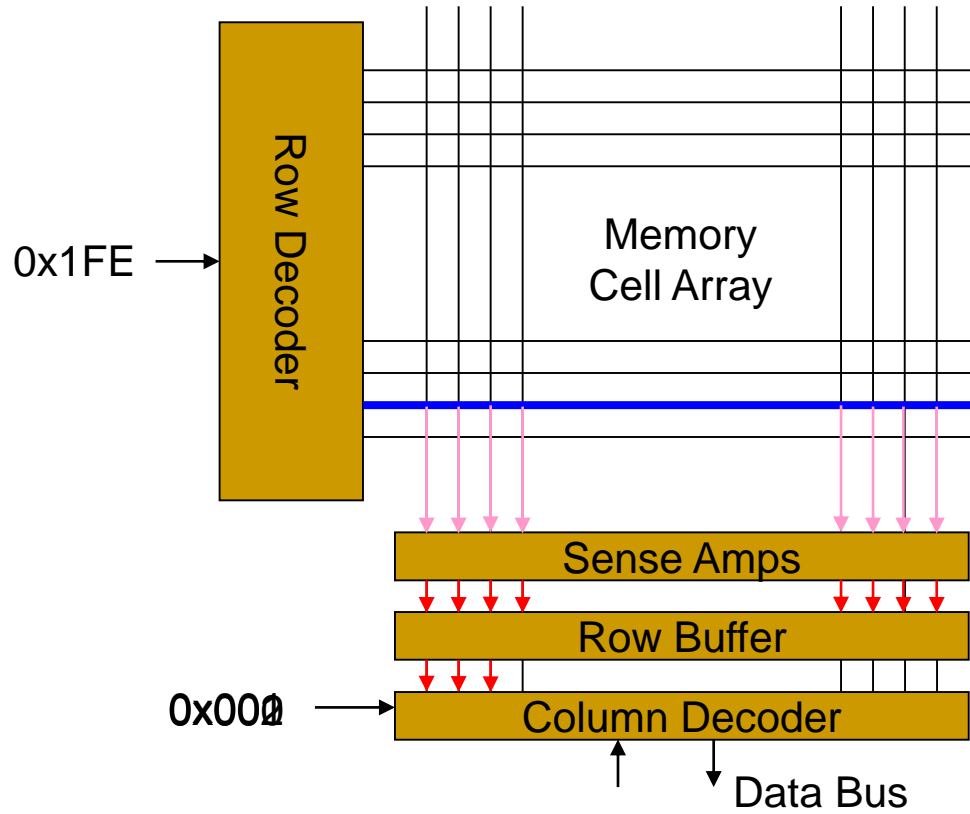
Ví dụ : 16 Mb DRAM (4M x 4)



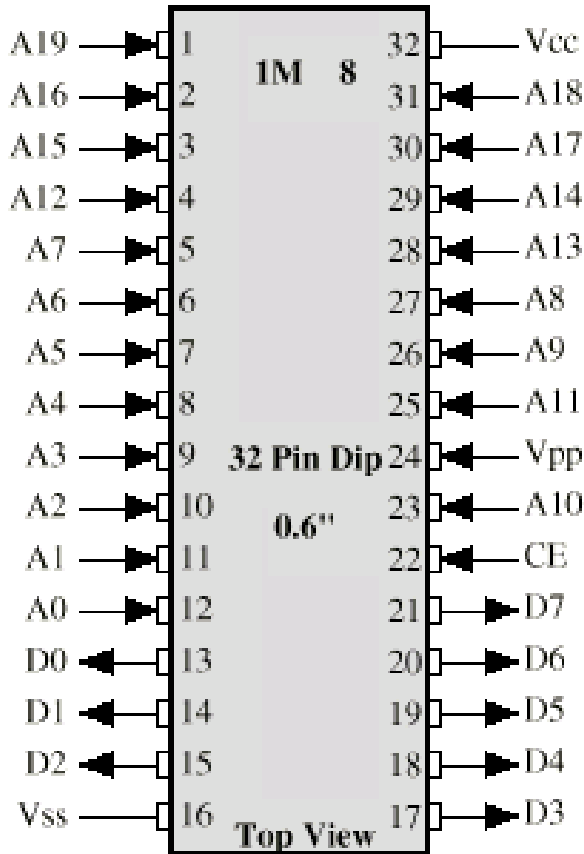
RAS: Row Address Select; CAS: Column Address Select

OE: Output Enable ; WE: Write Enable

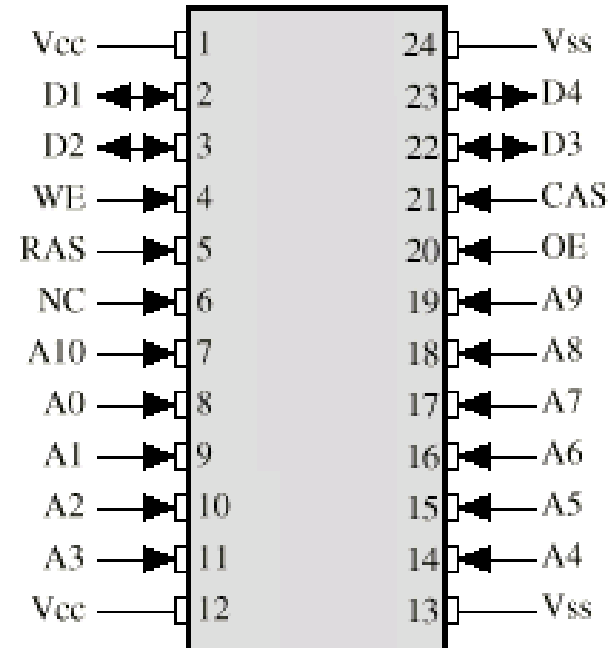
DRAM Read Operation



Packaging

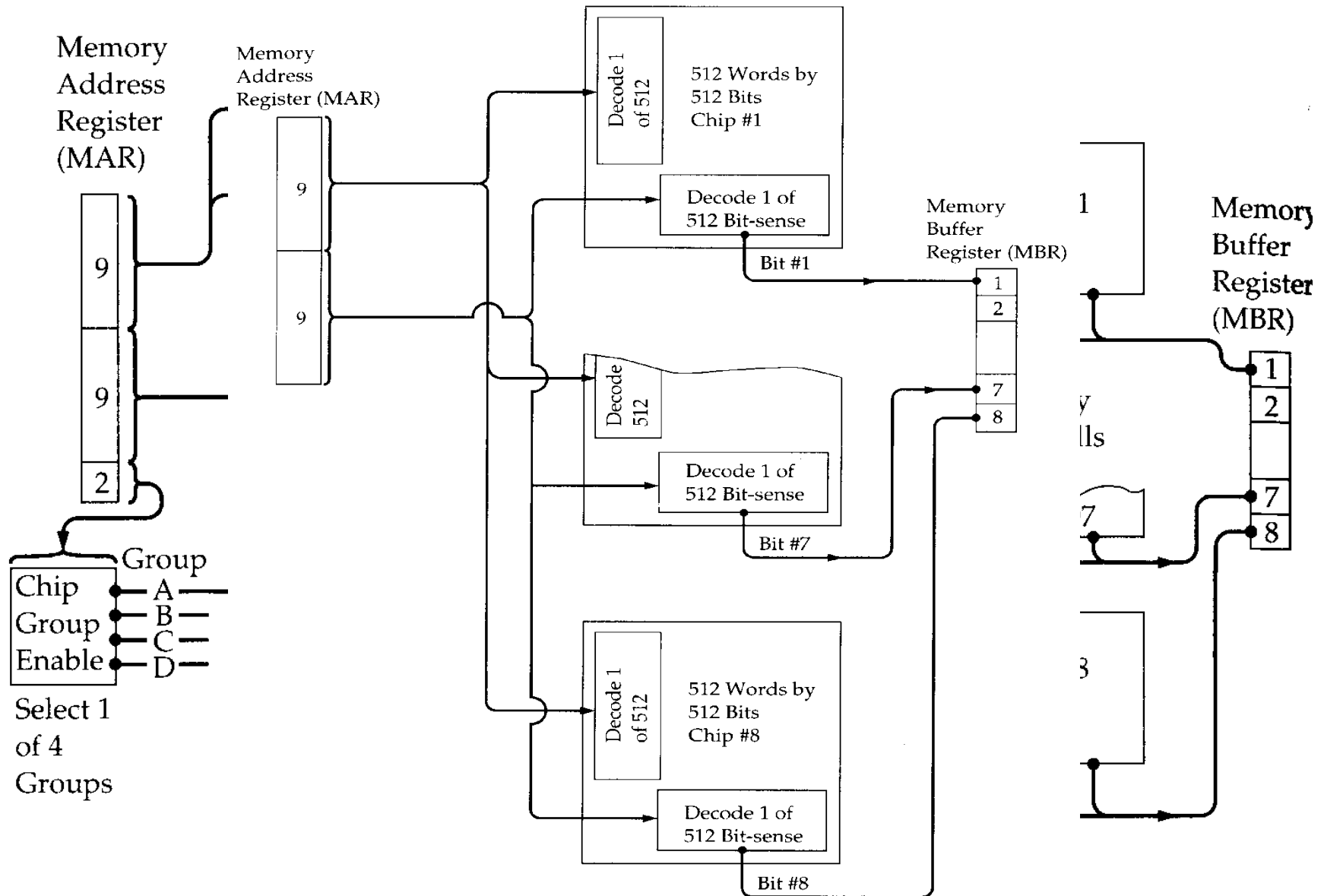


(a) 8 Mbit EPROM



(b) 16 Mbit DRAM

Tổ chức bộ nhớ lớn



Ví dụ

- Xây dựng một không gian nhớ 2^{16} từ 32 bits
 - 4096 từ nhớ trong RAM
 - 4096 từ nhớ trong ROM

Sử dụng:

- chip RAM 1024×8 bits
- chip ROM 4096×8 bits
- Cách thức:
 - Từ = 4 chips 8 bits song song
 - Bản đồ địa chỉ nhớ
 - 4096 từ nhớ RAM ở địa chỉ thấp nhất (0 - 4095)
 - 4096 từ nhớ ROM ở địa chỉ cao nhất (61440 - 65535)
 - Những địa chỉ khác không sử dụng

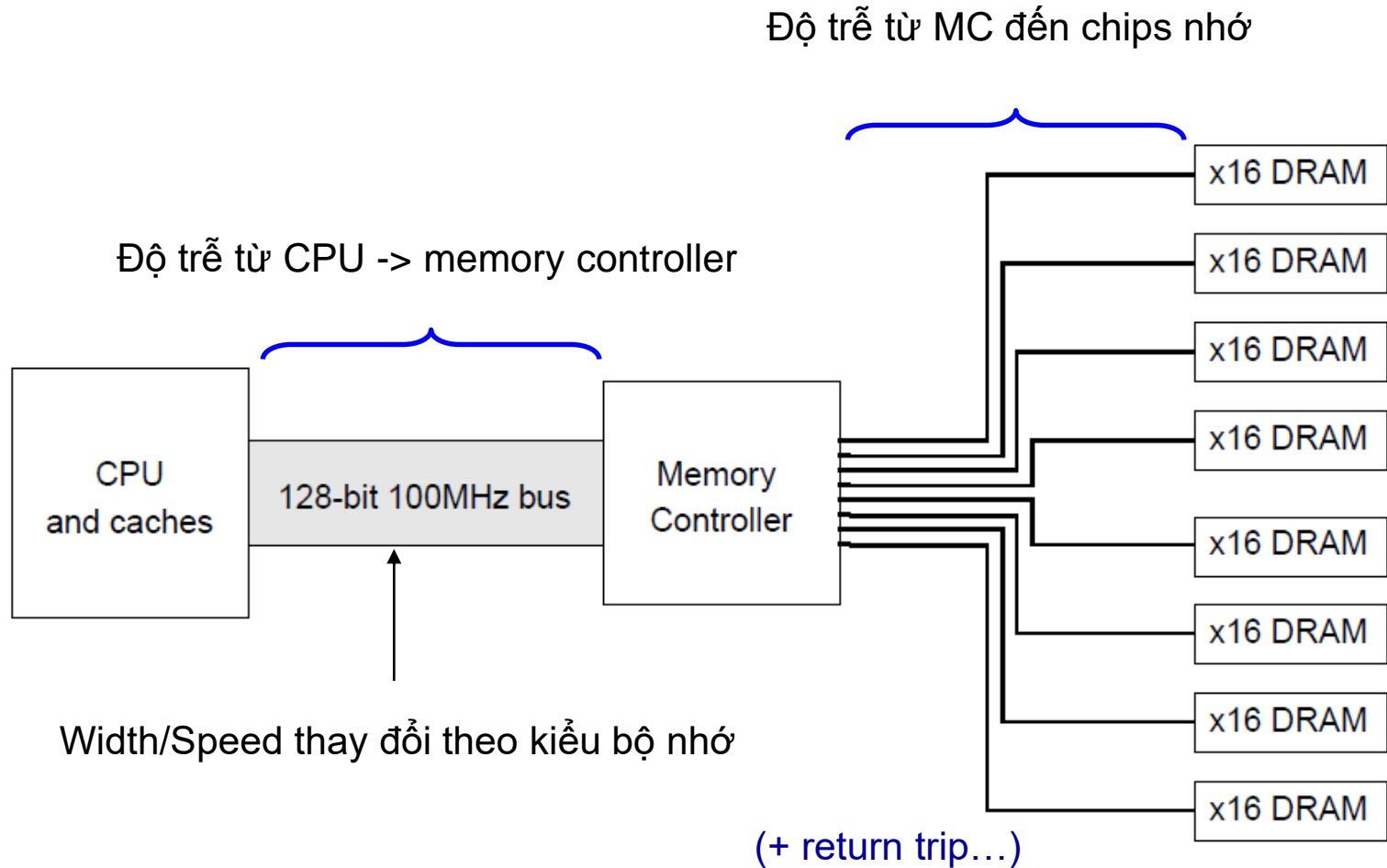
Ví dụ...

- 4 tầng RAM (0 - 4095)
 - 0 -1023
 - 1024 - 2047
 - 2048 - 3071
 - 3072 - 4095
- 1 tầng ROM (61440 - 65535)
- Big Endian (Unix) <> Little Endian (Windows)
 - VD: 0x12.34.56.78

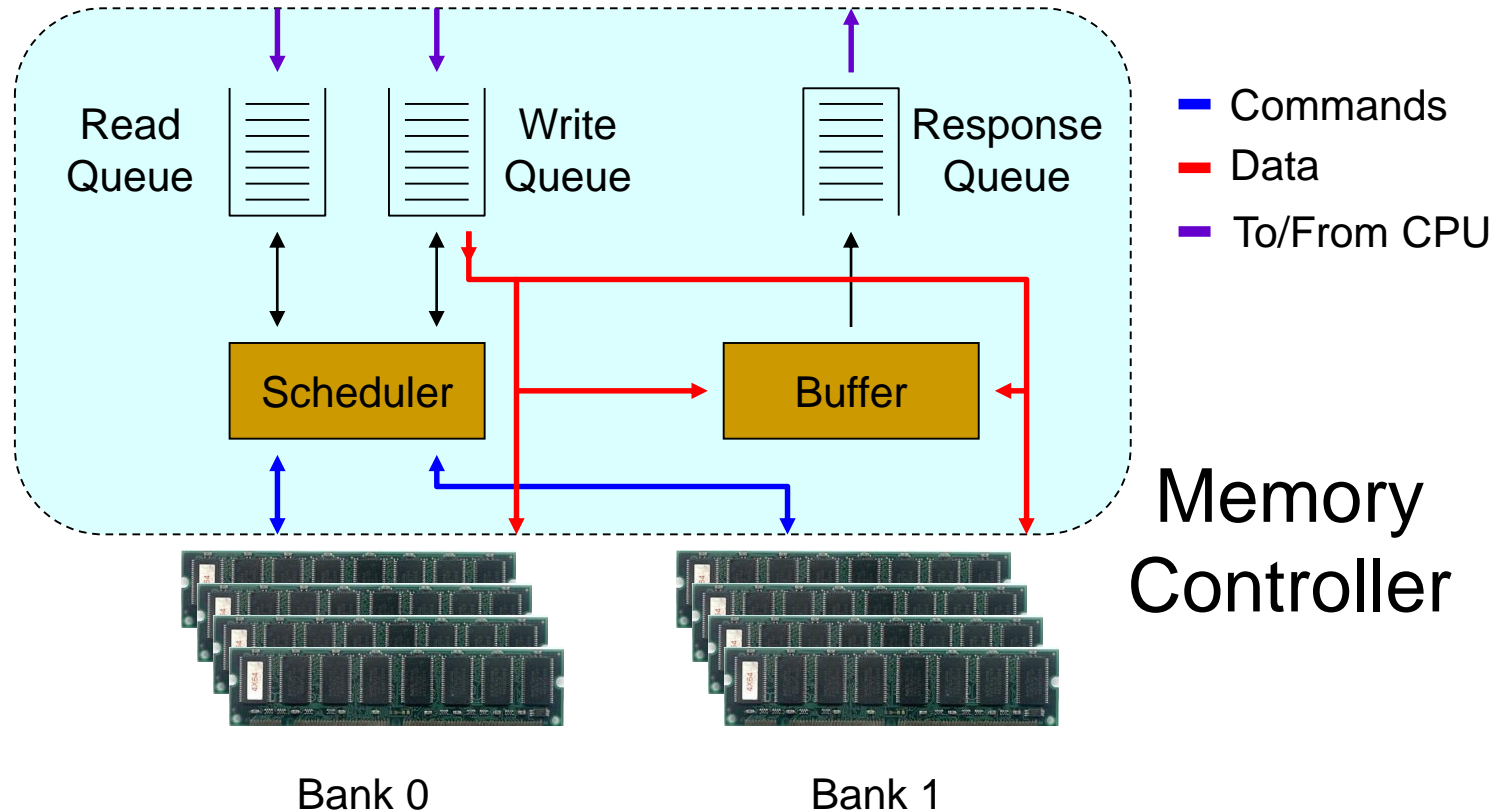
Tầng	Bits địa chỉ
RAM 0	0000 00xx xxxx xxxx
RAM 1	0000 01xx xxxx xxxx
RAM 2	0000 10xx xxxx xxxx
RAM 3	0000 11xx xxxx xxxx
Inoccupied	
ROM	1111 xxxx xxxx xxxx

Adresse	Big Endian	Little Endian
@x	12	78
1+@x	34	56
2+@x	56	34
3+@x	78	12

Latency



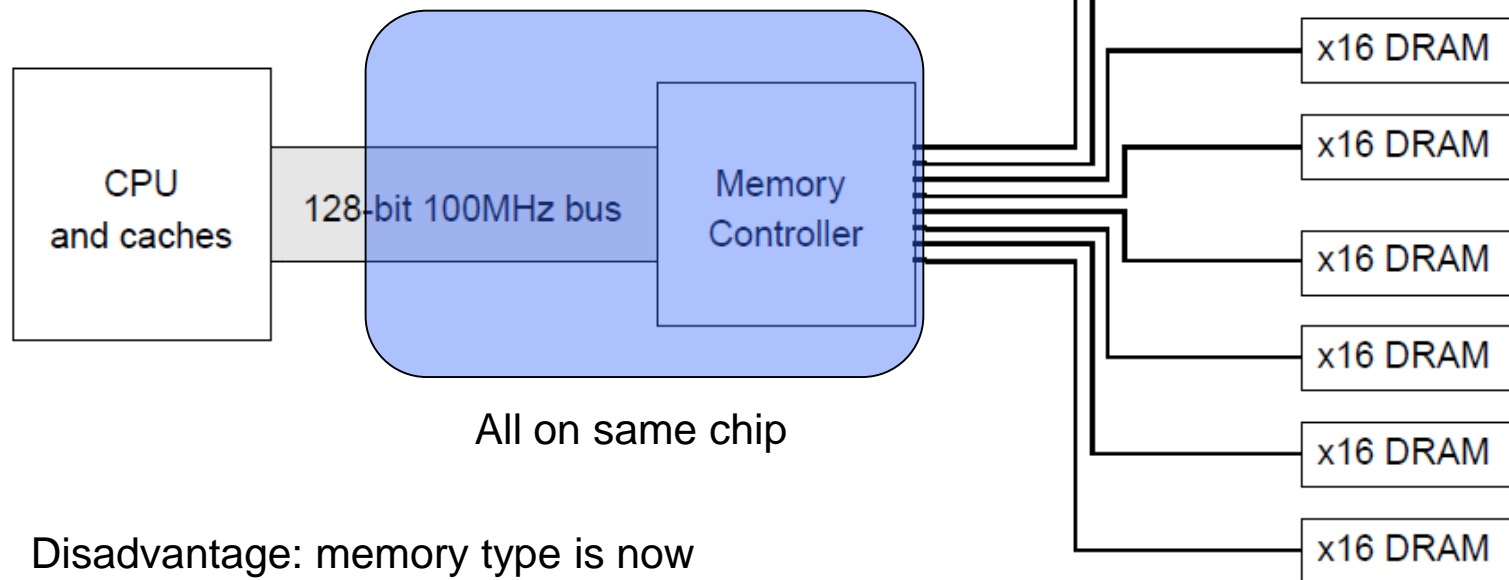
Memory Controller



On-Chip Memory Controller

Also: cơ chế lập lịch phức tạp hơn

Memory controller thi hành cùng tốc độ
CPU thay vì tốc độ xung
FSB



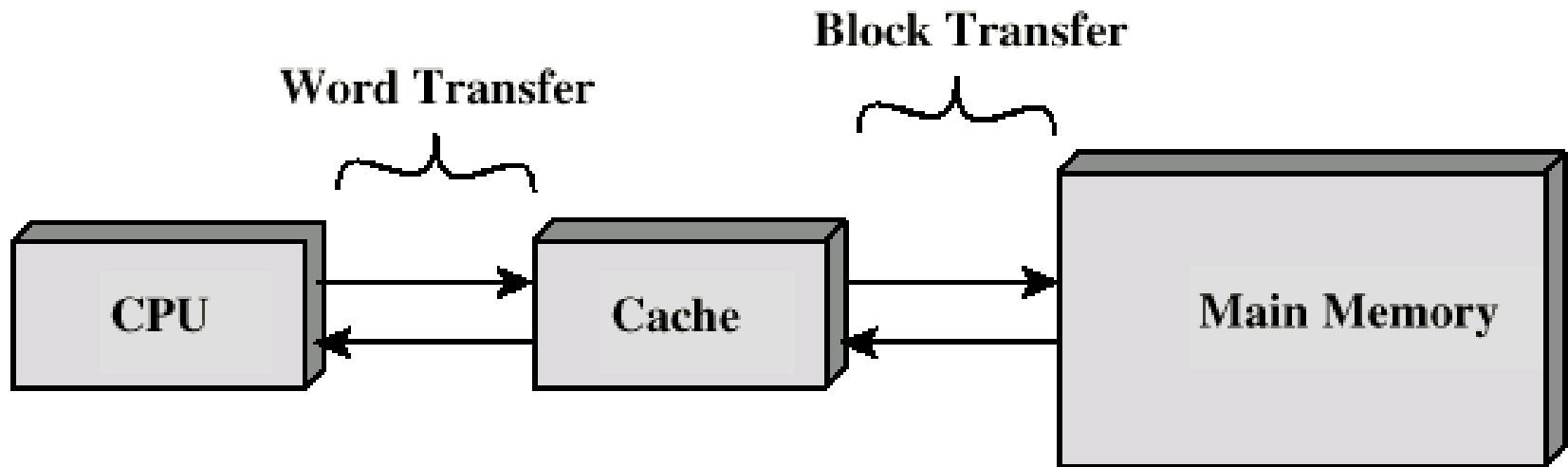
Disadvantage: memory type is now
tied to the CPU implementation

Memory Latency is *Long*

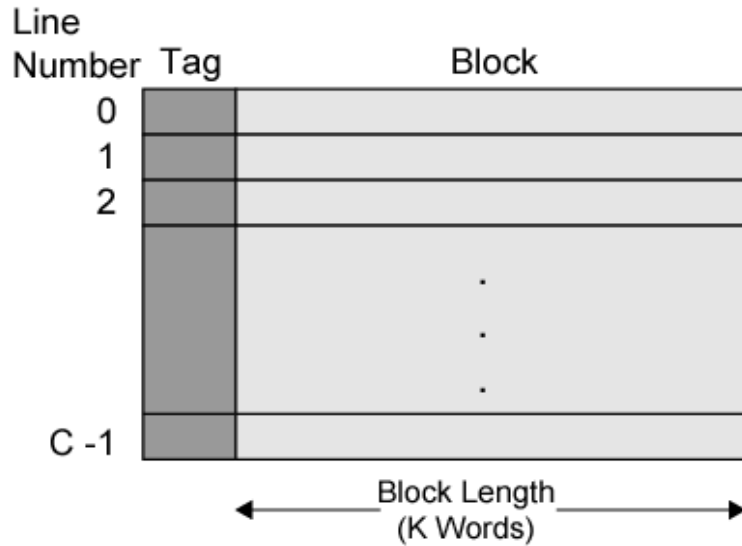
- Thường từ 60-100ns
- Lãng phí thời gian của CPU:
 - 2GHz CPU
 - → 0.5ns / cycle
 - 100ns memory → 200 cycle CPU latency!
- Solution: sử dụng bộ nhớ đệm có tốc độ cao hơn - Caches

3. Cache

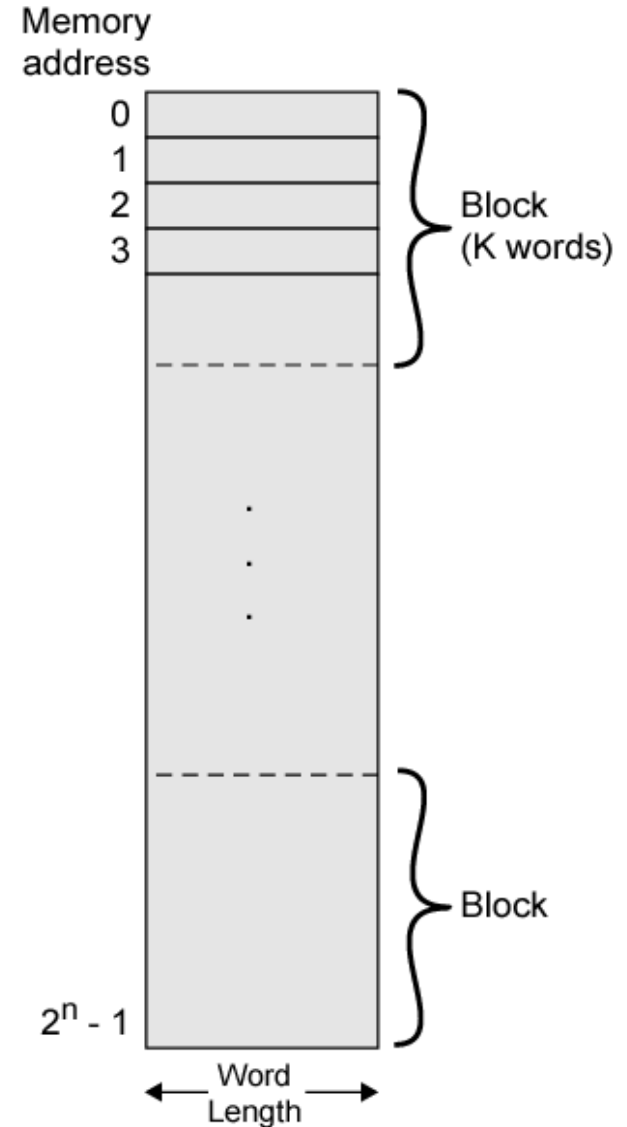
- Bộ nhớ có tốc độ truy cập nhanh
- Thường nằm giữa main memory và CPU
- Có thể nằm trong CPU chip hoặc một module riêng



Cấu trúc Cache/Main Memory



(a) Cache

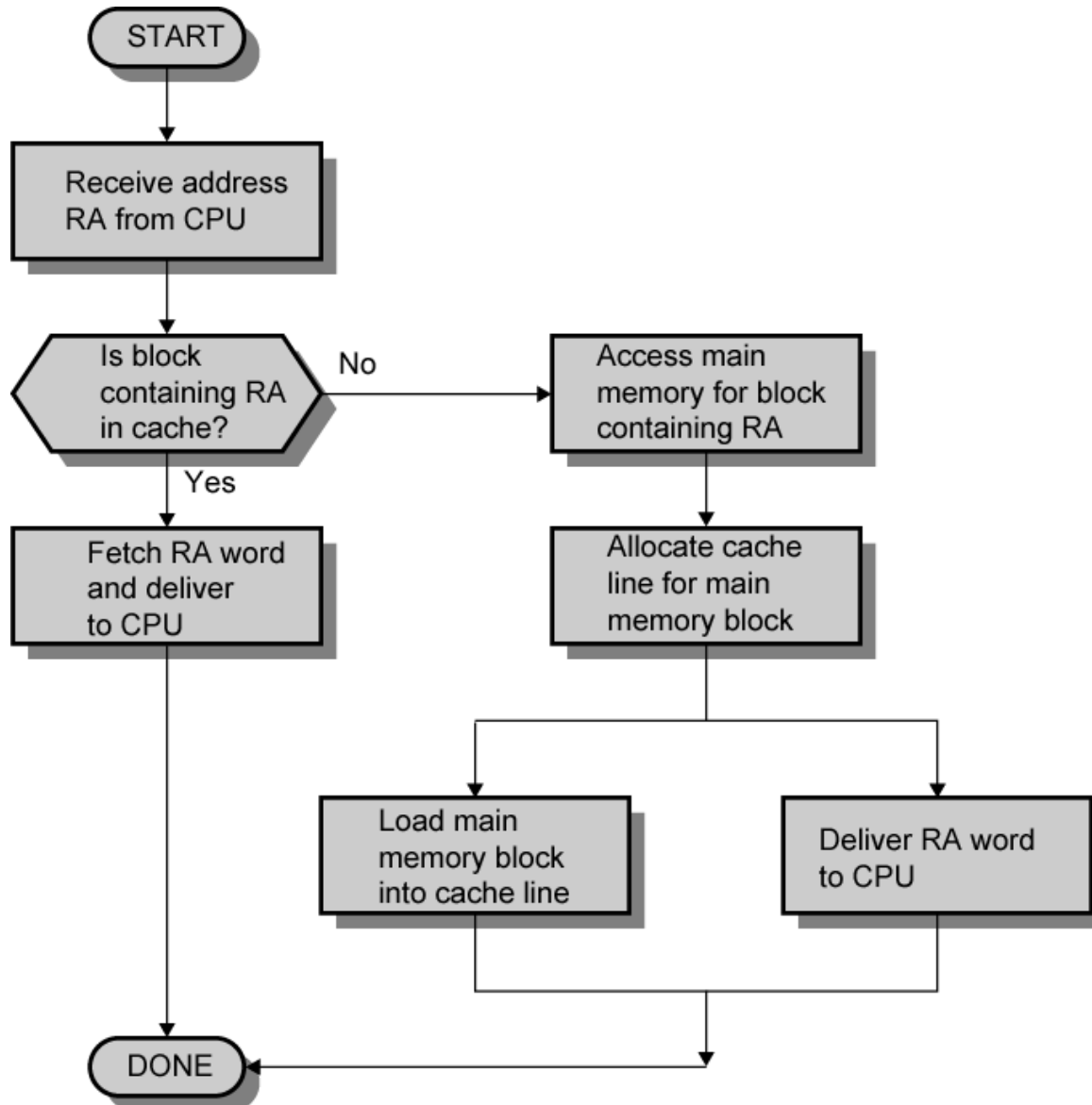


(b) Main memory

Cơ chế hoạt động

- CPU yêu cầu nội dung nhớ tại một vùng nhớ
- Kiểm tra cache đối với dữ liệu đó
- Nếu có, lấy dữ liệu đó từ cache (→ faster)
- Nếu không, đọc block yêu cầu từ main memory vào cache
- Chuyển dữ liệu đó từ cache đến CPU
- Cache sử dụng tags để xác định block nào của main memory trong mỗi khe cache (cache slot)

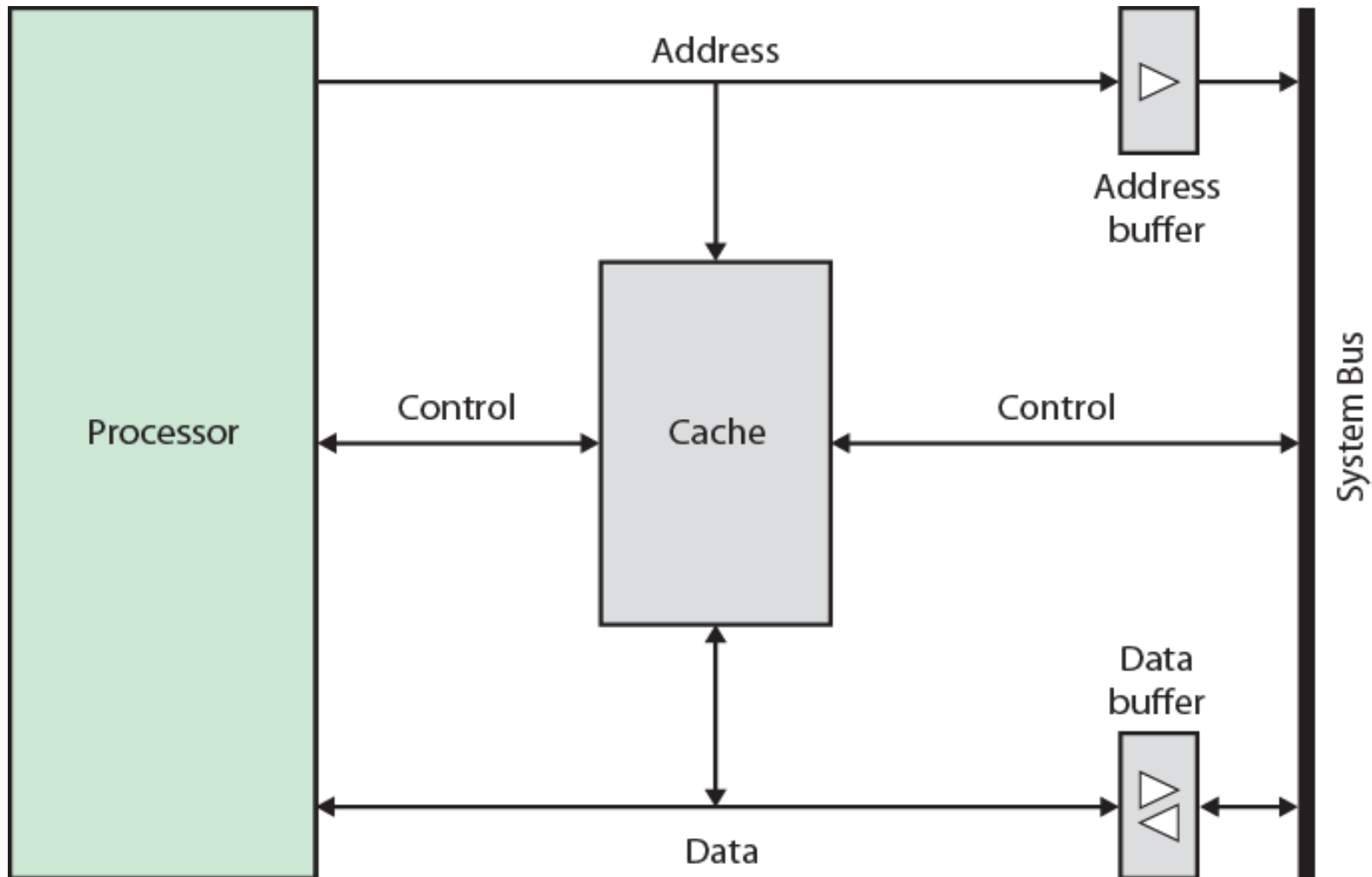
Biểu đồ



Thiết kế Cache

- Size: tốc độ và chi phí
 - More cache is faster (up to a point)
 - Checking cache for data takes time
 - More cache is expensive
- Mapping Function
 - Ánh xạ trực tiếp
 - Ánh xạ kết hợp
 - Ánh xạ kết hợp theo tập
- Replacement Algorithm
- Write Policy
- Block Size
- Number of Caches
 - 1 hay nhiều caches ở mức 1 (bên trong)
 - 1 cache ở mức 2 (trong hoặc ngoài) (thường $V_{L2} \geq 4V_{L1}$)
 - Đôi khi có thể sử dụng 1 cache ở mức 3 (ngoài)

Tổ chức cache điển hình



Bảng so sánh caches một số CPU

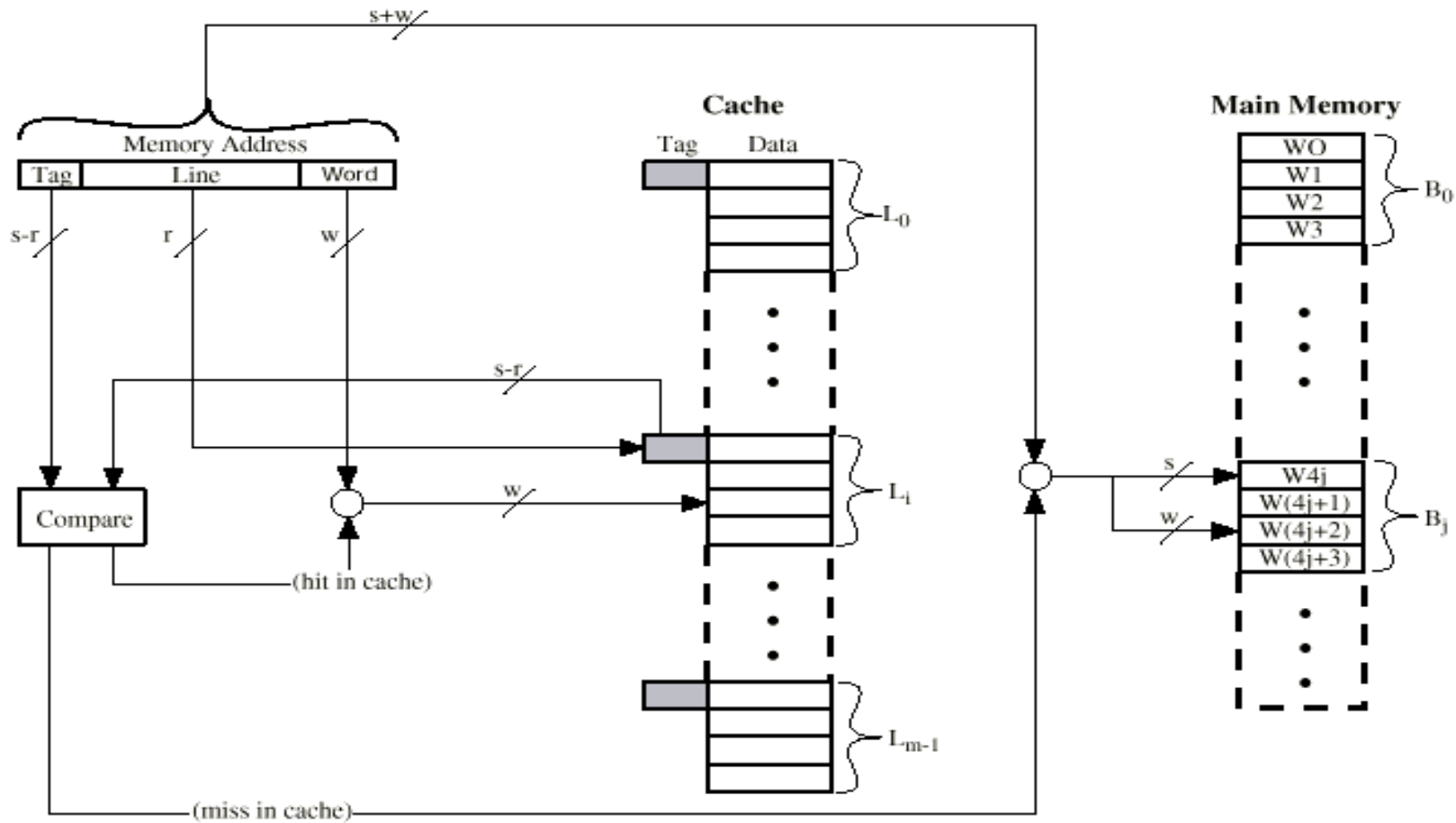
Processor	Type	Year of Introduction	L1 cache ^a	L2 cache	L3 cache
IBM 360/85	Mainframe	1968	16 to 32 KB	—	—
PDP-11/70	Minicomputer	1975	1 KB	—	—
VAX 11/780	Minicomputer	1978	16 KB	—	—
IBM 3033	Mainframe	1978	64 KB	—	—
IBM 3090	Mainframe	1985	128 to 256 KB	—	—
Intel 80486	PC	1989	8 KB	—	—
Pentium	PC	1993	8 KB/8 KB	256 to 512 KB	—
PowerPC 601	PC	1993	32 KB	—	—
PowerPC 620	PC	1996	32 KB/32 KB	—	—
PowerPC G4	PC/server	1999	32 KB/32 KB	256 KB to 1 MB	2 MB
IBM S/390 G4	Mainframe	1997	32 KB	256 KB	2 MB
IBM S/390 G6	Mainframe	1999	256 KB	8 MB	—
Pentium 4	PC/server	2000	8 KB/8 KB	256 KB	—
IBM SP	High-end server/ supercomputer	2000	64 KB/32 KB	8 MB	—
CRAY MTA ^b	Supercomputer	2000	8 KB	2 MB	—
Itanium	PC/server	2001	16 KB/16 KB	96 KB	4 MB
SGI Origin 2001	High-end server	2001	32 KB/32 KB	4 MB	—
Itanium 2	PC/server	2002	32 KB	256 KB	6 MB
IBM POWER5	High-end server	2003	64 KB	1.9 MB	36 MB
CRAY XD-1	Supercomputer	2004	64 KB/64 KB	1MB	—

Ánh xạ cache – main memory

- **Ánh xạ trực tiếp:**
 - Mỗi block của bộ nhớ chính được đặt chỉ vào 1 dòng cache
- **Ánh xạ kết hợp:**
 - 1 block của bộ nhớ chính có thể được đặt vào bất kỳ dòng nào trong cache
- **Ánh xạ kết hợp theo tập:**
 - 1 block của bộ nhớ chính có thể được đặt vào bất kỳ dòng nào trong 1 tập n-blocks của cache

Tag s-r	Cache slot r	Word w
----------------	---------------------	---------------

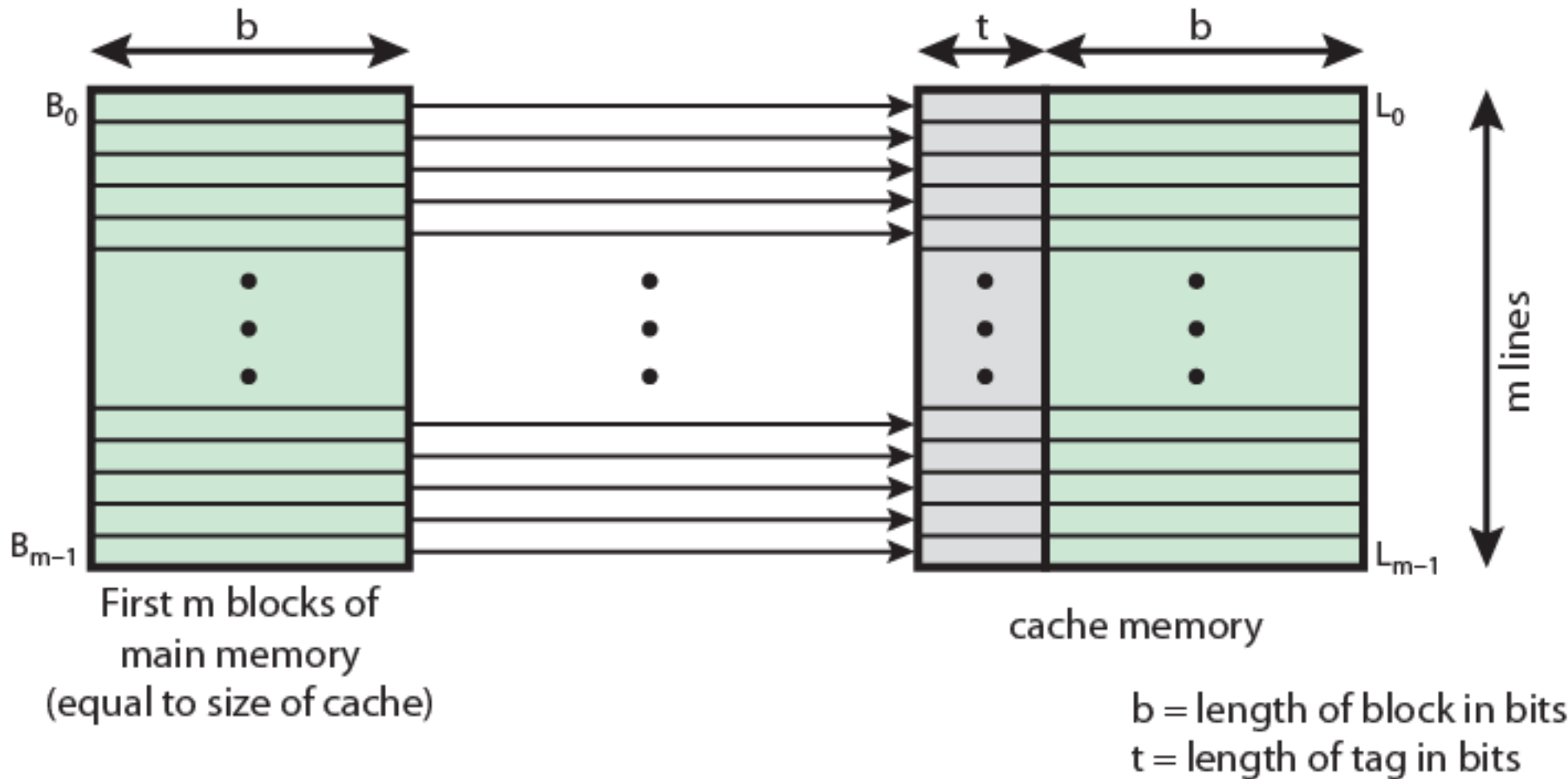
Direct Mapping



- Độ dài: $s+w$ bits
- Số lượng block = 2^s
- Tag = $s-r$ bits

Độ lớn block = 2^w từ
Số slots = 2^r

Ánh xạ trực tiếp từ Cache đến MM



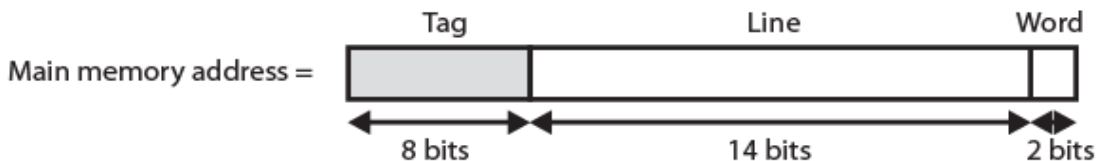
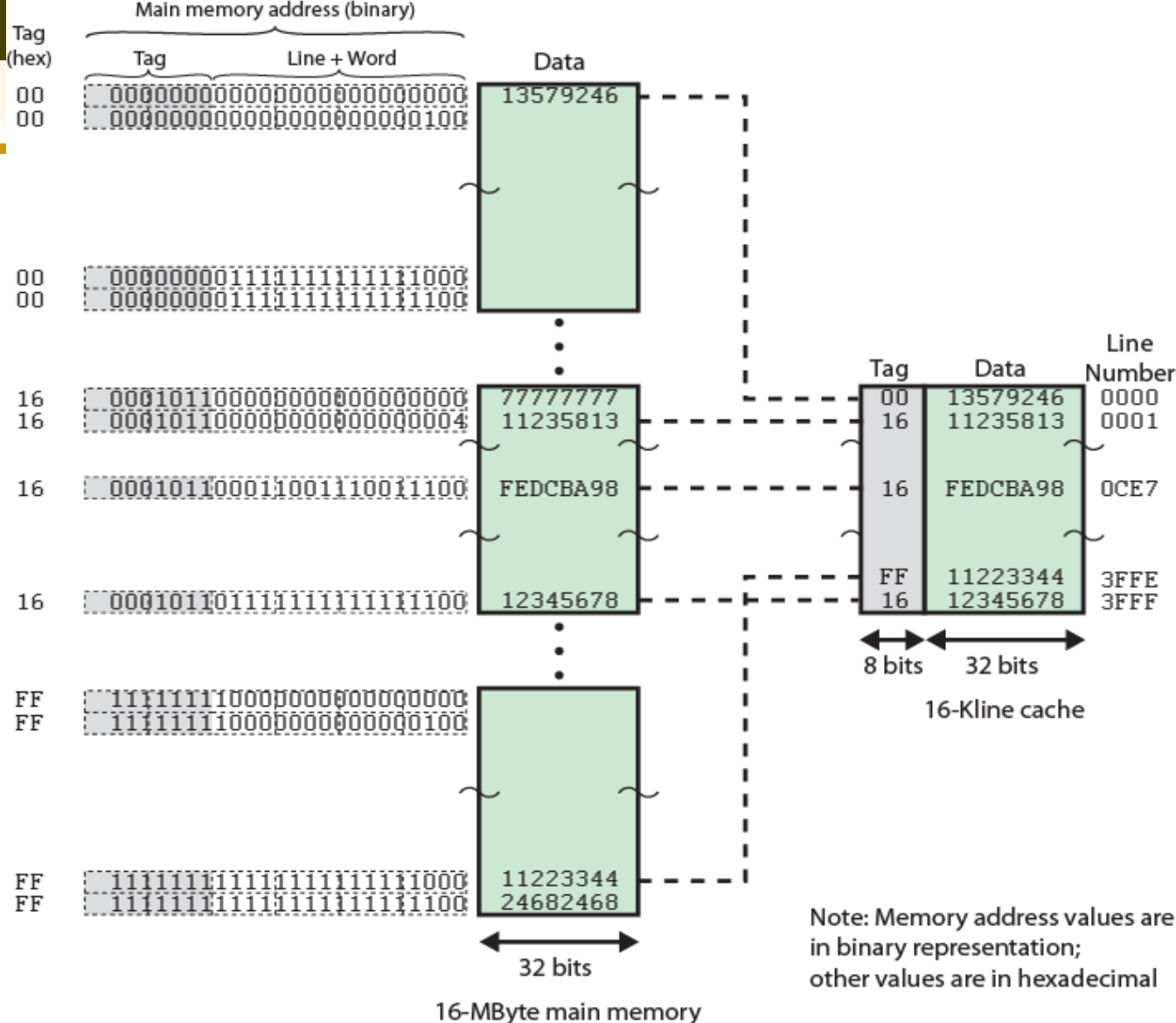
(a) Direct mapping

Bảng ánh xạ trực tiếp

Vị trí slot (cache line)	Khối MM có thể chứa	Tag
0	0, m, 2m, 3m...2s-m	
1	1, m+1, 2m+1...2s-m+1	
...		
m-1	m-1, 2m-1, 3m-1...2s-1	

Ví dụ

- Cache size = 64kByte,
- Cache block = 4 bytes -> có 16k (2^{14}) lines
- Main memory = 16MBytes, 24 bit address ($2^{24}=16M$)

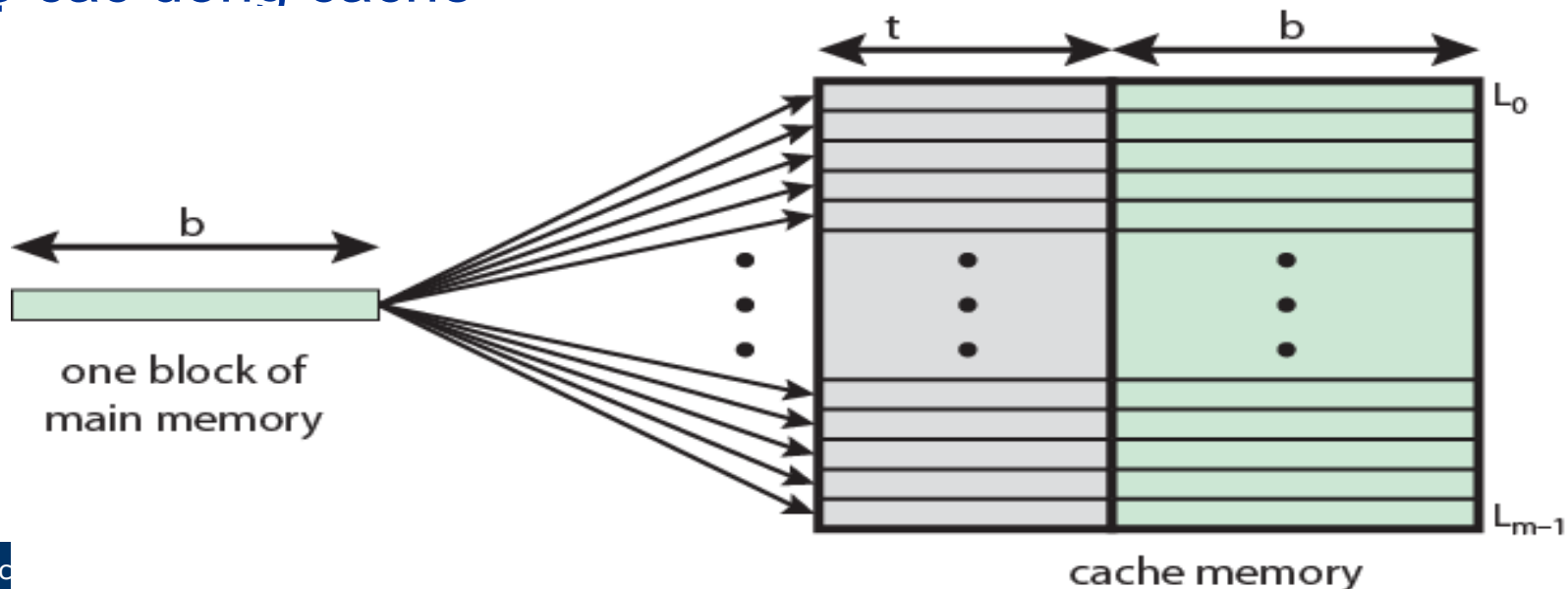


Ưu/Nhược của ánh xạ trực tiếp

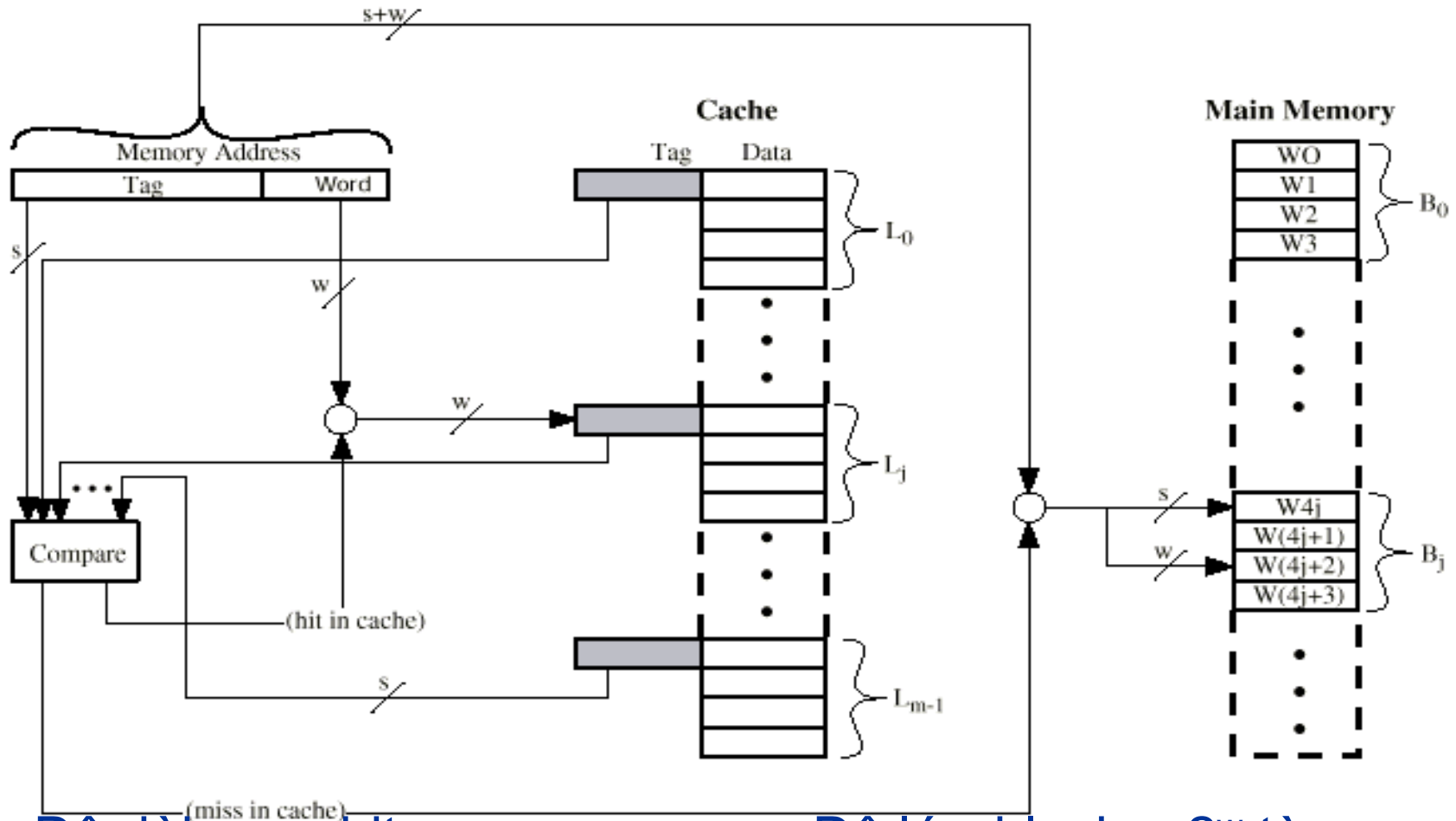
- Đơn giản, không cần thời gian tính toán nhiều
- Chi phí cài đặt thấp
- Mỗi block chỉ có một vị trí duy nhất trong cache (slot tương ứng) → có thể dẫn đến tỷ lệ cache miss tăng lên

Ảnh xạ kết hợp

- MM block có thể được đưa vào bất kỳ dòng cache nào
- Địa chỉ từ nhớ sẽ được chia thành hai phần: tag và word
 - Tag sẽ cho phép xác định địa chỉ block trong MM
 - Word cho phép xác định vị trí từ trong dòng cache tương ứng với địa chỉ Tag
- Việc so khớp cache hit/miss được tiến hành dựa trên trường Tag, có thể dẫn đến trường hợp phải so khớp toàn bộ các dòng cache



Ảnh xạ kết hợp...



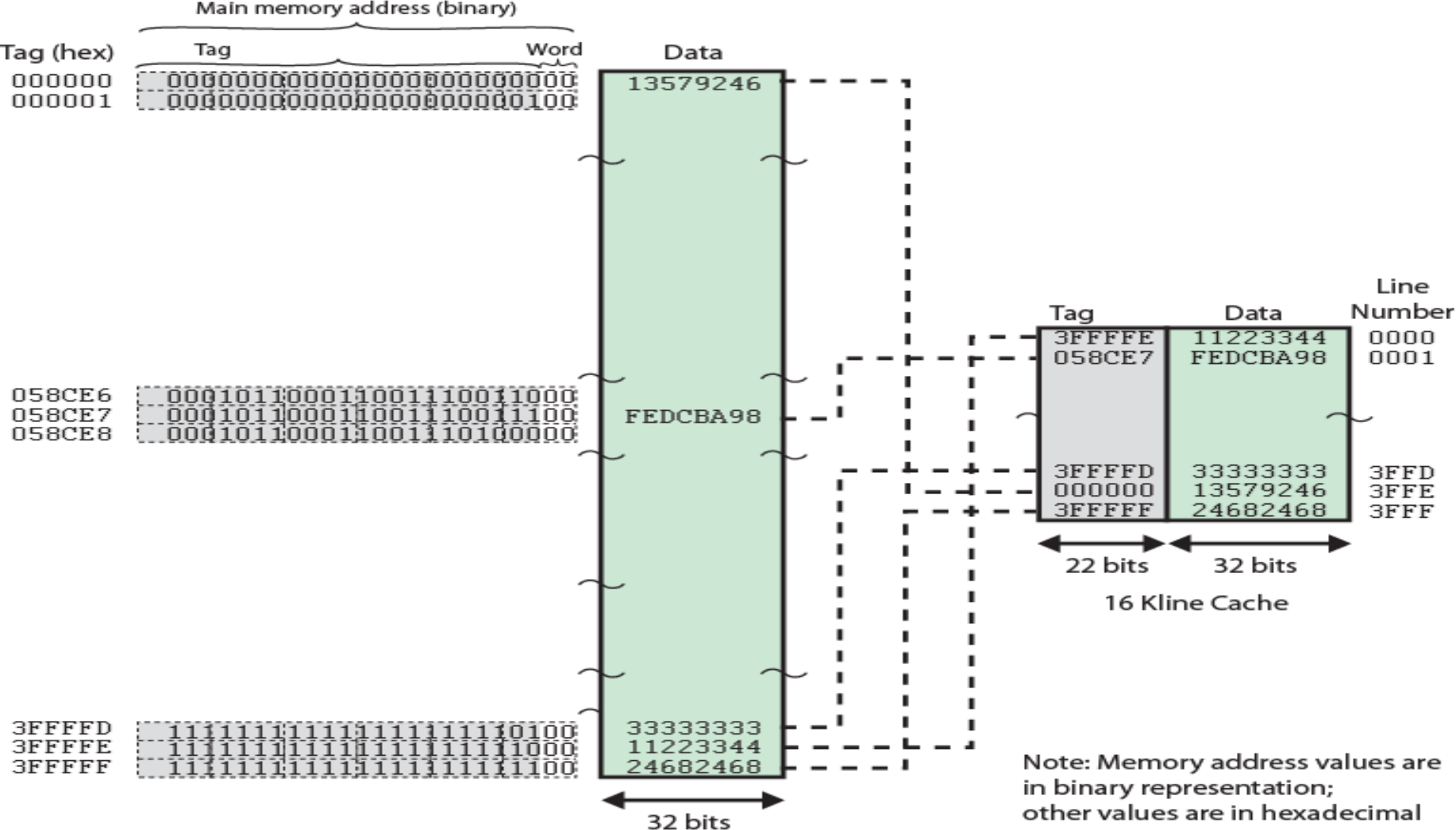
- Độ dài: $s+w$ bits

- Số lượng block = 2^s

- Tag = s bits

Độ lớn block = 2^w từ

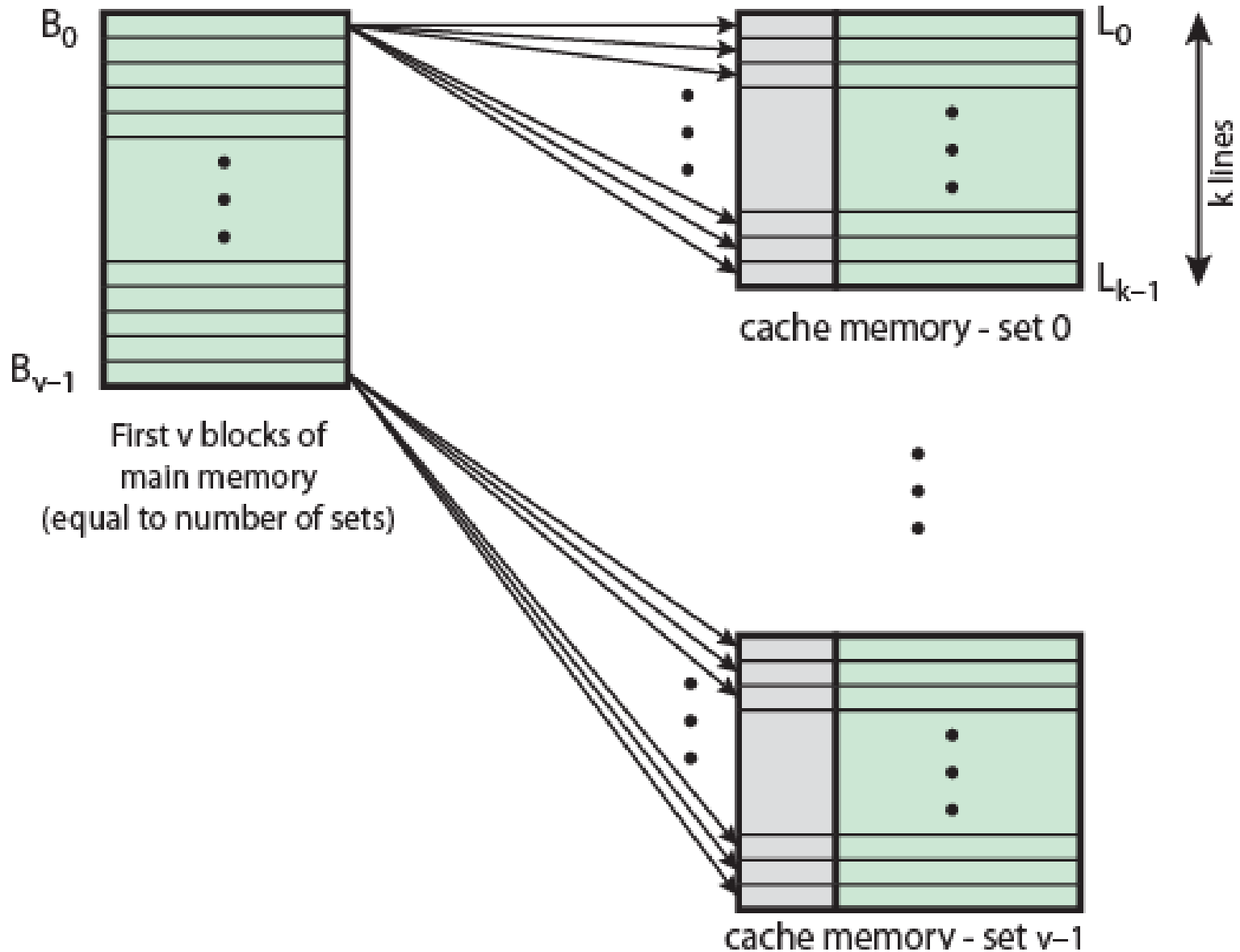
Số slots = undefined



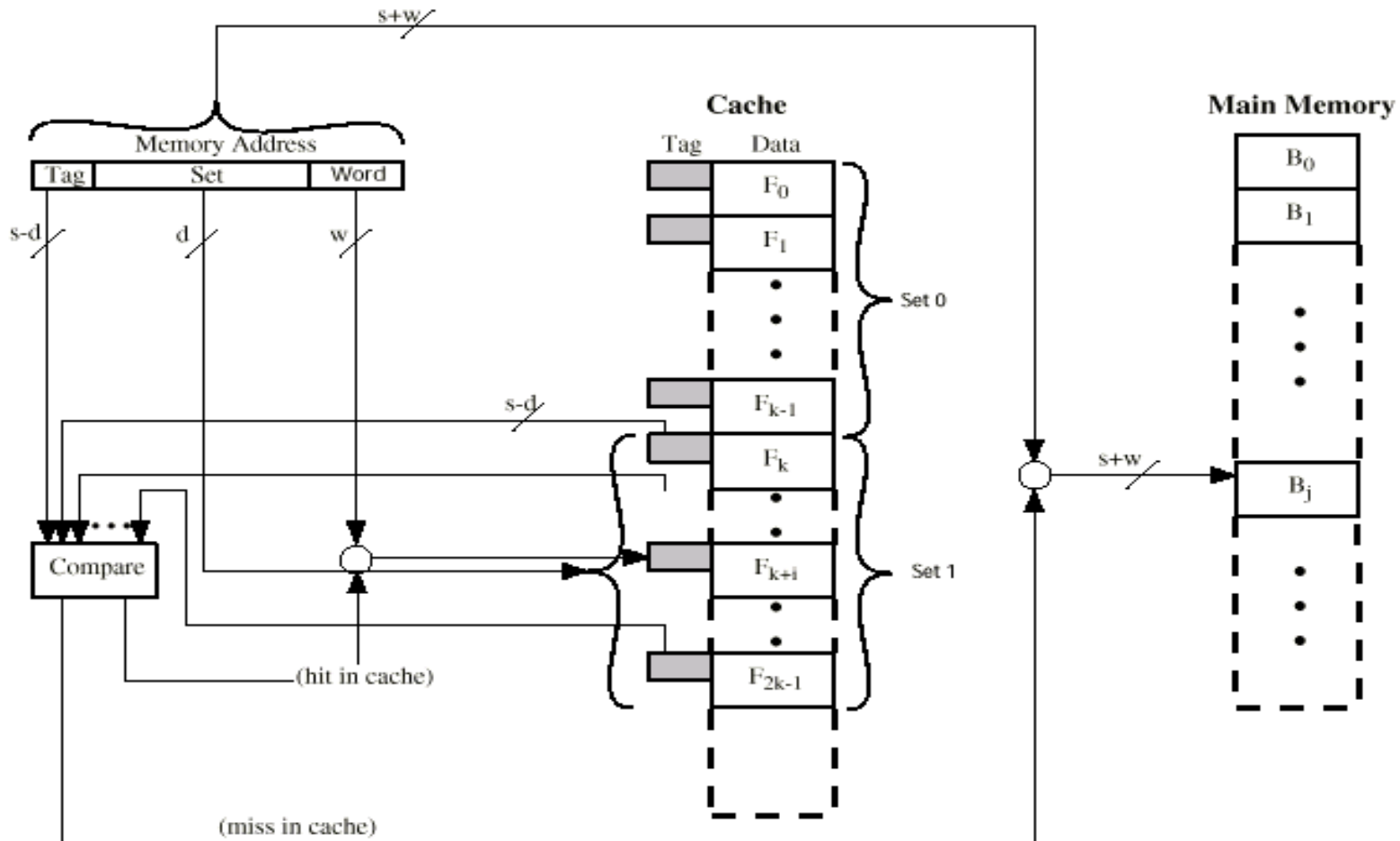
Ánh xạ kết hợp theo tập – Set Associative Mapping

- Cache được chia thành nhiều tập, mỗi tập chứa một số slots
- Mỗi block trong MM được đưa vào một tập duy nhất theo cơ chế ánh xạ trực tiếp; tuy nhiên, nó có thể được đặt vào bất kỳ slot nào trong tập đó theo cơ chế ánh xạ kết hợp
 - Chẳng hạn với 2 lines mỗi tập, mỗi block có thể ở 1 trong 2 lines tại vị trí một tập duy nhất
- Với k dòng, cơ chế ánh xạ này được gọi là k-Way Set Associative Mapping

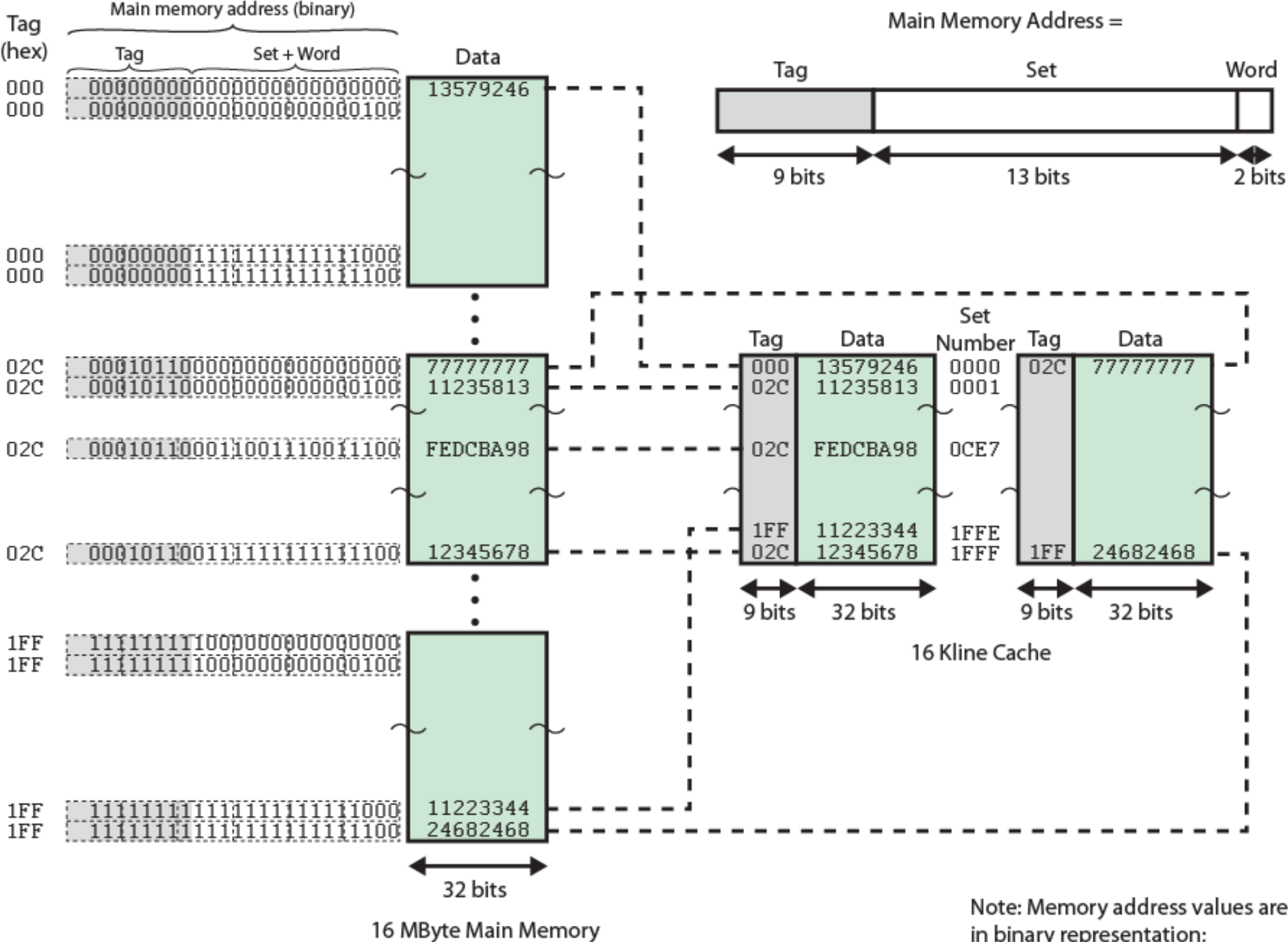
Set Associative Mapping...



k-Way Set Associative

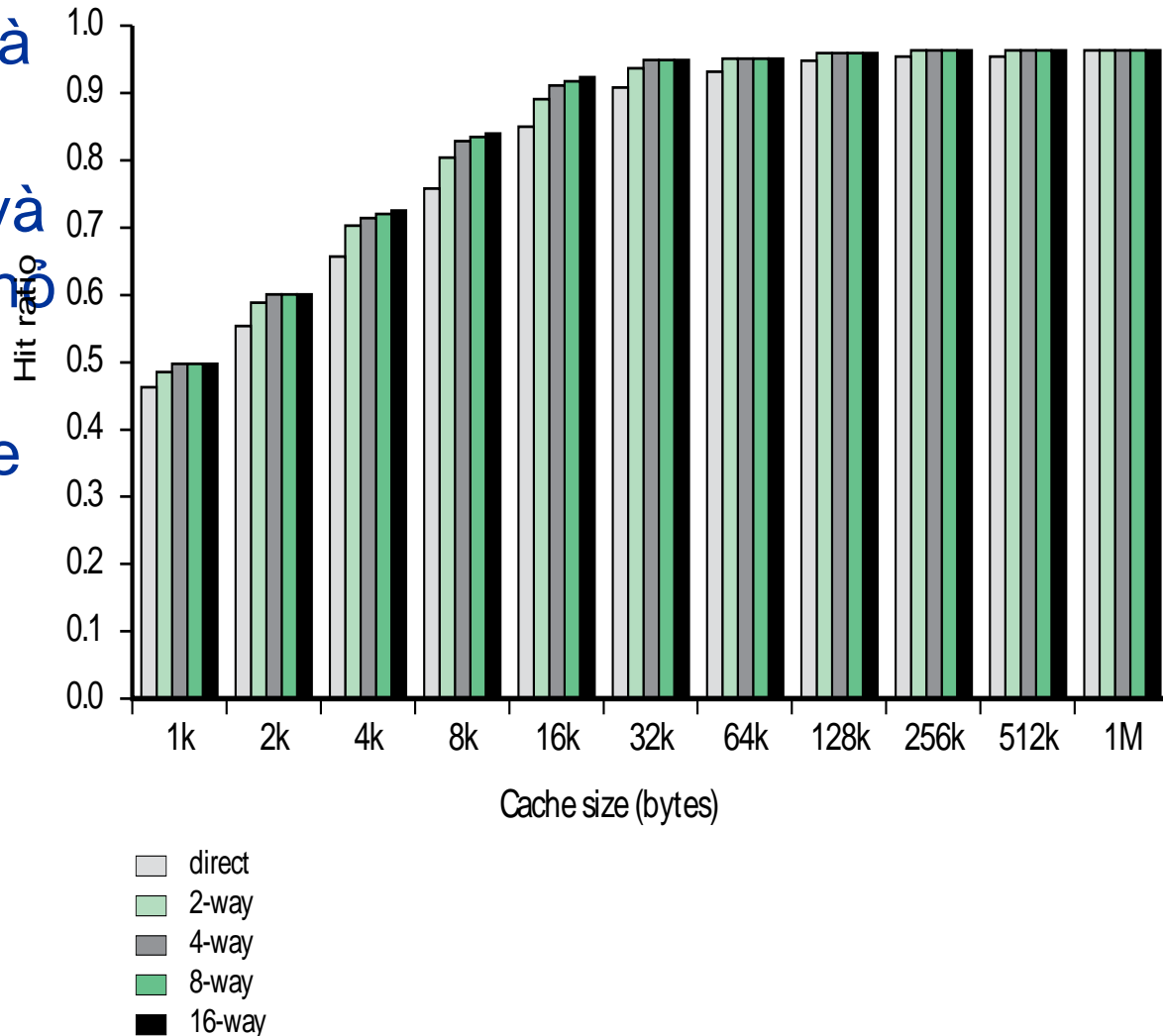


- Độ dài: $s+w$ bits
 - Số lượng block = 2^s
 - Số set $v = 2^d$, Số slots = $k.v$
- Độ lớn block = 2^w từ
Số lines/set = k
Tag = $s-d$ bits



So sánh hiệu năng

- Khác biệt giữa direct và k-way tối đa đến 64kB
- Khác biệt giữa 2-way và 4-way tại cache 4kB nhỏ hơn so với cache 8kB
- Độ phức tạp của cache tăng theo độ kết hợp
- Không có lý do gì để không tăng cache lên 8kB hay 16kB
- Từ 32kB trở lên không cải thiện nhiều về hiệu năng



Truy cập cache

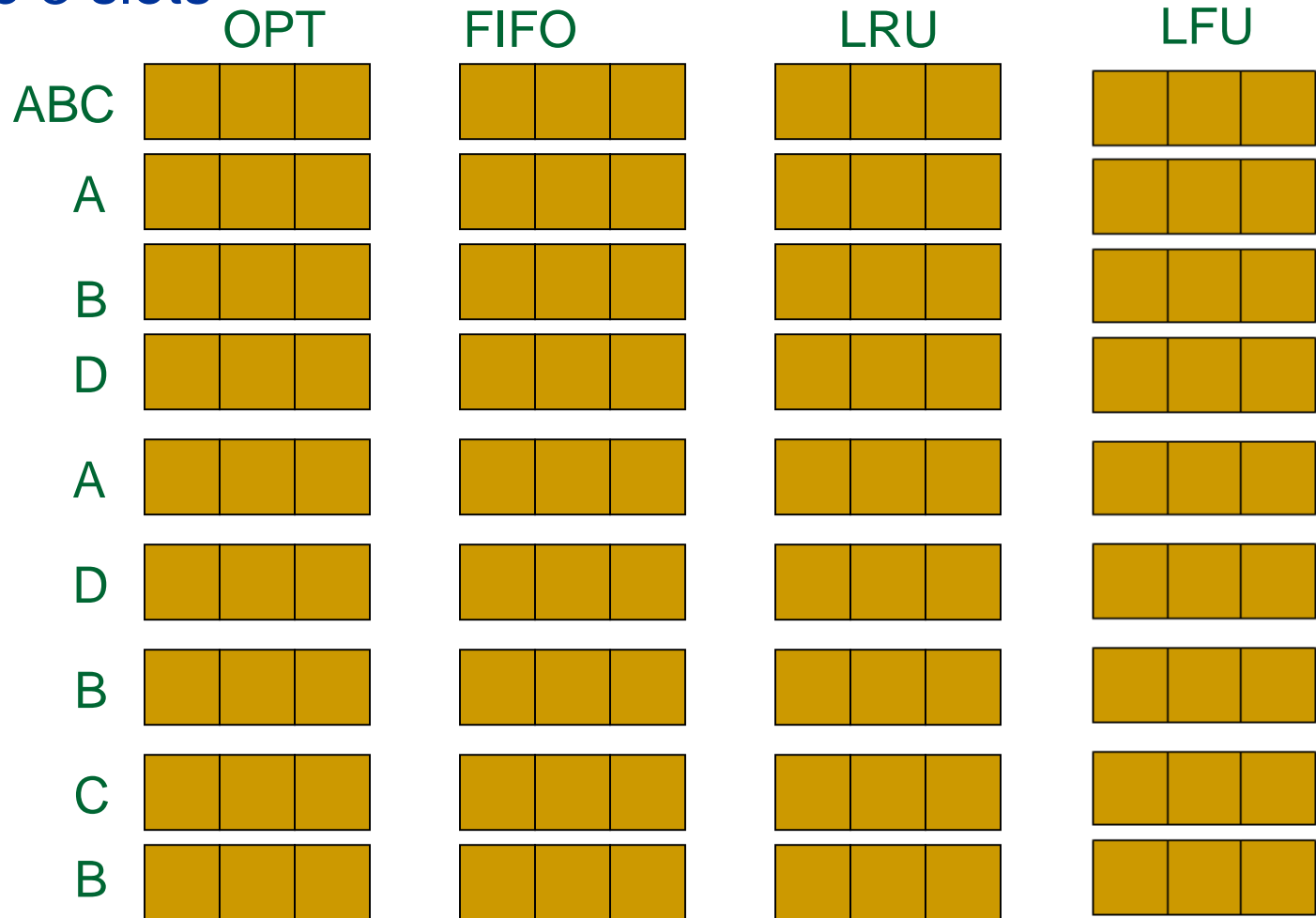
- Xây dựng địa chỉ nhớ dựa trên hàm ánh xạ
- Địa chỉ nhớ 1 từ cho phép tìm được:
 - block mà nó phụ thuộc
 - Vị trí của nó trong block đó
 - Vị trí block đó trong cache
- Cấu trúc: 2 phần
 - Số block:
 - index: vị trí block trong cache
 - tag: xác định block trong số những blocks có cùng vị trí
 - Độ rời: địa chỉ của từ trong block

Giải thuật thay thế

- Ánh xạ trực tiếp: không có sự lựa chọn
 - 1 block được đặt chỉ vào 1 vị trí
- Ánh xạ kết hợp – kết hợp tập:
 - Chọn block ngẫu nhiên
 - FIFO: Chọn block **cũ nhất** trong cache
 - Chọn block được sử dụng sớm nhất (LRU Least Recently Used)
 - Chọn block có tần xuất sử dụng ít nhất (LFU Least Frequently Used)
- Hiệu quả nhất: LFU, LRU, ngẫu nhiên
- Dễ cài đặt: ngẫu nhiên, FIFO

Ví dụ

- Xét chuỗi truy cập các khối MM có địa chỉ: ABCABDADBCBCBCB
- Cache có 3 slots



Chính sách Write

- Không được ghi đè một cache block trừ khi CPU cập nhật dữ liệu đó
- Mỗi CPU có thể có cache riêng
- I/O có thể truy cập trực tiếp main memory
- Phân biệt:
 - Write through
 - Write back

Write through

- Ghi cả main memory lẫn cache
- Nếu có nhiều CPU, cần quản lý lưu lượng của bộ nhớ chính để giữ cache cục bộ được cập nhật
- Giảm tốc độ ghi, gây nhiều traffic

Write back

- Cập nhật đầu tiên trong cache, lưu trạng thái trong bit cập nhật của cache slot (update bit = set)
- Nếu block có yêu cầu thay đổi dữ liệu, ghi nội dung hiện thời vào main memory chỉ khi update bit đã được thiết lập
- I/O phải truy cập main memory thông qua cache
- Khoảng 15% truy cập bộ nhớ là thao tác ghi

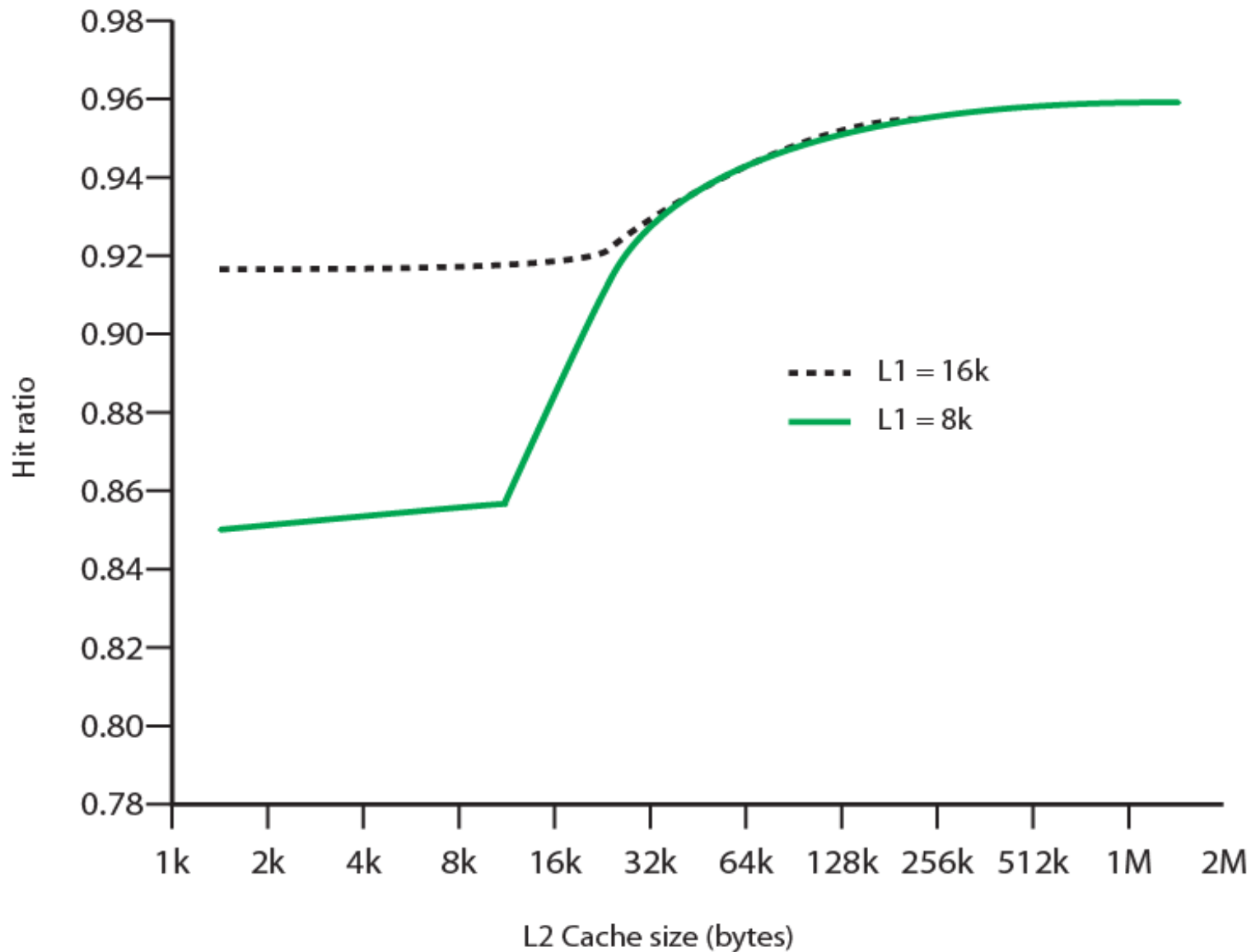
Độ lớn đường cache (slot size)

- Tăng kích thước slot sẽ tăng tỷ lệ hit (dựa trên nguyên lý cục bộ của chương trình)
- Tỷ lệ hit sẽ giảm nếu kích thước block quá lớn
 - Xác suất sử dụng dữ liệu mới tải lên nhỏ hơn xác suất sử dụng lại dữ liệu được thay thế
- Block lớn hơn
 - Giảm số blocks nằm trong cache
 - Khả năng dữ liệu bị đè sau khi được tải vào tăng lên
 - Các từ nhớ có tính cục bộ nhỏ đi → giảm tỷ lệ hit
- Không có giá trị tối ưu được xác định chính thức, từ 8 đến 64 bytes là giá trị hợp lý
- Với những hệ thống tính toán hiệu năng cao, kích thước block thường từ 64 đến 128 bytes

Cache đa mức - Multilevel Caches

- Hầu hết các CPU ngày này đều có cache on chip (do có mật độ logic gates tăng lên)
 - ➔ Nhanh hơn vì không cần sử dụng bus hệ thống
 - ➔ Bus được sử dụng cho những mục đích khác
- Thường thì cache đa mức được cài đặt trong cả trong và ngoài chip
 - L1 on chip, L2 off chip in static RAM
 - L2 access much faster than DRAM or ROM
 - L2 often uses separate data path
 - L2 may now be on chip
 - Resulting in L3 cache
 - Bus access or now on chip...

Tỷ lệ Hit (L1 & L2) với 8 kB và 16 kB L1



Unified v Split Caches

- Unified Cache: cache chung cho cả lệnh và dữ liệu
 - ➔ Tỷ lệ hit cao hơn; cân bằng tải lệnh và dữ liệu; chỉ phải thiết kế, cài đặt một bộ nhớ cache
- Split Cache: một cache cho lệnh và một cache cho dữ liệu
 - ➔ Loại được sự cạnh tranh cache (cache contention) trong chu trình tải/giải mã lệnh và chu trình thi hành (nâng cao hiệu năng pipeline)

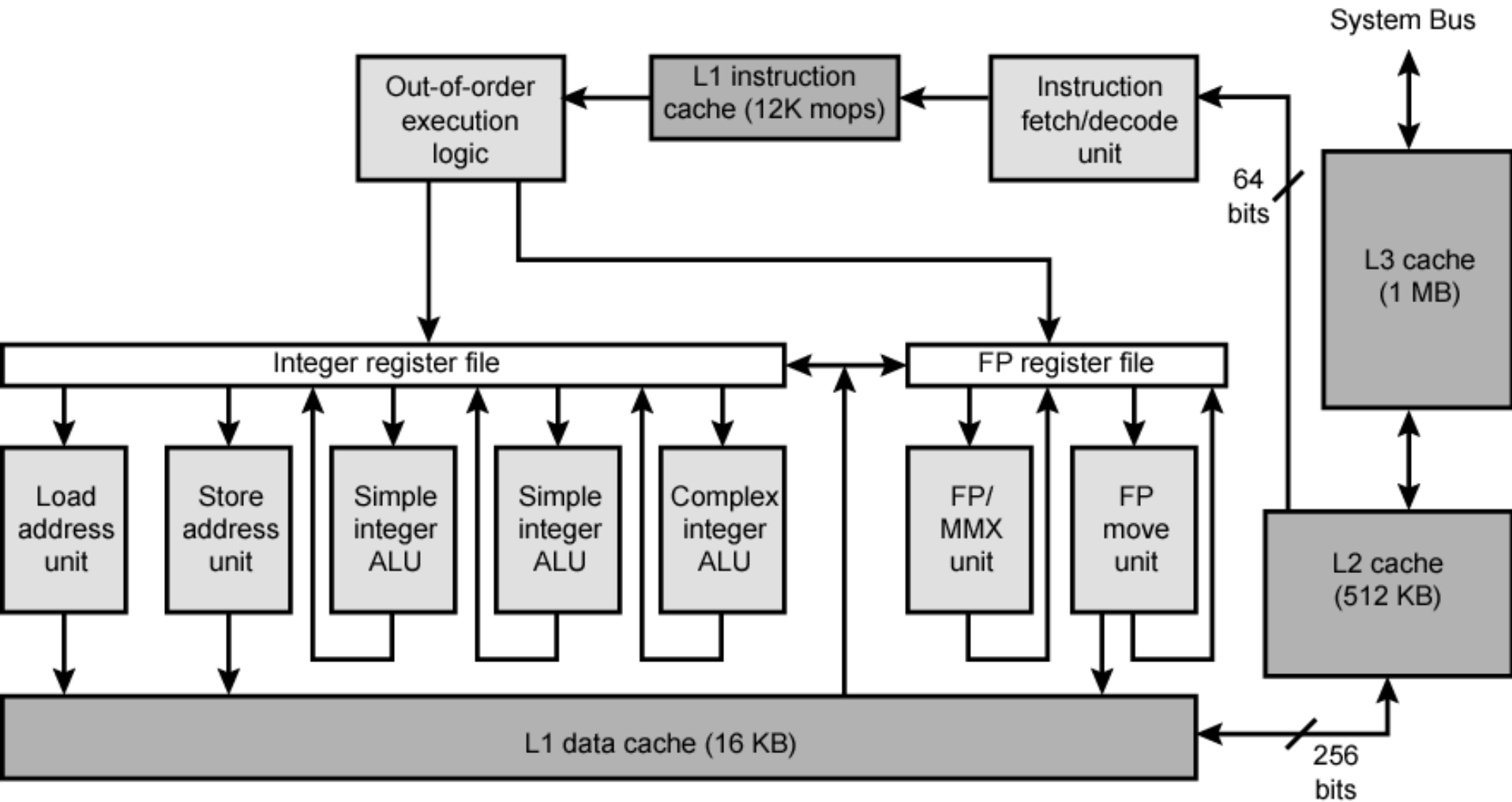
Hiệu năng

- Thời gian truy cập trung bình = Thời gian truy cập thành công + tỷ lệ thất bại x penalty thất bại
- Thời gian truy cập = Thời gian truy cập một dữ liệu trong cache
- Tỷ lệ thất bại = số lần dữ liệu cần đọc không chứa trong cache / số lần truy cập cache

Pentium Cache

- 80386 – no on chip cache
- 80486 – 8k using 16 byte lines and four way set associative organization
- Pentium (all versions) – two on chip L1 caches
 - Data & instructions
- Pentium III – L3 cache added off chip
- Pentium 4
 - L1 caches
 - 8k bytes
 - 64 byte lines
 - four way set associative
 - L2 cache
 - Feeding both L1 caches
 - 256k
 - 128 byte lines
 - 8 way set associative
 - L3 cache on chip

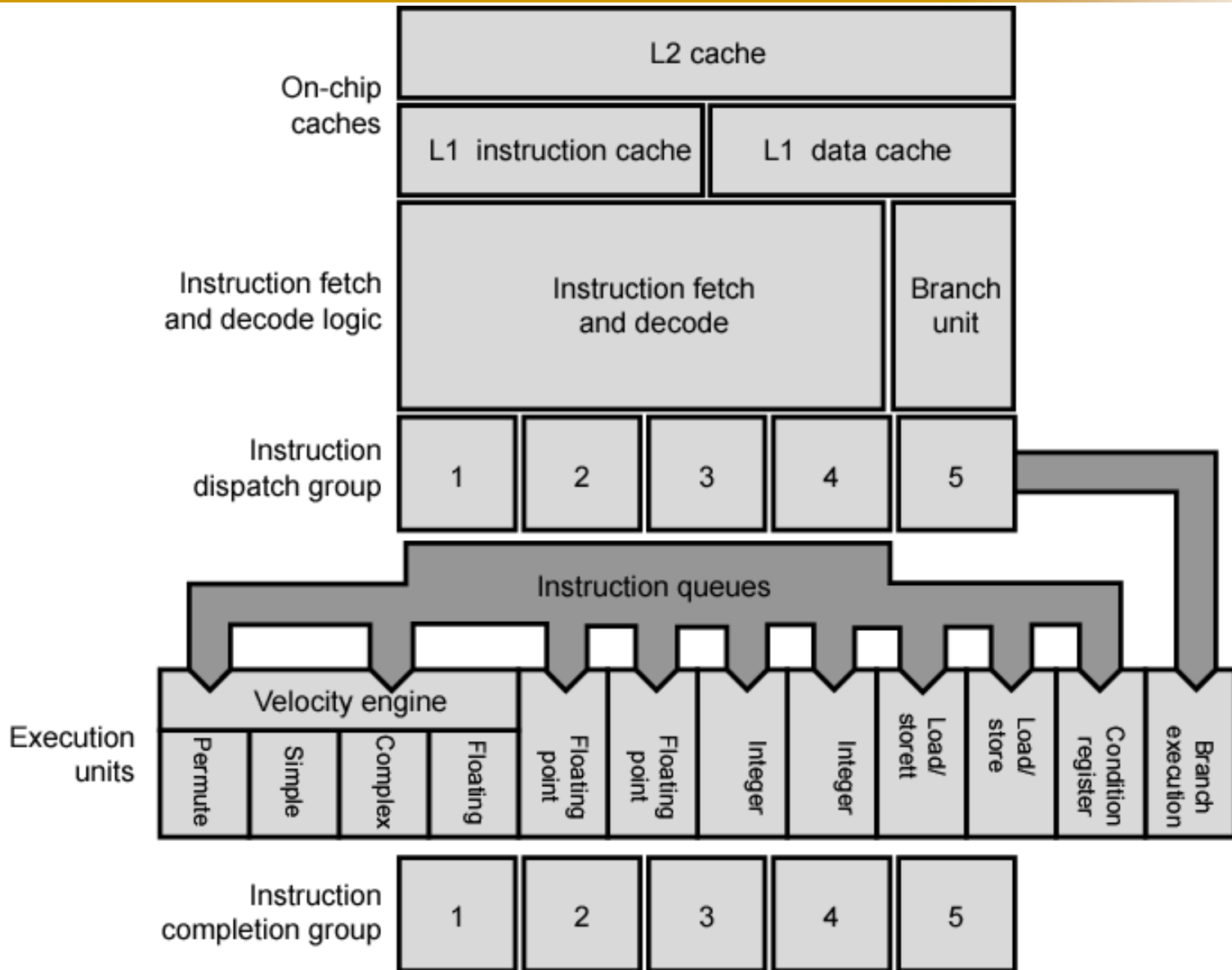
Pentium 4 Block Diagram



PowerPC Cache

- 601 – single 32kb 8 way set associative
- 603 – 16kb (2 x 8kb) two way set associative
- 604 – 32kb
- 620 – 64kb
- G3 & G4
 - 64kb L1 cache
 - 8 way set associative
 - 256k, 512k or 1M L2 cache
 - two way set associative
- G5
 - 32kB instruction cache
 - 64kB data cache

PowerPC G5 Block Diagram



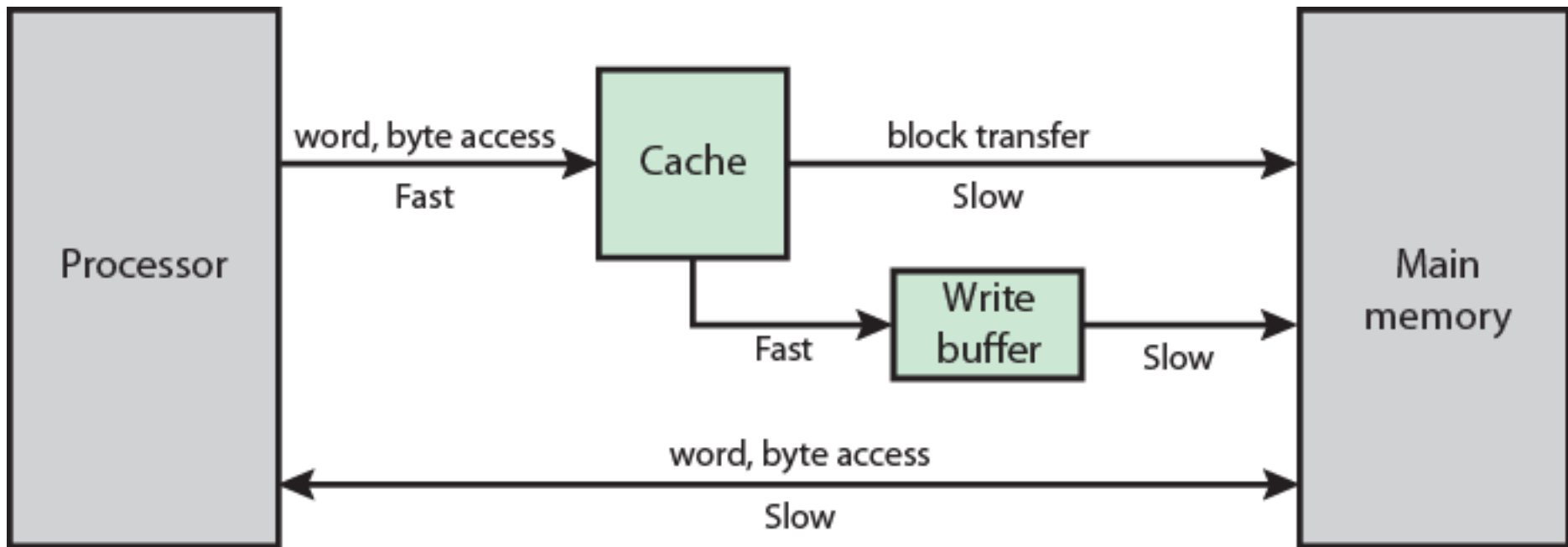
ARM Cache

Core	Cache Type	Cache Size (kB)	Cache Line Size (words)	Associativity	Location	Write Buffer Size (words)
ARM720T	Unified	8	4	4-way	Logical	8
ARM920T	Split	16/16 D/I	8	64-way	Logical	16
ARM926EJ-S	Split	4-128/4-128 D/I	8	4-way	Logical	16
ARM1022E	Split	16/16 D/I	8	64-way	Logical	16
ARM1026EJ-S	Split	4-128/4-128 D/I	8	4-way	Logical	8
Intel StrongARM	Split	16/16 D/I	4	32-way	Logical	32
Intel Xscale	Split	32/32 D/I	8	32-way	Logical	32
ARM1136-JF-S	Split	4-64/4-64 D/I	8	4-way	Physical	32

TỔ chức ARM Cache

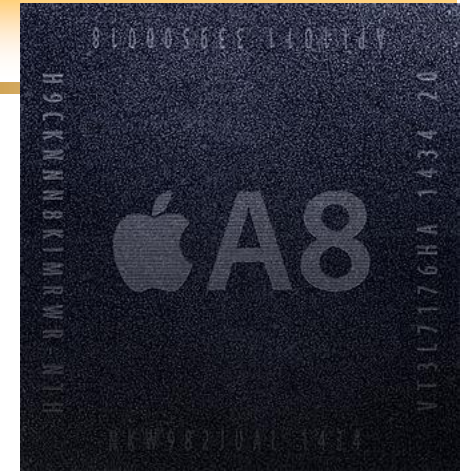
- Small FIFO write buffer
 - Enhances memory write performance
 - Between cache and main memory
 - Small c.f. cache
 - Data put in write buffer at processor clock speed
 - Processor continues execution
 - External write in parallel until empty
 - If buffer full, processor stalls
 - Data in write buffer not available until written
 - So keep buffer small

Tổ chức ARM Cache ...



Apple A8 CPU

- 2 cores
- Max. CPU clock: 1.38 GHz
- Min. feature size: 20 nm
- Instruction set: ARMv8-A
- L1 cache: Per core: 64 KB instruction + 64 KB data
- L2 cache: 1 MB shared
- L3 cache: 4 MB
- 1 GB of LPDDR3 RAM included in the package
- GPU: PowerVR Series 6XT GX6450 (quad core)
- 2 billion transistors, physical size reduced by 13% to 89 mm²
- Produced by Taiwan Semiconductor Manufacturing Company Limited (TSMC)



Apple A8X, 10/2014

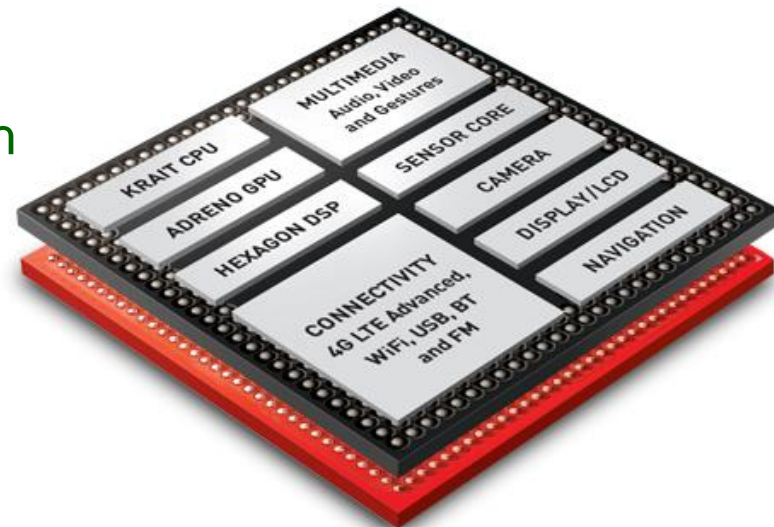
- Cores: 3
- Max. CPU clock rate: 1.5 GHz
- Min. feature size: 20 nm[1]
- Instruction set: A64, A32, T32
- Microarchitecture: Typhoon ARMv8-A-compatible
- L1 cache Per core: 64 KB instruction + 64 KB data
- L2 cache 2 MB shared
- L3 cache 4 MB[
- Predecessor Apple A7
- Successor Apple A9X
- GPU PowerVR Series 6XT GXA6850 (octa-core)



Qualcomm Snapdragon

■ Snapdragon 805

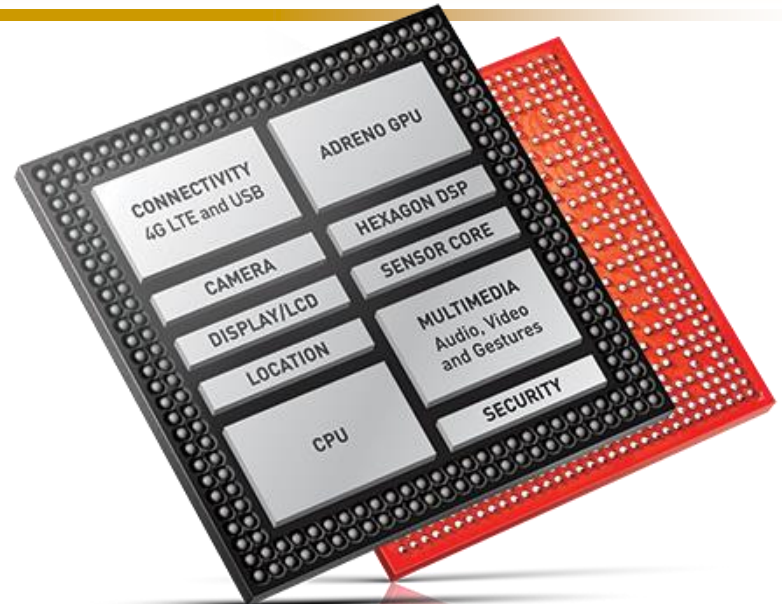
- ❑ ARMv7-A, Quad-core Krait 450 CPU at up to 2.7 GHz per core
- ❑ 16 KiB / 16 KiB L1 cache per core; 2 MiB L2 cache
- ❑ 4K UHD video upscale & play
- ❑ Dual camera image signal processor supporting up to 55 Megapixel, stereoscopic 3D
- ❑ Adreno 420 GPU
- ❑ LPDDR3 25.6 GB/s memory bandwidth
- ❑ IZat Gen8B GNSS location technology
- ❑ USB 2.0 and 3.0
- ❑ Hexagon, QDSP6V5A, 600 MHz
- ❑ e-MMC V5.0, UFS V2.0
- ❑ BT4.1, 802.11ac Wi-Fi
- ❑ 28 nm HPm (high performance mobile)
- ❑ Devices: Samsung S5, G G3, Samsung Note 4 , Note Edge



Qualcomm Snapdragon...

■ Snapdragon 810

- ❑ ARMv8-A, 4+4 cores, 2GHz?
- ❑ 16 KiB / 16 KiB L1 cache per core; 2 MiB L2 cache
- ❑ H.265/HEVC encoding/decoding
- ❑ eMMC 5.0 support
- ❑ 14-bit dual-ISP
- ❑ support for triple-band (i.e. IEEE 802.11, IEEE 802.15 (Bluetooth) and IEEE 802.11ad (60 GHz).
- ❑ Qualcomm acquired Wilocity
- ❑ Adreno 430 GPU
- ❑ LPDDR4 25.6 GB/s memory bandwidth
- ❑ BT4.1, 802.11ac Wi-Fi
- ❑ 20 nm



Exynos 7 Octa 7420, 2015

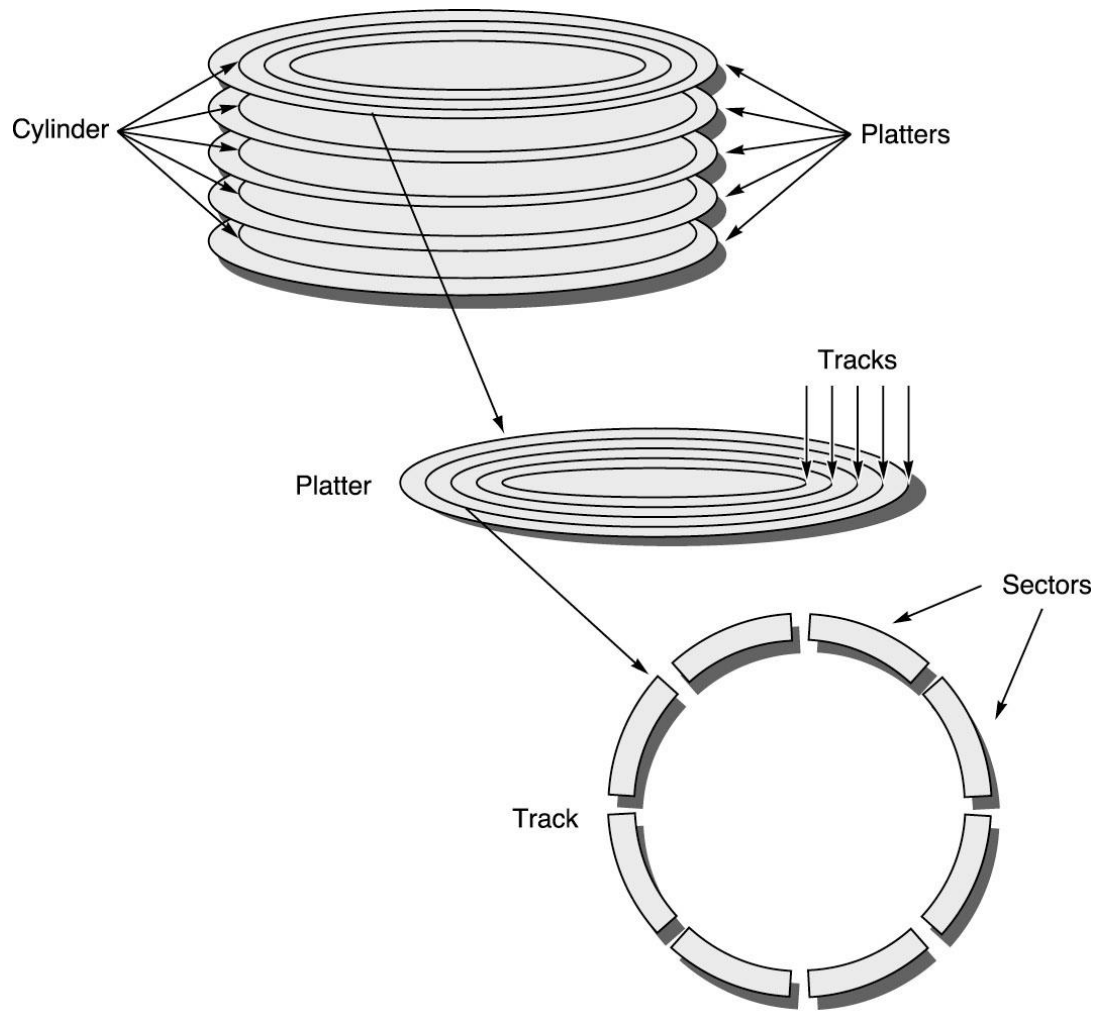
- Technology: 14 nm LPE
 - Instruction Set: ARMv8-A
 - Microarchitecture: Cortex-A57+
Cortex-A53 (big.LITTLE with GTS)
 - Cores: 4 (2.1GHz) + 4 (1.5GHz)
 - GPU: Mali-T760 MP8 @ 772 MHz
-
- Samsung Galaxy S6, Samsung Galaxy S6 Edge, Samsung Galaxy S6 Active, Samsung Galaxy S6 Edge+, Samsung Galaxy Note 5



4. Bộ nhớ ngoài

- RAM for storage
 - Flash
- Đĩa từ - Magnetic Disk
 - HDD
 - RAID
 - Removable
- Đĩa quang - Optical
 - CD-ROM
 - CD-Recordable (CD-R)
 - CD-R/W
 - DVD
- Băng từ - Magnetic Tape

Magnetic Disks



Magnetic Disks

- Ưu điểm: rẻ(\$/MB), độ tin cậy chấp nhận được
 - Primary storage, memory swapping
- Nhược: chỉ có thể đọc/ghi toàn bộ một sector
 - → không thể truy cập trực tiếp như bộ nhớ chính
- Thời gian truy cập đĩa
 - Queuing delay
 - Wait until disk gets to do this operation
 - Seek time
 - Head moves to correct track
 - Rotational latency
 - Correct sector must get under the head
 - Data transfer time and controller time

Khuyñh hướng của đĩa từ

- Dung lượng: ~gấp đôi sau mỗi năm
- Thời gian truy cập trung bình
 - 5-12ms (việc cải thiện tốc độ tương đối chậm)
- Latency quay trung bình (1/2 full rotation)
 - 5,000 RPM to 10,000 RPM to 15,000 RPM
 - Cải thiện chậm, phức tạp (reliability, noise)
- Tốc độ truyền dữ liệu
 - Phụ thuộc vào bề mặt đĩa, dữ liệu trên một track

Đĩa quang

- Giới hạn bởi các chuẩn
 - CD and DVD capacity fixed over years
 - Technology actually improves, but it takes time for it to make it into new standards
- Kích thước bé, dễ thay thế
 - Good for backups and carrying around

Băng từ

- Thời gian truy cập rất chậm
 - Must rewind tape to correct place for read/write
- Chi phí thấp (\$/MB)
 - It's just miles of tape!
 - But disks have caught up anyway...
- Được sử dụng để sao lưu dữ liệu (secondary storage)
 - Large capacity & Replaceable

RAM for Storage

- Disks are about 100 times cheaper (\$/MB)
- DRAM is about 100,000 faster (latency)
- Solid-State Disks
 - Actually, a DRAM and a battery
 - Much faster than disk, more reliable
 - Expensive (not very good for archives and such)
- Flash memory
 - Much faster than disks, but slower than DRAM
 - Very low power consumption
 - Can be sold in small sizes (few GB, but tiny)

Một số vấn đề lỗi

- Phân loại theo nguyên nhân
 - Hardware Faults
 - Hardware devices fail to perform as designed
 - Design Faults
 - Faults in software and some faults in HW
 - E.g. the Pentium FDIV bug was a design fault
 - Operation Faults
 - Operator and user mistakes
 - Environmental Faults
 - Fire, power failure, sabotage, etc.

Khắc phục: cải thiện độ tin cậy

- Fault Avoidance
 - Prevent occurrence of faults by construction
- Fault Tolerance
 - Prevent faults from becoming failures
 - Typically done through redundancy
- Error Removal
 - Removing latent errors by verification
- Error Forecasting
 - Estimate presence, creation, and consequences of errors

Disk Fault Tolerance with RAID

- Redundant Array of Independent/Inexpensive Disks
 - Several smaller disks play a role of one big disk
- Can improve performance
 - Data spread among multiple disks
 - Accesses to different disks go in parallel
- Can improve reliability
 - Data can be kept with some redundancy

RAID

Tập các đĩa vật lý thành một đĩa logic nhìn bởi hệ điều hành

- RAID 0: không dư thừa dữ liệu
 - Truy cập song song các đĩa
 - Dữ liệu có thể phân tán trên nhiều đĩa
- RAID 1:
 - Mirrored disks
 - 2 copies được lưu trên 2 đĩa khác nhau
 - Đọc từ 1 trong 2, ghi đồng thời lên 2
 - Đắt
- RAID 2:
 - Các đĩa được đồng bộ hoá
 - Dư thừa nhiều, giá thành cao

RAID

■ RAID 3

- Như RAID 2
- Chỉ sử dụng 1 đĩa dự thừa

■ RAID 4

- Mỗi đĩa hoạt động một cách độc lập
- Parity được lưu trong một đĩa parity riêng

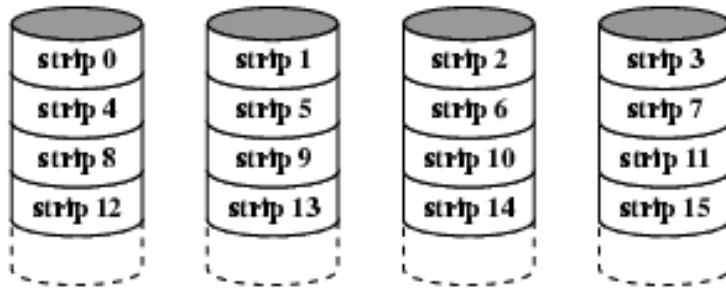
■ RAID 5

- Như RAID 4
- Parity được phân ra trên tất cả các đĩa

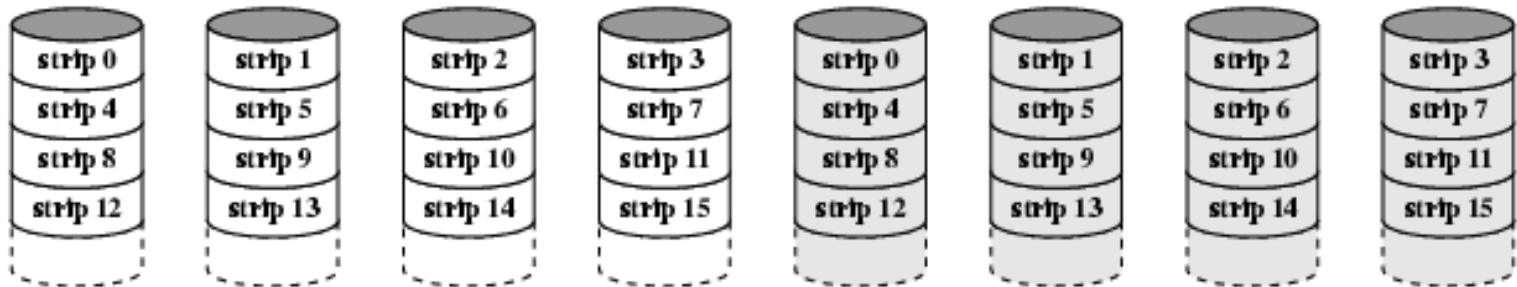
■ RAID 6

- Hai lần tính parity
- Lưu các block khác nhau trên các đĩa khác nhau

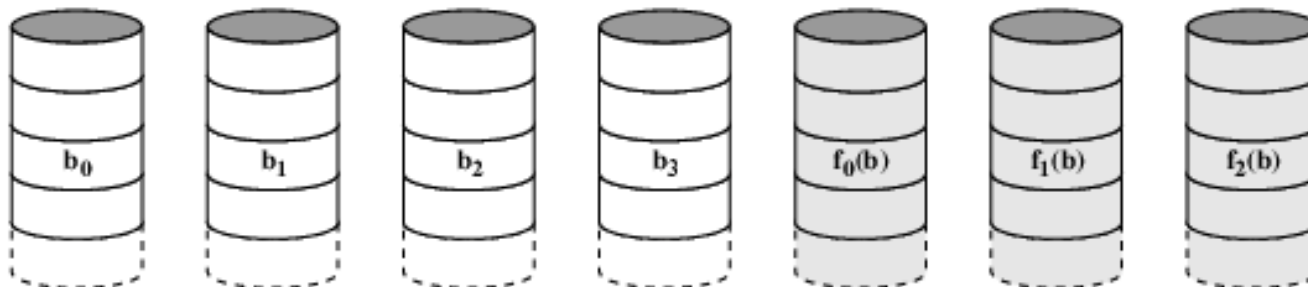
RAID 0, 1, 2



(a) RAID 0 (non-redundant)

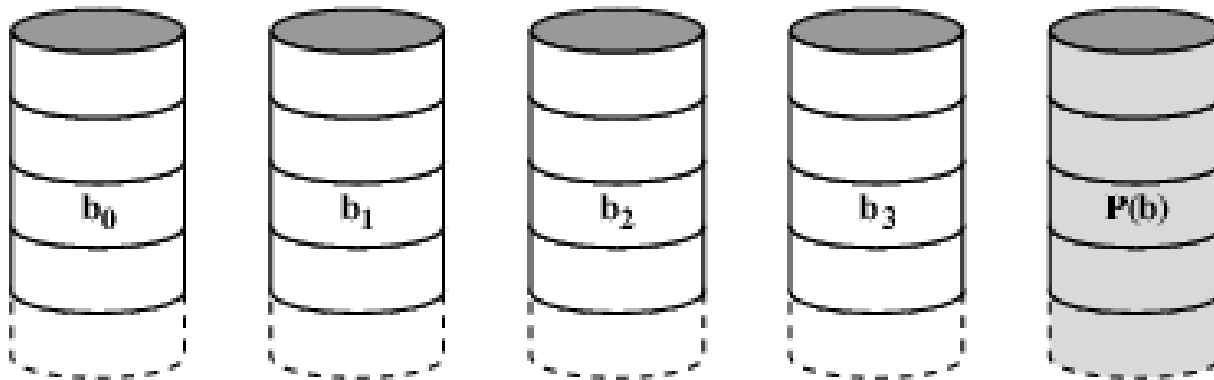


(b) RAID 1 (mirrored)

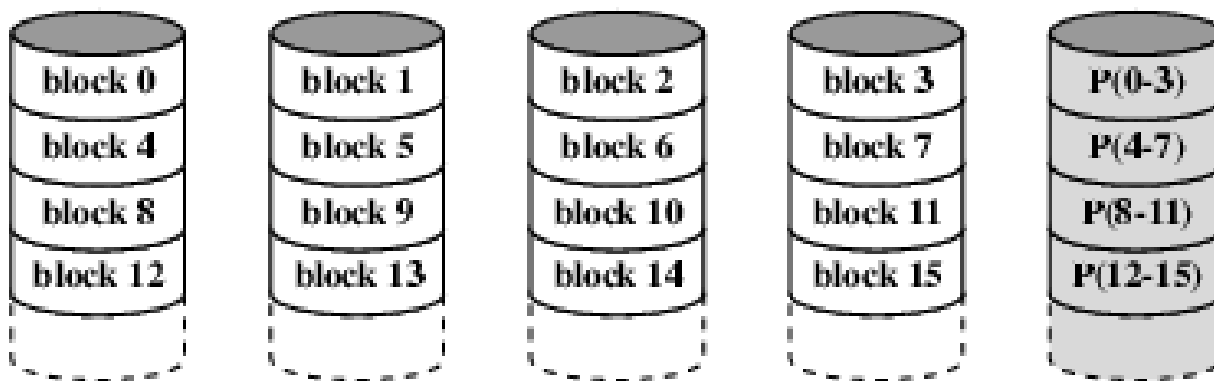


(c) RAID 2 (redundancy through Hamming code)

RAID 3 & 4

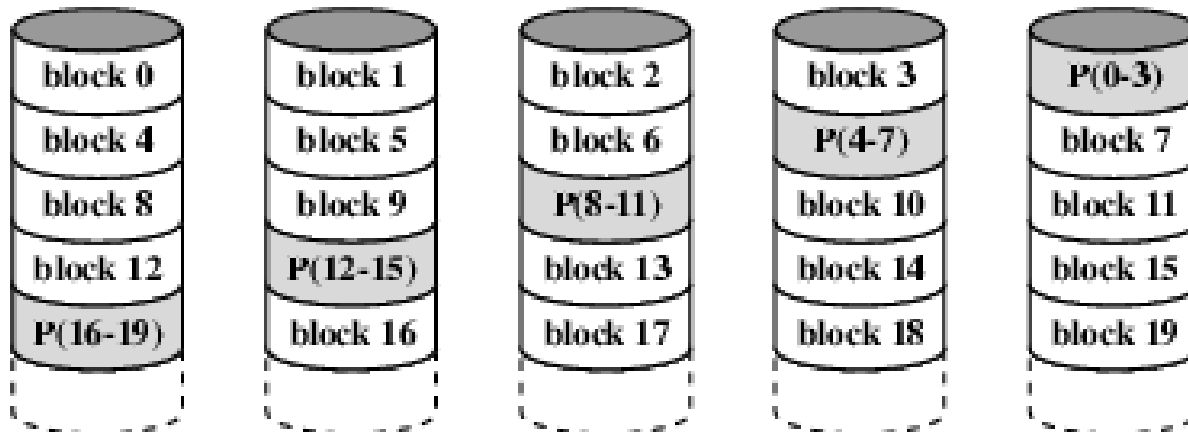


(d) RAID 3 (bit-interleaved parity)

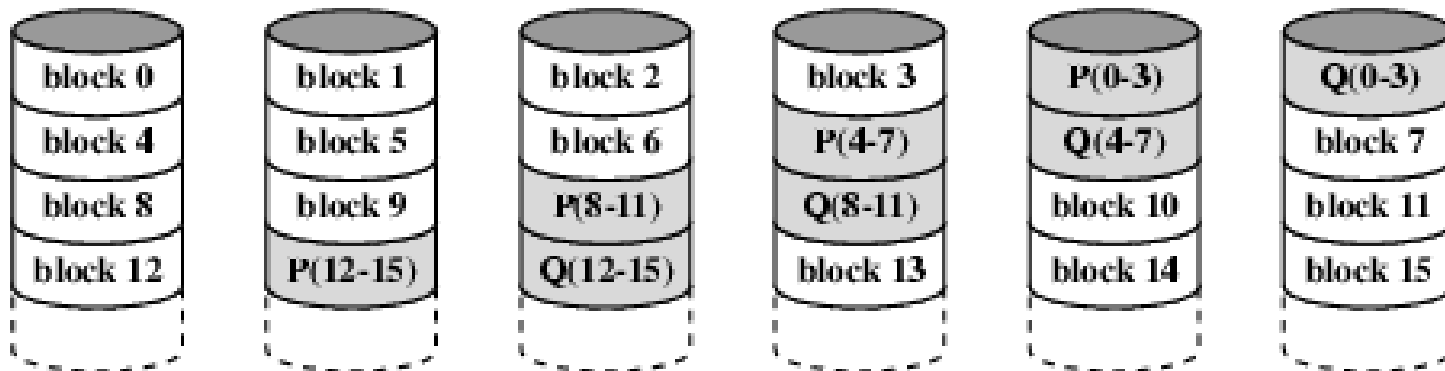


(e) RAID 4 (block-level parity)

RAID 5 & 6



(f) RAID 5 (block-level distributed parity)



(g) RAID 6 (dual redundancy)

Tổng kết

- Khái niệm về bộ nhớ trong máy tính, các đặc điểm chính, ...
- Bộ nhớ chính
 - Nguyên tắc
 - Phân loại
 - Tổ chức bộ nhớ lớn
- Bộ nhớ cache
 - Nguyên tắc chung
 - Các phương pháp ánh xạ
 - Các giải thuật thay thế, chính sách ghi, ...
- Bộ nhớ ngoài: ổ cứng, quang, RAID, ...