Chương 5:

BỘ NHỚ

Sự Phân Cấp Bộ Nhớ

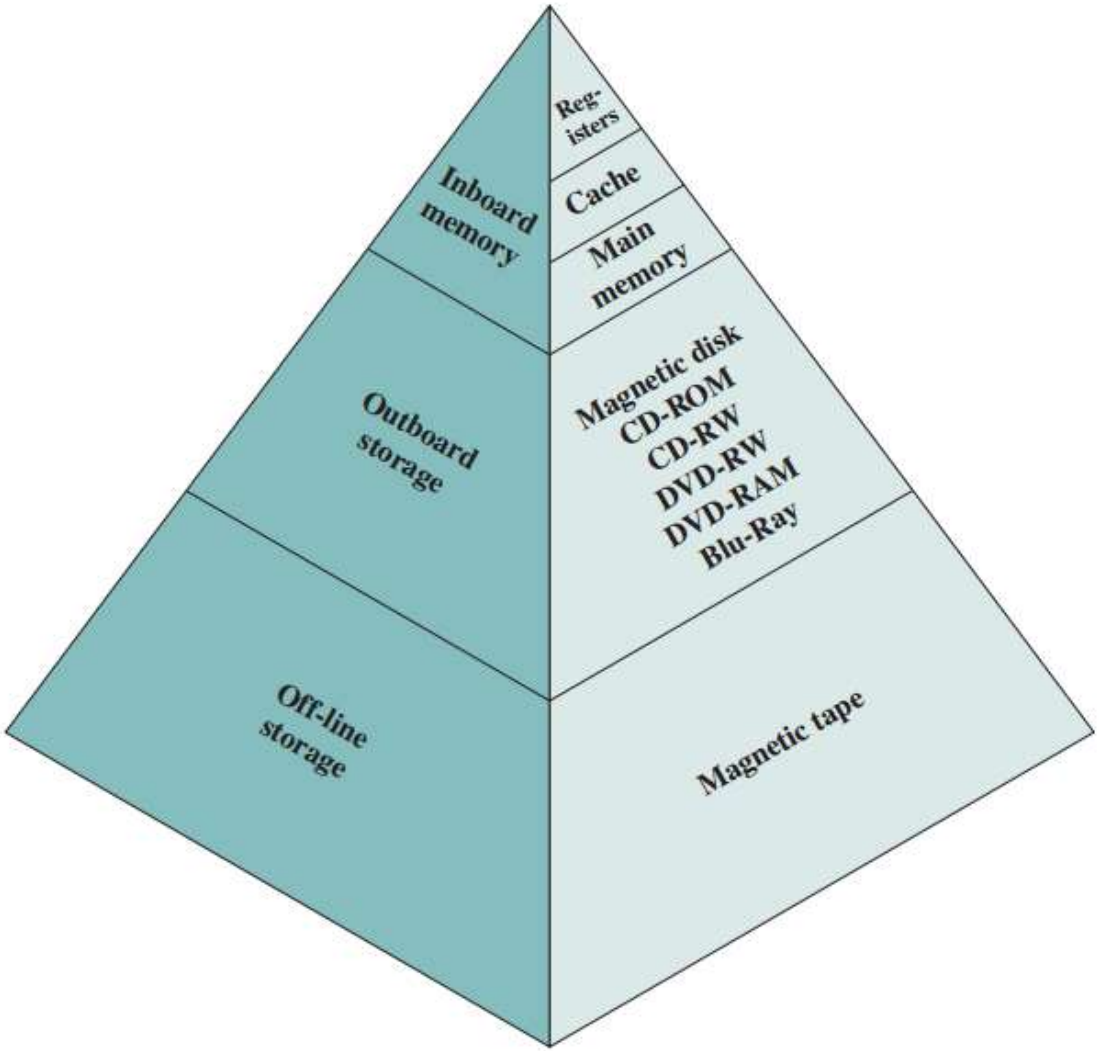
- Trong trường hợp lý tưởng, đó là sự cân bằng giữa ba đặc trưng chính của bộ nhớ: dung lượng, thời gian truy xuất và giá thành. Nhiều công nghệ được sử dụng để chế tạo bộ nhớ và chúng có các mối quan hệ như sau:
  - Thời gian truy xuất nhanh hơn, giá thành của mỗi bit nhớ lớn hơn.
  - Dung lượng lớn hơn, giá thành mỗi bit nhớ nhỏ hơn.
  - Dung lượng lớn hơn, thời gian truy xuất chậm hơn.
- Vấn đề khó khăn mà nhà thiết kế phải đối

Sự Phân Cấp Bộ Nhớ

mặt là rõ ràng. Nhà thiết kế muốn sử dụng các công nghệ bộ nhớ để có thể chế tạo bộ nhớ có dung lượng lớn hơn nhưng giá thành thấp hơn. Tuy nhiên, nếu cần phải đáp ứng yêu cầu về hiệu suất, phải chấp nhận giá thành cao, dung lượng nhỏ hơn nhưng thời gian truy xuất ngắn.

- Giải pháp để giải quyết vấn đề này thì không dựa vào một kiểu bộ nhớ hay công nghệ nào mà là áp dụng sự phân cấp bộ nhớ (memory hierarchy). Sự phân cấp tiêu biểu được minh hoạ trong Hình 4.1.

Sự Phân Cấp Bộ Nhớ



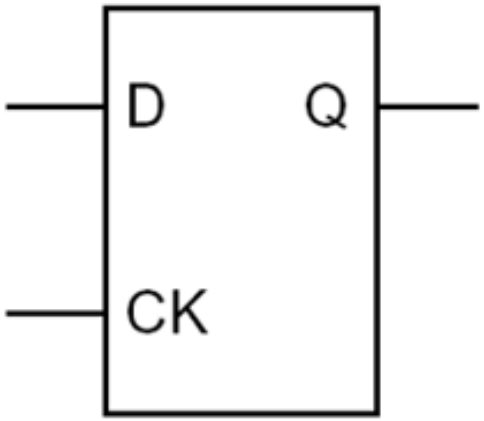
Hình 4.1: Sự phân cấp bộ nhớ.

Sự Phân Cấp Bộ Nhớ

- Xét Hình 4.1 từ trên xuống, chúng ta có nhận xét như sau:
  - Giảm giá thành mỗi bit nhớ.
  - Tăng dung lượng.
  - Tăng thời gian truy xuất.
  - Giảm tần suất truy xuất bộ nhớ của bộ xử lý.

Bộ Nhớ Trong – Bit Nhớ

- Bit nhớ được tạo từ Flip-flop D hoặc mạch cài/chốt D (D latch). Ký hiệu và bảng sự thật của bit nhớ như sau:



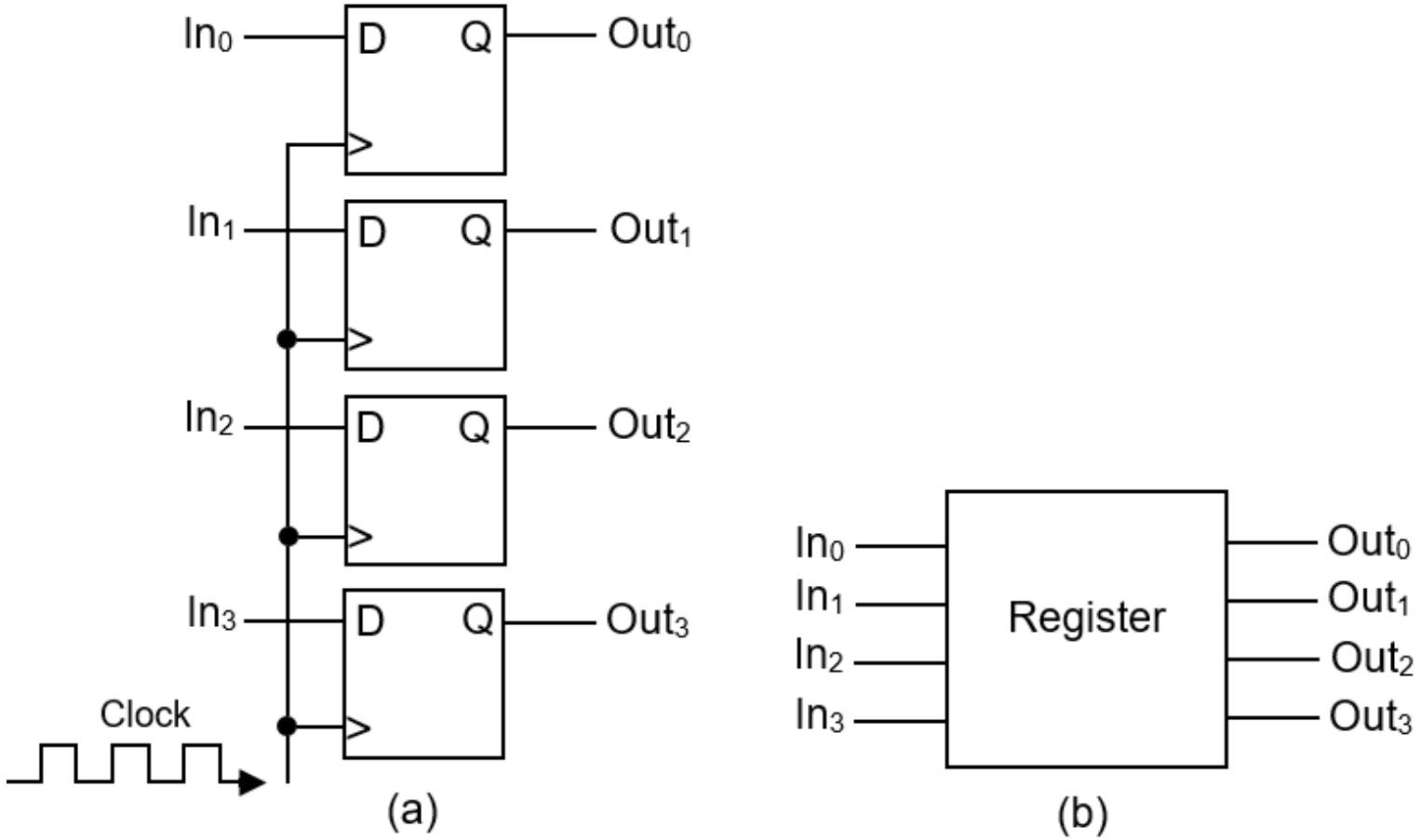
Input		Output
D	Clk	Q
0	↑	0
1	↑	1

Hình 4.2: Ký hiệu và bảng sự thật của bit nhớ.

Bit Nhớ

- Flip-flop D hay mạch chốt D là đơn vị cấu tạo bộ nhớ, chúng có khả năng nhớ 1 bit.
- Muốn mạch nhớ nhiều bit, chúng ta mắc nối tiếp nhiều flip-flop. Mạch như vậy còn gọi là mạch ghi dịch (*shift register*).
- Các bit đang lưu trữ có thể dịch chuyển sang phải hay sang trái khi có tín hiệu clock, nên mạch ghi dịch có nhiều ứng dụng quan trọng. Cụ thể như dùng làm thanh ghi (*register*) hoặc dịch (sang trái, sang phải, vòng quanh) các số nhị phân.
- Hình 4.3 cho thấy cấu trúc thanh ghi 4 bit (a) và sơ đồ khối của nó (b).

Bit Nhớ



Hình 4.3: (a) Thanh ghi 4 bit. (b) Sơ đồ khối.

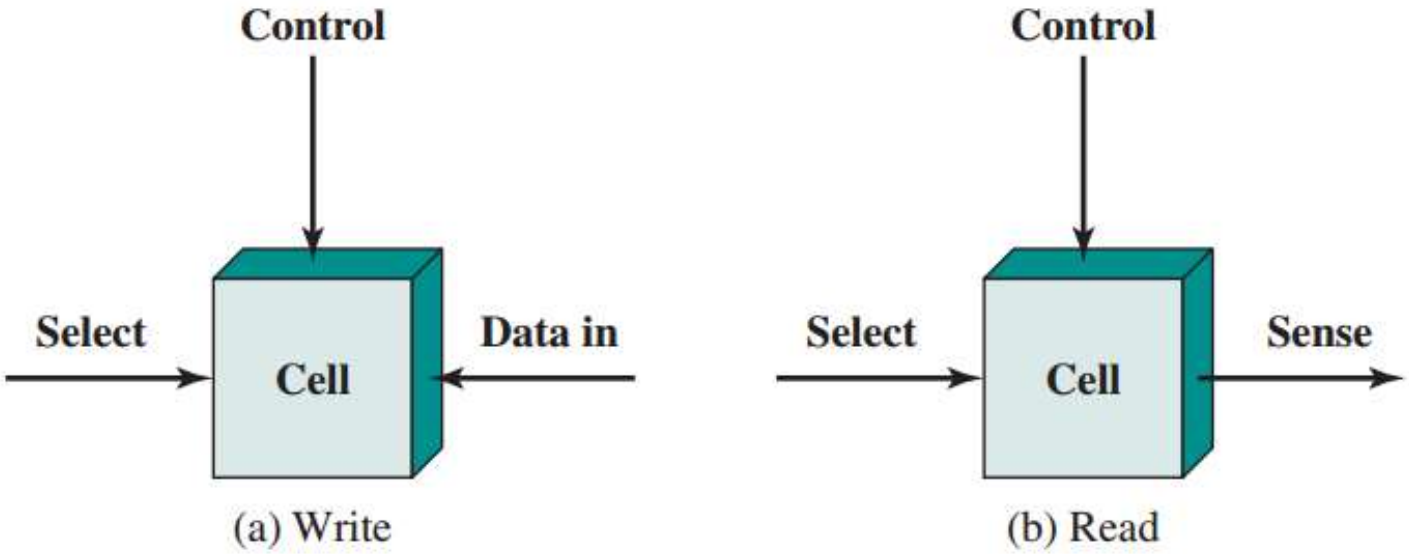
Tổ Chức Bộ Nhớ

- Thành phần cơ bản của bộ nhớ bán dẫn (semiconductor memory) là các ô nhớ (memory cell). Mỗi ô nhớ có một địa chỉ riêng biệt và là đơn vị hoạt động của bộ nhớ. Thông thường mỗi ô nhớ có dung lượng là 1 byte. Các byte được ghép thành từ (word). Một từ có thể dài 2, 4 hay 8 byte tuỳ vào kiến trúc máy tính. Mặc dù có nhiều công nghệ điện tử được sử dụng, nhưng hầu hết các ô nhớ bán dẫn có một số đặc điểm chung như sau:
  - Có hai trạng thái (được biểu diễn bởi giá trị nhị phân là 1 và 0).

Tổ Chức Bộ Nhớ

- Có khả năng được ghi vào (thiết lập trạng thái).
- Có khả năng được đọc ra (nhận biết trạng thái).

❖ Hình 4.4 mô tả hoạt động của một ô nhớ.

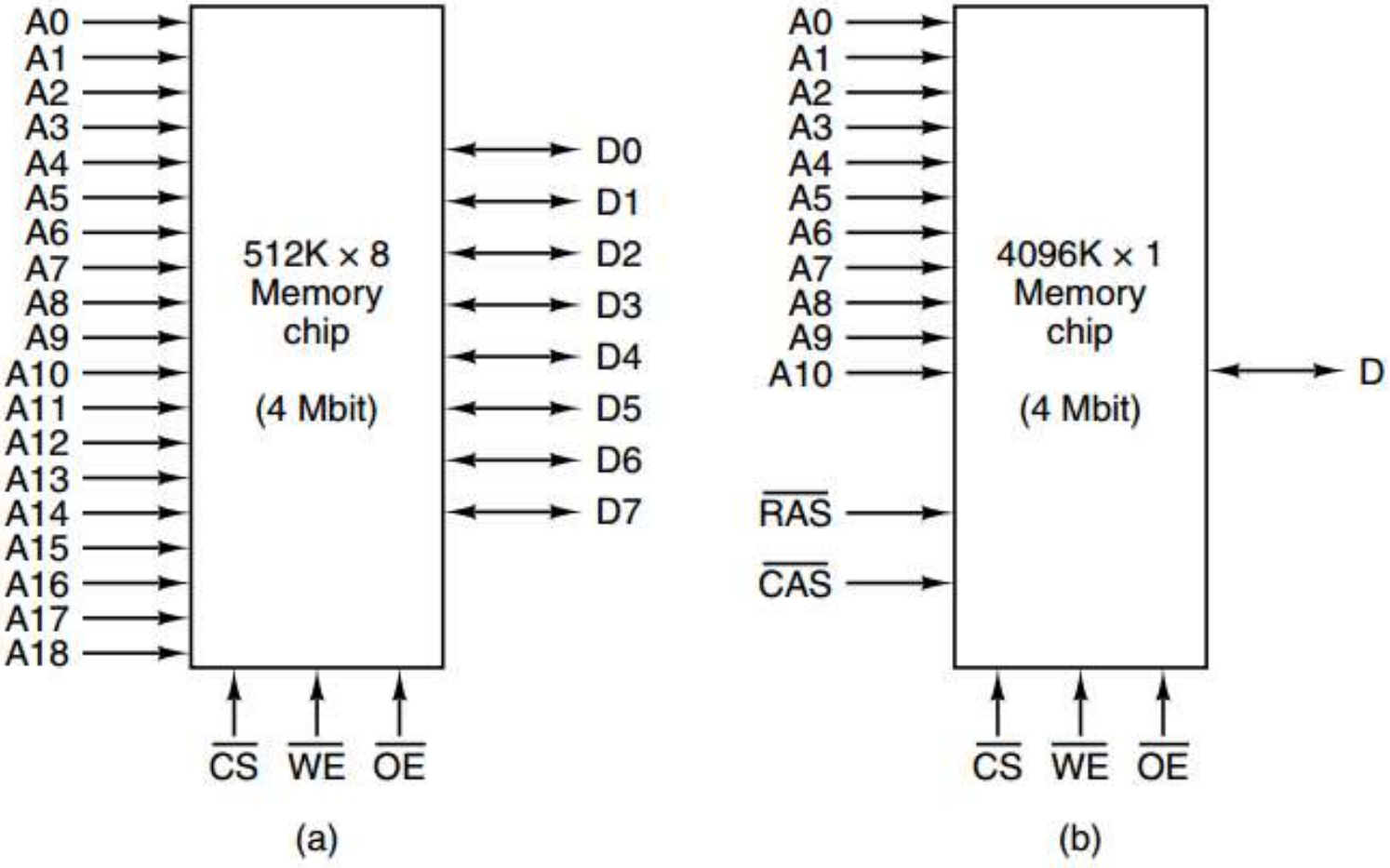


Hình 4.4: Hoạt động của ô nhớ.

Tổ Chức Bộ Nhớ

- Với kích thước bộ nhớ cho trước, có nhiều cách để tổ chức vi mạch (IC/chip) nhớ.
- Hình 4.5 cho thấy hai tổ chức có thể của vi mạch nhớ trước đây có kích thước là 4 Mbit: 512 K × 8 và 4096 K × 1 (kích thước vi mạch nhớ thường được biểu diễn bằng bit hơn là byte).

Tổ Chức Bộ Nhớ



Hình 4.5: Hai cách tổ chức chip nhớ 4 Mbit.

RAM

- Bộ nhớ mà chúng ta biết đến bây giờ có thể đọc và ghi. Bộ nhớ như thế gọi là RAM (*Random Access Memory* – Bộ nhớ truy xuất ngẫu nhiên).
- Thuật ngữ này có thể dùng nhầm bởi vì tất cả các chip nhớ đều có thể truy xuất ngẫu nhiên. Tuy nhiên, từ RAM đã được sử dụng quá quen thuộc. RAM được chia làm hai loại: RAM tĩnh và RAM động.
- RAM tĩnh (*Static RAM – SRAM*) thì được tạo bởi các mạch giống như flip-flop D. Loại bộ nhớ này có đặc tính là nội dung của nó duy trì miễn chừng nào còn nguồn điện.

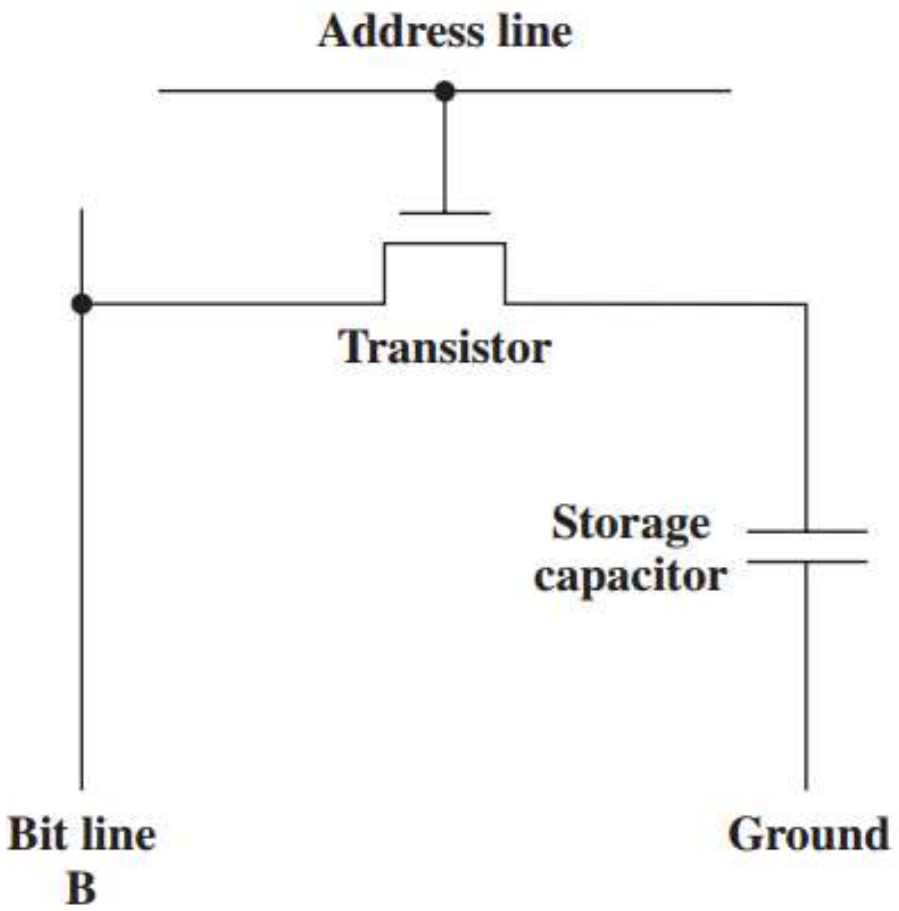
RAM

- RAM động (*Dynamic RAM - DRAM*) thì ngược lại, nó không dùng các flip-flip. Thay vào đó, RAM động là một dãy các ô, mỗi ô chứa một linh kiện bán dẫn (*transistor*) và một tụ điện (*capacitor*) nhỏ.
- Các tụ điện có thể được nạp và xả, cho phép các số 0 và 1 được lưu trữ. Bởi vì điện tích trong tụ bị rò rỉ, nên mỗi bit trong RAM động phải được làm tươi (*refresh*) tức nạp điện lại cho tụ trong khoảng thời gian vài mili giây để ngăn ngừa mất dữ liệu.
- Hình 4.6 là một cấu trúc tiêu biểu của RAM động cho một ô nhớ riêng biệt để lưu trữ 1 bit.

RAM

- Đường địa chỉ được kích hoạt khi giá trị của bit từ ô nhớ này được đọc hay được ghi.
- Transistor hành động giống như công tắc và nó đóng lại (cho phép dòng điện đi qua) nếu có điện áp trên đường địa chỉ và mở ra (không có dòng điện đi qua) nếu không có điện áp trên đường địa chỉ.

RAM



Hình 4.6: Ô nhớ của RAM động (DRAM).

RAM

- Có nhiều kiểu chip RAM động:
  - SDRAM (*Synchronous DRAM* – RAM động đồng bộ).
  - DDR (*Double Data Rate*) SDRAM.
  - RDRAM (*Rambus Dynamic* RAM).
- Bảng 4.1 sau cho thấy sự so sánh hiệu suất của một số loại RAM động.

RAM

	Clock Frequency (MHz)	Transfer Rate (GB/s)	Access Time (ns)	Pin Count
SDRAM	166	1.3	18	168
DDR	200	3.2	12.5	184
RDRAM	600	4.8	12	162

Bảng 4.1: So sánh hiệu suất một số RAM động.

- **Một số RAM động được phân loại theo hình dạng như sau:**
  - **SIMM (*Single Inline Memory Module* – Bộ nhớ một hàng chân). Xem Hình 4.7.**

RAM

SIMM

Hình 4.7: SIMM.

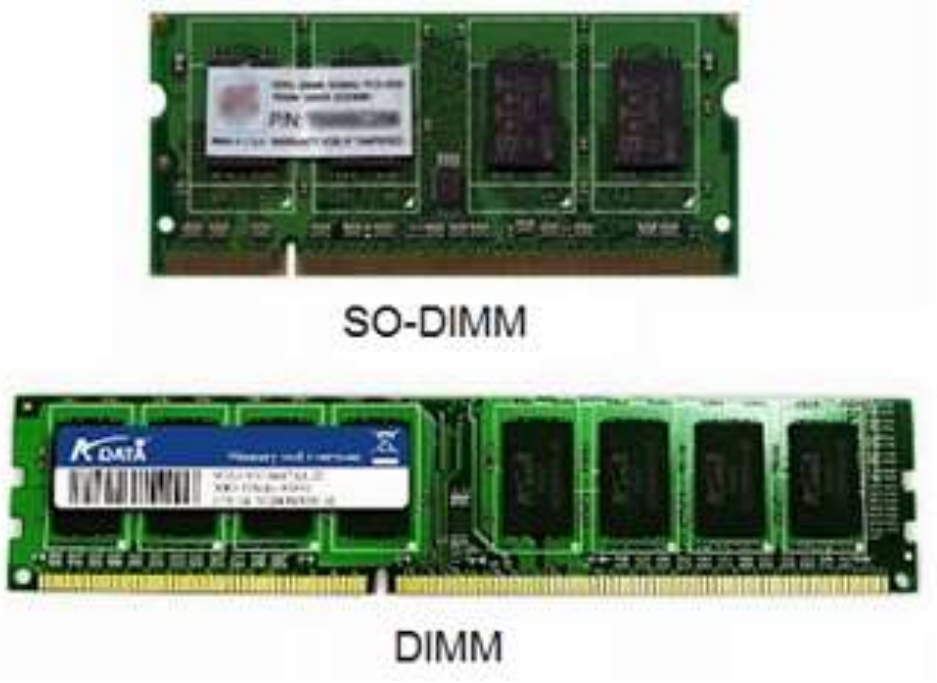
- **DIMM (*Dual Inline Memory Module* - Bộ nhớ hai hàng chân). Xem Hình 4.8.**

DIMM

Hình 4.8: DIMM.

RAM

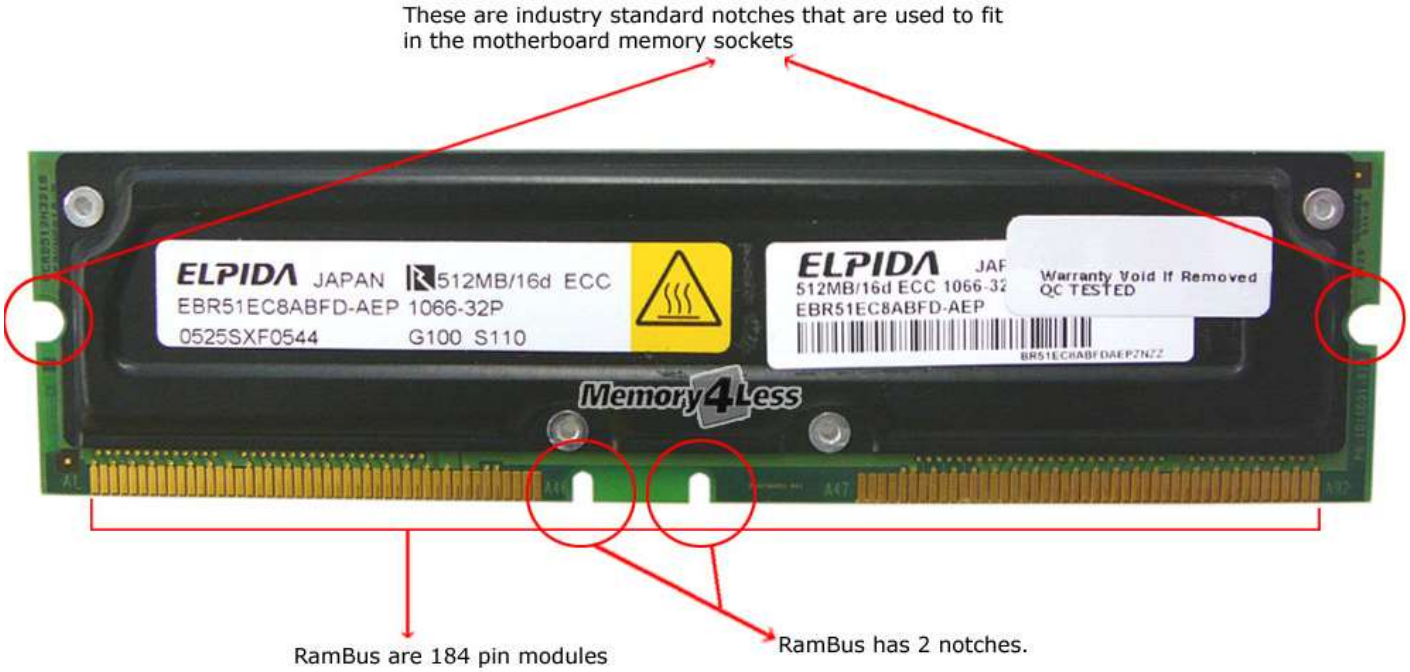
- **SO-DIMM (*Small Outline DIMM*). Xem Hình 4.9.**



Hình 4.9: SO-DIMM và DIMM.

RAM

- **RIMM (Rambus Inline Memory Module). Xem Hình 4.10.**



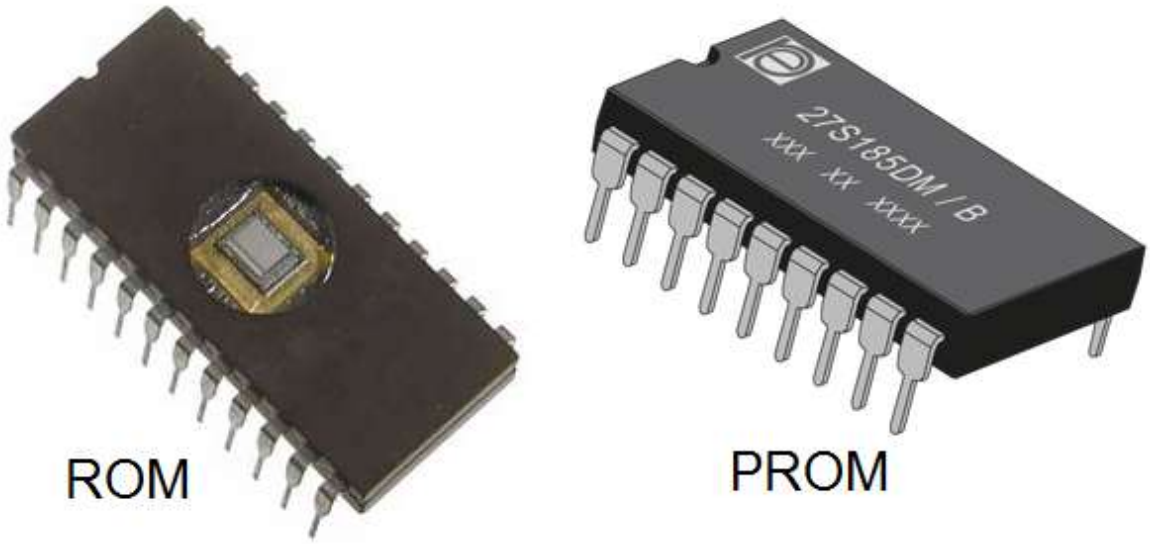
Hình 4.10: RIMM.

ROM

- ROM (*Read Only Memory* – Bộ nhớ chỉ đọc), là loại bộ nhớ không thể thay đổi hay xoá có chủ đích.
- Dữ liệu trong ROM được thêm vào trong lúc sản xuất ra nó, bằng cách là thông qua vật liệu cảm quang của mặt nạ để khắc lên trên bề mặt ROM các bit của chương trình.
- ROM thì rẻ hơn RAM nhiều. Tuy nhiên, nó không linh hoạt, bởi vì nó không thể thay đổi sau khi sản xuất.

ROM

- **PROM (*Programmable ROM* – ROM lập trình được): giống như ROM, ngoại trừ nó có thể được lập trình 1 lần bằng thiết bị chuyên dụng. Xem Hình 4.11**



Hình 4.11: ROM và PROM.

ROM

- **EPROM (*Erasable PROM* – ROM có thể lập trình và xoá), loại ROM này không chỉ lập trình được mà còn xoá được. Xem Hình 4.12.**

Hình 4.12: EPROM.

ROM

- **EEPROM (*Electrically EPROM* – ROM có thể lập trình và xoá bằng điện). EEPROM có thể xoá từng byte dữ liệu bằng các xung điện (bằng phần mềm) thay vì đặt nó vào thiết bị đặc biệt để xoá bằng tia cực tím.**
- **Ngoài ra, EEPROM có thể lập trình lại bằng xung điện trong khi EPROM phải đặt vào thiết bị lập trình EPROM đặc biệt. Xem Hình 4.13.**

ROM

Hình 4.13: EEPROM.

- **Một loại EEPROM gần đây là flash memory (bộ nhớ cực nhanh). Không giống như EPROM được xoá bằng tia cực tím và EEPROM có thể xoá từng byte, bộ nhớ flash có thể xoá và ghi lại theo khối.**

ROM

- Bảng 4.2 tóm tắt các loại bộ nhớ và Hình 4.14 là bộ nhớ flash.

Type	Category	Erasure	Byte alterable	Volatile	Typical use
SRAM	Read/write	Electrical	Yes	Yes	Level 2 cache
DRAM	Read/write	Electrical	Yes	Yes	Main memory (old)
SDRAM	Read/write	Electrical	Yes	Yes	Main memory (new)
ROM	Read-only	Not possible	No	No	Large-volume appliances
PROM	Read-only	Not possible	No	No	Small-volume equipment
EPROM	Read-mostly	UV light	No	No	Device prototyping
EEPROM	Read-mostly	Electrical	Yes	No	Device prototyping
Flash	Read/write	Electrical	No	No	Film for digital camera

Bảng 4.2: So sánh các kiểu bộ nhớ khác nhau.

ROM

Hình 4.14: Bộ nhớ flash.

Bộ Nhớ Ảo – Khái Niệm

- Những ngày trước đây, bộ nhớ máy tính nhỏ và đắt. IBM 650 dẫn đầu về khoa học máy tính ở những ngày này (sau thập niên 1950), bộ nhớ chỉ có 2000 từ.
- Một trong các trình biên dịch ngôn ngữ ALGOL 60 đầu tiên được viết chỉ chiếm 1024 từ bô nhớ.
- Người lập trình tốn nhiều thời gian để đưa các chương trình vào trong bô nhớ nhỏ này.
- Một giải pháp truyền thống là sử dụng bộ nhớ thứ cấp, như đĩa chẳng hạn. Người lập trình chia chương trình thành nhiều phần nhỏ gọi là các *overlay*, mỗi phần có dung lượng vừa với bộ nhớ.

Phân Trang

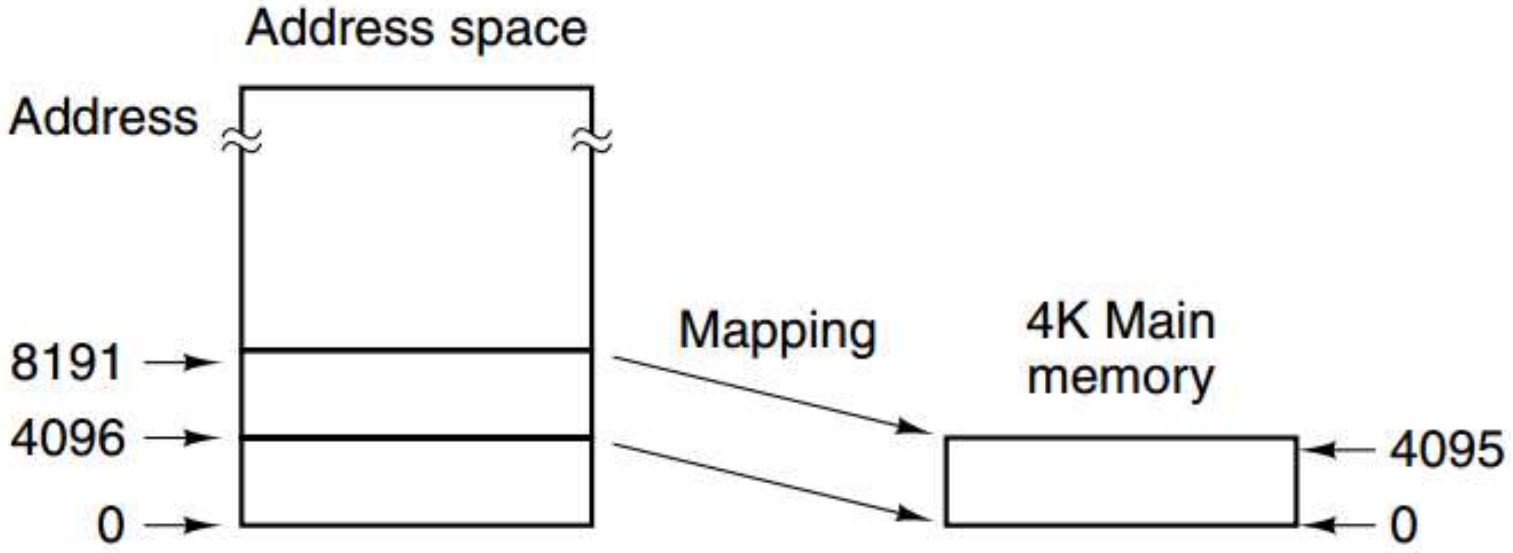
- Ý tưởng này được đưa ra bởi nhóm Manchester là những khái niệm của không gian địa chỉ (*address space*) và các vị trí nhớ (*memory location*).
- Xét một máy tính tiêu biểu của thời kỳ đó, nó có một vùng địa chỉ 16 bit trong các lệnh của nó và 4096 từ bộ nhớ.
- Một chương trình trên máy tính này có thể định địa chỉ 65536 từ nhớ. Mỗi địa chỉ tương ứng với một từ nhớ khác nhau.
- Không gian địa chỉ của máy tính này gồm các số 0, 1, 2, …, 65535, bởi vì đó là một tập từ nhớ có thể định địa chỉ.

Phân Trang

- Trước khi bộ nhớ ảo được phát minh, không có sự phân biệt giữa không gian địa chỉ và các vị trí nhớ, bởi vì phần cứng bắt buộc chỉ có tương ứng một-một giữa chúng.
- Ý tưởng của sự phân biệt không gian địa chỉ và các vị trí nhớ như sau: ở một thời điểm bất kỳ, 4096 từ nhớ có thể được truy xuất trực tiếp (từ địa chỉ 0 đến 4095).
- Chúng ta có thể cho máy tính "biết" là từ đây về sau bất kỳ khi nào địa chỉ 4096 được truy xuất, từ nhớ tại địa chỉ 0 được sử dụng. Bất kỳ khi nào địa chỉ 4097 được truy xuất, từ nhớ tại địa chỉ 1 được sử dụng, …

Phân Trang

- Nói cách khác, chúng ta đã định nghĩa một ánh xạ không gian địa chỉ lên các vị trí nhớ thực sự, như Hình 4.15.



Hình 4.14: Một ánh xạ địa chỉ ảo từ 4096 đến 8191 lên các địa chỉ của bộ nhớ chính từ 0 đến 4095.

Phân Trang

- Một câu hỏi đặt ra là: “Điều gì xảy ra nếu một chương trình rẽ nhánh đến địa chỉ giữa 8192 và 12287?”.
- Trên máy tính không có bộ nhớ ảo, chương trình đó làm cho lỗi phát sinh và một thông báo được in ra, chẳng hạn “*Nonexistent memory referenced*” và chương trình kết thúc.
- Trên máy tính có bộ nhớ ảo, các bước sau đây sẽ xảy ra:
  1. Nội dung của bộ nhớ chính sẽ được lưu lên đĩa.

Phân Trang

2. Các từ có địa chỉ từ 8192 đến 12287 sẽ được xác định vị trí trên đĩa.
3. Các từ có địa chỉ từ 8192 đến 12287 sẽ được nạp và bộ nhớ chính.
4. Sự ánh xạ địa chỉ sẽ được thay đổi để ánh xạ các địa chỉ từ 8192 đến 12287 lên bộ nhớ từ vị trí 0 đến 4095.
5. Sự thực thi sẽ tiếp tục như không có điều gì bất thường xảy ra.

- Kỹ thuật này được gọi là phân trang (*paging*) và các đoạn chương trình đọc từ đĩa được gọi là các trang (*page*).

Phân Trang

- Để tránh lẫn lộn, chúng ta sẽ gọi các địa chỉ mà chương trình có thể tham chiếu là không gian địa chỉ ảo (*virtual address space*) và các vị trí nhớ vật lý là không gian địa chỉ vật lý (*physical address space*).
- Bảng trang (*page table*) sẽ chỉ định mỗi địa chỉ ảo tương ứng với địa chỉ vật lý cụ thể. Chúng ta giả định rằng dung lượng đĩa chứa đủ toàn bộ không gian địa chỉ ảo.

Thực Hiện Phân Trang

- Không gian địa chỉ ảo được chia nhỏ thành nhiều trang bằng nhau. Kích thước trang từ 512 đến 64 KB. Kích thước trang luôn là luỹ thừa của 2, ví dụ 2^k, để tất cả các địa chỉ có thể được xác định bằng k bit.
- Không gian địa chỉ vật lý được chia thành các phần theo cách tương tự, mỗi phần có cùng kích thước như một trang. Những phần đó được gọi là các khung trang (page frame).
- Trong Hình 4.14 bộ nhớ chính chỉ chứa một khung trang. Trong thực tế thiết kế nó sẽ thường chứa hàng ngàn trang.

Thực Hiện Phân Trang

- **Hình 4.15 (a) minh họa cách có thể để chia 64 KB đầu tiên của không gian địa chỉ ảo thành các trang 4 KB.**

Thực Hiện Phân Trang

Page	Virtual addresses
≈	≈
15	61440 – 65535
14	57344 – 61439
13	53248 – 57343
12	49152 – 53247
11	45056 – 49151
10	40960 – 45055
9	36864 – 40959
8	32768 – 36863
7	28672 – 32767
6	24576 – 28671
5	20480 – 24575
4	16384 – 20479
3	12288 – 16383
2	8192 – 12287
1	4096 – 8191
0	0 – 4095

(a)

Hình 4.15: (a) 64 KB đầu tiên của không gian địa chỉ ảo được chia thành 16 trang, mỗi trang 4 K.

(b) Bộ nhớ chính 32 KB được chia thành 8 khung trang, mỗi khung 4 KB.

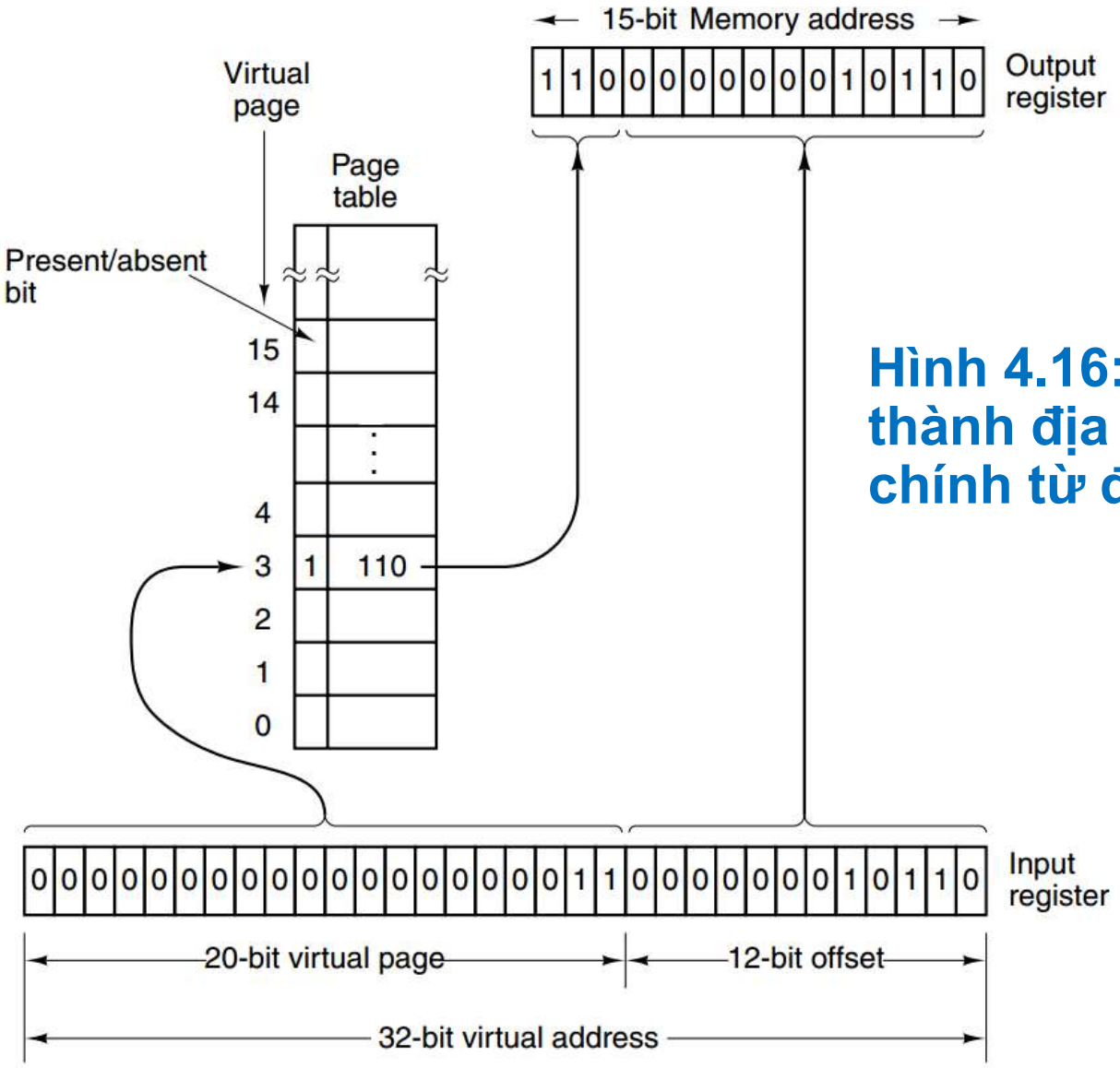
Page frame	Bottom 32K of main memory Physical addresses
7	28672 – 32767
6	24576 – 28671
5	20480 – 24575
4	16384 – 20479
3	12288 – 16383
2	8192 – 12287
1	4096 – 8191
0	0 – 4095

(b)

Thực Hiện Phân Trang

- Bây giờ hãy xem xét cách một địa chỉ ảo 32 bit có thể được ánh xạ trên địa chỉ vật lý của bộ nhớ chính.
- Mỗi máy tính với bộ nhớ ảo có một thiết bị để thực hiện ánh xạ ảo-vật lý. Thiết bị này được gọi là MMU (*Memory Management Unit* - Đơn vị quản lý bộ nhớ).
- Để xem cách MMU hoạt động, hãy xét ví dụ của Hình 4.16.

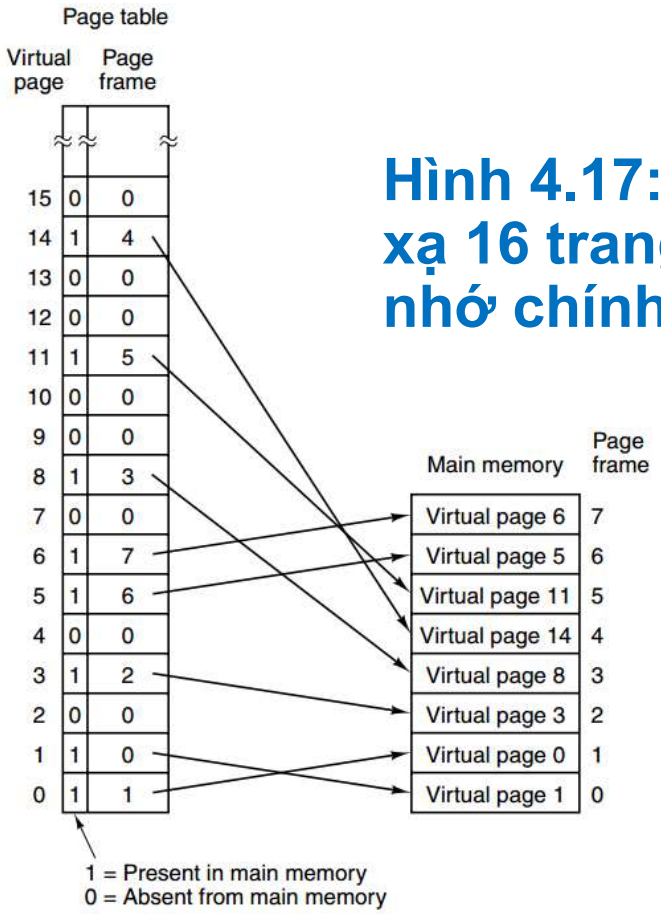
Thực Hiện Phân Trang



Hình 4.16: Sự tạo thành địa chỉ bộ nhớ chính từ địa chỉ ảo.

Thực Hiện Phân Trang

- Hình 4.17 cho thấy một khả năng ánh xạ giữa các trang ảo và các khung trang vật lý.



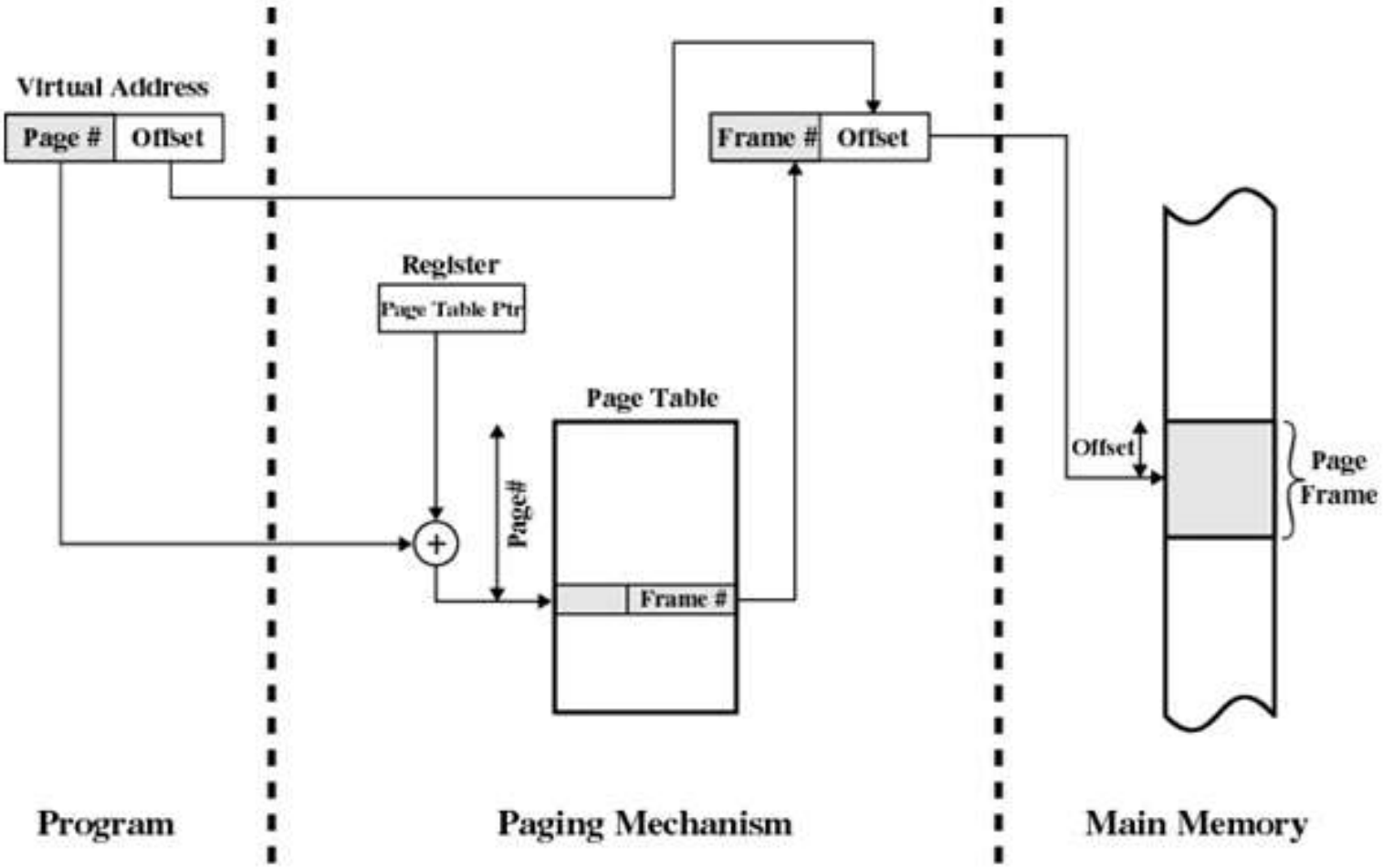
Hình 4.17: Một trường hợp ánh xạ 16 trang ảo đầu tiên vào bộ nhớ chính có 8 khung trang.

Chuyển Địa Chỉ Ảo Thành Vật lý

- **Chuyển đổi địa chỉ ảo thành địa chỉ vật lý:**
  1. **Địa chỉ ảo:**
     <Page number> <Offset>
     - **Page number: số thứ tự trang.**
     - **Offset: độ dời.**
  2. **Địa chỉ vật lý:**
     <Page frame number> <Offset>
     - **Page frame number: số thứ tự khung trang.**
     - **Offset: độ dời.**

- **Hình 4.18 minh hoạ cơ chế chuyển đổi địa chỉ ảo thành địa chỉ vật lý.**

Chuyển Địa Chỉ Ảo Thành Vật lý



Hình 4.18: Chuyển đổi địa chỉ ảo thành địa chỉ vật lý.

Truy Xuất Dữ Liệu Theo Địa Chỉ Ảo

- Tóm lại, cách truy xuất dữ liệu theo địa chỉ ảo như sau:
  1. Tách địa chỉ ảo thành số thứ tự trang và offset (độ dời).
  2. Chuyển số thứ tự trang thành số thứ tự khung trang bằng cách truy xuất bảng trang.
  3. Kiểm tra bit hiện diện/vắng mặt:
     a) Nếu bit này bằng 1 thì:
        - Thay số thứ tự trang thành số thứ tự khung trang.
        - Truy xuất dữ liệu trên khung trang với độ dời.

Truy Xuất Dữ Liệu Theo Địa Chỉ Ảo

b) Nếu bit này bằng 0 thì:

- Tìm trang trên đĩa.
- Tìm một khung trống (có thể phải thay thế trang nếu các khung đầy).
- Sao chép trang vào khung trống.
- Cập nhật bảng trang (bit hiện diện/vắng mặt = 1, số thứ tự khung trang mới).
- Thực hiện truy xuất như bước (a).

Chính Sách Thay Thế Trang

- Khi một chương trình tham chiếu một trang không có trong bộ nhớ chính, trang cần thiết phải được tìm nạp từ đĩa.
- Tuy nhiên, để có chỗ cho nó, một số trang khác thường sẽ phải được ghi trở lại cho đĩa. Như vậy một thuật toán quyết định trang nào cần di chuyển là cần thiết.
- Chọn một trang để loại bỏ một cách ngẫu nhiên có lẽ không phải là một ý tưởng tốt.
- Phần lớn, các hệ điều hành đều cố gắng để dự đoán những trang nào trong bộ nhớ là ít hữu dụng nhất theo nghĩa là sự vắng mặt của nó sẽ có tác động bất lợi nhỏ nhất đối với sự thực thi chương trình.

Chính Sách Thay Thế Trang

- **Một thuật toán phổ biến loại bỏ trang ít nhất được sử dụng gần đây nhất (*Least Recently Used* - LRU). Một thuật toán thay thế trang khác là FIFO (*First-In First-Out*). FIFO loại bỏ trang được nạp ít nhất trong thời gian gần đây.**

Bộ Nhớ Ảo Dạng Phân Đoạn

- Bộ nhớ ảo được thảo luận ở trên là một chiều bởi vì địa chỉ ảo đi từ 0 đến địa chỉ tối đa nào đó, địa chỉ này sau địa chỉ khác. Trong nhiều vấn đề, có hai hoặc nhiều không gian địa chỉ riêng biệt có thể tốt hơn nhiều khi chỉ có một. Cho ví dụ, trình biên dịch có thể có nhiều bảng được tạo ra trong quá trình biên dịch, bao gồm:
  - Bảng biểu tượng, chứa tên và các thuộc tính của các biến.
  - Mã nguồn của chương trình.
  - Một bảng có chứa tất cả các hằng số nguyên và các hằng số dấu chấm động.

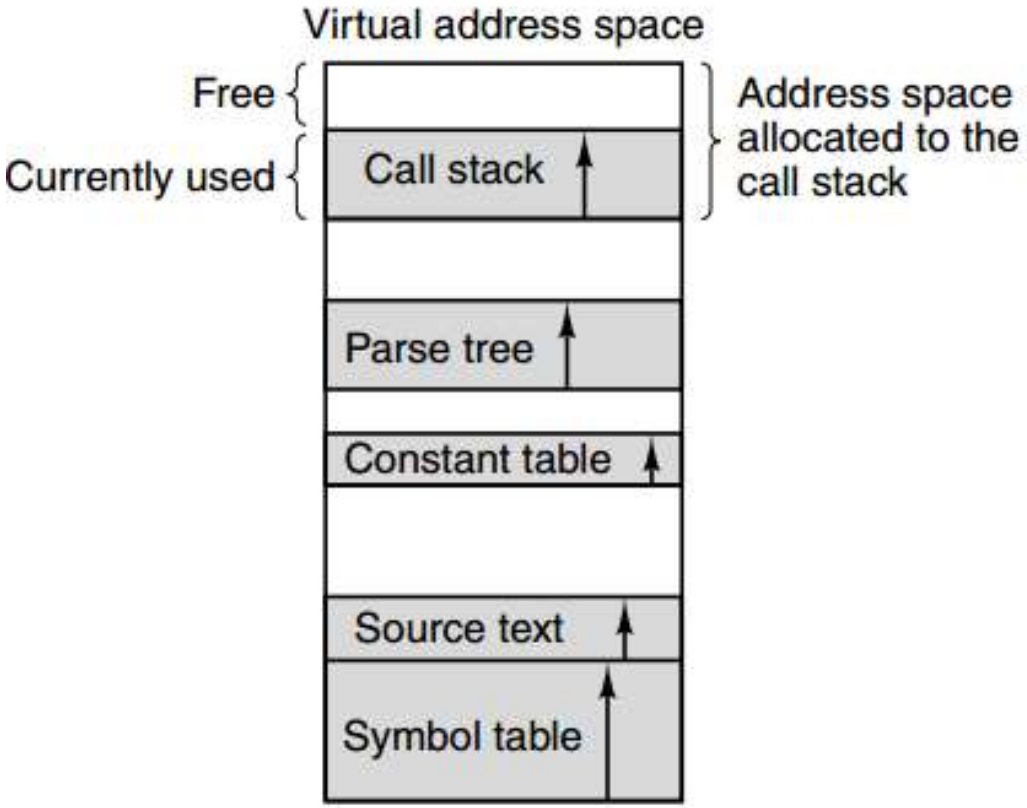
Bộ Nhớ Ảo Dạng Phân Đoạn

- Cây phân tích, chứa kết quả phân tích cú pháp của chương trình.
- Các stack được sử dụng cho các lời gọi thủ tục trong trình biên dịch.

❖ Mỗi bảng trong bốn bảng đầu tiên tăng trưởng liên tục trong quá trình biên dịch. Bảng cuối cùng tăng trưởng và co lại theo những cách không thể đoán trước trong lúc biên dịch.

❖ Trong bộ nhớ một chiều, năm bảng này sẽ phải được phân bổ như các khối liền kề của không gian địa chỉ ảo, như trong Hình 4.19.

Bộ Nhớ Ảo Dạng Phân Đoạn



Hình 4.19: Trong không gian địa chỉ 1 chiều với các bảng tăng trưởng, một bảng có thể chạm vào bảng khác.

Bộ Nhớ Ảo Dạng Phân Đoạn

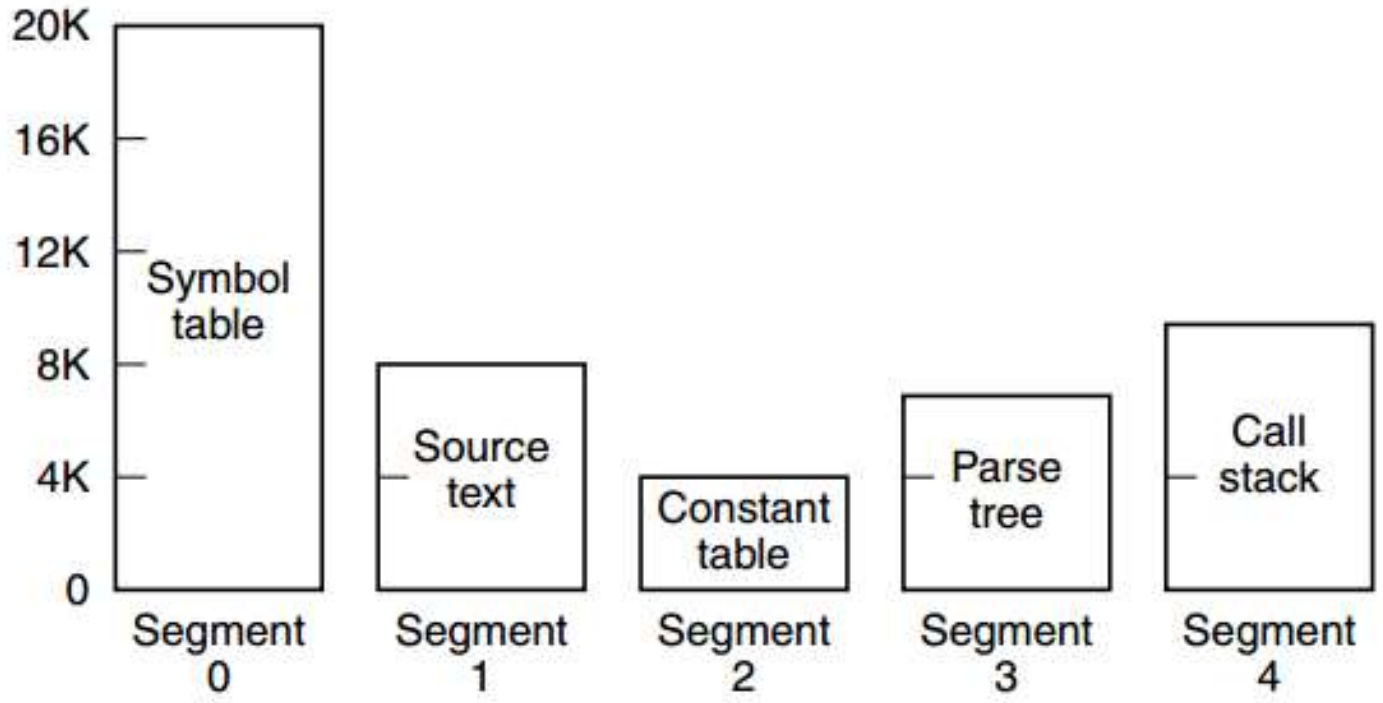
- Hãy xem điều gì sẽ xảy ra nếu chương trình có số lượng các biến lớn một cách khác thường. Đoạn không gian địa chỉ được phân bổ cho bảng biểu tượng (*symbol*) có thể chạm vào bảng mã nguồn.
- Một giải pháp đơn giản là cung cấp nhiều không gian địa chỉ hoàn toàn độc lập, được gọi là các đoạn (*segment*). Mỗi đoạn bao gồm một chuỗi các địa chỉ tuyến tính, từ 0 đến tối đa. Chiều dài của mỗi đoạn có thể là từ 0 đến tối đa cho phép. Các đoạn khác nhau có thể có chiều dài khác nhau.
- Bởi vì mỗi đoạn tạo thành không gian địa

Bộ Nhớ Ảo Dạng Phân Đoạn

chỉ riêng biệt, các đoạn khác nhau có thể tăng lên hoặc co lại một cách độc lập mà không ảnh hưởng đến nhau.

- Để chỉ định một địa chỉ trong bộ nhớ phân đoạn hoặc bộ nhớ hai chiều này, chương trình phải cung cấp một địa chỉ gồm hai phần: số thứ tự đoạn và địa chỉ bên trong đoạn.
- Hình 4.20 minh họa một bộ nhớ phân đoạn đang được sử dụng cho các bảng trình biên dịch đã thảo luận ở trên.

Bộ Nhớ Ảo Dạng Phân Đoạn

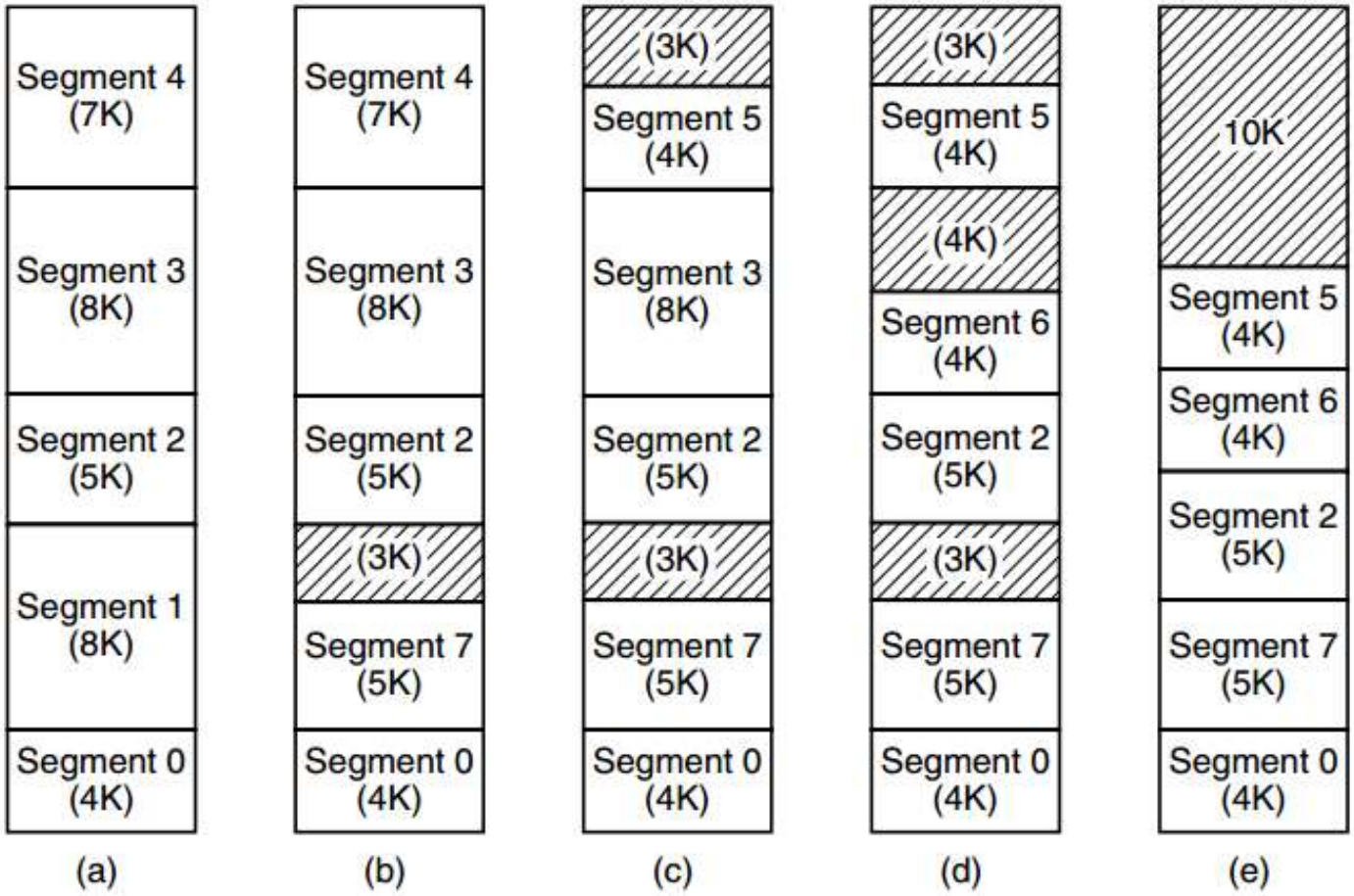


Hình 4.20: Bộ nhớ phân đoạn cho phép mỗi bảng có thể tăng lên hay co lại một cách độc lập.

Thực Hiện Phân Đoạn

- Tuy nhiên, việc thực hiện phân đoạn khác với phân trang trong một cách rất cốt lõi: các trang có kích thước cố định và các đoạn thì không. Xem Hình 4.21.

Thực Hiện Phân Đoạn



Hình 4.21: (a) – (d) Sự phát triển phân mảnh ngoài. (e) Xoá phân mảnh ngoài bằng cách nén.

Câu Hỏi và Bài Tập

1. Sự khác nhau giữa truy xuất tuần tự và truy xuất ngẫu nhiên là gì?
2. Việc sử dụng nhiều mức bộ nhớ là gì?
3. Các tính chất chính của bộ nhớ bán dẫn là gì?
4. Giải thích thế nào bộ nhớ truy xuất ngẫu nhiên?
5. Sự khác nhau giữa DRAM và SRAM về mặt ứng dụng của nó là gì?
6. Một số ứng dụng của ROM là gì?
7. Xét một RAM động mà nó phải được làm tươi 64 lần mỗi ms. Mỗi thao tác làm tươi cần 150 ns. Tính phần trăm thời gian tiêu tốn để làm tươi bộ nhớ trong mỗi ms.

Câu Hỏi và Bài Tập

8. Một máy tính có không gian địa chỉ ảo có thể định địa chỉ 32 bit. Mỗi địa chỉ là 1 byte. Kích thước trang là 8 KB. Bao nhiêu trang của không gian địa chỉ ảo có thể tồn tại?
9. Có cần phải có kích thước trang là luỹ thừa của 2? Có thể nói là một trang có kích thước 4000 byte được thực hiện trong lý thuyết? Nếu vậy, nó có thực tế không?
10. Một bộ nhớ ảo có kích thước trang là 1024 từ, 8 trang ảo và 4 khung trang vật lý. Bảng trang được cho như sau:

Câu Hỏi và Bài Tập

Virtual page	Page frame
0	3
1	1
2	not in main memory
3	not in main memory
4	2
5	not in main memory
6	0
7	not in main memory

Hãy cho biết địa chỉ vật lý (dạng nhị phân) ứng với các địa chỉ ảo (dạng thập phân) sau: 0, 1023, 1024, 3728, 4096, 7800.