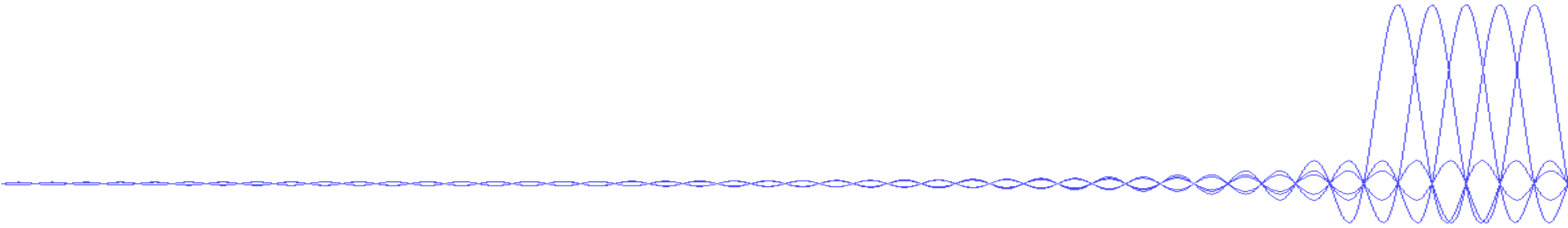
COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

HỆ ĐIỀU HÀNH
Chương 7 – Quản lý bộ nhớ (2)

14/03/2017



COMPUTER ENGINEERING

Câu hỏi ôn tập chương 7-1

- Chuyển đổi địa chỉ là gì? Địa chỉ nhớ được biểu diễn như thế nào trong quá trình chạy 1 chương trình?
- Khi nào địa chỉ lệnh và dữ liệu được chuyển thành địa chỉ thật?
- Thế nào là dynamic linking? Nêu ưu điểm?
- Thế nào là dynamic loading?
- Nêu cơ chế overlay? Swapping?
- Nêu các mô hình quản lý bộ nhớ?

COMPUTER ENGINEERING

Câu hỏi ôn tập chương 7-1 (tt)

- Thế nào là phân mảnh ngoại? Phân mảnh nội? Cho ví dụ?
- Fixed partitioning là gì? Các chiến lược placement?
- Dynamic partitioning là gì? Các chiến lược placement?

COMPUTER ENGINEERING

Câu hỏi ôn tập chương 7-1 (tt)

Giả sử bộ nhớ chính được cấp phát các phân vùng có kích thước là 600K, 500K, 200K, 300K (theo thứ tự), sau khi thực thi xong, các tiến trình có kích thước 212K, 417K, 112K, 426K (theo thứ tự) sẽ được cấp phát bộ nhớ như thế nào, nếu sử dụng: Thuật toán First fit, Best fit, Next fit (con trỏ đang ở vị trí 500K), Worst fit? Thuật toán nào cho phép sử dụng bộ nhớ hiệu quả nhất trong trường hợp trên

COMPUTER ENGINEERING

Mục tiêu chương 7-2

- Hiểu và vận dụng các cơ chế quản lý bộ nhớ:
  - Cơ chế phân trang
  - Cơ chế phân đoạn

COMPUTER ENGINEERING

Nội dung chương 7-1

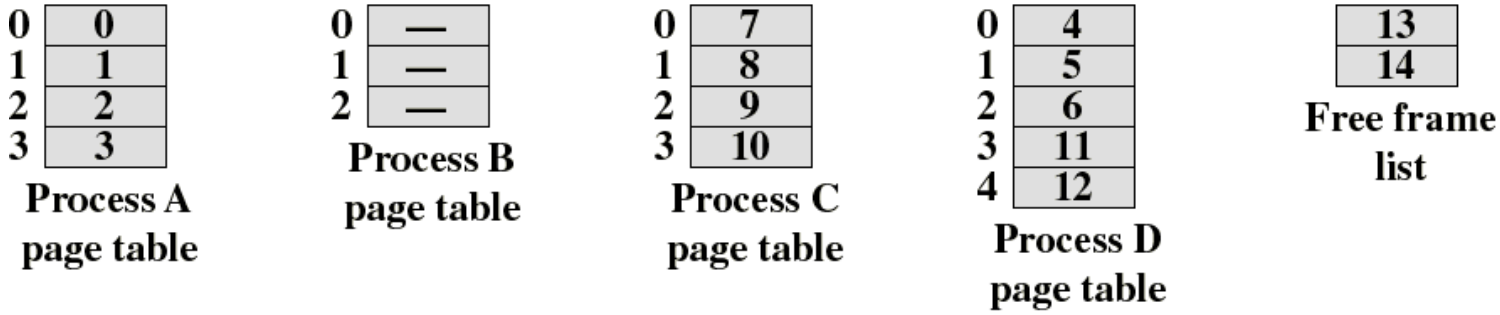
- Cấp phát không liên tục
  - Cơ chế phân trang
  - Cơ chế phân đoạn
  - Cơ chế kết hợp phân trang và phân đoạn

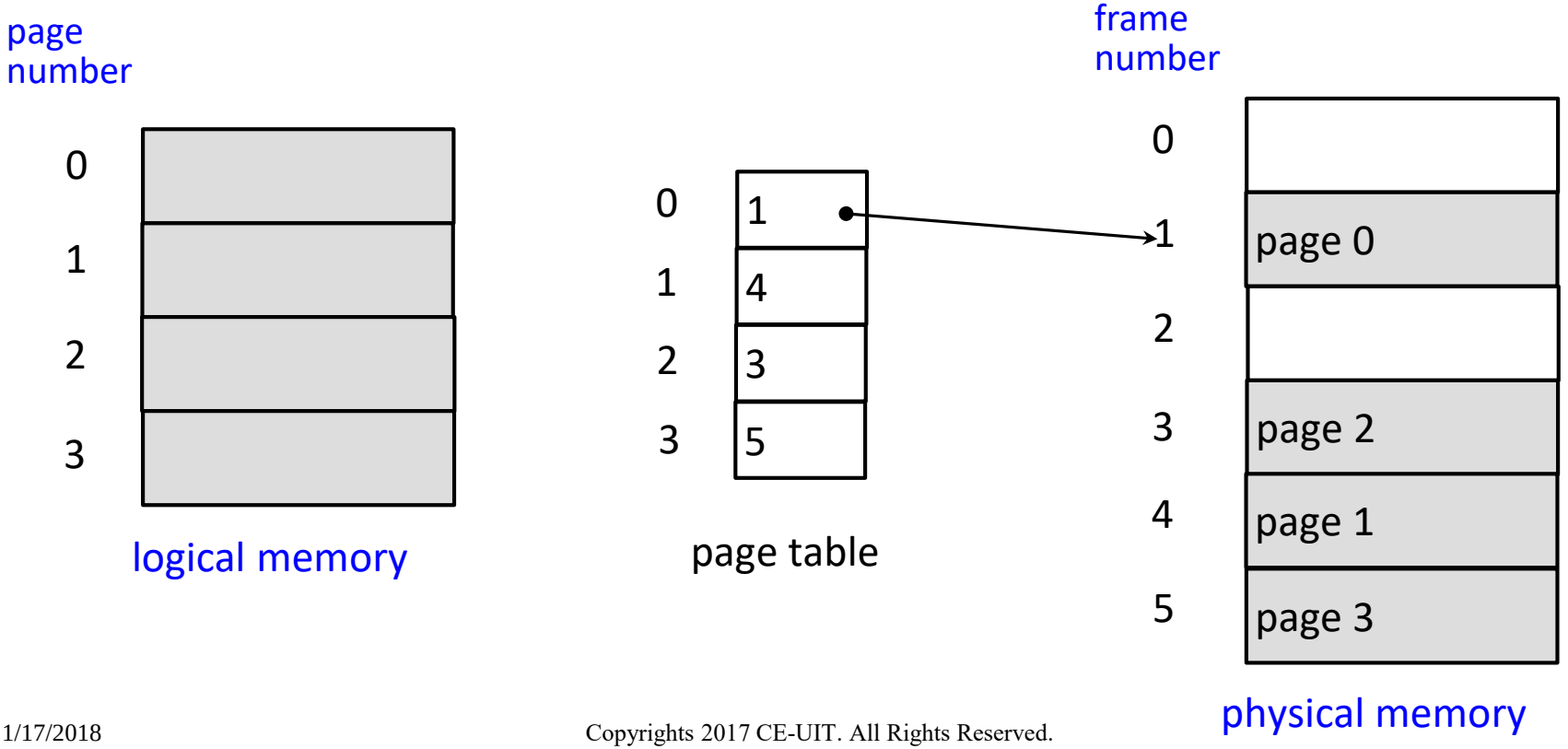
COMPUTER ENGINEERING

Cơ chế phân trang

- Bộ nhớ vật lý → khung trang (frame).
  - Kích thước của frame là lũy thừa của 2, từ khoảng 512 byte đến 16MB.
- Bộ nhớ luận lý (logical memory) hay không gian địa chỉ luận lý là tập mọi địa chỉ luận lý mà một chương trình bất kỳ có thể sinh ra → page.
  - Ví dụ
  - MOV REG,1000 //1000 là một địa chỉ luận lý
- Bảng phân trang (page table) để ánh xạ địa chỉ luận lý thành địa chỉ thực

Cơ chế phân trang (tt)





COMPUTER ENGINEERING

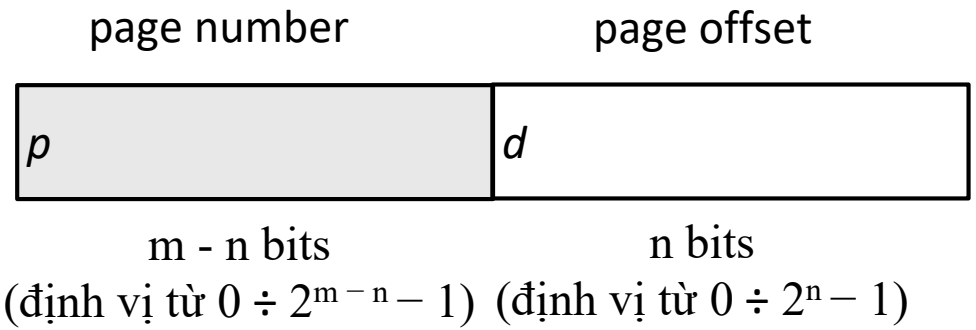
Cơ chế phân trang (tt)

- Chuyển đổi địa chỉ trong paging
- Cài đặt bảng trang
- Effective access time
- Tổ chức bảng trang
- Bảo vệ bộ nhớ

COMPUTER ENGINEERING

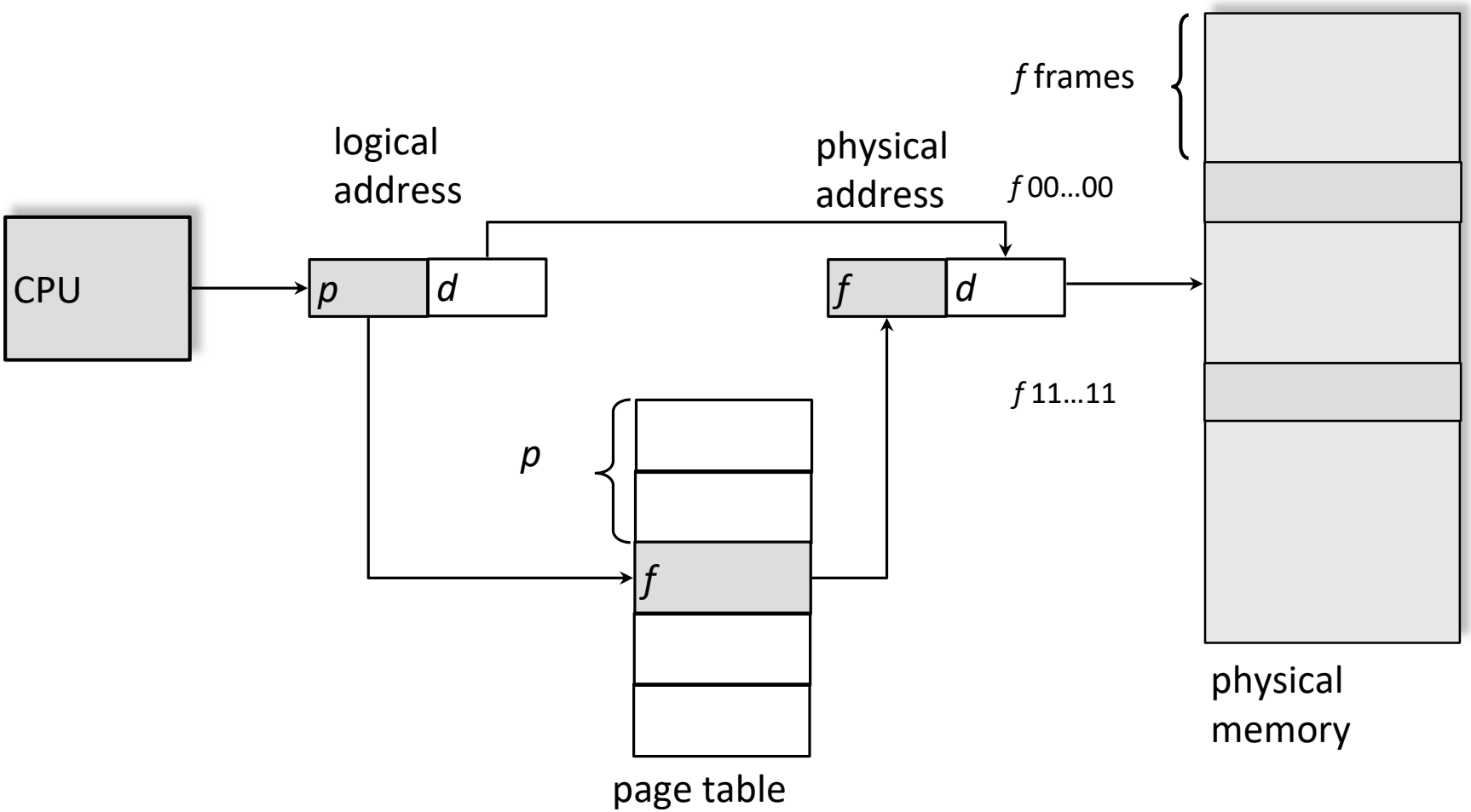
Chuyển đổi địa chỉ trong paging

- Địa chỉ luận lý gồm có:
  - Số hiệu trang (Page number) p
  - Địa chỉ tương đối trong trang (Page offset) d
- Nếu kích thước của không gian địa chỉ ảo là 2^m, và kích thước của trang là 2^n (đơn vị là byte hay word tùy theo kiến trúc máy) thì



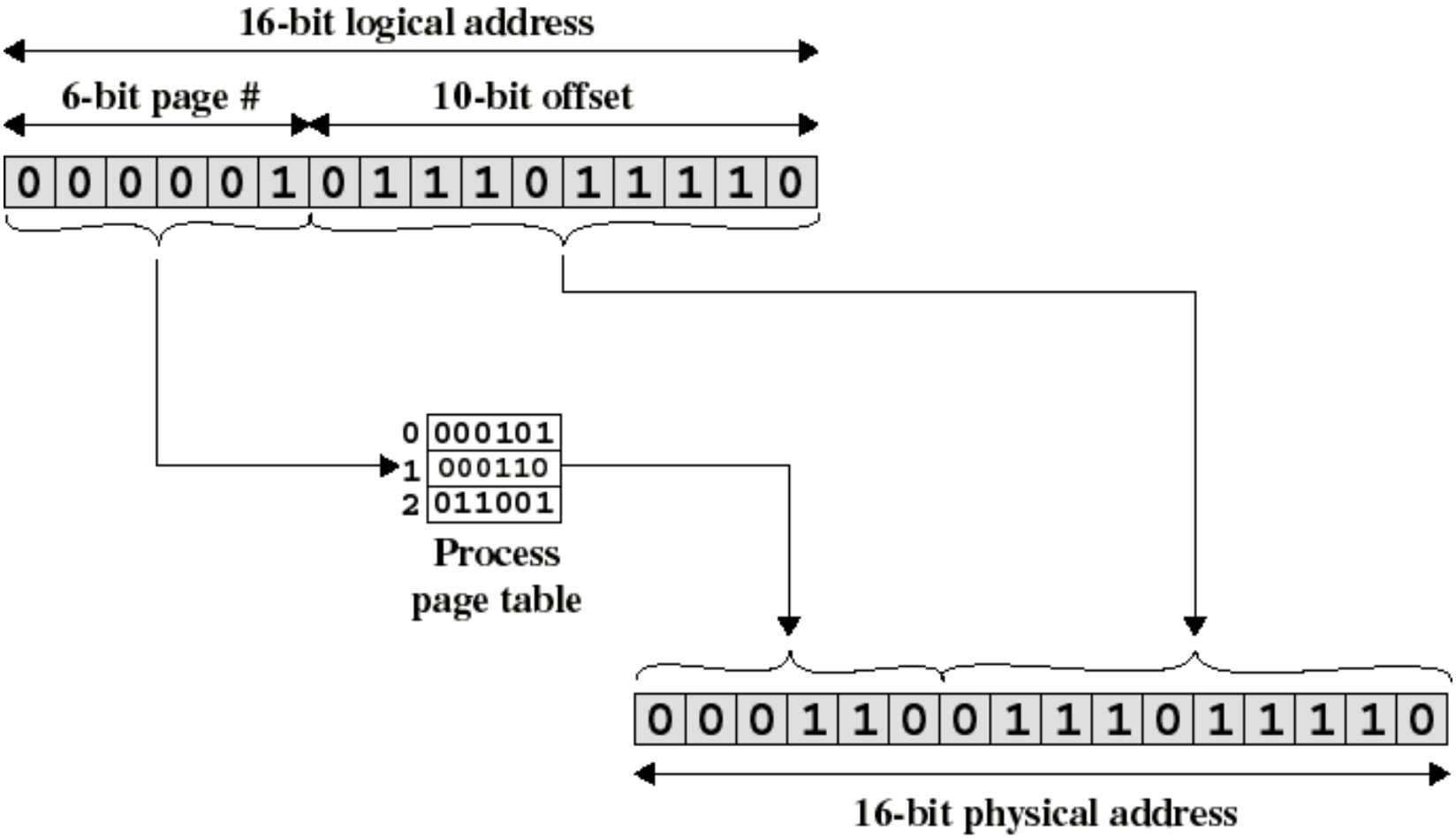
Bảng trang sẽ có tổng cộng $2^m/2^n = 2^{m-n}$ mục (entry)

Chuyển đổi địa chỉ trong paging

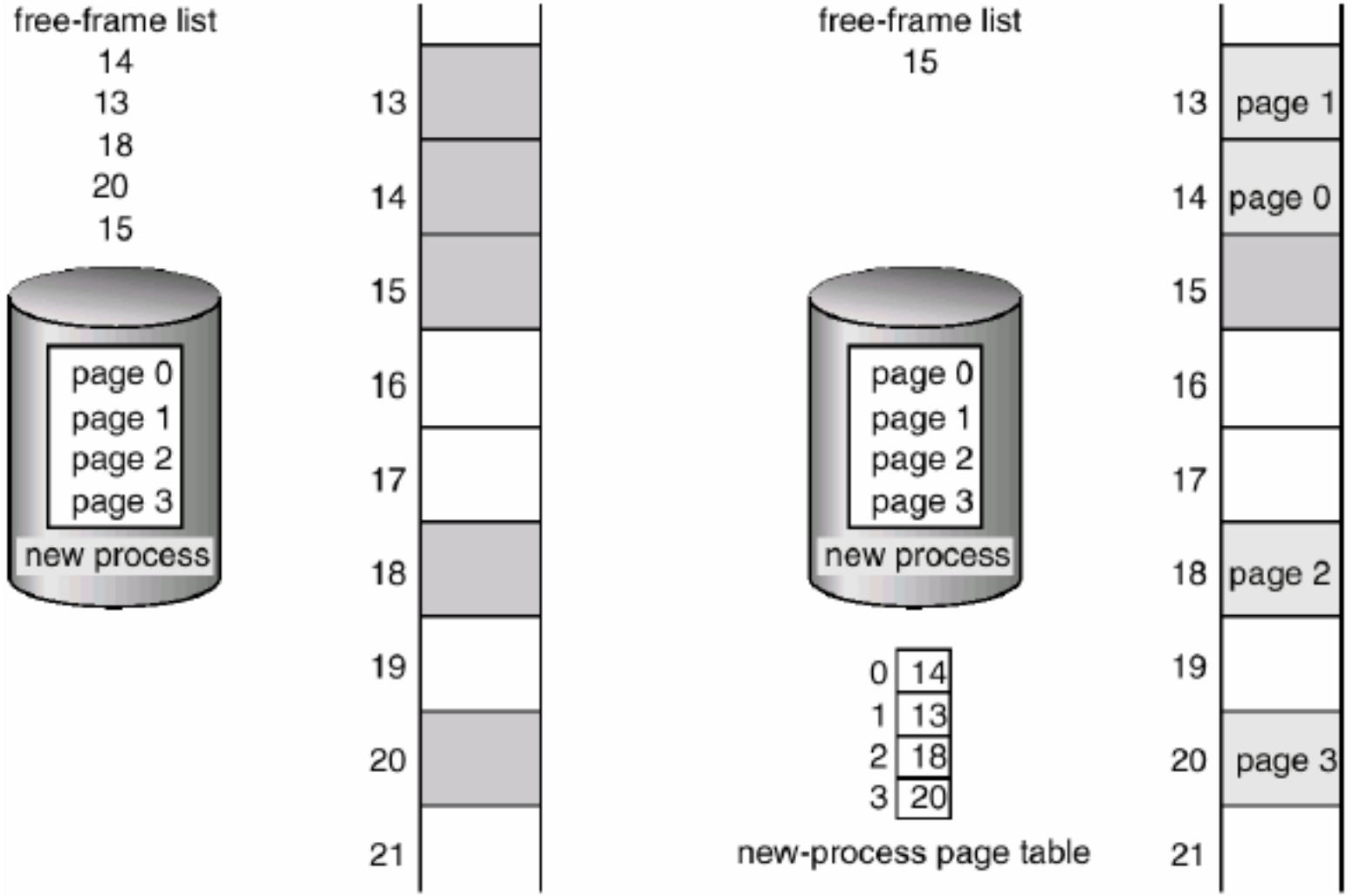


Chuyển đổi địa chỉ trong paging (tt)

- Ví dụ:



Cơ chế phan trang (tt)



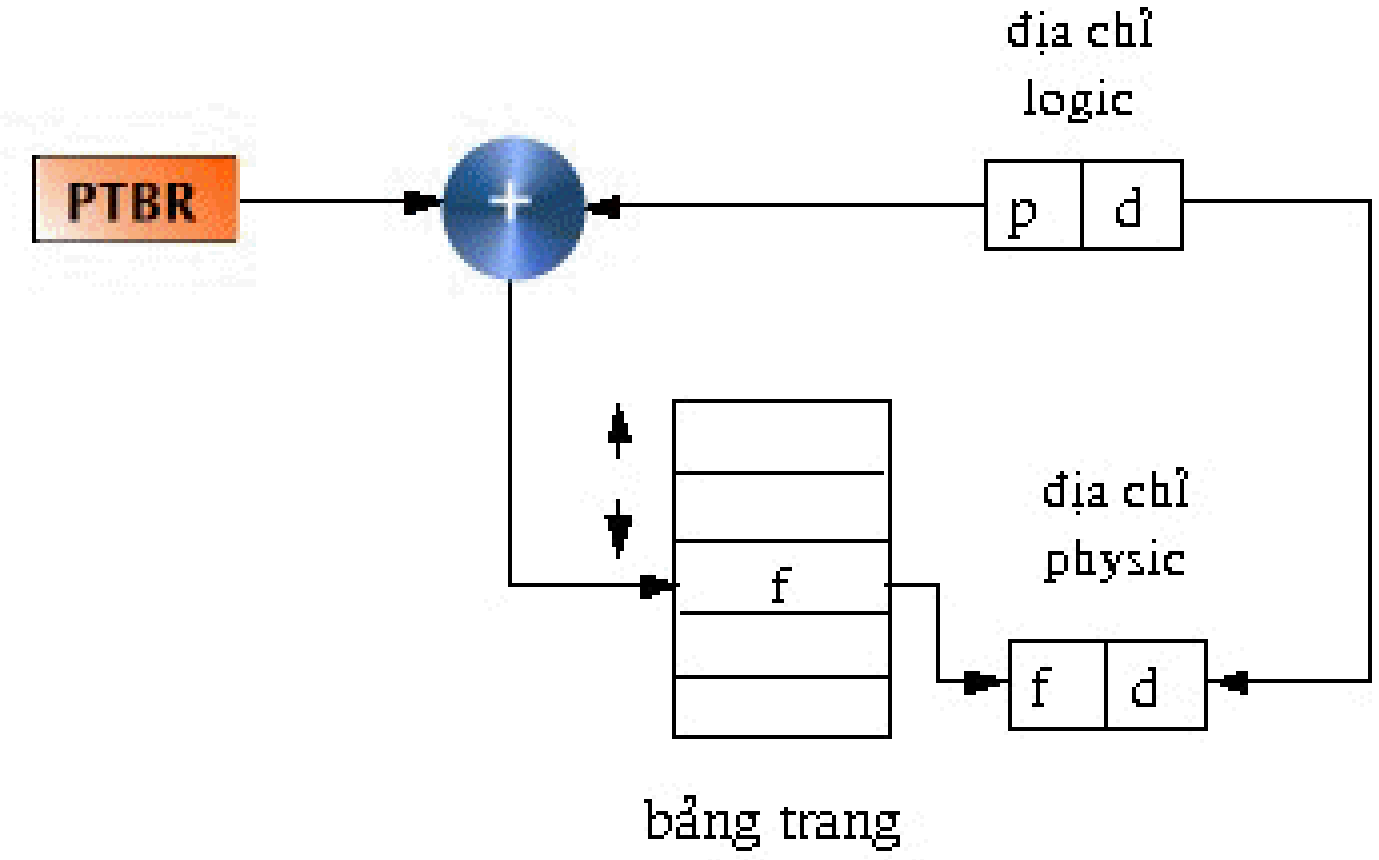
COMPUTER ENGINEERING

Cài đặt bảng trang (paging hardware)

- Bảng phân trang thường được lưu giữ trong bộ nhớ chính
  - Mỗi process được hệ điều hành cấp một bảng phân trang
  - Thanh ghi page-table base (PTBR) trỏ đến bảng phân trang
  - Thanh ghi page-table length (PTLR) biểu thị kích thước của bảng phân trang (có thể được dùng trong cơ chế bảo vệ bộ nhớ)
- Thường dùng một bộ phận cache phần cứng có tốc độ truy xuất và tìm kiếm cao, gọi là thanh ghi kết hợp (associative register) hoặc translation look-aside buffers (TLBs)

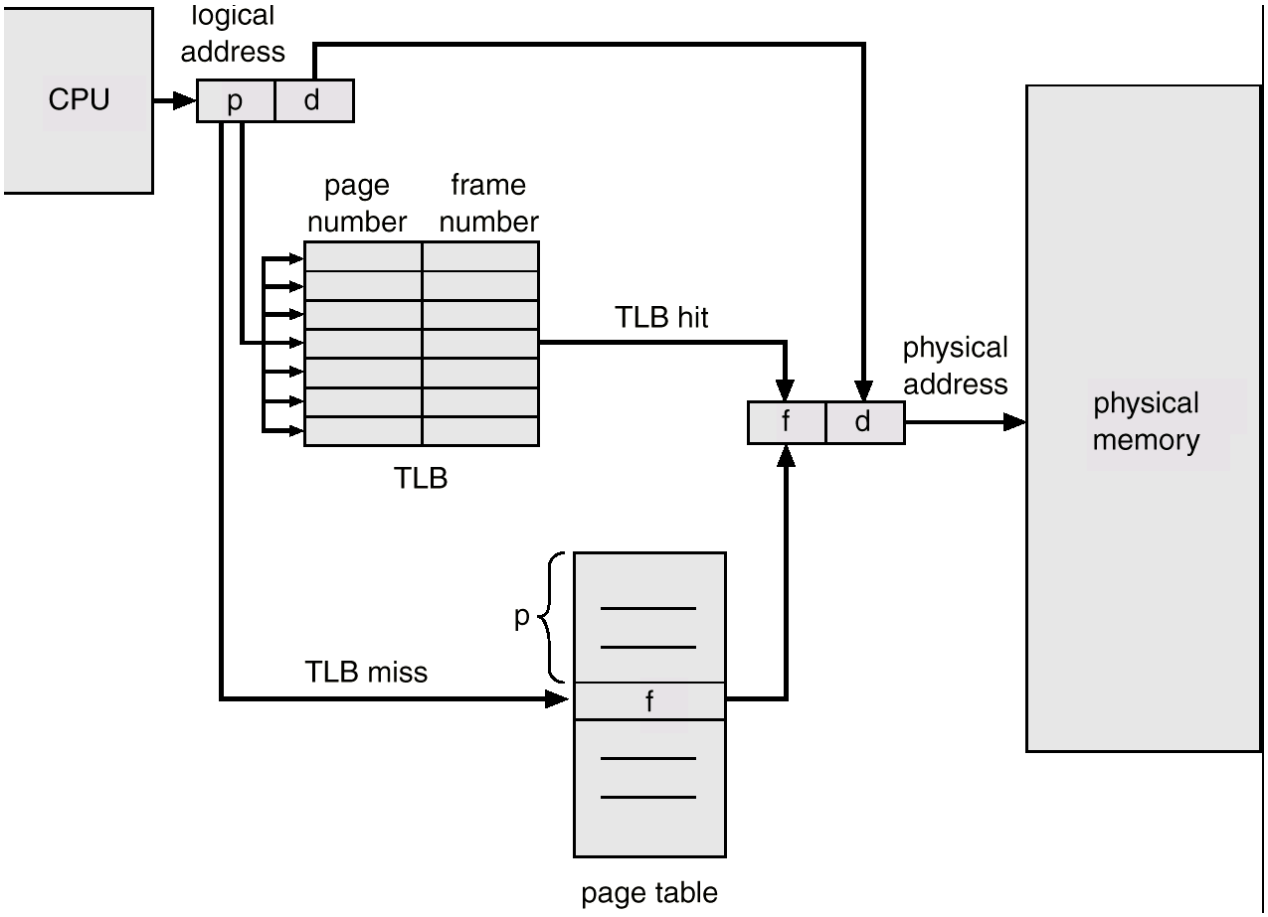
Cài đặt bảng trang (tt)

- Dùng thanh ghi Page-Table Base Register (PTBR)



Cài đặt bảng trang (tt)

- Dùng TLB



COMPUTER ENGINEERING

Effective access time (EAT)

- Tính thời gian truy xuất hiệu dụng (effective access time, EAT)
- Thời gian tìm kiếm trong TLB (associative lookup): ε
- Thời gian một chu kỳ truy xuất bộ nhớ: x
- Hit ratio: tỉ số giữa số lần chỉ số trang được tìm thấy (hit) trong TLB và số lần truy xuất khởi nguồn từ CPU
  - Kí hiệu hit ratio: α
- Thời gian cần thiết để có được chỉ số frame
  - Khi chỉ số trang có trong TLB (hit) $\varepsilon + x$
  - Khi chỉ số trang không có trong TLB (miss) $\varepsilon + x + x$
- Thời gian truy xuất hiệu dụng

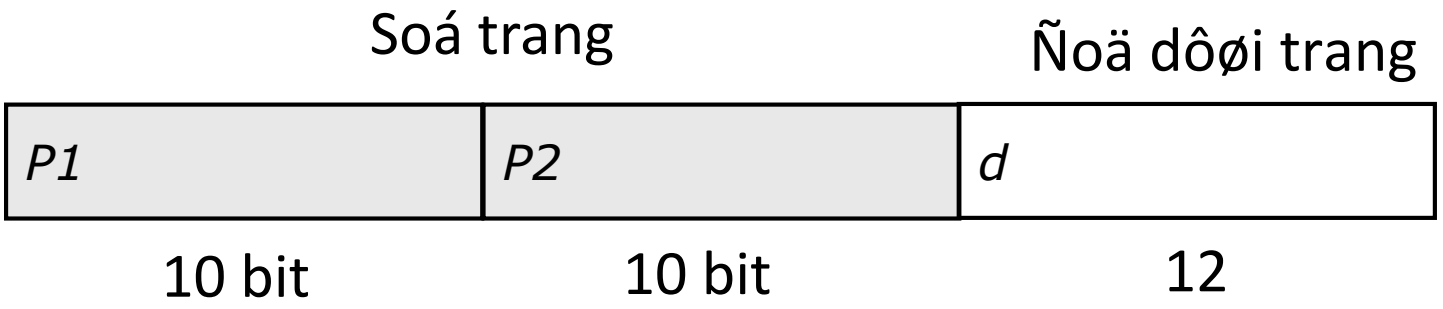
$$EAT = (\varepsilon + x)\alpha + (\varepsilon + 2x)(1 - \alpha)$$
$$= (2 - \alpha)x + \varepsilon$$

Effective access time (EAT) (tt)

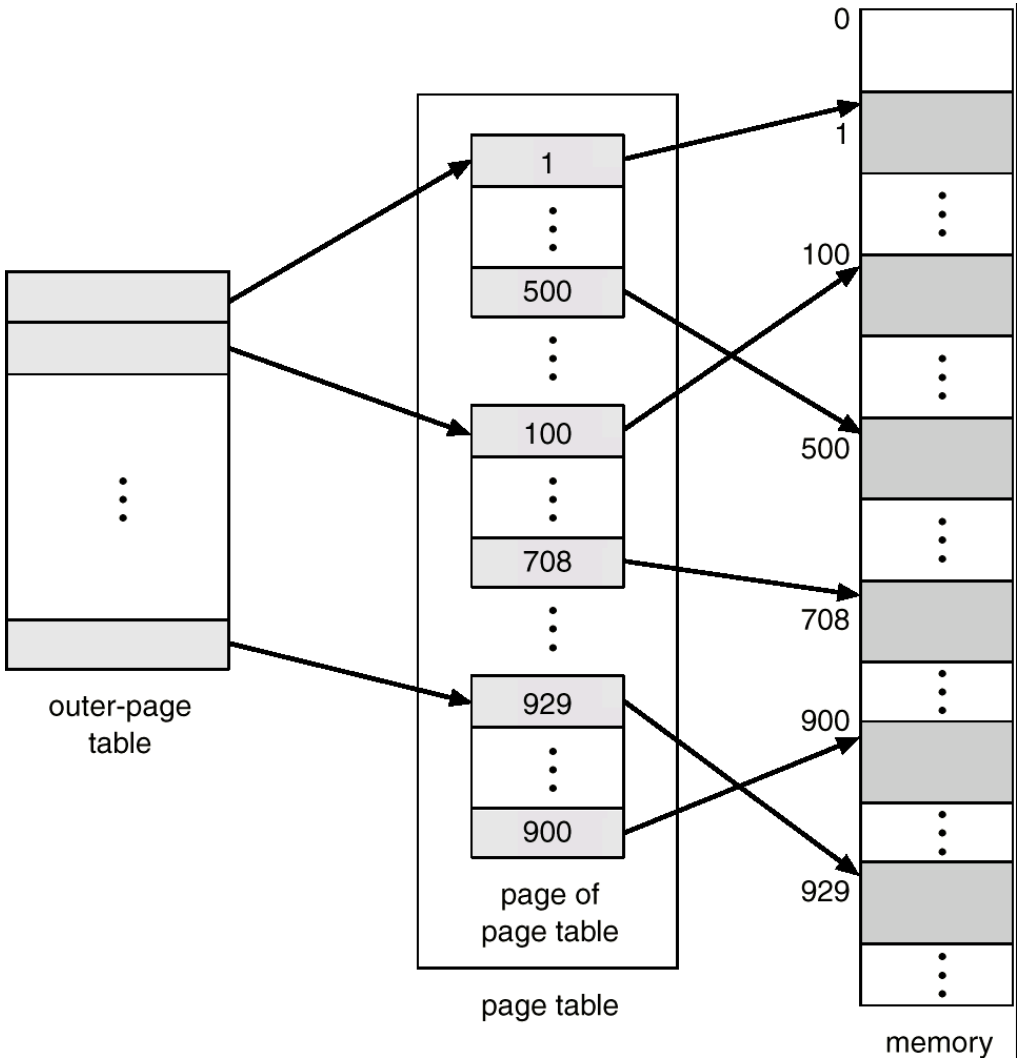
- Ví dụ 1: đơn vị thời gian nano giây
  - Associative lookup = 20
  - Memory access = 100
  - Hit ratio = 0.8
  - EAT = (100 + 20) × 0.8 + (200 + 20) × 0.2
    = 1.2 × 100 + 20
    = 140
- Ví dụ 2: đơn vị thời gian nano giây
  - Associative lookup = 20
  - Memory access = 100
  - Hit ratio = 0.98
  - EAT = (100 + 20) × 0.98 + (200 + 20) × 0.02
    = 1.02 × 100 + 20
    = 122

Tổ chức bảng trang

- Các hệ thống hiện đại đều hỗ trợ không gian địa chỉ ảo rất lớn (2^{32} đến 2^{64}), ở đây giả sử là 2^{32}
  - Giả sử kích thước trang nhớ là 4KB ($= 2^{12}$)
    $\Rightarrow$ bảng phân trang sẽ có $2^{32}/2^{12} = 2^{20} = 1$M mục.
  - Giả sử mỗi mục gồm 4 byte thì mỗi process cần 4MB cho bảng phân trang
    - Ví dụ: Phân trang 2 cấp



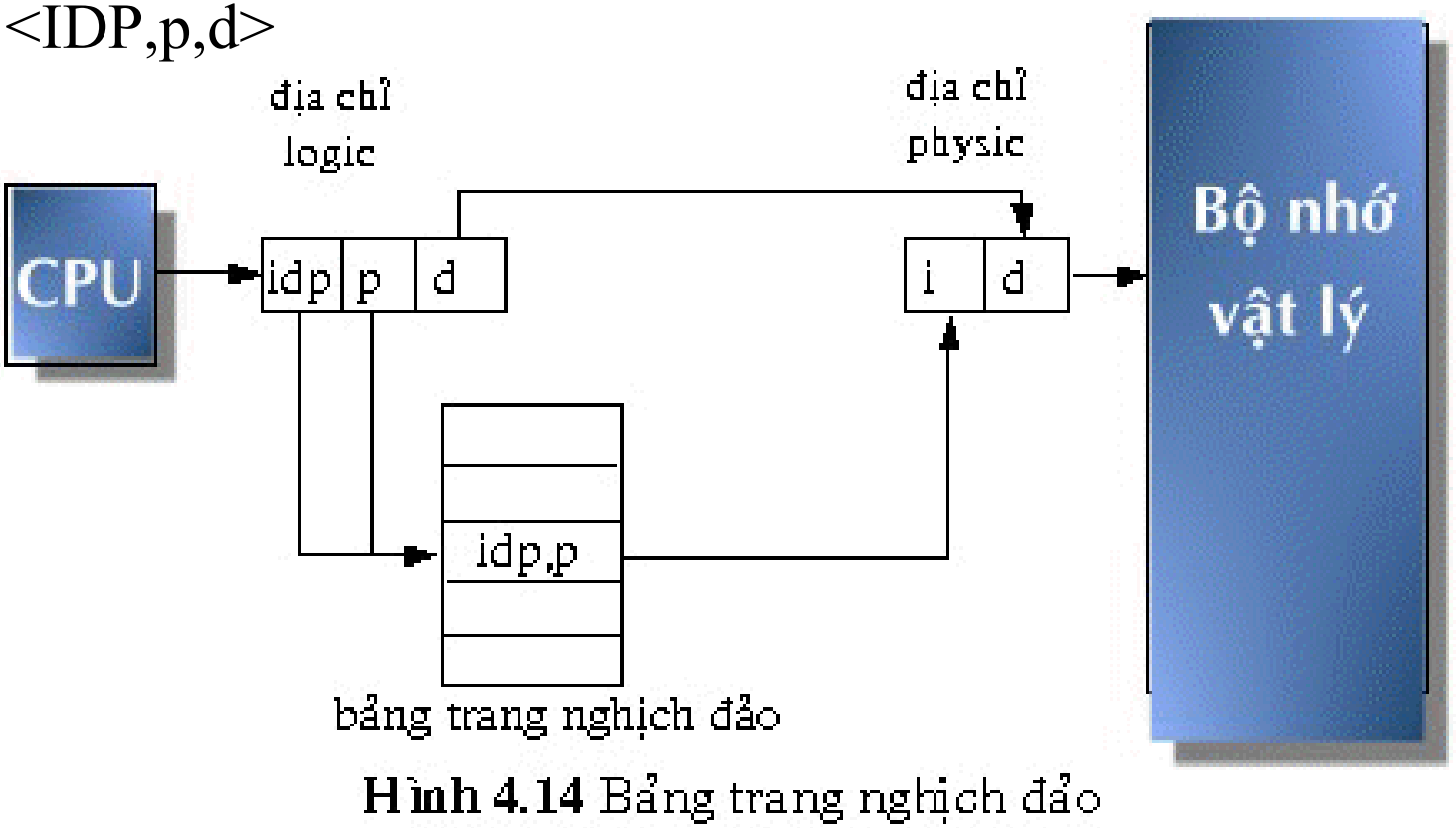
Tổ chức bảng trang (tt)



Tổ chức bảng trang (tt)

Bảng trang nghịch đảo (IBM system/38, IBM RISC, IBM RT): sử dụng cho tất cả các Process

<IDP,p,d>



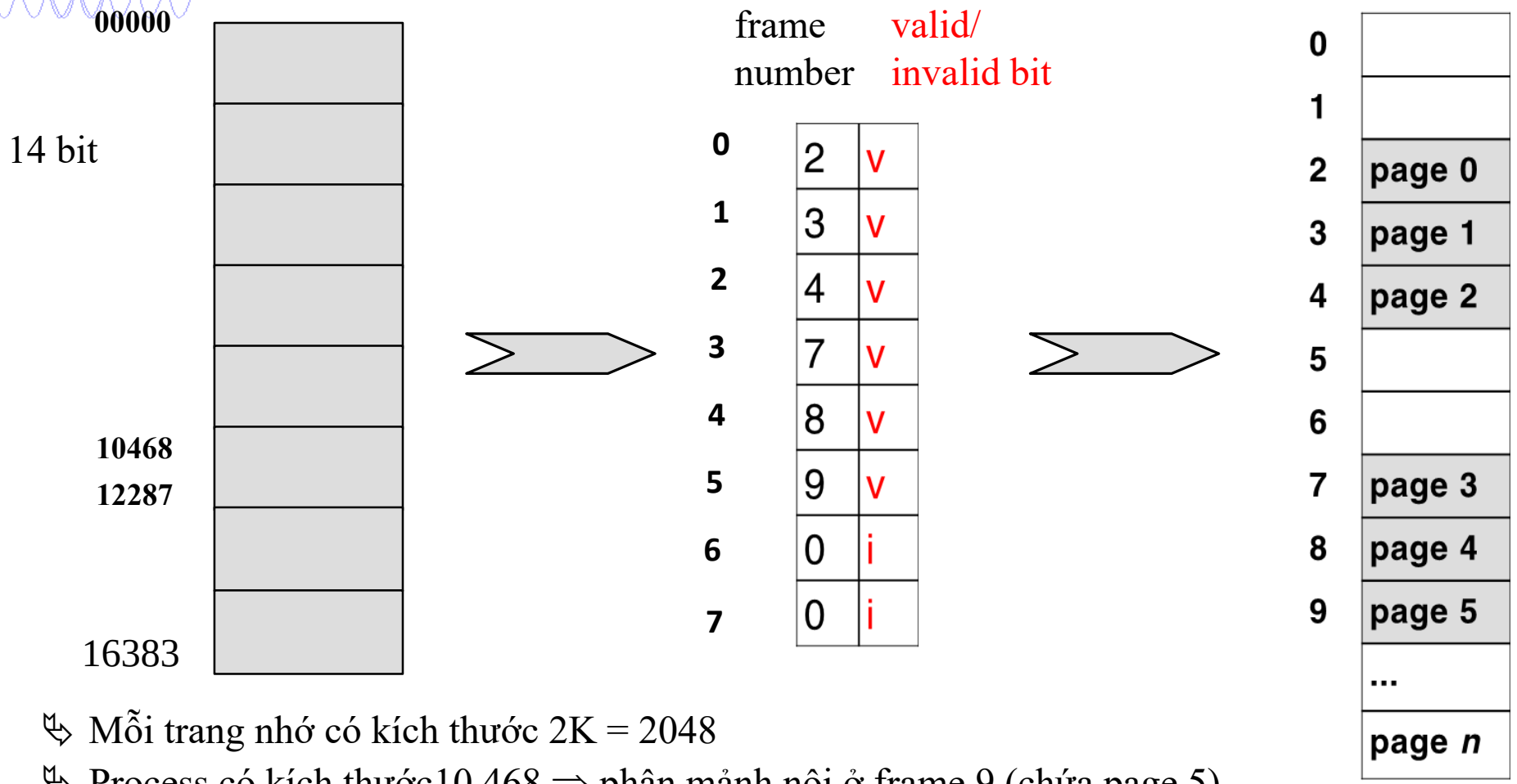
Hình 4.14 Bảng trang nghịch đảo

COMPUTER ENGINEERING

Bảo vệ bộ nhớ

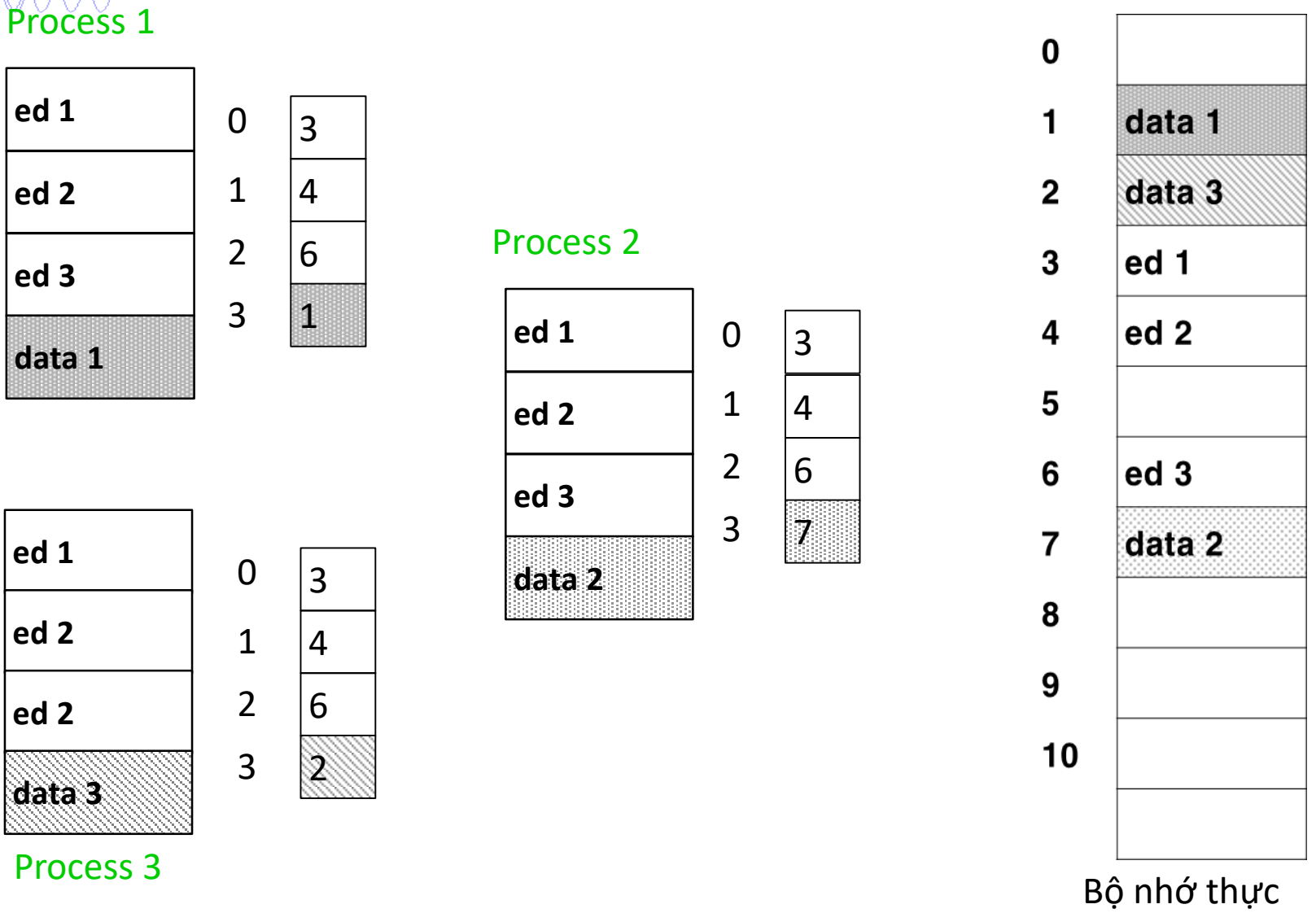
- Việc bảo vệ bộ nhớ được hiện thực bằng cách gắn với frame các bit bảo vệ (protection bits) được giữ trong bảng phân trang. Các bit này biểu thị các thuộc tính sau
  - read-only, read-write, execute-only
- Ngoài ra, còn có một valid/invalid bit gắn với mỗi mục trong bảng phân trang
  - "valid": cho biết là trang của process, do đó là một trang hợp lệ.
  - "invalid": cho biết là trang không của process, do đó là một trang bất hợp lệ.

Bảo vệ bằng valid/invalid bit



- Mỗi trang nhớ có kích thước 2K = 2048
- Process có kích thước10,468 ⇒ phân mảnh nội ở frame 9 (chứa page 5), các địa chỉ ảo > 12287 là các địa chỉ invalid.
- Dùng PTLR để kiểm tra truy xuất đến bảng phân trang có nằm trong bảng hay không.

Chia sẻ các trang nhớ



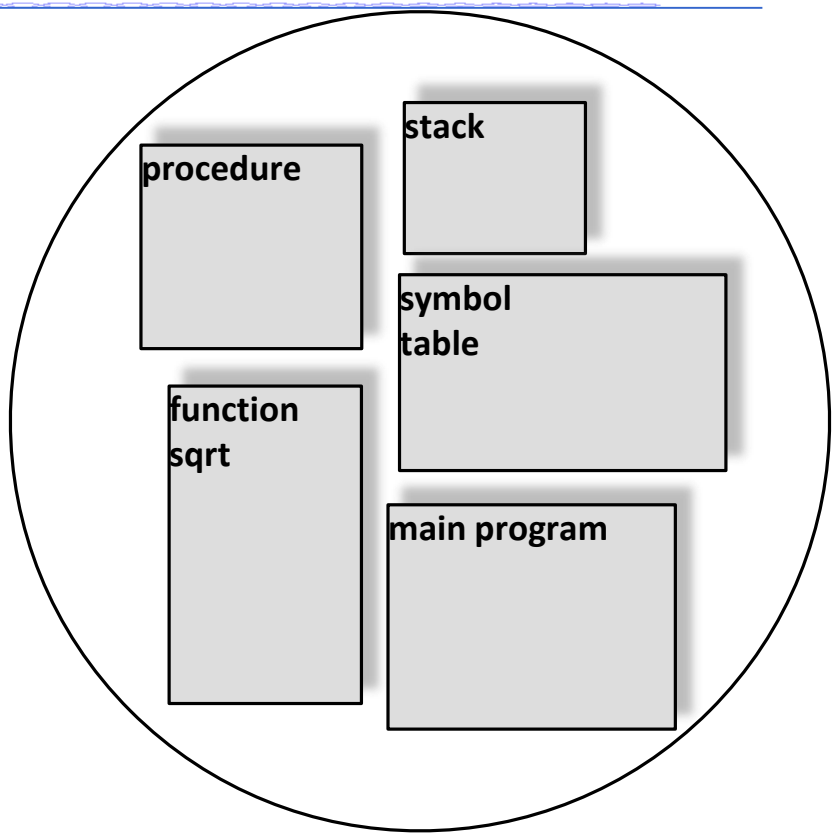
COMPUTER ENGINEERING

Phân đoạn (segmentation)

- Nhìn lại cơ chế phân trang
  - user view (không gian địa chỉ ảo) tách biệt với không gian bộ nhớ thực. Cơ chế phân trang thực hiện phép ánh xạ user-view vào bộ nhớ thực.
- Trong thực tế, dưới góc nhìn của user, một chương trình cấu thành từ nhiều đoạn (segment). Mỗi đoạn là một đơn vị luận lý của chương trình, như
  - main program, procedure, function
  - local variables, global variables, common block, stack, symbol table, arrays,…

User view của một chương trình

- Thông thường, một chương trình được biên dịch. Trình biên dịch sẽ tự động xây dựng các segment.
- Ví dụ, trình biên dịch Pascal sẽ tạo ra các segment sau:
  - Global variables
  - Procedure call stack
  - Procedure/function code
  - Local variable
- Trình loader sẽ gán mỗi segment một số định danh riêng.



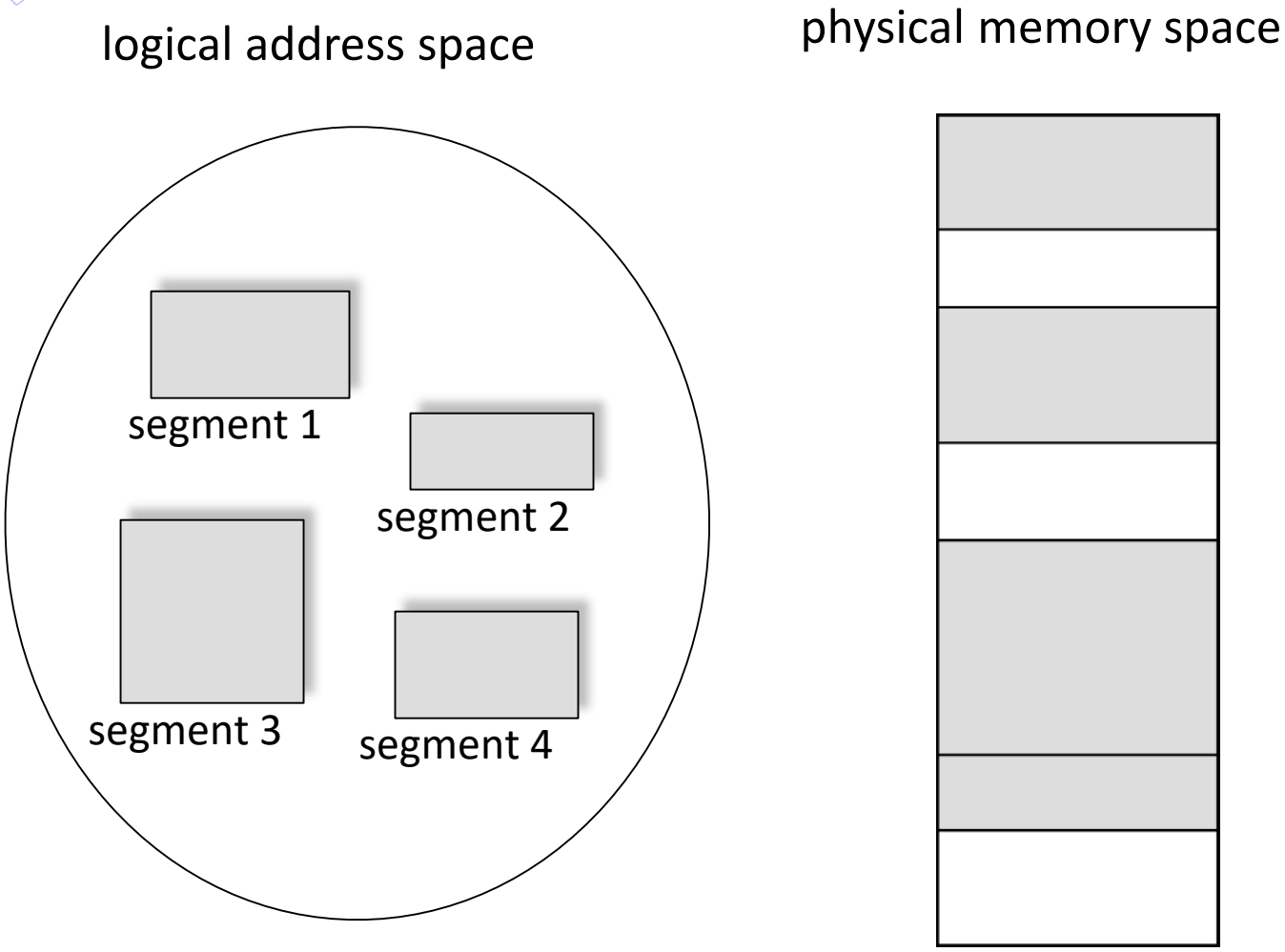
Logical address space

COMPUTER ENGINEERING

Phân đoạn

- Dùng cơ chế phân đoạn để quản lý bộ nhớ có hỗ trợ user view
  - Không gian địa chỉ ảo là một tập các đoạn, mỗi đoạn có tên và kích thước riêng.
  - Một địa chỉ luận lý được định vị bằng tên đoạn và độ dời (offset) bên trong đoạn đó (so sánh với phân trang!)

Phân đoạn (tt)



COMPUTER ENGINEERING

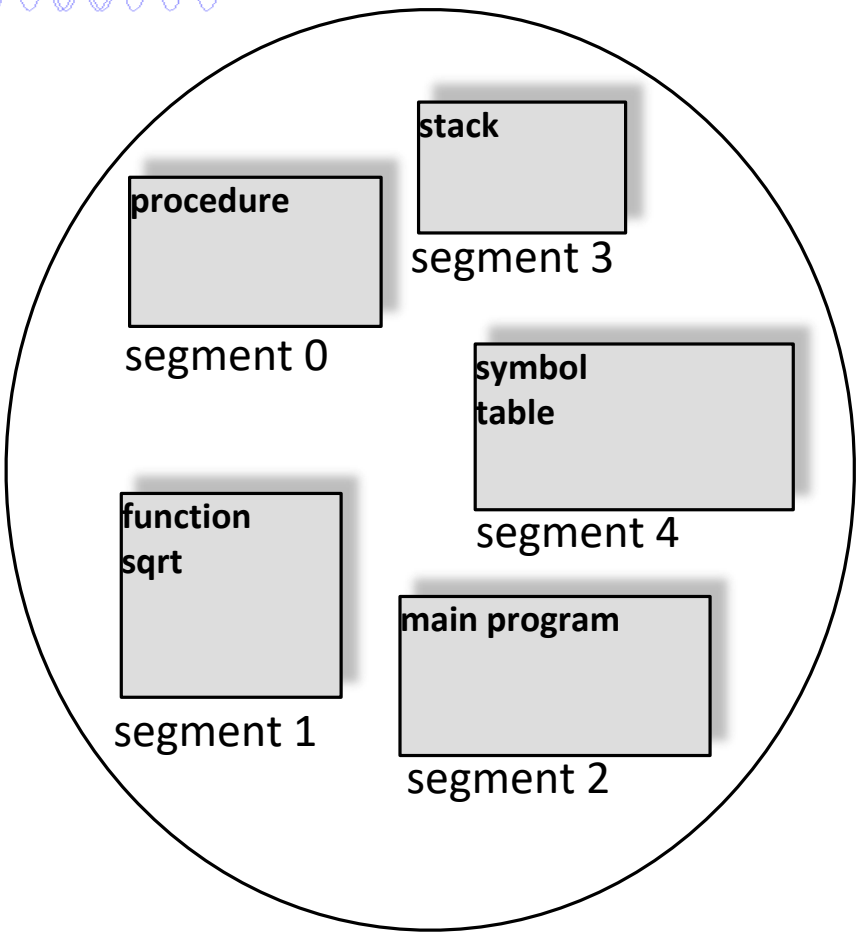
Cài đặt phan đoạn

- Địa chỉ luận lý là một cặp giá trị

  (segment number, offset)

- Bảng phân đoạn (segment table): gồm nhiều mục, mỗi mục chứa
  - base, chứa địa chỉ khởi đầu của segment trong bộ nhớ
  - limit, xác định kích thước của segment
- Segment-table base register (STBR): trỏ đến vị trí bảng phân đoạn trong bộ nhớ
- Segment-table length register (STLR): số lượng segment của chương trình
- ⇒ Một chỉ số segment s là hợp lệ nếu s < STLR

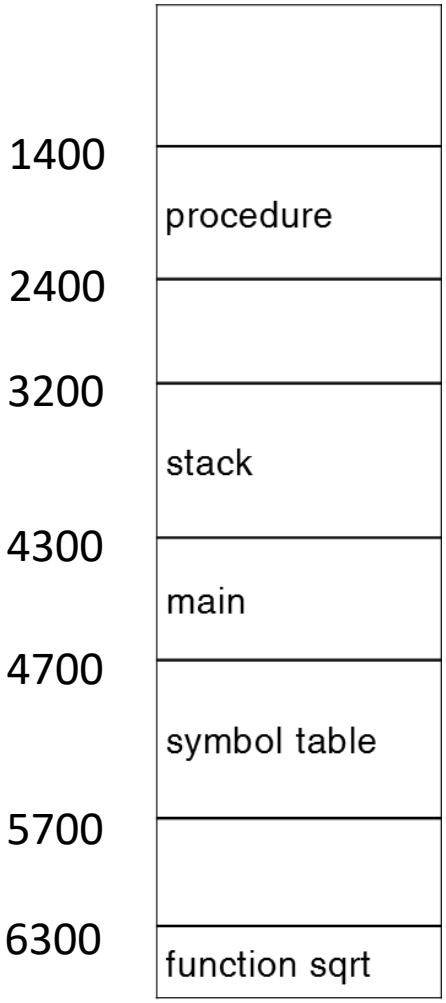
Một ví dụ về phân đoạn



logical address space

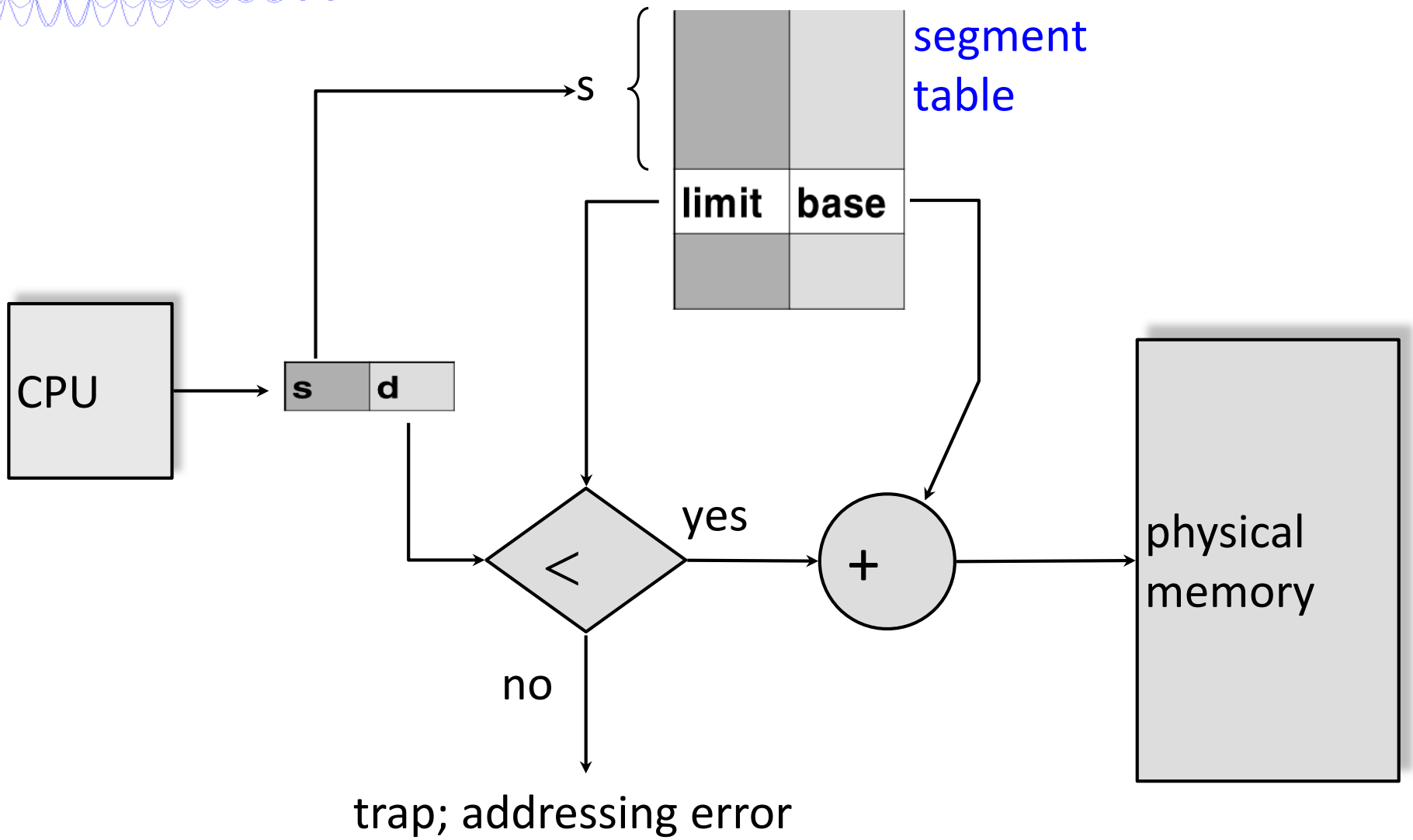
	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

segment table

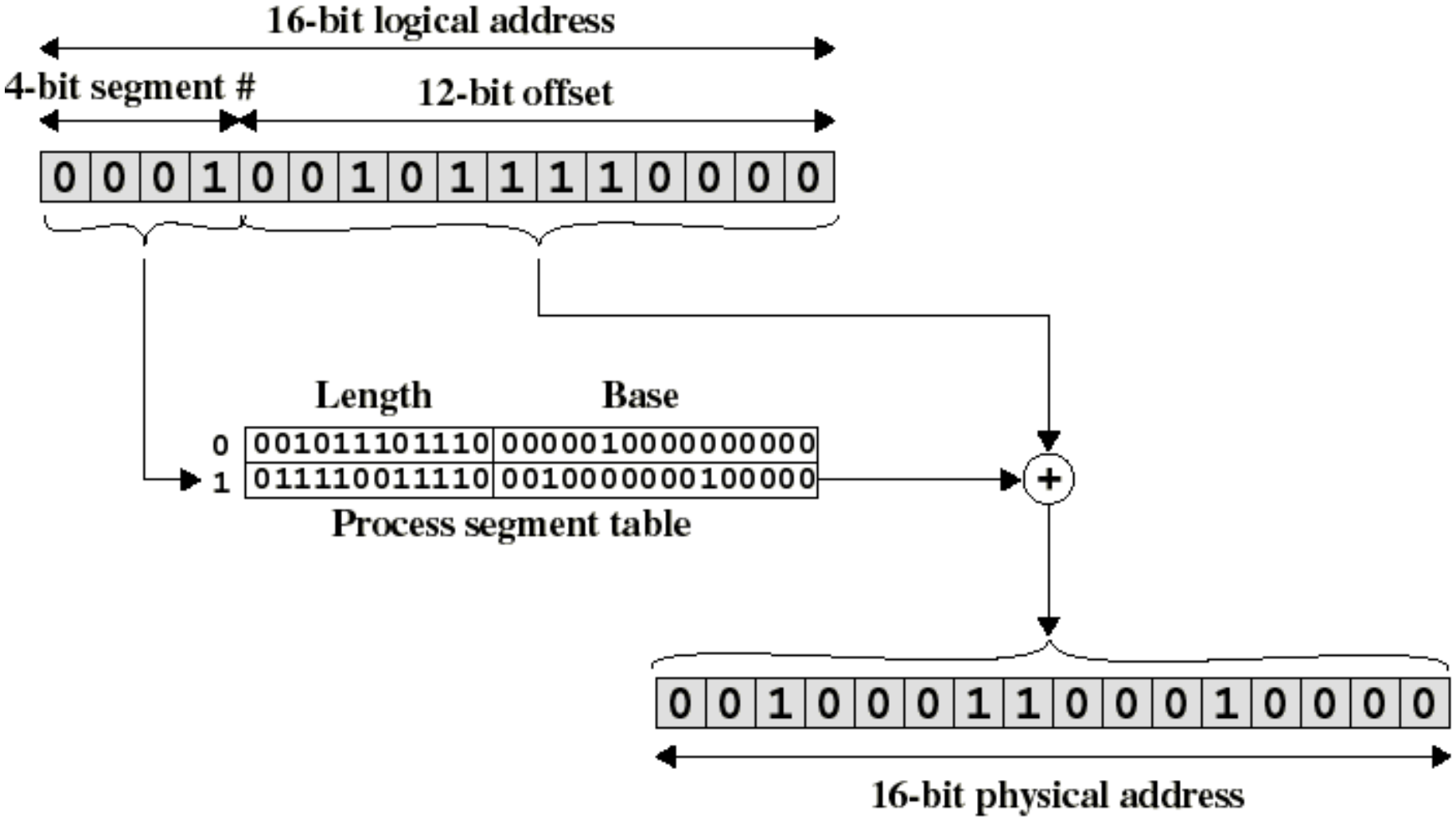


physical memory space

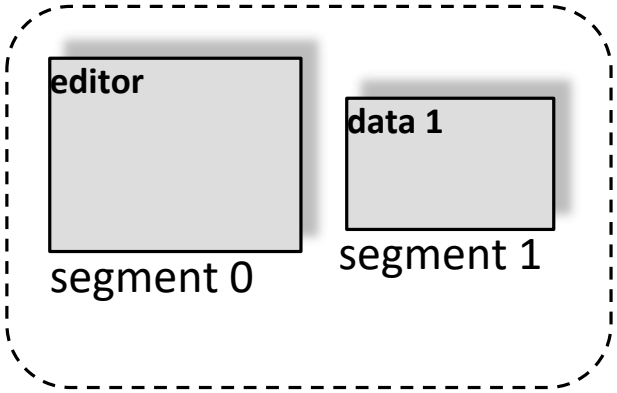
Phần cứng hỗ trợ phân đoạn



Chuyển đổi địa chỉ trong cơ chế phan đoạn



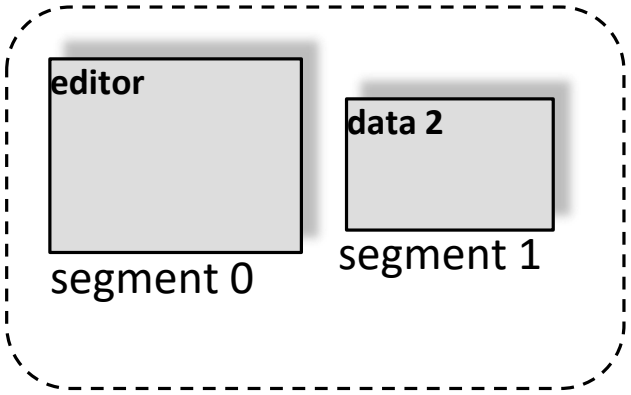
Chia sẻ các đoạn



logical address space
process P_1

	limit	base
0	25286	43062
1	4425	68348

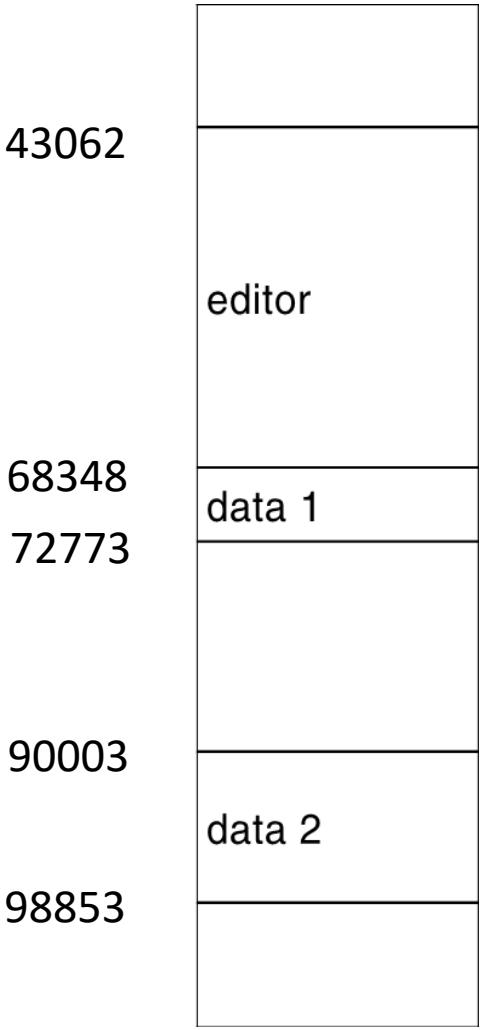
segment table
process P_1



logical address space
process P_2

	limit	base
0	25286	43062
1	8850	90003

segment table
process P_2



physical memory

COMPUTER ENGINEERING

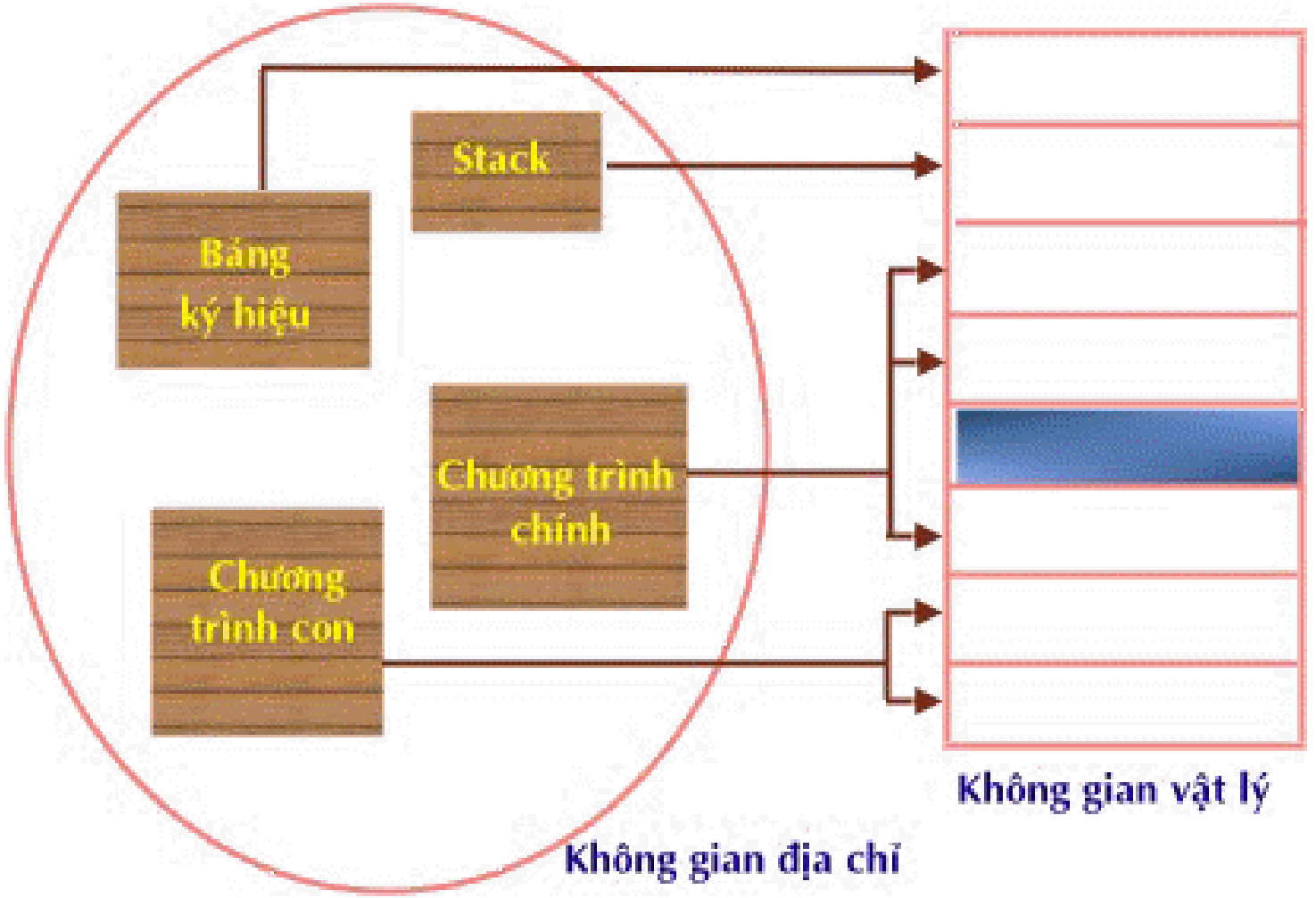
Kết hợp phân trang và phân đoạn

- Kết hợp phân trang và phân đoạn nhằm kết hợp các ưu điểm đồng thời hạn chế các khuyết điểm của phân trang và phân đoạn:
  - Vấn đề của phân đoạn: Nếu một đoạn quá lớn thì có thể không nạp nó được vào bộ nhớ.
  - Ý tưởng giải quyết: paging đoạn, khi đó chỉ cần giữ trong bộ nhớ các page của đoạn hiện đang cần.

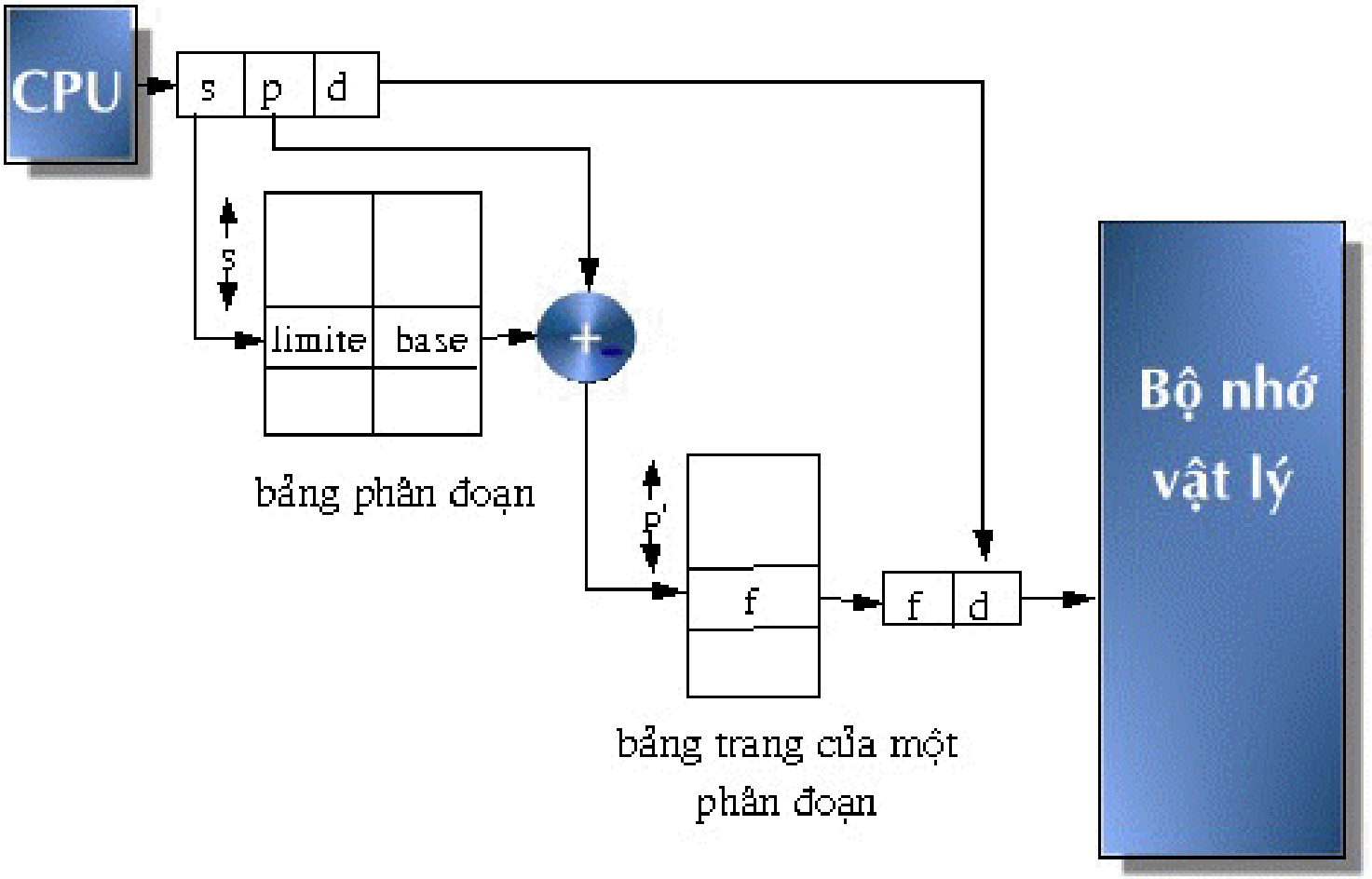
Logic Addr = <s,p,d>

COMPUTER ENGINEERING

Kết hợp phân trang và phân đoạn (tt)



Cài đặt phân đoạn



Hình 4.23 Cơ chế phần cứng của sự phân đoạn kết hợp phân trang

COMPUTER ENGINEERING

Tóm tắt lại nội dung buổi học

- Cấp phát không liên tục
  - Cơ chế phân trang
  - Cơ chế phân đoạn
  - Cơ chế kết hợp phân trang và phân đoạn

COMPUTER ENGINEERING

Bài tập 1

Xét một không gian địa chỉ có 12 trang, mỗi trang có kích thước 2K, ánh xạ vào bộ nhớ vật lý có 32 khung trang.

a. Địa chỉ logic gồm bao nhiêu bit?

b. Địa chỉ physic gồm bao nhiêu bit?

COMPUTER ENGINEERING

Bài tập 2

Xét một hệ thống sử dụng kỹ thuật phân trang, với bảng trang được lưu trữ trong bộ nhớ chính.

a. Nếu thời gian cho một lần truy xuất bộ nhớ bình thường là 200ns thì mất bao nhiêu thời gian cho một thao tác truy xuất bộ nhớ trong hệ thống này?

b. Nếu sử dụng TLBs với hit-ratio là 75%, thời gian để tìm tròn TLBs xem như bằng 0, tính thời gian truy xuất bộ nhớ trong hệ thống

COMPUTER ENGINEERING

Bài tập 3

Xét bảng phân đoạn trong hình

Tính địa chỉ vật lý tương ứng với các địa chỉ logic sau đây:

a. 430
b. 110
c. 2500
d. 3400
e. 4112

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96