

# Cấu trúc 8088/86

- Địa chỉ logic có dạng:  
Địa chỉ đoạn (16 bit): offset (16 bit)
- Địa chỉ vật lý (20 bit) = địa chỉ đoạn \* 10h + offset
- Ví dụ:  
Có địa chỉ logic 1234h:0076h  
 $\Rightarrow$  địa chỉ vật lý =  $1234h * 10h + 0076h = 123B6h$
- Người lập trình chỉ lập trình với địa chỉ logic, còn việc chuyển sang địa chỉ vật lý là do bộ vi xử lý thực hiện.

# Giới thiệu lập trình hợp ngữ

- Chương trình hợp ngữ gồm các dòng lệnh, mỗi lệnh viết trên một dòng, không phân biệt chữ hoa, chữ thường.
- Khi dịch thành mã máy thì chỉ có các lệnh của bộ vi xử lý mới được dịch

Cấu trúc của một dòng lệnh :

Tên	Thao tác	Toán hạng	Chú thích
( Name	Operation	Operand	Comment )

- Giữa các trường phải có ít nhất một dấu cách (hoặc TAB)
- Ví dụ:

**MAIN        PROC**

**BAT\_DAU: MOV   CX, 50 ; khai tạo bo dem**

# Giới thiệu lập trình hợp ngữ

- Hợp ngữ cho phép biểu diễn dữ liệu dưới dạng:
  - Số nhị phân: 1011b, 1011B, ...
  - Số thập phân: 35, 35d, 35D, ...
  - Số Hexa: 4Ah, 0ABCDh, 0FFFFH, ...
  - Kí tự: "A", 'HELLO', "Bach Khoa", ...
- Tất cả các kiểu dữ liệu trên sau đó đều được trình dịch Assembler dịch ra mã nhị phân.
- Mỗi kí tự được dịch thành mã ASCII tương ứng
  - Chương trình không phân biệt 'A' với 41h hay 65

# Giới thiệu lập trình hợp ngữ

Chỉ thị	Biểu diễn
DB	Định nghĩa byte
DW	Định nghĩa word (2 byte)
DD	Định nghĩa double word (4 byte)
DQ	Định nghĩa quadword (8 byte liên tiếp)
DT	Định nghĩa tenbyte (10 byte liên tiếp)

## ■ Biến Byte:

### ◦ Khai báo:

Ten_bien	DB	Gia_tri_khoi_dau
Ten_bien	DB	?

### ◦ Ví dụ:

Age	DB	25	; Khởi tạo giá trị ban đầu Age = 25
Alpha	DB	?	; Ban đầu Alpha không xác định

# Giới thiệu lập trình hợp ngữ

## ■ Biến Word:

### ◦ Khai báo:

Ten_bien	DW	Gia_tri_khoi_dau
Ten_bien	DW	?

### ◦ Ví dụ:

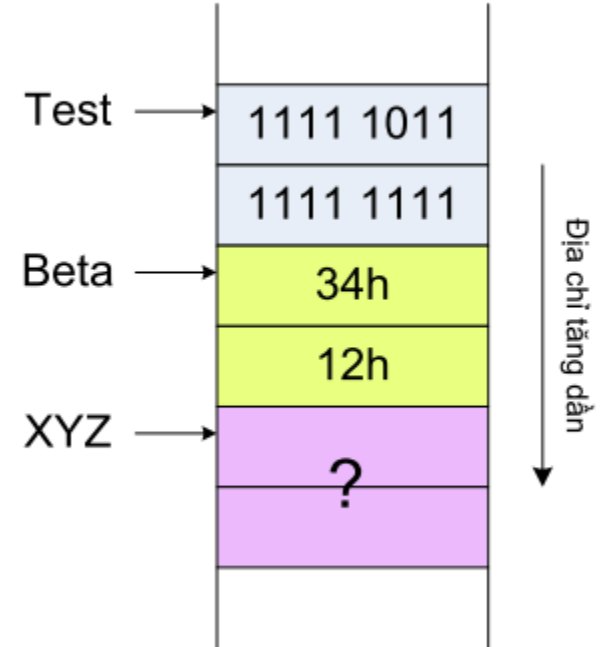
Test DW -5 ; -5 = 1111111111111011b

Beta DW 1234h ; 1234h = 0001001000110100b

XYZ DW ?

### ◦ Khoảng xác định của biến Word:

- Số không dấu: [0, 65535]
- Số có dấu: [-32768, 32767]



# Giới thiệu lập trình hợp ngữ

## ■ Biến mảng:

### ◦ Mảng Byte:

MangB DB 10h, 20h, 30h, 40h

Buffer DB 100 dup (?)

### ◦ Mảng Word:

MangW DW -12, 127, 0A48Bh

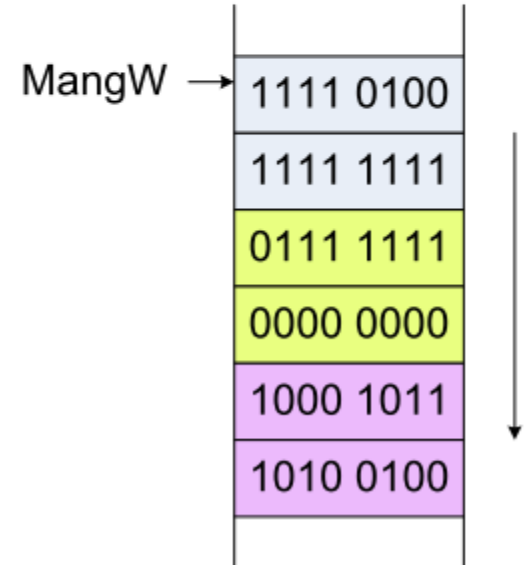
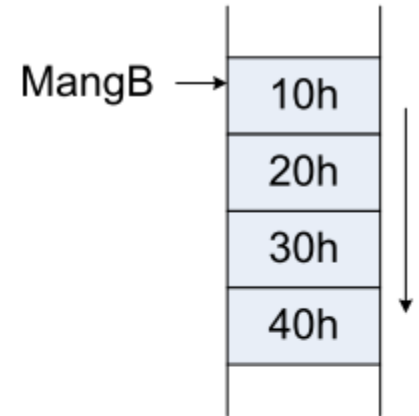
### ◦ Mảng kí tự:

▪ Thực chất là mảng Byte

▪ Ví dụ: 2 cách viết sau là tương đương

M DB 'ABC'

M DB 41h, 42h, 43h



# Giới thiệu lập trình hợp ngữ

- Chương trình mã máy khi được thực thi sẽ chiếm 3 vùng nhớ cơ bản trong bộ nhớ chính:
  - Vùng nhớ lệnh (Code)
  - Vùng dữ liệu (Data)
  - Vùng ngăn xếp (Stack)
- Chương trình hợp ngữ cũng được tổ chức tương tự như vậy.



# Giới thiệu lập trình hợp ngữ

## Cấu trúc chương trình hợp ngữ:

.Model      Small

.Stack100h

.Data

; khai báo biến, hằng ở đây

.Code

Main Proc

; các lệnh của chương trình chính ở đây

Main EndP

; các chương trình con khác ở đây

End Main

# Giới thiệu lập trình hợp ngữ

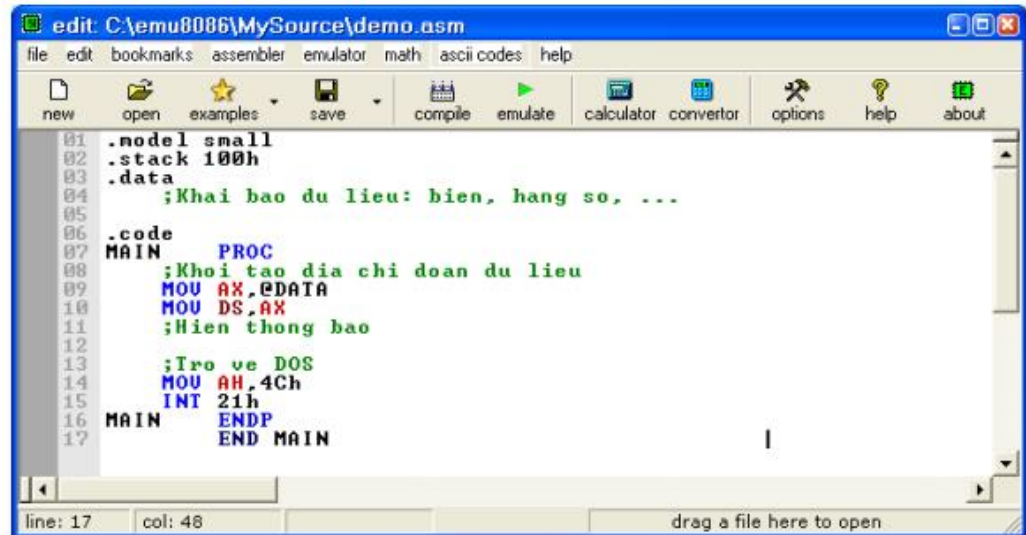
```
.Model      Small
.Stack     100h
.Data
    ; khai báo biến và hằng ở đây
.Code
Main Proc
    mov     ax, @Data
    mov     ds, ax      ; khởi tạo DS trỏ đến đoạn Data
    ; mov     es, ax      ; bỏ dấu ; để khởi tạo ES = DS

    ; thân chương trình

    mov     ah, 4Ch      ; hàm thoát về DOS
    int     21h
Main EndP
End Main
```

# Giới thiệu lập trình hợp ngữ

- Cho phép:
  - Viết mã hợp ngữ
  - Mô phỏng/Debug



# Ví dụ 01

Cho đoạn chương trình khai báo dữ liệu như sau:

```
.model small
```

```
.stack 100h
```

```
.data
```

```
mem1 dw 500
```

```
mem2 dw -50
```

```
vec2 db 10, 20, -10, -20, -30, -40
```

```
.code
```

```
main proc
```

```
mov ax,@data
```

```
mov ds,ax
```

```
mov es,ax
```

```
mov bx,1
```

```
mov ax,es:[bx]
```

```
main endp
```

Offset

mem1	→	F4h	0
		01h	1
mem2	→	CEh	2
		FFh	3
vec2	→	0Ah	4
		14h	5
		F6h	6
		ECh	7
		E2h	8
		D8h	9

Xác định nội dung thanh ghi AX sau khi thực hiện đoạn lệnh

a) 01F4

b) 0A14

c) F4FF

d) 14F6

e) CE01

# Ví dụ 02

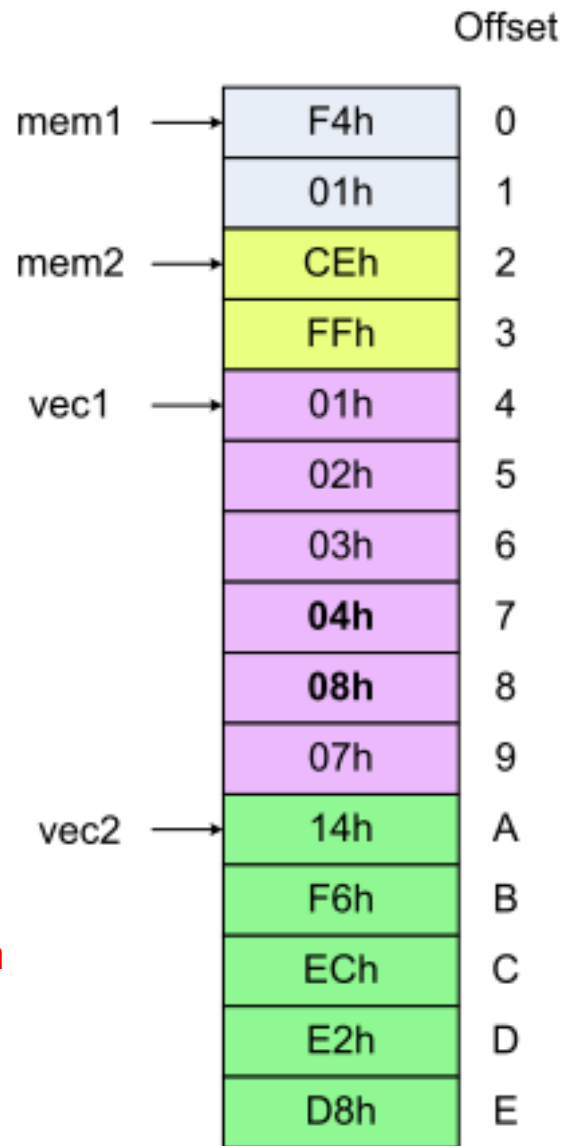
Cho đoạn chương trình khai báo dữ liệu như sau:

```
.model small
.stack 100h
.data
    mem1 dw 500
    mem2 dw -50
    vec1 db 1,2,3,4,8,7
    vec2 db 20,-10,-20,-30,-40
.code
main proc
    mov ax,@data
    mov ds,ax
    mov bx,OFFSET vec2
    mov cx,[bx+2]
main endp
```

1. Xác định nội dung thanh ghi CX sau khi thực hiện đoạn lệnh

2. Kết quả thay đổi thế nào khi khai báo:

vec2 **dw** 20,-10,-20,-30,-40



# Một số bài tập

**Bài 1: Cho đoạn lệnh thực hiện bằng hợp ngữ như sau:**

```
      MOV AX,00H
      MOV CX,08H
K1:   INC AX
      PUSH AX
      LOOP K1
      MOV CX,04H
K2:   POP AX
      MOV BX,AX
      ADD DX,AX
      LOOP K2
      POP DX
      POP CX
      POP AX
```

Xác định nội dung các thanh ghi AX, BX, CX, DX sau khi thực hiện đoạn lệnh trên

- 1) AX: 0005      BX:0005      CX:0000      DX:001A
- 2) AX:

**Bài 2: Cho đoạn lệnh thực hiện bằng hợp ngữ như sau, hãy xác định nội dung các thanh ghi AX, BX, CX, DX sau khi thực hiện đoạn lệnh**

**MOV AX,100(0064H)**

**MOV BX,200(00C8H)**

**MOV CX,300(012C)**

**PUSH AX**

**PUSH BX**

**PUSH CX**

**POP BX**

**POP CX**

**POP DX**

**AND AX,123**

**PUSH AX**

**OR BX,456**

**PUSH BX**

**POP CX**

**POP DX**

**Bài tập:** Cho đoạn chương trình với các lệnh trong bộ nhớ như sau. Hãy mô tả trình tự thực hiện chương trình và nội dung của bộ nhớ ngăn xếp.

LOC:	MACHINE CODE	SOURCE
0100:	E8 04 00	ORG 100h CALL m1
0103:	B8 02 00	MOV AX, 2
0106:	C3	RET
0107:	E8 09 00	m1 PROC CALL m2
010A:	BB 05 00	MOV BX, 5
010D:	E8 03 00	CALL m2
0110:	03 C3	ADD AX, BX
0112:	C3	RET
0113:		m1 ENDP
0113:	BB 02 00	m2 PROC MOV BX, 2
0116:	C3	RET
		m2 ENDP
		END



**Bài tập:** Cho đoạn chương trình với các lệnh trong bộ nhớ như sau. Hãy mô tả trình tự thực hiện chương trình và nội dung của bộ nhớ ngăn xếp.

```
=====
LOC: MACHINE CODE      SOURCE
=====

:                      ORG      100h
:
0100: B0 01             MOV      AL, 1
0102: B3 02             MOV      BL, 2
:
0104: E8 0A 00          CALL     m2
0107: E8 07 00          CALL     m2
010A: E8 04 00          CALL     m2
010D: E8 01 00          CALL     m2
:
0110: C3               RET                      ; return to operating system.
:
0111:                  m2      PROC
0111: F6 E3             MUL      BL              ; AX = AL * BL.
0113: C3               RET                      ; return to caller.
:                  m2      ENDP
:
:                      END
:
=====
```

## Bài tập: Một vùng trong bộ nhớ RAM có nội dung như sau

07 00:01 00	update															<input checked="" type="radio"/> table	<input type="radio"/> list
0700:0100	B0	01	B3	02	E8	0A	00	E8-07	00	E8	04	00	E8	01	00	☒ ☒...☒...☒♦.☒☒.	
0700:0110	C3	F6	E3	C3	90	90	90	90-90	90	90	90	90	90	90	90	÷Π ÉÉÉÉÉÉÉÉÉÉÉÉÉÉ	
0700:0120	90	90	90	90	90	90	90	90-F4	00	00	00	00	00	00	00	ÉÉÉÉÉÉÉÉÉÉÉÉÉÉ	
0700:0130	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	.....	
0700:0140	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	.....	
0700:0150	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	.....	
0700:0160	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	.....	
0700:0170	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	.....	

Xác định nội dung của các thanh ghi AX, BX, CX, DX và bộ nhớ thay đổi thế nào khi thực hiện các lệnh sau:

MOV BX,0111h

MOV AX,[BX]

PUSH AX

AND AX,BX

MOV CX,AX

POP DX

MOV [BX],AX

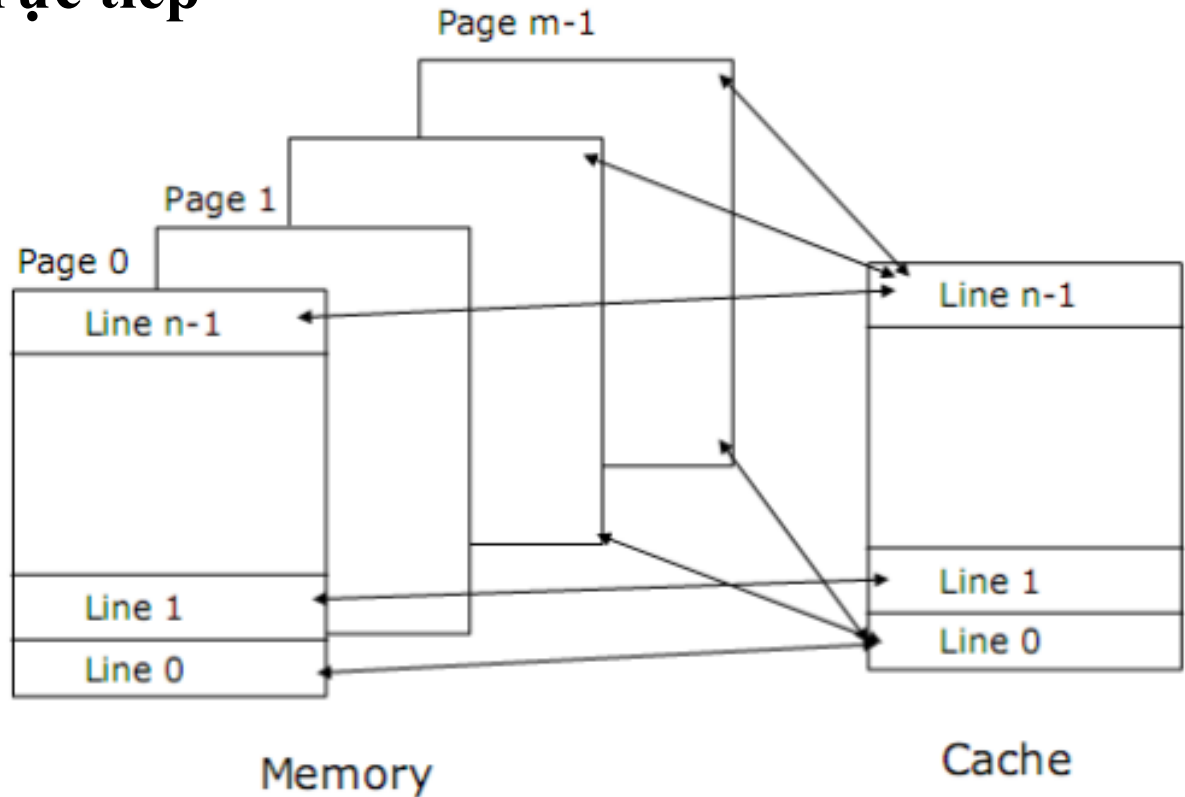
# Bài tập về bộ nhớ

Hệ thống máy tính có bộ nhớ RAM dung lượng 8GB với  $t_{ac}=80\text{ns}$ , bộ nhớ Cache dung lượng 2MB với  $t_{ac}=4\text{ns}$ , kích thước line của bộ nhớ cache là 32bytes. Hãy xác định số bit trong các trường của địa chỉ dữ liệu trong các trường hợp:

- Bộ nhớ cache ánh xạ trực tiếp
- Bộ nhớ cache ánh xạ kết hợp đầy đủ
- Bộ nhớ cache ánh xạ tập kết hợp (với 4 ways)
- Xác định thời gian truy cập trung bình của hệ thống nhớ khi có 1000000 truy xuất, với tỉ lệ cache hit là 96%.

# Bài tập về bộ nhớ

## a) Cache ánh xạ trực tiếp

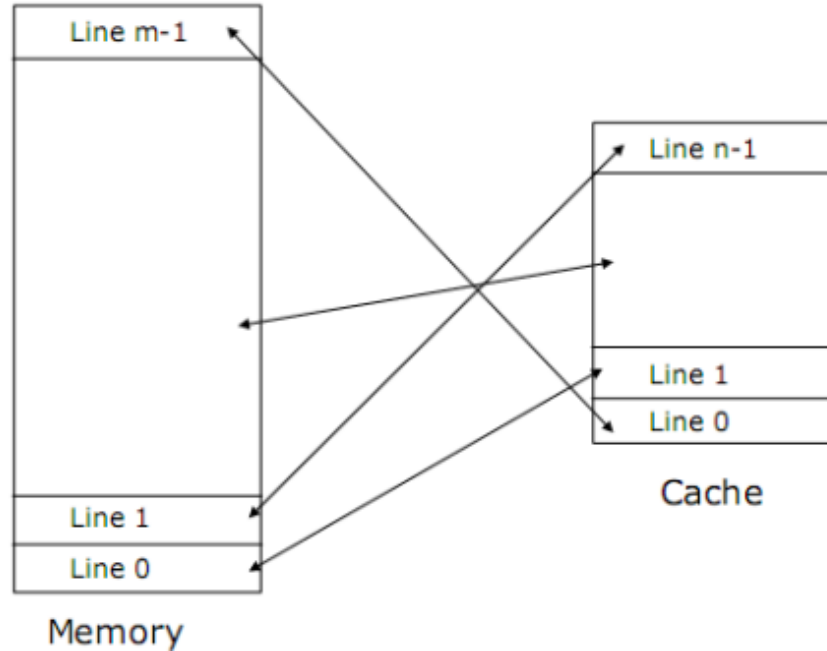


Tag	Line	Word
-----	------	------

- Tag (bit): là địa chỉ của trang trong bộ nhớ
- Line (bit): là địa chỉ của line trong cache
- Word (bit): là địa chỉ của word trong line

# Bài tập về bộ nhớ

## b) Cache ánh xạ kết hợp đầy đủ

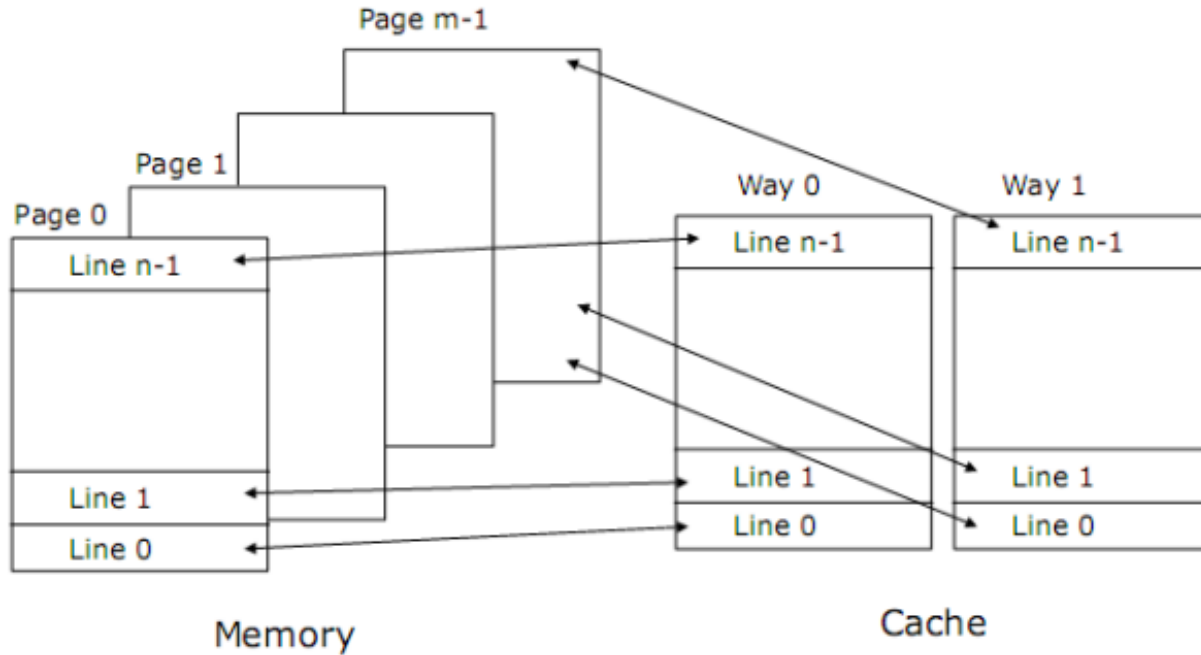


Tag	Word
-----	------

- Tag (bit) là địa chỉ của line trong bộ nhớ (page =1)
- Word (bit) là địa chỉ của từ trong line

# Bài tập về bộ nhớ

## c) Cache ánh xạ tập kết hợp



Tag	Set	Word
-----	-----	------

Tag (bit): là địa chỉ của trang trong bộ nhớ

Set (bit): là địa chỉ của line trong way của cache

Word (bit): là địa chỉ của word trong line

# Bài tập về bộ nhớ

## d) Thời gian truy suất trung bình

$$t_{\text{access}} = t_{\text{cache}} + (1 - H) * (t_{\text{memory}})$$

$$T_{\text{tb}} = [T_1 * \text{hit\_rate} + (T_1 + T_2) * \text{miss\_rate}] / \text{số lần truy cập}$$

$$T_1 = 4\text{ns}; T_2 = 80\text{ns};$$

$$\text{Số truy cập} = 1000000$$

$$H = 0,96$$

# Bộ nhớ ảo

## Bài 1:

Một hệ thống máy tính với bộ nhớ chính có kích thước **320MB**. Hệ thống sử dụng địa chỉ logic **48** bit. Kích thước trang được sử dụng là **8KB**. Yêu cầu xác định các thông số sau:

- Cho biết số bit dùng cho địa chỉ offset.
- Số khung trang vật lý.
- Số trang logic trong không gian tiến trình.
- Cho địa chỉ logic 20030, yêu cầu đổi sang dạng  $\langle p, d \rangle$ .

### *a) Số bit dùng cho địa chỉ offset?*

Là số bit cần dùng để mô tả tất cả các địa chỉ trong một trang

Kích thước một trang:  $8KB = 2^3 \cdot 2^{10}B = 8192B = 2^{13}B \rightarrow$  Số bit cần dùng là **13 bit**.



# Bộ nhớ ảo

## Bài 1:

Một hệ thống máy tính với bộ nhớ chính có kích thước **320MB**. Hệ thống sử dụng địa chỉ logic **48** bit. Kích thước trang được sử dụng là **8KB**. Yêu cầu xác định các thông số sau:

- Cho biết số bit dùng cho địa chỉ offset.
- Số khung trang vật lý.
- Số trang logic trong không gian tiến trình.
- Cho địa chỉ logic 20030, yêu cầu đổi sang dạng  $\langle p, d \rangle$ .

### *b) Số khung trang vật lý*

$$\frac{\text{Kích thước bộ nhớ vật lý}}{\text{Kích thước trang}} = \frac{320 \text{ MB}}{8 \text{ KB/trang}} = \frac{320 * 2^{20} \text{ B}}{8 * 2^{10} \text{ B/trang}} = 40 * 2^{10} = 40960 \text{ trang}$$

# Bộ nhớ ảo

## Bài 1:

Một hệ thống máy tính với bộ nhớ chính có kích thước **320MB**. Hệ thống sử dụng địa chỉ logic **48** bit. Kích thước trang được sử dụng là **8KB**. Yêu cầu xác định các thông số sau:

- Cho biết số bit dùng cho địa chỉ offset.
- Số khung trang vật lý.
- Số trang logic trong không gian tiến trình.
- Cho địa chỉ logic 20030, yêu cầu đổi sang dạng  $\langle p, d \rangle$ .

c) *Số trang logic trong không gian tiến trình*

$$\frac{\text{Kích thước không gian tiến trình}}{\text{Kích thước trang}} = \frac{2^{48} \text{ B}}{8 \text{ KB/trang}} = \frac{2^{48} \text{ B}}{2^{13} \text{ B/trang}} = 2^{35} \text{ trang}$$

# Bộ nhớ ảo

## Bài 1:

Một hệ thống máy tính với bộ nhớ chính có kích thước **320MB**. Hệ thống sử dụng địa chỉ logic **48** bit. Kích thước trang được sử dụng là **8KB**. Yêu cầu xác định các thông số sau:

- Cho biết số bit dùng cho địa chỉ offset.
- Số khung trang vật lý.
- Số trang logic trong không gian tiến trình.
- Cho địa chỉ logic 20030, yêu cầu đổi sang dạng  $\langle p, d \rangle$ .

*d) Đổi địa chỉ 20030 sang dạng  $\langle p, d \rangle$*

Do kích thước trang là 8192, lấy 20030 chia cho 8192 được 2 dư 3646.

20030 được đổi thành  $\langle p = 2, d = 3646 \rangle$ .

# Bộ nhớ ảo

## Bài 2:

Một máy tính sử dụng địa chỉ logic 64bit có dung lượng bộ nhớ 64MB. Hệ điều hành sử dụng 12 bit để làm địa chỉ offset. Yêu cầu tính:

- a) Số trang logic
- b) Số trang vật lý
- c) Kích thước trang.

# Bộ nhớ ảo

## Bài 3:

Một hệ thống máy tính với bộ nhớ chính có kích thước 800MB. Hệ thống sử dụng địa chỉ logic 32 bit. Kích thước trang được sử dụng là 16KB. Yêu cầu xác định các thông số sau:

- a) Cho biết số bit dùng cho địa chỉ offset.
- b) Số khung trang vật lý.
- c) Số trang logic trong không gian tiến trình.
- d) Cho địa chỉ logic 203030, yêu cầu đổi sang dạng  $\langle p, d \rangle$ .

# Bộ nhớ ảo

## Bài 4:

Cho một hệ thống máy tính sử dụng bộ nhớ ảo với cơ chế toàn cục (nghĩa là khi chọn trang nạn nhân, hệ thống có thể chọn trang của một tiến trình khác). Hệ thống có 3 khung trang, kích thước của mỗi trang là 1024 bytes. Hệ thống sử dụng địa chỉ logic 16 bit. :

- a) Cho biết số bit dùng cho địa chỉ offset và số trang logic tối đa trong không gian tiến trình
- b) Cho địa chỉ logic 263168, yêu cầu đổi sang dạng  $\langle p, d \rangle$ , nêu nhận xét về tính hợp lệ của địa chỉ logic trên.

# Bộ nhớ ảo

## Bài 5:

Cho một hệ thống máy tính sử dụng bộ nhớ ảo với cơ chế toàn cục (nghĩa là khi chọn trang nạn nhân, hệ thống có thể chọn trang của một tiến trình khác). Hệ thống có 3 khung trang, kích thước của mỗi trang là 1024 bytes. Hệ thống sử dụng địa chỉ logic 16 bit:

- a) Cho biết số bit dùng cho địa chỉ offset và số trang logic tối đa trong không gian tiến trình
- b) Cho địa chỉ logic 263168, yêu cầu đổi sang dạng  $\langle p, d \rangle$ , nêu nhận xét về tính hợp lệ của địa chỉ logic trên.

# Bộ nhớ ảo

## Bài 6:

Giả sử trong quá trình quản lý bộ nhớ ảo dạng phân đoạn, HĐH duy trì bảng phân đoạn (Segment Table) như sau:

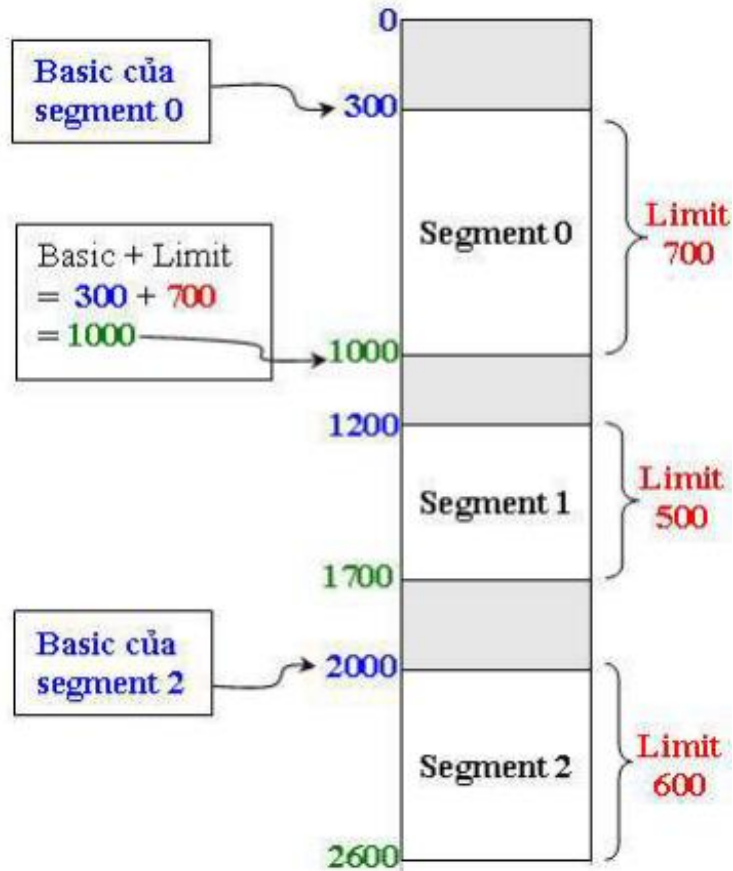
Segment	Base	Limit
0	300	700
1	1200	500
2	2000	600

- Hãy tính địa chỉ vật lý cho mỗi địa chỉ logic sau: (1,200), (1,0), (0,700), (2,0), (2,600)

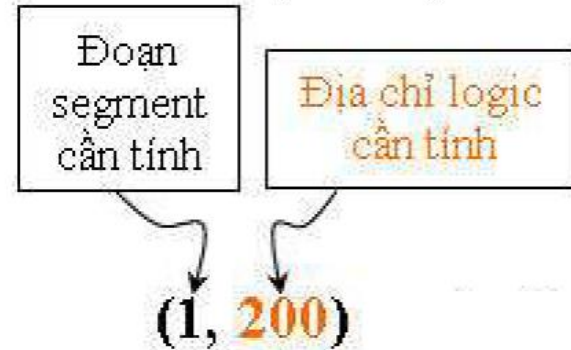


# Bộ nhớ ảo

## 1) Vẽ vùng bộ nhớ vật lý dạng các đoạn segment:



## 2. Cách tính địa chỉ logic



====>  $(1, 200) = 1200 + 200 = 1400$  (hợp lệ vì  $< 1700$ )  
+  $(1, 0) = 1200 + 0 = 1200$  (hợp lệ)  
+  $(0, 700) = 300 + 700 = 1000$  (hợp lệ)  
+  $(2, 0) = 2000 + 0 = 2000$  (hợp lệ)  
+  $(2, 600) = 2000 + 600 = 2600$  (hợp lệ)

# Bộ nhớ ảo

## Bài 7:

Giả sử có bảng đoạn sau:

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

Hãy tính địa chỉ vật lý cho mỗi địa chỉ logic sau:

- a) (0,430)
- b) (1,010)
- c) (2,500)
- d) (3,400)
- e) (4,112)

# Bộ nhớ ảo

## Bài 8:

Xét bảng phân đoạn sau đây:

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

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

- a) 0, 430
- b) 1, 10
- c) 2, 500
- d) 3, 400
- e) 4, 112