

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



## KIẾN TRÚC MÁY TÍNH

---

ĐỀ: 03

# Bài tập cá nhân

---

GVHD: Nguyễn Xuân Minh  
SV thực hiện: Phạm Công Võ – 2313946

Tp. Hồ Chí Minh, Tháng 05/2025



## Mục lục

<b>1</b>	<b>Đề Bài</b>	<b>2</b>
1.1	Đề 3: . . . . .	2
<b>2</b>	<b>Bài làm</b>	<b>2</b>
2.1	Dùng bộ nhớ cache Direct-mapped có 32 block, mỗi block chứa 1 word. . . . .	2
2.2	Dùng bộ nhớ cache Direct-mapped có 16 block, mỗi block chứa 2 word . . . . .	6
2.3	Tổng số bit bộ nhớ cần dùng để xây dựng bộ nhớ cache . . . . .	8
2.3.1	Direct-mapped 32 block, mỗi block chứa 1 word . . . . .	8
2.3.2	Dùng bộ nhớ cache Direct-mapped có 16 block, mỗi block chứa 2 word . . . . .	8



# 1 Đề Bài

## 1.1 Đề 3:

**Câu 1:** Cho biết khi lấy ngẫu nhiên một điểm trong hình vuông có cạnh là 1, xác suất để điểm đó nằm trong hình tròn nội tiếp hình vuông là  $\pi/4$  (diện tích hình tròn chia diện tích hình vuông). Áp dụng vào hệ tọa độ Đề Cát trong khoảng ( $0 < x < 1$  và  $0 < y < 1$ ), viết chương trình MARS MIPS dùng chức năng set seed (syscall 40) theo time (syscall 30) và các chức năng phát số ngẫu nhiên để phát ra tọa độ số thực (x,y) ( $0 < x < 1$ ,  $0 < y < 1$ ) của 50000 điểm dùng để xác định số PI theo gợi ý trên (dùng hình tròn nội tiếp bán kính 0.5 hoặc  $\frac{1}{4}$  hình tròn bán kính 1 đều được). Lưu kết quả chạy chương trình lên tập tin PI.TXT gồm các thông tin như sau:

**Số điểm nằm trong hình tròn: dddd/50000**

**Số PI tính được: f.fffff**

**Câu 2:** Cho danh sách địa chỉ 32-bit truy xuất theo địa chỉ word như sau:

5, 172, 43, 37, 253, 88, 173, 5, 183, 44, 186, 252

a) Nếu dùng bộ nhớ cache Direct-mapped có 32 block, mỗi block chứa 1 word. Hãy xác định địa chỉ theo bit, từ đó suy ra các vùng tag, index lưu trữ vào cache. Cho biết trạng thái Hit/Miss của chuỗi truy xuất trên.

b) Làm lại câu a) với bộ nhớ cache Direct-mapped có 16 block, mỗi block chứa 2 word.

c) Hãy xác định tổng số bit bộ nhớ cần dùng để xây dựng bộ nhớ cache trong cả 2 trường hợp. Biết rằng 1 phần tử cache sẽ chứa 1 bit V, các bit tag và dữ liệu.

# 2 Bài làm

## 2.1 Dùng bộ nhớ cache Direct-mapped có 32 block, mỗi block chứa 1 word.

- Xác định vùng tag, index:

Cache có 32 block. Suy ra index = 5 bit.

Mỗi block chứa 1 word = 4 byte. Suy ra byteoffset = 2 bit.

Đường địa chỉ 32 bit. Suy ra tag =  $32 - 5 - 2 = 25$  bit.

- Trạng thái Hit/Miss của chuỗi truy xuất:

Do truy xuất theo địa chỉ word nên 2 bits offset ở cuối được xem như ẩn đi, không cần xét trong phần này. Ta sử dụng đường địa chỉ 32 bit. Nhưng giá trị của tất cả địa chỉ trong dãy địa chỉ trên nhỏ hơn 256 (Theo hệ 10), nên chỉ có 8 bit tiếp theo của 2 bit offset có giá trị khác 0. Do đó, ta chỉ quan tâm đến 8 bit đó.

Địa chỉ khối	Địa chỉ khối nhị phân(8 bit)	Tag	Index	Index(10)
5	00000101	000	00101	5
172	10101100	101	01100	12
43	00101011	001	01011	11
37	00100101	001	00101	5
253	11111101	111	11101	29
88	01011000	010	11000	24
173	10101101	101	01101	13
5	00000101	000	00101	5
183	10110111	101	10111	23
44	00101100	001	01100	12
186	10111010	101	11010	26
252	11111100	111	11100	28



Để đơn giản cho việc quan sát, chúng ta chỉ hiện thị các index đã có dữ liệu, ở những ô chưa có ta tạm thời ẩn đi. Đầu tiên, truy xuất tại 5.

Địa chỉ khối	Tag	Cache block	Hit/Miss
5	000	00101	Miss

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	000	Mem[00000101]

Tiếp theo, truy xuất tại 172.

Địa chỉ khối	Tag	Cache block	Hit/Miss
172	101	01100	Miss

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	000	Mem[00000101]
12	01100	Y	101	Mem[10101100]

Tiếp theo, truy xuất tại 43.

Địa chỉ khối	Tag	Cache block	Hit/Miss
43	001	01011	Miss

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	000	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	101	Mem[10101100]

Tiếp theo, truy xuất tại 37.

Địa chỉ khối	Tag	Cache block	Hit/Miss
37	001	00101	Miss

Trong cache tại Index = 00101 đã có dữ liệu với Tag = 000 nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	001	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	101	Mem[10101100]

Tiếp theo, truy xuất tại 253.

Địa chỉ khối	Tag	Cache block	Hit/Miss
253	111	11101	Miss

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:



	Index	Valid	Tag	Data
5	00101	Y	001	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	101	Mem[10101100]

Tiếp theo, truy xuất tại 90.

Địa chỉ khối	Tag	Cache block	Hit/Miss
90	010	11010	Miss

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	001	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	101	Mem[10101100]
29	11101	Y	111	Mem[11111101]

Tiếp theo, truy xuất tại 88.

Địa chỉ khối	Tag	Cache block	Hit/Miss
88	010	11000	Miss

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	001	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	101	Mem[10101100]
24	11000	Y	010	Mem[01011000]
29	11101	Y	111	Mem[11111101]

Tiếp theo, truy xuất tại 173.

Địa chỉ khối	Tag	Cache block	Hit/Miss
173	101	01101	Miss

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	001	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	101	Mem[10101100]
13	01101	Y	101	Mem[10101101]
24	11000	Y	010	Mem[01011000]
29	11101	Y	111	Mem[11111101]

Tiếp theo, truy xuất tại 5.

Trong cache tại Index = 00101 đã có dữ liệu với Tag = 001 nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:



	Index	Valid	Tag	Data
5	00101	Y	000	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	101	Mem[10101100]
13	01101	Y	101	Mem[10101101]
24	11000	Y	010	Mem[01011000]
29	11101	Y	111	Mem[11111101]

Tiếp theo, truy xuất tại 183. Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	000	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	101	Mem[10101100]
13	01101	Y	101	Mem[10101101]
23	10111	Y	101	Mem[10110111]
24	11000	Y	010	Mem[01011000]
29	11101	Y	111	Mem[11111101]

Tiếp theo, truy xuất tại 44.

Trong cache tại Index = 01100 đã có dữ liệu với Tag = 101 nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	000	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	001	Mem[00101100]
13	01101	Y	101	Mem[10101101]
23	10111	Y	101	Mem[10110111]
24	11000	Y	010	Mem[01011000]
29	11101	Y	111	Mem[11111101]

Tiếp theo, truy xuất tại 186. Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
5	00101	Y	000	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	001	Mem[00101100]
13	01101	Y	101	Mem[10101101]
23	10111	Y	101	Mem[10110111]
24	11000	Y	010	Mem[01011000]
26	11010	Y	101	Mem[10111010]
29	11101	Y	111	Mem[11111101]

Tiếp theo, truy xuất tại 252. Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:



	Index	Valid	Tag	Data
5	00101	Y	000	Mem[00000101]
11	01011	Y	001	Mem[00101011]
12	01100	Y	001	Mem[00101100]
13	01101	Y	101	Mem[10101101]
23	10111	Y	101	Mem[10110111]
24	11000	Y	010	Mem[01011000]
26	11010	Y	101	Mem[10111010]
28	11100	Y	111	Mem[11111100]
29	11101	Y	111	Mem[11111101]

Trạng thái Hit/Miss của chuỗi truy xuất trên: M M M M M M M M M M M M

## 2.2 Dùng bộ nhớ cache Direct-mapped có 16 block, mỗi block chứa 2 word

Với bộ nhớ cache Direct-mapped có 16 block, mỗi block chứa 2 word, ta cần xác định các vùng tag, index và offset của cache. Mỗi block có 2 word và mỗi word có 4 byte, vì vậy ta cần 3 bit để đánh địa chỉ offset. Với bộ nhớ cache Direct-mapped có 16 block, số bit index sẽ là  $\log_2(16) = 4$  bits. Số bit tag sẽ là  $32 - 4 - 3 = 25$  bits. Ta sử dụng đường địa chỉ 32 bit. Nhưng giá trị của tất cả địa chỉ trong dãy địa chỉ trên nhỏ hơn 256 (Theo hệ 10), nên chỉ có 8 bit tiếp theo của 2 bit offset có giá trị khác 0. Do đó, ta chỉ quan tâm đến 8 bit đó. Do truy xuất theo địa chỉ word nên 2 bits offset ở cuối được xem như ẩn đi, không cần xét trong phần này. Do 1 block có 2 word nên ta phải tính Block address.

Word address	Block address	Block address(Binary address)	Tag	Cache block	Cache block(10)
5	2	0000010	000	0010	2
172	86	1010110	101	0110	6
43	21	0010101	001	0101	5
37	18	0010010	001	0010	2
253	126	1111110	111	1110	14
88	44	0101100	010	1100	12
173	86	1010110	101	0110	6
5	2	0000010	000	0010	2
183	91	1011011	101	1011	11
44	22	0010110	001	0110	6
186	93	1011101	101	1101	13
252	126	1111110	111	1110	14

Ta sử dụng đường địa chỉ 32 bit. Nhưng giá trị của tất cả Block address nhỏ hơn 128 (Theo hệ 10), nên chỉ có 7 bit tiếp theo của 3 bit offset có giá trị khác 0. Do đó, ta chỉ quan tâm đến 7 bit đó.

Đầu tiên, truy xuất tại Block address 2.(5)

Địa chỉ khối	Tag	Cache block	Hit/Miss
2	000	0010	Miss

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:



	Index	Valid	Tag	Data
0	0000	N		
1	0001	N		
2	0010	Y	000	Mem[0000010]
3	0011	N		
4	0100	N		
5	0101	N		
6	0110	N		
7	0111	N		
8	1000	N		
9	1001	N		
10	1010	N		
11	1011	N		
12	1100	N		
13	1101	N		
14	1110	N		
15	1111	N		

Truy xuất tại Block address 86(172)

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:

	Index	Valid	Tag	Data
0	0000	N		
1	0001	N		
2	0010	Y	000	Mem[0000010]
3	0011	N		
4	0100	N		
5	0101	N		
6	0110	Y	101	Mem[1010110]
7	0111	N		
8	1000	N		
9	1001	N		
10	1010	N		
11	1011	N		
12	1100	N		
13	1101	N		
14	1110	N		
15	1111	N		

Truy xuất tại Block address 21(43)

Địa chỉ này truy xuất lần đầu và chưa có trong cache nên ghi nhận là miss. Bộ nhớ cache cập nhật như sau:





	Index	Valid	Tag	Data
0	0000	N		
1	0001	N		
2	0010	Y	000	Mem[0000010]
3	0011	N		
4	0100	N		
5	0101	Y	001	Mem[0010101]
6	0110	Y	101	Mem[1010110]
7	0111	N		
8	1000	N		
9	1001	N		
10	1010	N		
11	1011	N		
12	1100	N		
13	1101	N		
14	1110	N		
15	1111	N		

Làm tương tự cho đến hết ta được chuỗi:

M M M M M M H M M M M H

## 2.3 Tổng số bit bộ nhớ cần dùng để xây dựng bộ nhớ cache

Tổng số bit bộ nhớ cần dùng =  $(1 + \text{tag} + \text{data}) * \text{số block}$

### 2.3.1 Direct-mapped 32 block, mỗi block chứa 1 word

- Số bit =  $(1 + 25 + 32) * 32 = 1856$  (bit)

### 2.3.2 Dùng bộ nhớ cache Direct-mapped có 16 block, mỗi block chứa 2 word

- Số bit =  $(1 + 25 + 2*32) * 16 = 1440$  (bit)