

VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY  
UNIVERSITY OF TECHNOLOGY  
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



**Kiến Trúc Máy Tính (CO2008)**

---

**BÀI TẬP CÁ NHÂN**

---

Advisor: Nguyễn Xuân Minh  
Student: Trần Phước Nhật  
Student ID: 2212412

HO CHI MINH CITY, DECEMBER 2023

## Báo cáo bài tập lớn cá nhân

1. Đề bài:  $((2212412 + 2) \% 4) + 1 = 3$

### Đề số 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  $\pi/4$ . 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 100000 số ngẫu nhiên  $x$  ( $0 < x < 1$ ) dùng để xác định và in ra số PI theo gợi ý trên. Lưu kết quả chạy chương trình lên tập tin PI.TXT gồm các thông tin như sau:

So diem nam trong hình tron: ddddd

So PI tinh duoc: f.ffffff

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.

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

### Câu 2:

#### a) Direct-mapped Cache với 32 Blocks

Direct-mapped Cache với 32 Blocks, mỗi block chứa 1 word (4 bytes), chúng ta có thể chia 32-bit địa chỉ thành những phần như sau:

- Index:  $\log_2(32) = 5$  bits
- Tag:  $32 - 5 = 27$  bits

Chúng ta có thể biểu diễn bits như trên để nhận định vị trí của một word trong bộ nhớ cache. Trạng thái Hit/Miss sẽ phụ thuộc vào việc một word đã được lưu vào bên trong cache từ trước hay chưa. Với Direct-mapped Cache, ta có bảng trạng thái của dãy 5, 172, 43, 37, 253, 88, 173, 5, 183, 44, 186, 252 như sau:

Word Address	Bits	Tag	Index	Trạng Thái
5	00000000000000000000000000000000_00101	0	5	Miss
172	00000000000000000000000000000101_01100	5	12	Miss
43	00000000000000000000000000000001_01011	1	11	Miss
37	00000000000000000000000000000001_00101	1	5	Miss
253	00000000000000000000000000000111_11101	7	29	Miss
88	00000000000000000000000000000010_11000	2	24	Miss
173	00000000000000000000000000000101_01101	5	13	Miss
5	00000000000000000000000000000000_00101	0	5	Miss
183	00000000000000000000000000000101_10111	5	23	Miss
44	00000000000000000000000000000001_01100	1	12	Miss
186	00000000000000000000000000000101_11010	5	26	Miss
252	00000000000000000000000000000111_11100	7	28	Miss

### b. Direct-mapped Cache với 16 Blocks

Direct-mapped Cache với 16 Blocks, mỗi block chứa 2 word (8 bytes), chúng ta có thể chia 32-bit địa chỉ thành những phần như sau:

- Index:  $\log_2(16) = 4$  bits
- Tag:  $32 - 4 = 28$  bits

Tương tự như câu a) nhưng với kích thước block là 8 bytes, tổng số block là 16. Ta có bảng trạng thái của dãy 5, 172, 43, 37, 253, 88, 173, 5, 183, 44, 186, 252 như sau:

Word Address	Bits	Tag	Index	Trạng Thái
5	00000000000000000000000000000000_0101	0	5	Miss
172	000000000000000000000000000000001010_1100	10	12	Miss
43	0000000000000000000000000000000010_1011	2	11	Miss
37	0000000000000000000000000000000010_0101	2	5	Miss
253	000000000000000000000000000000001111_1101	15	13	Miss
88	00000000000000000000000000000000101_1000	5	8	Miss
173	000000000000000000000000000000001010_1101	10	13	Miss
5	00000000000000000000000000000000_0101	0	5	Hit
183	000000000000000000000000000000001011_0111	11	7	Miss
44	0000000000000000000000000000000010_1100	2	12	Miss
186	000000000000000000000000000000001011_1010	11	10	Miss
252	000000000000000000000000000000001111_1100	15	12	Miss

### c. Tổng số bits bộ nhớ cần dùng để xây dựng bộ nhớ cache trong cả 2 trường hợp

Để xác định được tổng số bits bộ nhớ cần dùng để xây dựng nên bộ nhớ cache, chúng ta cần phải tính toán được số bits cần cho mỗi phần tử của cache (**bao gồm 1 bit V, các bits tag và dữ liệu**). Ta có 1 byte = 8 bits, chúng ta có thể tính tổng số bit như sau:

Tổng số bit cần = Số block trong cache \* (1 bit V + bits tag + bits dữ liệu)

- Trường hợp a)  $32 \times (1 + 7 + 4 \times 8) = 1280$  bits
- Trường hợp b)  $32 \times (1 + 15 + 8 \times 8) = 2560$  bits

Vậy, ở trường hợp a) ta cần 1280 bits và ở trường hợp b) ta cần 2560 bits để xây dựng bộ nhớ cache.