2018113003

Q8)

→ Code:

```
int x[4][128];
int i;
int sum = 0;

for (i=0; i<128; ++i) {
    sum += x[0][i] * x[1][i] * x[2][i] * x[3][i];
}
```

sizeof(int) = 4

Array x begin at memory address 0x0

& is stored in row-major order.

Cache is initially empty

Memory accesses are only to x, everything else in register.

Size of (x) = $4 * 128 *$ size of (int)

$= (4)^2 (128)$

$= 2^4 (2^7) = 2^{11} = 2048$

a)

→ Cache block size = 16 bytes

Cache lines = $\frac{512}{16} = 32$

∴ Index 0 → 0-15, 128-143, 256-271, 384-399

Index 1 → 16-31, 144-160, ...

⋮

∴ Memory access patern

0, 128, 256, 384, 1, 129, ..

∴ 128 → cold misses

384 → conflict misses

∴ 50 cache misses

∴ Miss rate 100%

b) If cache is fully associative, then there are no conflict misses, only capacity ones

Address in miss : 0, 128, 256, 384, ..

Hits : 1-15, 129-143, ..

⇒ There are 4 parallel channels, getting queried

∴ 4-way set associative cache is ideal

⇒ Optimal organization is 4-way parrallel.

c) :

→ In column form, equivalent misses will be:

Address access : 0, 1, .. , 51,

∴ Cache misses : 1/16 = 6.25%

→ First is cold miss

⇒ For 32 bit databus, miss rate is 25% (1/4) → Only 3 ints cached, when asked for 4