

## ◆◆ Problem – week 17 (02-06-2023)

Name 1 \_\_\_\_\_

This is another problem that tests your ability to analyze the cache behavior of C code. Assume we execute the three summation Functions copyA, copyB y copyC in Figure 1 under the following conditions:

- The machine has a 32-bytes cache with a 16-byte block size. 512 DRAM
- Within the two loops, the code uses memory accesses only for the array data a y b.
- The loop indices and the value sum are held in registers.
- Array a is stored starting at memory address SRAM 0x108.
- $t_{\text{main}} = 30\text{ns}$   $t_{\text{cache}} = 5\text{ns}$

Fill in the table for the approximate **cache miss rate** for the cases  $N = 4$ .

Assume two machine.

Case 1 a machine with direct Mapped cache

Case 2:a machine with Two-Way Set-Associative Cache

N=4	Case 1: Miss Rate Direct Mapped Cache	Time Access memory	Case 2: Miss Rate Two-Way Set-Associative Cache	Time Access memory
copyA				
copyB				
copyC				

```
int a,b array_t[N][N];

void copyA(a,b)
{
    int i, j;
    short sum = 0;
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++) {
            a[i][j]= b[i][j];
        }
    return;
}

void copyB (a,b)
{
    int i, j;
    int sum = 0;
    for (j = 0; j < N; j++)
        for (i = 0; i < N; i++) {
            a[i][j]= b[i][j];
        }
    return;
}

int copyC(a,b)
{
    int i, j;
    int sum = 0;
    for (j = 0; j < N; j += 2)
        for (i = 0; i < N; i += 2) {
            a[i][j]=      b[i][j]
            a[i + 1][j]=    b[i + 1][j]
            a[i][j + 1]=    b[i][j + 1]=
            a[i + 1][j + 1]= b[i + 1][j + 1];
        }
    return;
}
```

<https://www3.ntu.edu.sg/home/smitha/ParaCache/Paracache/start.html>

- A) draw the conformation of the cache memory in both cases, indicating the sets  
 B) indicate the total accesses and total failures for each of the three functions

i	j	address	A	B	C							address	A	B	C	Data
		0x100										0x190				
		0x104										0x194				
		0x108										0x198				
		0x10C										0x19C				
		0x110										0x1A0				
		0x114										0x1A4				
		0x118										0x1A8				
		0x11C										0x1AC				
		0x120										0x1B0				
		0x124										0x1B4				
		0x128										0x1B8				
		0x12C										0x1BC				
		0x130										0x1C0				
		0x134										0x1C4				
		0x138										0x1C8				
		0x13C										0x1CC				
		0x140										0x1D0				
		0x144										0x1D4				
		0x148										0x1D8				
		0x14C										0x1DC				
		0x150										0x1E0				
		0x154										0x1E4				
		0x158										0x1E8				
		0x15C										0x1EC				
		0x160										0x1F0				
		0x164										0x1F4				
		0x168										0x1F8				
		0x16C										0x1FC				
		0x170										0x200				
		0x174										0x204				
		0x178										0x208				
		0x17C										0x20C				
		0x180										0x210				
		0x184										0x214				
		0x188										0x218				
		0x18C										0x21C				