存储器单元作业

1．Consider a 2 GHz processor with two levels of caches in its memory hierarchy. Loads and stores first check a level 1 cache to determine if the desired word is available. If the word is absent from the cache, a larger secondary cache, called a level two cache, is checked for the same word. Finally, if the word is missing from the level two cache, the data is obtained from main memory.
Here is a table of the cache behavior:

Access Behavior	Hit Rate	Hit time
Level 1 Cache	80 %	4 ns
Level 2 Cache	85 %	40 ns
Main Memory	100%	400 ns

Consider a program where 20 % of the instructions are loads, 15 % are stores, and the average CPI of the other instructions is 2.
(a) What is the average memory access time?
(b) What is the average CPI of the processor?
(c) What is the speedup of this processor configuration over a system that does not have the L2 cache?

2: The Microsoft Xbox
Microsoft recently began selling a new video game console called the Xbox. The system is based on a 733 MHz. .Intel processor that is believed to be a scaled-down Pentium III. You have been hired to help design a competing product. But, before you begin planning this new system, you decide to first review the organization of Xbox architecture.
After some research, you assemble the facts below. While many of the details have not been released to the public *, it is likely that the system will have the specifications given below.

System:

Characteristic	**Values**
Virtual Addresses	32 bits
Physical Addresses	36 bits
Addressing Mode	Byte Addressing
Page Size	4 KB

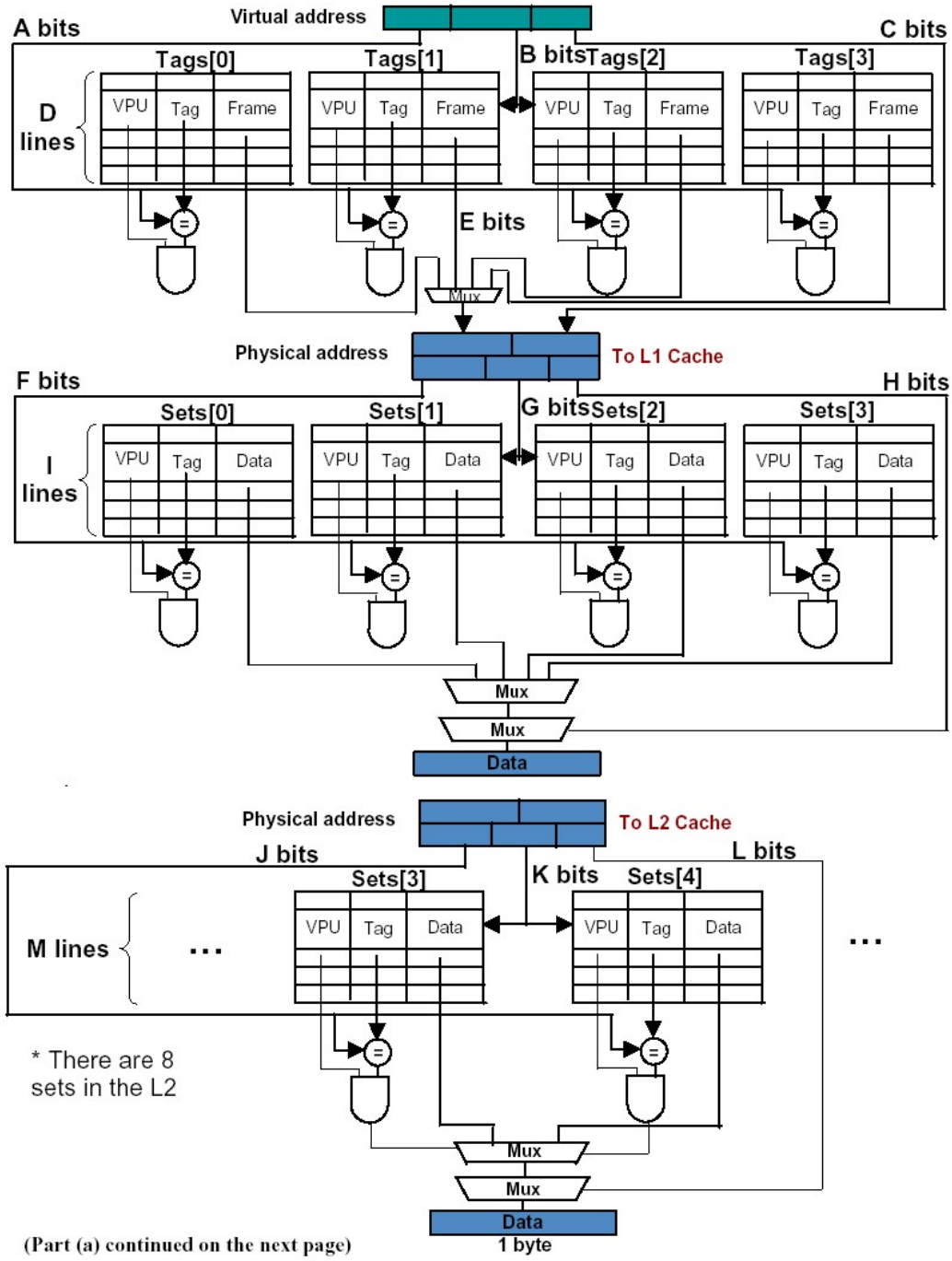
Cache Organization:

Characteristic	**TLB**	**Level1**	**Level2**
Organization	Split I/D	Split I/D	Unified
Cache size	I: 32 Entries D: 64 Entries	I: 16 KB D: 16 KB	128 KB
Cache associativity	4-way Set Associative	4-way Set Associative	8-way Set Associative
Replacement	Pseudo-LRU	Pseudo-LRU	Pseudo-LRU
Block size	N/A	32 Bytes	32 Bytes

Write policy	N/A	I: Write-back or Write-through (programmable) D: Write-back	Write-through or Write-back (programmable)

* In addition to being a "best guess" at some of these details, this table has simplified some issues for the sake of this problem.

Below is the diagram of the memory organization for data accesses.



(Part (a) continued on the next page)

(a) For the diagram illustrating a **data** access on the previous page, fill in the width in bits or number of entries, as appropriate, for each of the letters A through M shown.

A _____B _____C _____D _____E _____F _____

G _____H _____I _____J _____K _____L _____

M _____

(b) The number of entries in the data TLB is greater than the number in the instruction TLB. Aside from die space, give one reason why the designers chose to make the data TLB larger.

(c) In designing your new game embedded processor, you decide to increase the cache block size. Describe **three** potential impacts on performance caused by the increase in block size.

(d) Your first design for this new system included a TLB and the operating system used paging and provided virtual memory. Unfortunately, due to the overhead of virtual memory disk accesses, you later decide to remove the virtual memory support from the OS. One of the other engineers on the team asks whether the TLB should be removed or not given the change to the VM. Give an answer and justify why or why not.

(e) In building the operating system for this new game console, you opt to use an inverted page table. Explain when this decision is sound and justify **quantitatively** why this is an improvement over a simple paging scheme.

3．某磁盘每道有 1024 个扇区，每个扇区存放有 1KB 数据，转速为 14400RPM。该盘在一个磁道上的持续数据速度为多少？

4．某按字节编址的计算机层次存储器系统参数如下：

a) 32 位虚拟地址
b) 32 位实地址
c) 页大小为 2048 字节（Byte）
d) Cache 大小为 64KB，Cache 块大小为 32Byte，采用直接映射方式。

（1）求下列地址字段的位数。

- 虚页号
- 实页号
- Cache 标记字段
- Cache 块号字段
- Cache 块内地址

（2）为简便起见，**假定 Cache 大小为 128 字节**，其组成如下图所示。我们在该计算机上分别运行程序 A 和程序 B。

```
/*    Program A */
int a[100];
  for (i=0; i<4; i++)
    for (j=0; j<2; j++)
      a[i+j*32]++;
```

```
/* Program B */
 int a[100];
 for (j=0; j<2; j++)
   for (i=0; i<4; i++)
     a[i+j*32]++;
```

程序 A 访问存储器的顺序为：load a[0], store a[0], load a[32], store a[32], load a[1], store a[1], …。程序 B 访问存储器的顺序为：load a[0], store a[0], load a[1], store a[1], load a[2], store a[2], …。（提示：整数为 4 个字节长。）

假定 a[0]的地址为 0x420，所有有效位（valid）初始值为 0。请分别填写程序 A 和程序 B 运行结束后下表的内容，包括标记字段和数据（数据不用填具体值，只需

填写 a[？]）。并分别计算命中率。

程序 A 运行结束后的 Cache 状态：

	标记	有效位	w0	w1	w2	w3	w4	w5	w6	w7
0										
1										
2										
3										

程序 B 运行结束后的 Cache 状态：

	标记	有效位	w0	w1	w2	w3	w4	w5	w6	w7
0										
1										
2										
3										