存储器作业

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?

做这个题首先要搞明白这里的Hit Rate的含义，根据层次存储器的原理，L1中的内容在L2中可以找到，L2中的内容在Main Memory中可以找到，从题目可以看出来，对于所有访问的内容，80%可以在查找L1时hit，85%的内容在L2中hit，但L2的这85%的有80%在L1中就可以找到，所以真正会在L2才命中的内容只有5%，同理要在主存中才能命中的占15%。

总结一下，假设现在有一系列的内存地址要访问，那么这些地址的内容有80%可以在L1中找到，5%在L2中找到，剩下的15%需要在主存中找到。

(a)平均的访问时间为：

T=4*80%+44*5%+444*15%

=72ns

因为80%数据需要4ns访问时间，5%的数据要44ns访问时间，15%的数据要444ns的访问时间

(b) 2GHz,则时钟周期为0.5ns，即1ns为2个时钟周期。

load和store同样也需要IF/ID/EX等阶段，故所需时间为其他指令的执行时间(2 CPI)加上读写数据的时间，即72×2+2个周期，故为

((72* 2+2)*(20% + 15%) + 2 * (1 - 20% - 15%)) = 52.4CPI

(c)若没有L2，则a中的结果为T=4*80%+404*20%=84ns

从而平均CPI为：

(84*2+2)*(20%+15%)+2*(1-20%-15%)=60.8CPI

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:

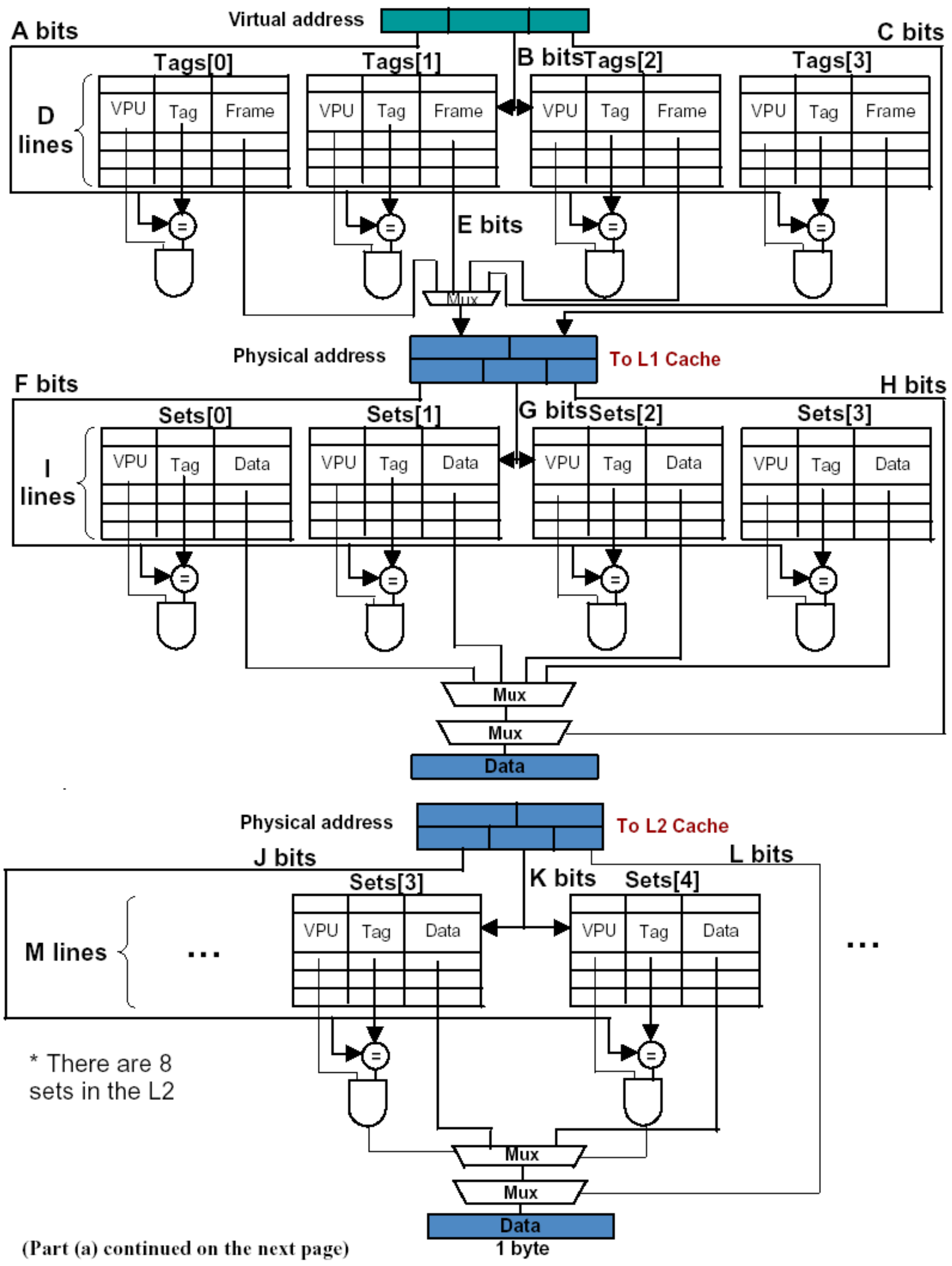
Characteristic	**Values**
Virtual Addresses	32 bits
Physical Addresses	36 bits
Addressing Mode	Byte Addressing
Page Size	4 KB

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.

C	块表页大小4KB，所以要12位页地址，C=12
D	64个单元，4块，所以一块有16个单元，D=16
A	虚拟地址32位，A=32-B-C=16
B	2B= D，所以B=4
E	物理地址36位，C=12，所以E=36-C=24
I	L1大小16KB，块大小32B，I = 16KB/(4*32)B=128
G	2^G=I,G=7
H	块大小32B，2^H=32,H=5
F	F=36-G-H=24
M	M=128K/256B = 512
K	2^K=M,K=9
L	2^L=32,L=5
J	J=36-K-L=22

2),游戏处理中，数据的访问量远远大于指令量，所以数据TLB设置要比指令TLB大，从而能够提高效率。
3),块大小增加后
一次读入的内容更多，在空间连续性意义上，命中率会提高(不过，是有限度的)
发生缺失的时候，由内存到Cache填充时间增加，缺失损失增加
块变大，块中无用数据就会增加，所以Cache利用率下降
2
4),TLB可以保留。拿掉虚存以后TLB还可以用来做32位地址到36位地址的转换。主要原因是32位操作系统都设计好了，市面上的32位的游戏都开发好了，总不能因为去掉了虚存就迫使人家把操作系统推倒重来或者让游戏开发者为了你这款产品重新写一套吧？
5),使用逆向页表可以节省页表空间开销。
假设这款机器的内存大小是S字节，进程数N(假设N<=16，一个游戏没有那么多进程的，实际上现在最热门的游戏都是单进程的，设定这个N只是要了解进程数对页表空间开销的影响)。由于一页大小为4KB，所以： 物理页帧有S/4K个，虚拟页面有2^32/4K = 1M个。
如果使用索引式页表（不使用诸如页目录等等优化手段）： 每一个页表项要放有物理页帧编号，需要36-12 =24位，加上诸如替换、有效等等位，每个入口项需要4字节。一个页表有1M项，所以一个页表占4MB，所有进程的页表加起来共占4N MB内存。
如果使用逆向页表：每一个页表项要存放<虚拟页面编号，进程号>这样的数据，所以

一个页表项就需要(32-12) + log16 = 24bit = 3 bytes；页表项个数与物理页帧相等，为S/4K个，则页表大小为3B*S/4K = S* 0.00075B。
当S * 0.00075B<<4N MB,即S<<N*5.33G(或者N>>S*0.75/4G）的时候，逆向页表能够大大节省页表的空间开销。
作为验证我们可以把S = 1GB代进去，得到N 0:19，也就是说1G内存的话，两个进程左右的时候用逆向页表就能够节省近90%的页表空间开销了，即使只有一个进程也赚了。如果N = 1，那么S<<5.33G，也就是说S大约为512M左右的时候如果只跑1个进程用逆向页表就能够节省90%的页表空间开销了。
3. 某虚拟存储器系统采用页式管理，且仅有一级页表。每页大小为 2P 字节，虚拟地址为 V 位。虚地址组成如下：

虚页号	页内偏移量

页表起始地址为PTBL，页表中每个表项为4个字节长。
请回答下列问题：
1）虚地址中页内偏移量为几位长？
2）虚页号又有几位长？
3）页表中共有多少个表项？页表占用的最大地址是多少？
4）页表本身有多少页？
5）如果要使页表仅占用一页的空间，则P值最小可以是多少？
1)每页大小为2^p，故页内偏移为P位
2）虚地址一共V位，故虚页号为V-P位
3）虚页号为V-P位，故共2^(V-P)个页表项：每个表项占4B，总共4*2^(V-P),所以最大地址为：PTBL+4*2^(V-P)-1
4）页表大小除以每页大小即可，(4*2^(V-P))/2^P = 2^(V-2P+2)
5）要满足2^(V-2P+2)<=1，得$P_{\min} = \left\lceil \frac{V+2}{2} \right\rceil$

4. 某磁盘每道有1024 个扇区，每个扇区存放有4KB 数据，转速为7200RPM。该盘在一个磁道上的持续数据速度为多少？
1024*4KB*7200/60s = 480MB/s