1.The system in question has 1MiB of physical memory, 32-bit virtual addresses, and 256 physical pages. The memory management system uses a fully associative TLB with 128 entries and an LRU replacement scheme.

a. What is the size of the physical pages in bytes?

2^12 bytes

b. What is the size of the virtual pages in bytes?

2^12 bytes

c. What is the maximum number of virtual pages a process can use?

2^20 pages

d. What is the minimum number of bits required for the page table base address register?

记录页表长度和页表地址

20 bits

2.Everybody Got Choices

e. Answer "True!" or "False!" to the following questions

i. The page table is stored in main memory True**!**

ii. Every virtual page is mapped to a physical page False**!**

iii. The TLB is checked before the page table True**!**

iv. The penalty for a page fault is about the same as the penalty for a cache miss False**!**

v. A linear page table takes up more memory as the process uses more memory False**!**

3. Example: Mapping VAs to PAs

Suppose • virtual memory of 2^32 (4G) bytes • physical memory of 2^30 (1G) bytes • page size is 2^14 (16 K) bytes

1). How many pages can be stored in physical memory at once?

$2^{30-14} = 2^{16} = 64K$

2). How many entries are there in the page table?

$2^{32-14} = 2^{18} = 256K$

3). How many bits are necessary per entry in the page table? (Assume each entry has PPN, resident bit, dirty bit)

16 (PPN) + 2 = 18

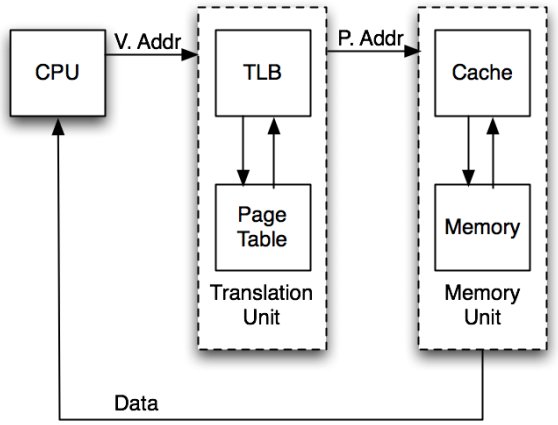
4). A portion of the page table is given to the below. What is the physical address for virtual address 0x00004110?

VPN	R	D	PPN
0	0	0	2
1	1	1	7
2	1	0	0
3	1	0	5
4	0	0	5
5	1	0	3
6	1	1	2
7	1	0	4
8	1	0	1

虚拟页号 1，查表，实页 7

011100 0001 0001 0000 1D110

4. Virtual Addressing



Virtual Address (VA) What your program uses

Virtual Page Number (VPN)	Page Offset

Physical Address (PA) What actually determines where in memory to go

Physical Page Number (PPN)	Page Offset

A processor has 16-bit addresses, 256 byte pages, and an 8-entry fully associative TLB, 16-bit Physical Address. At some time instant, the TLB, Page table, L1 data cache for the current process is given in the table below. Describe 1. 0x0Bf0 2. 0x0C01 **the access process, and final result.**

Access Pattern

1. 0x0Bf0 2. 0x0C01

1. 0x0BF0 VPN=0x0B TLB hit PPN=0x14 PA=0x14F0

0x14F0 0001010011 1100 00 offset=0,index=1100=0C,tag=53,76832135,字节 35

2. 0x0C01

VPN=0xoc, TLB miss page table ppn=0x19 0x1901

0x1901 0001 1001 0000 0001 index=0, tag=64 1256c9ac, 字节 c9