(5 pts) Exercise 7-21

Suppose a cache divides addresses as follows:

	4 bits	3 bits
tag	index	byte offset

Fill in the values for a direct-mapped or 4-way associative cache:

	Direct-mapped	4-way associative
Block size		
Number of blocks		
Total size of cache (e.g. 32 * 128 – don't have to multiply out)		
Tag size (# bits)		

(5 pts) Exercise 7-22

1. **Suppose cache has:**
   - **4 byte blocks**
   - **128 blocks**

   Show how to break the following address into the tag, index, & byte offset.

   0000 1000 0101 1100 0001 0001 0111 1001

2. **Same cache, but now 8-way associative. How does this change things?**

   0000 1000 0101 1100 0001 0001 0111 1001

(5 pts) Exercise 7-23

- **Given a cache that is:**
  - **4-way associative**
  - **32 blocks**
  - **16 byte block size**

  What is the cache index and byte offset for the following address:

  `0x3ab12395`

  Cache index =
  Byte offset =

  And this one:

  `0x70ff1213`

  Cache index =
  Byte offset =

Do these addresses conflict in the cache?

(10 pts) Exercise 7-26

Suppose a 32-bit address is divided up as follows for caching:

6 bits – byte offset
5 bits – index
21 bits – tag

Fill in the following table for the two given types of caches

	Direct-mapped	2-way associative
Block size		
Number of blocks		
Total cache size		

(10 pts) Exercise 7-27

Suppose a direct-mapped cache has 16 byte blocks and a total of 128 blocks (N=128). The machine has 64 bit addresses.

1. **How many address bits are used for the byte offset?**

2. **How many address bits are used for the index?**

3. **How many address bits are used for the tag?**

Now suppose the cache is 4-way set associative. Answer again:

1. **How many address bits are used for the byte offset?**

2. **How many address bits are used for the index?**

3. **How many address bits are used for the tag?**

(5 pts) Exercise 7-31

- **Given system with**
  - **20 bit virtual addresses**
  - **16 bit physical addresses**
  - **256 byte page sizes**
- **How to split a virtual address?**

Virtual page #	Page offset

- **What will the physical address look like?**

Physical page #	Page offset

- **How many entries in the page table?**

(5 pts) Exercise 7-32 (new problem – not related to 7-31)

Translate the following addresses:

1. **B004890**
2. **B002123**
3. **B006001**

Page Table

	Valid?	Physical Page or Disk Block #
B000	1	B004
B001	1	A120
B002	0	AB00
B003	0	8003
B004	1	7590
B005	1	5800
B006	1	F4C0

…

(5 pts) Exercise 7-33

Given the fragment of a page table on the right, answer the following questions assuming a page size of 1024 bytes

1. **What is the virtual address size (# bits)**
2. **What is the physical address size (# bits)**
3. **Number of entries in page table?**

Page Table

	Valid?	Physical Page #
B000	1	B0
B001	1	A0
B002	0	AB
B003	0	80
B004	1	90
B005	1	58
B006	1	F4

…

(5 pts) Exercise 7-36

Suppose we have a virtual memory system with a 32-bit address space and 16 KB pages.

1. **Show how to divide an address into the page offset and the virtual page number.**

2. **How many entries are needed in the page table?**

(10 pts) Exercise 7-37

For a virtual memory system, we usually want…

1. **Direct-mapped OR 4-way associative OR fully associative**

2. **Why?**

3. **Write-back OR Write-through**

4. **Why?**