

**FIGURE 7.20 Mapping from a virtual to a physical address.** The page size is  $2^{12} = 4$  KB. The number of physical pages allowed in memory is  $2^{18}$ , since the physical page number has 18 bits in it. Thus, main memory can have at most 1 GB, while the virtual address space is 4 GB.

3. As we saw in class, page tables require fairly large amounts of memory even when most of the entries are invalid. One way to reduce the memory footprint of page tables is to use a hierarchical page table. This way each process's first-level page table can easily fit in main memory and only the necessary lower-level page tables need to be kept in memory. In such a system an address's virtual page number, as described in the figure above, is further divided into two fields: a "page table number" and a "page table offset". The page table number is used to index the first-level page table which provides the physical address for a second-level page table. The page table offset is then used to index the second-level page table to retrieve the physical page number which is then concatenated with the page offset to provide the physical address. One way to organize such a system is to have each second-level page table occupy exactly one page of memory. Answer the following questions for a system which has a 48-bit virtual address space, 32-bit physical address space and a 4 KiB page size. Recall that page table entries are the next largest power of two in size which can contain all the necessary bits.
  - a. Ignoring the valid and dirty bits, how big is a page table entry in bytes?
  - b. How many bits of the virtual address are used to index the second-level page table (the page table offset)?
  - c. How many bits of the virtual address are used to index the first-level page table (the page table number)?
  - d. How many pages does a first-level page table require?

#### 4. Virtual Memory

Given the following stream of 16-bit virtual addresses, update the TLB and Page Table shown below. Assume a 2 KiB page size, a four-entry fully-associative TLB with true LRU replacement where the least recently used entry is the invalid one followed by the entry with Tag = 11, the entry with Tag = 7 and the entry with Tag = 3 (so the most recently used entry is the entry with Tag = 3). If pages must be brought in from disk, increment to the next highest physical page number (so 13, 14 and so on).

**Addresses: 4095, 31272, 15789, 15000, 7193, 4096, 8912**

**TLB**

Valid	Tag	Physical Page
1	11	12
1	7	4
1	3	6
0	4	9

**Page Table**

Valid	Physical page or in Disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12
0	Disk
1	3
1	7
0	Disk
0	Disk
1	1
0	Disk
1	4
0	Disk
1	8
0	Disk
1	9
1	2
0	Disk

1	10
0	Disk
1	6
0	11
0	Disk
1	12

b. Repeat the question, but this time assume that pages are 8 KiB.

**Addresses: 4095, 31272, 15789, 15000, 7193, 4096, 8912**

**TLB**

<b>Valid</b>	<b>Tag</b>	<b>Physical Page</b>
1	11	12
1	7	4
1	3	6
0	4	9

**Page Table**

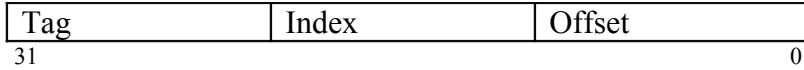
<b>Valid</b>	<b>Physical page or in Disk</b>
1	5
0	Disk
0	Disk
1	6
1	9
1	12
0	Disk
1	4

c. What is an advantage of having a larger page size? What is a disadvantage?

### 5. Virtual vs. Physical

When accessing the data cache, complete the following chart to indicate whether each of the three fields (Tag, Index and Offset as shown below) should come from the Physical or Virtual address and explain why.

Address:



Field	Physical/Virtual	Reason
Tag		
Index		
Offset		