Virtual Memory (VM)

TRANSPARENCY - VM locations “look” the same to program whether on DISK or in RAM

Pagemap Characteristics:
- One entry per virtual page
- RESIDENT bit (R) = 1 for pages stored in RAM
- DIRTY bit says we’ve changed this page since loading from disk (and will need to write back to disk eventually)

Arithmetic
- \(2^p\) bytes per physical page
- \((v+p)\) bits in virtual address
- \((m+p)\) bits in physical address
- \(2^v\) number of virtual pages
- \(2^m\) number of physical pages
- \((m+2)2^v\) bits in page map

When our page map gets too big, we store it in “main memory” (DRAM)
To allow fast VM->PM translation, we cache the most common VM pages indices in the Translation Look-aside Buffer (TLB)
Problem 1
Part G:
The table to the left shows the first 8 entries in the page map. Recall that the valid bit is 1 if the page is resident in physical memory and 0 if the page is on disk or hasn't been allocated.

If there are 1024 ($2^{10}$) bytes per page, what is the physical address corresponding to the decimal virtual address 3956?

<table>
<thead>
<tr>
<th>Virtual page</th>
<th>Valid bit</th>
<th>Physical page</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Problem 2
A particular 32-bit microprocessor includes support for paged virtual memory addressing with $2^{12}$ byte pages. The mapping of virtual to physical addresses requires two translation steps:
1. The most significant 10 bits of the virtual address (the Dir field) are multiplied by 4 and appended to the 20 most significant bits of the dirbase (directory base) register to get the address in main memory of a page directory entry. Each entry in the page directory is a 32-bit record composed of a 20-bit PTBL field and various control bits (Present, Dirty, Read-only, etc.).
2. The bits of the Page field (virtual address bits 21 to 12) are multiplied by 4 and appended to the PTBL field to form the page-table address. This page table address references a 32-bit page table entry. Each page table entry is composed of a 20-bit physical page number (PPN) and a series of control bits.

All page-table entries and the page directory are stored in main memory. The results of these translations are cached in a fully-associative translation look-aside buffer (TLB) with a total of 64 entries, and a LRU replacement strategy is used on TLB misses.