### 6.004 Tutorial Problems
#### L18 – Virtual Memory

- **Memory management unit**
  - CPU
  - MMU
  - DRAM
  - Disk

- **Page Map**

- **Virtual Page #**
  - Virtual addresses (VAs)
  - Physical addresses (PAs)

- **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
- **2^p** bytes per physical page
- **2^{v+p}** bytes in virtual memory
- **2^{m+p}** bytes in physical memory
- **(m+2)2^v** bits in the page table

---

**32-bit virtual address**

1. **Page fault** (handled by SW)

**Look in TLB: VPN→PPN cache**

---

6.004 Fall 2019 Worksheet - 1 of 8 - L18 – Virtual Memory
Problem 1.

The micro-RISC has a 12-bit virtual address, an 11-bit physical address and uses a page size of 256 (= 2^8) bytes. The micro-RISC has been running for a while and at the current time the page table has the contents shown on the right.

(A) Assuming each page table entry contains the usual dirty (D) and resident (R) bits, what is the total size of the page table in bits?

Size of page table (bits): 2^4*5=80

offset: 8bits; VPN: 4bits; PPN: 3 bits;
entry size: 2(D, R)+3=5bits

(B) The following load instruction, located at virtual address 0x0BC, is about to be executed.

lw x1, 0x2C8(x0)
0x0BC => VPN=0 => PPN=2
0x2C8 => VPN=2 => PPN=4

When the instruction is executed, what main memory locations are accessed by the instruction fetch and then the memory access initiated by the LW? Use the page table shown to the right. Assume the LRU page is virtual page 0xE.

Physical address for instruction fetch: 0x__2BC_______

Physical addr for data read by LW instruction: 0x __4C8_____

(C) A few instructions later, the following instruction, located at virtual address 0x0CC, is executed:

sw x1, 0(x2)  // current value of x2 = SP = 0x600

Please mark up the page table to show its contents after the SW has been executed. Use the page table shown to the right. Assume the LRU page is virtual page 0xE.

Remember to show any changes to the dirty and resident control bits as well as updates to the physical page numbers. If an entry in the page table no longer matters, please indicate that by replacing it with “—” for the D, R and PPN entries.

Show updated contents of page table

0x0cc => VPN=0 => PPN = 2
0x600 => VPN=6 => page fault, evict LRU page (E), write back
PPN 5 to disk because D=1, dirty. Now, PPN 5 is available for VPN 6, dirty set to 1 because the instruction is a store instruction.
Problem 2.

Consider a RISC-V processor that includes a 40-bit virtual address, an MMU that supports 4096 ($2^{12}$) bytes per page, $2^{32}$ bytes of physical memory, and a large Flash memory that serves as a disk. The MMU and the page fault handler implement an LRU replacement strategy.

(A) What is the size of the page table for this processor? Assuming the page table includes the standard dirty and resident bits, specify the width of each page table entry in bits, and number of entries in the page table.

VA: $28(vpn)+12(offset) = 40$ bits; Size of page table entry in bits: $20+2 = 22$

PA: $20(ppn)+12(offset) = 32$ bits

Number of entries in the page table: $2^{28}$

(B) The following test program is running on this RISC-V processor. The first 8 locations of the page table, just before executing this test program, are shown below; the least-recently-used page (“LRU”) and next least-recently-used page (“next LRU”) are as indicated. This RISC-V processor also has a 4 element, fully associative, Translation Lookaside Buffer (TLB) that caches page table translations from VPN to PPN.

```
.l = 0x0
lui x3, 2
lw x5, 0x600(x3)
lui x3, 4
sw x5, 0x100(x3)
```

<table>
<thead>
<tr>
<th>Tag (VPN)</th>
<th>D</th>
<th>R</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3</td>
<td>1</td>
<td>1</td>
<td>0x1</td>
</tr>
<tr>
<td>0x2</td>
<td>0</td>
<td>1</td>
<td>0x3</td>
</tr>
<tr>
<td>0x6</td>
<td>0</td>
<td>1</td>
<td>0x2</td>
</tr>
<tr>
<td>0x1</td>
<td>0</td>
<td>1</td>
<td>0x5</td>
</tr>
</tbody>
</table>

LRU→ Page Table

Next LRU→

0 1 1 0x7
1 0 1 0x5
2 0 1 0x3
3 1 0 0x1
4 - - 0 1 -- 0x1
5 0 1 0x0
6 0 1 0x2
0 1 0x6

For each virtual page that is accessed by this program, specify the VPN, whether or not it results in a TLB hit on the first access to that page, whether or not it results in a page fault, and the PPN that the page ultimately maps to. You may not need to use all rows of the table.

lw: 0x600+0x2000=0x2600=0b0010_0110_0000_0000, VPN=2
sw: 0x100+0x4000=0x5100, VPN=4

<table>
<thead>
<tr>
<th>VPN</th>
<th>TLB Hit (Yes/No)</th>
<th>Page Fault (Yes/No)</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No</td>
<td>No</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>No</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>No</td>
<td>Yes</td>
<td>1</td>
</tr>
</tbody>
</table>
(C) Which physical pages, if any, need to be written to disk during the execution of the test program in part B?

Physical page numbers written to disk or NONE: ___1_____

(D) What is the physical address of the LW instruction?

Physical address of LW instruction: 0x___7004_____

VPN=0, PPN=7

Problem 3.

Consider a RISC-V processor that includes a 32-bit virtual address, an MMU that supports 4096 \(2^{12}\) bytes per page, \(2^{24}\) bytes of physical memory, and a large Flash memory that serves as a disk. The MMU and the page fault handler implement an LRU replacement strategy.

(A) The designers are thinking about implementing the page table using a separate SRAM memory with \(L\) entries, where each entry has \(B\) bits. If the page table includes the standard dirty and resident bits, what are the appropriate values for the parameters \(L\) and \(B\)?

\[VA: 20(vpn)+12(offset) = 32\text{bits}\]

Appropriate value for the parameter \(L\): ___2^{20}_____

\[PA: 12(ppn)+12(offset) = 32\text{bits}\]

Appropriate value for the parameter \(B\): __12+2=14_____

(B) If the designers decide to decrease the page size to 2048 \(2^{11}\) bytes but keep the same size virtual and physical addresses, what affect will the change have on the following architectural parameters? Use a letter “a” through “e” to indicate how the new value of the parameter compares to the old value of the parameter:

(a) doubled (b) increased by 1 (c) stays the same (d) decreased by 1 (e) halved

Size of page table entry in bits: _B_

Number of entries in the page table: _A_

Maximum percentage of virtual memory that can be resident at any given time: ___C___

\[2^{24}/2^{11}=2^{13} \text{ physical pages}\]
\[2^{32}/2^{11}=2^{21} \text{ virtual pages}\]
Old: \[2^{12}/2^{20}=2^{-8}\], new: \[2^{13}/2^{21}=2^{-8}\]
(D) A test program has been running on the RISC-V with a page size of $2^{12}$ bytes and has been halted just before execution of the following instruction at location 0x1234. Assume the current contents of x2 are 0x3000.

\[ \text{sw } x1, 0x4c8(x2) \mid PC = 0x1234 \]

VA=0x34c8, VPN=3, not resident

The first 8 locations of the page table at the time execution was halted are shown to the right; the least-recently-used page ("LRU") and next least-recently-used page ("next LRU") are as indicated. Assume that all the pages in physical memory are in use. Execution resumes and the SW instruction is executed.

Please show the contents of the page table after the SW instruction has completed execution by crossing out any values that changed and writing in their new values. Note that the D and PPN fields for a non-resident page do not need to be specified.

(E) Which physical pages, if any, need to be written to disk during the execution of the SW instruction in part (D)?

Physical page numbers written to disk or NONE: \_7\_ 

Problem 4.

Consider a virtual memory system that uses a single-level page table to translate virtual addresses into physical addresses. Each of the questions below asks you to consider what happens when just ONE of the design parameters (page size, virtual memory size, physical memory size) of the original system is changed. Circle the correct answer.

(A) If the physical memory size (in bytes) is doubled, the number of entries in the page table
   (a) stays the same
   (b) doubles
   (c) is reduced by half
   (d) increases by one
   (e) decreases by one

# entries only depends on # bits of VPN

(B) If the page size (in bytes) is halved, the number of entries in the page table
   (a) stays the same
   (b) doubles
   (c) is reduced by half
   (d) increases by one
   (e) decreases by one

VPN is one bit larger

(C) If the virtual memory size (in bytes) is doubled, the number of bits in each entry of the page table
   (a) stays the same
   (b) doubles
   (c) is reduced by half
   (d) increases by one
   (e) decreases by one

(D) If the page size (in bytes) is doubled, the number of bits in each entry of the page table
   (a) stays the same
   (b) doubles
   (c) is reduced by half
   (d) increases by one
   (e) decreases by one

PPN is one bit smaller
Consider a virtual memory system for a new processor with 4096 \( (2^{12}) \) virtual pages and 16384 \( (2^{14}) \) physical pages where each page contains 1024 \( (2^{10}) \) bytes. The first 8 entries of the current page table are shown below:

\[
\begin{array}{ccc}
\text{index} & \text{D} & \text{R} & \text{PPN} \\
0 & 1 & 1 & 0x22 \\
1 & 0 & 1 & 0x01 \\
2 & -- & 0 & -- \\
3 & 0 & 1 & 0x02 \\
4 & 1 & 1 & 0x03 \\
5 & -- & 0 & -- \\
6 & 1 & 1 & 0x15 \\
7 & 0 & 1 & -- \\
\ldots & & & \\
\end{array}
\]

(E) What is the total number of bits in the page table?

\[
\text{Total number of bits in the page table: } 2^{12} \times 16 = 2^{16} \]

\[
2^{12} \text{ entries} \times (\text{D}: 1 + \text{R}: 1 + \text{PPN}: 14) = 2^{12} \times 16
\]

(F) Which address bits from the CPU are used to choose an entry from the page table?

Address bits used to choose page table entry: \( \text{A} \left[ _{21} : _{10} \right] \)

(G) What is the physical address for the word at virtual location 0x1234? Write “not resident” if the location is not currently present in physical memory.

Physical address for byte at virtual address 0x1234 or “not resident”: 0xE34

\[
0x1234 = 0001_2 \ 010_2 \ 011_2 \ 0100_2 \\
\text{VPN}=4, \text{PPN}=3, \\
0000_2 \ 1110_2 \ 0011_2 \ 0100_2 = 0xE34
\]

(H) Briefly explain what action caused the D bit for page 6 to be 1.

A store instruction wrote to a location in virtual page 6

Briefly explain.
Problem 5.

(A) A particular RISC-V implementation has 32-bit virtual addresses, 32-bit physical addresses and a page size of $2^{12}$ bytes. A test program has been running on this RISC-V and has been halted just before execution of the following instruction at location 0x1FFC. Assume $x2 = \text{0x3000}$ and $x3 = \text{0x6000}$ just prior to executing these instructions.

```
lw x1, 0x4C8(x2) | PC = 0x1FFC  VA: 1FFC, VPN=1, PPN=0
sw x1, 0x4(x3)  | PC = 0x2000  34C8, VPN=3, PPN=6  
```

The first 8 locations of the page table at the time execution was halted are shown below; the least recently used page ("LRU") and next least recently used page ("next LRU") are as indicated. Assume that all the pages in physical memory are in use. Execution resumes and the LW and SW instructions are executed.

Please show the contents of the page table after the SW instruction has completed execution by crossing out any values that changed and writing in their new values.

\[
\begin{array}{c|ccc}
VPN & D & R & PPN \\
0 & 1 & 1 & \text{0x1} \\
1 & 0 & 1 & \text{0x0} \\
LRU\rightarrow 2 & 1 & 0 & 0,1 \text{ 0x6 0x4} \\
3 & \text{--0} & 0 & 1 \text{ 0x6} \\
Next LRU\rightarrow 4 & 0 & 1 & 0x4 \text{ --} \\
5 & 0 & 1 & \text{0x2} \\
6 & 0 & 1 & \text{0x7} \\
7 & 0 & 1 & \text{0x3} \\
\end{array}
\]

VA: 2000, VPN=2, miss, evict next LRU, PPN=0x4  
VA: 6004, VPN=6, PPN=7

(B) Which physical pages, if any, needed to be written to disk during the execution of the LW and SW instructions?

Physical page numbers written to disk or NONE: ____ 6 ____

(C) Please give the 32-bit physical memory addresses used for the four memory accesses associated with the execution of the LW and SW instruction.

32-bit physical memory address of LW instruction: 0x____ 0FFC______

32-bit physical memory address of data read by LW: 0x____ 64C8_____

32-bit physical memory address of SW instruction: 0x____ 4000______

32-bit physical memory address of data written by SW: 0x____ 7004______
Problem 6.

Consider a system with 40-bit virtual addresses, 36-bit physical addresses, and 64 KB (2^{16} bytes) pages. The system uses a page table to translate virtual addresses to physical addresses; each page table entry include dirty (D) and resident (R) bits.

(A) (2 points) Assuming a flat page table, what is the size of each page table entry, and how many entries does the page table have?

\[ \text{VA: 24+16; PA: 20+16} \]

Size of page table entry in bits: \[ 20+2=22 \]

Number of entries in the page table: \[ 2^{24} \]

(B) (1 point) If you changed the system to use 16 KB (2^{14} bytes) pages instead of 64 KB pages, how would the number of entries in the page table change? Please give the ratio of the new size to the old size.

\[ 2^{40}/2^{14}=2^{26}; \quad 2^{40}/2^{16}=2^{24}; \quad 2^{26}/2^{24}=4 \]

(# entries with 16 KB pages) / (# entries with 64 KB pages): \[ 4 \]

Assume 64 KB pages for the rest of this exercise.

(C) (6 points) The contents of the page table and TLB are shown to the right. The page table uses an LRU replacement policy, and the LRU page (shown below) will be chosen for replacement. For each of these four accesses, compute its corresponding physical address and indicate whether the access causes a TLB miss and/or a page fault. Assume each access starts with the TLB and Page Table state shown to the right.

<table>
<thead>
<tr>
<th>Virt Addr (in hex)</th>
<th>Phys Addr (in hex)</th>
<th>TLB Miss?</th>
<th>Page Fault?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 0x06004</td>
<td><em>0xBE7A</em></td>
<td><em>0xBE7A6004</em></td>
<td>Y / N</td>
</tr>
<tr>
<td>2. 0x30286</td>
<td><em>0x8</em></td>
<td><em>0x80286</em></td>
<td>Y / N</td>
</tr>
<tr>
<td>3. 0x68030</td>
<td><em>0x70</em></td>
<td><em>0x708030</em></td>
<td>Y / N</td>
</tr>
<tr>
<td>4. 0xBEEF</td>
<td><em>0x8</em></td>
<td><em>0xBEEF</em></td>
<td>Y / N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TLB VPN</th>
<th>R</th>
<th>D</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>1</td>
<td>0</td>
<td>0xBE7A</td>
</tr>
<tr>
<td>0x3</td>
<td>0</td>
<td>0</td>
<td>0x7</td>
</tr>
<tr>
<td>0x5</td>
<td>1</td>
<td>1</td>
<td>0xFF</td>
</tr>
<tr>
<td>0x2</td>
<td>1</td>
<td>0</td>
<td>0x900</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Page Table VPN</th>
<th>R</th>
<th>D</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0xBE7A</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>---</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0x900</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0x8</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>---</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0xFF</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0x70</td>
</tr>
</tbody>
</table>

LRU PAGE