LE16.3

LE16.3.1: Page Map Arithmetic

0.0/1.0 point (ungraded)

Consider a virtual memory system that uses a single-level page map 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. Select the correct answer.

1. If the physical memory size (in bytes) is doubled, the number of entries in the page table
   - ○ stays the same
   - ○ doubles
   - ○ is reduced by half
   - ○ increases by one
   - ○ decreases by one

2. If the page size (in bytes) is halved, the number of entries in the page table
   - ○ stays the same
   - ○ doubles
   - ○ is reduced by half
   - ○ increases by one
   - ○ decreases by one

3. If the virtual memory size (in bytes) is doubled, the number of bits in each entry of the page table
   - ○ stays the same
   - ○ doubles
   - ○ is reduced by half
   - ○ increases by one
   - ○ decreases by one

4. If the page size (in bytes) is doubled, the number of bits in each entry of the page table
   - ○ stays the same
   - ○ doubles
   - ○ is reduced by half
   - ○ increases by one
   - ○ decreases by one

Consider a virtual memory system for the Gamma 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 map are shown below:

index	D	R	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	0x04

...

5. What is the total number of bits in the page map?

Total number of bits in the page map (you can express your answer as a mathematical expression like 2.5*2^7, for example):

6. Which address bits from the CPU are used to choose an entry from the page table?

Address bits used to choose page table entry: A[:]

7. What is the physical address for the word at virtual location 0x1234?

Physical address for byte at virtual address 0x1234: 0x

8. What action caused the D bit for page 6 to be 1?

- A LD of address 0x1856
- A LD of virtual page 6.
- A ST instruction wrote to a location in physical page 6.
- A ST instruction wrote to a location in virtual page 6.

Submit

LE16.3.2: Memory access time

0.0/1.0 point (ungraded)

The pagemap resides to main memory and accessed via a translation lookaside buffer (TLB). Each main memory access takes 50 ns and each TLB access takes 10 ns. Each virtual memory access involves:

- mapping VPN to PPN using TLB [10 ns]
- if TLB miss: mapping VPN to PPN using page map in main memory [50 ns]
- accessing main memory at appropriate physical address [50 ns]

Assuming no page faults (i.e., all virtual memory pages are resident) what TLB hit rate is required for an average virtual memory access time of 61 ns?

TLB hit rate for 61 ns access time:

Submit

LE16.3.3: TLB

0.0/1.0 point (ungraded)

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

1. Assuming a flat page map, what is the size of each page map entry, and how many entries does the page map have? You can express your answers as mathematical expressions if you like, e.g., 10*(2^17).

   Size of page map entry in bits: 22 Answer: 22

   Explanation
   Physical addresses are 36 bits long. 16 of those bits are used for the page offset, so 20 bits are used for the PPN. In addition, we need a dirty bit and a resident bit per page map entry, so we end up with 22 bits per entry.

   Number of entries in the page map: 2**24 Answer: 2^24

   Explanation
   Virtual addresses are 40 bits long. 16 of those bits are used for the page offset, so 24 bits are used for the VPN. Since we need one entry per virtual page in the page map, that means that we have 2^{24} entries in the page map.

2. If we 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 map change? Please give the ratio of the new size to the old size.

   (# entries with 16 KB pages) / (# entries with 64 KB pages): 4 Answer: 4

   Explanation
   If the page size goes down by a factor of 4 while the virtual memory size remains the same, that means that there must be 4 times as many virtual pages.

   Assume 64 KB pages for the rest of this exercise.

3. The contents of the page map and TLB are shown below. The page map 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 Map state shown below.

   TLB:

VPN(tag)	V	D	PPN
0x0	1	0	0xBE7A
0x3	0	0	0x7
0x5	1	1	0xFF
0x2	1	0	0x900

   Page Table:

VPN	D	R	PPN
0x0	0	1	0xBE7A
0x1	--	0	--
0x2	0	1	0x900
LRU → 0x3	0	1	0x8
0x4	--	0	--
0x5	1	1	0xFF

0x6	0	1	0x70

Fill in the table below

Virtual Addr	PPN (in hex)	Physical Addr (in hex)	TLB Miss?	Page Fault?
0x06004	0x: BE7A Answer: BE7A	0x: BE7A6004 Answer: BE7A6004	No Answer: No	No Answer: No
0x30286	0x: 8 Answer: 8	0x: 80286 Answer: 80286	Yes Answer: Yes	No Answer: No
0x68030	0x: 70 Answer: 70	0x: 708030 Answer: 708030	Yes Answer: Yes	No Answer: No
0x4BEEF	0x: 8 Answer: 8	0x: 8BEEF Answer: 8BEEF	Yes Answer: Yes	Yes Answer: Yes

Explanation
VA 0x06004: VPN = 0x0, page offset = 0x6004. Results in TLB hit which provides PPN = 0xBE7A. A TLB hit also means that there was no page fault because the page is in main memory. Our physical address is produced by concatenating the VPN with the page offset. Since the page offset happens to be a multiple of 4, it's easy to do in hex. So physical address = 0xBE7A6004.
VA 0x30286: VPN = 0x3, page offset = 0x0286. Although there is an entry for VPN 0x3 in the TLB, it's valid bit is 0, so it cannot be used. So we look it up in the page map instead. There we find a resident page with PPN = 0x8. Concatenating the two together results in a physical address of 0x80286. This example showed a TLB miss together with a page map hit (i.e., no page fault).
VA 0x68030: VPN = 0x6, page offset = 0x8030. Looking up page 6 in the TLB results in a TLB miss. Looking it up in the page map results in a hit with a resulting PPN of 0x70. So the physical address is 0x708030.
VA 0x4BEEF: VPN = 0x4, page offset = 0xBEEF. Looking up page 4 in the TLB results in a TLB miss. Looking up page 4 in the page map also results in a miss. This means that we have a page fault in this case and must bring the page into main memory. In order to do that, we must first remove the LRU page from main memory. The LRU page is VPN 3 whose dirty bit is 0, so the page does not need to be written back to disk. We set it's resident bit to 0, and now we can use physical page 0x8 for VPN 4. We set the R bit for VPN 4 to 1, and the PPN to 0x8. So the resulting PPN is 8, and the resulting physical address is 0x8BEEF.

Submit

Answers are displayed within the problem

Discussion

Topic: 16. Virtual Memory / LE16.3

Hide Discussion

Add a Post

Show all posts by recent activity

Physical page number outnumbers virtual page number — 3
Physical page number outnumbers virtual page number in LE16.3.1. How is that possible?

‹ Previous Next ›

edX

Legal

Connect