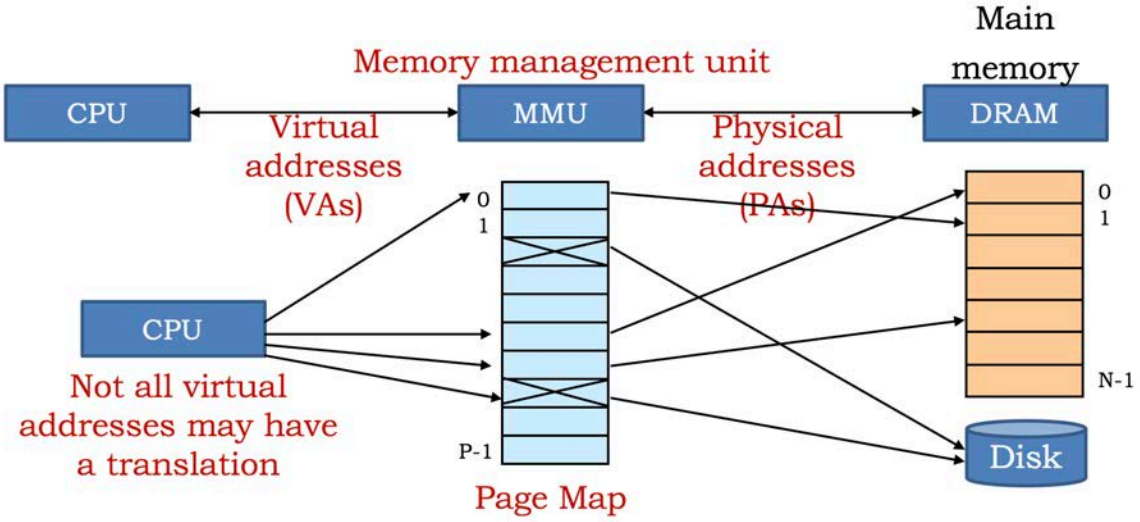
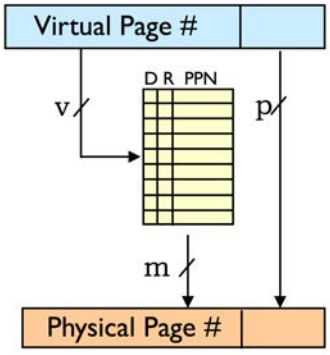
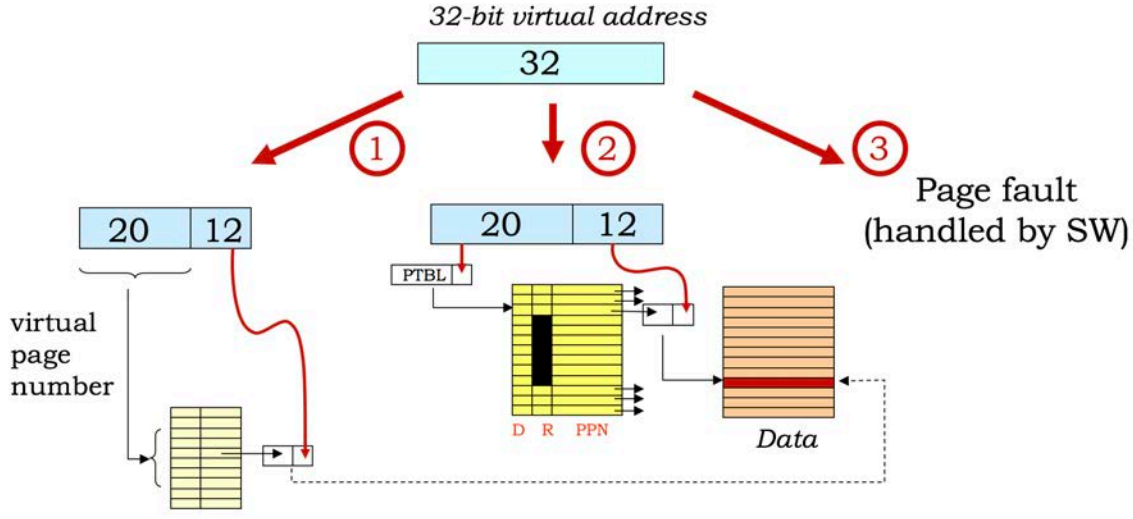
Computation Structures

Virtual Memory Worksheet





(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 map



Look in TLB: VPN→PPN cache
Usually implemented as a small fully-associative cache

Problem 1.

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

VPN	*D*	*R*	*PPN*
0	0	1	2
1	—	0	—
2	0	1	4
3	—	0	—
4	1	1	0
5	1	1	1
6	—	0	—
7	—	0	—
8	—	0	—
9	—	0	—
A	—	0	—
B	—	0	—
C	1	1	7
D	1	1	6
LRU→E	1	1	5
F	0	1	3

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

Size of page map (bits): _____

(B) (The following instruction, located at virtual address 0x0BA, is about to be executed.

```
LD(R31, 0x2C8, R0)
```

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

Physical address for instruction fetch: 0x _____

Physical addr for data read by LD instruction: 0x _____

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

```
ST(BP, -4, SP)     // current value of SP = 0x604
```

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

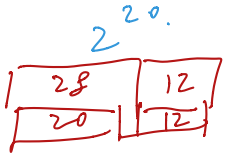
Remember to show any changes to the dirty and resident control bits as well as updates to the physical page numbers. If an entry in the page map no longer matters, please indicate that by replacing it with "— 0 — " for the D, R and PPN entries.

Show updated contents of page map

Problem 2.

Consider a Beta 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 map for this processor? Assuming the page map includes the standard dirty and resident bits, specify the width of each page map entry in bits, and number of entries in the page map.



Size of page map entry in bits: 22

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

(B) The following test program is running on this Beta 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 Beta processor also has a 4 element, fully associative, Translation Lookaside Buffer (TLB) that caches page map translations from VPN to PPN.

```
. = 0x0
ADDC(R31,0x2800,R3)
LD(R3,0,R5)
ST(R5,0x4100,R31)
```

0x0
0x4.

TLB

	Tag (VPN)	*D*	*R*	*PPN*
LRU→	~~0x3~~ 0 0	1	1	~~0x1~~
	0x2	0	1	0x3
	0x6	0	1	0x2
Next LRU→	0x1	0	1	0x5

0x7

Page Map

VPN	*D*	*R*	*PPN*
0	1	1	0x7
1	0	1	0x5
2	0	1	0x3
LRU→ 3	1	1	0x1
4	--	0	--
5	0	1	0x0
6	0	1	0x2
Next LRU→ 7	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.*

VPN	TLB Hit (Yes/No)	Page Fault (Yes/No)	PPN
0x 0	No	No	0x7
~~0x 0~~	~~Yes~~	~~No~~	~~0x7~~
0x 2	Yes	No	0x3.
0x4	No	Yes.	0x1

(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: none

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

Physical address of LD instruction: 0x 7004

Problem 3.

Consider a Beta 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 map using a separate SRAM memory with L entries, where each entry has B bits. If the page map includes the standard dirty and resident bits, what are the appropriate values for the parameters L and B?

Appropriate value for the parameter L: 2^{20}

Appropriate value for the parameter B: 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 map entry in bits: b

Number of entries in the page map: a

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

(D) (4 points) A test program has been running on the Beta with a page size of 2^{12} bytes and has been halted *just before* execution of the following instruction at location 0x1234:

```
ST(R1,0x34C8,R31)    | PC = 0x1234
```

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 ST instruction is executed.

VPN	*D*	*R*	*PPN*
0	1	1	0x1
1	0	1	0x0
2	1	1	0x6
3	-- 1	0	-- 0x7
Next LRU→ 4	0	1	0x4
5	0	1	0x2
LRU→ 6	~~1~~	~~1~~ 0	~~0x7~~
7	0	1	0x3

Please **show the contents of the page table** after the ST 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) (1 point) Which physical pages, if any, need to be written to disk during the execution of the ST instruction in part (D)?

Physical page numbers written to disk or NONE: 0x7

Problem 4.

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. **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

(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

(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 1-bit smaller.

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	
...			

22.

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

$2^{12} \cdot (14+2)$

Total number of bits in the page map: _____

$= 2^{16}$

(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: A[21 **:** 10 **]**

0007 11 10 3 x

(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": 0x.E34

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

Briefly explain.

A ST instruction write data to visual page 6.

Problem 5.

(A) A particular Beta 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 Beta and has been halted *just before* execution of the following instruction at location 0x1FFC:

```
LD(R31,0x34C8,R1)    | PC = 0x1FFC
ST(R1,0x6004,R31)    | PC = 0x2000
```

VPN = 20.
PPN = 20

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 LD and ST instructions are executed.

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

VPN	*D*	*R*	*PPN*
0	1	1	0x1
1	0	1	0x0
LRU→ 2	~~1~~ 0	1	~~0x6~~ 0x4
3	-- 0	~~0~~ 1	-- 0x6
Next LRU→ 4	~~0~~ --	~~1~~ 0	~~0x4~~
5	0	1	0x2
6	~~0~~ 1	1	0x7
7	0	1	0x3

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

Physical page numbers written to disk or NONE: 0x6

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

32-bit physical memory address of LD instruction: 0x 0FFC.

32-bit physical memory address of data read by LD: 0x 64c8

32-bit physical memory address of ST instruction: 0x 4000

32-bit physical memory address of data written by ST: 0x 7004

Problem 6.

VPN = 24. PPN = 20

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 include dirty (D) and resident (R) bits.

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

Size of page map entry in bits: 22

Number of entries in the page map: 2^{24}

(B) (1 point) If 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

Assume 64 KB pages for the rest of this exercise.

(C) (6 points) The contents of the page map and TLB are shown to the right. 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 to the right.

TLB

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

Fill in table below

	Virt Addr	PPN (in hex)	Phys Addr (in hex)	TLB Miss?	Page Fault?
1.	**0x06004**	BE7A	~6004	Y / (N)	Y / (N)
2.	**0x30286**	8	80286	(Y) / N	Y / (N)
3.	**0x68030**	70	708030.	(Y) / N	Y / (N)
4.	**0x4BEEF**	8	8BEEF	(Y) / N	(Y) / N

Page Map

VPN	R	D	PPN	
0	1	0	0xBE7A	
1	0	0	---	
2	1	0	0x900	
3	1	0	0x8	← **LRU PAGE**
4	0	0	---	
5	1	1	0xFF	
6	1	0	0x70	
...				