Homework 10

Due Jun 3 by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip, tar, gz, pdf, txt, hex, circ, ys, and yo
Available after May 28 at 11:59pm

Homework 10: The Hardware/OS interface

In this assignment, you will gain a deeper understanding of the Y86 "missing pieces" of privilege and MMUs (which will be important in your later CS career).

Start with your Y86 from HW7 or HW8, or our sample solution from HW7.

Q1: Privilege (20pts)

Add a privmode bit to your LogiSim Y86. (Per convention, "0" means "priviliged" aka "kernel mode.") At boot time, the bit should be cleared to 0.

Wire your memory mapped I/O so that privmode needs to be zero in order for any I/O memory accesses to go through.

Add two new instructions:

- enter_os, which does the same as jmp 0x0001000, except it also clears privmode to 0. (E.g., it traps to the special privileged code living in ROM.)
- leave_os, which does the same as jmp 0x0000100, except it also sets privmode to be 1.

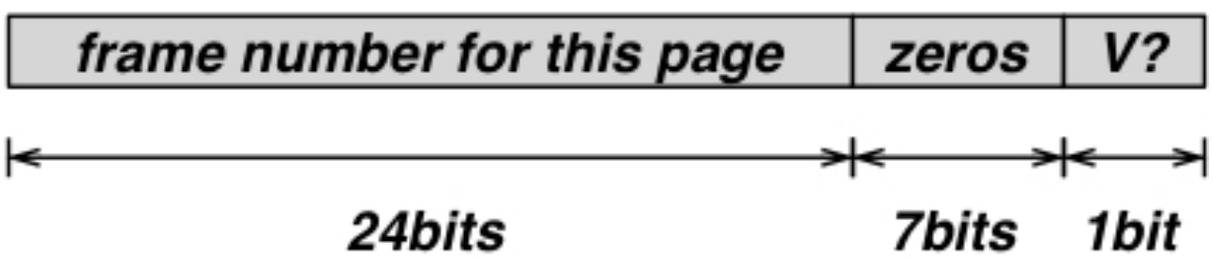
Q2: Exception/Trap (15pts)

Add a flip/flop to indicate whether or not a memory error exception has occurred. At boot time, the flip/flop should be cleared to zero. When the "memory trap" line is asserted, the F/F should be set to one, light up a red LED, and stop the clock. (In a real system, a hardware trap such as this would trigger a special "call into OS" call.)

Q3: MMU (25pts)

Build a simple MMU that sits between the processor (e.g., the SHIMS) and memory (RAM, ROM, I/I) and does the following:

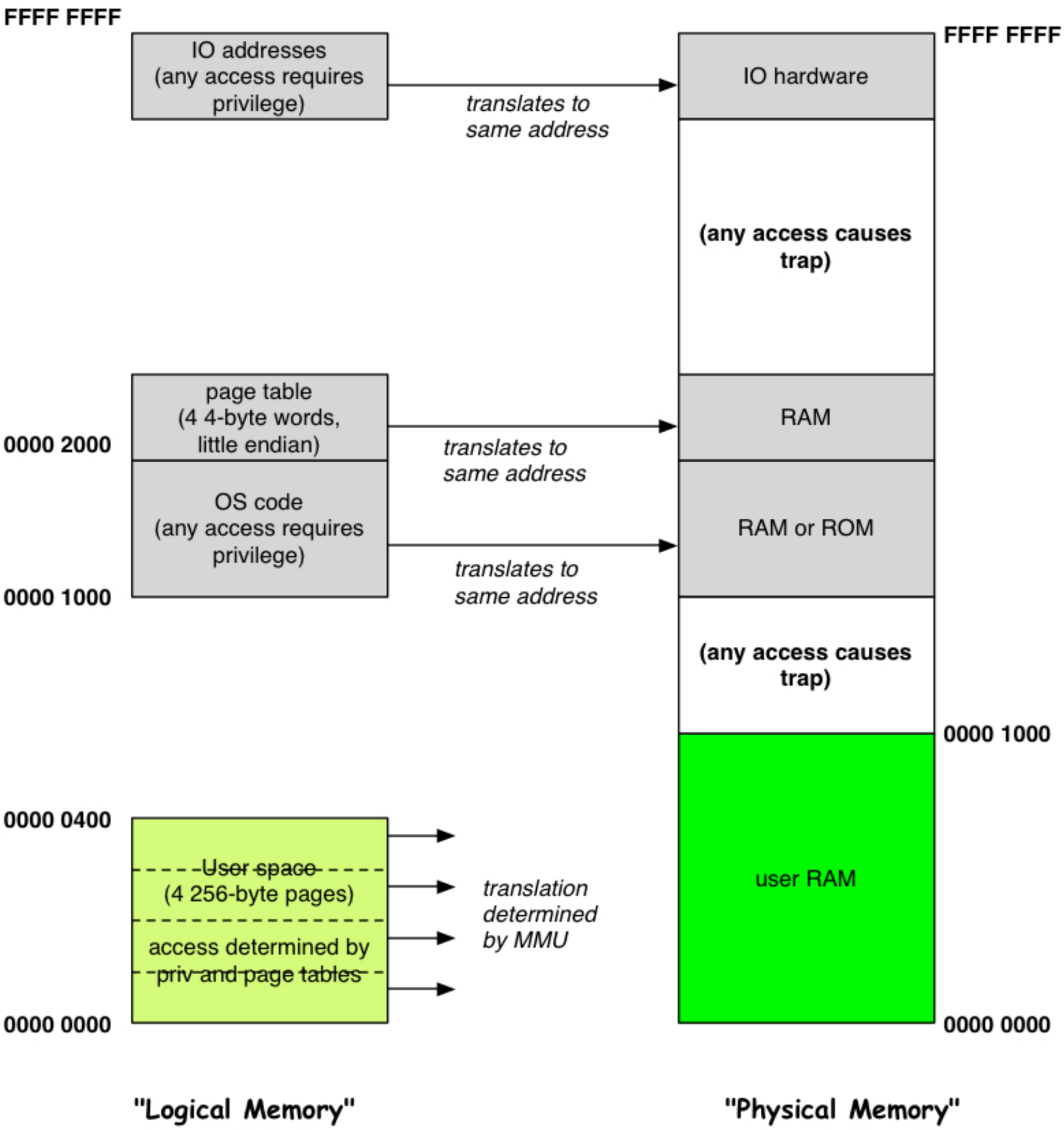
- The MMU receives the privilege bit as an input, and RAM_USE and the address to indicate the processor is attempting a memory access.
- The MMU has a page table consisting of four entries; each entry is as follows:



- If priv == 0 (that is, we're in privileged mode), the MMU passes the request on to physical memory.
- If priv == 1 and the requested address is above 0x0000 0400, the MMU throws a trap.
- If priv == 1 and the requested address is 0x0000 0NZY (for N < 4), the MMU consults the page table entry #N. If the valid bit is 0, the MMU throws a trap. If the valid bit is 1, the MMU uses the frame number to calculate a new address (by concatenating the frame number to the offset ZY), and passes that on physical memory.
- If physical memory receives an access request to an address where nothing lives, it throws a trap.

Q4: Page Table (20pts)

Wire your MMU's page table to live as 32 bytes in the physical address space at address 0x0000 2000 (so privileged code can thus read and write these bytes). The overall memory map thus looks like this:



Q5: Multiple Processes (20pts)

- Write a simple user program "Looper" that loops for a while and then invokes enter_os. It should start at logical address 0x100.
- Write a simple user program "NullPointer" that read from address 0x0000 0000, and then invokes enter_os. It should start at logical address 0x100.
- Install Looper and NullPointer in the user RAM. Write a simple OS (starting at 0x0000 1000) that, the first time through:
  - configures the page tables so that Looper lives at 0x100 (and page 0) is invalid
  - invokes enter_user, to run Looper

  and the second time through:

  - configures the page tables so that NullPointer lives at 0x100 (and page 0) is invalid
  - invokes enter_user, to run NullPointer.
- Show that it works---and that NullPointer crashes the system!

Homework 10		
Criteria	**Ratings**	**Pts**
Q1. enter_os view longer description		10 pts
Q1. leave_os view longer description		10 pts
Q2. Trap LED		5 pts
Q2. Stopping the clock on trap		5 pts
Q2. Clarity, other specs		5 pts
Q3. in: priv, RAM_USE, RAM_addr; out: RAM_addr, RAM_USE and having a page table		5 pts
Q3. If priv == 0 (that is, we're in privileged mode), the MMU passes the request on to physical memory		5 pts
Q3. If priv == 1 and the requested address is above 0x0000 0400, the MMU throws a trap		5 pts
Q3. If priv == 1 and the requested address is 0x0000 0NZY (for N < 4), the MMU consults the page table entry #N. If the valid bit is 0, the MMU throws a trap. If the valid bit is 1, the MMU uses the frame number to calculate a new address (by concatenating the frame number to the offset ZY), and passes that on physical memory		5 pts
Q3. If physical memory receives an access request to an address where nothing lives, it throws a trap		5 pts
Q4. Elegance, clarity, neatness, etc.		5 pts
Q4. Attempt		5 pts
Q4. Correctness		10 pts
Q5. Looper view longer description		5 pts
Q5. NullPointer view longer description		5 pts
Q5. OS view longer description		5 pts
Q5. Showing it works		5 pts
		Total Points: 100