HW8 Documentation and Testing:

Question 1:
You can see the privilege bit in a flip flop to the right of my DShim. Also, note that I anded the on/off values with IMemReady and IMemReq to avoid having the privilege bit flip too soon and trigger a memory trap in certain edge cases (compatability with the modifications in q2).

I tested this with test1.ys (specifically documented therein with inline comments).

Question 2:
The memory trap flip flop is to the left of the main memory. you can see that, initially I tested it by programming in a "memory trap" instruction by hand (but later disabled this by disconnecting the comparator result from the OR-gate leading into the memory trap flip flop. When the flip flop is triggered, it stops the clock by flipping the value opening a controlled buffer on both the clock and fast clock.

The flip can also be triggered if the physical address does not exist in physical ram (ie. it's greater than 0xffff and less that 0x00fffe00 OR it's greater than 0x00fffe0f).

Initial tests were conducted with the test2.ys code (and the manual "memory trap" command, which later was disabled in the microarchitecture).

Question 3:
My solution to this question is implemented in my MMU subcircuit. It converts virtual to physical addresses and throws a memory trap if certain privilege conditions are not met. I tested this by inputing values by hand and verifying that the intended results were output and written.

Question 4:
See above (page table implemented in and tested with the MMU subcircuit).

Question 5:
See multi-proc.ys to see the mini-OS I wrote for this program. I installed it on the program, and tested it by watching it run all the way to the end (and then cause a memory trap).

The addresses in the PC always were displayed as the logical addresses (when privilege mode was enabled), but they converted in the MMU to the correct physical address and each program got run. Then, program 2 causes a memory trap.