**Review for Exam over Chapter 2**

Hardware

Vocabulary to know

pipelining, clock, cache, evict, block, line, dirty bit, write back, write through

Make sure you understand these beyond just being able to explain them

Here are some examples

What is the difference between write back and write through?

Explain pipelining

When cache evicts something, what does it evict, and why does it do it?

Know the steps in the CPU cycle and what happens in each one

Know the registers the CPU itself uses (MAR, MDR, IR, PC) and how the CPU uses each one

Know what the FLAGS register contains.

Each flag is a bit

You don’t have to know each bit, but know several flags and what they mean

The bits (flags) are set after compares and math operation

Be familiar with the registers

rax, rbx, rcx, rdx: all 64 bit registers

eax, ebx, ecx, edx: all 32 bit registers

ax, bx, cx, dx: the lower 16 bits of registers corresponding to those above

al, bl, cl, dl: the lower 8 bits of registers corresponding to those above

ah, bh, ch, dh: the upper 8 bits of registers listed in Line 3 above

Lab activities

Know what a stack is. Know what push, pop do.

Be able to trace what a stack looks like after some pushes and pops

Know the addressing modes

Register (Data is in a register)

Direct (Data is in memory)

Immediate (Data is in the command, as in MOV $3, %eax)

Indirect (The register contains the address of the data)

Know these commands and what they do

add — Add

dec — Decrement by 1

mov — Copy data

jne, jnz, jmp — Jump instructions

**Some Sample Questions**

1. Explain these terms:

clock, cache, block, line, dirty bit,

2. What is the difference between write back and write through?

3. When cache evicts something, what does it evict, and why does it do it?

2. Tell the steps in the CPU cycle and what happens in each one

3. Explain how the CPU uses the MAR and MDR

4. How is the IR used in the machine cycle?

5. How does the CPU use the PC?

6. List and explain three bits in the FLAGS register

7. Explain the difference between the ah, al, and ax registers.

8. Explain the ax, rax, and eax registers and the relation between them.

9. What is a stack?

10. Explain how the push, pop functions work on a stack.

11. Give a common example of a stack.

12. Be able to trace what an integer stack looks like after these pushes and pops. Assume the stack is empty before any of this code is executed.

push (32);

push (56);

int va = pop();

…

13. Explain each addressing mode with an example.

14. Each of these instructions has an error. Find and explain the error.

decl %ax

movl %ax,%bx

addd %eax, %bax

15. What is the difference between a hardware instruction and an assembler directive?