CSC 462/562 Homework #8 Intel architecture
Due: Wednesday, April 17

Word process all answers. Answer question 1 (which counts as 2 questions) and any 3 other questions.

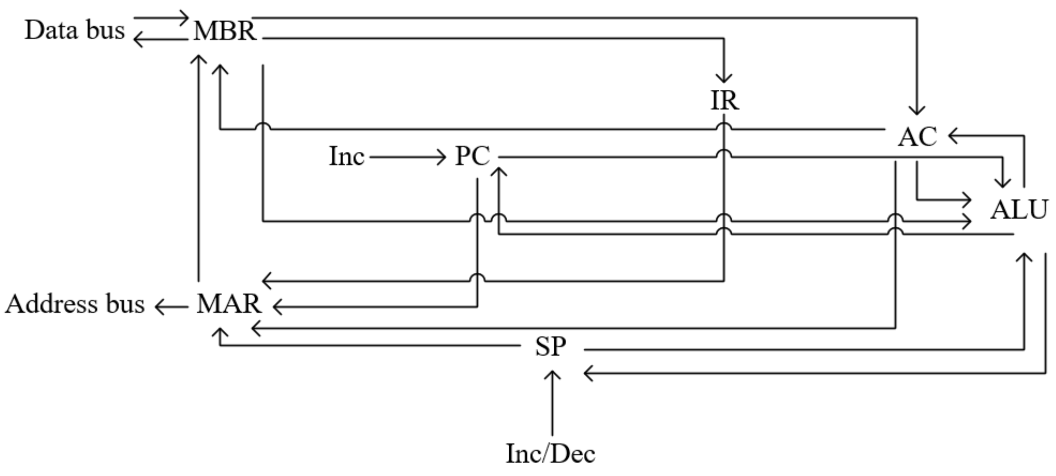
1. Use the following micro-architecture with the control signals shown below to provide the microprograms needed for the operations below. An example is shown first.

C0 = PC → MAR	C1 = MAR → address bus	C2 = memory read
C3 = memory write	C4 = Data bus → MBR	C5 = MBR → Data bus
C6 = MBR → IR	C7 = MBR → AC	C8 = AC → MBR
C9 = IR (address portion) → MAR	C10 = IR (address portion) → ALU	
C11 = MBR → MAR	C12 = AC → ALU	C13 = MBR → ALU
C14 = Inc PC	C15 = PC → ALU	C16 = ALU → PC
C17 = Inc SP	C18 = Dec SP	C19 = SP → MAR
C20 = SP → ALU	C21 = ALU → SP	C22 = ALU add
C23 = ALU subtract	C24 = ALU negation	C25 = ALU Inc
C26 = ALU → AC	C27 = AC < 0	C28 = AC == 0
C29 = AC > 0	C30 = PC → MBR	C31 = microbranch

To use C27, C28, C29, a value has to be moved to the ALU (either previously computed by for instance ALU subtract, or from the AC) and then an if-else statement follows in which, if the condition tested is true, the if clause executes, otherwise a microbranch executes.

As an example, instruction fetch would entail:

T0: C0	PC → MAR
T1: C1, C2, C4	MAR → addr bus, memory read, data bus → MBR
T2: C6, C14	MBR → IR, Inc PC



a. Write the microprogram for the assembly instruction PUSH, which stores what is in the AC to the top of the stack as pointed to by the SP, incrementing the SP appropriately.
b. Write the microprogram for the assembly instruction StoreIndirect X where address X is part of the instruction in the IR. This requires two memory accesses, one to retrieve the pointer to the location and one to store the AC value to that memory location.
c. Write the microprogram for the assembly instruction JG Foo, which branches to location PC + Foo if the value in the AC > 0. Foo is in the address portion of the IR.

    d. Write the microprogram for the assembly instruction Neg X (negate) where X is part of the instruction in the IR. Remember to write X back to memory when done.
    e. Write the microprogram for the assembly instruction JnS X. This instruction branches to location X + 1 but first saves the current PC value to location X. X is part an address stored as part of the instruction in the IR.

2. On slide 2 of the Intel power point notes, notice the four segment registers are left shifted by 4 bits. This was done because the original x86 processors used 16 bit registers and therefore 16-bit addresses. But the early IBM PC computers were expandable from 64KB to 1MB (these computers were byte addressable). Therefore, in order to access into 1M of space, an address was left shifted and then added to an offset in another register. This can be seen in the upper left hand portion of slide 3. Explain how this works and what the four segments were used for (you can find a partial description in the sample problems, you might need to research this to finish the question).

3. Examine the various architectures of the 8086, 286, 386 and 486 processors on slides 3, 6, 8 and 13, and answer the following questions.
    a. Which processor(s) permitted virtual memory?
    b. How many adders did each of the 8086, 286 and 386 have?
    c. Which processor first introduced an on-chip cache and what was its size? Which processor introduced the page cache and what was it used for?
    d. Each processor had an instruction queue. What was the number of instructions or length of each queue? The 286 and 386 also had a decoded instruction queue. What is the difference between the prefetch queue and the decoded instruction queue?

4. The Intel x86 architecture started as a CISC instruction set. For backward compatibility sake, the CISC features have been maintained throughout each successive generation. Therefore, the iCore remains a CISC architecture. We emphasized RISC features throughout the course to promote fewer stalls in a pipeline. The original 486 pipeline suffered badly because of the CISC nature of the instruction set. With respect to that 5-stage pipeline, as covered on slides 10-12 of the power point notes) and based on what you know of Intel assembly instructions, explain the areas where the Intel assembly instruction set violates RISC principles and thus causes stalls or penalties in the pipeline. If you want to look over example Intel assembly instructions, see the power point notes used in CSC 362 (http://sappho.nku.edu/~foxr/CSC362/NOTES08/assembly.pptx). Also take into account addressing mode execution times (slide 4 of the Intel notes).

5. In your own words, explain why pipelining microcode resolves may of the deficiencies of trying to pipeline a CISC processor, such as the deficiencies noted that you may have noted in #4 regarding the 486 5-stage pipeline.

6. We have covered many concepts in this class to promote ILP: superscalar, dynamic issue, branch prediction, etc. Discuss the innovations implemented in the iCore 7 with respect to the hardware items we have covered previously in the course (appendix B & C, chapters 2-3).