You may bring one page of notes, front and back.

Questions will be in short-answer format with partial credit for partial answers.

Questions will require you to read and write C and x86 code.

Topics:

- All midterm topics, although the exam will focus on material since the midterm
- Y86 instructions (all): functionality, operands, byte encoding
- Circuits: AND, OR, NOT gates, truth tables, multiplexers
- Pipelining: performance implications, CPI, purpose of each stage (IF, ID, EX, MEM, WB)
- Hazards: data dependencies, forwarding, branch hazards, return hazards
- Optimizations: removing procedure calls and memory references, loop unrolling
- Compiler limitations: procedure calls, memory aliasing
- Memory: SRAM vs DRAM, memory hierarchy
- Caching: locality, direct-mapped cache, 2-way set associative cache

Sample questions:

1. [5] The Y86 addl instruction is encoded as 2 bytes. Describe what information is stored in these 2 bytes:

2. [25] Consider the following Y86 instructions:

   irmovl $314 %eax
   mrmovl 4(%ecx) %ebx
   addl %ebx %eax

   a. (5) Identify all data dependencies in the above code:
   b. (10) How many stalls are required when running the above code without data forwarding? Why?
   c. (10) How many stalls are required when running the above code with data forwarding? Why?

3. [15] Consider the following C code:

   int sum = 0;
   for (int i = 0; i < n; ++i)
       for (int j = 0; j < n; ++j)
           sum += a[j][i];

   a. (5) Describe an element of temporal locality in the above code:
   b. (5) Describe an element of spatial locality in the above code:
   c. (5) Is the above cache friendly? Why or why not?