

Fall 2006 - CS 333 – Sample Exam Questions

1. Draw a block diagram of a micro-programmed control unit. Explain the function of each block in your diagram.
2. Explain the concept of horizontal programming. (One to two sentences).
3. Explain the concept of vertical programming. (One to two sentences).
4. The following is a hexadecimal representation of the operation code for the following SRC instruction:

<u>SRC Instruction</u>	<u>Hexadecimal Representation</u>
andi r8, r8, constant	AA11E1E0

The constant is specified in the instruction.

- a. Show the 17-bit value of the constant as defined in the instruction (in binary and in hexadecimal).
 - b. Show the 32-bit value of the constant after it is sign-extended during the execution of the instruction.
5. Explain the concept of static multiple issue (and VLIW). Make sure to explain how and when decisions are made.
 6. Explain the concept of dynamic multiple issue (superscalar). Make sure to explain how and when decisions are made.
 7. Compare and contrast the differences between abstract RTN and concrete RTN.
 8. Chapter 4. Problems similar in style to: 4.1-4.5, etc. (Write concrete RTN and control sequences)
 9. Why does a branch delay slot exist for some pipelined machines?
 10. A central processing unit (CPU) has the following hardware components:
 - Program Counter (PC)
 - Instruction Register (IR)
 - 1 Arithmetic Logic Unit (ALU) that can perform 8 logical and 8 arithmetic operations
 - 4 general purpose integer registers (R0-R7)
 - 4 floating point registers (F0-F7)
 - A memory address register (MA)
 - A memory data register (MD)

- A register to temporarily hold the result of the ALU (ALU_RESULT)
- A register to temporarily hold one operand input to the ALU (OPER1)

Two internal CPU buses (32 bits wide each) are used to connect these hardware resources.

- Draw the block diagram of the datapath for this machine.
 - List the control signals needed in your datapath.
- Describe the process for developing a microprogrammed control unit. (Describe each step in detail).
 - Consider a pipeline that has 5 stages: (1) instruction fetch – 200 ps, (2) instruction decode - 100 ps, (3) instruction execution – 200 ps, (4) memory read – 200 ps, (5) register write – 100 ps
 - If the machine is not pipelined, but the latencies for the various operations involved in instruction fetch and execute are the same as above, what is the latency of a single instruction ? (Assume all steps must occur).
 - What is the latency of an instruction flowing through the pipeline?
 - What is the execution time of a 100 instruction program without pipelining?
 - What is the execution time of the same 100 instruction program with pipelining ?(no hazards)
 - What is the idealized speedup? (Ignore pipeline fill and pipeline drain)
 - What is the fastest clock frequency for the non-pipelined machine?
 - What is the fastest clock frequency for the pipelined machine?
 - What is the throughput of a 100 instruction program executing on a non-pipelined machine?
 - What is the throughput of a 100 instruction program executing on the pipeline described above?
 - Problems similar to 5.3, 5.4, 5.8, 5.9.
 - Be able to trace/diagram a set of instructions through a pipeline.
 - Identify hazards (know different types of hazards – data, control, branch)
 - Be able to give examples of different types of data (RAW, WAR, WAW) and control hazards
 - Be able to discuss/explain different techniques for handling hazards
 - The exception process allows modification of instruction flow based on internal or external events. Describe the items/tasks that the processor MUST do in order to handle exceptions. Give some examples of different types of exceptions and briefly explain them.
 - What are pipeline registers, what sorts of information do they carry, and why are they needed?
 - Describe the process that needs to be followed in order to design a pipelined architecture.