EECS 583 – Fall 2011 – Midterm Exam

Monday, November 14, 2011
10:40-12:30: open book, open notes

Name: ______________________________________

Please sign indicating that you have upheld the Engineering Honor Code at the University of Michigan.

"I have neither given nor received aid on this examination."

Signature: ______________________________________

There are 13 questions divided into 3 sections. The point value for each question is specified with that question. Please show your work unless the answer is obvious. If you need more space, use the back side of the exam sheets.

Part I: Short Answer
5 questions, 25 pts total Score:_____

Part II: Short Problems
6 questions, 65 pts total Score:_____

Part II: Longer Problems
2 questions, 40 pts total Score:_____

Total (130 possible): _______

Part I. Short Answer (Questions 1-5) (25 pts)

1) A definition that is available must always be reaching. Is this statement *True* or *False*, briefly explain. (5 pts)

2) Given a loop in SSA form with no stores, LICM can always be applied to a load with invariant operands. Is this statement *True* or *False*, briefly explain. (5 pts)

3) When scheduling a basic block, can an instruction be scheduled after its Lstart time? Briefly explain why or why not. (5 pts)

4) For a loop that iterates a large number of times, which modulo schedule will finish the loop in fewer cycles: (a) II = 2, SC = 9; (b) II = 3, SC = 3? Briefly explain your answer. (5 pts)

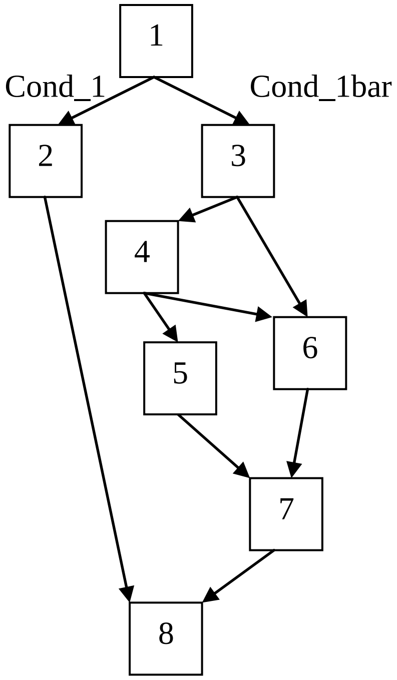
5) Which class of opcodes does a compiler generally get the most benefit from speculation? Briefly explain why. (5 pts)

Part II. Short Problems (Questions 6-11) (65 pts)

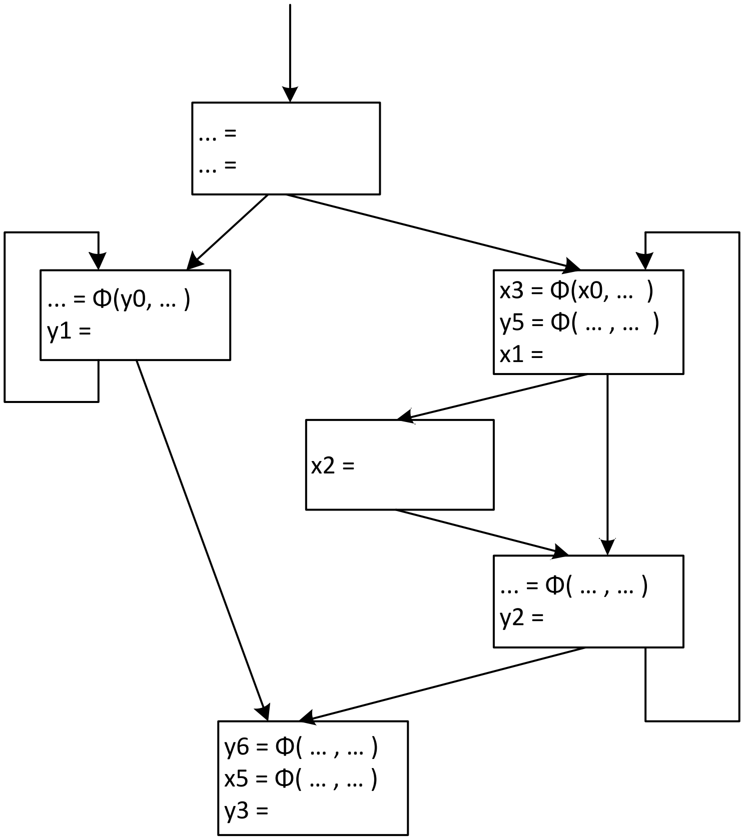
6) For each part below, draw a control flow graph (CFG) consisting of 4 nodes that satisfies the stated properties: (15 pts)

a. A dominates all nodes, B dominates C but not D.
b. D post dominates A, B, C, but A, B, C, and D do not dominate any of the other nodes.
c. B and D are control dependent on A, C and D are control dependent on B, no other control dependences exist.

7) For the following CFG: (a) what is the fewest number of predicates required to if-convert the entire graph; (b) Compute the predicate for BB 5 assuming the branch condition is *Cond_i* and *Cond_ibar* for each BBi with 2 outgoing edges. (10 pts)



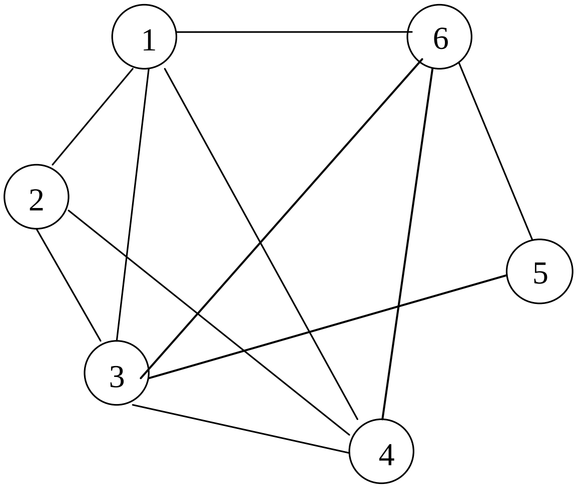
8) Fill in the blanks (locations indicated by …) in the following CFG such that it is in SSA form. You are not allowed to insert extra instructions or Phi nodes. Note, there is no need to go through the algorithm from class in all detail, coming up with the answer by inspection is fine. (10 pts)



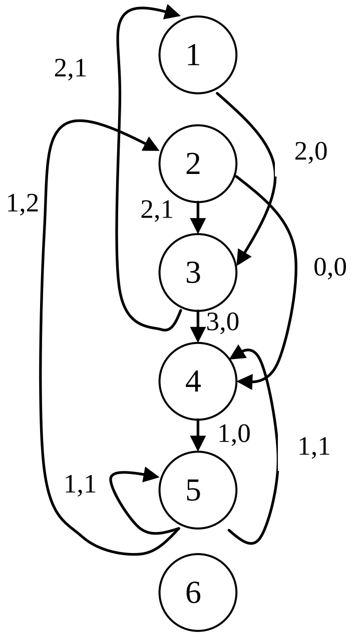
9) Consider the following rolled modulo scheduled code. Specify the II and the unrolled schedule (list the operations by cycle). Assume that p1[0] is the staging predicate for the first stage and the index increases by 1 for subsequent stages. (10 pts)

```
Cycle 0: Load if p1[0]; Add if p1[1]; Mpy if p1[1];
Cycle 1: Sub if p1[0]; Or if p1[2]; Branch if p1[3];
```

10) Consider the following interference graph. When applying the graph coloring algorithm discussed in class for 3 physical registers, which live ranges are spilled to memory? Assume the reference count of each live range is 100. (10 pts)



11) Compute ResMII, RecMII, and MII for the following dependence graph and processor model. (10 pts)



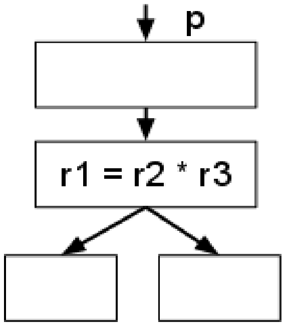
Processor model
4 fully pipelined function units
2 ALU, 1 MEM, 1 BR

Instructions 1 and 4 are memory
Instructions 2, 3, and 5 are ALU
Instruction 6 is a branch

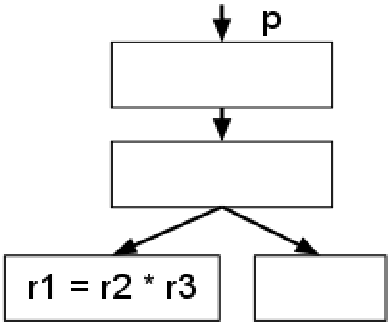
All instructions have 0,0 edges to instruction 6

Part III. Longer Problems (Questions 12-13) (40 pts)

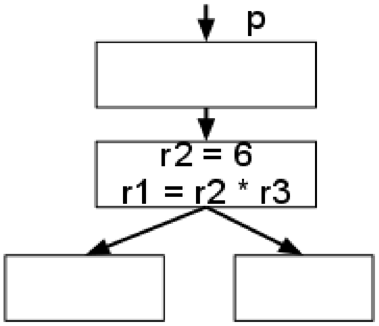
12) An expression e is very busy at point p if in all paths from p to Exit, expression e is evaluated before the value of e is changed. Very busy expressions are good candidates for loop invariant code motion. If a loop invariant expression is very busy then we know that it must be used in the future and evaluating it outside the loop would be worthwhile. The following illustrates whether or not the expression r2*r3 is very busy at point p.



very busy at p



not very busy at p



not very busy at p

Please briefly describe what kind of data flow analysis you will use (forward/backward? must/may?). Define the set of dataflow equations to solve for very busy expression. You should define GEN, KILL, IN and OUT sets. (20 pts)

13) You are the professor in a compiler construction class and you have to write an exam question on optimization. Create a small pseudo assembly function which reduces to:
Store (r1, 100)
after the following optimizations are applied in this order:

1. Common sub-expression elimination
2. Copy propagation
3. Constant propagation
4. Constant folding
5. Constant propagation
6. Dead code elimination

Create the answer key by showing the code before and after each optimization. Note, you should assume that each invocation of your optimizer exhaustively applies the optimization in question before moving on to next optimization and that in each step the optimization must fire at least one time. You can only use all or a subset of {r2, r3, r4, r5} as extra registers, and your starting code *should not contain any dead code*. (20 pts)