Computer Science 320: Final Examination

May 22, 2015

Name ______________________

You have exactly 3 hours to answer the following questions. This final is closed book. One two-sided 8.5 x 11 sheet of notes is permitted. For partial credit, show all work. Put your name on the bottom of every page. Write out and sign the Honor Code pledge on this page before turning in the test.

"I pledge my honor that I have not violated the Honor Code during this examination."

Problem 1: (15%)
Is the following grammar in SLR?

$$
\begin{aligned}
S' &\rightarrow S\$ \\
S &\rightarrow \text{ b } B \text{ c} \\
S &\rightarrow \text{ a } A \text{ c} \\
A &\rightarrow \text{ d} \\
A &\rightarrow A \text{ a } B \\
B &\rightarrow \text{ d}
\end{aligned}
$$

Problem 2: (15%)

Consider the following code:

```
while (ptr = ptr->next) {
  ptr->val = ptr->val + 1
}
```

and the resulting assembly code:

```
   LOOP:
A    r1 = M[r1]
B    branch r1 == 0, EXIT
C    r2 = r1 + 4
D    r3 = M[r2]
E    r4 = r3 + 1
F    M[r2] = r4
G    jump LOOP

   EXIT:
```

A: Draw the dependence graph. Exclude anti-dependence and output-dependences. Draw inter-iteration dependences with solid lines, and draw loop-carried dependences with dashed lines.

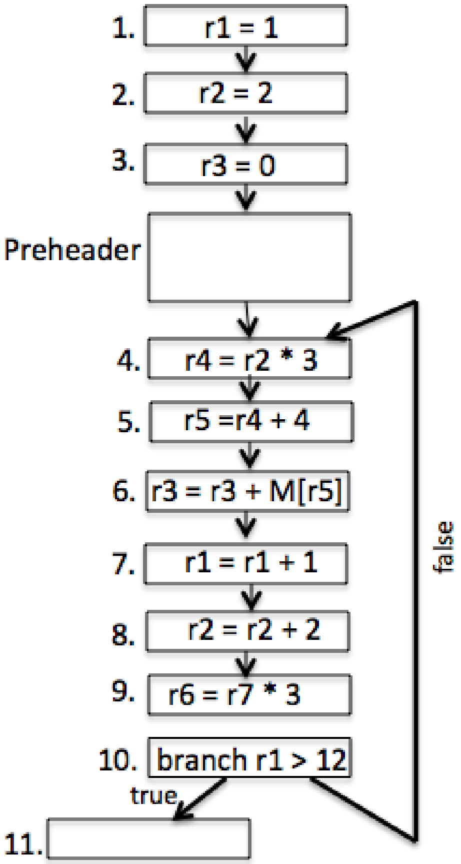
B: Draw the DAG_{SCC}.

C: Apply DSWP for two threads. Use the "produce" instruction to send a register value to the other thread. Use the "consume" instruction to receive a register value from the the other thread. No need for "spawn" since both threads are assumed established.

Problem 3: (15%)

Optimize the following loop using any three different loop-specific optimizations discussed in class. Perform each optimization separately. You may start each from the original version of the code or from the result of prior optimization. Do not unroll the loop. For each optimization :

- State the name of the optimization
- Write out the code of the loop after the optimization, and
- State how many fewer (or more) cycles the optimized version of the code takes relative to the original version. (Assume that multiply and load instructions take 2 cycles. All other instructions take 1. The machine is pipelined and can only start one instruction per cycle.)



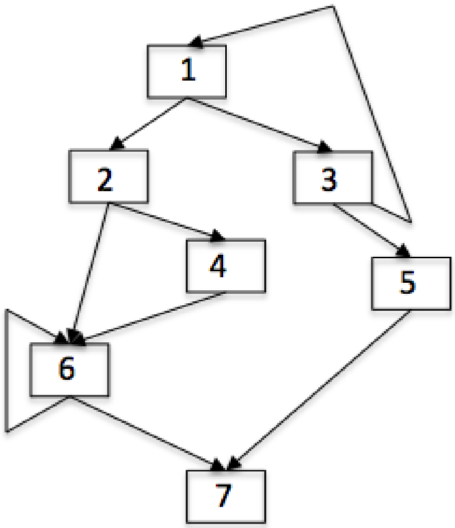
Problem 4: (15%)

Explain how each of these optimizations, especially performed in isolation, could result in decreased performance:

- Loop Unrolling
- Common Subexpression Elimination
- Instruction Scheduling
- Speculative DOALL
- Loop Invariant Code Motion

Problem 5: (10%)

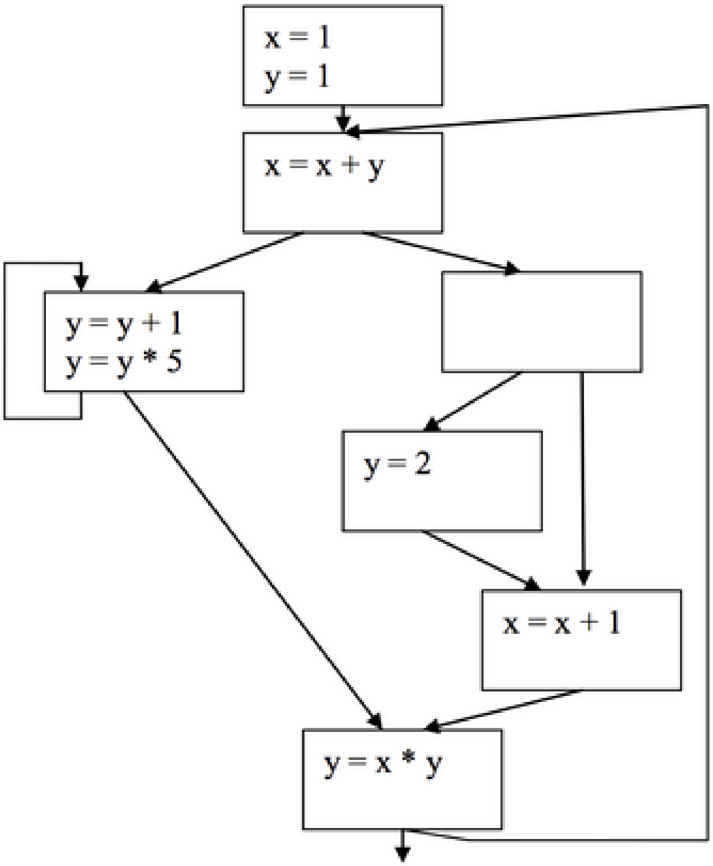
Compute the post dominator (PDOM) set for each basic block in the following control flow graph.



BB	PDOM
1	
2	
3	
4	
5	
6	
7	

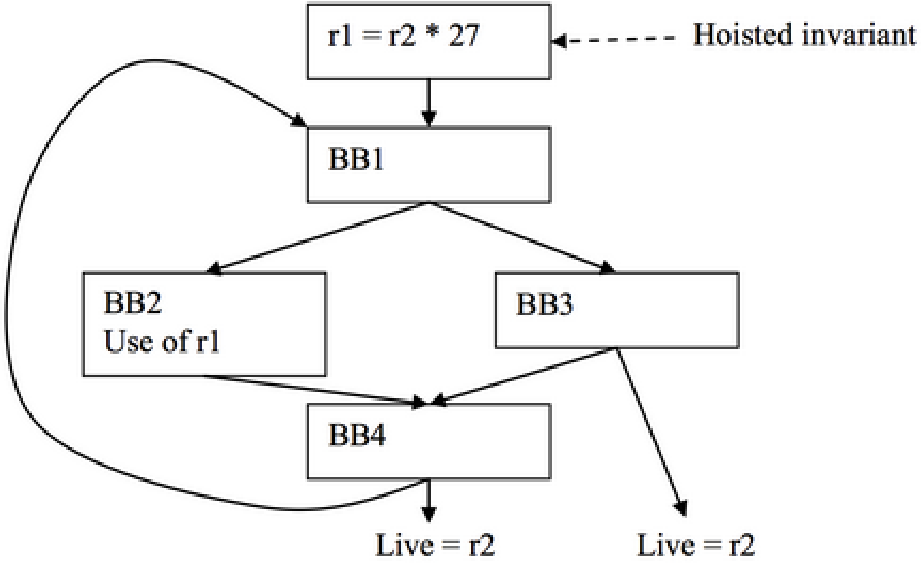
Problem 6: (15%)

Convert the following program segment into single static assignment (SSA) form. You should perform the necessary renames and show the Phi nodes. The number of Phi nodes should be minimized.



Problem 7: (15%)

In the following loop consisting of basic blocks 1-4 with 2 exit points, Loop Invariant Code Motion has been applied to the instruction “r1 = r2 * 27”. Assuming that the optimization was legal, which block(s) could *not* be the original home of the invariant instruction? Explain.


r1 = r2 * 27
Hoisted invariant
BB1
BB2
Use of r1
BB3
BB4
Live = r2
Live = r2