

Optcomp9.tex.

Optimising Compilers

1999

p9q7

AM

### Model Answer

1. Bookwork: Instruction scheduling is the compile-time process whereby instructions are re-arranged so that fewer run-time pipeline stalls occur. It will slow down (only a little one hopes) the compiler but increase execution speed of compiled code.
2. Bookwork: Form the dependency dag of a *basic block* and do a topological sort according to heuristics in the notes. The presented algorithm is  $O(n^2)$ .
- 3(a). True for the naive algorithm presented in part 2, but by delaying choice of physical register for virtual register we might hope to be able to allocated (say) r1 for the last 2 occurrences of r0 and thereby interleave the two loads and stores.
- 3(b) This is arrant nonsense of course. Such a processor may not have delayed loads, but scheduling could act on an occurrence of 4 separate independent consecutive dependency chains of  $n$  instructions (which would be assembled into  $4n-3$  instructions (about 1 instruction per cycle)) into full use (4 exploited parts for each of  $n$  instructions, i.e. 4 instructions per cycle).