Lecture 18 (Part 2)
Global Scheduling

Reading: Chapter 10.4

Assume each clock can execute 2 operations of any kind.

```
if (a==0) goto L
LD R6 <- 0(R1)
nop
B1

e = d + d

LD R7 <- 0(R2)
nop
ST 0(R3) <- R7
L:
```

```
LD R6 <- 0(R1) ; LD R8 <- 0(R4)
LD R7 <- 0(R2)
ADD R8 <- R8,R8 ; BEQZ R6, L
L: ST 0(R5) <- R8
       B1
        B1
L: ST 0(R5) <- R8 ; ST 0(R3) <- R7
```

Result of Code Scheduling

Terminology

Control equivalence:
• Two operations o1 and o2 are control equivalent
  if o1 is executed if and only if o2 is executed.

Control dependence:
• An op o2 is control dependent on op o1
  if the execution of o2 depends on the outcome of o1.

Speculation:
• An operation o is speculatively executed if it
  is executed before all the operations it
  depends on (control-wise) have been executed.

Requirements:
• does not raise an exception
• satisfies data dependences
**Code Motions**

**Goal:** Shorten execution time **probabilistically**

**Moving instructions up:**
- Move instruction to a cut set (from entry)
- Speculation: even when not anticipated.

**Moving instructions down:**
- Move instruction to a cut set (from exit)
- May execute extra instruction
- Can duplicate code

---

**A Note on Data Dependences**

- \( a = 0 \)
- \( a = 1 \)

---

**General-Purpose Applications**

- Lots of data dependences
- Key performance factor: memory latencies
- **Move memory fetches up**
  - Speculative memory fetches can be expensive
- **Control-intensive:** get execution profile
  - **Static estimation**
    - Innermost loops are frequently executed
    - Back edges are likely to be taken
  - Edges that branch to exit and exception routines are not likely to be taken
  - **Dynamic profiling**
    - Instrument code and measure using representative data

**A Basic Global Scheduling Algorithm**

- Schedule innermost loops first
- Only upward code motion
- No creation of copies
- Only one level of speculation
Program Representation

- A region in a control flow graph is:
  - a set of basic blocks and all the edges connecting these blocks,
  - such that control from outside the region must enter through a single entry block.
- A function is represented as a hierarchy of regions
  - The whole control flow graph is a region
  - Each natural loop in the flow graph is a region
  - Natural loops are hierarchically nested
- Schedule regions from inner to outer
  - treat inner loop as a black box unit
    - can schedule around it but not into it
  - ignore all the loop back edges \(\rightarrow\) get an acyclic graph

Algorithm

Compute data dependences;
For each region from inner to outer {
  For each basic block B in prioritized topological order {
    CandBlocks = ControlEquiv(B) \(\cup\) Dominated-Successors(ControlEquiv(B));
    CandInsts = ready operations in CandBlocks;
    For (t = 0, 1, ... until all operations from B are scheduled) {
      For (n in CandInst in priority order) {
        if (n has no resource conflicts at time t) {
          S(t) = (B, t)
          Update resource commitments
          Update data dependences
        }
      }
    }
  }
}

Priority functions: non-speculative before speculative

Extensions

- Prepass before scheduling: loop unrolling
- Especially important to move operation up loop back edges

Summary

- Global scheduling
  - Legal code motions
  - Heuristics