

### Instruction scheduling

1. Assume that your machine has two ALUs, which can execute ADDs in a single cycle, SUBs in two cycles, and *one* of the two ALUs can execute MULs in two cycles. There is also a single LD/ST unit. LOADs take up the ALU for 1 cycle and the LD/ST the next, while STs take up the LD/ST for one cycle. Draw the reservation tables for this machine.
2. (ECE 573 only): How many states would the scheduling automaton for this machine have?
3. Draw the scheduling DAG for the following program:

```
1.  LOAD(A) R1
2.  LOAD(B) R2
3.  LOAD(C) R3
4.  LOAD(D) R4
5.  R5 = R1 + R2
6.  R6 = R5 * R3
7.  R7 = R1 + R6
8.  R8 = R6 + R4
9.  R9 = R8 + R7
10. ST(R9) E;
```

4. Give the schedule you obtain after performing height-based list scheduling on the above code.

### Loop transformations (I)

For the following problems, consider the code below:

```
1.  X = 2;
2.  Y = 10;
3.  if (X < Y) goto 13
4.    A = Y + 10;
5.    B = X * 4 + 10;
6.    Z = 10;
7.    if (B < Z) goto 11
```

```
8.      D = Y + A;
9.      Z = Z - 1;
10.     goto 7;
10.     X = X + 1;
12.     goto 3;
13.     Y = D;
14.     halt;
```

1. Draw the CFG for the code above. Identify the loops in the code.
2. Which statements are loop invariant? Can they be moved outside their enclosing loop? Show the code that results after hoisting any loop invariant code outside the loop.
3. Identify the induction variables in this code. Show the code that results after performing any possible strength reduction.
4. Show the code after performing any possible linear test replacement.

### **Loop transformations (II)**

Consider a machine that *does not have a cache*. For the following loop transformations, explain whether the transformation would still be worth doing or not.

1. Loop interchange
2. Loop fusion
3. Loop unrolling