Problem 1. Software Pipelining

Consider a simple loop below

```c
for(i = 1; i<n; ++i) {
    B[i] = 2 * A[i];
}
```

Assume LD and ST take 1 clock, ADD and MUL take 2 clocks. The machine has two MEM units that can execute one LD and one ST, and two ALU units that can execute ADD and MUL in one clock. The machine can autoincrement address registers.

1. Draw the data dependence graph by showing three types of nodes, LD, ADD, and ST.
2. Label the edges of the dependency graph according to the type of dependency (true dependency, anti-dependency or output dependency)
3. What is the lower bound on the initiation interval?
Problem 2. Data Dependency Analysis

Consider the loop nest below:

```c
for(i = 1; i<n; i++) {
    for(j = i+1; j<m; j++) {
        Z[i, j] = 2 * Y[2*i+1];
    }
}
```

1. Draw the iteration space for this loop.

2. What are the possible data dependencies in this loop? Hint: a data dependency is something of the form “read/write A[i], read/write B[j].”

3. What are the data dependency equations that must be satisfied for the loop to be parallelizable?

4. Is this loop nest 1 or 2-d parallelizable without loop transformations? You do not need to show the exact solution to the equations, but justify your answer.

5. Is this loop nest 1 or 2-d parallelizable with loop transformations? Again, you do not need to show the exact solution, but justify your answer.
Problem 3. Global Instruction Scheduling

SLOWMO is a statically scheduled machine that can only issue one operation every clock. Its memory load operation (denoted by the * dereference operator) has a latency of three clocks; all other operations have a latency of one clock. Consider the following locally scheduled program:

Assume that \( r \) is the only live variable at the end of the program. Each branch in the flow graph is labeled with the likelihood that it is taken dynamically.

1. Assuming that \( p = 0.01 \), what is the best globally scheduled code that you can generate for the above program? You may apply any of the code motions discussed in class and introduce extra nops to fit the schedule. What is the expected execution time of the scheduled program?

2. Repeat part 1 for \( p = 0.99 \).

3. Find the expected execution time of the scheduled program in part 1 and part 2 under \( p = 0.40 \). Are they better than the original program?
Problem 4. More Software Pipelining

Consider the following dependence graph for a single iteration of a loop, with resource constraints:

Instructions A, D and E take 2 clock cycles and B and C take 1 cycle each.

1. What is the bound on the initiation interval T according to the precedence and resource constraints for this program?

2. What is the minimum initiation interval? Show a modulo reservation table for an optimal software pipelined schedule. Also show the code and schedule for an iteration in the source loop.

3. Can the scheduling algorithm described in class produce the optimal schedule for this loop? Discuss whether and how the original algorithm should be improved.