Homework 6

CIS 501, Spring 1998
Due on April 1, 1995

1. For the following code fragment, list the control dependencies. For each control dependency, tell whether the statement can be scheduled before the if statement. Assume that all data references are shown, that all values are defined before use, and that only `b` and `c` are used again after this segment. You may ignore any possible exceptions.

```
if ( a > c ) {
  d = d + 5;                      /*  S1  */
  a = b + d + e;                  /*  S2  */
} else {
  e = e + 2;                      /*  S3  */
  f = f + 2;                      /*  S4  */
  c = c + f;                      /*  S5  */
}
b = a + f;                        /*  S6  */
```

2. Consider the following code fragment:

```
ADD     R3,R2,R4
LW      R4,8(R5)
MULT    R3,R3,R2
ADD     R5,R4,R3
LW      R3,8(R4)
SUB     R6,R6,R2
LW      R7,8(R6)
ADD     R7,R7,R6
```

Assume that:

- Reading register operands requires 1 cycle.
- Once operands are read, execution takes 1 cycle for the ALU, 2 for the Load/Store unit, and 6 for the floating point MULT unit.
- For the static pipeline, at most one new instruction can be initiated per cycle.
- For the dynamic pipelines, at most 2 instructions can be fetched per cycle. Moreover, at most 3 instructions can be issued in a cycle.
- The DLX machine has 1 ALU (integer math unit), 1 MULT/DIV (integer multiply/divide unit) and 1 LSU (load/store unit).
- If a result is ready to write back, you can do so—there are always enough writeback busses.
- Once execution has completed, writing the results back takes one cycle, but the writeback happens at the start of the cycle.
- Bypassing is possible only in the static pipeline. On a dynamic machine, the value isn't available until the writeback phase. Bypassing is possible, but the mechanisms we have presented are NOT enough to manage it. So **do not** try to do bypassing in the dynamic pipelines.
- The first cycle is cycle 0.

For each of the following, draw a timing diagram showing the cycle(s) in which each instruction is fetched, issued and decoded, the operands become available, and the results are written back.

(a) For a simple, nondynamic pipeline.

(b) For a pipeline managed by scoreboarding.

3. Suppose we have a deeply pipelined processor, for which we implement a branch-target buffer for the conditional branches only. Assume that the misprediction penalty is always 4 cycles and the buffer miss penalty is always 3 cycles. Assume a 90% hit rate and a 90% accuracy, and 15% branch frequency. How much faster is the processor with the branch-target buffer versus a processor that has a fixed 2-cycle branch penalty? Assume a base CPI without branch stalls of 1.