EE 4720 **Homework 3** **Due: 13 March 1998**

The program below is referred to in the problems.

```
        !! r4 holds a limit
        !! r5 holds the first array element address
        add     r2, r0, r0    ! Clear sum register.
        add     r5, r10, rll  ! Set r5 to first element.
LOOP:
        lw      r6, 0(r5)
        add     r2, r2, r6
        slt     r3, r2, r4
        addi    r5, r5, #4
        bneq    r3, LOOP
```

The program executes on the DLX Chapter-3 implementation in which the branch address is computed in the ID stage, as shown in *the corrected version* of Figure 3.22 (COPYRIGHT 1990, 1996 MORGAN KAUFMANN PUBLISHERS, INC. ALL RIGHTS RESERVED), to the right. The pipeline also includes bypass (forwarding) paths to the ALU (not shown).

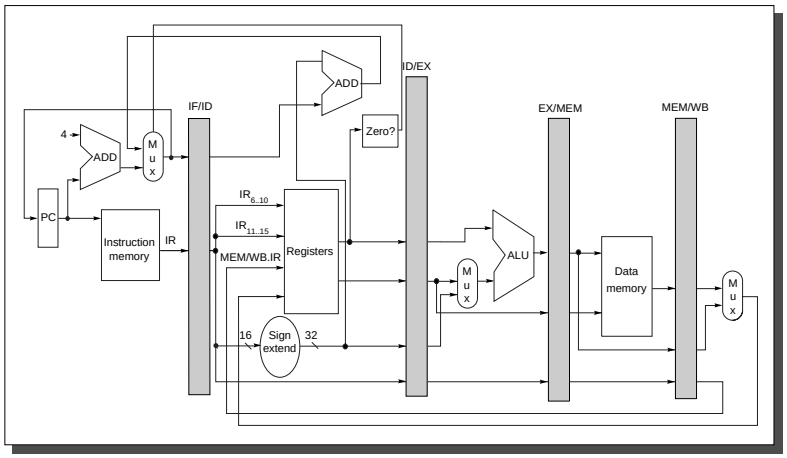

FIGURE 3.22 The stall from branch hazards can be reduced by moving the zero test and branch target calculation into the ID phase of the pipeline.

Problem 1: Draw a pipeline execution diagram showing the first two iterations of the program executing on the implementation above. What is the CPI while executing the loop?

Problem 2: Explain how adding forwarding paths to the ID stage would speed the execution of the branch instruction.

Problem 3: Consider an implementation that uses the ID-stage forwarding paths mentioned in the problem above and which also has a branch delay slot. Re-write the program above so that it executes as fast as possible. Draw a pipeline execution diagram showing the first two iterations and compute the CPI of the loop execution.