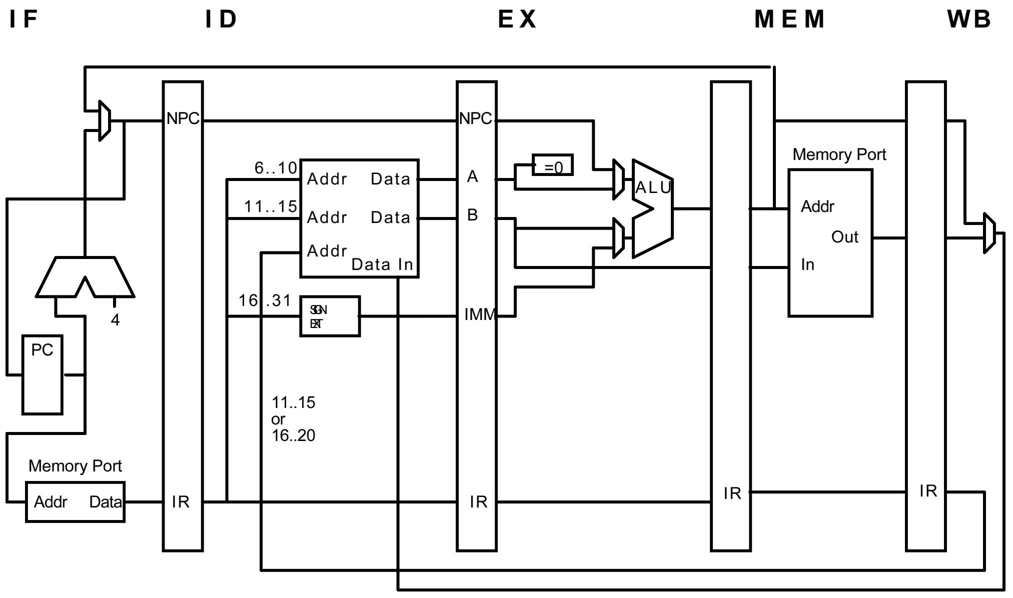
EE 4720 Homework 2 Due: 6 March 1998

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, r11  ! 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:
```

Problem 1: Draw a pipeline execution diagram showing the first three iterations of the program above (assuming that it executes for at least three iterations, of course). The program runs on the DLX implementation shown in Figure 3.4 of the text (and below) in which a register can be read from the register file in the same cycle it was written to the register file. The processor will stall in the face of hazards, but there's no data forwarding or other neat stuff covered later in Chapter 3. Note that the execution of the first iteration is different than subsequent iterations.

Problem 2: Show the contents of the pipeline registers (stage latches) and the register file the first time the addi instruction (just before `bneq`) is in the memory stage. Assume that the first element of the array is 10, the second element is 20, etc. Show only registers that are being used (*e.g.*, don't show `r22`), use a "??" for pipeline register values that cannot be determined or are not being used.



Problem 3: What is the CPI while running the loop above? Ignore initialization and the first iteration.