LSU EE 4720 **Homework 2** Solution **Due: 13 March 2006**

Problem 1: The code fragment below runs on the illustrated implementation. Assume the branch is always taken.

(*a*) Show a pipeline execution diagram covering execution to the beginning of the third iteration of the loop.
See below.

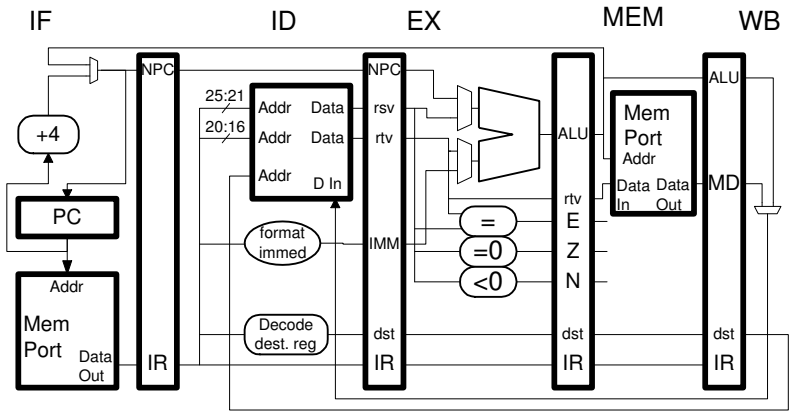
(*b*) What is the CPI for a large number of iterations?

Hint: Pay close attention to dependencies and carefully add the stalls to handle them; also pay close attention to the timing of the branch. Work from the illustrated implementation, **do not** *adapt the solution from a similar past assignment, that would be like preparing for a 10 km run by driving around the jogging trail.*

An iteration has four instructions. The first iteration takes $10 - 0 = 10$ cycles as does the second iteration: $20 - 10 = 10$ cycles. Both the second and third iterations start with the pipeline in the same state (lw in IF, add in ME, bneq in WB) and so the third iteration will be identical as will every subsequent iteration and so the CPI is $\frac{10}{4}$.

```
# Cycle:             0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
LOOP:
 lw $s0, 0($s1)     IF ID EX ME WB
 addi $s3, $s0, 4      IF ID ----> EX ME WB
 bneq $s3, $0 LOOP        IF ----> ID ----> EX ME WB
 add $s1, $s1, $s2                 IF ----> ID EX ME WB
 xor $t0, $t1, $t2                          IF IDx
 or  $t3, $t4, $t5                             IFx
# Cycle:             0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
 lw $s0, 0($s1)                                IF ID EX ME WB
 addi $s3, $s0, 4                                 IF ID ----> EX ME WB
 bneq $s3, $0 LOOP                                   IF ----> ID ----> EX ME WB
 add $s1, $s1, $s2                                             IF ----> ID EX ME WB
 xor $t0, $t1, $t2                                                      IF IDx
 or  $t3, $t4, $t5                                                         IFx

 lw $s0, 0($s1)                                                               IF ID ----> ...
# Cycle:             0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
```



Problem 2: The code fragment below (the same as the one above) runs on the illustrated implementation (different than the one above—and better!). Assume the branch is always taken.

(*a*) Show a pipeline execution diagram covering execution to the beginning of the third iteration of the loop.

See below. Note that there is no bypass for the branch condition.

(*b*) What is the CPI for a large number of iterations?

An iteration is $14 - 7 = 7$ cycles, the CPI is $\frac{7}{4}$.

(*c*) An A points to a wire on the illustration. On the pipeline execution diagram show the value of that wire in every cycle that the corresponding stage holds a "live" instruction.

See diagram. The A points to the integer register number to write. Both the value, and for convenience, the register name are shown. Note that the branch specifies register 0 as a destination, because it does not write any real register.

(*d*) A B points to a wire on the illustration. On the pipeline execution diagram add a row labeled `B`, and on it place an `X` in a cycle if the value on the wire can be changed without changing the way the program executes.

Through B the rs register value from the register file goes to the ALU. It is only used if the instruction uses the rs register value and if that value is not bypassed. A lower-case ex (`x`) is placed in positions where there is no instruction or if the instruction does not use the rs register value (bypassed or note). An upper-case ex (`X`) is placed in positions where the instruction uses a bypassed rs value (the value from the register file is outdated).

```
# Cycle:              0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18
LOOP:
 A:                        16    19       0  17 16    19       0  17 16
                           s0    s3       r0 s1 s0    s3       r0 s1 s0
 B:                  x  x     x  X  x  x  x     X  x  X  x  x  x     X...
 lw $s0, 0($s1)      IF ID EX ME WB
 addi $s3, $s0, 4       IF ID -> EX ME WB
 bneq $s3, $0 LOOP      IF -> ID ----> EX ME WB
 add $s1, $s1, $s2            IF ----> ID EX ME WB
# Cycle:              0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18
 lw $s0, 0($s1)                        IF ID EX ME WB
 addi $s3, $s0, 4                         IF ID -> EX ME WB
 bneq $s3, $0 LOOP                           IF -> ID ----> EX ME WB
 add $s1, $s1, $s2                                 IF ----> ID EX ME WB

 lw $s0, 0($s1)                                                IF ID EX ME WB
# Cycle:              0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18
```