LSU EE 4720 **Homework 6** **Due: 25 April 2003**

Problem 1: Show the execution of the MIPS code fragment below for three iterations on a four-way dynamically machine using method 3 (physical register file). Though the machine is four-way, assume that there can be any number of write-backs per cycle.

- Assume that the branch and branch target are correctly predicted in IF so that when the branch is in ID the predicted target is being fetched.
- The FP multiply functional unit is three stages (M1, M2, and M3) with an initiation interval of 1.
- There are an unlimited number of functional units.

(*a*) Show the pipeline execution diagram, indicate where each instruction commits.

(*b*) Determine the CPI for a large number of iterations. (The method used for statically scheduled systems will work here but will be very inconvenient. There is a much easier way to determine the CPI.)

```
LOOP:  # LOOP = 0x1000
 ldc1 f0, 0(t1)
 mul.d f2, f2, f0
 bneq t1, t2 LOOP
 addi t1, t1, 8
```

Problem 2: The execution of a MIPS program on a one-way dynamically scheduled system is shown below. The value written into the destination register is shown to the right of each instruction. Below the program are tables showing the contents of the ID Map, Commit Map, and Physical Register File (PRF) at each cycle. The tables show initial values (before the first instruction is fetched), in the PRF table the right square bracket "]" indicates that the register is free. (Otherwise the right square bracket shows *when* the register is freed.)

(*a*) Show where each instruction commits.

(*b*) Complete the ID and Commit Map tables.

(*c*) Complete the PRF table. Show the values and use a "[" to indicate when a register is removed from the free list and a "]" to indicate when it is put back in the free list. Be sure to place these in the correct cycle.

```
# Cycle          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 (Result)
lw   r1, 0(r2)   IF ID Q  L1 L2                L2 WB                   (0x100)
ori  r1, r1, 6      IF ID Q                       EX WB                (0x106)
subi r2, r1, 2         IF ID Q                       EX WB             (0x104)
xor  r1, r3, r3           IF ID Q  EX WB                               (0)
addi r2, r1, 0x700           IF ID Q  EX WB                            (0x700)
subi r1, r2, 4                  IF ID Q  EX WB                         (0x6fc)
# Cycle          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
ID Map
r1       96
r2       92
# Cycle          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
Commit Map
r1       96
r2       92
# Cycle          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
Physical Register File
99       112 ]
98       583 ]
97       174 ]
96       309
95       606 ]
94       058 ]
93       285 ]
92      1234
91       518 ]
90       207 ]
# Cycle          0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
```