EE 4720 **Homework 4** **Due: 17 April 2000**

Problem 1: The diagram below shows the execution of code on a dynamically scheduled machine that uses physical register numbers to name destination operands. Show the state of the ID register map, the commit register map, their free lists, and the physical register file for each cycle of the execution below. In the register maps and file show only values related to registers `f0` and `f3`. Initially, `f0`=0, `f1`=10, `f2`=20, etc. Initially, register `f0` is assigned to physical register 12 and `f3` is assigned to physical register 15 (ignore the other architected registers). Initially, both free lists contain physical register numbers $\{7, 8, 9, 10, 11\}$.

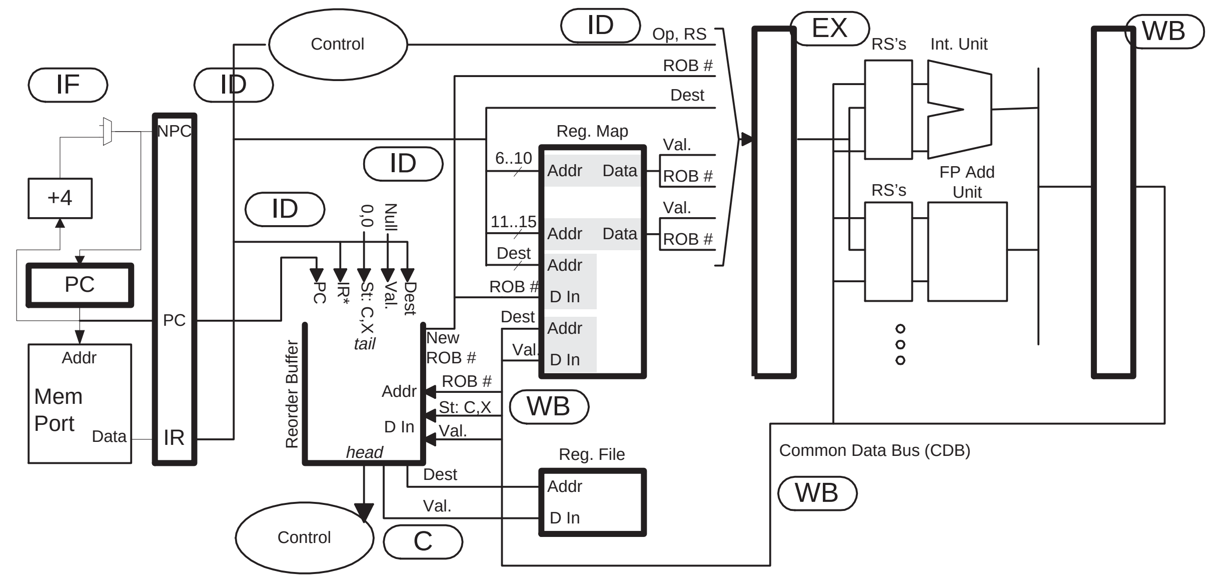
Note: As originally assigned the initial free lists did not contain register 11 and the pipeline execution diagram showed reservation station (RS) segments. Both were mistakes and have been corrected.

```
! Cycle             0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18
multf f0, f1, f2    IF ID Q           M0 M1 M2 M3 M4 M5 WC
addf  f3, f0, f2       IF ID Q                             A0 A1 WC
subf  f0, f4, f5          IF ID Q  A0 A1 WB                         C
addf  f3, f0, f5             IF ID Q        A0 A1 WB                   C
addf  f0, f2, f1                IF ID Q        A0 A1 WB                   C
```

Problem 2: Repeat the problem above assuming that there is an exception in stage A1 of the execution of `addf f3, f0, f5`, as shown below: The solution can start at the cycle in which the tables will differ from the solution above.

```
! Cycle             0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18
multf f0, f1, f2    IF ID Q           M0 M1 M2 M3 M4 M5 WC
addf  f3, f0, f2       IF ID Q                             A0 A1 WC
subf  f0, f4, f5          IF ID Q  A0 A1 WB                         C
addf  f3, f0, f5             IF ID Q        A0*A1*WB                   Cx
addf  f0, f2, f1                IF ID Q        A0 A1 WB
```

Problem 3: The diagram below, of a dynamically scheduled processor, omits hardware that checks whether the register map should be updated in the WB stage. (The hardware was described in class.) Add the hardware to the diagram (at the same level of detail as other parts of the diagram).



Problem 4: Draw a pipeline execution diagram for the DLX code below running on a dynamically scheduled 4-way superscalar implementation with the following characteristics:

- Dynamically scheduled using a reorder buffer to name registers (method 1).
- One load/store functional unit with stages L1 and L2.
- No dynamic (hardware) branch prediction, all branches are predicted not taken. Branch predictor uses the `B` functional unit and must wait for its operand like any other instruction.
- Four integer execution units.

Find the IPC for an execution of a large number of iterations. Show the execution for 14 cycles or until there is enough information to compute the IPC, which ever is shorter.

```
! Note: runs for many iterations.
 add  r3, r0, r0
LOOP:! LOOP = 0x1000
 lw   r1, 4(r2)
 add  r3, r3, r1
 lw   r2, 8(r2)
 bneq r2, LOOP
 xor  r0, r0, r0
```

Problem 5: Repeat the problem above when the branch is statically predicted as taken and the branch target is computed in the ID stage.

Problem 6: Repeat the superscalar problem when the branch is statically predicted taken and in which the address of LOOP it 0x1004.

```
! Note: runs for many iterations.
 add  r3, r0, r0
LOOP:! LOOP = 0x1004
 lw   r1, 4(r2)
 add  r3, r3, r1
 lw   r2, 8(r2)
 bneq r2, LOOP
 xor  r0, r0, r0
```