ENGN 2910A Homework 06 (80 points) – Due Date: Nov 12 2015

Professor: Sherief Reda

School of Engineering, Brown University

---

1. (20 points) Identify all the hazards (RAW, WAW, WAR) that could emerge from executing the following code on an OOO execution engine.

```
LD      F2, 0(Rx)
MULTD   F2, F0, F2
DIVD    F8, F2, F0
LD      F4, 0(Ry)
ADDD    F4, F0, F4
ADDD    F10, F8, F2
SD      F4, 0(Ry)
ADDI    Rx, Rx, 8
ADDI    Ry, Ry, 8
SUB     R20, R4, Rx
```

2. (30 points) Illustrate on a cycle-by-cycle basis the execution of the following program on a dynamic execution engine with a ROB to support speculative execution and precise exception. Assume a dual dispatch and a dual common data bus (CDBs). Add latency is two cycles, and multiply latency is three cycles. An instruction can begin execution in the same cycle that is dispatches assuming all dependences are satisfied. Show the content of ROB, reservation stations (RS), register file (RF) and the register alias table (RAT) for each cycle. Indicate which instruction is executing in each functional unit in each cycle. Also indicate any results forwarding across the CDBs.

```
w:  R4  ←  R0+R8
x:  R2  ←  R0*R4
y:  R4  ←  R4+R8
z:  R8  ←  R4*R2
```

3. (30 points) Consider the execution of the following loop, which increments each element of an integer array, on a two-issue processor, once without speculation and once with speculation:

```
LOOP:  LD       R2, 0(R1)      ; R2 = array element
       DADDIU   R2, R2, #1     ; increment R2
       SD       R2, 0(R1)      ; store result
       DADDIU   R1, R1, #8     ; increment pointer
       BNE      R2, R3, LOOP   ; branch if not last statement
```

Assume that there are separate integer functional units for effective address calculation, for ALU operations, and for branch condition evaluation with execution latency of 1 cycle. Create a

table for the first three iterations of this loop for both processors. Assume that up to to two instructions of any type can retire per clock cycle. I have the first few entries for both cases. Note that the execution of a load and store instruction means to compute the effective address.

Iteration number	Instruction	Dispatches clock cycle number	Executes at clock cycle number	Memory access at clock cycle number	Write CDB at clock cycle number
1	LD R2, 0(R1)	1	2	3	4
1	DADDIU R2, R2, #1	1	5		6
1	SD R2, 0(R1)	2	3	7	
1	DADDIU R1, R1 #8	2	3		4
⋮	⋮	⋮	⋮	⋮	⋮

Table 1: Non-speculative execution

Iteration number	Instruction	Dispatches clock cycle number	Executes at clock cycle number	Memory access at clock cycle number	Write CDB at clock cycle number	retires at clock cycle number
1	LD R2, 0(R1)	1	2	3	4	5
1	DADDIU R2, R2, #1	1	5		6	7
1	SD R2, 0(R1)	2	3			7
1	DADDIU R1, R1 #8	2	3		4	8
⋮	⋮	⋮	⋮	⋮	⋮	

Table 2: speculative execution