Computer Design and Organization
Assignment #3

Due: Wednesday October 30

The purpose of this third assignment is to test your understanding of Tomasulo's algorithm and of its extension using a reorder buffer. This paper and pencil assignment should be done individually.

The code that you'll be studying is the well-known SAXPY loop used in linear algebra. SAXPY stands for the vector computation $S = a \times X + Y$ where S, X, Y are vectors of the same length and a is a scalar. In fact the loop for this exercise is $Y = a \times X + Y$. Your book calls it DAXPY where D stands for double precision. You can ignore this fact and assume that all operations are single-precision (i.e., use single floating-point registers).

Your assignment is to do **Problem 3.6 (a) and (c)** in your textbook.

Note that there are inconsistencies in the statement of the problem. So, assume that all functional units are pipelined with the latencies given in Figure 3.63. Ignore the "cycles in EX" column of Figure 3.62. If there is a conflict to access the CBD, priority is given in this order: F-P multiply, F-P add, Load, integer unit. The F-P add and the integer unit are considered busy if they cannot broadcast their result. F-P store does not require the CDB.

A reservation station for an instruction is released in the cycle following the cycle in which the WR occurs for that instruction. To see why, assume two reservation stations for unit xyz with operation FOO. If the program contains the segment of code:
FOO R1,R1,R8
FOO R2,R2,R9
FOO R3,R1,R2
At the third instruction, the register status table would say that R1's value is coming from the xyz1 reservation station and R2's value from the Integer 2 reservation station. We cannot say R3's value is coming from the Integer 1 reservation station because R1 is listening for that on the CDB!

You can assume an unlimited instruction queue.

A result broadcast at cycle i is read during the same cycle, i.e. an instruction dependent only on this result can start executing at cycle $i + 1$.

If there are still some possibilities left for misinterpretation or confusion, please holler

For part (a), the timings should be shown in a table with the following format (in the comments column, indicate the reason for a stall of the corresponding instruction):

Instruction	Unit + Res. Station		Cycle Number			Comments
		IS	EX	Mem	WR	
LD F2,0(r1)	Load - 1	1	2	3	4	

The status of non-empty reservation stations should be shown in a table with the following format:

Indicate the status of the floating-point registers awaiting a pending result with a table like:

Unit + Res. Station	Busy	V_j	V_k	Q_j	Q_k

Field	F#	F#	etc...	F#

In these tables a tag should appear without "()", e.g., Load - 1, and a value that comes from a functional unit or a register should appear within "()", e.g., (Load - 1).

For part (c), your first table should have one additional column for "Commit" as in:

Instruction	Unit + Res. Station		Cycle Number				Comments
		IS	EX	Mem	WR	Commit	
LD F2,0(r1)	Load - 1	1	2	3	4	5	

Show also the contents of the reorder buffer.