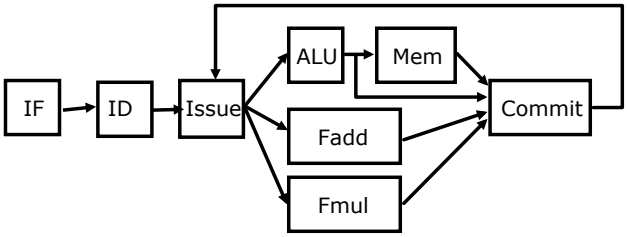
CSE 490/590 Computer Architecture
Homework 2

1. Suppose that you have the following out-of-order datapath with 1-cycle ALU, 2-cycle Mem, 3-cycle Fadd, 5-cycle Fmul, no branch prediction, and in-order fetch and commit.



Consider the following sequence of instructions.

Instruction Number	Instruction
I_1	LD F4, 0 (R1)
I_2	LD F3, 0 (R2)
I_3	FADD F6, F3, F4
I_4	FMUL F1, F6, F3
I_5	LD F5, 0 (R3)
I_6	FADD F2, F5, F4
I_7	FMUL F5, F2, F5
I_8	FADD F3, F1, F4
I_9	FMUL F5, F3, F2
I_{10}	FADD F6, F2, F4

Fill in the ROB and renaming table (next page).

Tag	op	dst	src1	src2
T1	LD	T1	R1	0
T2	LD			
T3	FADD			
T4	FMUL			
T5	LD			
T6	FADD			
T7	FMUL			
T8	FADD			
T9	FMUL			
T10	FADD			

	R1	R2	R3	F1	F2	F3	F4	F5	F6
I_1							T1		
I_2									
I_3									
I_4									
I_5									
I_6									
I_7									
I_8									
I_9									
I_{10}									

2. Consider the following instructions. Assume that the initial values for R1, R2, and R3 are all 0.

```
loop:
        SUBI R2, R1, 2
        BNEZ R2, target1
        ADDI R3, R3, 1
target1:
        ADDI R1, R1, 1
        SUBI R4, R1, 3
        BNEZ R4, loop
```

(a) Explain what the code does.

(b) Change the code to minimize the number of registers necessary.

(c) Assume that we have a 1-bit branch predictor that stores the result of the last branch and makes the prediction based on the result, i.e., the prediction is "take" if the last branch was taken and the other way round for "not take". Show the results of all predictions throughout the execution.

(d) Assume that we have one 2-bit branch predictor. Show the results of all predictions throughout the execution.

(e) Assume that we have four 2-bit branch predictors per branch instruction as well as one 2-bit shift register that stores the result of the last two branch instructions (i.e., we have a two-level branch predictor). Show the results of all predictions throughout the execution.

3. (Example on p.77) Consider the following code:

```
loop:
        LD F0, 0(R1)
        FADD F4, F0, F2
        ST F4, 0(R1)
        ADDI R1, R1, -8
        BNE R1, R2 loop
```

Show how to unroll the loop so that there are four copies of the loop body, assuming that R1 - R2 is initially a multiple of 32, which means that the number of loop iterations is a multiple of 4. Eliminate any obviously redundant computations and do not reuse any of the registers.

4. (Example on p.116) Suppose we have a VLIW that could issue two memory references, two FP operations, and one integer operation or branch in every clock cycle. Show an unrolled version of the loop $x[i] = x[i] + s$ (see p.76 for the MIPS code) for such a processor. Unroll as many times as necessary to eliminate any stalls. Ignore delayed branches.

5. Suppose that you have a multithreading single-issue in-order datapath with 1-cycle ALU, 2-cycle Mem, 3-cycle Fadd, 5-cycle Fmul, no branch prediction, and in-order fetch and commit. Consider the following instructions:

```
loop:
        LD F0, 0(R1)
        FADD F3, F3, F2
        ADDI R1, R1, -8
        BNE F0, F2 loop
```

What is the minimum number of threads necessary to fully utilize the datapath for each of the following strategies?

(a) Fixed switching: the CPU switches to a different thread every cycle in a round-robin fashion.

(b) Data-dependent switching: the CPU switches to a different thread when an instruction cannot proceed due to a dependency.