

Lecture 9 Out-of-order Execution Exercises

Q1:

Considering the following code, how many cycles will one iteration (execute from `lw` to `bnez`) take in a pipeline with out-of-order execution capability?

```
loop:    lw    r2, 40(r6)
         mult  r4, r3, r2
         sw    r4, 80(r7)
         addi  r6, r6, 4
         addi  r7, r7, 4
         addi  r1, r1, -1
         bnez  r1, loop
```

Assumptions:

- The pipeline has all kinds of data bypassing available.
- The pipeline stalls if a structural hazard would occur.
- The pipeline has branch prediction with 100% accuracy.
- `mult` has 4 pipelined execution stages P1, P2, P3, P4, which takes 4 cycles in total; `add` has a 1-cycle eXecution stage. `mult` and `add` use different ALUs.
- `mult` does not have M stage; all other instructions have M stage.
- `bnez` comparison is done in the **Decode** stage (rather than X stage).

Insn	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
<code>lw r2, 40(r6)</code>																				
<code>mult r4, r3, r2</code>																				
<code>sw r4, 80(r7)</code>																				
<code>addi r6, r6, 4</code>																				
<code>addi r7, r7, 4</code>																				
<code>addi r1, r1, -1</code>																				
<code>bnez r1, loop</code>																				
<code>lw r2, 40(r6)</code>																				

Solution: 14 cycles

/ represents waiting in reservation station

// represents waiting in reorder buffer

Insn	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
lw r2, 40(r6)	F	D	X	M	W																											
mult r4, r3, r2		F	D	/	P1	P2	P3	P4	W																							
sw r4, 80(r7)			F	D	/	/	/	X	M	W																						
addi r6, r6, 4				F	D	X	M	//	//	//	W																					
addi r7, r7, 4					F	D	X	M	//	//	//	W																				
addi r1, r1, -1						F	D	/	X	M	//	//	W																			
bnez r1, loop							F	D	D	D	X	M	//	W																		
lw r2, 40(r6)										F	D	X	M	//	W																	
mult r4, r3, r2											F	D	/	P1	P2	P3	P4	W														
sw r4, 80(r7)												F	D	/	/	/	X	M	W													
addi r6, r6, 4													F	D	X	M	//	//	//	W												
addi r7, r7, 4														F	D	X	M	//	//	//	W											
addi r1, r1, -1															F	D	/	X	M	//	//	W										
bnez r1, loop																F	D	D	D	X	M	//	W									
lw r2, 40(r6)																		F	D	X	M	//	W									

Q2:

With the code and pipeline in Q1, how many cycles will five iterations (execute from `lw` to `bnez` for **five** times) take?

Solution: 50 cycles

As given that “The pipeline has branch prediction with 100% accuracy”, the branch predictor is supposed to always predict “taken” during the five iterations. Therefore, the pipeline diagram of the second and later iterations should look like the above.

As the next iteration of the loop starts to fetch at cycle 10, one iteration with 14 cycles in total (see above pipeline diagram) can be divided into $1 * 9 + 5$ cycles, where the 5 cycles is the tail

Therefore, five iterations = $5 * 9$ cycles + 5 = 50 cycles