CSCI 4591: Computer Architecture
HW Assignment # 7
Due Date: April 11, 2021 @ 11:55 PM

1) In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Also, assume that instructions executed by the processor are broken down as follows:

ALU/Logic	Jump/Branch	LDUR	STUR
45%	20%	20%	15%

1. What is the clock cycle time in a pipelined and non-pipelined processor?
    - pipeline = max = 350 ps
        - n instructions = 350n + 1250 ps ??
    - non pipeline = 250 + 350 + 150 + 300 + 200 = 1250ps
        - n instructions 1250 ps

2. What is the total latency of an LDUR instruction in a pipelined and non-pipelined processor?
    - pipeline = 350 * 5 = 1750ps
    - non pipeline = 250 + 350 + 150 + 300 + 200 = 1250ps

3. If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?
    - I would split the ID of 350ps to 175ps each or 200/150. There are many ways to split the ID stage . The new clock cycle time would have a pipeline of 300 ps from MEM

4. Assuming there are no stalls or hazards, what is the utilization of the data memory?
   LDUR + STUR = 20% + 15% = 35%
   Only those two go through DM, the rest pass through it.

5. Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?
   LDUR + ALU/Logic = 20% + 45% = 65%

2) Assume that R1 is initialized to 11 and R2 is initialized to 22. Suppose you executed the code below on a version of the pipeline that does not handle data hazards (i.e., the programmer is responsible for addressing data hazards by inserting NOP instructions where necessary) and we forgot to add the required NOP. What would the final values of register R5 be? Assume the register file is written at the beginning of the cycle and read at the end of a cycle. Therefore, an ID stage will return the results of a WB state occurring during the same cycle.

solve it as normal

No NOPs needed since each of the ALUs can pass off the information down to the next code without issue.

ADDI R1, R2, #5

- r1 ← r2 + 5 = 27

ADD R3, R1, R2

- r3 ← 27 + 22 = 49

ADDI R4, R1, #15

- r4 ← 27 + 15 = 42

ADD R5, R1, R1

- r5 ← 27 + 27 = 54

3) Add NOP instructions to the code below so that it will run correctly on a pipeline that does not handle data hazards.

```
ADDI R1, R2, #5
NOP
NOP
NOP
ADD R3, R1, R2
ADDI R4, R1, #15
ADD R5, R3, R2
```

4) Consider the fragment of LEGv8 assembly below:

```
STUR R16, [R6,#12]
LDUR R16, [R6, #8]
SUB R7, R5, R4
CBZ R7, Label
ADD R5, R1, R4
SUB R5, R15, R4
```

Suppose we modify the pipeline so that it has only one memory (that handles both instructions and data). In this case, there will be a structural hazard every time a program needs to fetch an instruction during the same cycle in which another instruction accesses data.

CBZ needs R7, when R7 is in SUB, stall
Since CBZ is stalled, ADD is stalled
SUB is stalled since it needs R5 and it is in ADD
All of the stalls are in the first position, so it will be stalled in the initial steps

1. Draw a pipeline diagram to show where the code above will stall.
   Stall = X

	1	2	3	4	5	6	7	8	9	10	11	12
STUR	IF	ID	EX	MEM	WB							
LDUR		IF	ID	EX	MEM	WB						
SUB			IF	ID	EX	MEM	WB					
CBZ				X	X	IF	ID	EX	MEM	WB		
ADD					X	X	IF	ID	EX	MEM	WB	
SUB						X	X	IF	ID	EX	MEM	WB

2. In general, is it possible to reduce the number of stalls/NOPs resulting from this structural hazard by reordering code?
   Yes

SUB R7, R5, R4
CBZ R7, Label
ADD R5, R1, R4
SUB R5, R15, R4
STUR R16, [R6,#12]
LDUR R16, [R6, #8]

3. Must this structural hazard be handled in hardware? We have seen that data hazards can be eliminated by adding NOPs to the code. Can you do the same with this structural hazard? If so, explain how. If not, explain why not.

STUR R16, [R6,#12]
LDUR R16, [R6, #8]
SUB R7, R5, R4
NOP
NOP

CBZ R7, Label
ADD R5, R1, R4
SUB R5, R15, R4

4. Approximately how many stalls would you expect this structural hazard to generate in a typical program? (Use the instruction mix from Exercise 1)
    - 2 stalls

5) Problems in this exercise refer to the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath:

```
ADD R5, R2, R1
NOP
NOP
LDUR R3, [R5, #4]
LDUR R2, [R2, #0]
NOP
ORR R3, R5, R3
NOP
NOP
STUR R3, [R5,#0]
```

1. If there is no forwarding or hazard detection, insert NOPs to ensure correct execution.

2. Now, change and/or rearrange the code to minimize the number of NOPs needed. You can assume register R7 can be used to hold temporary values in your modified code.
   if we changed it to no code, then no NOPs necessary lolol or

   group LDURs together, STUR at the end
   ADD
   LDUR
   LDUR
   LDUR
   STUR