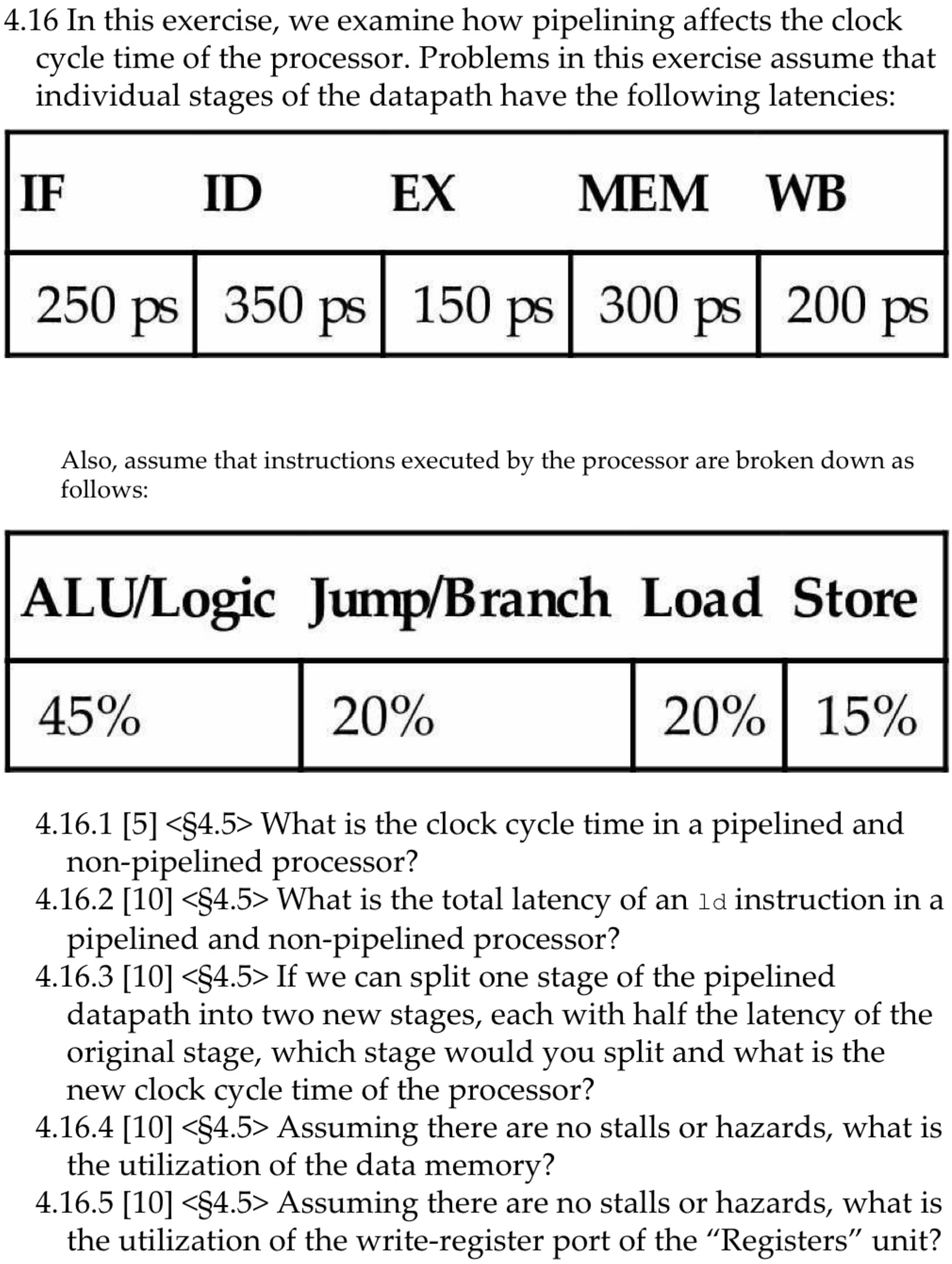
COD HW 5

吴骏东

PB20111699

2022.5.2

4.16



4.16 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
250 ps	350 ps	150 ps	300 ps	200 ps

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

ALU/Logic	Jump/Branch	Load	Store
45%	20%	20%	15%

4.16.1 [5] <§4.5> What is the clock cycle time in a pipelined and non-pipelined processor?

4.16.2 [10] <§4.5> What is the total latency of an `ld` instruction in a pipelined and non-pipelined processor?

4.16.3 [10] <§4.5> 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?

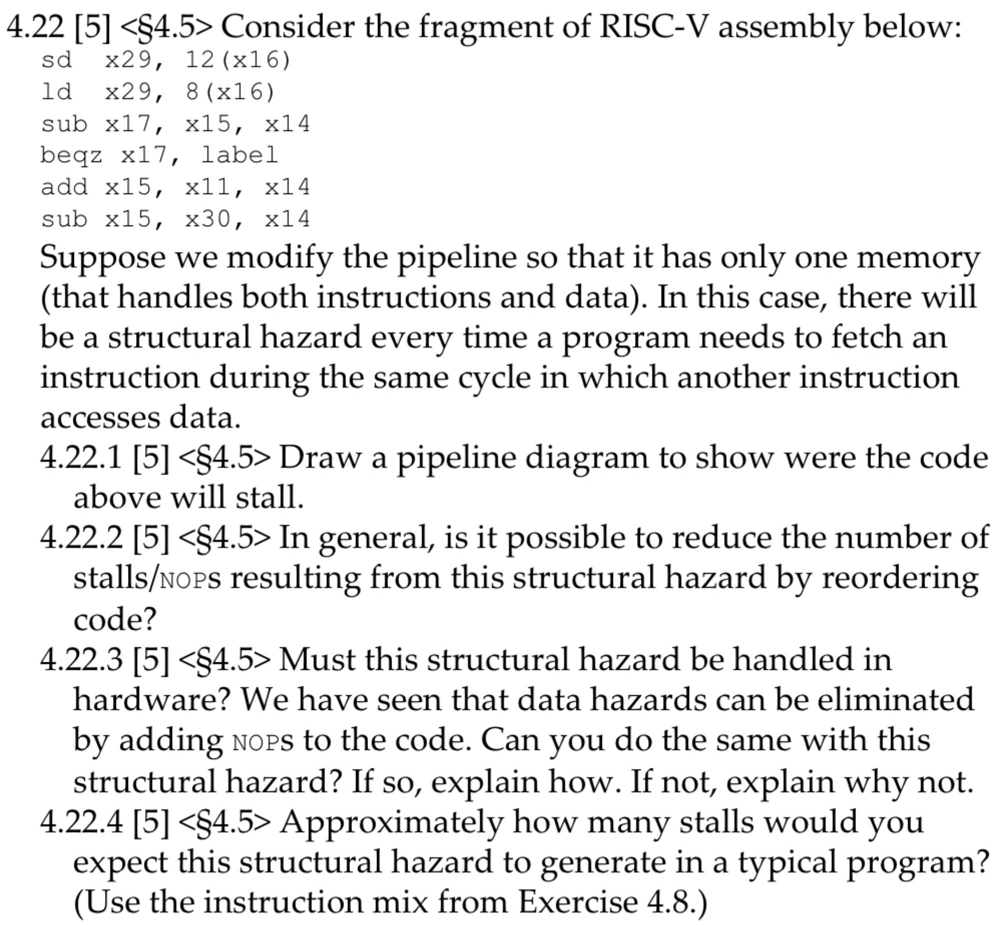
4.16.4 [10] <§4.5> Assuming there are no stalls or hazards, what is the utilization of the data memory?

4.16.5 [10] <§4.5> Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?

4.16.4 数据存储的利用率为 $\frac{\text{Load} + \text{Store}}{5} = 7\%$

4.16.5 寄存器堆写回的利用率为 $\frac{\text{ALU} + \text{Load}}{5} = 13\%$

4.22



4.22 [5] <§4.5> Consider the fragment of RISC-V assembly below:

```
sd  x29, 12(x16)
ld  x29, 8(x16)
sub x17, x15, x14
beqz x17, label
add x15, x11, x14
sub x15, x30, x14
```

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.

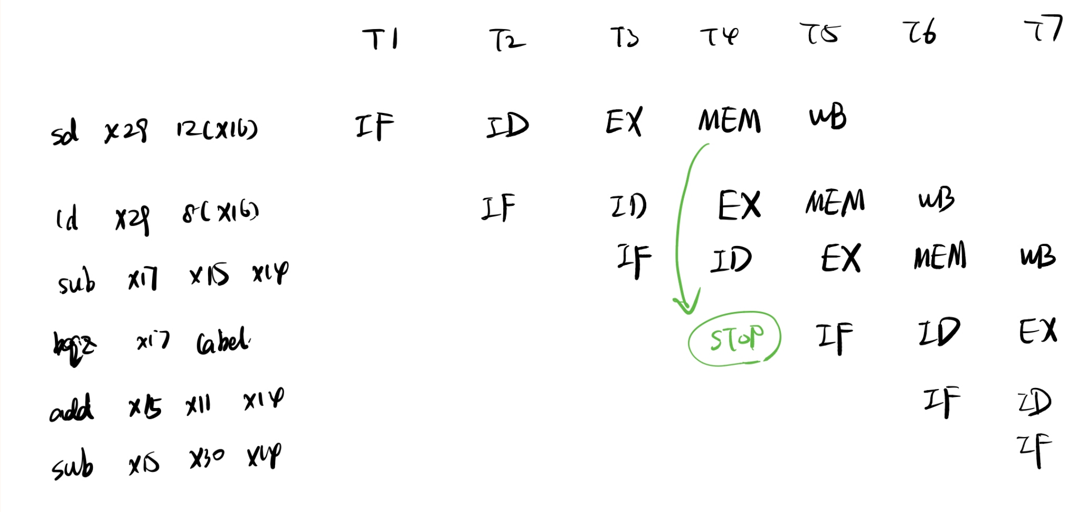
4.22.1 [5] <§4.5> Draw a pipeline diagram to show were the code above will stall.

4.22.2 [5] <§4.5> In general, is it possible to reduce the number of stalls/NOPs resulting from this structural hazard by reordering code?

4.22.3 [5] <§4.5> 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.

4.22.4 [5] <§4.5> Approximately how many stalls would you expect this structural hazard to generate in a typical program? (Use the instruction mix from Exercise 4.8.)

4.22.1 如下图所示，在 `sd x29 12(x16)` 执行到 MEM 段时，需要将对应指令的 IF 段停顿一个周期。



	T1	T2	T3	T4	T5	T6	T7
sd x29 12(x16)	IF	ID	EX	MEM	WB		
ld x29 8(x16)		IF	ID	EX	MEM	WB	
sub x17 x15 x14			IF	ID	EX	MEM	WB
beqz x17 label				STOP	IF	ID	EX
add x15 x11 x14						IF	ID
sub x15 x30 x14							IF

4.22.2 不可以。因为所有的指令都会经历 IF 阶段。无论怎样改变顺序，此时如果需要写入存储器都必须插入停顿。

4.22.3 必须由硬件解决。由于单存储器只有一个地址输入端口，即使在冲突时插入空指令也会造成该输入端口多驱动（来自 IF 和 MEM 的双重地址）。所以该结构冒险必须用硬件解决。

4.22.4 需要产生约 11/5 = 2.2%的停顿。

4.25

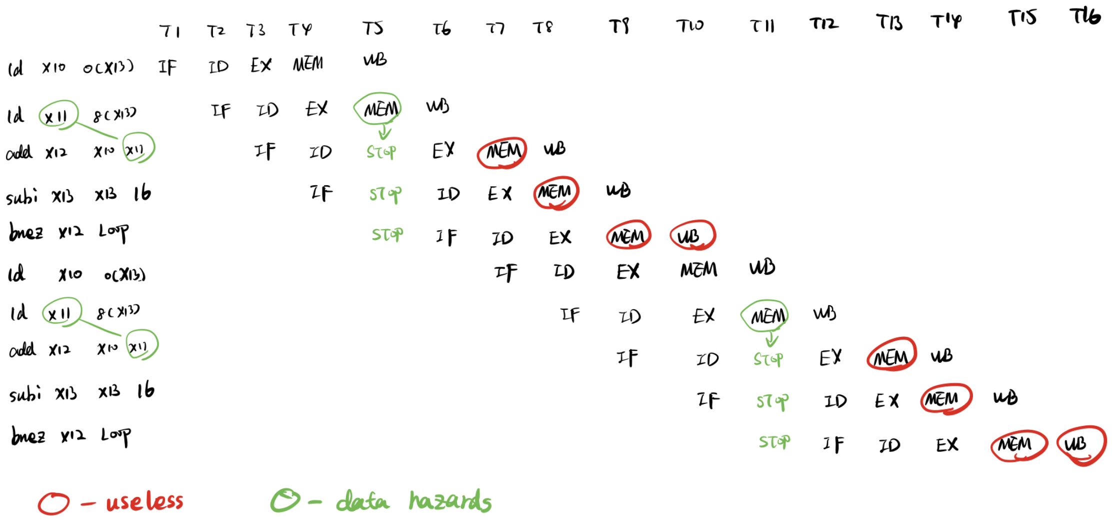
4.25 Consider the following loop.

```
LOOP: ldx10, 0(x13)
 ldx11, 8(x13)
 addx12, x10, x11
 subix13, x13, 16
 bnezx12, LOOP
```

Assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, that the pipeline has full forwarding support, and that branches are resolved in the EX (as opposed to the ID) stage.

4.25.1 [10] <§4.7> Show a pipeline execution diagram for the first two iterations of this loop.

4.25.2 [10] <§4.7> Mark pipeline stages that do not perform useful work. How often while the pipeline is full do we have a cycle in which all five pipeline stages are doing useful work? (Begin with the cycle during which the subi is in the IF stage. End with the cycle during which the bnez is in the IF stage.)



	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16
ld x10 0(x13)	IF	ID	EX	MEM	WB											
ld x11 8(x13)		IF	ID	EX	MEM	WB										
add x12 x10 x11			IF	ID	STOP	EX	MEM	WB								
subi x13 x13 16				IF	STOP	ID	EX	MEM	WB							
bnez x12 Loop					STOP	IF	ID	EX	MEM	WB						
ld x10 0(x13)							IF	ID	EX	MEM	WB					
ld x11 8(x13)								IF	ID	EX	MEM	WB				
add x12 x10 x11									IF	ID	STOP	EX	MEM	WB		
subi x13 x13 16										IF	STOP	ID	EX	MEM	WB	
bnez x12 Loop											STOP	IF	ID	EX	MEM	WB

○ – useless　　○ – data hazards

4.25 如上图所示。**无论什么时候，五级流水线都不会全部在进行有用操作。**