20171645 박찬우

1. critical path:

I-Mem	400 ps
MUX	100 ps
Regs	220 ps
Mux	100 ps
ALU	180 ps
D-Mem	1000 ps
MUX	100 ps
	2100 ps

, control unit은 I-Mem 이 읽힌 다음에 Memwrite control block에서 Mem write signal 생성, 따라서 I-Mem 생략 가능

$2100 - 400 = 1700\,ps$

2.

2-1) pipelined : 가장 큰 250 ps

nonpipelined : 모두 더한 1000 ps

2-2) lw는 IF, ID, EX, MEM, WB 를 모두 거치므로

pipelined : $250 \times 5 = 1250\,ps$

nonpipelined : 1000 ps

2-3) 가장 큰 IF를 반으로 split 하게됨, 이에 따라 새 clock cycle은 240 ps

2-4) data memory를 사용하는 instruction은 lw, sw

∴ utilization of data memory = 25 + 20 = 45%

2-5) write-register 를 하려면 operation이 필요, ∴ ALU 사용

register에 write 하므로 lw

∴ 30 + 25 = 55%

3.

두번째 Instruction 인 sw $4, -16($4) 와 세번째 lw $3, -20($4) 는 dependency 관계가 없으므로 한 cycle을 stall 시켜줘야함.

4.

IF	ID	EX	MEM	WB					
	IF	ID	EX	MEM	WB				
		XXX	IF	ID	EX	MEM	WB		
				IF	ID	EX	MEM	WB	
					IF	ID	EX	MEM	WB

4.

add \$1, \$2, \$3 ··· I_1

sw \$2, 0(\$1) ··· I_2

lw \$1, 4(\$2) ··· I_3

add \$2, \$2, \$1 ··· I_4

4-1) RAW : \$1 ($I_1 \rightarrow I_2$) \$1 ($I_3 \rightarrow I_4$)
WAR : \$2 ($I_1 \rightarrow I_4$) \$2 ($I_2 \rightarrow I_4$) \$2 ($I_3 \rightarrow I_4$) \$1 ($I_1 \rightarrow I_3$), \$1 ($I_2 \rightarrow I_3$)
WAW : \$1 ($I_1 \rightarrow I_3$)

4-2) with forwarding, hazard는 전부 해결됨
without forwarding, RAW dependency 들이 data hazard가 된다. (\$1($I_1 \rightarrow I_2$), \$1 ($I_3 \rightarrow I_4$))

5. data dependency는

① add \$1, \$5, \$3
sw \$1, 0(\$2)
에서 RAW, 즉 add의 write 과정이 완전히 끝난후 read가 수행되어야 함, nop 2

② lw \$1, 4(\$2)
add \$5, \$5, \$1
에서 RAW, load 다음 Raw dependency는 stall이 2번 일어남, nop 2

∴ add \$1, \$5, \$3
nop
nop
sw \$1, 0(\$2)
lw \$1, 4(\$2)
nop
nop
add \$5, \$5, \$1
sw \$1, 0(\$2)

6.

6-1)

	1	2	3	4	5	6	7	8	9	10	11	12
add	IF	ID	EX	MEM	WB							
sw		IF	ID	EX	MEM	WB						
add			IF	ID	EX	MEM	WB					
beq				IF	ID	EX	MEM	WB				
add					X	X	IF	ID	EX	MEM	WB	
sw								IF	ID	EX	MEM	WB

(branch에서 stall이
2번 일어나게 된다.)

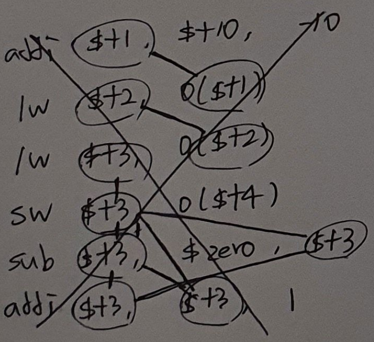
6-2)

	1	2	3	4	5	6	7	8	9	10	11
add	IF	ID	EX	MEM	WB						
sw		IF	ID	EX	MEM	WB					
add			IF	ID	EX	MEM	WB				
beq				IF	ID	EX	MEM	WB			
add					IF	ID	EX	MEM	WB		
sw						X	IF	ID	EX	MEM	WB

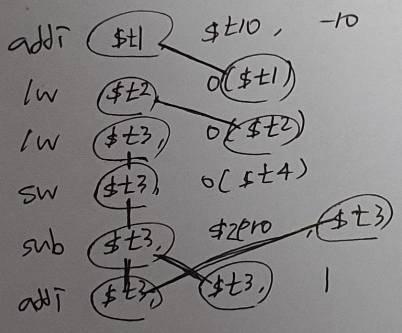
delay slot이 사용되므로
branch 다음 instruction 가 delay slot을 사용하는
instruction 이 되므로
add에서의 stall이 사라짐.

7.

7-1)



```
addi $t1, $t10, -10
lw   $t2, 0($t1)
lw   $t3, 0($t2)
sw   $t3, 0($t4)
sub  $t3, $zero, $t3
addi $t3, $t3, 1
```



```
addi $t1, $t10, -10
lw   $t2, 0($t1)
lw   $t3, 0($t2)
sw   $t3, 0($t4)
sub  $t3, $zero, $t3
addi $t3, $t3, 1
```

7-2)

1	2	3	4	5	6	7	8	9	10	11	12
IF	ID	EX	MEM	WD							
	IF	ID	↓ EX	MEM	WD						
		IF	--	ID	↓ EX	MEM	WD				
			--	IF	--	ID	↓ EX	MEM	WD		
			--		--	IF	ID	EX	MEM	WD	
			--		--		IF	ID	↓ EX	MEM	WD