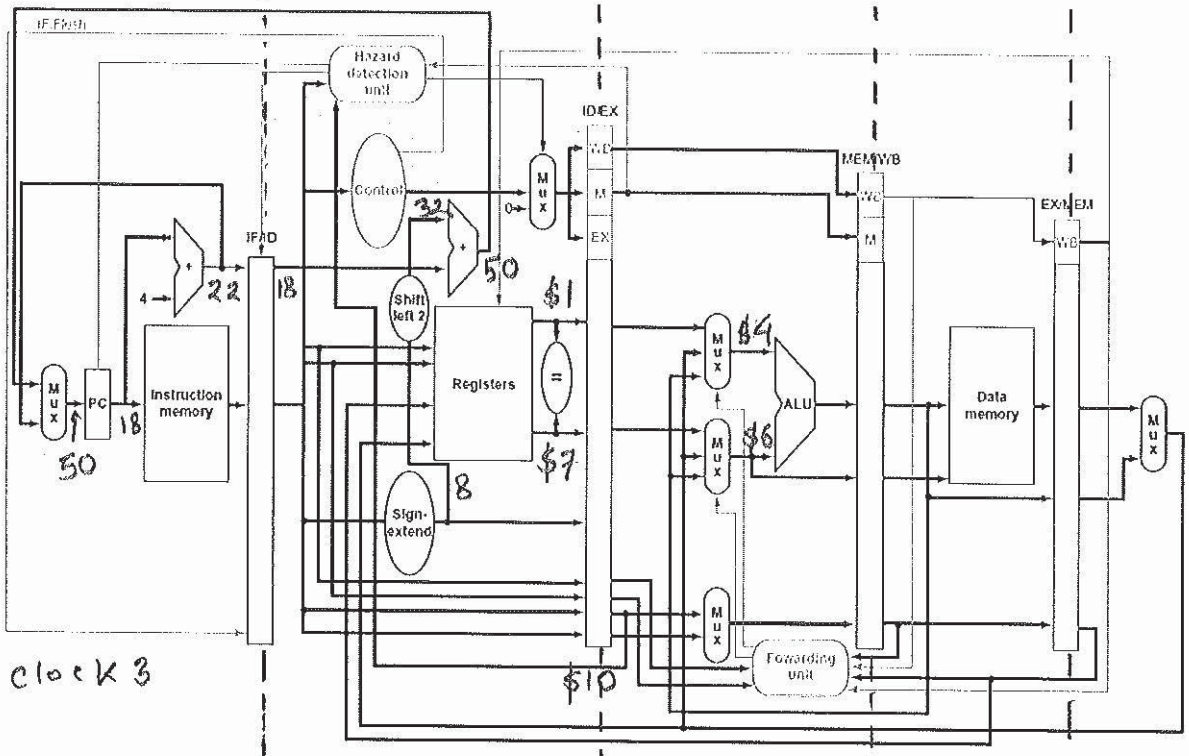
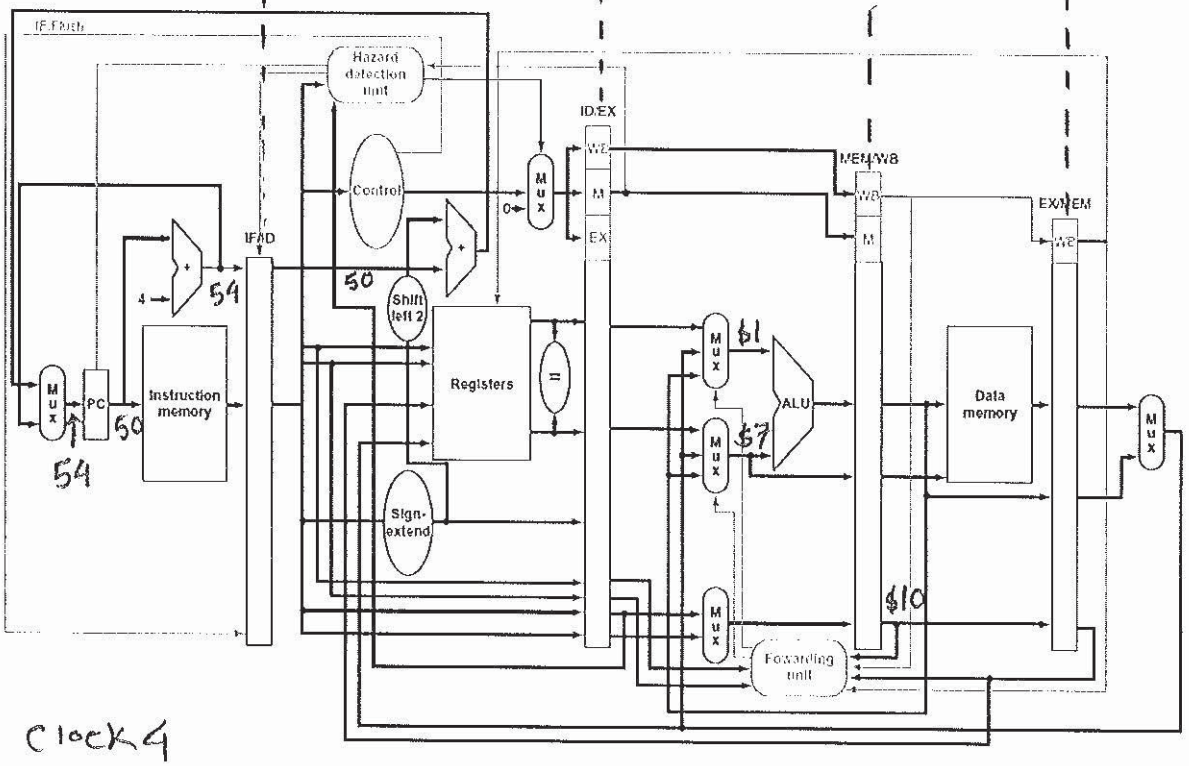
Q1

or $13, $2, $6 | beq $1, $7, 8 | add $10, $4, $6 | before <17 | before <17



clock 3

sw $4, 50($7) | bubble (nop) | beq $1, $7, 8 | add $10, $4, $6



clock 4

4.8.1 Pipelined

ID stage takes the longest (i.e. 350ps)

∴ clock cycle time = 350 ps

Non-pipelined

(IF + ID + Ex + MEM + WB)

clock cycle time = (250 + 350 + 150 + 300 + 200) ps

⇒ clock cycle time: 1250 ps

4.8.2 LW uses all stages

Non-pipelined

total latency = (250 + 350 + 150 + 300 + 200) ps

⇒ total latency = 1250 ps

Pipelined

Through put is improved, and latency only depends on the longest stage length.

⇒ total latency = (5 stages) · (ID stage)

= (5)(350 ps)

⇒ total latency = 1750 ps

4.8.3

Since ID stage is the longest, if you split its stage into two new stages, the time for each of those stages will be halved (i.e. 350/2 = 175ps). After the split, the new clock cycle time will depend on the longest stage, which is now MEM stage.

=> New clock cycle time after ID split = 300ps

4.8.4 Only "lw" and "sw" instructions deal with data memory.

Utilization = lw + sw = (20 + 15) %

=> Utilization = 35%

4.8.5 Only "lw" and "alu" instructions are utilized in this case.

Utilization = lw + alu = (20 + 45) %

=> Utilization = 65%

4.13

a.) sw R16, -100(R6)
lw R4, 8(R16)
add R5, R4, R4 } data hazard

4.13.1

In line 1, R6 is destination (offset by -100) and R16 is written in WB stage

In line 2, R4 is destination register and R16 offset by +8 is written in WB stage

In line 3 there is a hazard; source register (and rt) R4 is unavailable before WB stage from line 2 ∴ only 1 dependency

4.13.2 Data hazard between line 2 and 3

sw R16, -100(R6)
lw R4, 8(R16) } data hazard
add R5, R4, R4

⇒ :

```
sw R16, -100(R6)
lw R4, 8(R16)
NOP
NOP
add R5, R4, R4
```

9 cycles total

1	IF	ID	Ex	MEM	WB				
2		IF	ID	Ex	MEM	WB			
NOP			nop	nop	nop	nop	nop		
NOP				nop	nop	nop	nop	nop	WB

4.13.3 — 7 cycles

1	IF	ID	EX	MEM	WB			
2		IF	ID	EX	MEM	WB		
3			IF	ID	EX	MEM	WB	

With forwarding, one stage must be forwarded to get R4 value before WB stage in line 2.

4.13.4 Without forwarding

total ex. time = 9 clock cycles × 250ps = 2250 ps

With full forwarding

total ex. time = 7 clock cycles × 300 ps = 2100 ps

$$\text{speedup} = \frac{2250\ ps}{2100\ ps} = 1.07$$

so it's a slight speedup

4.13.5 — 9 cycles

1	IF	ID	Ex	MEM	WB			
2		IF	ID	Ex	MEM	WB		
3			IF	ID	NOP	NOP	Ex	MEM	WB

NOP: can't get R4 ↓ ; NOP: can't get ↓

4.13.6

Only ALU-ALU forwarding

Total ex. time = (9 cycles) · (290 ps)

Total ex. time = 2610 ps

No forwarding from before...

Total ex. time = ... = 2250 ps

$$\Rightarrow \text{speedup} = \frac{2250\ ps}{2610\ ps} = 0.86$$

, so it's slower now.