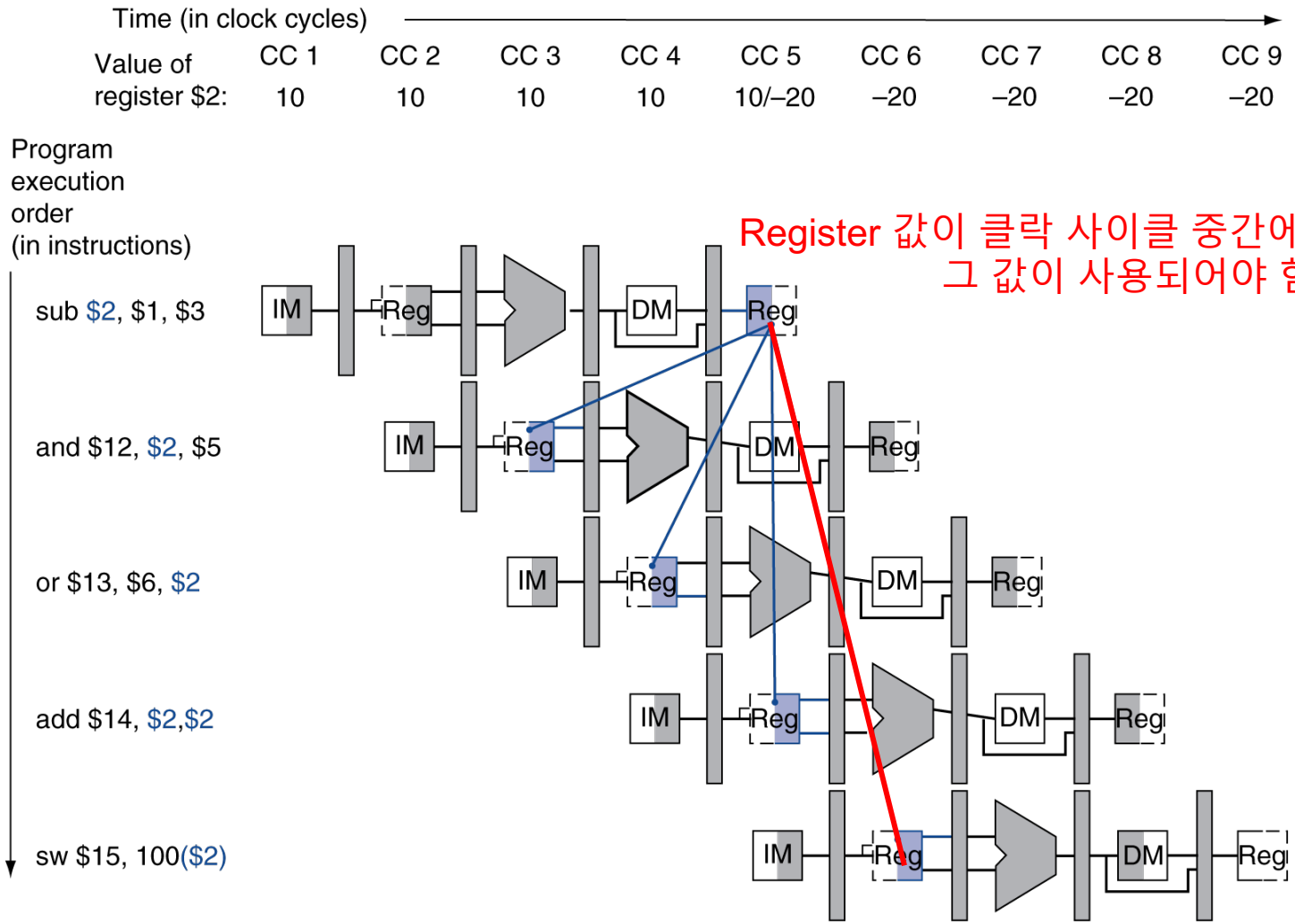
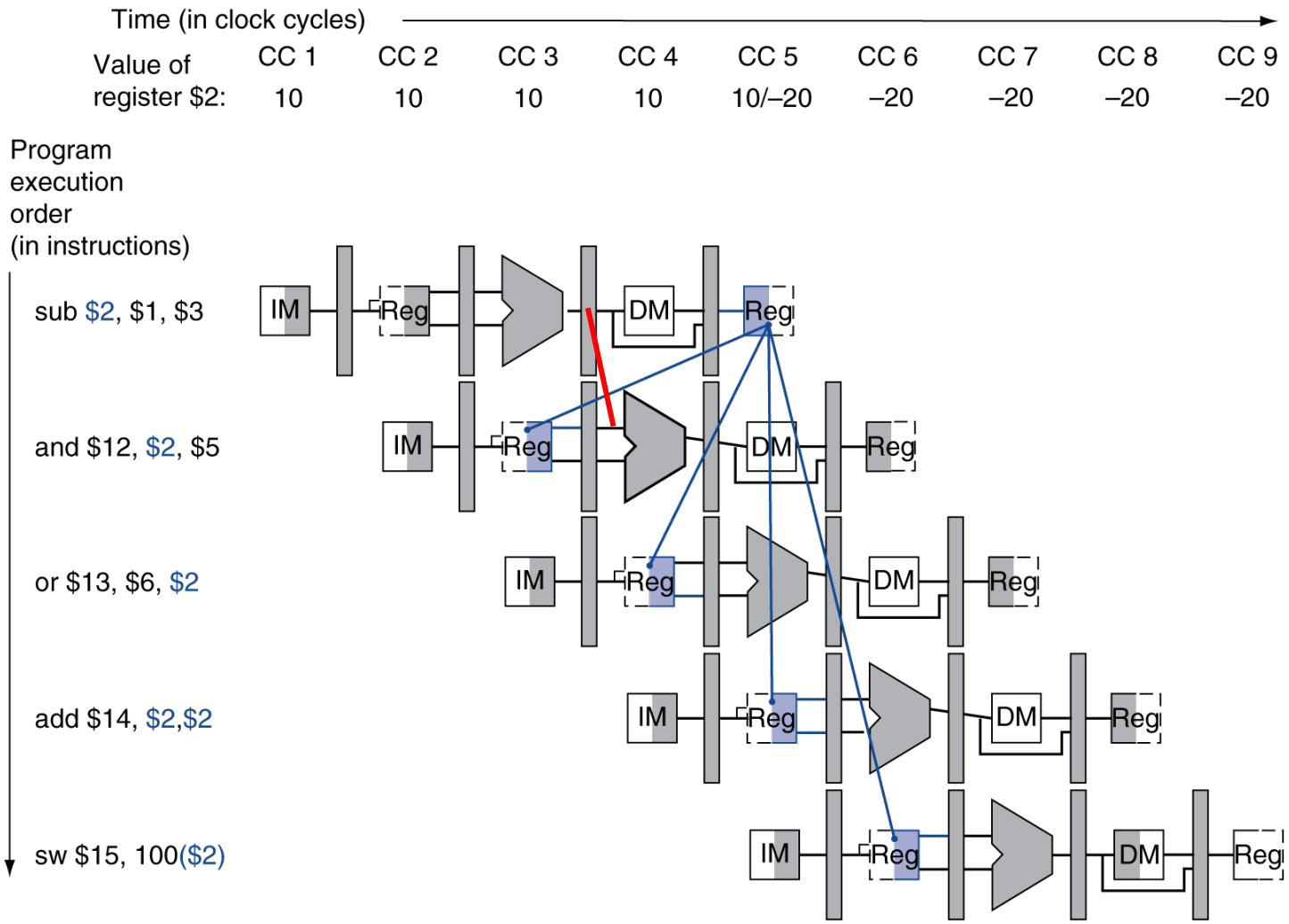
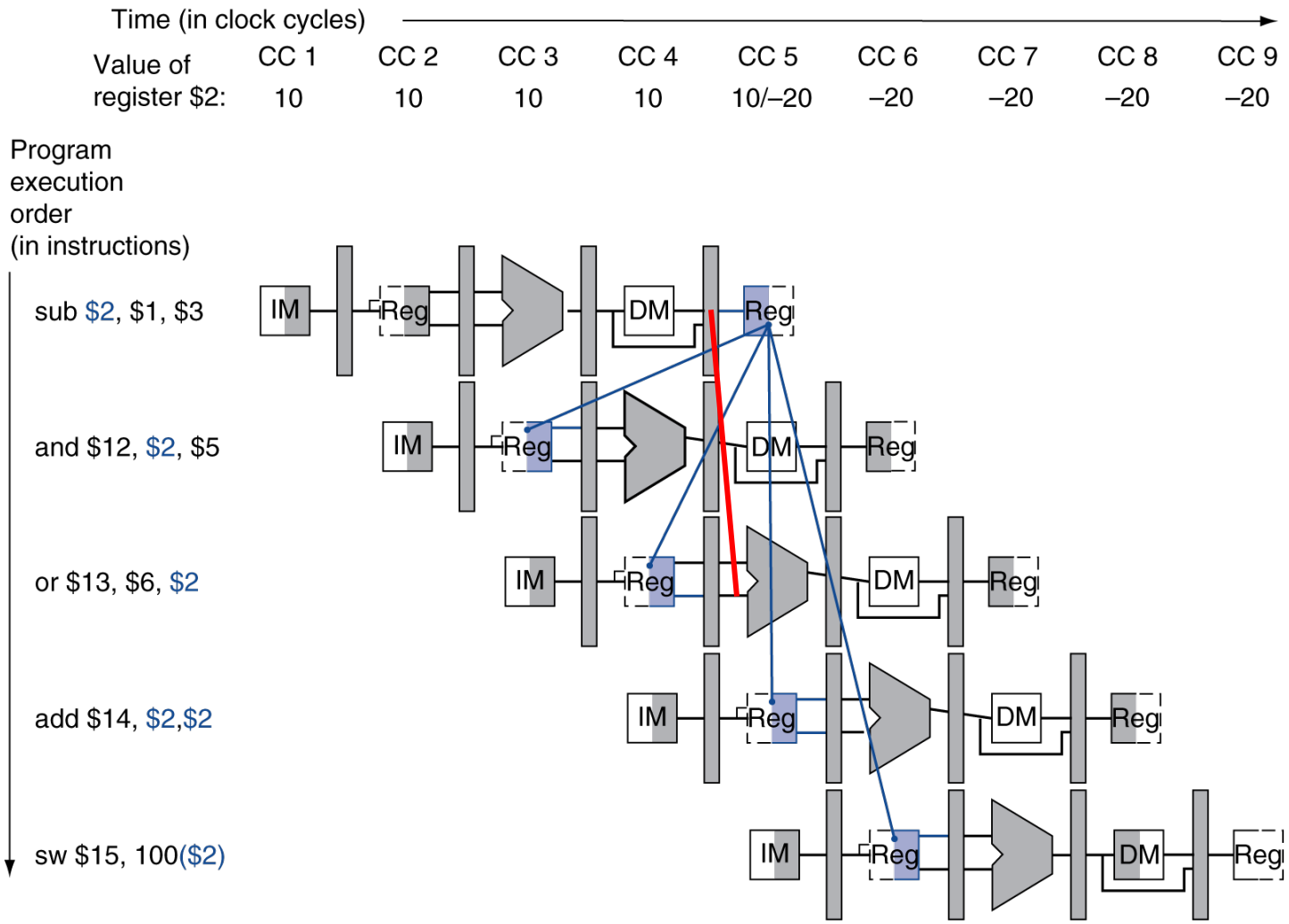
No Forwarding



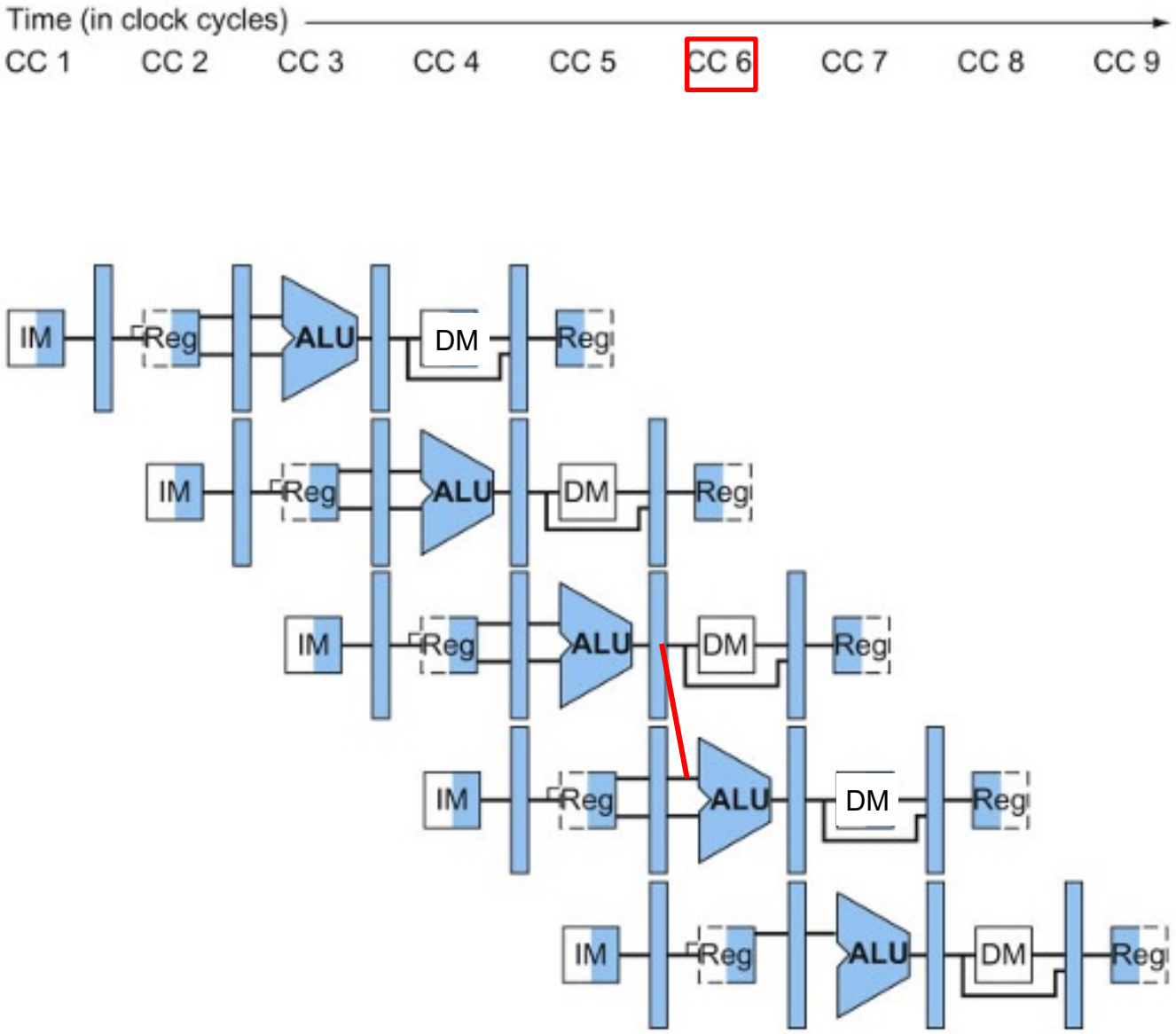
Solving EXE hazard



Solving MEM hazard



Solving Double hazard



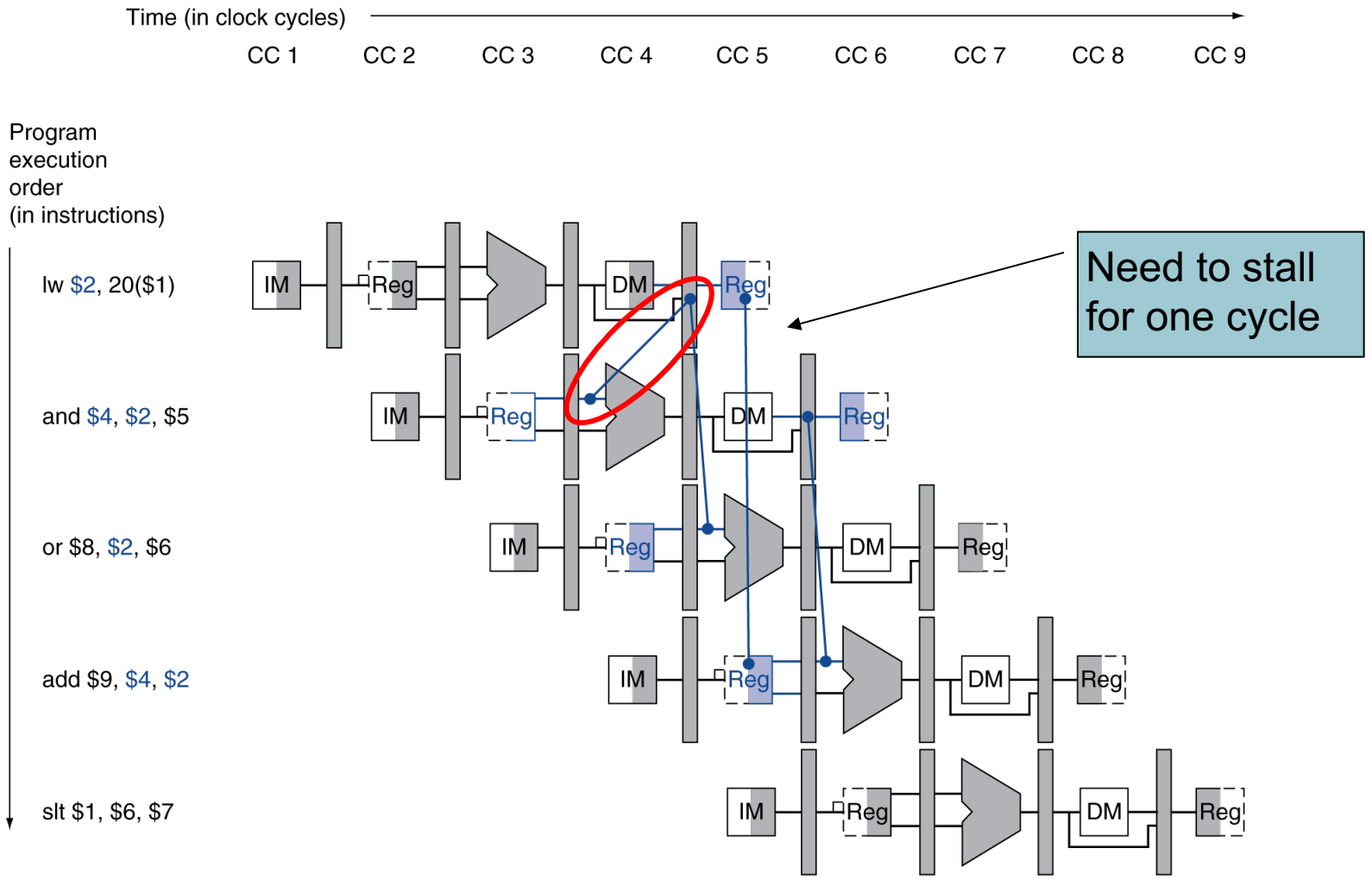
sub $2, x, x

and $4, $2, x

or $4, $4, $2

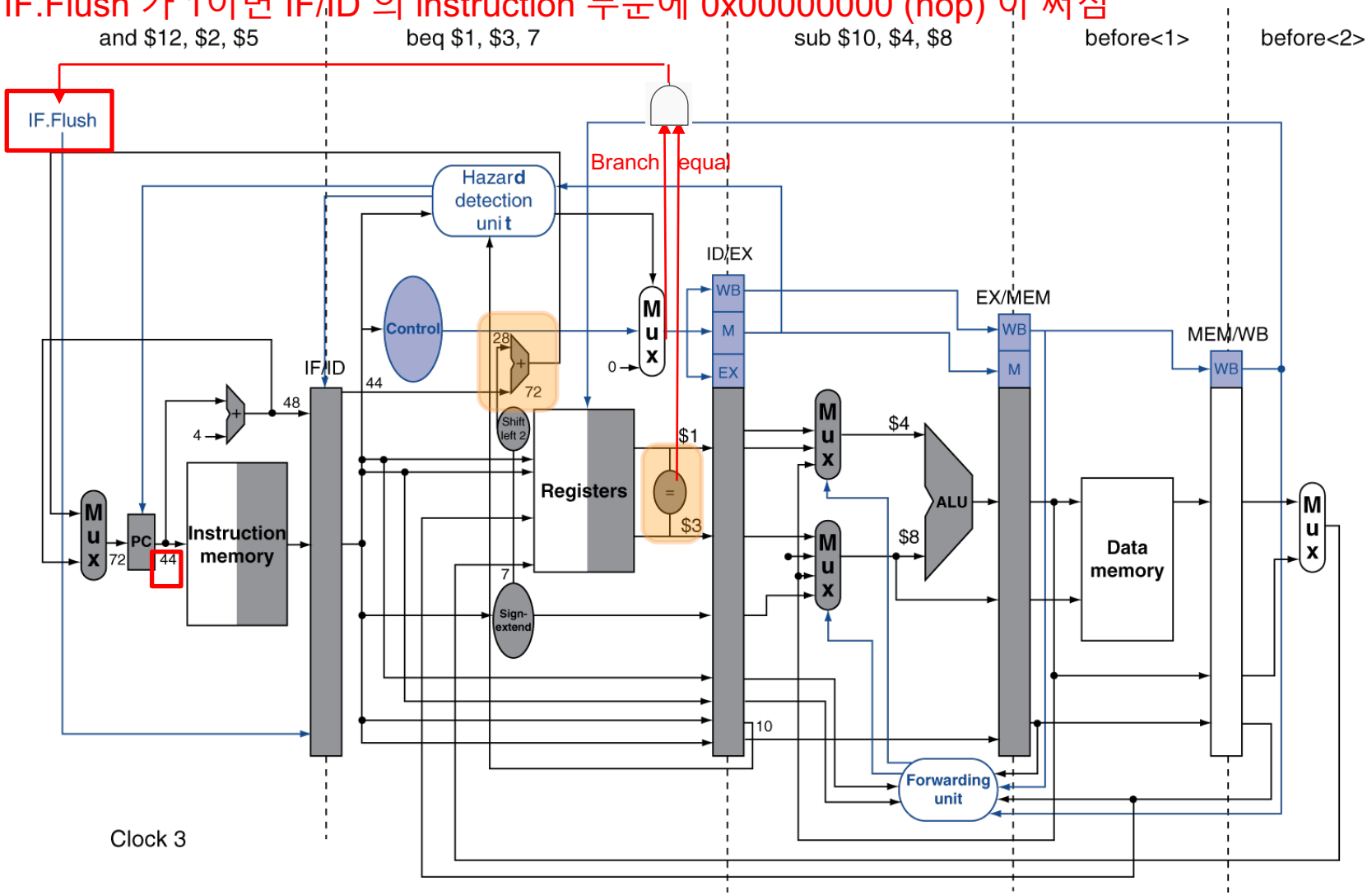
add $9, $4, $2

Load-Use Data Hazard



Stall for Branch Taken

IF.Flush 가 1이면 IF/ID 의 instruction 부분에 0x00000000 (nop) 이 써짐

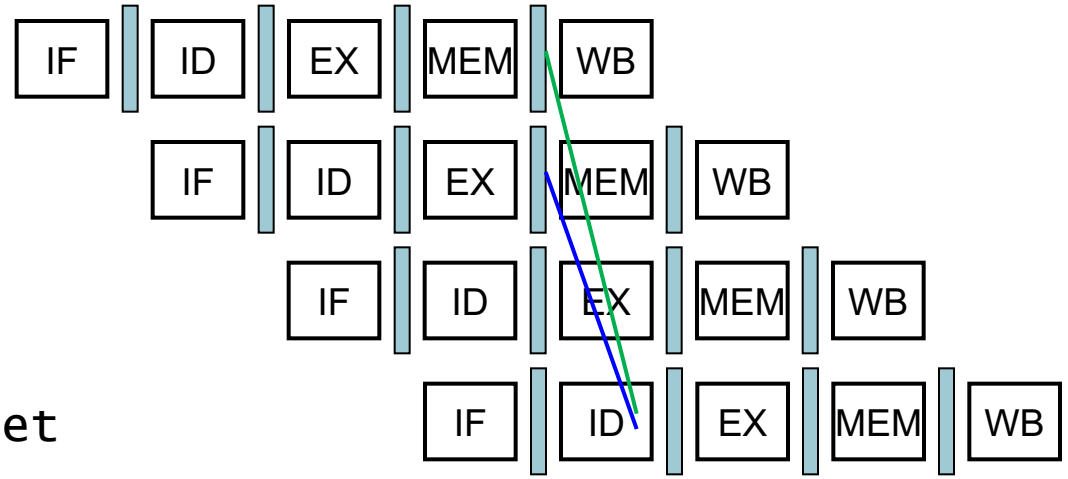


Stall for J

Data Hazards for Branches

- If a comparison register is a destination of 2nd or 3rd preceding ALU instruction

```
add $1, $2, $3
add $4, $5, $6
…
beq $1, $4, target
```

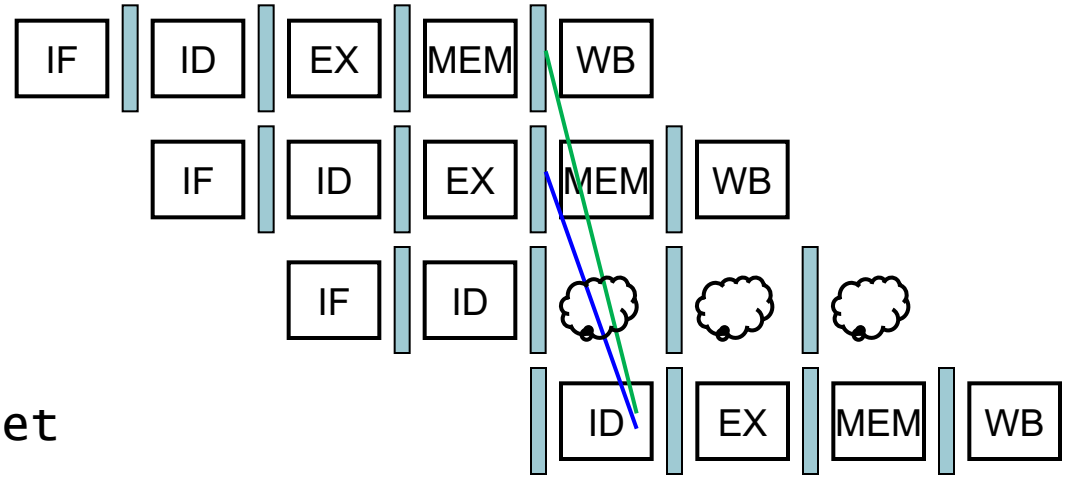


- Can resolve using forwarding

Data Hazards for Branches

- If a comparison register is a destination of preceding ALU instruction or 2nd preceding load instruction
  - Need 1 stall cycle

```
lw  $1, addr
add $4, $5, $6
beq stalled
beq $1, $4, target
```



Data Hazards for Branches

- If a comparison register is a destination of immediately preceding load instruction
  - Need 2 stall cycles

```
lw   $1, addr
beq  stalled
beq  stalled
beq  $1, $0, target
```