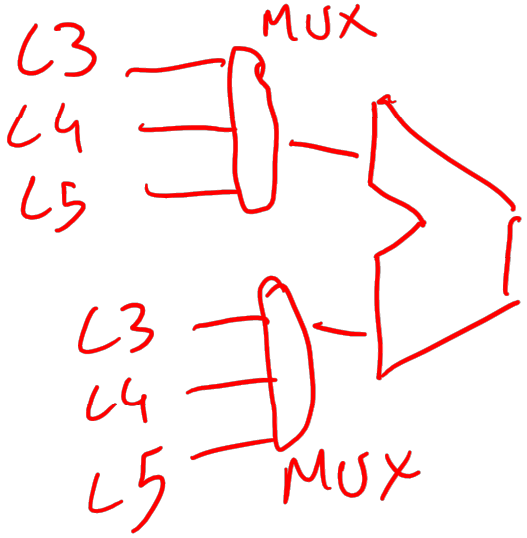
Lecture 18: Pipelining Hazards

- Today's topics:
  - Data hazards and instruction scheduling
  - Control hazards



HW 7 due Friday

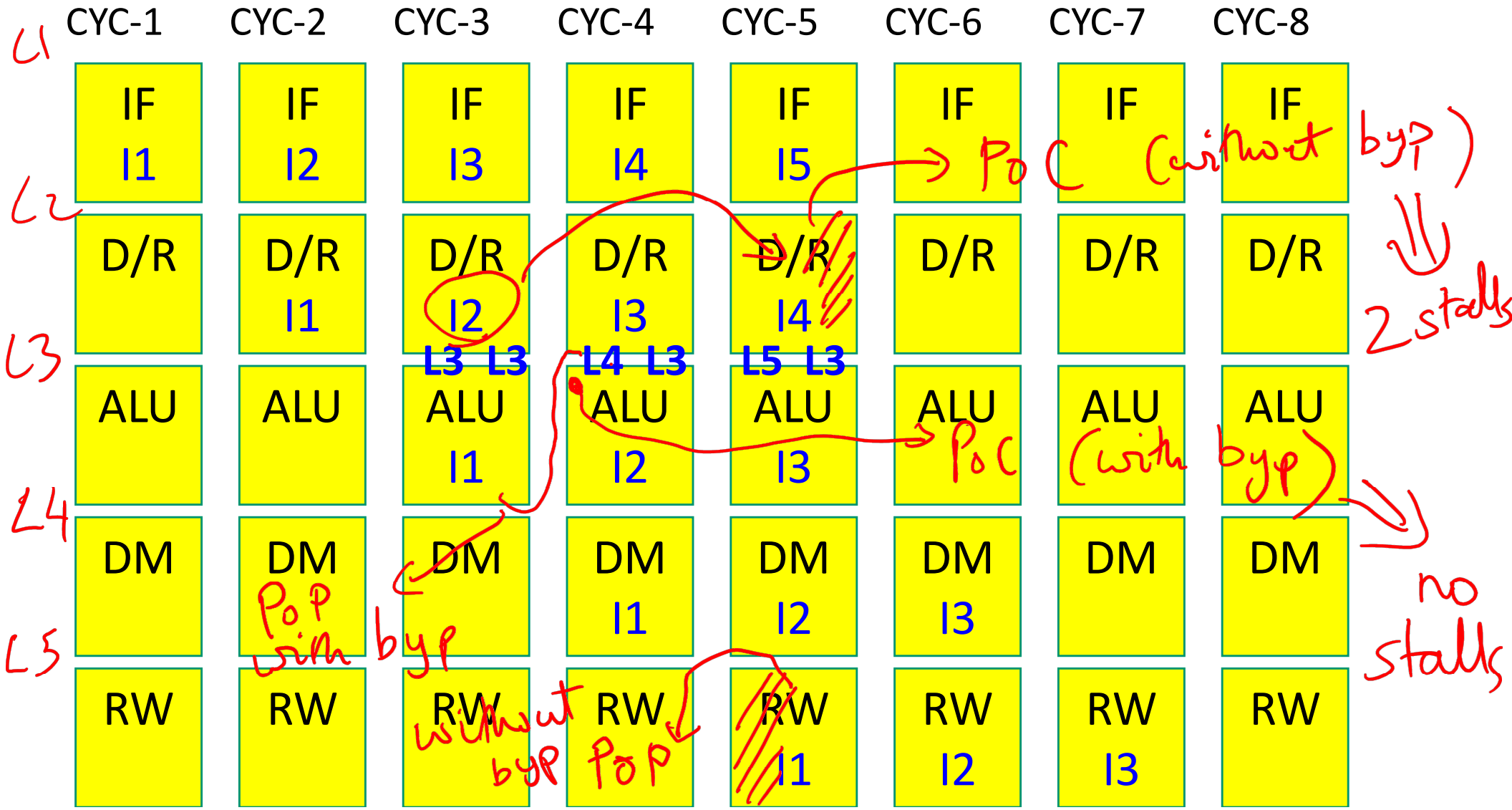
HW 8 due next Friday

MidTerm 2 wks away (Tuesday)

HW effort (byp) and/or Compiler effort (scheduling)

Example 2 – Bypassing

- Show the instruction occupying each stage in each cycle (with bypassing) if I1 is R1+R2→R3 and I2 is R3+R4→R5 and I3 is R3+R8→R9. Identify the input latch for each input operand.



Problem 0

D/R is where you stall
Decode and Reg Rd

```
add $1, $2, $3
add $5, $1, $4
```

I1 Prod
I2 Cons

● Point of Production

● Point of Consumption

Without bypassing:

add $1, $2, $3: IF DR AL DM RW

add $5, $1, $4: IF DR DR DR AL DM RW

↑ C1 ↑ C2

end of 1st half of 5th stage

3 DR instead of 1

With bypassing:

add $1, $2, $3: IF DR AL DM RW

add $5, $1, $4: IF DR AL DM RW

0 stalls

PoC is 2nd half of 2nd stage

2 stalls

D/R

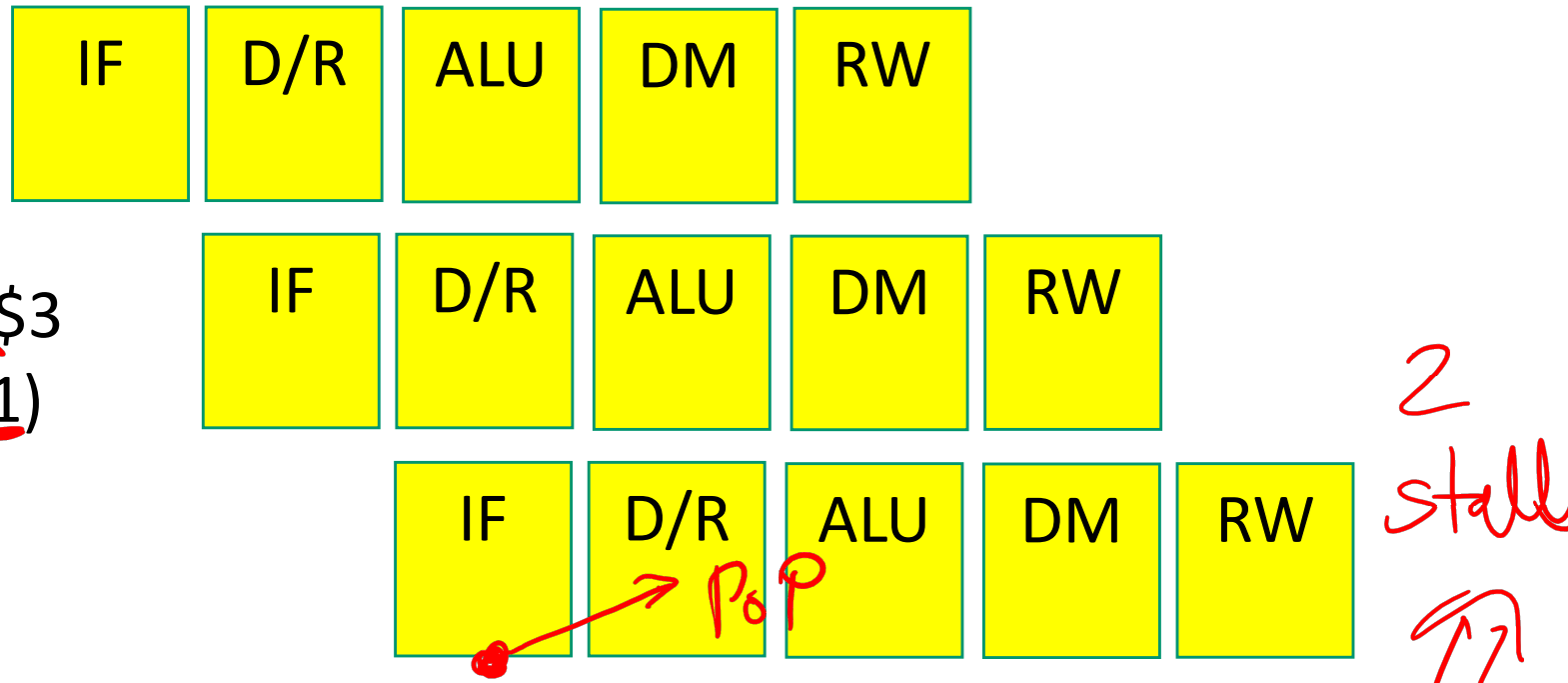
Problem 1 – No Byp

Add: PoP is stage 4:5

lw: PoC is stage 1:5

Prod add $1, $2, $3

Cons lw $4, 8($1)



2 stalls

Prod Add : IF DR AL DM RW

Cons lw : IF DR DR DR

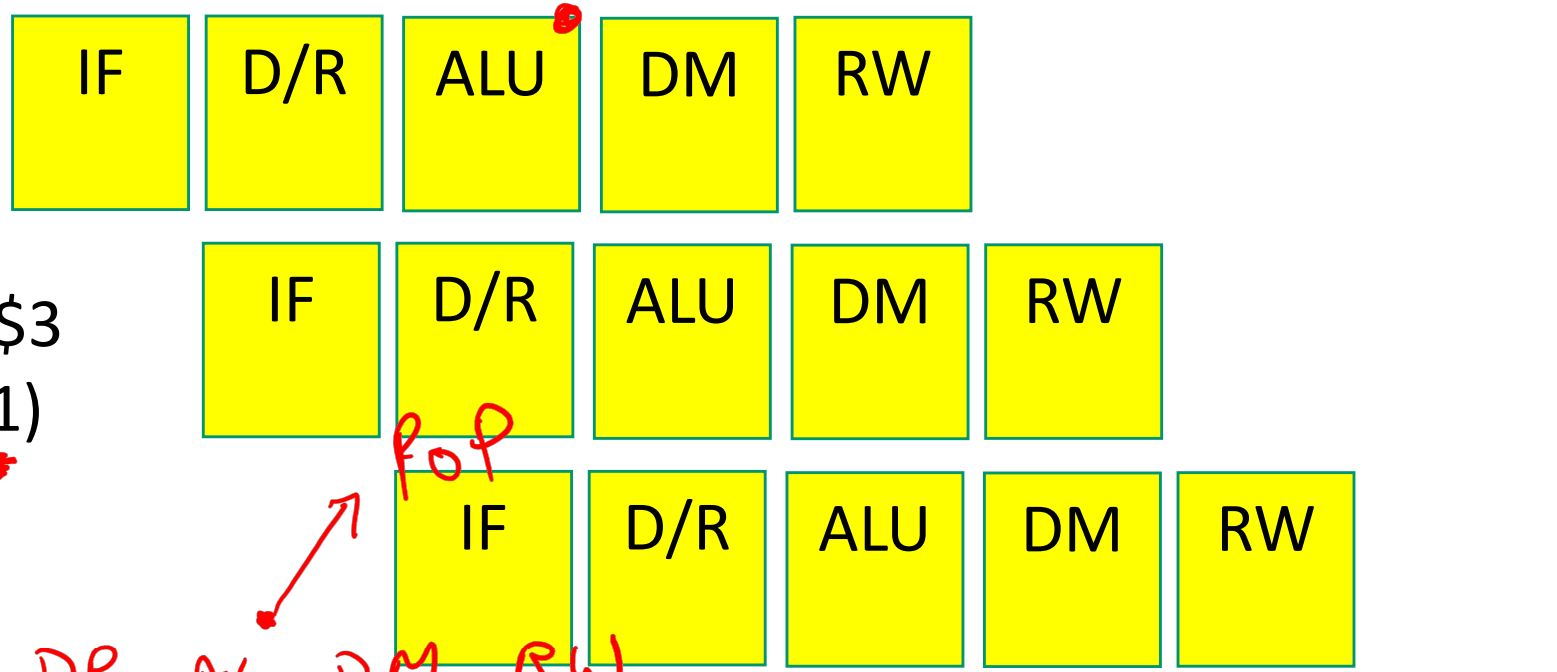
Erase → PoC PoC PoC

3 DR stages instead of 1

Problem 1 – with Byp

Add: PoP is end of ALU
lw: PoC is start of ALU

```
add  $1, $2, $3
lw   $4, 8($1)
```



ProdAdd: IF DR AL DM RW

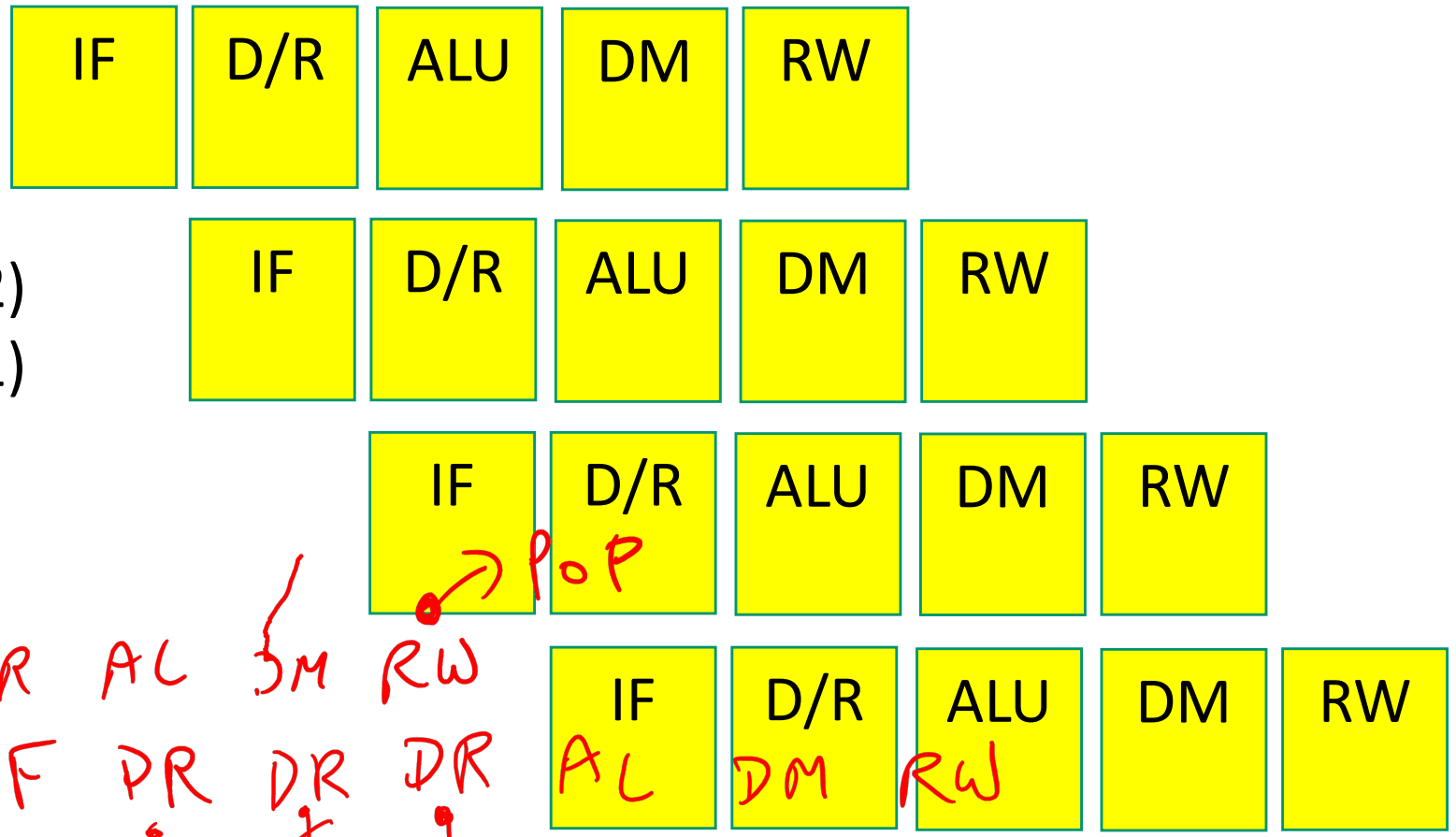
Cons lw: IF DR AL DM RW

1 DR stage ⇒ No stalls

Problem 2 – no Byp

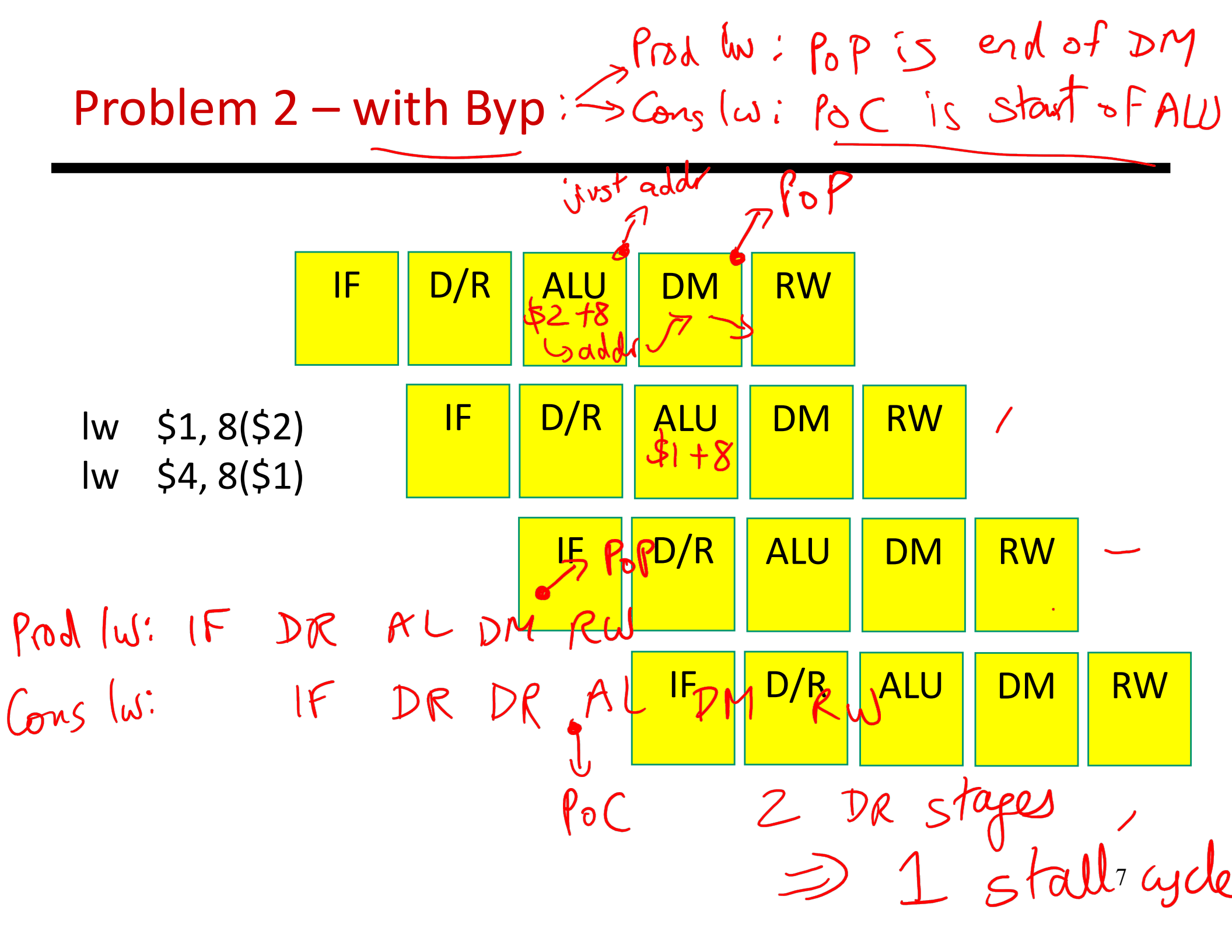
Prod: lw: PoP in stage 4:5
Cons: lw: PoC in stage 1:5

```
Prod lw   $1, 8($2)
Cons lw   $4, 8($1)
```

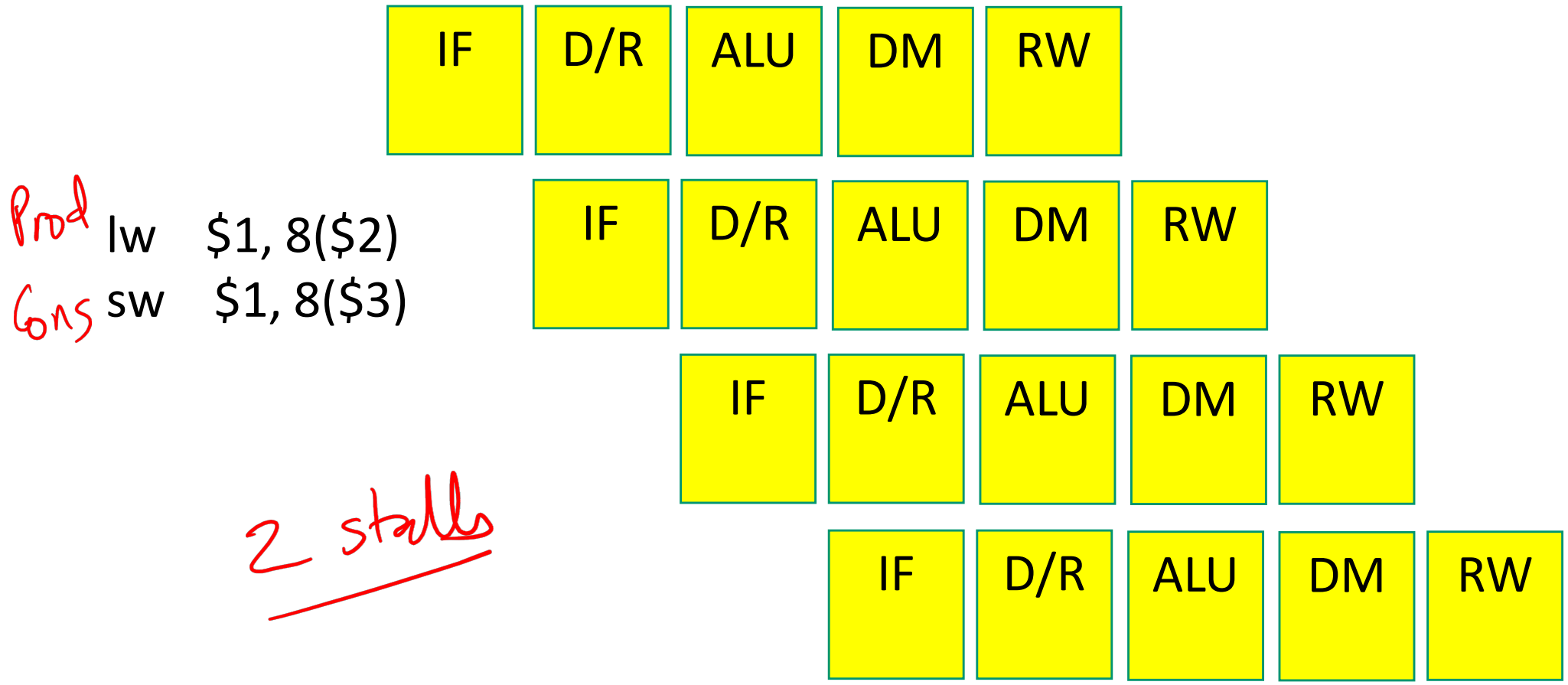


Problem 2 – with Byp

Prod lw: PoP is end of DM
Cons lw: PoC is start of ALU

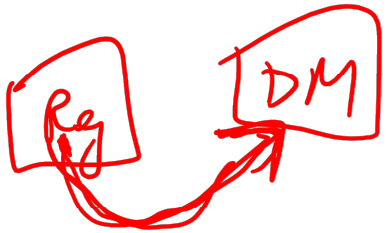


Problem 3 – no Byp



Problem 3 – with Byp

PoP for lw is end of DM
PoC for $1 is start of DM
PoC for $3 is start of ALU



PoP for lw

lw $1, 8($2)
sw $1, 8($3)

IF	D/R	ALU	DM	RW

	IF	D/R	ALU	DM	RW

$1
$3+8
add

Also solve when
sw $3, 8($1)
1 stall

PoP

	IF	D/R	ALU	DM	RW

Prod lw: IF DR AL DM RW

Cons sw: IF DR AL DM RW

		IF	D/R	ALU	DM	RW

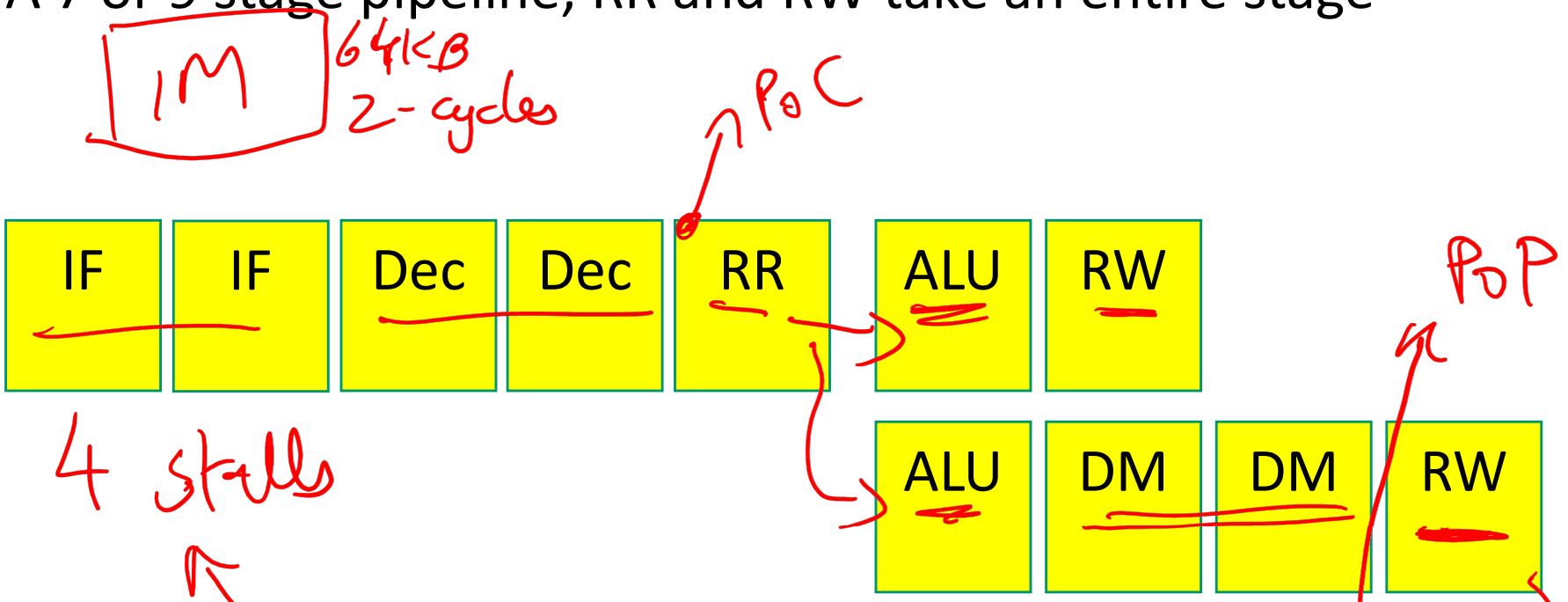
PoC (for $1)

0 stalls

Problem 4 – no Byp

Prod lw : PoP is end of st 9
Cons add: PoC is start of st 5

A 7 or 9 stage pipeline, RR and RW take an entire stage



lw $1, 8($2)

add $4, $1, $3

IF IF DE DE RR AL DM DM RW

IF IF DE DE DE DE DE DE RR $1 avail for others

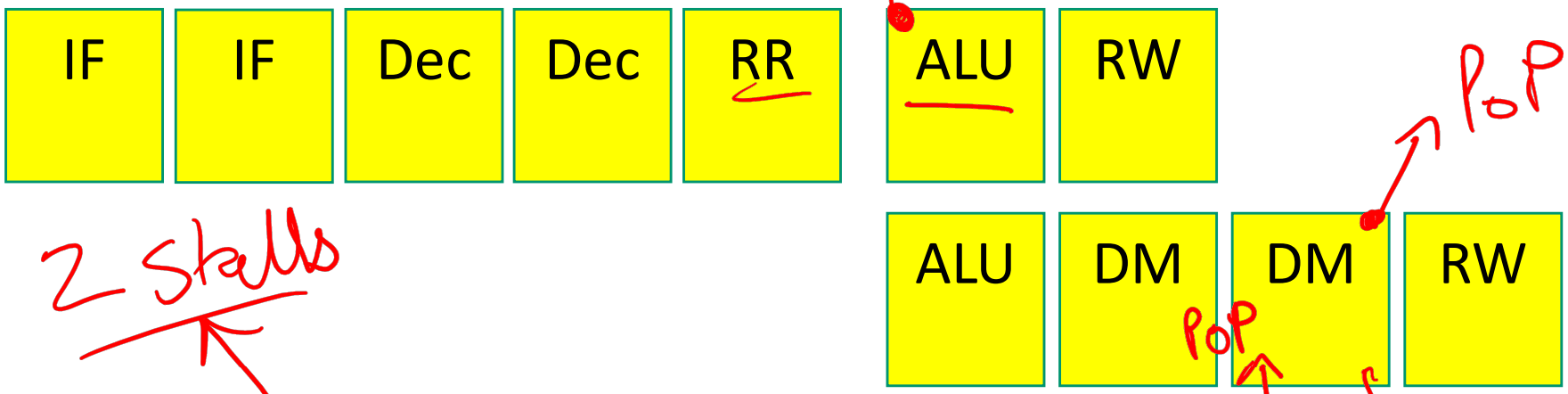
6 DE (instead of 2)

PoC

Problem 4 – with Byp

Prod lw; PoP is end of st 8
Cons add: PoC is start of st 6

A 7 or 9 stage pipeline, RR and RW take an entire stage



2 Stalls

lw $1, 8($2)

add $4, $1, $3

IF IF DE DE RR AL DM DM RW value for $1 is prod

IF IF DE DE DE DE RR AL

4 DEs (not 2)

PoC

Problem 4

Without bypassing: 4 stalls

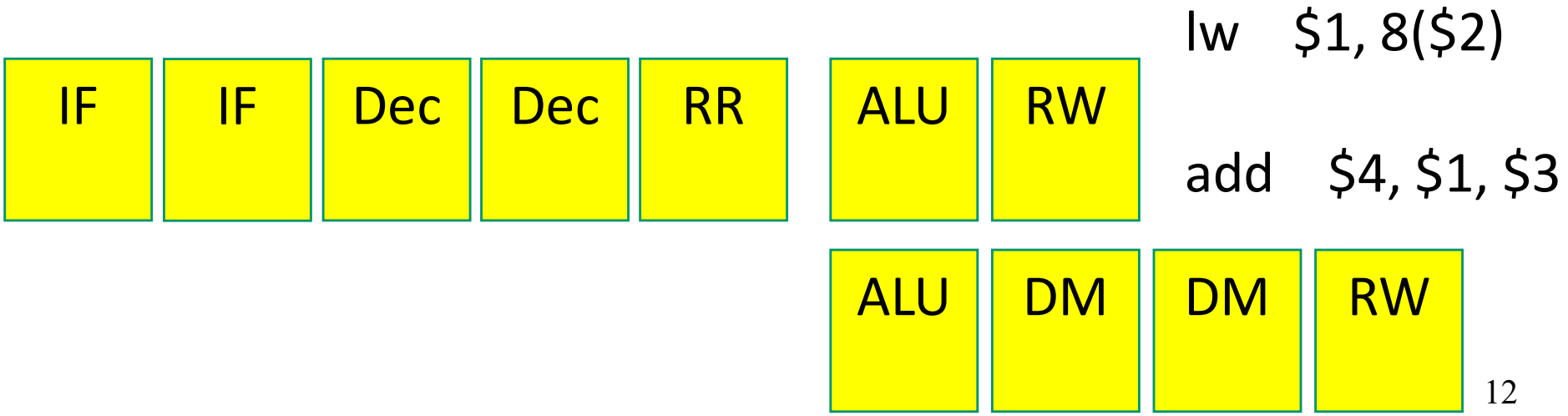
IF:IF:DE:DE:RR:AL:DM:DM:RW

IF: IF :DE:DE:DE:DE: DE :DE:RR:AL:RW

With bypassing: 2 stalls

IF:IF:DE:DE:RR:AL:DM:DM:RW

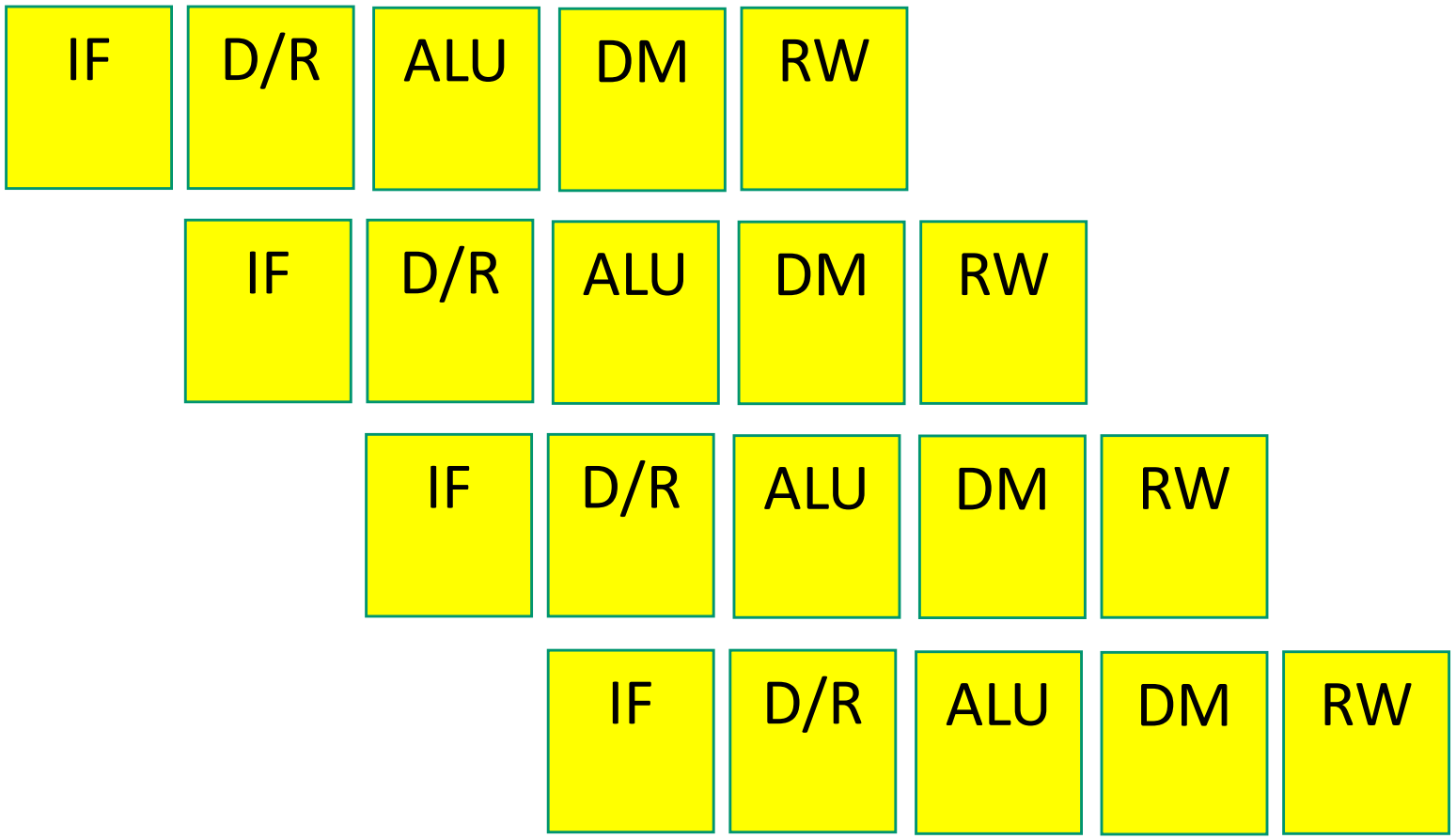
IF: IF :DE:DE:DE:DE: RR :AL:RW



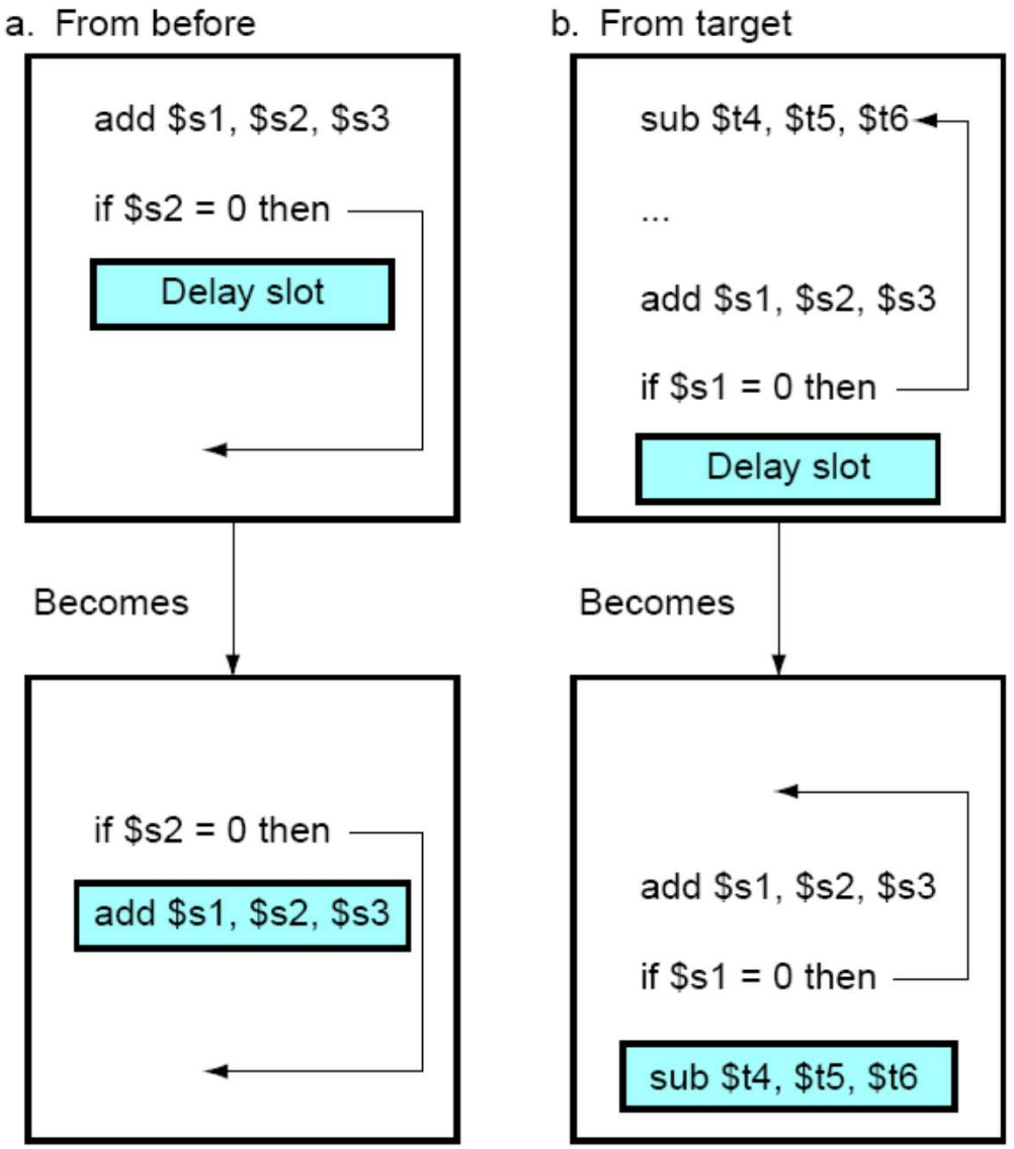
Control Hazards

- Simple techniques to handle control hazard stalls:
  - for every branch, introduce a stall cycle (note: every 6th instruction is a branch!)
  - assume the branch is not taken and start fetching the next instruction – if the branch is taken, need hardware to cancel the effect of the wrong-path instruction
  - fetch the next instruction (branch delay slot) and execute it anyway – if the instruction turns out to be on the correct path, useful work was done – if the instruction turns out to be on the wrong path, hopefully program state is not lost
  - make a smarter guess and fetch instructions from the expected target

Control Hazards



Branch Delay Slots



Source: H&P textbook