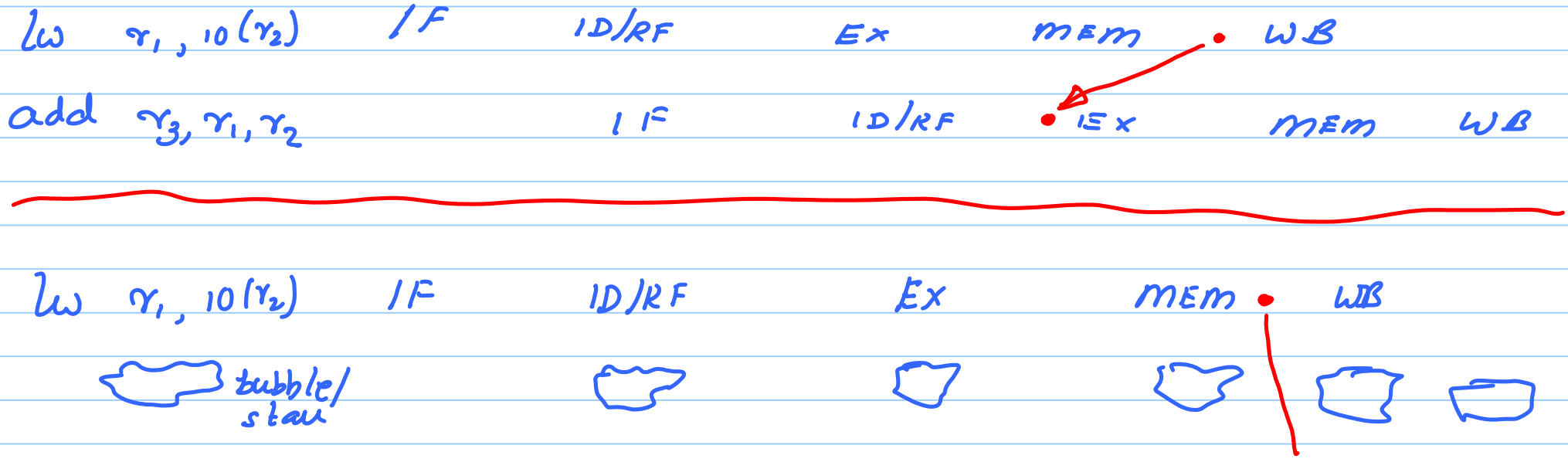
Oct-5

Note Title 05-10-2012

How do we fix a load-use hazard?

Space-Time Diagram.



add r_3, r_1, r_2 IF ID/RF EX MEM

bottom-right.

The way to fix a load-use hazard is to insert a stall/bubble between the load and the instruction that is using it (if these insts. are consecutive)

Time ↓

	Inst 1	Inst 2.
①	ALU	ALU
②	Ld	Ld
	St	St

Exercise: For these 9 pairs of instructions draw space-time diagrams, and figure out the need for forwarding.

Conclusion: Only for load-use a one cycle bubble is required

Quick look at the forwarding data path/ control path. (Slides)

SUMMARY:

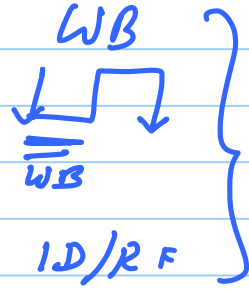
Forwarding involves adding an extra forwarding control unit to the pipeline.

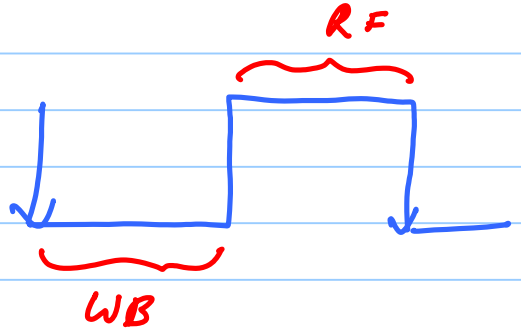
This unit generates control signals.
These control signals drive a mux that chooses between

a) default input

b) forwarded input

Optimization: No need to forward between WB & ID/RF



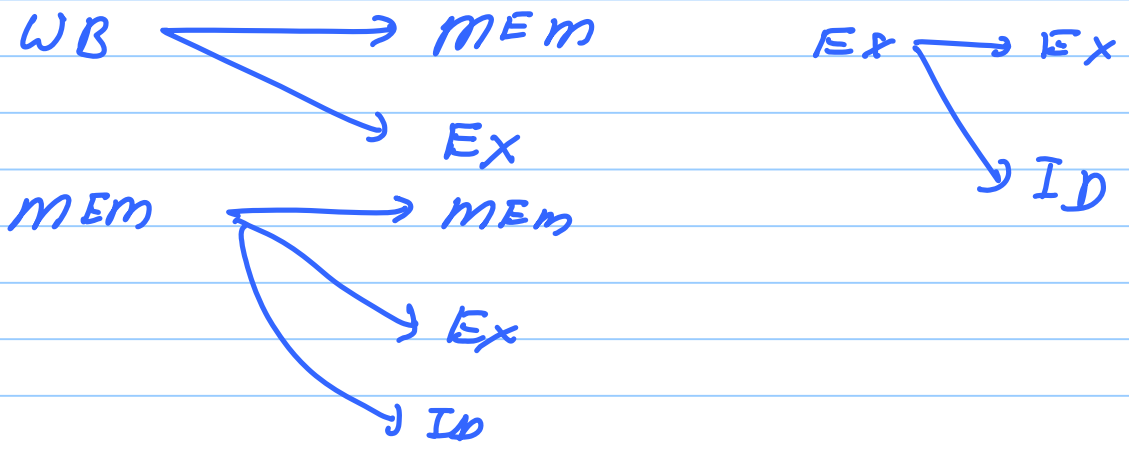


∴ The MIPS pipeline assumes:

WB does the register write in the early part of the clock cycle.

RF does the register read in the second half of the clock cycle

Forwarding Paths

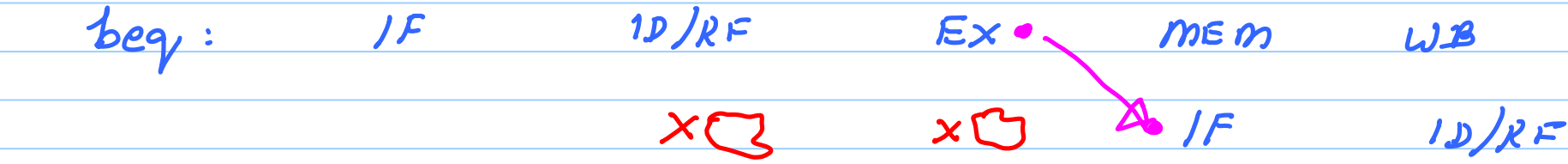


Control Hazards / Dependences.

Branch ???

1) Branch is evaluated in the EX stage

beq $r_1, r_2,$ <offset>



For control (branch) instructions)

1. Find the earliest possible point in the space-time diagram when the branch inst. is evaluated.

2. Find the earliest possible pt. at which you can fetch the target

3. Draw a line

bottom-left → Not possible

bottom-right → possible

When, the branch is evaluated in the Ex stage, we need to insert two pipeline bubbles.

Example.

Pipeline with 0 branch stall cycles: CPI 1.5

2 branch stall cycles

20% of my instr. are branches.

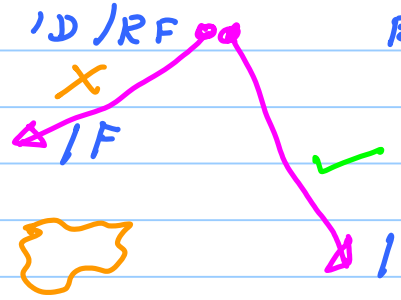
$$CPI_{new} = 1.5 + 0.2 \times 2$$
$$= 1.9$$

2) Branch is evaluated in the ID stage.

beq: IF ID/RF EX MEM WB

target: X IF

target: IF



If the branch is evaluated in the ID stage, we need to add 1 bubble after it

But!!

```
add   r1, r4, r5
beq   r1, r2, <20>
→target.
```

```
add   r1, r4, r5      IF    ID/RF    EX    MEM    WB
beq,  r1, r2, <20>          IF       ID/RF  EX     MEM   WB
                            IF       ID/RF  EX
                                     IF     ID/RF  EX
```

Summary

Branch inst. evaluated at:

	Before	After
ID	1 bubble maybe	1
Ex	3	2 (default)

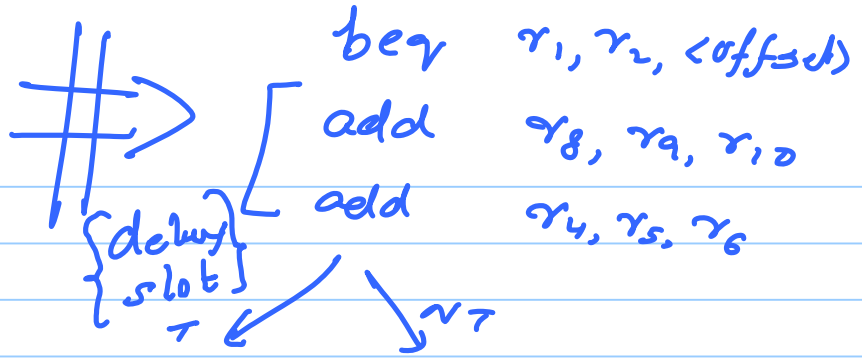
Q: How do you solve the issue of branch bubbles?

→ Delayed Branch (Today)
→ Branch Prediction (Tomorrow).

Delayed Branch:

beq r1 r2 ---

Ex:
```
add r8, r9, r10
add r4, r5, r6
beq r1, r2, <offset>
```



delayed branch:

compiler support: ① To place two instructions after the branch that will be executed all the time.

② These 2 instruction should not have any data dependence with the branch.

HW Support:

After a branch, do not stall. Execute the instructions in the delay slot.

If we are not able to find an inst. to fill in the delay slot insert a dummy inst. NOP