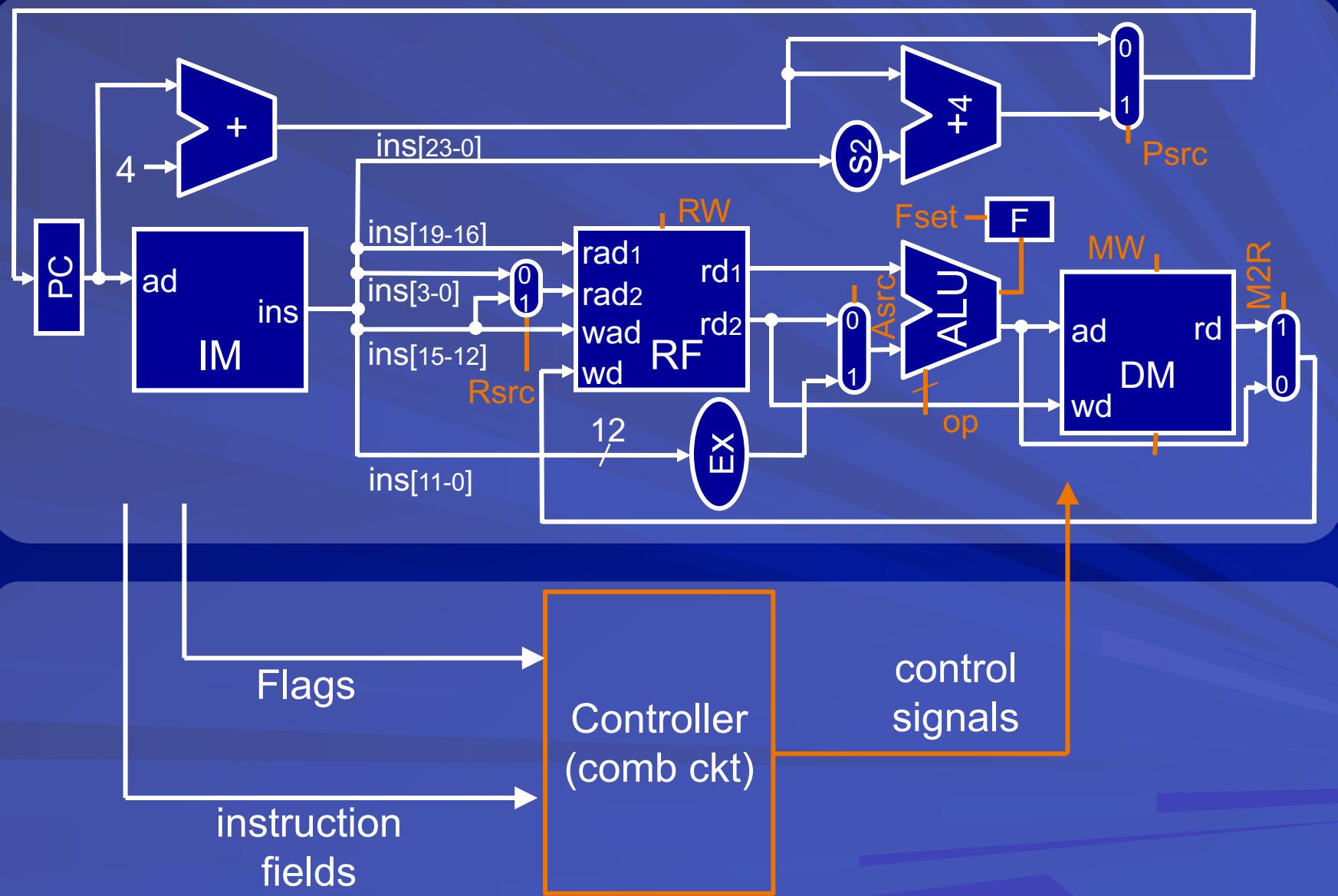


# COL216

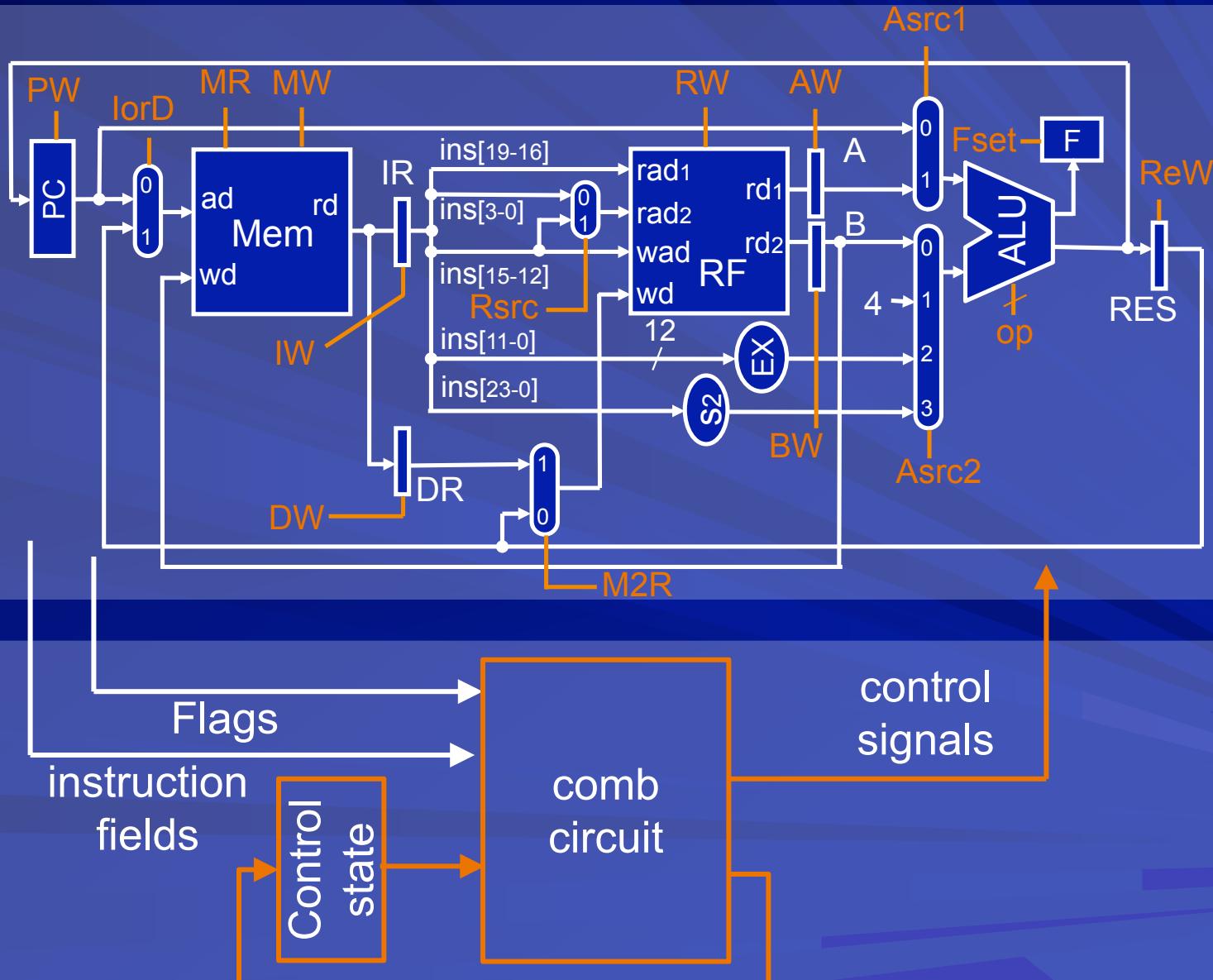
# Computer Architecture

Pipelined Processor design –  
Controller, Data forwarding  
10th February 2022

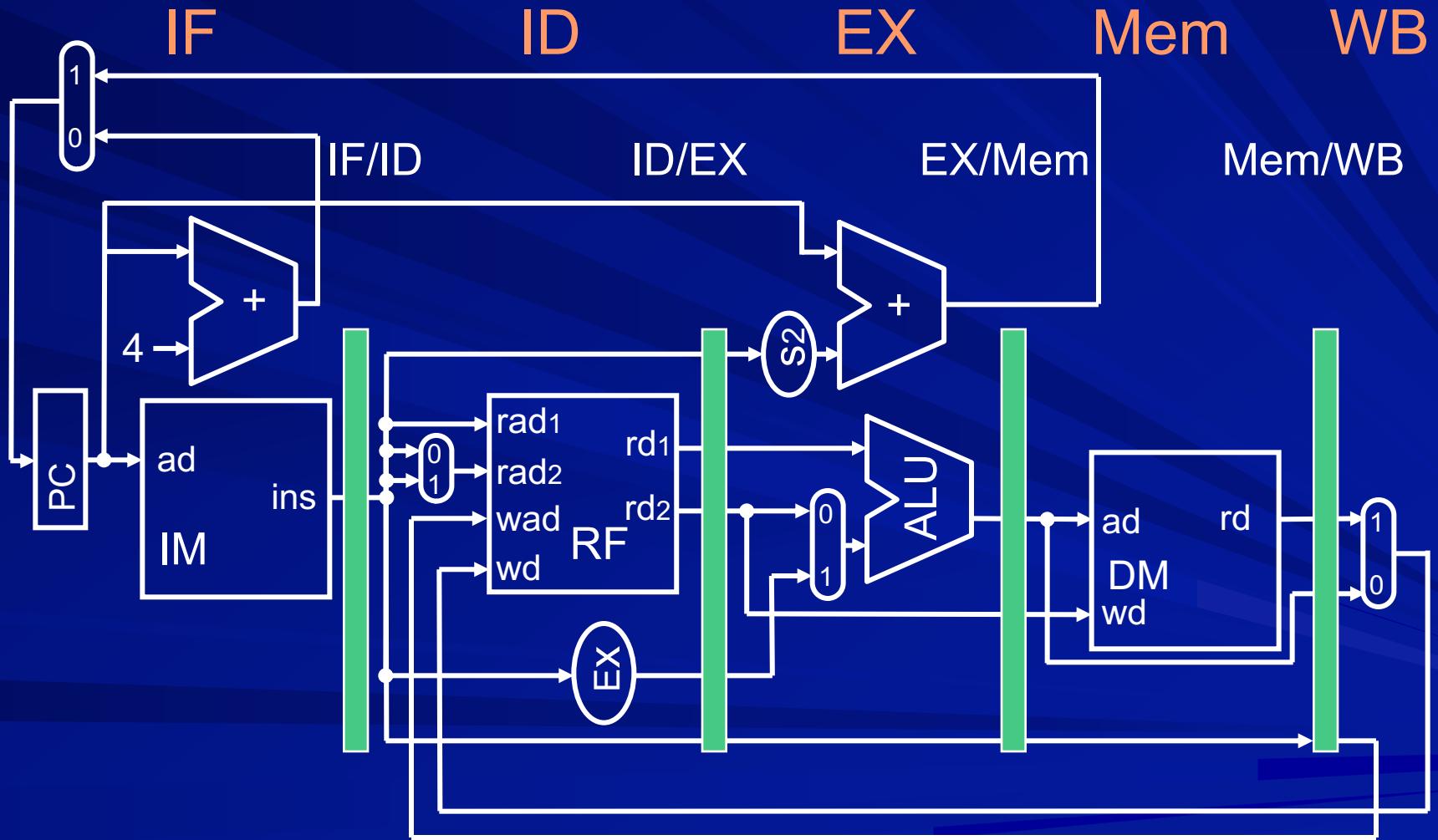
# Single Cycle Datapath + Controller



# Multi Cycle Datapath + Controller



# 5 Stage Pipeline



# Hurdles in instruction pipelining

## ■ Structural hazards

- Resource conflicts - two instruction require same resource in the same cycle

## ■ Data hazards

- Data dependencies - one instruction needs data which is yet to be produced by another instruction

## ■ Control Hazards

- Decision about next instruction needs more cycles

# Handling data hazards

- Assume no data hazards
  - leave it to compiler to remove hazards
- Introduce stalls/bubbles
  - requires hazard detection  
(check data dependence among instructions)
  - compiler may still help in reducing hazards
- Do data forwarding
  - this also requires hazard detection
  - stalls may also be required in some cases

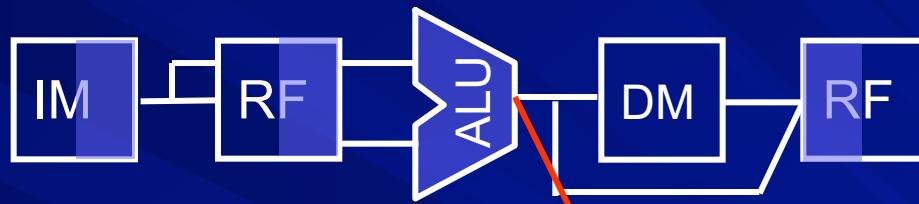
# Detecting data hazard

Condition to be checked:

Instruction in RF stage reads from a register in which instruction in ALU stage or DM stage is going to write

# Data forwarding path P1

I:add r6,...

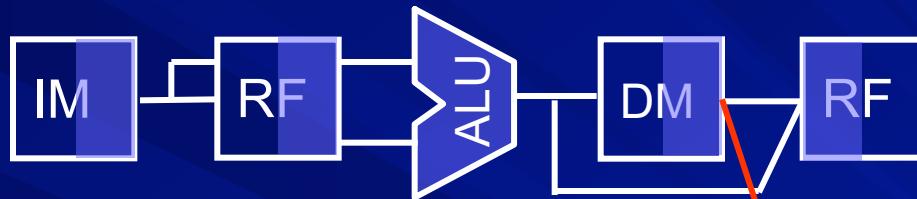


I+1:add r4, r6,..

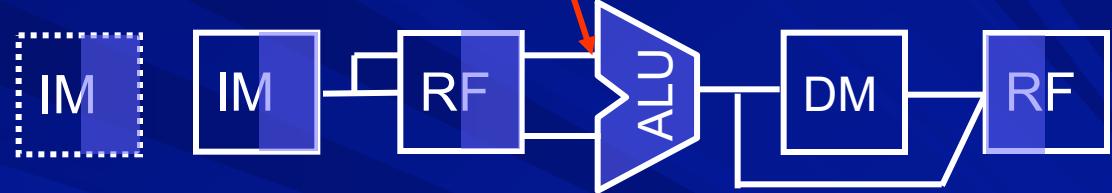


# Data forwarding path P2

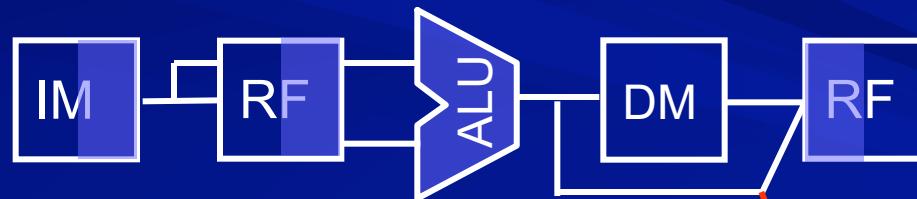
I:ldr r6,..



I+1:add r4,r6,..



I:add r6,..

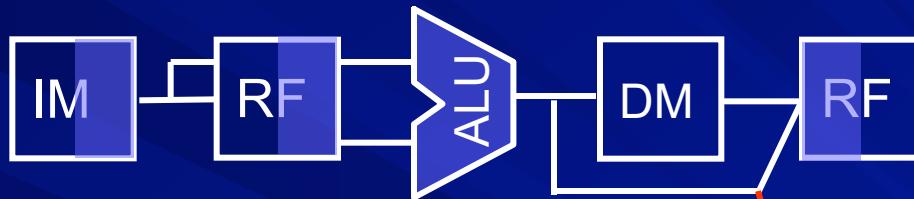


I+2:add r4,r6,..

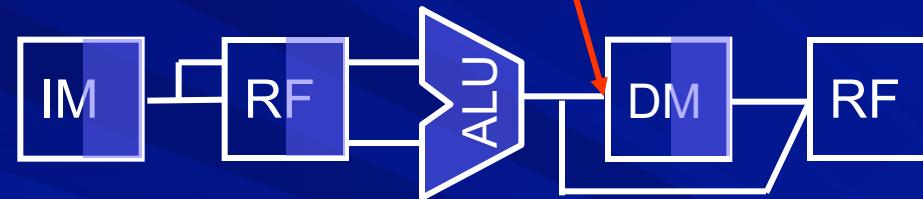


# Data forwarding path P3

I:add r6,...



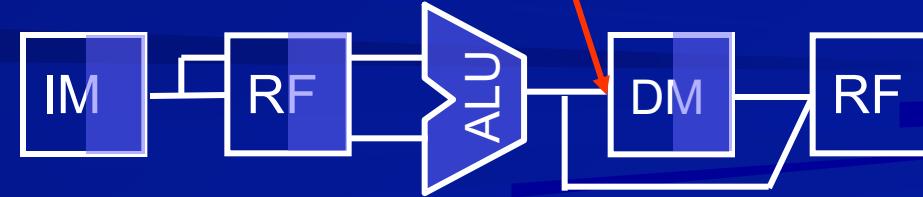
I+1:str r6,...



I:ldr r6,...



I+1:str r6,...



# Data forwarding path list

■ P1

from ALU out

to ALU in1/2

(EX/DM register)

■ P2

from DM/ALU out

to ALU in1/2

(DM/WB register)

■ P3

from DM/ALU out

to DM in

(DM/WB register)

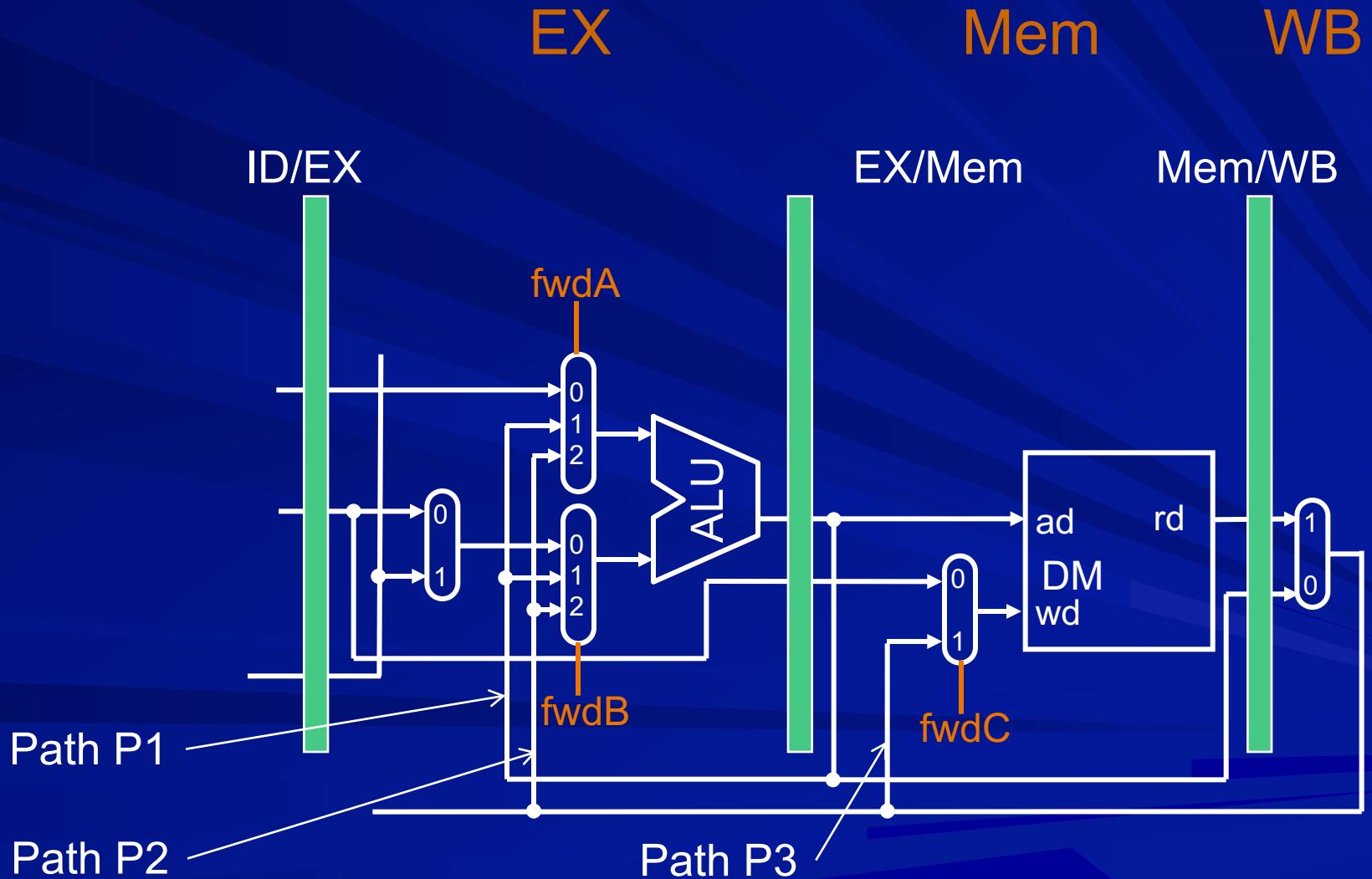
# Dependence check logic

Condition to be checked:

Operand of instruction in RF stage is a register in which instruction in ALU stage or DM stage is going to write

We need to ensure that instruction in RF stage actually reads Rn and/or Rm

# Data forwarding paths



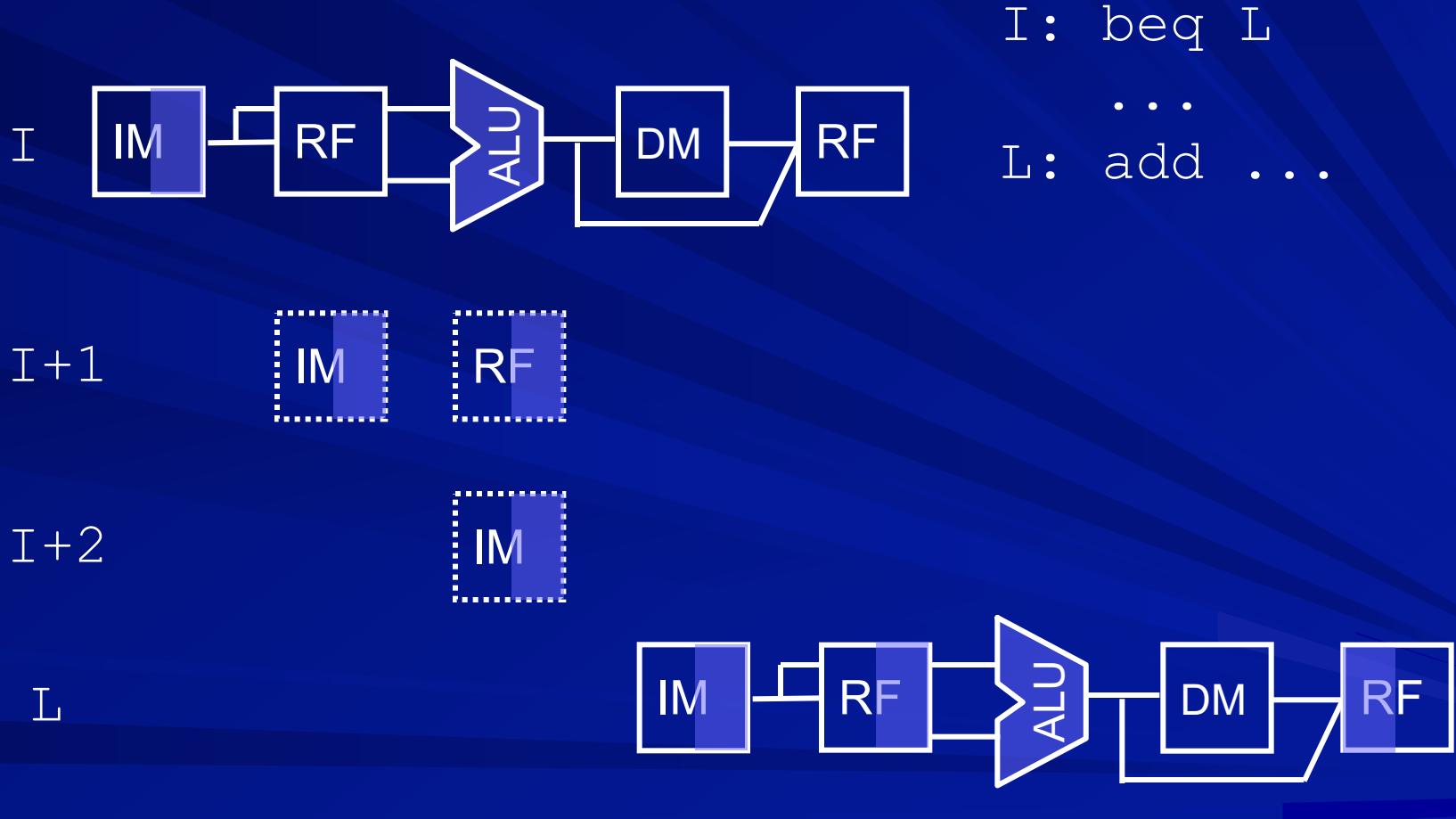
# Executing branch instructions

- In which cycle the instruction is found to be a branch instruction?
- In which cycle the branch decision is known?
- In which cycle the target address is computed?

# On decoding a branch instruction

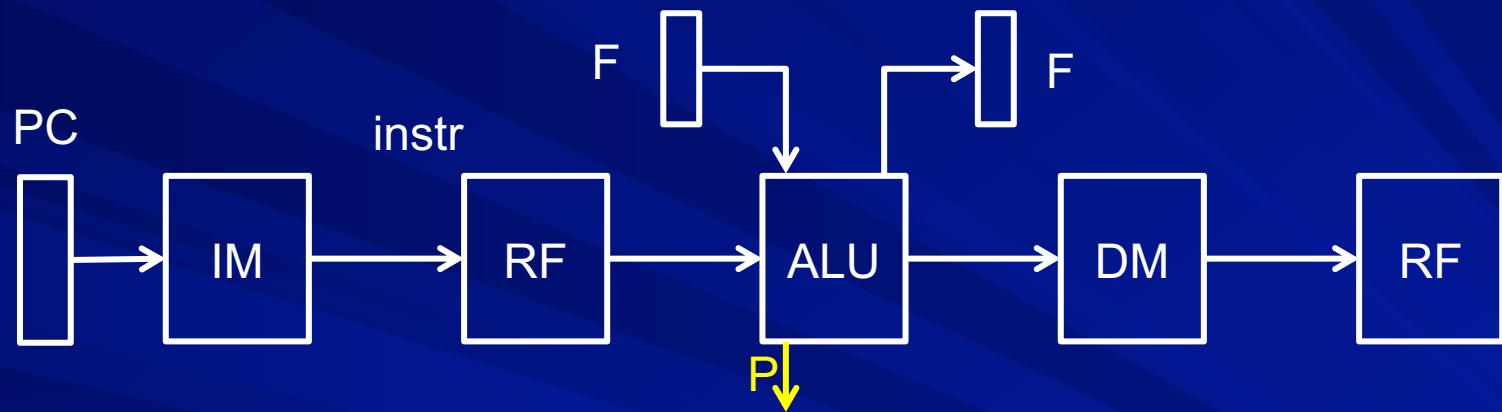
- Flush the inline instructions
- Freeze (stall) the inline instructions
- Allow the inline instructions to continue

# Stalls due to control hazards



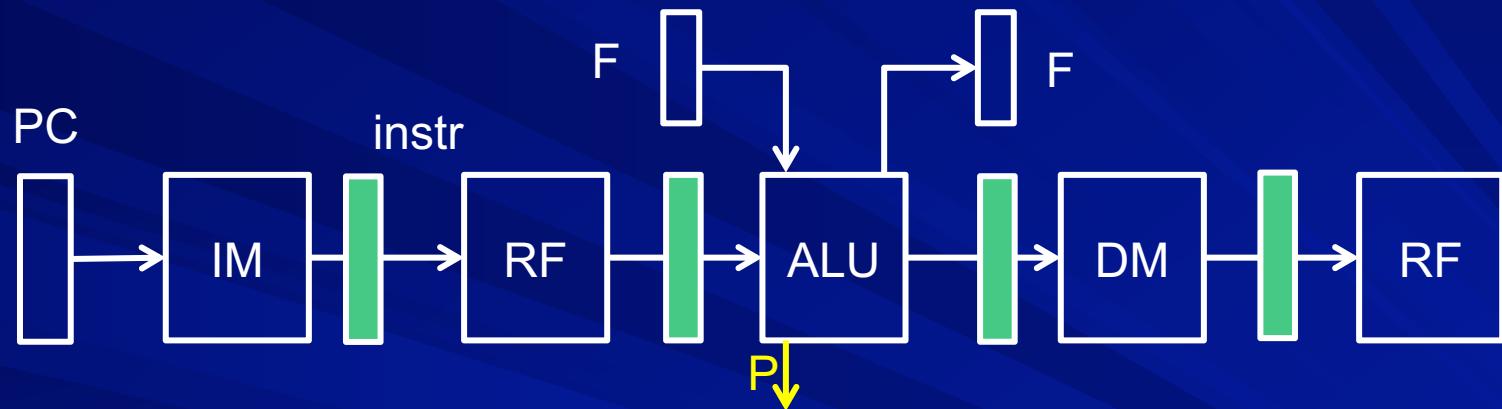
# Controller design

# Single cycle datapath



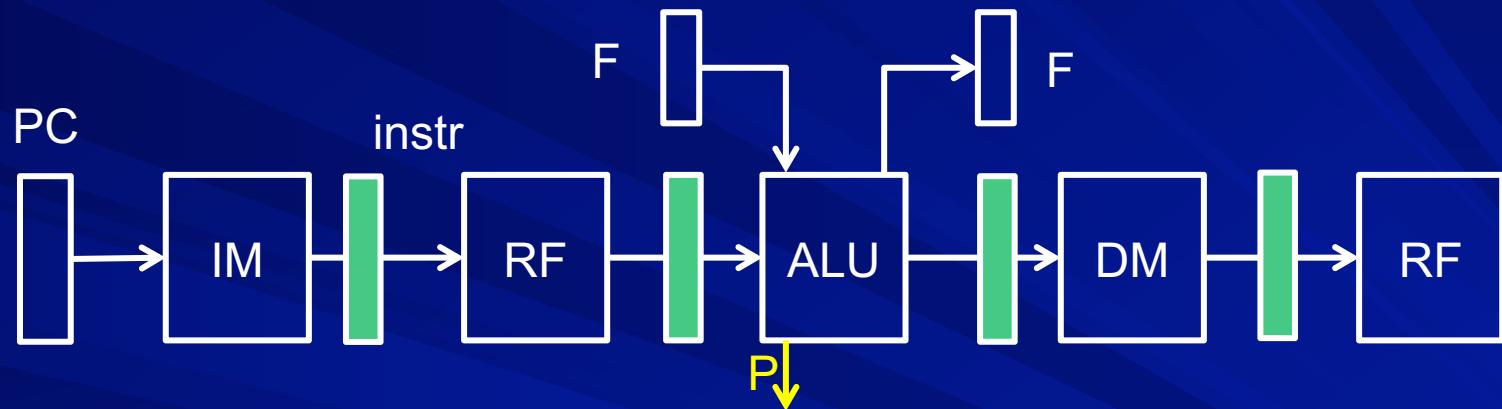
# Multi-cycle datapath

Resource sharing possible across cycles

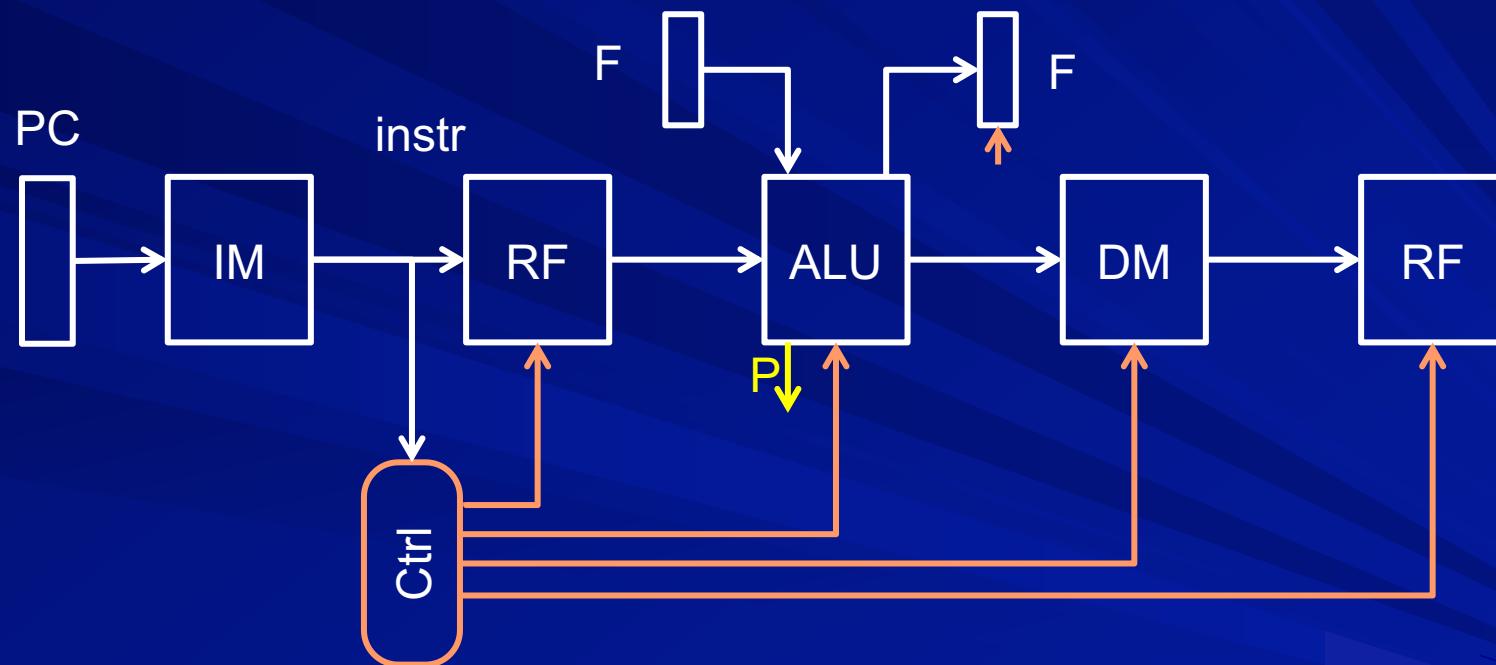


# Pipelined datapath

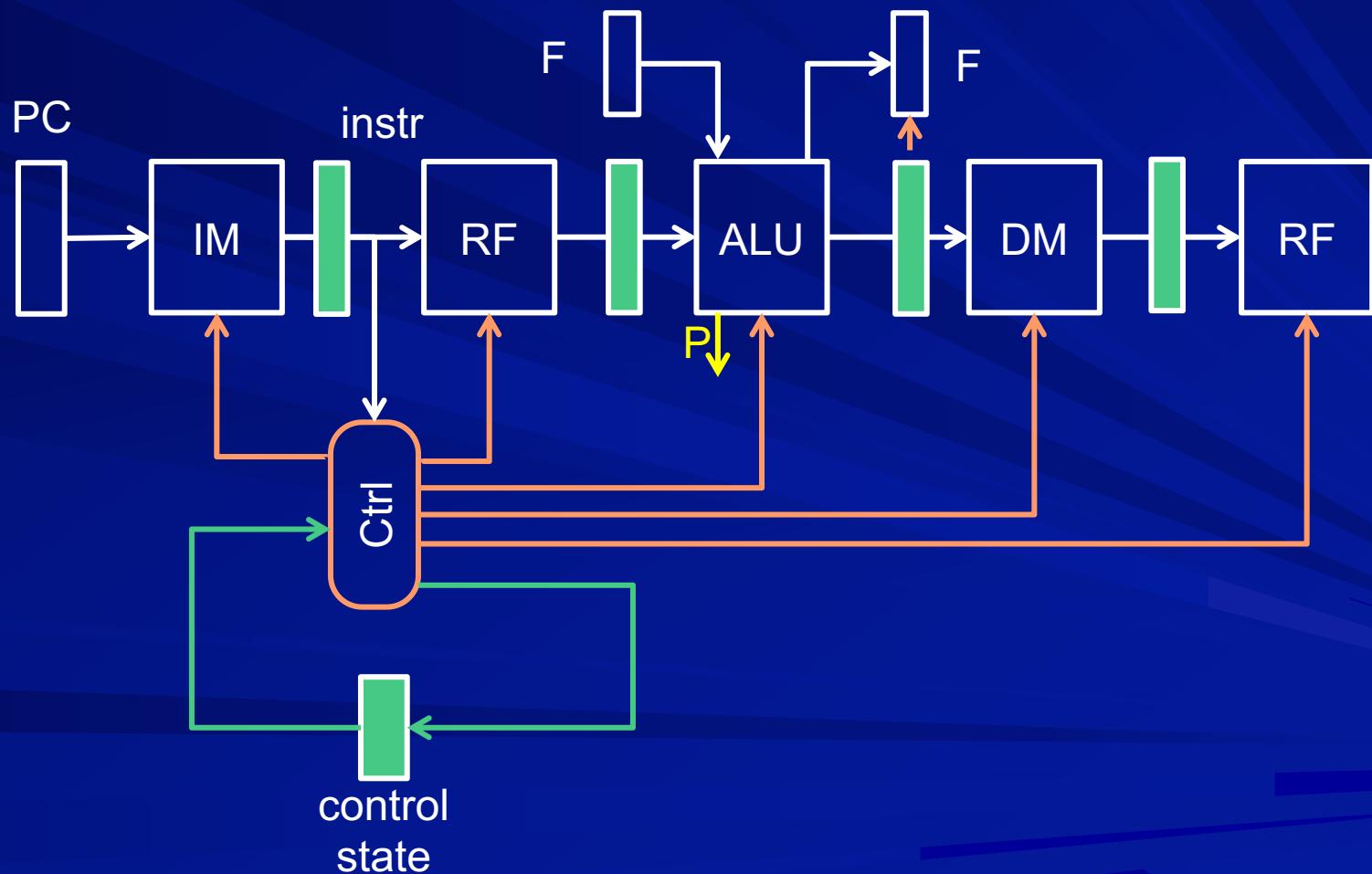
Resource sharing leads to hazards



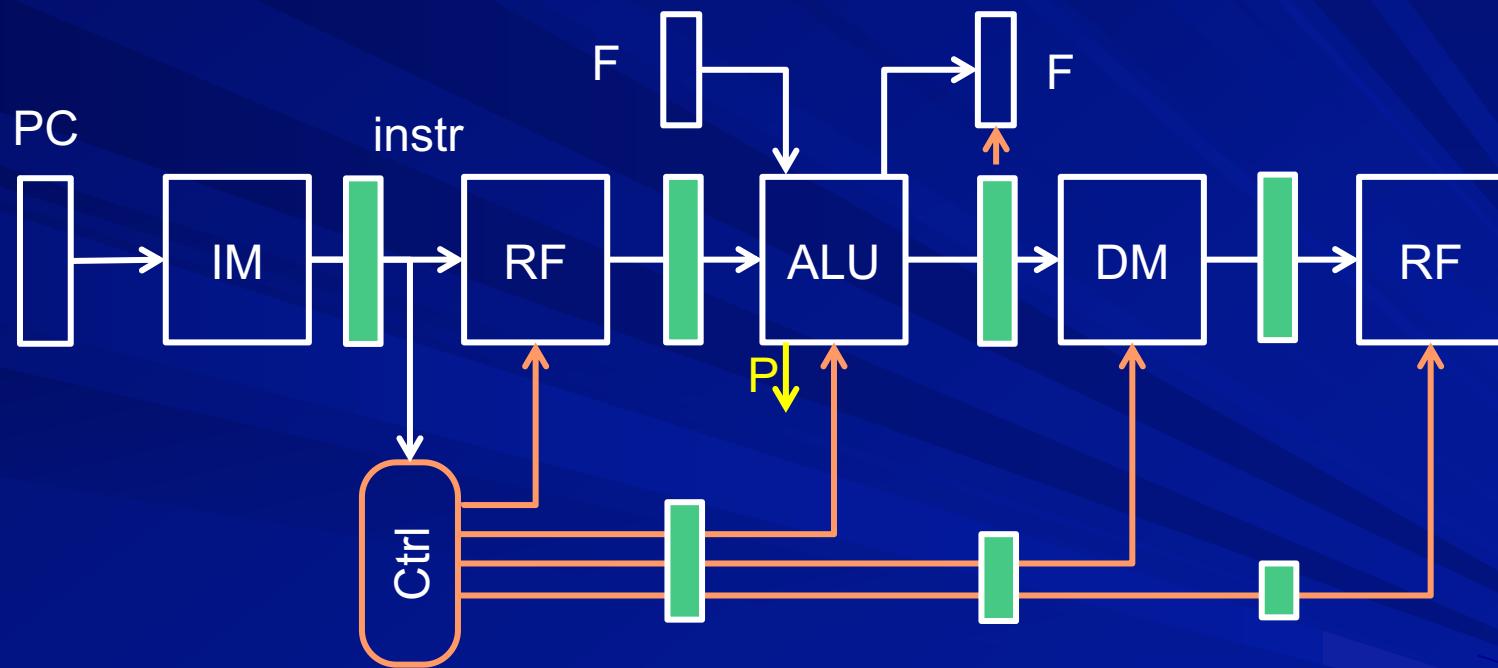
# Controller for single cycle DP



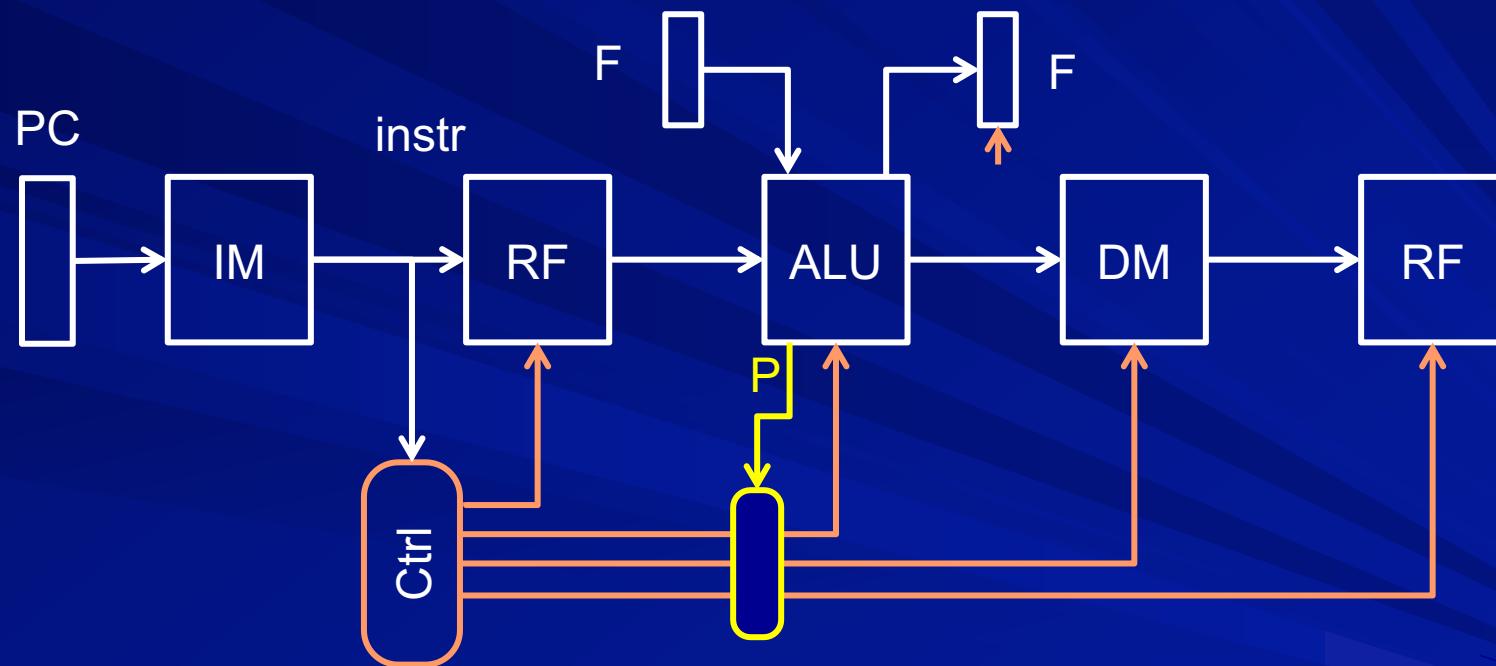
# Controller for multi-cycle DP



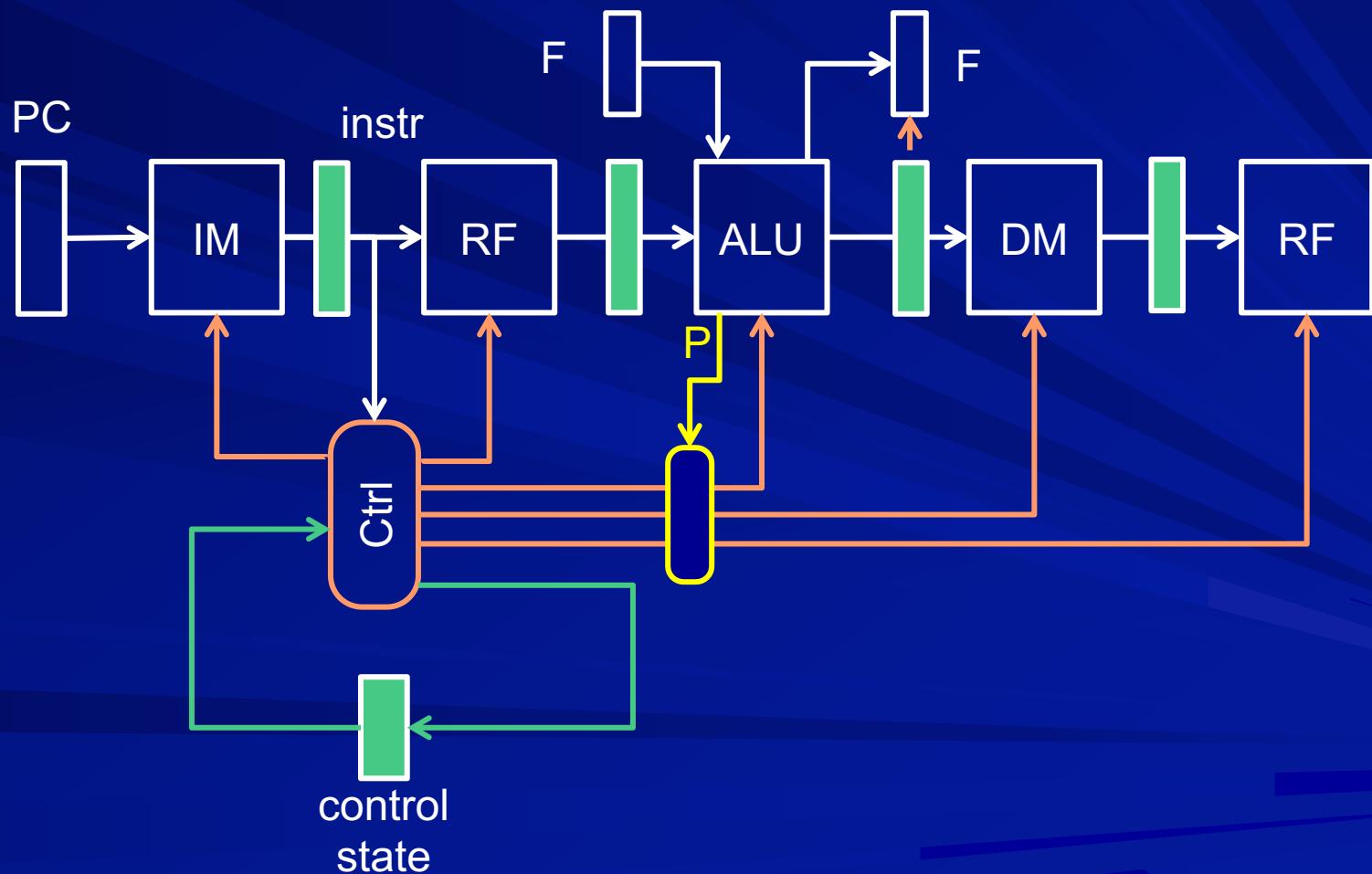
# Controller for pipelined DP



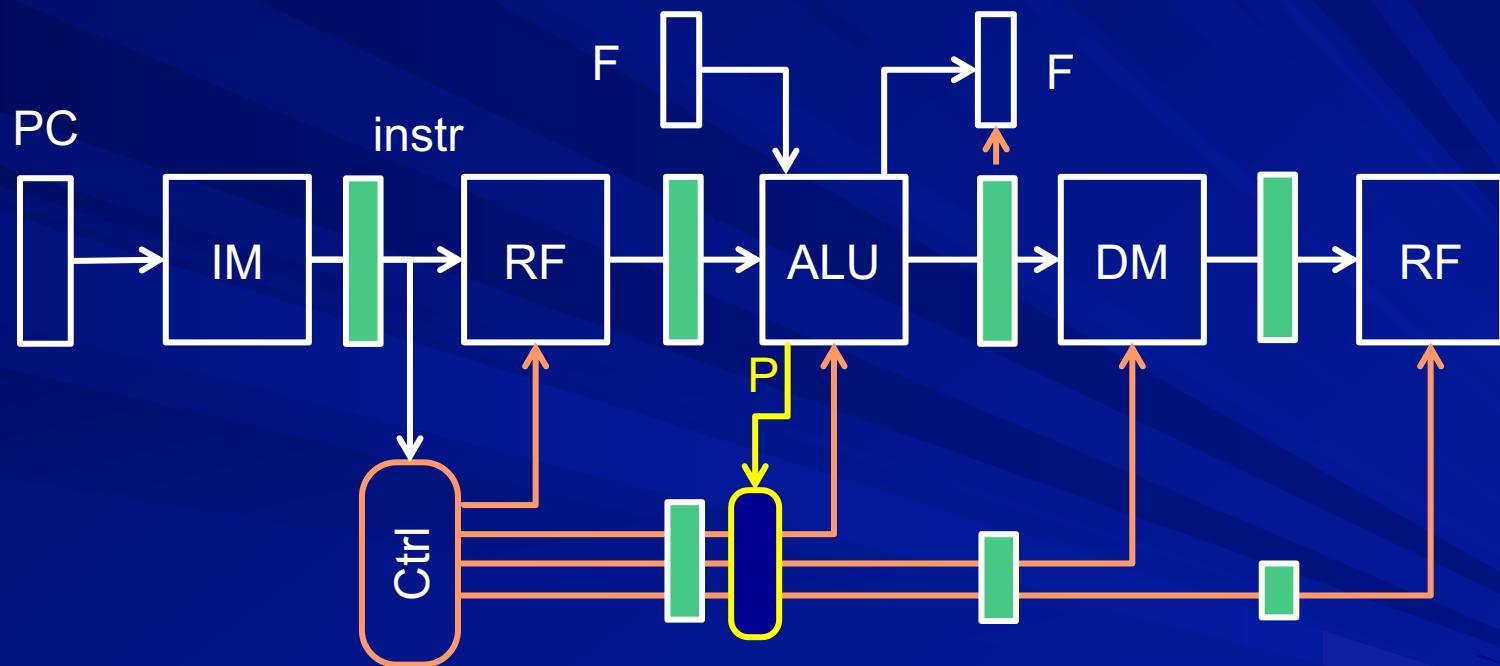
# Controller for single cycle DP



# Controller for multi-cycle DP



# Controller for pipelined DP



# Thanks