VE370 RC5

L8, L9, Project 2

Structure Harzard

--A required source is busy

Solution: separate instruction/data caches

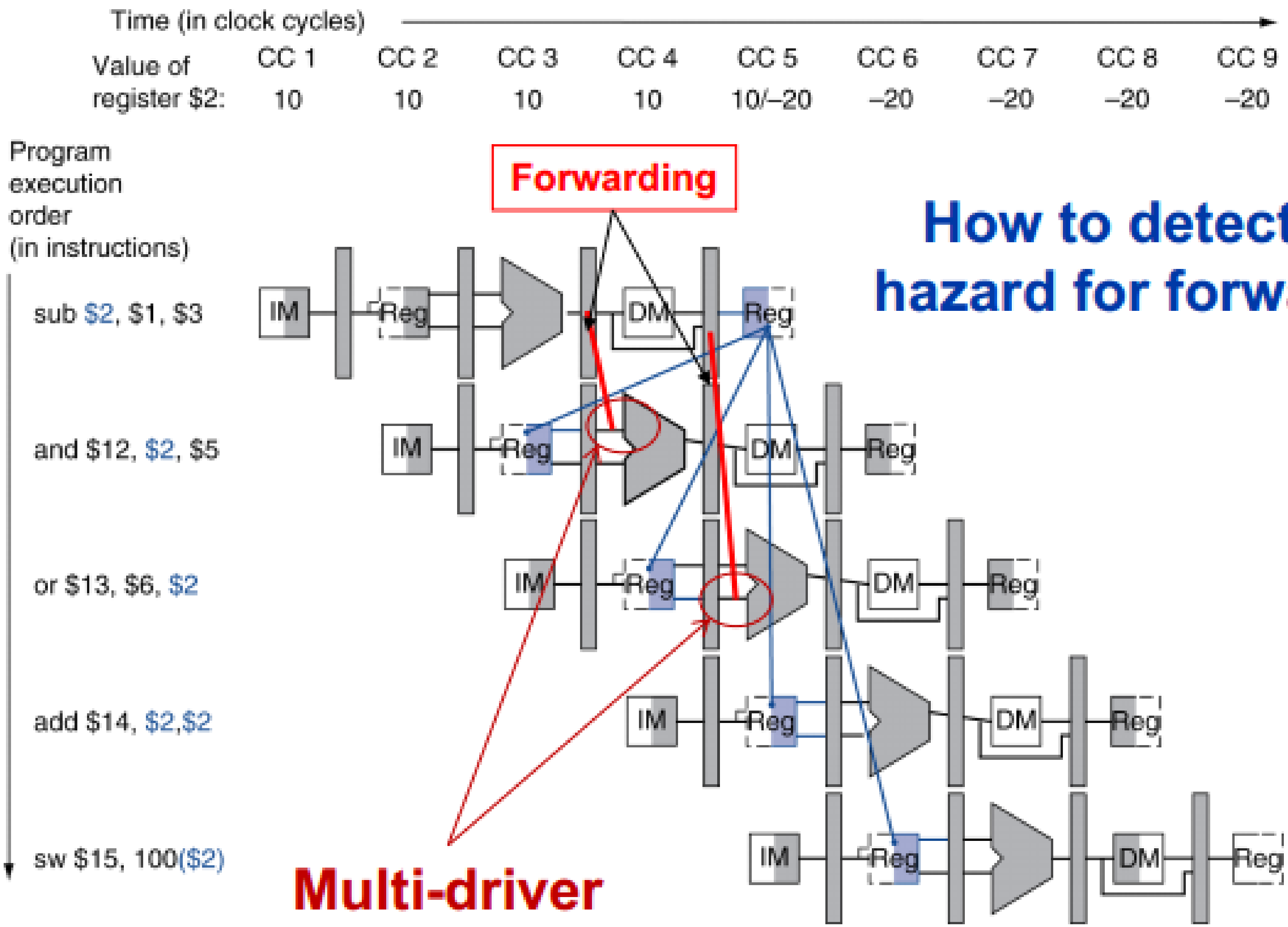
Data Hazard

--Data Dependency: need to wait for previous instruction

Solution: 1. Add stalls: how many nops should be inserted

2. Forwarding (By passing): Use result as soon as it is computed. Apply Forwarding unit [Mux] to decide which value we want to use

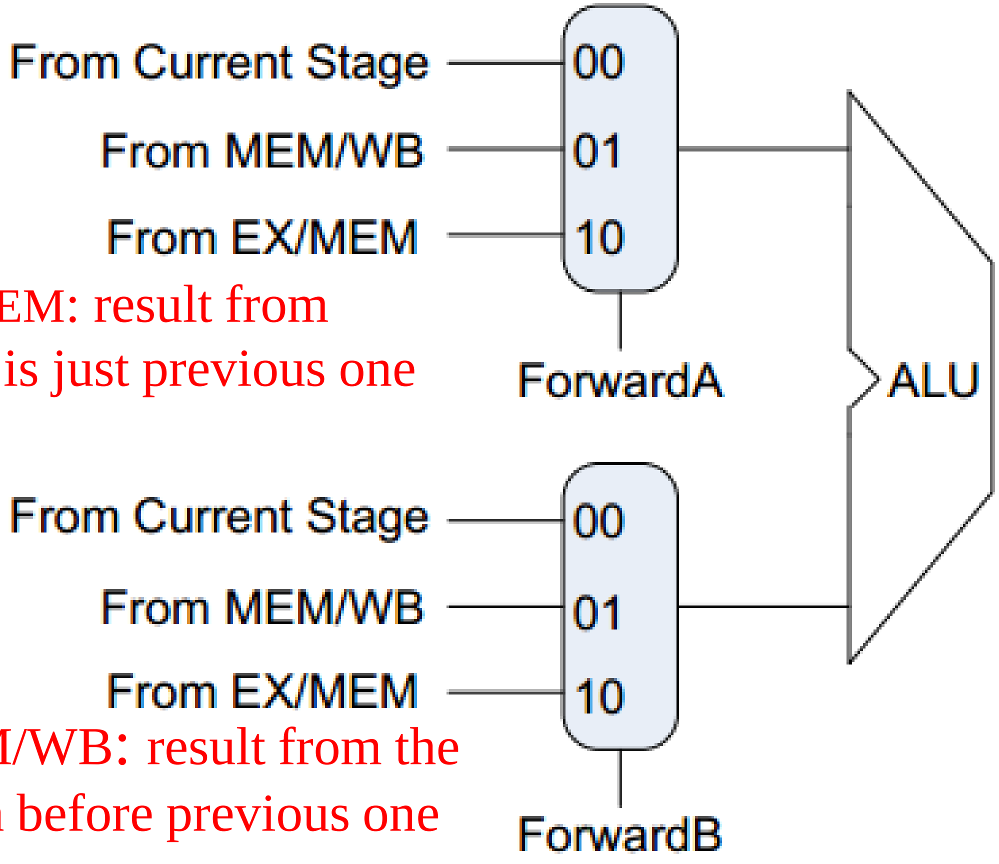
Dependencies & Forwarding



How to detect data hazard for forwarding

Forwarding Path

- Forwarding paths are created between stage pipeline registers and ALU inputs
  - By using MUXes



From EX/MEM: result from instruction that is just previous one

From MEM/WB: result from the instruction before previous one

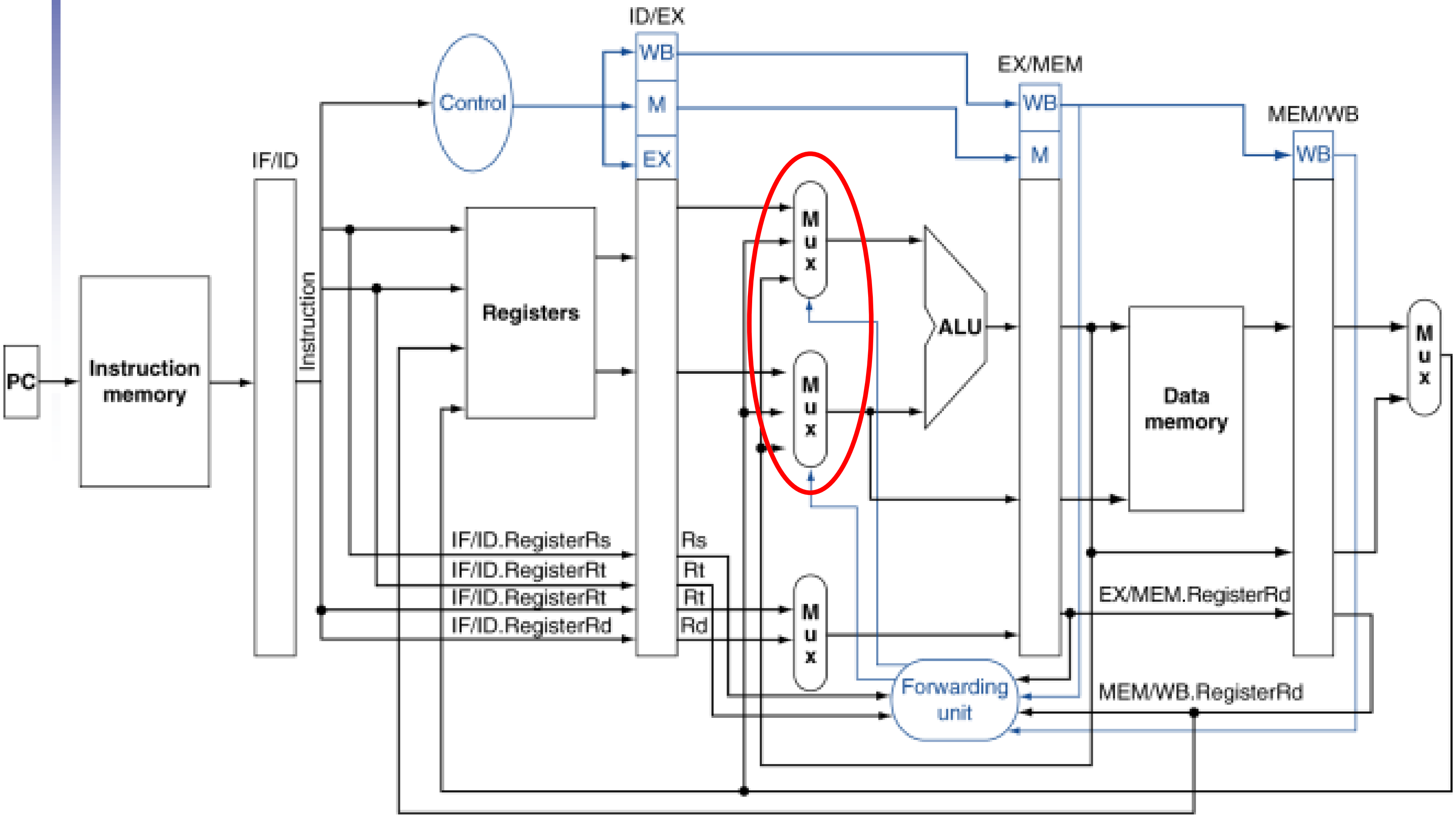
MEM hazard (not EX hazard)

- If (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
  and (MEM/WB.RegisterRd = ID/EX.RegisterRs)
  and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
  and (EX/MEM.RegisterRd = ID/EX.RegisterRs)))
  ForwardA = 01

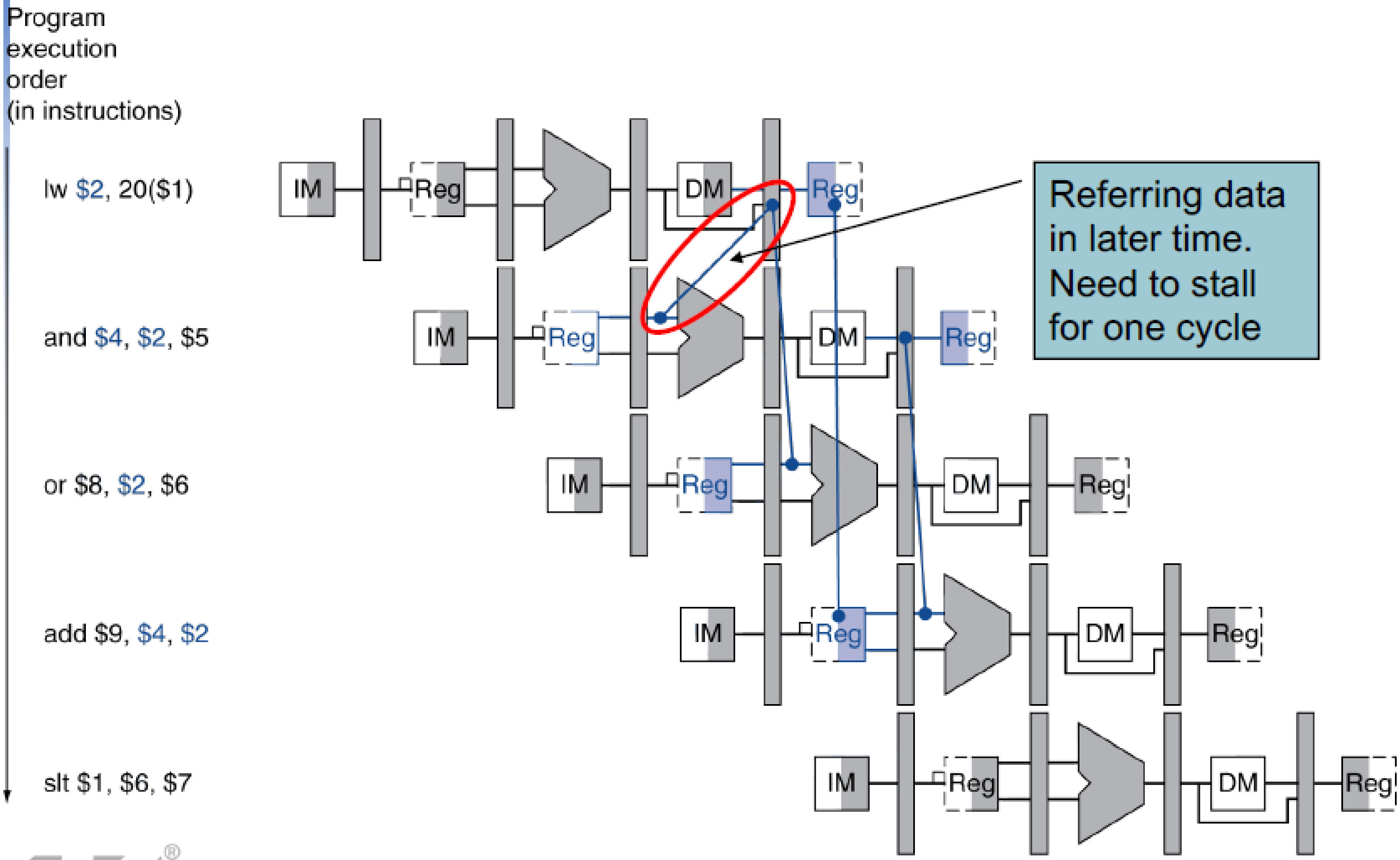
- If (MEM/WB.RegWrite and (MEM/WB.RegisterRd ≠ 0)
  and (MEM/WB.RegisterRd = ID/EX.RegisterRt)
  and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd ≠ 0)
  and (EX/MEM.RegisterRd = ID/EX.RegisterRt)))
  ForwardB = 01

Want to use most recent value when facing double data hazard

Datapath with Forwarding Unit

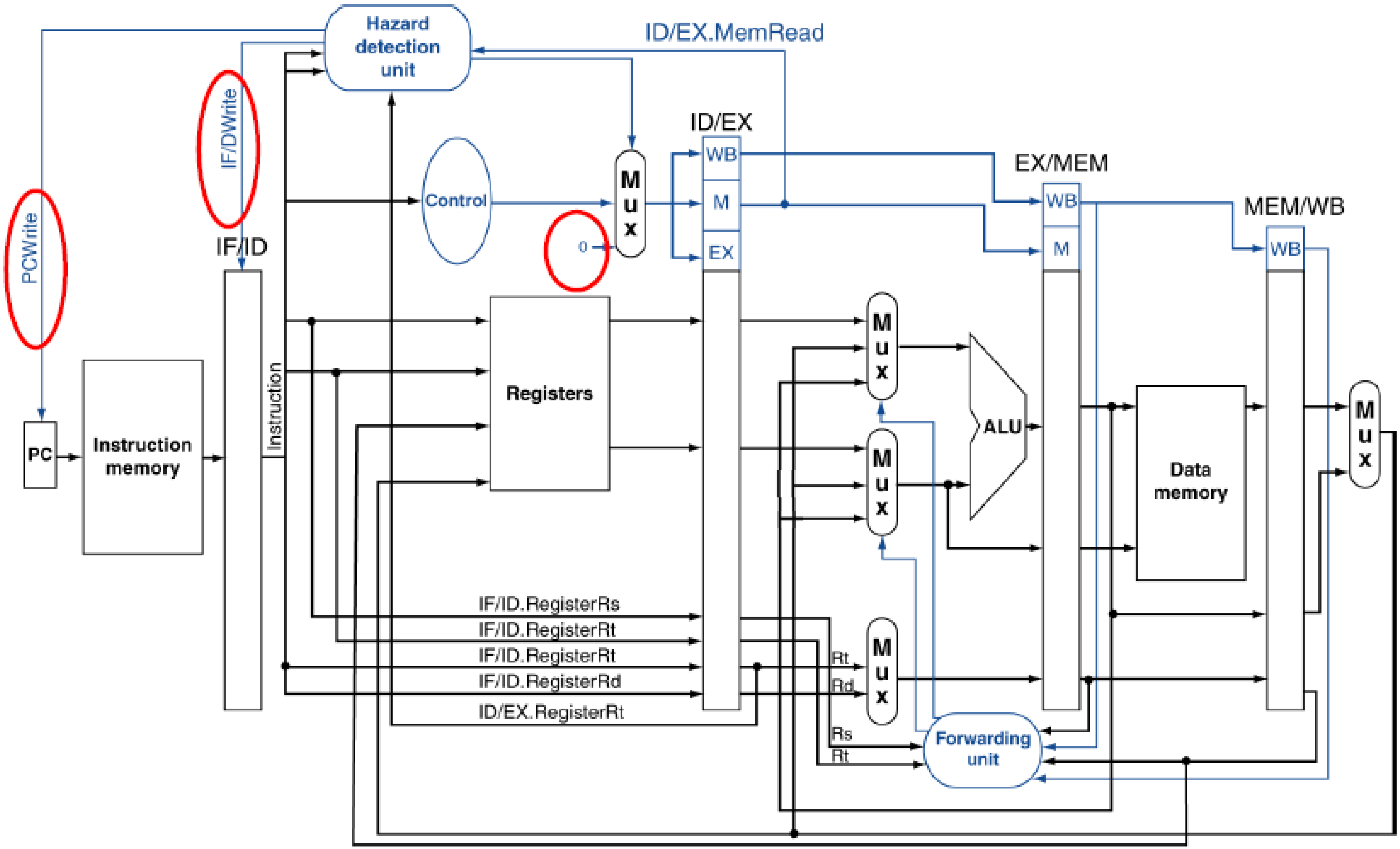


Load-Use Data Hazard



Referring data in later time. Need to stall for one cycle

Datapath with Hazard Detection



Control Hazard

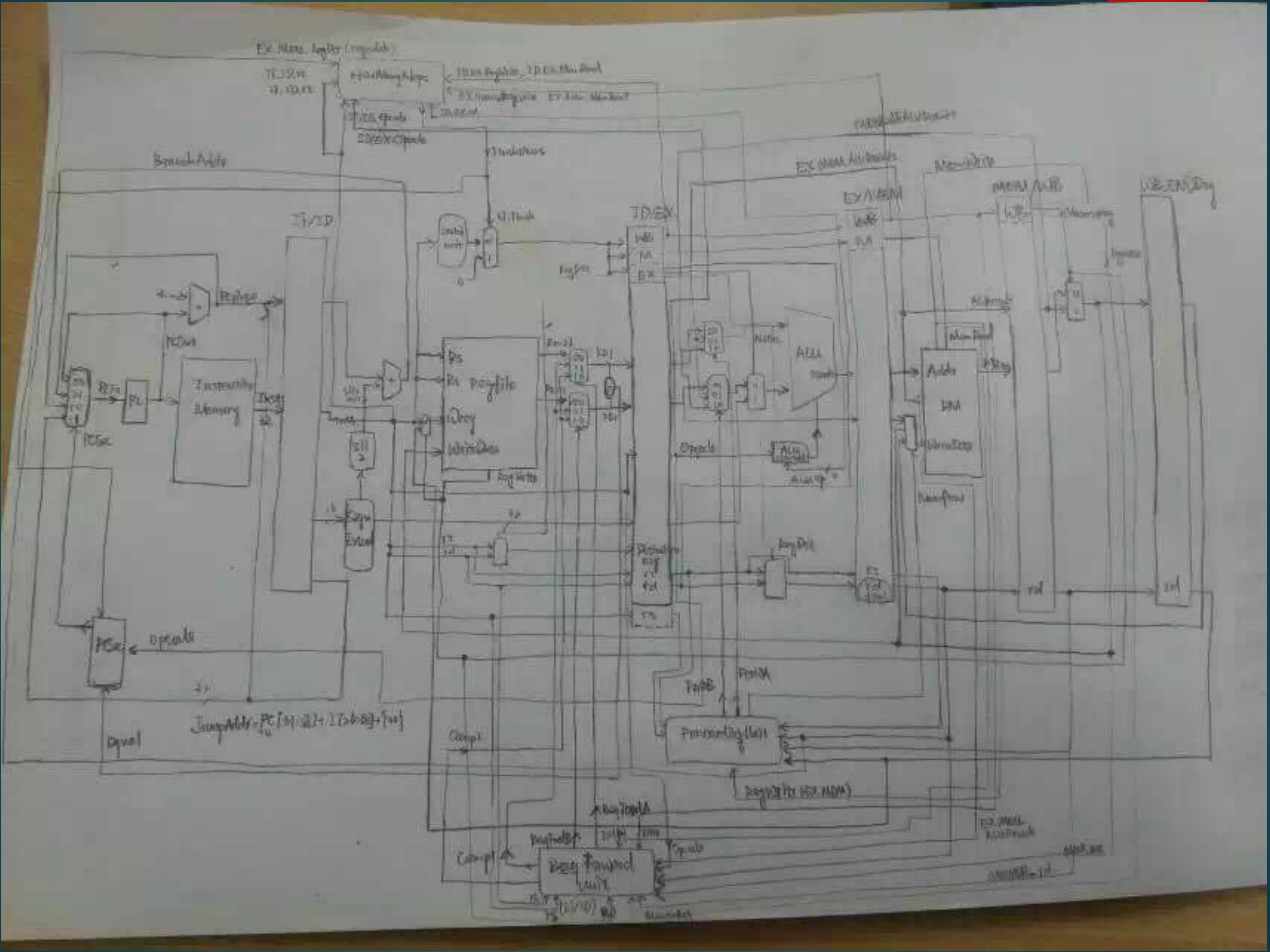
--Branch control signal comes late.
If no forwarding, 3 bubbles inserted if there is a branch

Solution: Compare early. Determine branch in ID stage. Only one bubble is need

-- Issue: Data hazards.

- 4 types of “Branch Hazard Resolution”

Project 2


EX.MEM.RegDst (regwrite)
IF.ID.rs
IF.ID.rt
HazardDetectionUnit
ID.EX.RegWrite, ID.EX.MemRead
EX.MEM.RegWrite EX.MEM.MemRead
ID.EX.Opcode
IF.ID.Opcode
ID/EX.Opcode
BranchAddr
IF.Flush
IF/ID
Control Unit
0
1
0
RegDst
ID/EX
WB
M
EX
EX/MEM
WB
M
MEM/WB
WB
WB_EN.Reg
EX.MEM.AluResult
MemWrite
PCAdd
PCSrc
PC
Instruction Memory
Inst
Rs
Rt
RegFile
WReg
WriteData
RegWrite
SLL 2
Sign Extend
rt
rd
ALUSrc
ALU
ALUop
ALU control
Opcode
AluResult
MemRead
Addr
DM
WriteData
RegDst
rd
rd
Opcode
PCSrc
JumpAddr=PC[31:28]+I[25:0]+{00}
Equal
Compt
ForwardB
ForwardA
ForwardingUnit
RegWrite(EX.MEM)
RegForwardA
RegForwardB
Opcode
Compt
Branch Forward Unit
rs
rt
EX.MEM.RdOrRt
MEM.WB.rd

Some advice

1. Try to draw the graph by yourself once, not only help for the project but also final. Understand the meaning of every bus.

2. Determine all the name of signals before you begin programming, also clarify them with your teammates for better cooperation

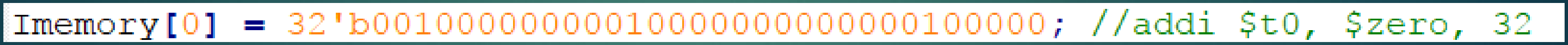
3. First test every component, then every stage, finally combine them together to debug, one file each component.

4. If not just one person work in a group, clarify the input and output name for everyone's part.

5. If only one person works… contact us early if you feel overwhelmed

1. Registers should always be written first then read

2. Instruction memory is initialized by yourself, we will give you the test case later. e.g.



```
Imemory[0] = 32'b00100000000010000000000000100000; //addi $t0, $zero, 32
```

3. Be careful with “always @”, all stages should be trigger at the same positive edge or negative edge
4. Need to implement forwarding in data hazard and control hazard.