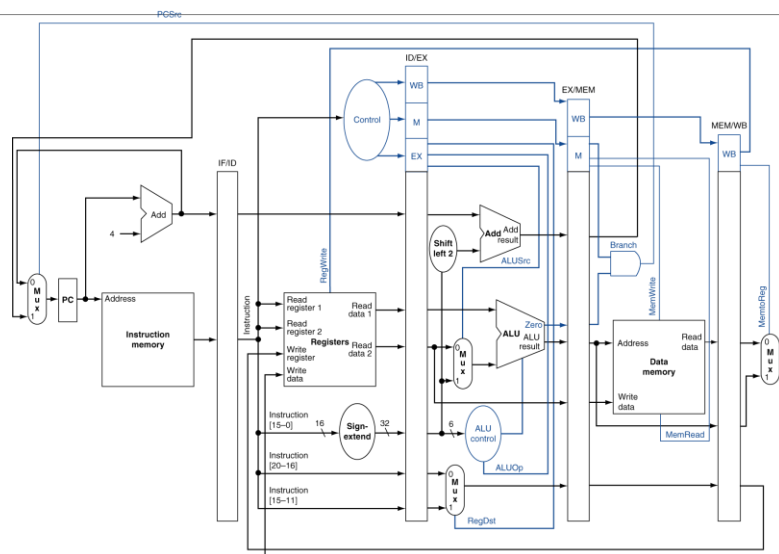


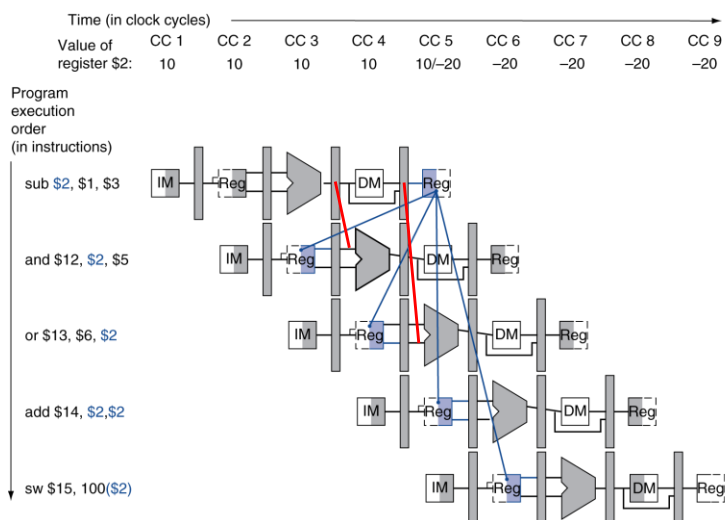
# Pipelined Control



## Data Hazards in ALU Instructions

- Consider this sequence:
  - sub \$2, \$1, \$3
  - and \$12, \$2, \$5
  - or \$13, \$6, \$2
  - add \$14, \$2, \$2
  - sw \$15, 100(\$2)
- We can resolve hazards with forwarding
  - How do we detect when to forward?

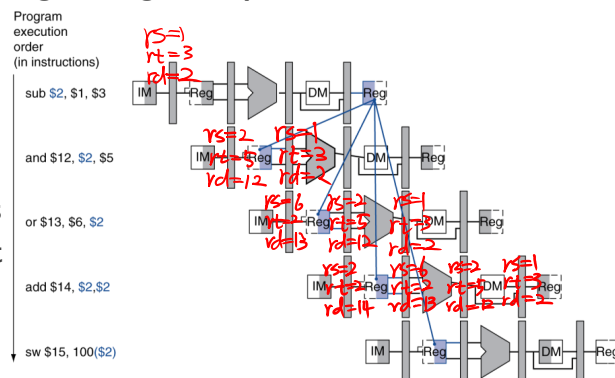
# Dependencies & Forwarding



CSULB

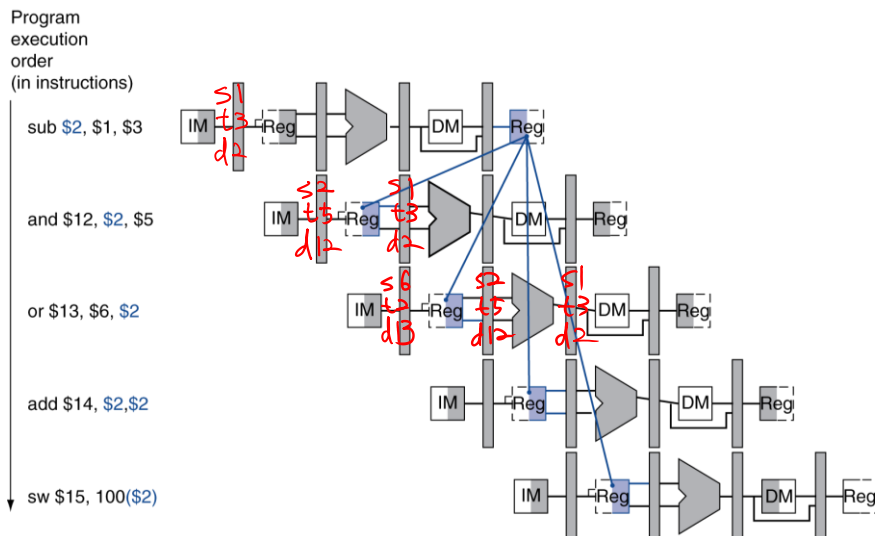
## Detecting the Need to Forward

- Pass register numbers along pipeline
  - e.g., ID/EX.RegisterRs = register number for Rs sitting in ID/EX pipeline register
- ALU operand register numbers in EX stage are given by
  - ID/EX.RegisterRs, ID/EX.RegisterRt
- Data hazards when
  - 1a. EX/MEM.RegisterRd = ID/EX.RegisterRs
  - 1b. EX/MEM.RegisterRd = ID/EX.RegisterRt
  - 2a. MEM/WB.RegisterRd = ID/EX.RegisterRs
  - 2b. MEM/WB.RegisterRd = ID/EX.RegisterRt



CSULB

## Detecting the Need to Forward



CSULB

## Detecting the Need to Forward

- But only if forwarding instruction will write to a register!
  - EX/MEM.RegWrite, MEM/WB.RegWrite
- And only if Rd for that instruction is not \$zero
  - EX/MEM.RegisterRd  $\neq$  0,  
MEM/WB.RegisterRd  $\neq$  0

CSULB

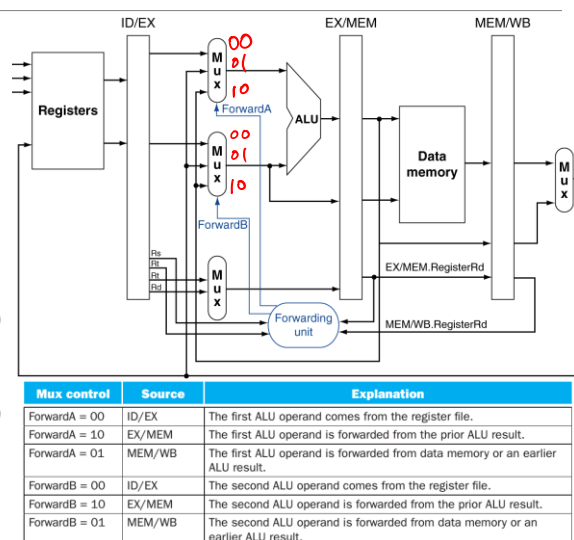
# Forwarding Conditions

## EX hazard

- if (EX/MEM.RegWrite and (EX/MEM.RegisterRd  $\neq$  0) and (EX/MEM.RegisterRd = ID/EX.RegisterRs))  
ForwardA = 10
- if (EX/MEM.RegWrite and (EX/MEM.RegisterRd  $\neq$  0) and (EX/MEM.RegisterRd = ID/EX.RegisterRt))  
ForwardB = 10

## MEM hazard

- if (MEM/WB.RegWrite and (MEM/WB.RegisterRd  $\neq$  0) and (MEM/WB.RegisterRd = ID/EX.RegisterRs))  
ForwardA = 01
- if (MEM/WB.RegWrite and (MEM/WB.RegisterRd  $\neq$  0) and (MEM/WB.RegisterRd = ID/EX.RegisterRt))  
ForwardB = 01



CSULB

# Double Data Hazard

## Consider the sequence:

add \$1, \$1, \$2

add \$1, \$1, \$3

add \$1, \$1, \$4

## Both hazards occur

- Want to use the most recent

## Revise MEM hazard condition

- Only fwd if EX hazard condition isn't true

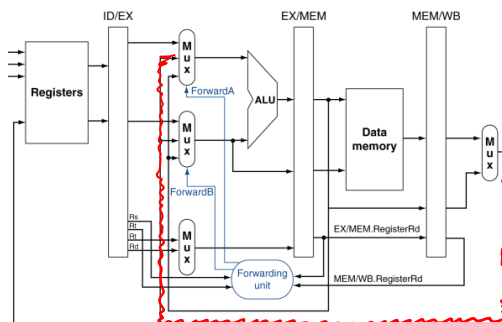
CSULB

# Revised Forwarding Condition

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

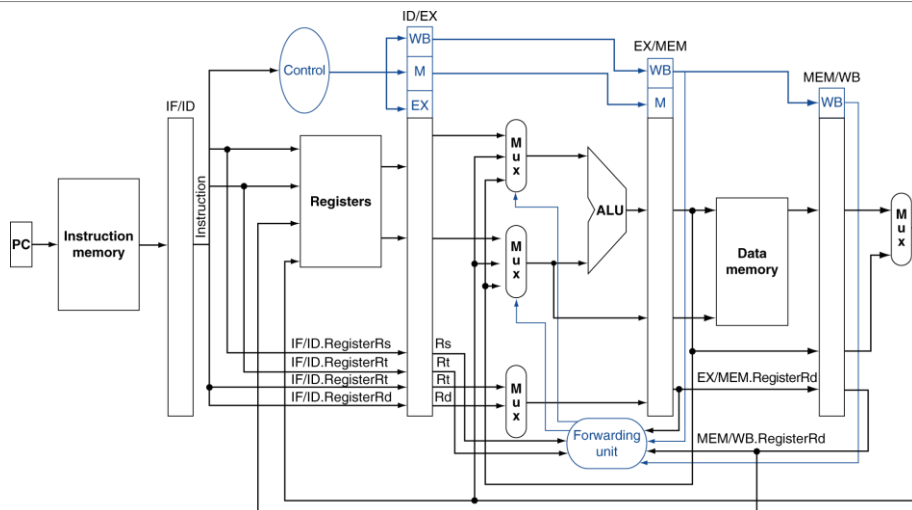
## MEM hazard

- if (MEM/WB.RegWrite and (MEM/WB.RegisterRd  $\neq$  0)  
and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd  $\neq$  0)  
and (EX/MEM.RegisterRd = ID/EX.RegisterRs) )  
and (MEM/WB.RegisterRd = ID/EX.RegisterRs))  
ForwardA = 01
- if (MEM/WB.RegWrite and (MEM/WB.RegisterRd  $\neq$  0)  
and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd  $\neq$  0)  
and (EX/MEM.RegisterRd = ID/EX.RegisterRt) )  
and (MEM/WB.RegisterRd = ID/EX.RegisterRt))  
ForwardB = 01



CSULB

# Datapath with Forwarding



CSULB