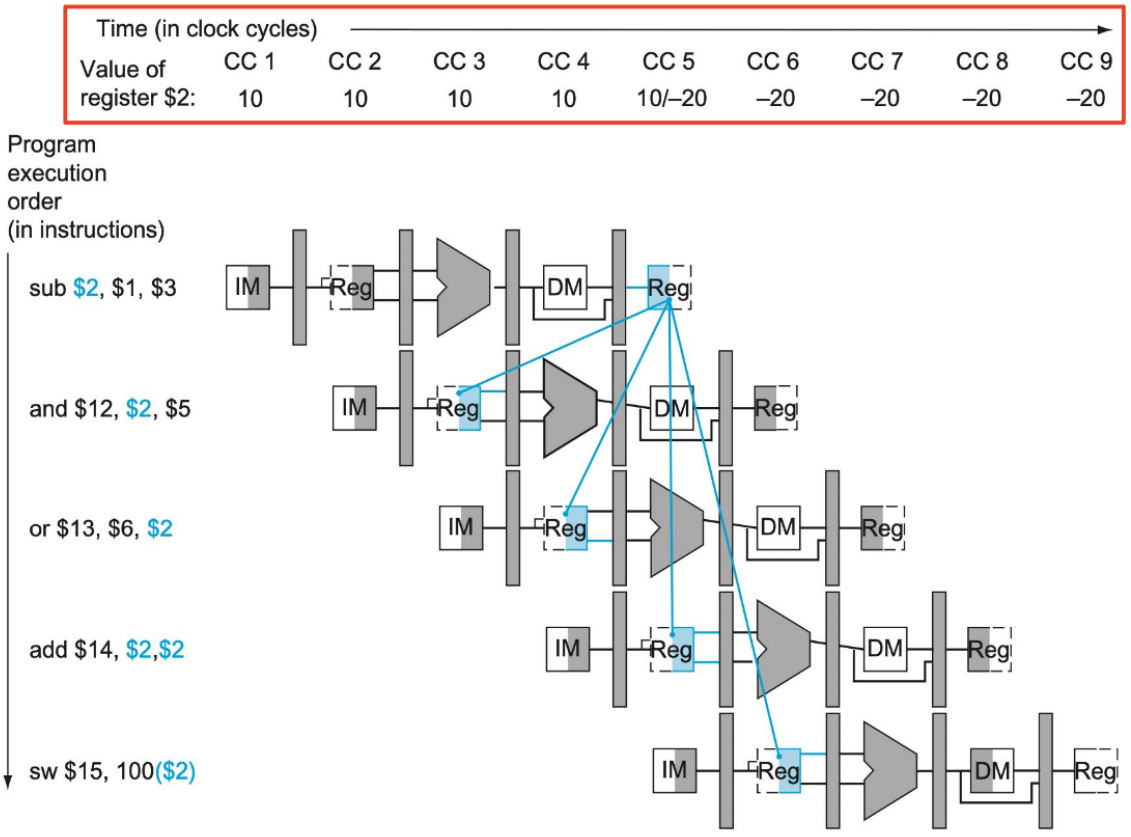
Computer Architecture (ENE1004)

▼ Lec 15

Lec 15: The Processor 8

Data Hazard - An Example Scenario (1)

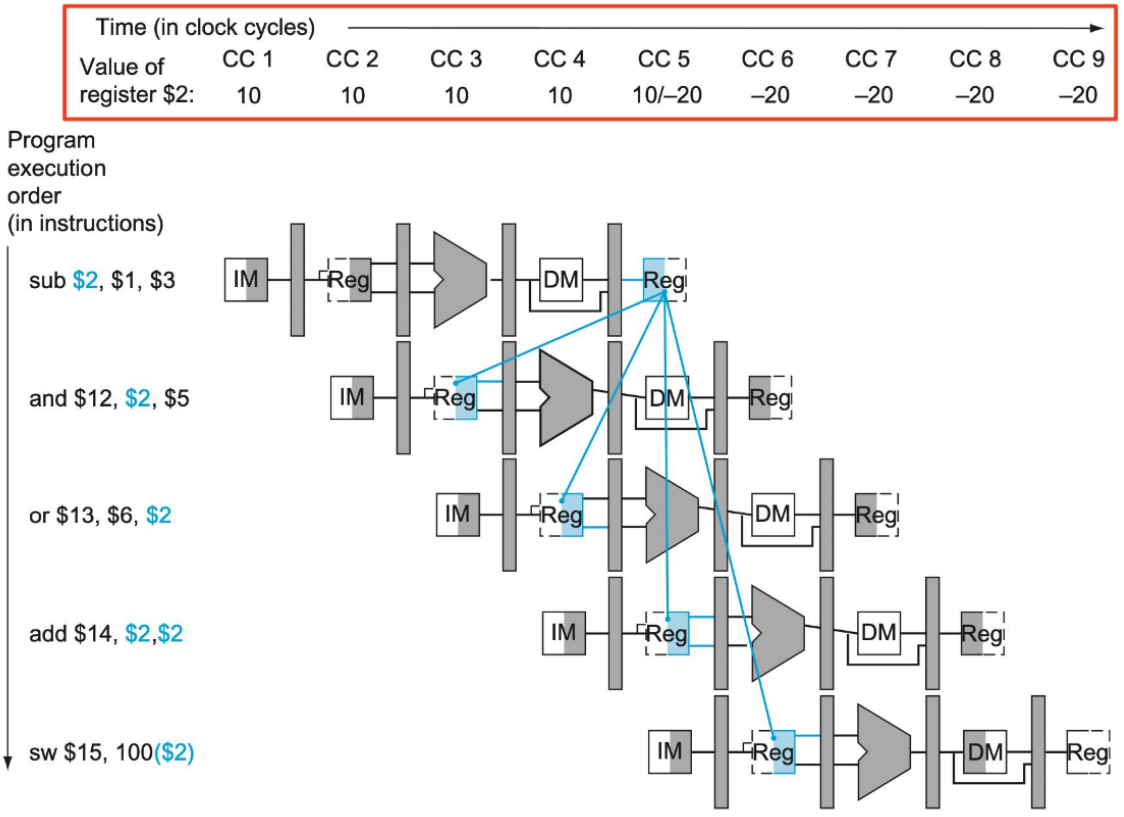


- Figure
  - Clock cycle (CC) : left → right
  - Instructions: top → down
- Instructions scenario: The destination register of 1st instruction ($2) is the source registers of following instructions (명령어 시나리오: 첫 번째 명령어($2)의 대상 레지스터는 다음 명령어의 소스 레지스터이다.)
- Assumption (가정)

- $2 holds “10” at first ($2는 처음에 “10”을 보유한다.)
- $2 is expected to hold “-20” after executing 1st instruction (sub $2,$1,$3) ($2는 첫 번째 명령어 (sub $2, $1, $3)을 실행 후 “-20”을 보유할 것으로 예상된다.)

- Let us focus on the value of $2 ($2의 값에 집중을 해보자.)
  - 1st instruction writes “-20” into $2 in WB stage in CC5 (첫 번째 명령어는 CC5의 WB 단계에서 “-20”을 $2에 쓴다.)
  - So, $2 holds “10” till CC4; $2 holds “-20” from CC5 (따라서 $2는 CC4까지 “10”을 보유하고, CC5 부터는 “-20”을 보유한다.)

Data Hazard - An Example Scenario (2)

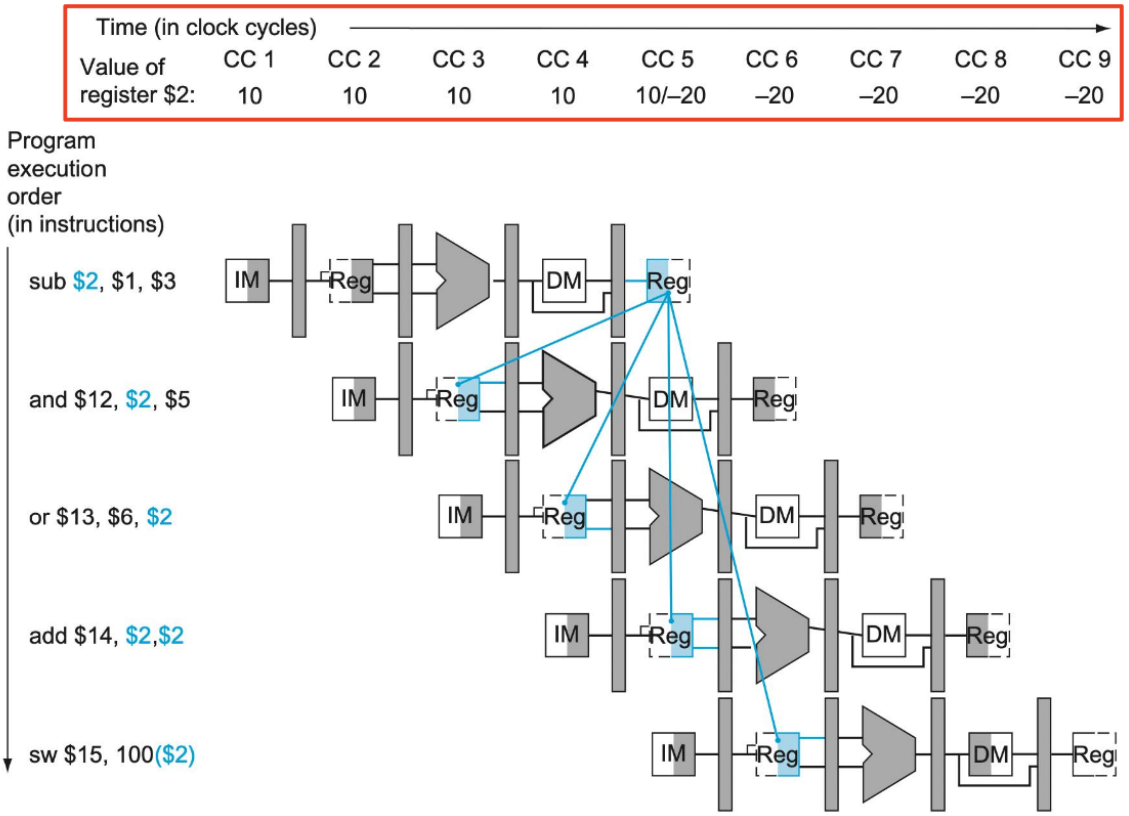


- and $12, $2, $5
  - The 2nd instruction expects to read the new value “-20” from $2 in the ID stage at CC3
  - However, $2 holds the old value “10” at CC3
- or $13, $6, $2

- The 3rd instruction expects to read the new value “-20” from $2 in the ID stage at CC4
- However, $2 holds the old value “10” at CC4

- We call these “data hazard”
  - A destination register of an instruction is read by following instructions as source registers (명령어의 대상 레지스터는 소스 레지스터와 같이 다음 명령어를 통해 읽는다.)

Data Hazard - An Example Scenario (3)

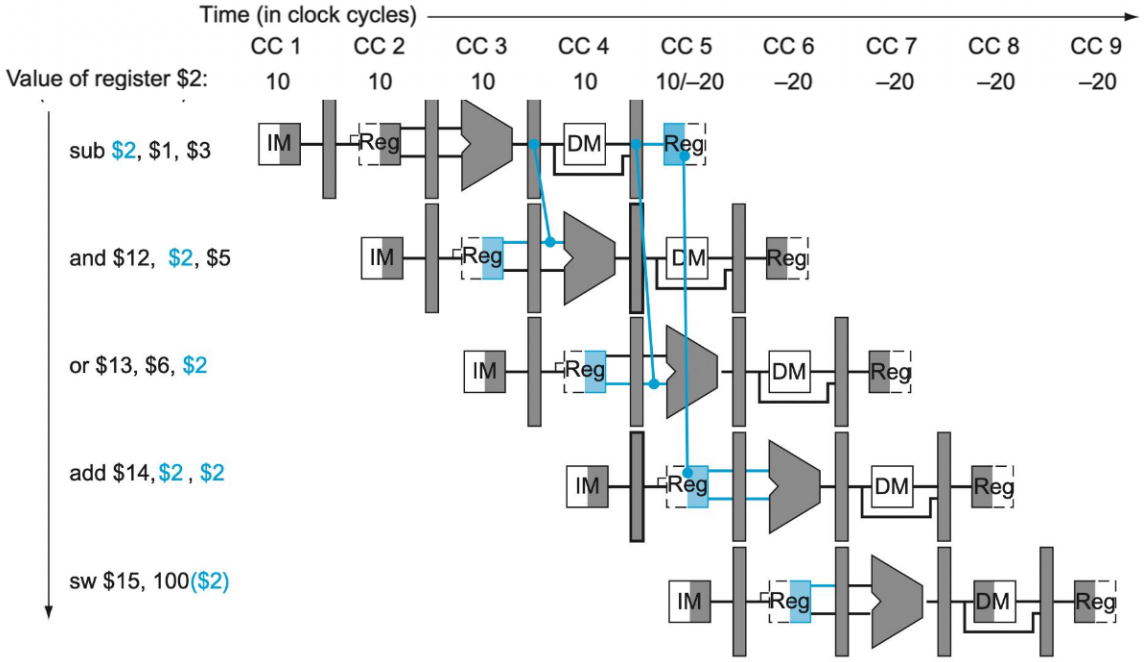


- add $14, $2, $2
  - The 3rd instruction expects to read the new value “-20” from $2 in the ID stage at CC5 (세 번째 명령은 CC5의 ID 단계에서 $2의 새 값 “-20”을 읽을 것으로 예상한다.)
  - The new value “-20” is written into $2 in the first half of CC5 by sub, which can be read by add in the second half of CC5 (새로운 값 “-20”은 CC5의 전반부에서 sub에 의해 $2에 쓰여지고, CC5의 후반부에서 덧셈으로 읽을 수 있다.)

- This is NOT a data hazard

- sw $15, 100($2)
  - The 4th instruction expects to read the new value “-20” from $2 in the ID stage at CC6 (4번쨔 명령은 CC6의 ID 단계에서 $2에서 새 값 “-20”을 읽을 것으로 예상한다.)
  - The new value “-20” is written into $2 at CC5, which can be read at CC6 (새 값 “-20”은 CC5에서 $2에 기록되며, 이는 CC6에서 읽을 수 있다.)
  - There is NO problwm at all

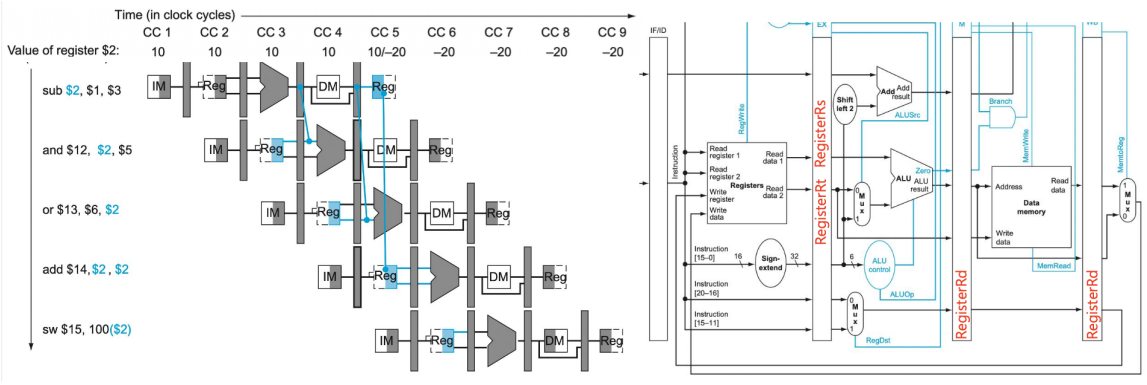
Data Forwarding (Solution) - An Example Scenario



- The new value “-20” is generated by ALU at the end of EX stage at CC3 (EX/MEM register) (새로운 값 “-20”은 CC3 (EX/MEM 레지스터)에서 EX 단계가 끝날 때 ALU에 의해 생성된다.)
- How about sending (forwarding) the new value to the sources of ALU? (이 새로운 값을 ALU의 소스로 전송(forwarding) 하는 것은 어떨까?)
  - For and, we can send the value from EX/MEM register to Rs (1st) port of the ALU at CC4 (and의 경우, EX/MEM 레지스터의 값을 CC4에서 ALU의 Rs(1번째) 포트에 전송할 수 있다.)
  - For or, we can send the value from MEM/WB register to Rt (2nd) port of the ALU at CC5 (or의 경우, CC5에서 MEM/WB 레지스터의 값을

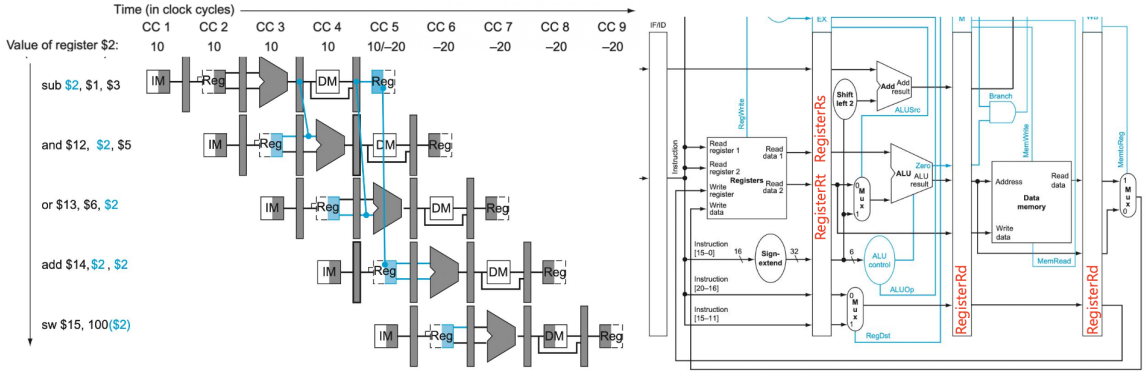
ALU의 Rt (2번째) 포트로 전송할 수 있다.)

Data Hazard Detection - An Example Scenario



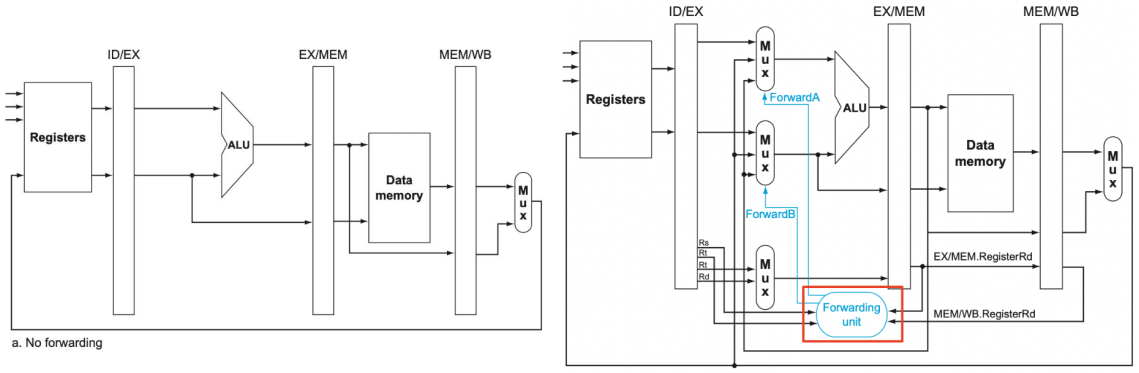
- Datapath should be able to detect when data hazard occurs (and forwarding is performed) (데이터 경로는 data hazard(위험) 발생 (및 forward 수행) 시점을 감지할 수 있어야 한다.)
- We need to check whether the destination register ($2) is used as a source register ($2) (대상 레지스터 ($2)가 소스 레지스터 ($2)로 사용되는지 확인해야 한다.)
    - For and, is EX/MEM.RegisterRd (of sub) equal to ID/EX.RegisterRs (of and)? (and의 경우, EX/MEM.RegisterRd(sub의)가 ID/EX.RegisterRs(and의)와 같은가?
    - For or, is MEM/WB.RegisterRd (of sub) equal to ID/EX.RegisterRt (of or)? (or의 경우, MEM/WB.Register(sub의)가 ID/EX.RegisterRt(or의)와 같은가?)

Data Hazard Detection - Generalization



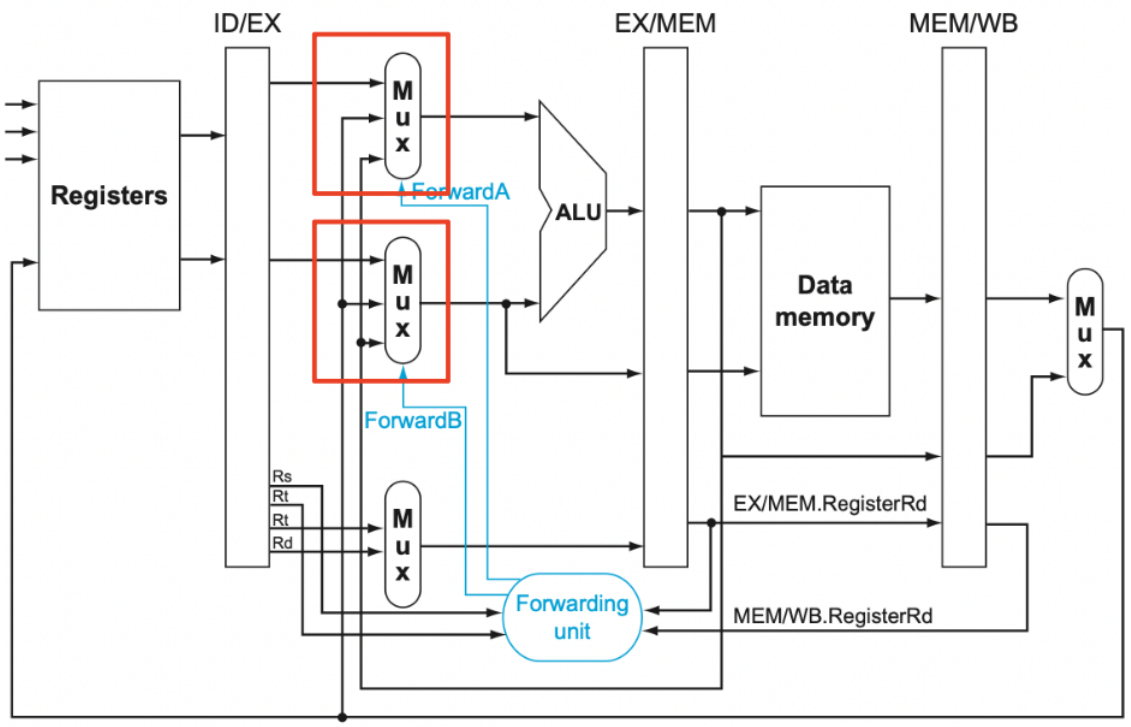
- Four different cases where the data hazards occur (data hazards(위험)이 발생하는 4가지 경우)
  - EX/MEM.RegisterRd = ID/EX.RegisterRs (btw 1st and 2nd instructions; 1st source of 2nd instruction) (1번째와 2번째 명령어, 2번째 명령어의 1번째 소스)
  - EX/MEM.RegisterRd = ID/EX.RegisterRt (btw 1st and 2nd instructions; 2nd source of 2nd instruction) (1번째와 2번째 명령어, 2번째 명령어의 2번째 소스)
  - MEM/WB.RegisterRd = ID/EX.RegisterRs (btw 1st and 3rd instructions; 1st source of 3rd instruction) (1번째와 3번째 명령어, 3번째 명령어의 1번째 소스)
  - MEM/WB.RegisterRd = ID/EX.RegisterRt (btw 1st and 3rd instructions; 2nd source of 3rd instruction) (1번째와 3번째 명령어, 3번째 명령어의 2번째 소스)

Data Forwarding - Implementation (1)



- We place the forwarding unit, which detects data hazard using the four inputs (네 개의 입력을 사용하여 data hazard를 감지하는 forwarding unit을 배치한다.
  - EX/MEM.RegisterRd
  - MEM/WB.RegisterRd
  - ID/EX.RegisterRs
  - ID/EX.RegisterRt

Data Forwarding - Implementation (2)

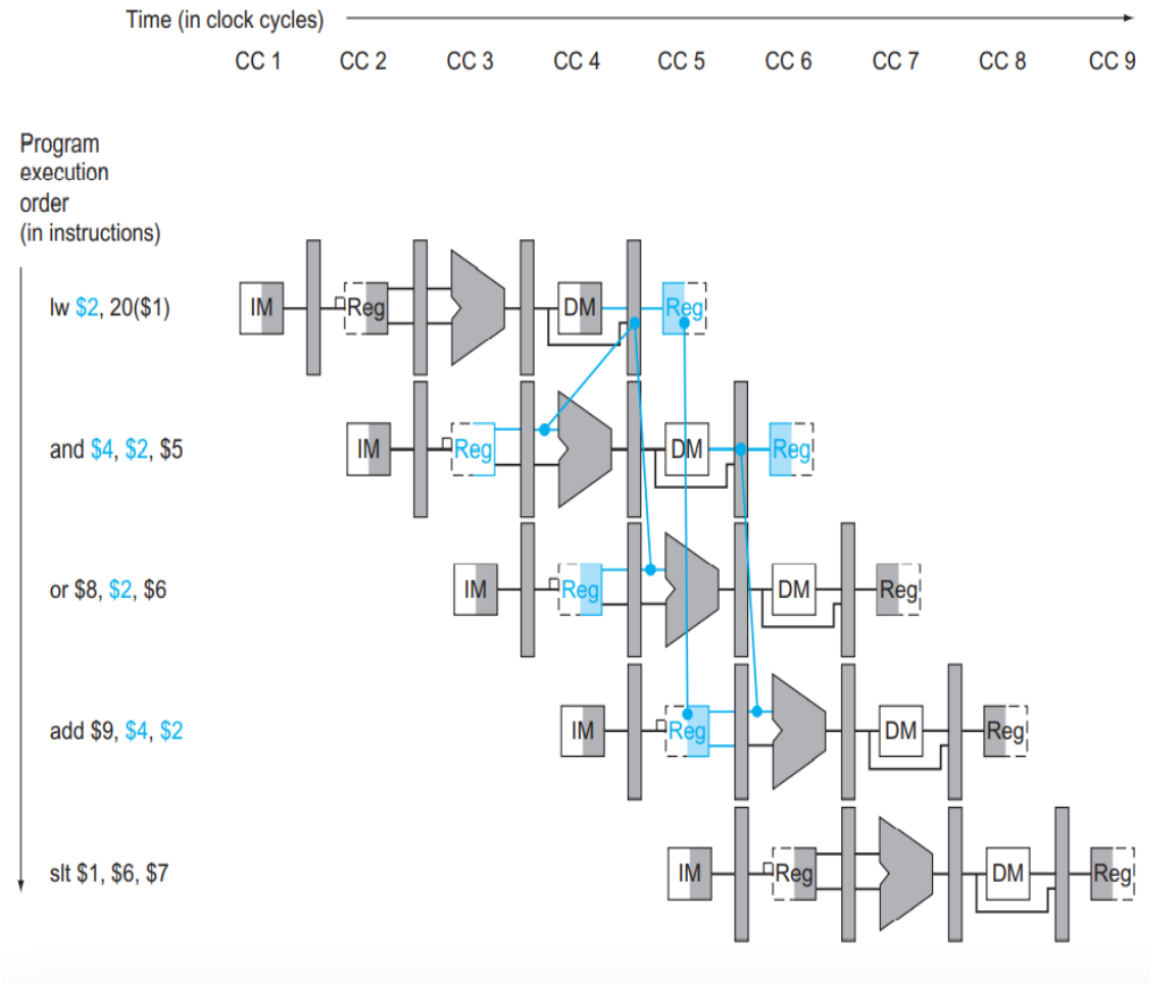


b. With forwarding

- We place multiplexers for two sources of ALU; the inputs of the multiplexers are (두 개의 ALU 소스에 대해 multiplexers를 배치한다. ; multiplexers의 입력은 다음과 같다.)
  - (1) from register file (ID/EX) - no data hazard (레지스터 파일(ID/EX)에서 - 데이터 위험 없음)
  - (2) from data memory or from an earlier ALU result (MEM/WB) (데이터 메모리 또는 이전 ALU 결과(MEM/WB)로부터의 데이터 입력)
  - (3) from the prior ALU result (EX/MEM) (이전 ALU 결과에서 (EX/MEM))
- The forwarding unit determines one of the three inputs using 2-bit signal (forwarding unit은 2비트 신호를 사용하여 세 가지 입력 중 하나를 결정한다.)
  - ForwardA for the first source of the ALU (ALU의 첫 번째 소스에 대한 ForwardA)
  - ForwardB for the second source of the ALU (ALU의 두 번째 소스에 대한 ForwardB)

Mux control	Source	Explanation
ForwardA = 00	ID/EX	The first ALU operand comes from the register file.
ForwardA = 10	EX/MEM	The first ALU operand is forwarded from the prior ALU result.
ForwardA = 01	MEM/WB	The first ALU operand is forwarded from data memory or an earlier ALU result.
ForwardB = 00	ID/EX	The second ALU operand comes from the register file.
ForwardB = 10	EX/MEM	The second ALU operand is forwarded from the prior ALU result.
ForwardB = 01	MEM/WB	The second ALU operand is forwarded from data memory or an earlier ALU result.

Data Hazard - Another Example Scenario



- There is a case where data forwarding cannot resolve the data hazard (데이터 전달로 데이터 위험을 해결할 수 없는 경우가 있다.)
  - Destination register of lw instruction (lw 명령의 대상 레지스터)
  - Next instruction attempts read the register (다음 명령어 시도는 레지스터를 읽는다)

    - $2 register in the example (예제에서 $2 레지스터)
- lw $2, 20($1)
    - The data is ready in MEM stage (MEM/WB register) at CC5 (데이터는 CC5의 MEM 단계 (MEM/WB 레지스터)에서 준비되었다.)
- and $4, $2, $5
    - The data is needed in EX stage at least by CC4 (EX 단계에서 적어도 CC4까지 데이터가 필요하다.)
- Data forwarding forwards the data from MEM/WB register to Rs port of the ALU; but, it is too late (데이터 포워딩은 MEM/WB 레지스터에서 ALU의 Rs 포트로 데이터를 전달하지만 너무 늦는다.)
    - How can we handle this problem? (이 문제를 어떻게 처리할 수 있을까?)