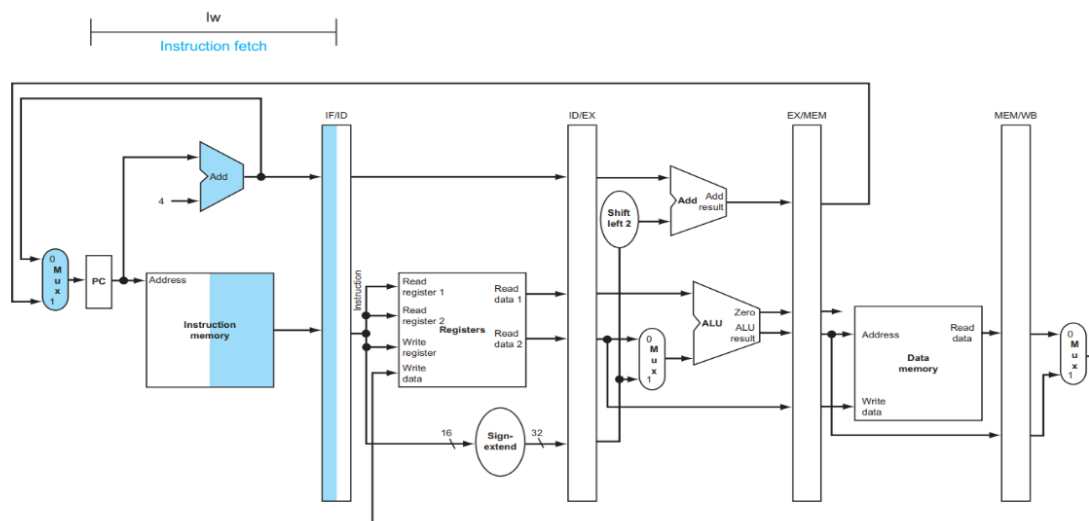


Computer Architecture (ENE1004)

▼ Lec 13

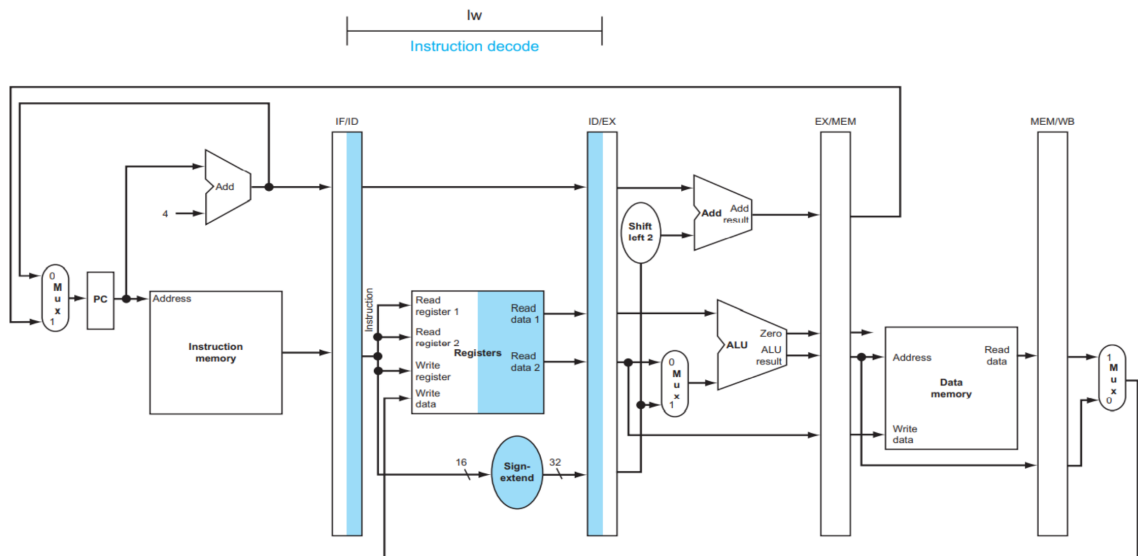
Lec 13: The Processor 6

lw Instruction in Pipelined Datapath - IF (1)



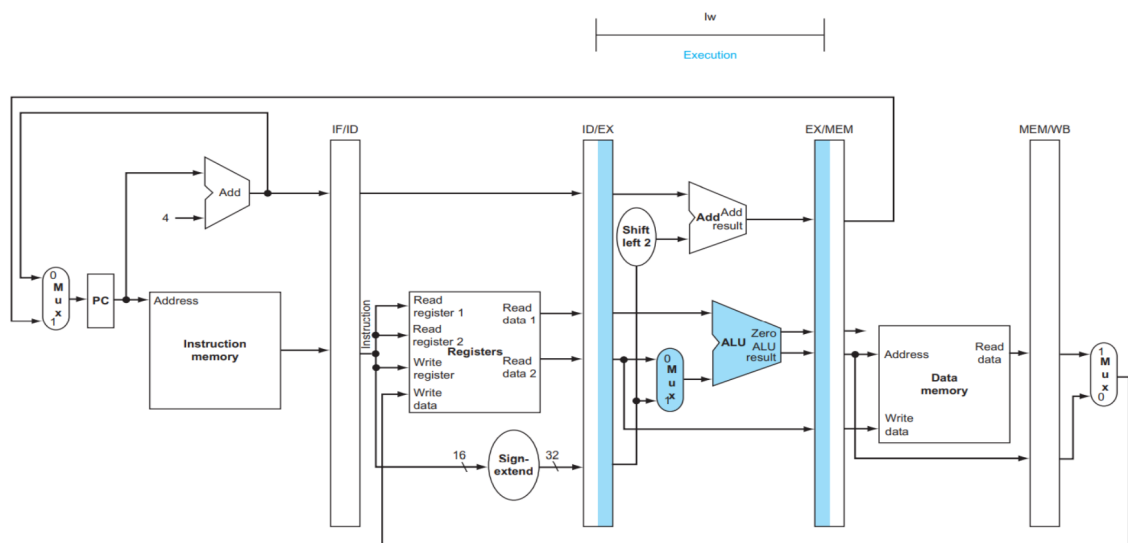
- The instruction is fetched from the instruction memory and stored in IF/ID register (명령어는 명령어 메모리에서 가져와 IF/ID 레지스터에 저장된다)
- PC is incremented by 4, which is written back into the PC for the next instruction (PC가 4씩 증가하여 다음 명령을 위해 다시 PC에 기록된다)
- The incremented PC is also saved in IF/ID register in case it is needed later for (beq) (증가된 PC는 나중에 (beq)에 필요할 경우를 대비하여 IF/ID 레지스터에도 저장된다)
 - The processor cannot know what instruction is being fetched; so, it must prepare for any case (프로세서는 어떤 명령어를 가져오는지 알 수 없으므로 모든 경우에 대비해야한다)

lw Instruction in Pipelined Datapath - ID (2)



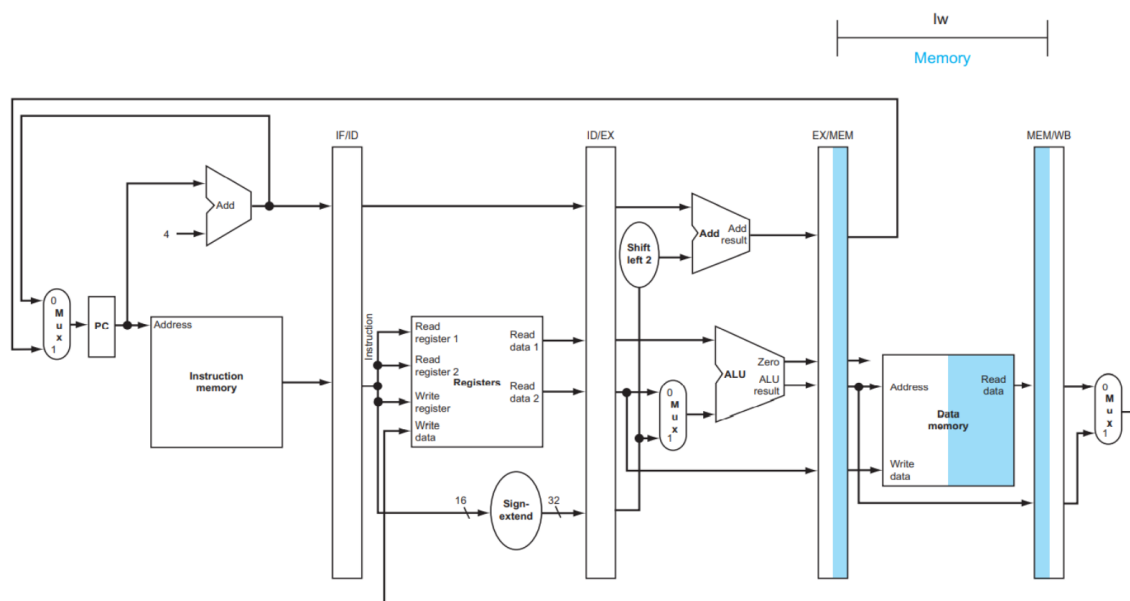
- IF/ID register supplies the instruction (now, the datapath knows that it is `lw`) (IF/ID 레지스터가 명령을 제공한다. (이제 데이터 경로는 `lw`임을 알고 있다))
- A set of data generated in ID stage are stored in ID/EX register (ID 단계에서 생성된 데이터 세트는 ID/EX 레지스터에 저장된다)
 - Two 32-bit data read from the two registers (두 레지스터에서 읽은 두 개의 32비트 데이터)
 - A sign-extended 32-bit immediate value (부호 확장된 32비트 immediate 값)
 - The incremented PC in IF/ID (IF/ID에서 증가된 PC)

lw Instruction in Pipelined Datapath - EX (3)



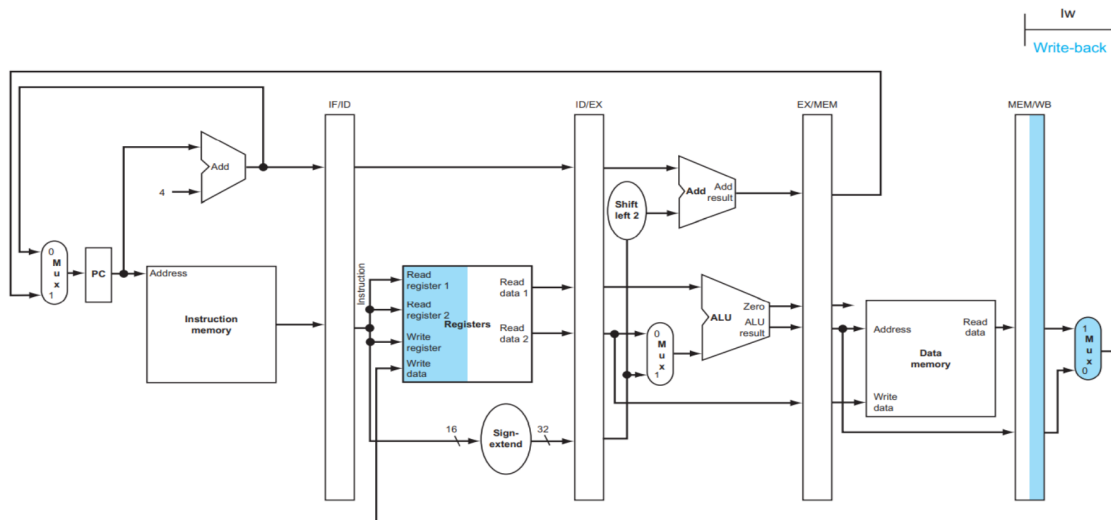
- (i) The value of register 1 and (ii) the sign-extended immediate value are read from ID/EX register ((i) 레지스터 1의 값과 (ii) 부호 확장된 immediate 값을 ID/EX 레지스터에서 읽는다)
- (i) and (ii) are added using the ALU to compute the target data memory address ((i) 및 (ii)는 ALU를 사용하여 대상 데이터 메모리 주소를 계산하기 위해 추가된다)
- The computed target address is stored in EX/MEM register (계산된 타겟 주소는 EX/MEM 레지스터에 저장된다)

lw Instruction in Pipelined Datapath - MEM (4)



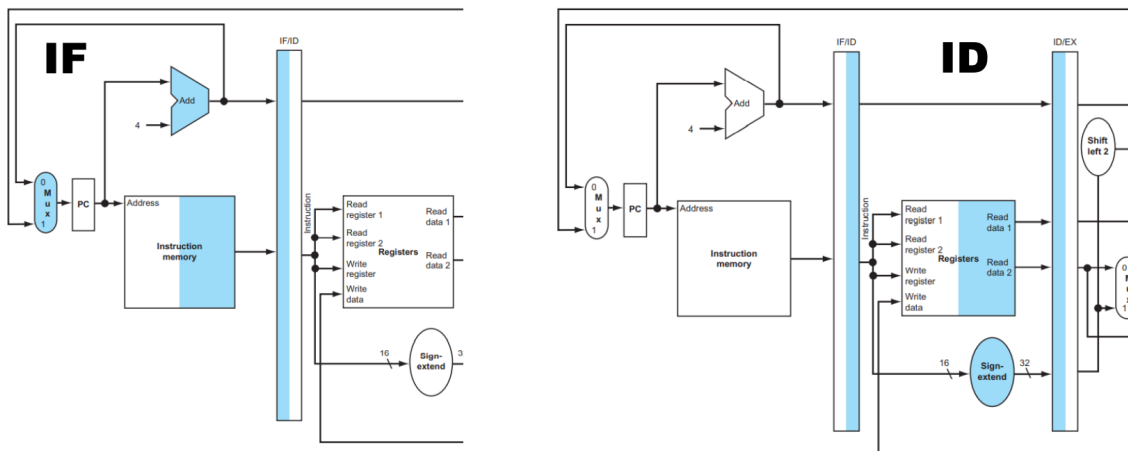
- Using the address from EX/MEM register, data is read from the data memory (EX/MEM 레지스터의 주소를 사용하여 데이터 메모리에서 데이터를 읽는다)
- The read data is stored in MEM/WB register (읽은 데이터는 MEM/WB 레지스터에 저장된다)

lw Instruction in Pipelined Datapath - WB (5)



- The data is read from MEM/WB register (MEM/WB 레지스터에서 데이터를 읽는다)
- The data is written into the register file (데이터가 레지스터 파일에 기록된다)

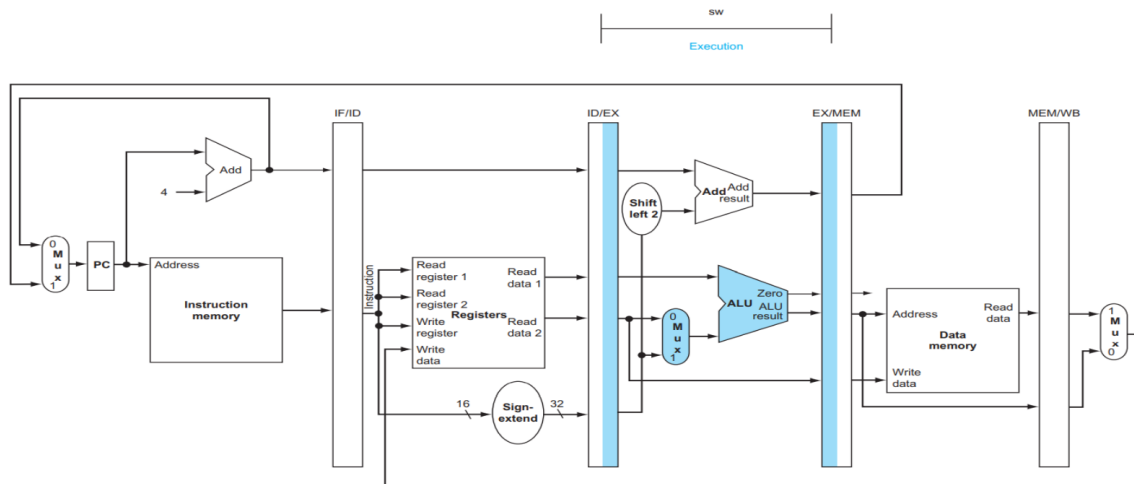
sw Instruction in Pipelined Datapath - IF/ID (1/2)



- The first two (IF/ID) stages are common in all the instructions (처음 두 단계 (IF/ID)는 모든 명령어에서 공통적으로 사용된다)
 - In ID stage, the datapath knows the fetched instruction (ID 단계에서 데이터 경로는 가져온 명령어를 알고 있다)
- IF stage: the fetched instruction & the incremented PC are stored in IF/ID register (IF 단계 : 가져온 명령어와 증가된 PC가 IF/ID 레지스터에 저장된다)

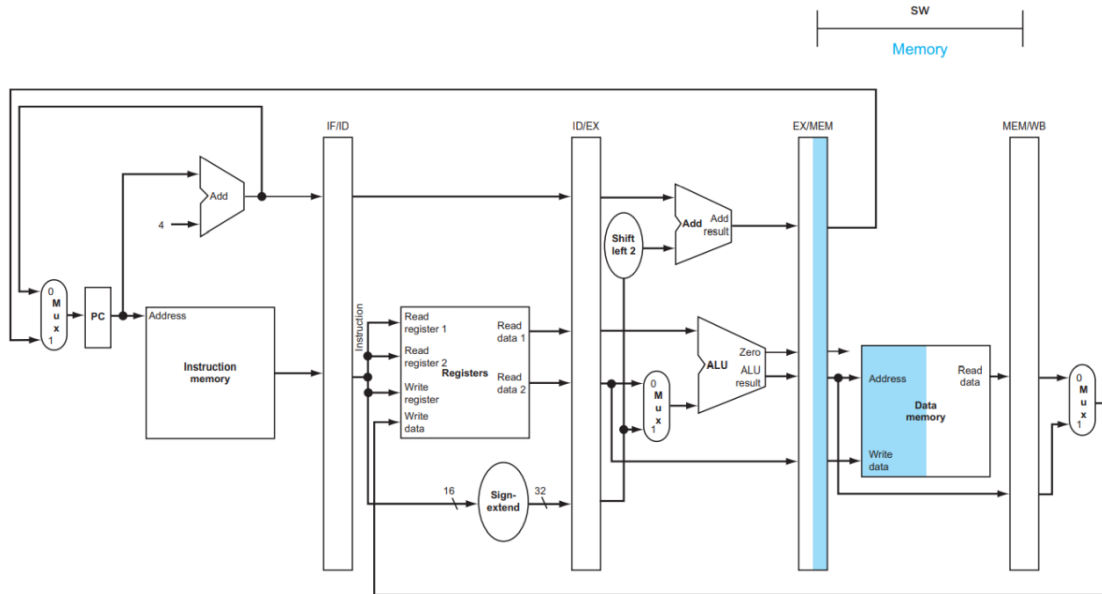
- ID stage: two register values & sign-extended value are stored in ID/EX register (ID 단계 : 두 레지스터 값과 부호 확장 값이 ID/EX 레지스터에 저장된다)

sw Instruction in Pipelined Datapath - EX (3)



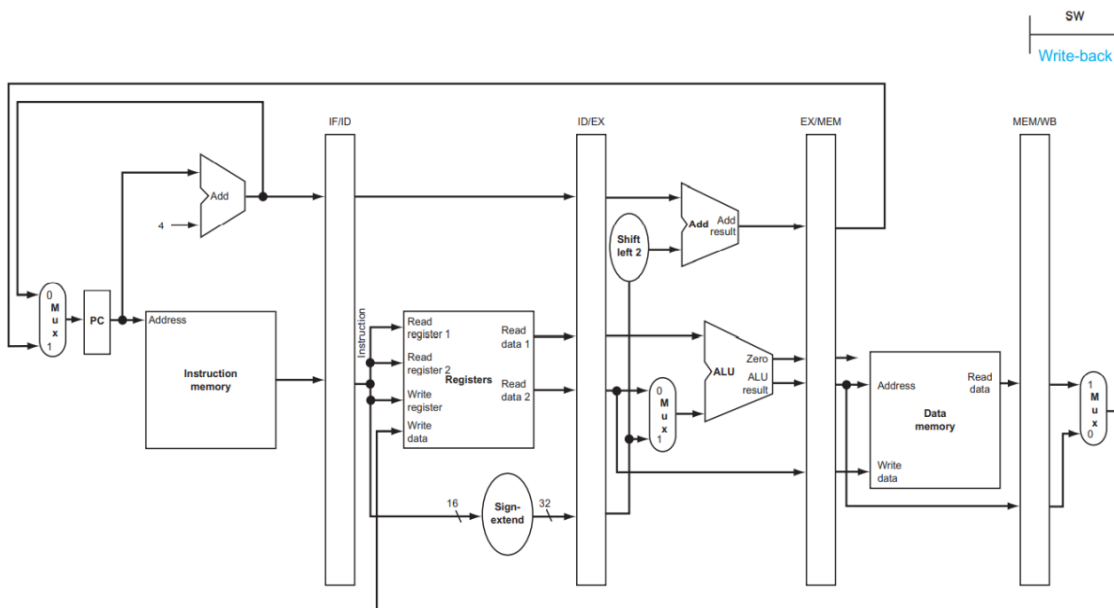
- (i) The register 1 value and (ii) the sign-extended value are read from ID/EX register, which are added using the ALU for the memory address ((i) 레지스터 1 값과 (ii) 부호 확장된 값이 ID/EX 레지스터에서 읽혀지고, 메모리 주소는 ALU를 사용하여 추가된다)
- The computed memory address is stored to EX/MEM register (계산된 메모리 주소는 EX/MEM 레지스터에 저장된다)
- The register 2 value, the data to be written into the data memory, is stored to EX/MEM (데이터 메모리에 기록할 데이터인 레지스터 2 값은 EX/MEM에 저장된다)

sw Instruction in Pipelined Datapath - MEM (4)



- The address and the data (register 2 value) are read from EX/MEM register (주소와 데이터(레지스터 2 값)는 EX/MEM 레지스터에서 읽는다)
- The data is written into the data memory at the address (데이터는 해당 주소의 데이터 메모리에 기록된다)

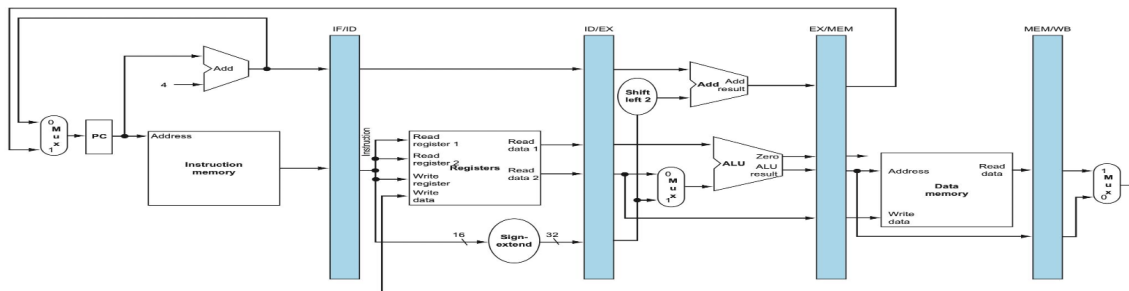
sw Instruction in Pipelined Datapath - WB (5)



- There is nothing left to complete the store instruction; so, nothing happens in WB stage (sw 명령을 완료하기 위해 남은 것이 없으므로 WB 단계에서는 아무 일도 일어나지 않는다)

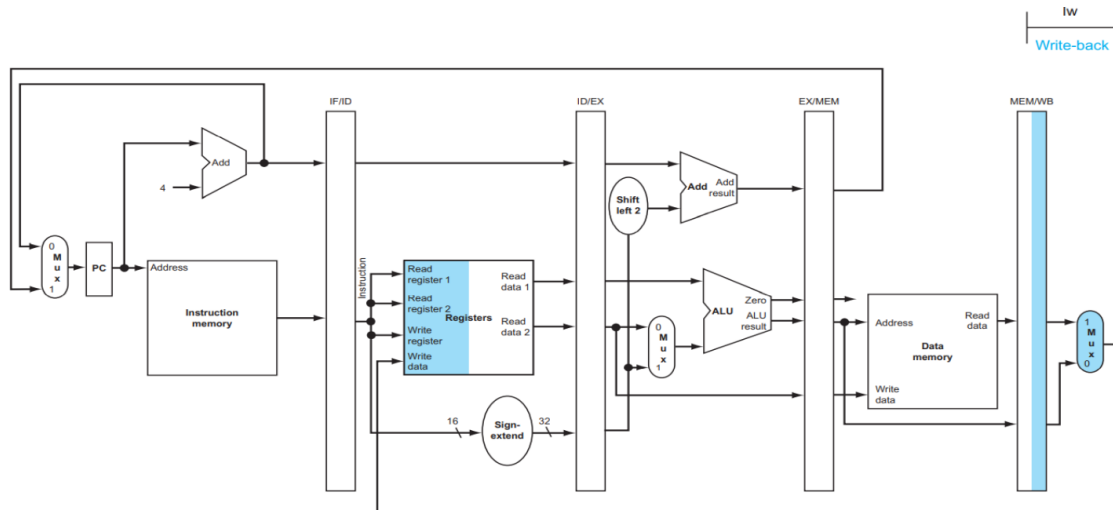
- In MEM stage of sw instruction, the data is written into the data memory (sw 명령의 MEM 단계에서는 데이터가 데이터 메모리에 기록된다)
- An instruction passes through a stage even if there is nothing to do in it, because all the following instructions are already in progress at the maximum rate (다음 모든 명령어가 이미 최대 속도로 진행 중이므로 수행할 작업이 없더라도 명령어는 해당 단계를 통과한다)

Two Key Points in Pipelined Datapath



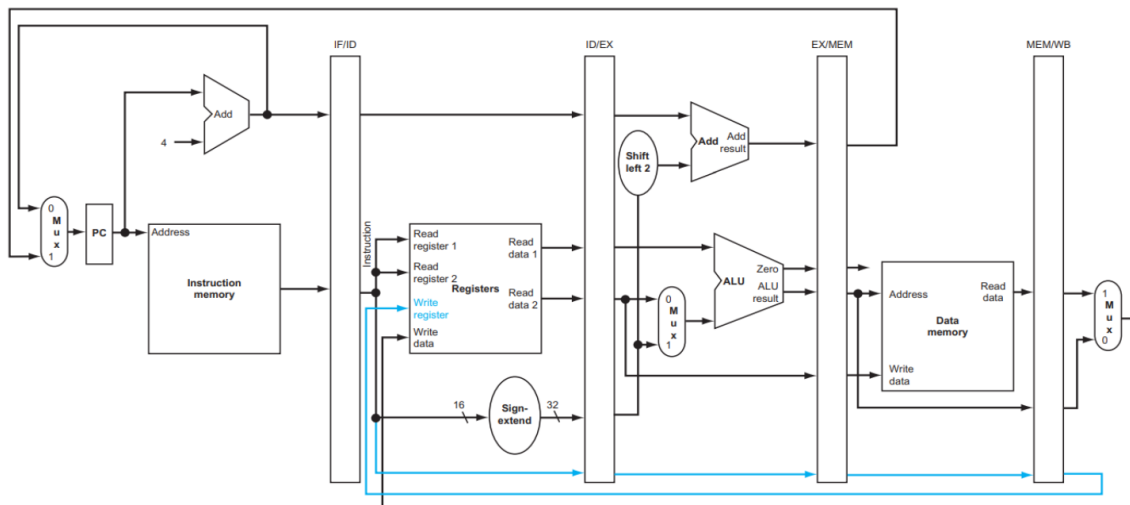
- (1) To pass something from a stage to another, it must be placed in a pipeline register ((1) 한 단계에서 다른 단계로 무언가를 전달하려면 pipeline 레지스터에 배치해야 한다)
 - Otherwise, the data or the information is lost when the next instruction enters that stage (그렇지 않으면, 다음 명령이 해당 단계에 들어갈 때 데이터 또는 정보가 손실된다)
 - For sw, the register 2 value obtained from ID is passed to EX (through ID/EX register) and again is passed to MEM (through EX/MEM register) (sw의 경우, ID에서 얻은 레지스터 2 값은 ID/EX 레지스터를 통해 EX로 전달되고, 다시 EX/MEM 레지스터를 통해 MEM으로 전달된다)
- (2) Each component - instruction memory, register read ports, ALU, data memory, register write port - can be used only within a single stage ((2) 명령어 메모리, 레지스터 읽기 포트, ALU, 데이터 메모리, 레지스터 쓰기 포트 등 각 구성 요소는 단일 스테이지 내에서만 사용할 수 있다)
 - Hardware cannot be shared by multiple instructions simultaneously (하드웨어를 여러 명령어가 동시에 공유할 수 없다)
 - Register file has separate ports for read/write; so, it can be considered as different components (레지스터 파일은 읽기/쓰기를 위한 별도의 포트가 있으므로 서로 다른 구성 요소로 간주할 수 있다)

A Corrected Pipelined Datapath: Bug



- In WB stage of lw instruction, the read data should be written into the write register (lw 명령의 WB 단계에서 읽기 데이터는 쓰기 레지스터에 기록되어야 한다)
- Then, at the moment, who supplies the write register number? (그렇다면 현재 쓰기 레지스터 번호는 누가 제공하는가?)
 - It is the instruction now in the ID stage, which is a later instruction of the load instruction (load 명령의 이후 명령인 ID 단계에 있는 명령이다)
- We need to preserve the write register number of the load instruction till it reaches WB stage (load 명령어의 쓰기 레지스터 번호는 WB 단계에 도달할 때까지 보존해야 한다)

A Corrected Pipelined Datapath: Fix



- To fix the bug, we need to pass the write register number from ID stage to WB stage (버그를 수정하려면 ID 단계에서 WB 단계로 쓰기 레지스터 번호를 전달해야 한다)
 - The write register number is stored in ID/EX register in the ID stage (쓰기 레지스터 번호는 ID 단계의 ID/EX 레지스터에 저장된다)
 - It is copied to EX/MEM register in the EX stage (EX 단계에서 EX/MEM 레지스터에 복사된다)
 - It is copied to MEM/WB register in the MEM stage (MEM 단계에서 MEM/WB 레지스터에 복사된다)
- Then, it can be read from MEM/WB register and used in WB stage (이후 MEM/WB 레지스터에서 읽어와 WB 단계에서 사용한다)