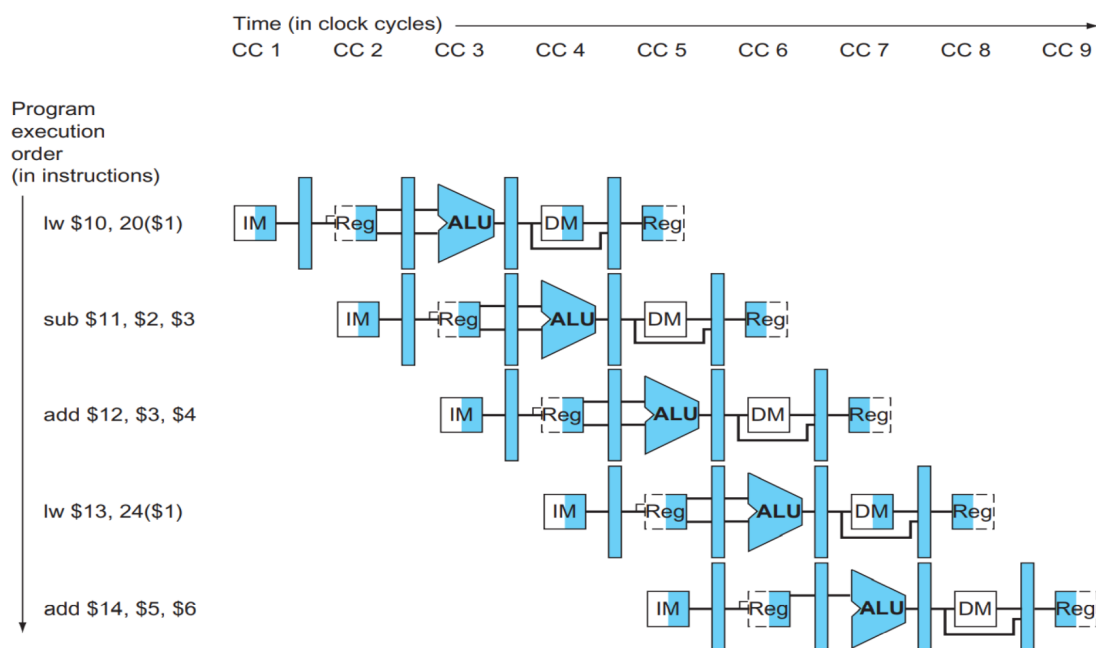


# Computer Architecture (ENE1004)

## ▼ Lec 14

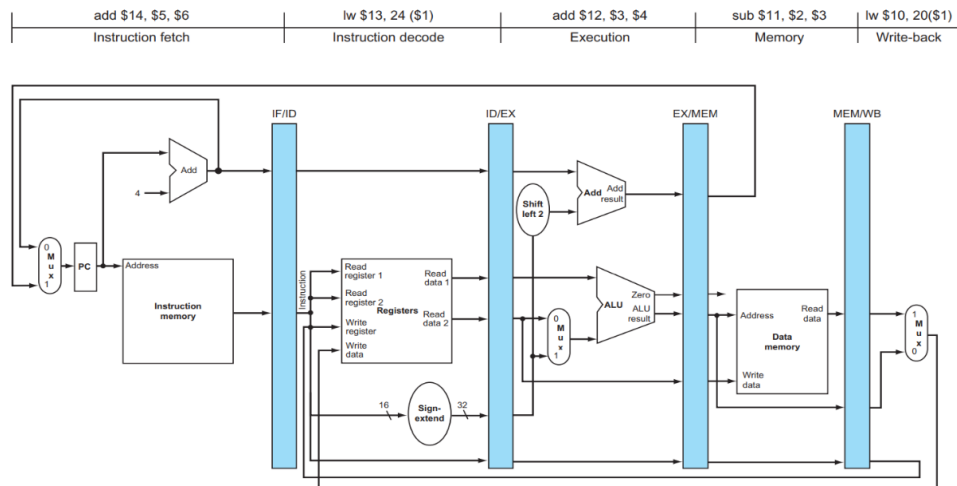
### Lec 14: The Processor 7

#### Example : Pipeline Diagram of Five Instructions



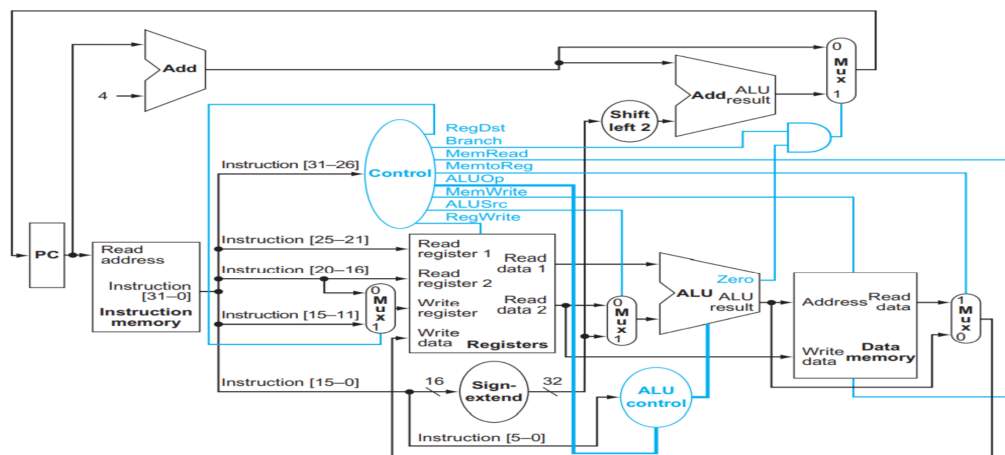
- Time advance from left to right; instructions advance in each clock cycle (CC) (왼쪽에서 오른쪽으로 시간 진행; 각 클럭 사이클(CC)에서 명령어 진행)
- Five instructions enter to the pipeline one after another (from top to bottom) (5개의 명령어가 차례로 파이프라인에 들어간다 (위에서 아래로))
- In each cycle, all instructions are in the different stages (use different components) (각 사이클에서 모든 명령어는 다른 단계에 있다 (다른 구성 요소 사용))
  - Register file can be read and written in the same cycle, thanks to its separate ports (별도의 포트 덕분에 동일한 사이클에서 레지스터 파일을 읽고 쓸 수 있다)

## Example : Five Instructions in CC5



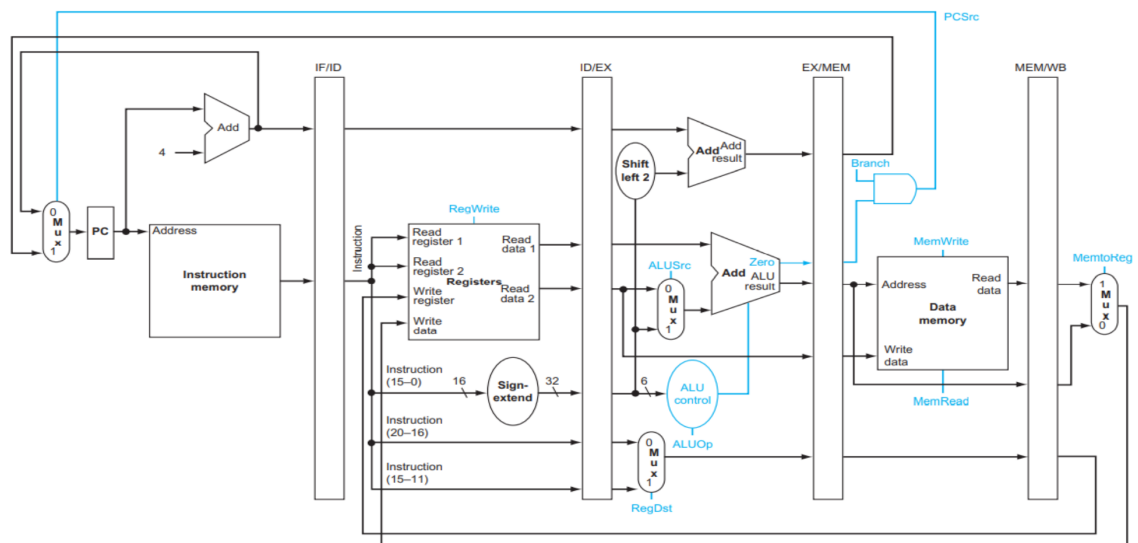
- There exists a (pipelined) datapath, which can process five instructions at a time (한 번에 5개의 명령어를 처리할 수 있는 (파이프라인) 데이터 경로가 존재한다.)
- In a given time (CC), all instructions are in the different stages (use different HWs) (주어진 시간(CC)에 모든 명령어는 서로 다른 단계에 있다 (서로 다른 HW 사용).)

## Control Signals for Non-Pipelined Datapath (Review)



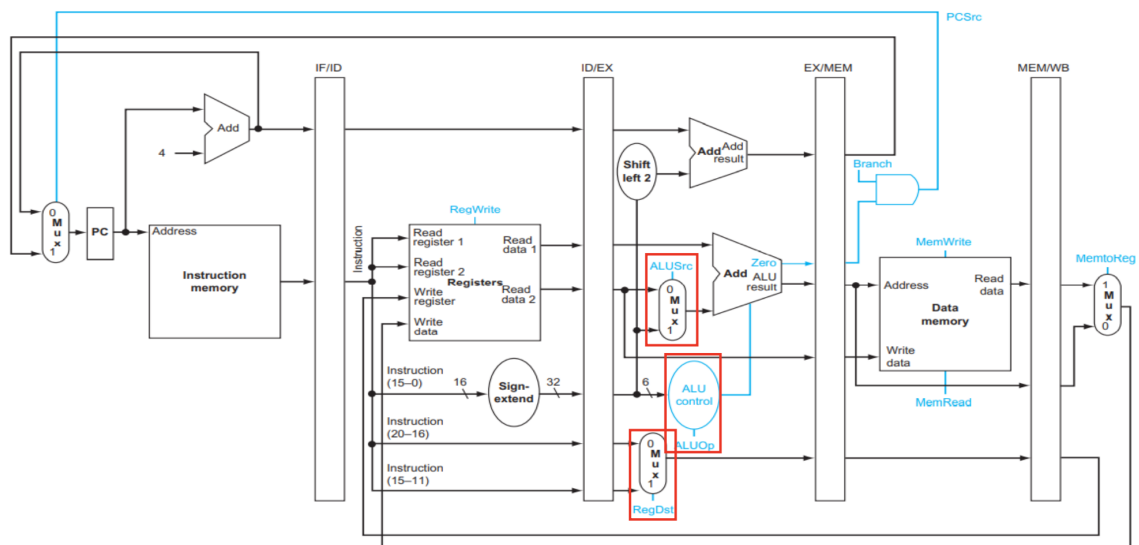
Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

## Control Signals for Pipelined Datapath (1)



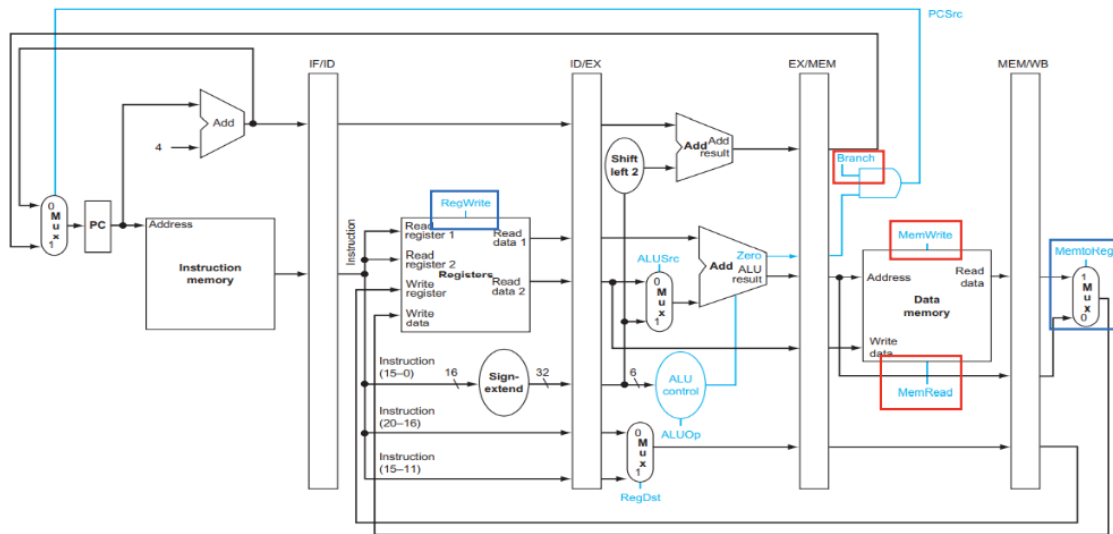
- We divide the control signals into five groups according to the stages (제어 신호를 단계에 따라 5개의 그룹으로 나눈다.)
  - Each control line is associated with an active component in a single stage (각 제어 라인은 단일 스테이지의 활성 구성 요소와 연결된다.)
- IF stage: There is nothing special to control (as is in non-pipelined datapath) (IF 단계: 제어할 특별한 것이 없다. (비파이프라인 데이터 경로에서와 마찬가지로이다.))
- ID stage: There is nothing special to control (as is in non-pipelined datapath) (ID 단계 : 특별히 제어할 것이 없다. (비파이브라인 데이터 경로에서와 마찬가지로이다.))
  - Note that we do not need a control signal for reading registers; RegWrite is used in [WB] stage (레지스터 읽기를 위한 제어 신호는 필요하지 않으며, [WB] 단계에서는 RegWrite가 사용된다.)

## Control Signals for Pipelined Datapath (2)



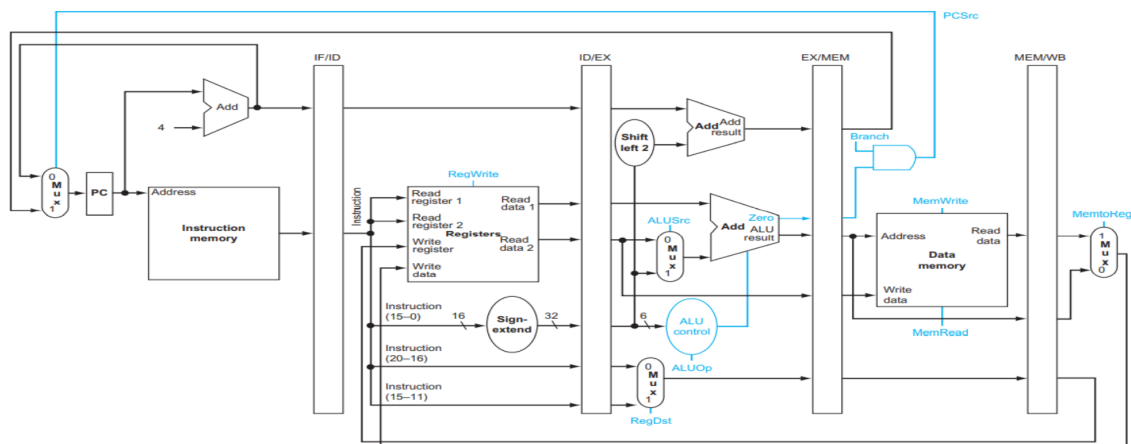
- EX stage: ALUSrc, ALUOp, and RegDst should be set (EX 단계 : ALUSrc, ALUOp 및 RegDst가 설정되어야 한다.)
  - ALUSrc selects either Read data 2 or a sign-extended immediate for the 2nd input of the ALU (ALUSrc는 ALU의 두 번째 입력에 대해 데이터 2 읽기 또는 sign-extended immediate를 선택한다.)
  - ALUOp selects the ALU operation in collaboration with 6-bit func field (the rightmost 6 bits) (ALUOp는 6비트 함수 필드(가장 오른쪽 6비트)와 함께 ALU 연산을 선택한다.)
  - RegDst selects the Write register (1 for R-type instructions; 0 for load instructions) (RegDst는 쓰기 레지스터를 선택한다. (R형 명령어의 경우 1, 로드 명령어의 경우 0))

### Control Signals for Pipelined Datapath (3)



- MEM stage: Branch, MemRead, and MemWrite should be set (MEM 단계: Branch, MemRead, MemWrite가 설정되어야 한다.)
  - Branch, MemRead, and MemWrite are set for branch equal, load, and store instructions (branch equal, load, and store 명령에 대해 Branch, MemRead, MemWrite가 설정되어 있다.)
- WB stage: MemtoReg and RegWrite should be set (WB 단계: MemtoReg 및 RegWrite가 설정되어야 한다.)
  - MemtoReg decides between sending the ALU result or the memory value to the register file (MemtoReg는 ALU 결과 또는 메모리 값을 레지스터 파일로 전송할지 여부를 결정한다.)
  - RegWrite writes the chosen value into the register (RegWrite는 선택한 값을 레지스터에 쓴다.)

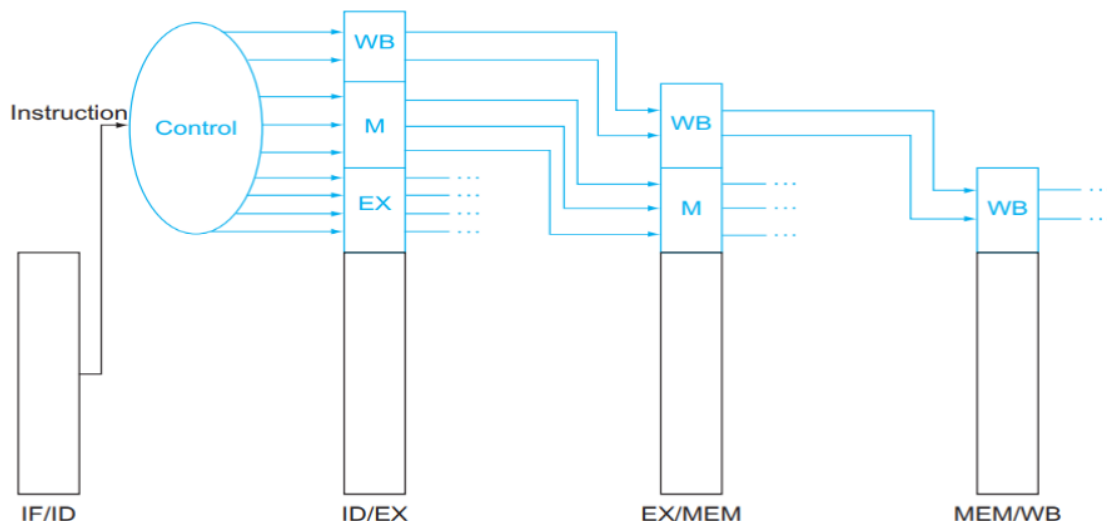
## Control Signals for Pipelined Datapath (4)



- The meaning of the control signals remains unchanged; we can use the same values in the table (제어 신호의 의미는 변경되지 않으므로 표에서 동일한 값을 사용할 수 있다.)
- But, the table is rearranged by grouping the signals by the pipeline stages (그러나 pipeline 단계별로 신호를 그룹화하여 테이블을 다시 정렬한다.)

Instruction	Execution/address calculation stage control lines				Memory access stage control lines			Write-back stage control lines	
	RegDst	ALUOp1	ALUOp0	ALUSrc	Branch	Mem-Read	Mem-Write	Reg-Write	Memto-Reg
R-format	1	1	0	0	0	0	0	1	0
lw	0	0	0	1	0	1	0	1	1
sw	X	0	0	1	0	0	1	0	X
beq	X	0	1	0	1	0	0	0	X

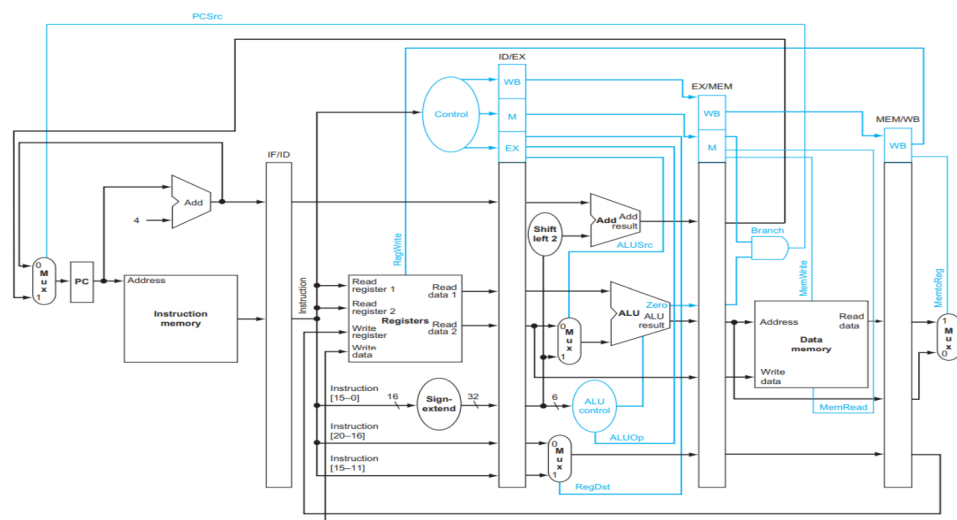
## Implementing Control Lines for Pipelined Datapath (1)



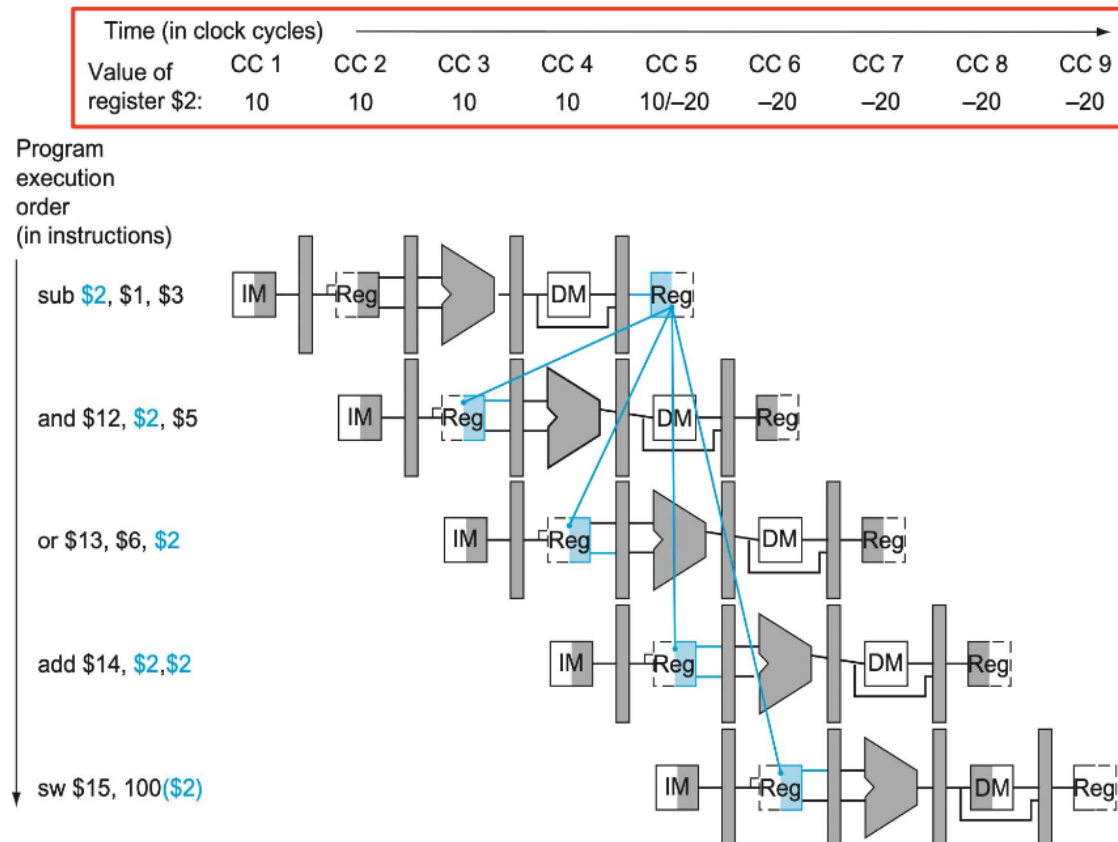
- For each instruction, the nine control signals should be set in each stage (각 명령어에 대해 각 단계에서 9개의 제어 신호를 설정해야 한다.)
  - (i) We can extend the pipeline registers to include control information (제어 정보를 포함하도록 파이프라인 레지스터를 확장할 수 있다.)
  - (ii) As an instruction advances, its control values can move from one pipeline register to the next (명령어가 진행됨에 따라 제어 값이 한 파이프라인 레지스터에서 다음 파이프라인 레지스터로 이동할 수 있다.)
- Values for an instruction can be obtained in ID stage and advance through the stages (명령어에 대한 값은 ID 스테이지에서 가져와서 스테이지를 통해 진행할 수 있다.)

- RegDst, ALUOp1/0, ALUSrc are used in EX stage; they are not copied to EX/MEM register (RegDst, ALUOp1/0, ALUSrc는 EX 단계에서 사용되며 EX/MEM 레지스터로 복사되지 않는다.)
- Branch, MemRead, MemWrite are used in MEM stage; they are not copied to MEM/WB register (Branch, MemRead, MemWrite는 MEM 단계에서 사용되며, MEM/WB 레지스터에 복사되지 않는다.)
- MemtoReg and RegWrite are used in WB stage (MemtoReg, RegWrite는 WB 단계에서 사용된다.)

## Implementing Control Lines for Pipelined Datapath (2)



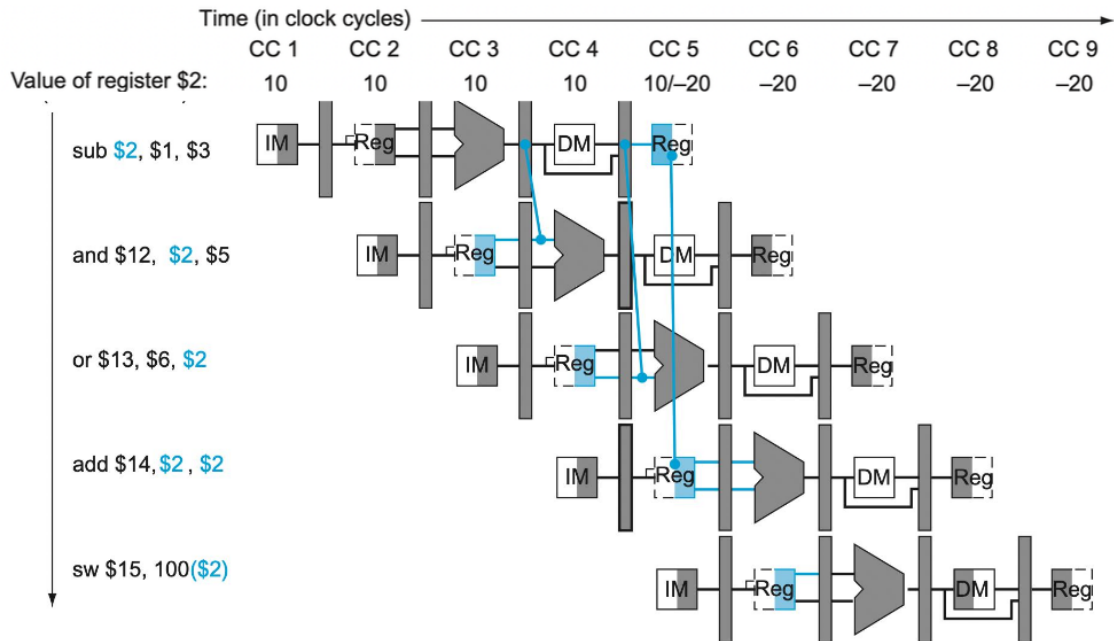
## Data Hazard in RegWrite Scenarios



- and \$12, \$2, \$5
  - The 2nd instruction expects to read the new value "-20" from \$2 in ID stage at CC3 (두 번째 명령은 CC3의 ID 단계에서 \$2에서 새 값 "-20"을 읽을 것으로 예상한다.)
  - However, \$2 holds the old value "10" at CC3 (그러나 \$2는 CC3에서 이전 값인 "10"을 보유한다.)
- or \$13, \$6, \$2
  - The 3rd instruction expects to read the new value "-20" from \$2 in ID stage at CC4 (세 번째 명령은 CC4의 ID 단계에서 \$2에서 새 값 "-20"을 읽을 것으로 예상한다.)
- We call this problem "data hazard" (우리는 이 문제를 "데이터 위험"이라고 부른다.)
  - A destination register of an instruction is read by following instructions as source registers (명령의 대상 레지스터는 소스 레지스터와 같이 다음 명령에 의해 읽혀진다.)

## Data Forwarding (Solution) in RegWrite Scenarios





- The new value "-20" is generated by sub at the end of EX stage at CC3 (EX/MEM register) (새로운 값 "-20"은 CC3(EX/MEM 레지스터)의 EX 단계 끝에서 sub에 의해 생성된다.)
- How about sending (forwarding) the new value of the sources of ALU? (이 새로운 값을 ALU의 소스로 전송(포워딩)하는 것은 어떨까?)
  - For and, we can send the value from EX/MEM register to Rs port of the ALU at CC4 (and의 경우, EX/MEM 레지스터의 값을 CC4에서 ALU의 Rs 포트에 전송할 수 있다.)
  - For or, we can send the value from MEM/WB register to Rt port of the ALU at CC5 (or의 경우, CC5에서 MEM/WB 레지스터의 값을 ALU의 Rt 포트에 전송할 수 있다.)