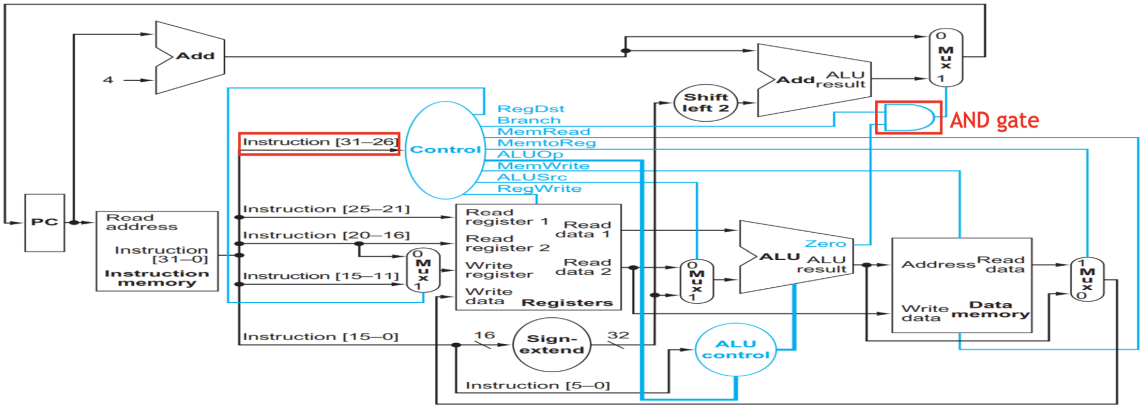
Computer Architecture (ENE1004)

▼ Lec 11

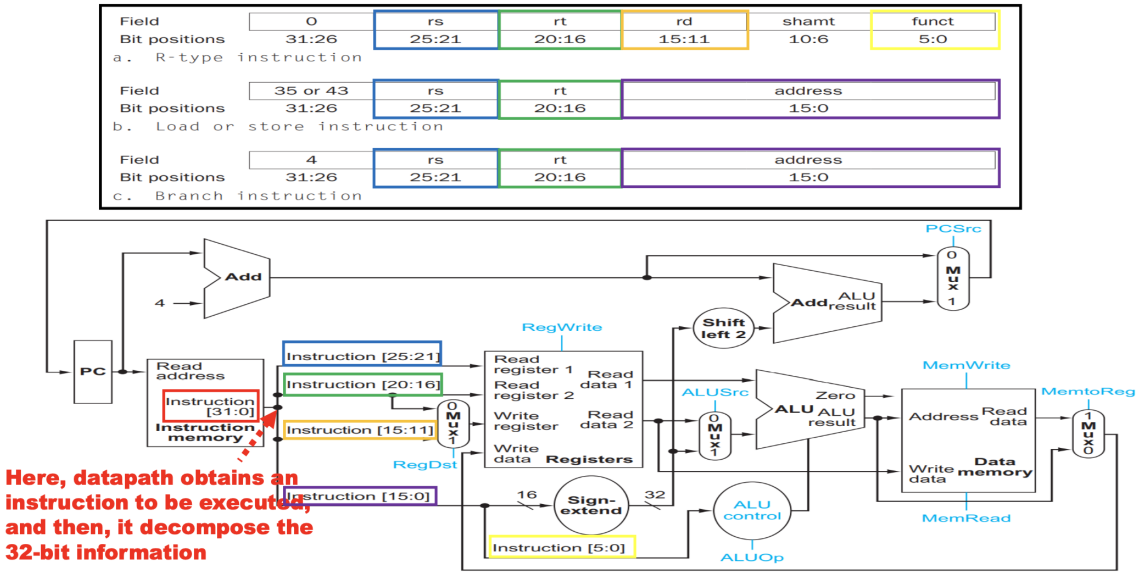
Lec 11: The Processor 4

A Control Unit for Datapath



- The control unit determines all the signal values based on the instruction type (control unit은 명령어 유형에 따라 모든 신호 값을 결정한다.)
  - Input: Instruction [31-26] & output: corresponding values of seven signals (입력: 명령어 [31-26] 및 출력: 7가지 신호의 해당 값)
- PCSrc is replaced by an AND gate of which two inputs are “Branch” and “Zero” (PCSrc는 두 개의 입력이 "Branch"와 "Zero"인 AND 게이트로 대체된다.)
  - If the instruction is a branch (Branch = 1) and the branch is taken (Zero = 1), then the target address is written to the pc (명령어가 브랜치(Branch = 1)이고 브랜치가 취해지면(Zero = 1), 대상 주소가 PC에 기록된다.)

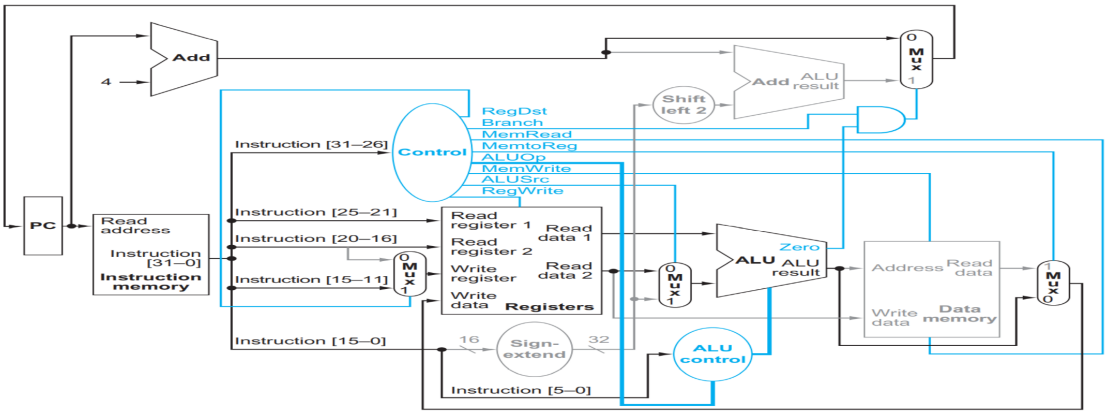
Revisit to Instruction Formats



Here, datapath obtains an instruction to be executed, and then, it decompose the 32-bit information (여기서 데이터경로는 실행할 명령을 얻은 다음 32비트 정보를 분해한다.)

Control Signals for R-Type Instructions

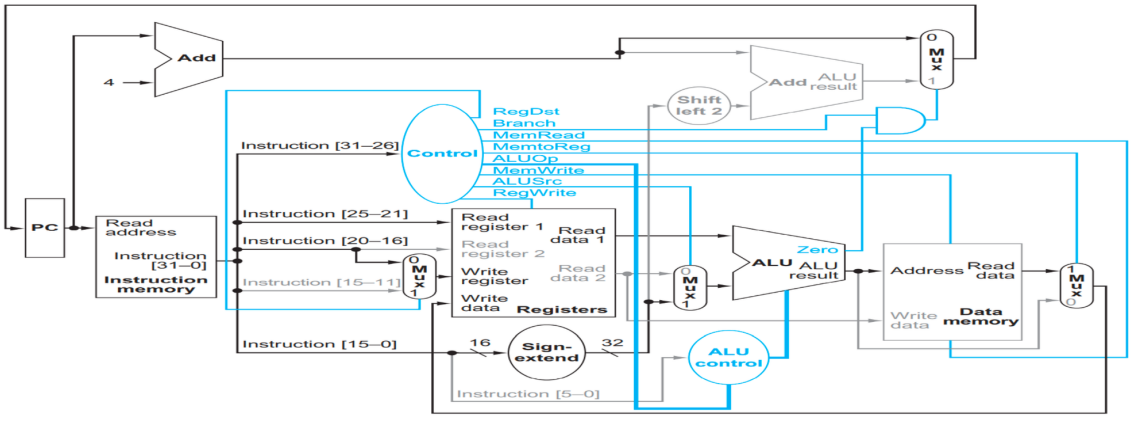
Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1



- ALUOp1 = 1, ALUOp0 = 0가 뭔가 ?
  - R-type 명령어에서 ALUOp1 = 1, ALUOp0 = 0인 경우, 이는 ALU에게 R-type 명령어의 연산을 수행하라는 것을 의미한다. R-type 명령어는 레지스터 간의 연산을 수행하는 명령어들로, 예를 들어 add, sub, and, or 등이 있다. 여기서, ALUOp 신호들은 ALU에게 함수 코드(Field)를 사용하여 정확한 연산을 결정하도록 지시한다.

Control Signals for Load Instructions

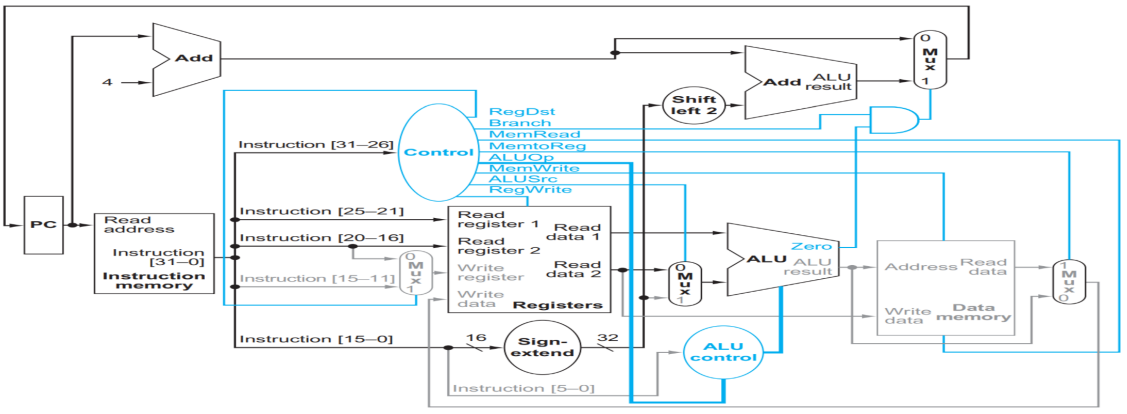
Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1



- ALUOp1 = 0, ALUOp0 = 0가 뭔가 ?
  - lw, sw 명령어에서 ALUOp1 = 0, ALUOp0 = 0인 경우, 이는 메모리 주소를 계산하는 간단한 덧셈 연산을 수행하라는 것을 의미한다. 이러한 명령어들은 메모리에서 데이터를 로드하거나 메모리에 데이터를 저장할 때, 주소를 계산하기 위해 기본적인 덧셈 연산이 필요하다. 따라서, ALU는 주소 계산을 위한 덧셈 연산을 수행한다.

Control Signals for Store Instructions

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

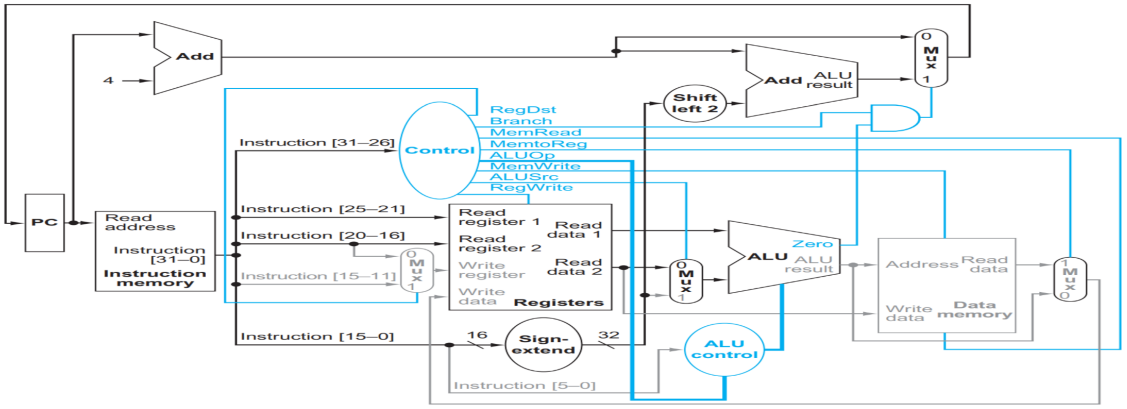


- ALUOp1 = 0, ALUOp0 = 0가 뭔가 ?

- lw, sw 명령어에서 ALUOp1 = 0, ALUOp0 = 0인 경우, 이는 메모리 주소를 계산하는 간단한 덧셈 연산을 수행하라는 것을 의미한다. 이러한 명령어들은 메모리에서 데이터를 로드하거나 메모리에 데이터를 저장할 때, 주소를 계산하기 위해 기본적인 덧셈 연산이 필요하다. 따라서, ALU는 주소 계산을 위한 덧셈 연산을 수행한다.

Control Signals for Branch-on-Equal Instructions

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



- ALUOp1 = 0, ALUOp0 = 1가 뭔가 ?
  - beq 명령어에서 ALUOp1 = 0, ALUOp0 = 1인 경우, 이는 분기 연산을 수행하라는 것을 의미한다. beq는 두 레지스터의 값이 같은지 비교하고, 같다면 지정된 위치로 분기(점프)하는 명령어이다. 이 경우, ALU는 두 값이 같은지 비교하는 연산을 수행한다.