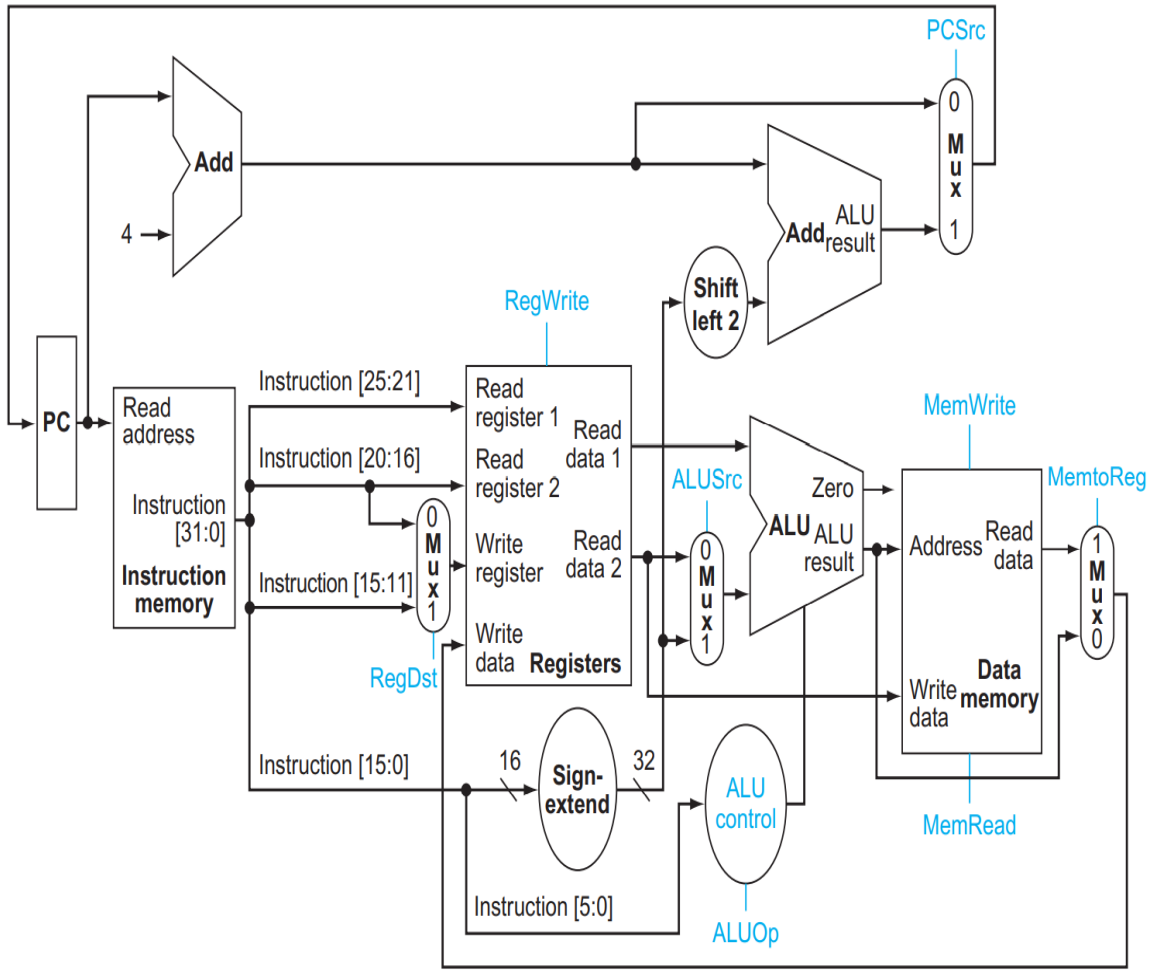
Computer Architecture
(ENE1004)

Lec – 13: The Processor (Chapter 4) - 4

Upcoming Schedule

- Apr. 20 (Thursday) – No class
  - The class right before the midterm exam
- Apr. 21/22 (Friday/Saturday) – Sample questions are uploaded to LMS
  - Keep your eyes on the notice
- Apr. 24 (Monday) - Midterm exam
  - In our classroom during our class hours
  - At 1:00pm for class # 24434
  - At 2:30pm for class # 23978

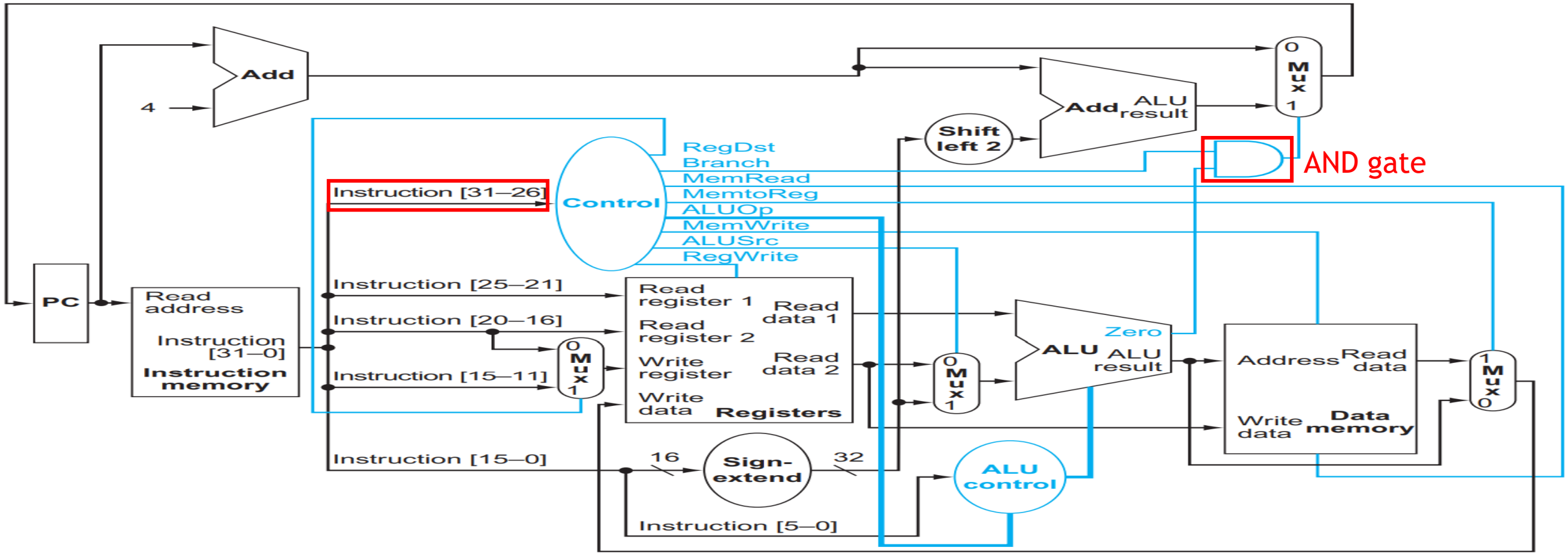
Seven Control Signals + 2-bit ALUOp



Signal name	A value of 0 Effect when deasserted	A value of 1 Effect when asserted
RegDst	The register destination number for the Write register comes from the rt field (bits 20:16).	The register destination number for the Write register comes from the rd field (bits 15:11).
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, lower 16 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

- The values of the seven signals + ALUOp signal are determined by the instruction type
- Depending on the instruction type, a control unit that determines all the values is required

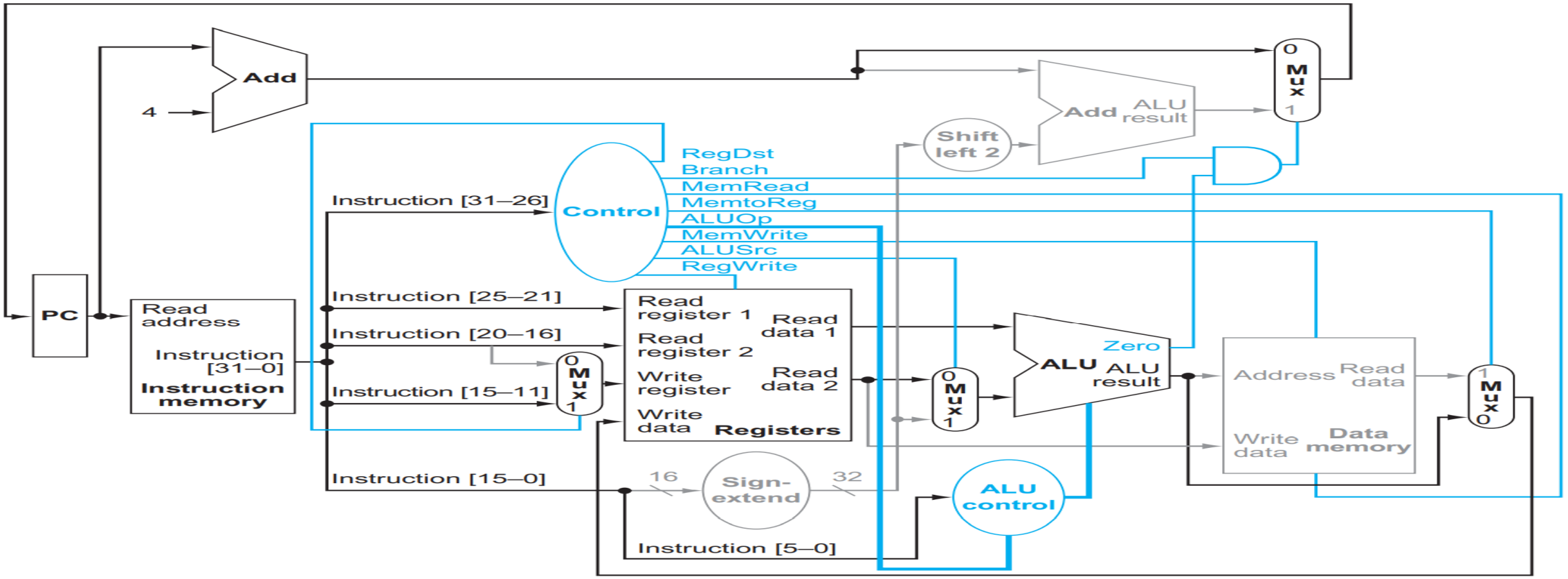
Control Unit for Datapath



- The control unit determines all the signal values based on the instruction type
  - Input: Instruction [31-26] & output: corresponding values of seven signals
- PCSrc is replaced by an AND gate of which two inputs are “Branch” and “Zero”
  - If the instruction is a branch (Branch = 1) and the branch is taken (Zero = 1), then the target address is written to the PC

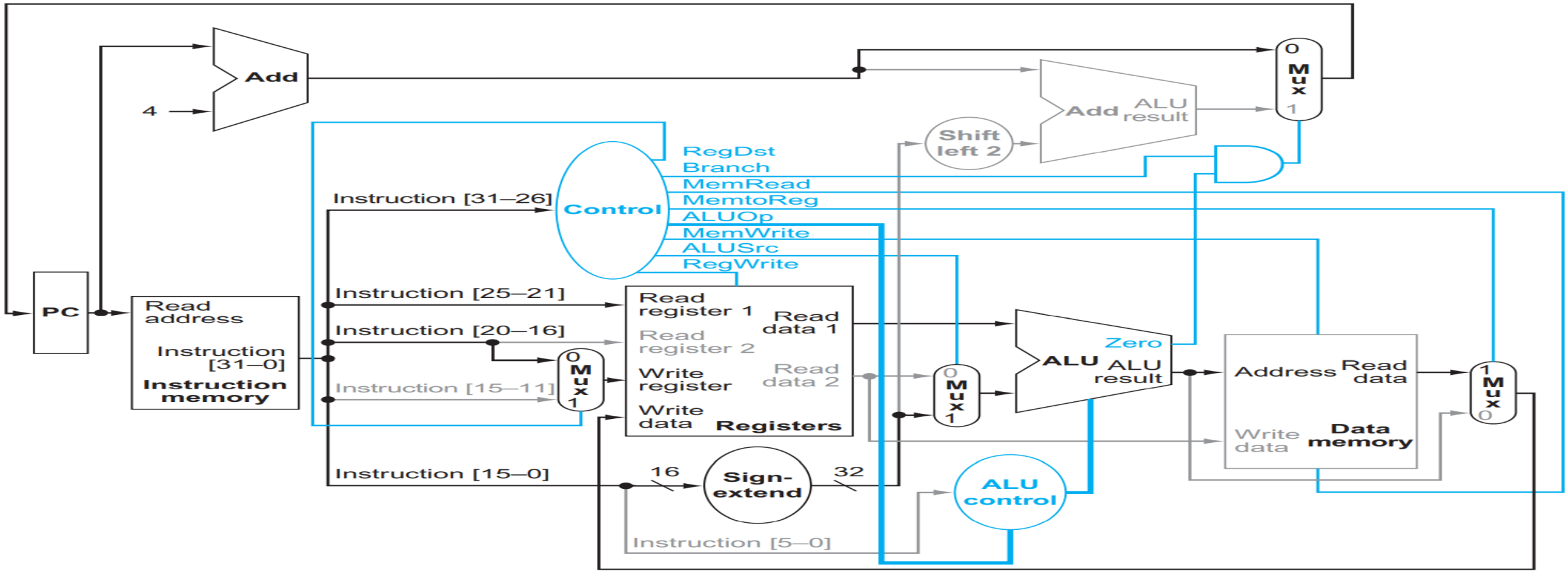
Control Unit for R-Type Instructions

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



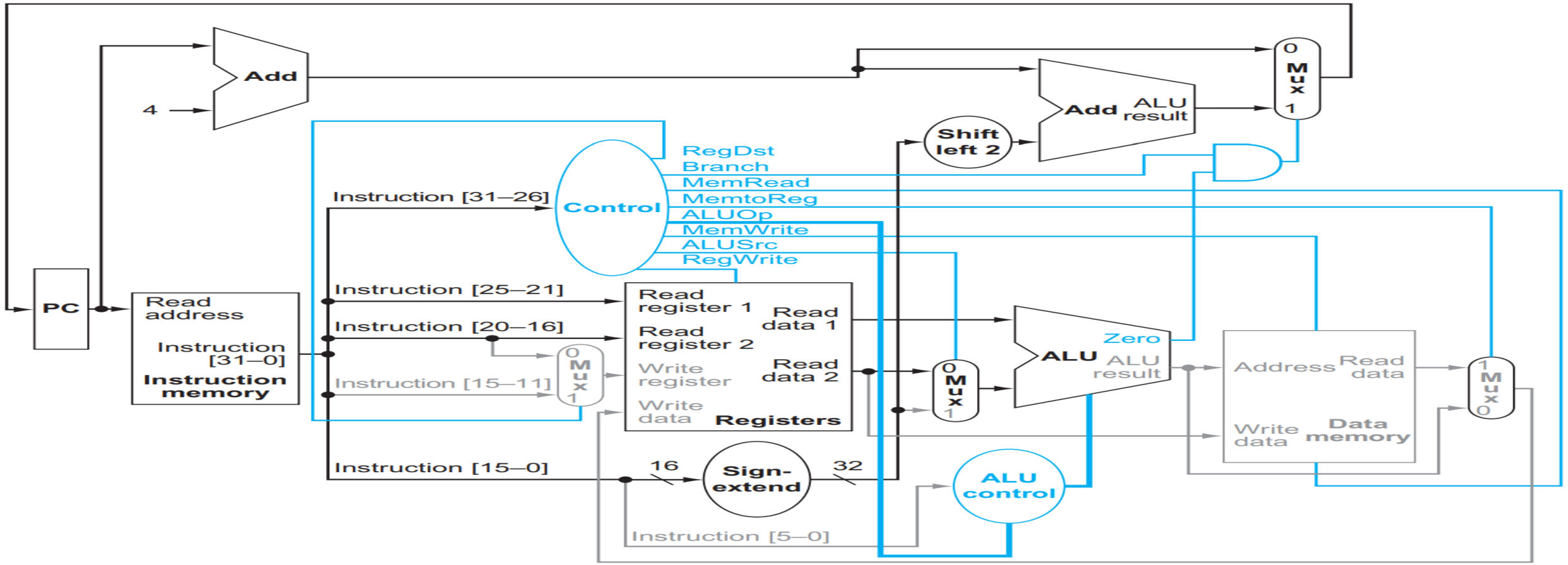
Control Unit for Load Instruction

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



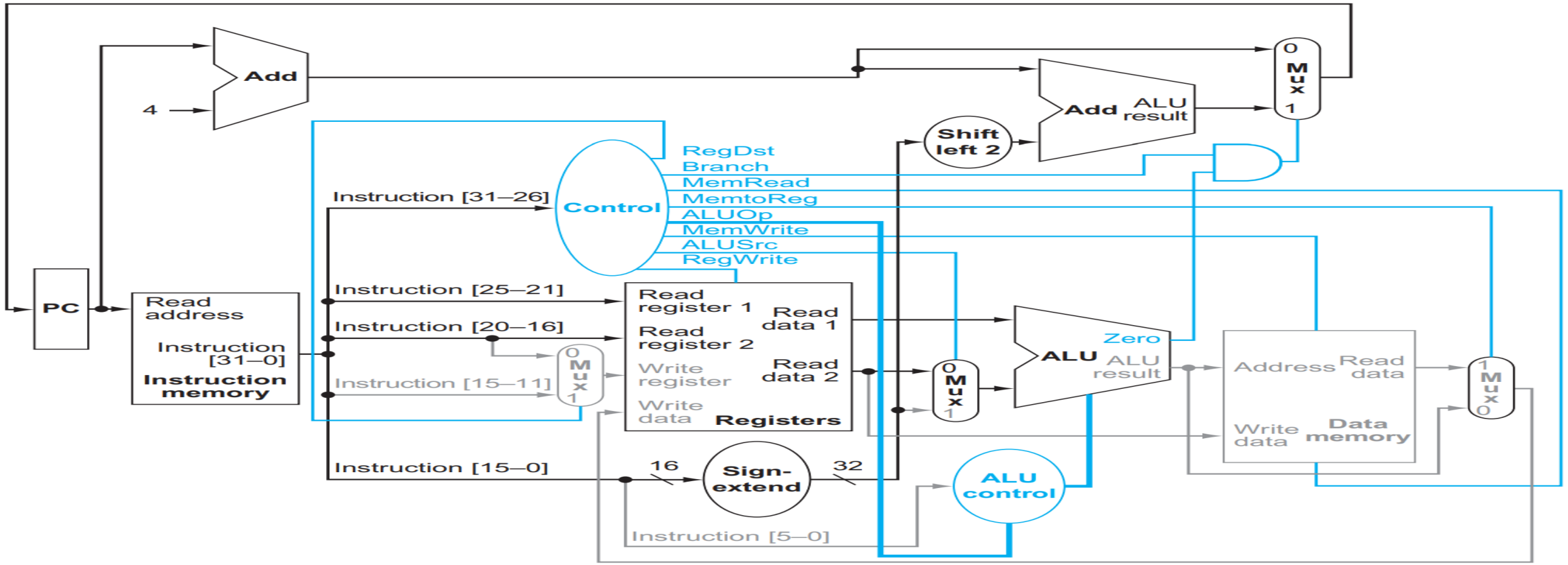
Control Unit for Store Instruction

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



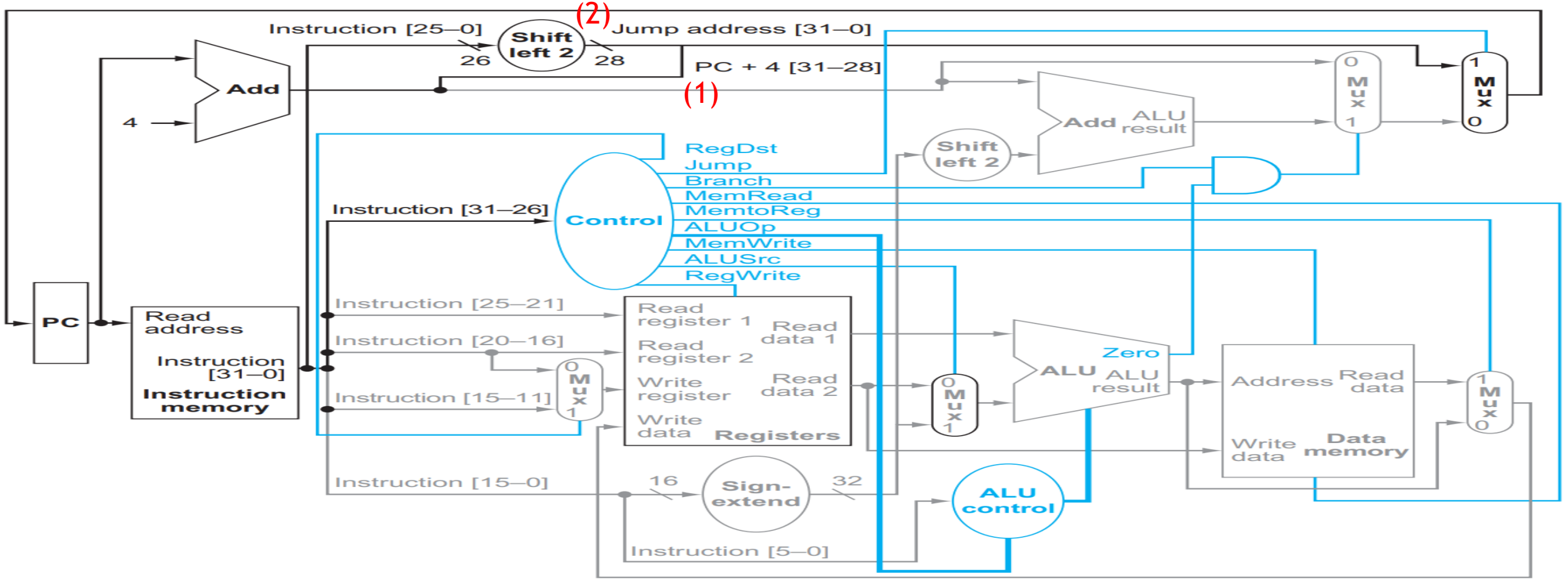
Control Unit for Branch-on-Equal Instruction

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



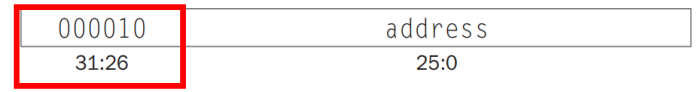
Implementing Jump Instruction

- A jump instruction computes its target address using pseudo-direct addressing
  - (1) The leftmost 4 bits of the next instruction ([31:28] of PC + 4) is extracted
  - (2) The offset 26 bits of the instruction is extended by two bits (multiplied by 4) ([25:0] → [27:0])
  - Jump target address [31:0] = (1) [31:28] ⊕ (2) [27:0]



Implementing Jump Instruction

- A multiplexer (Mux) is added
  - A PC value can be (i) PC+4, (ii) branch target, or (iii) jump target
  - The additional Mux has two inputs for (iii) and (i)/(ii)
  - "Jump" signal is added to select (iii) or (i)/(ii)



If Instruction [31:26] = 00010, Jump = 1