EE204: Computer Architecture

Chapter 4

Section 4.4

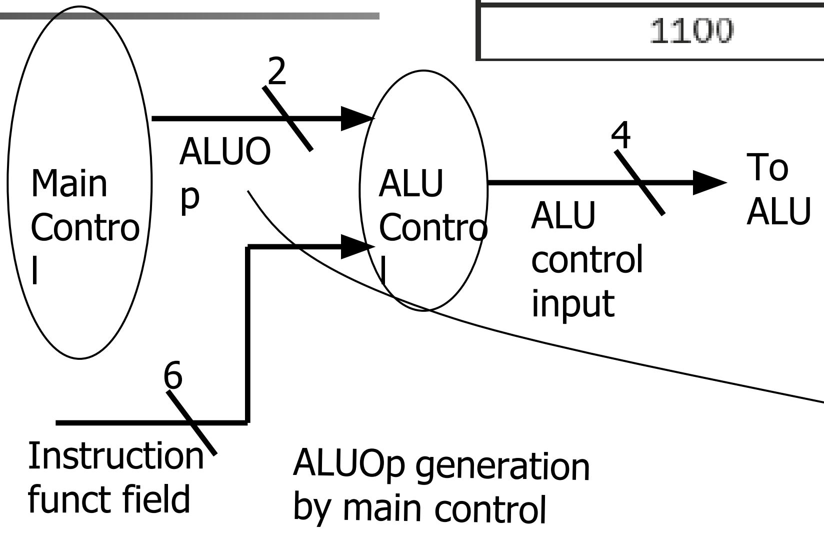
Control

- Control unit takes input from
  - the instruction opcode bits

- Control unit generates
  - ALU control input
  - write enable (possibly, read enable also) signals for each storage element
  - selector controls for each multiplexor

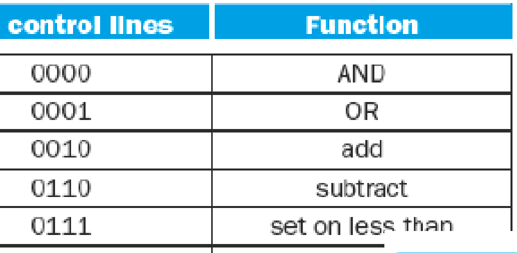
ALU Control

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

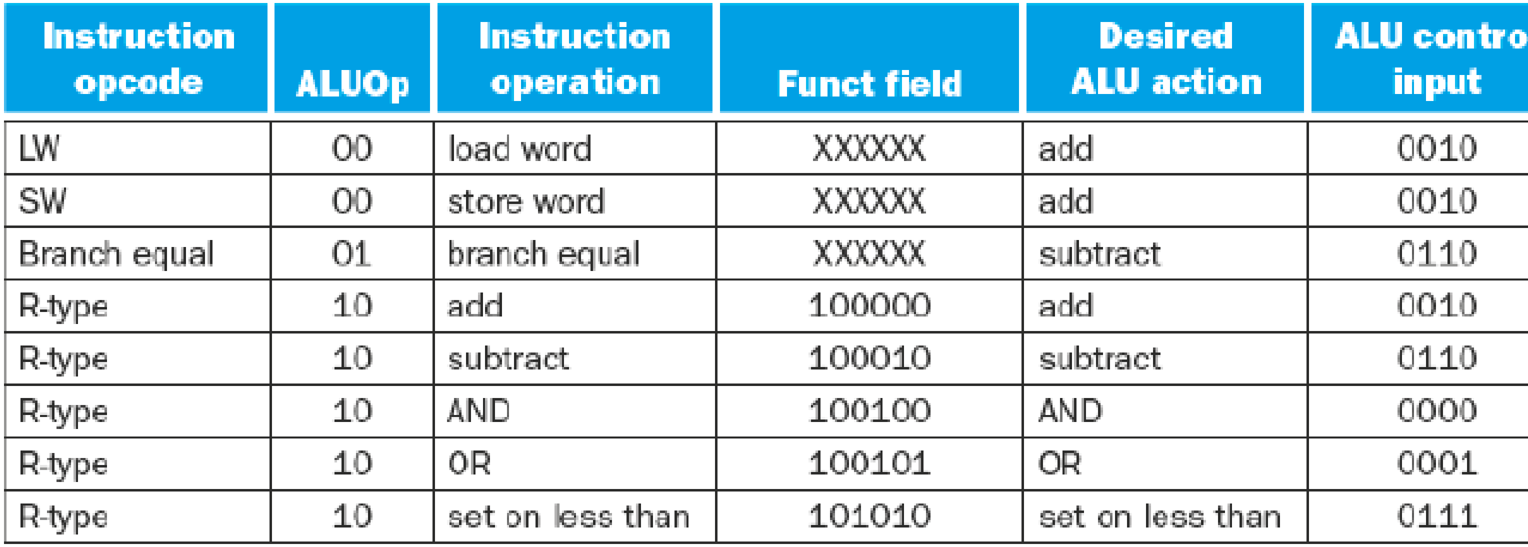


- ALU must perform
  - *add* for load/stores (ALUOp 00)
  - *sub* for branches (ALUOp 01)
  - one of *and* , *or* , *add* , *sub* , *slt* for R-type instructions, depending on the instruction’s 6-bit funct field (ALUOp 10)

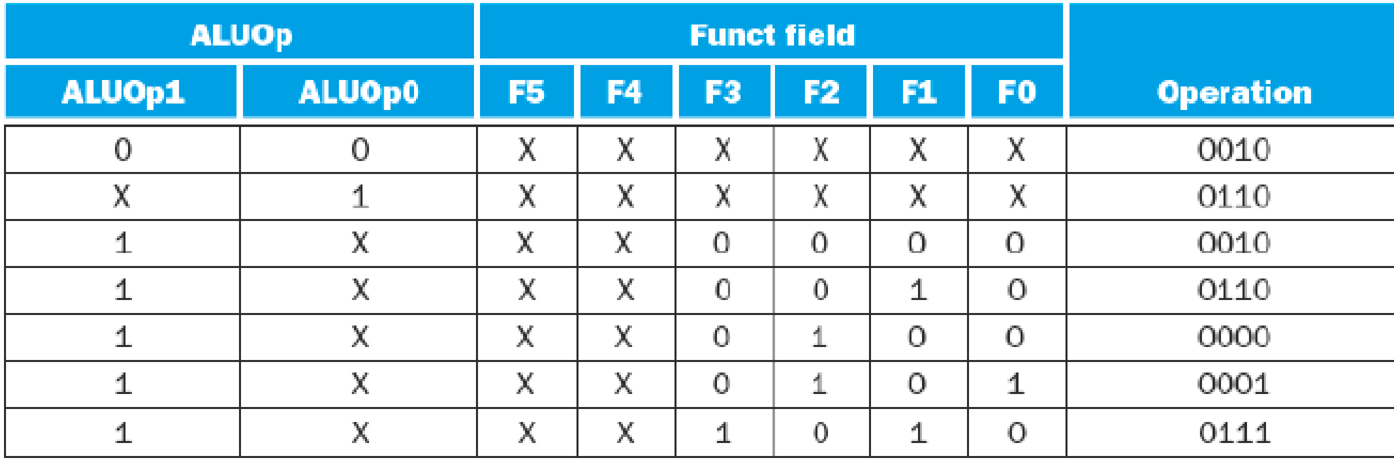
Setting up the ALU Control



control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

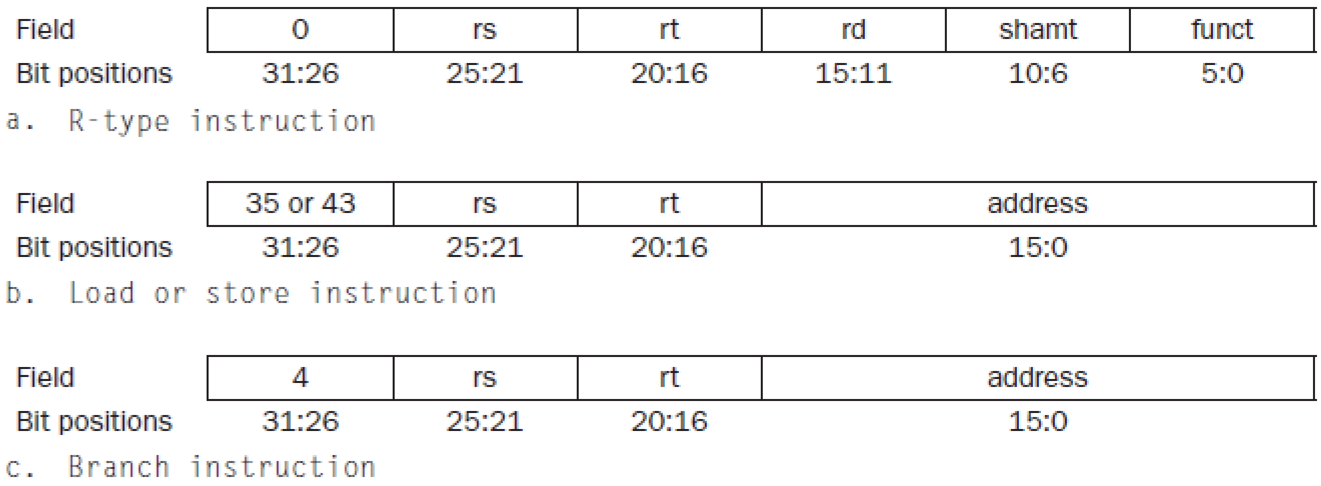


Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU contro input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111



ALUOp		Funct field						
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	Operation
0	0	X	X	X	X	X	X	0010
X	1	X	X	X	X	X	X	0110
1	X	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	X	X	X	0	1	0	0	0000
1	X	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

Designing the Main Control



Field	0	rs	rt	rd	shamt	funct
Bit positions	31:26	25:21	20:16	15:11	10:6	5:0

a. R-type instruction

Field	35 or 43	rs	rt	address
Bit positions	31:26	25:21	20:16	15:0

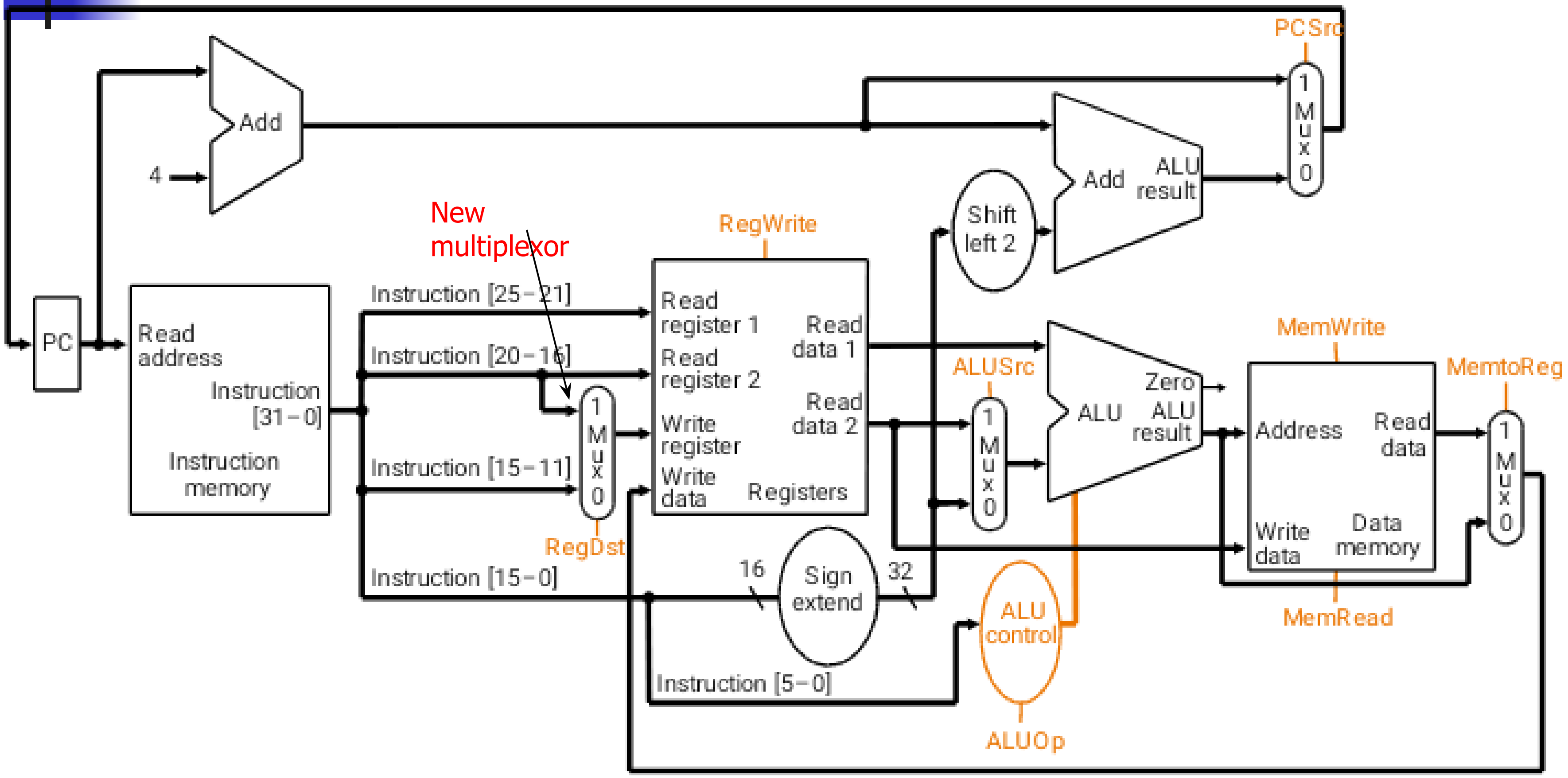
b. Load or store instruction

Field	4	rs	rt	address
Bit positions	31:26	25:21	20:16	15:0

c. Branch instruction

- Observations about MIPS instruction format
  - opcode is always in bits 31-26
  - two registers to be read are always rs (bits 25-21) and rt (bits 20-16)
  - base register for load/stores is always rs (bits 25-21)
  - 16-bit offset for branch equal and load/store is always bits 15-0
  - destination register for loads is in bits 20-16 (rt) while for R-type instructions it is in bits 15-11 (rd) (*will require multiplexor to select*)

Datapath with Control I



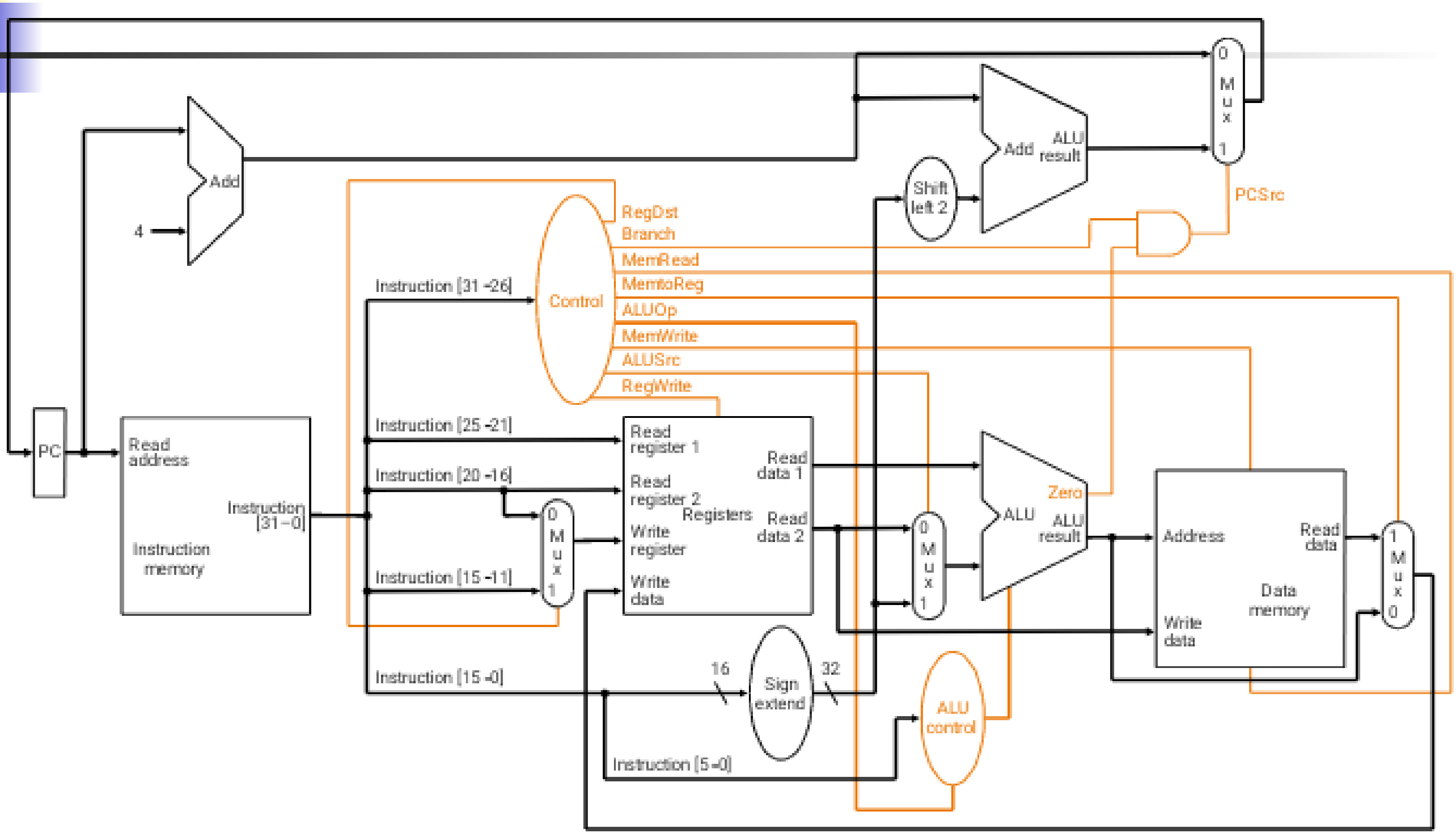
what are the functions of the control signals?

Control Signals

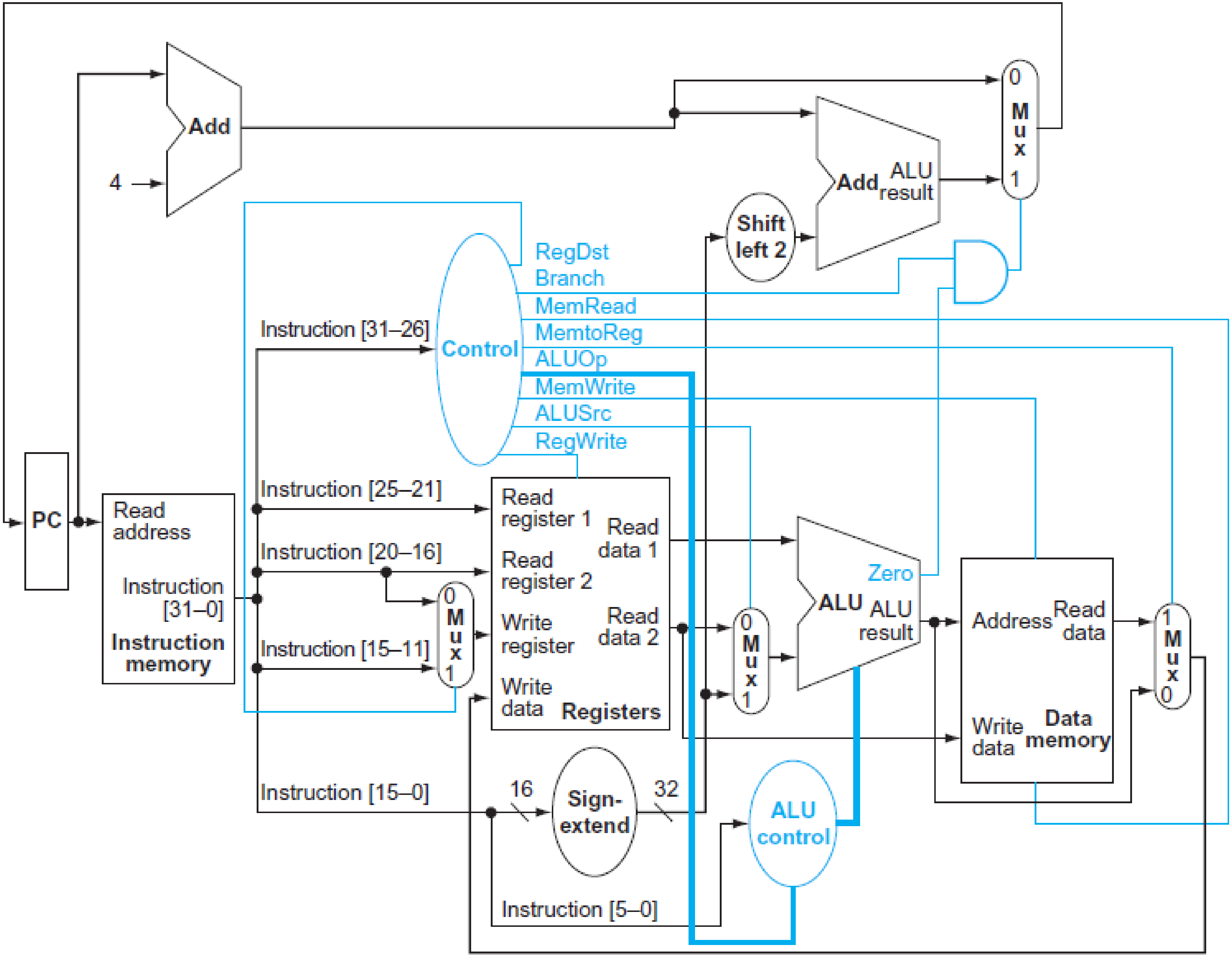
Signal Name	Effect when deasserted	Effect when asserted
RegDst	The register destination number for the Write register comes from the rt field (bits 20-16)	The register destination number for the Write register comes from the rd field (bits 15-11)
RegWrite	None	The register on the Write register input is written with the value on the Write data input
AlLUSrc	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 first Read data output
MemWrite	None	Data memory contents designated by the address input are replaced by the value of 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

Effects of the seven control signals

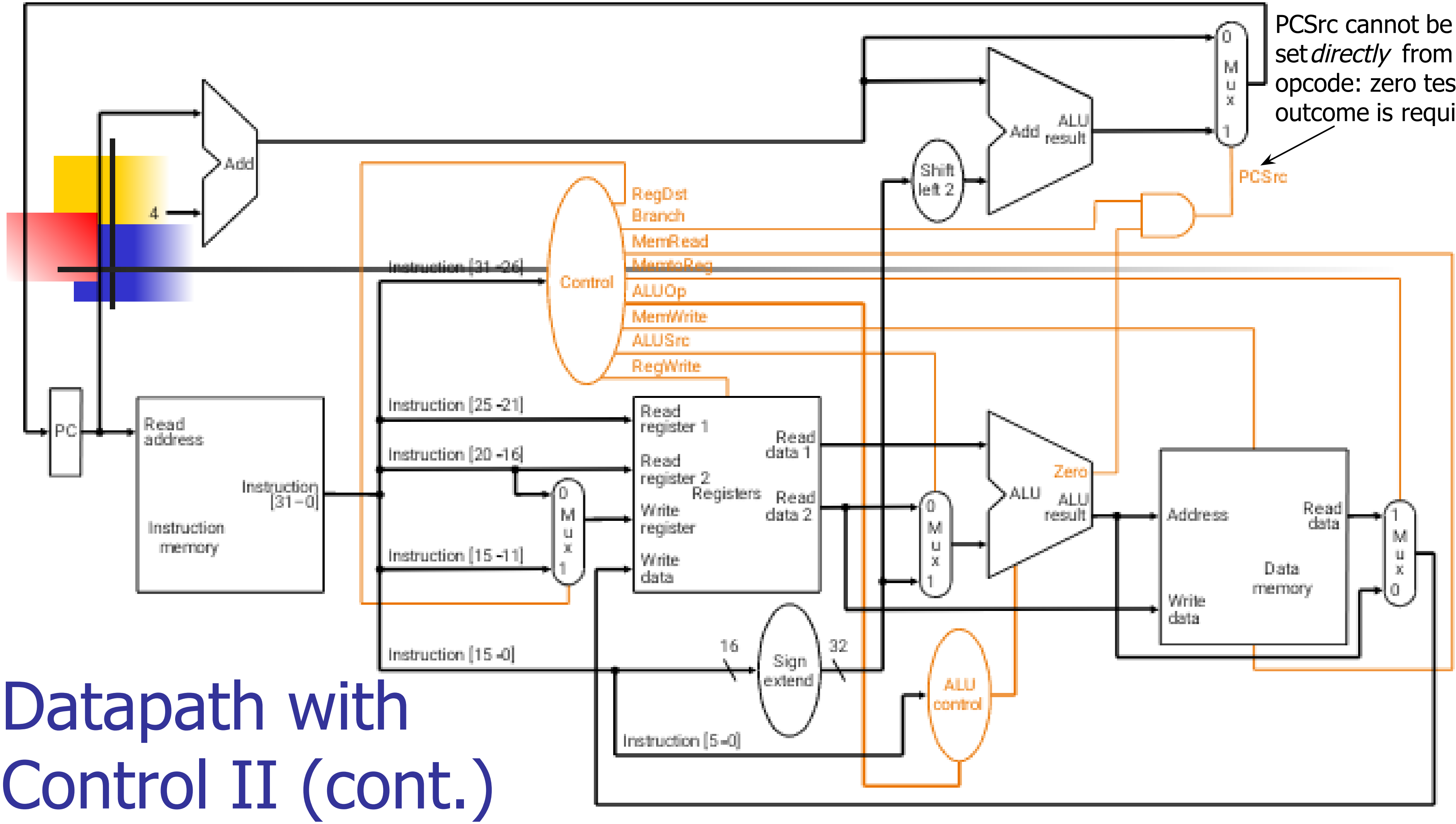
Datapath with Control



MIPS datapath with the control unit: input to control is the 6-bit instruction opcode field, output is seven 1-bit signals and the 2-bit ALUOp signal



Add
4
ALU
Add result
Shift left 2
0
M
u
x
1
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
0
M
u
x
1
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
0
M
u
x
1
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
1
M
u
x
0
Instruction [15–0]
16
Sign-extend
32
ALU control
Instruction [5–0]



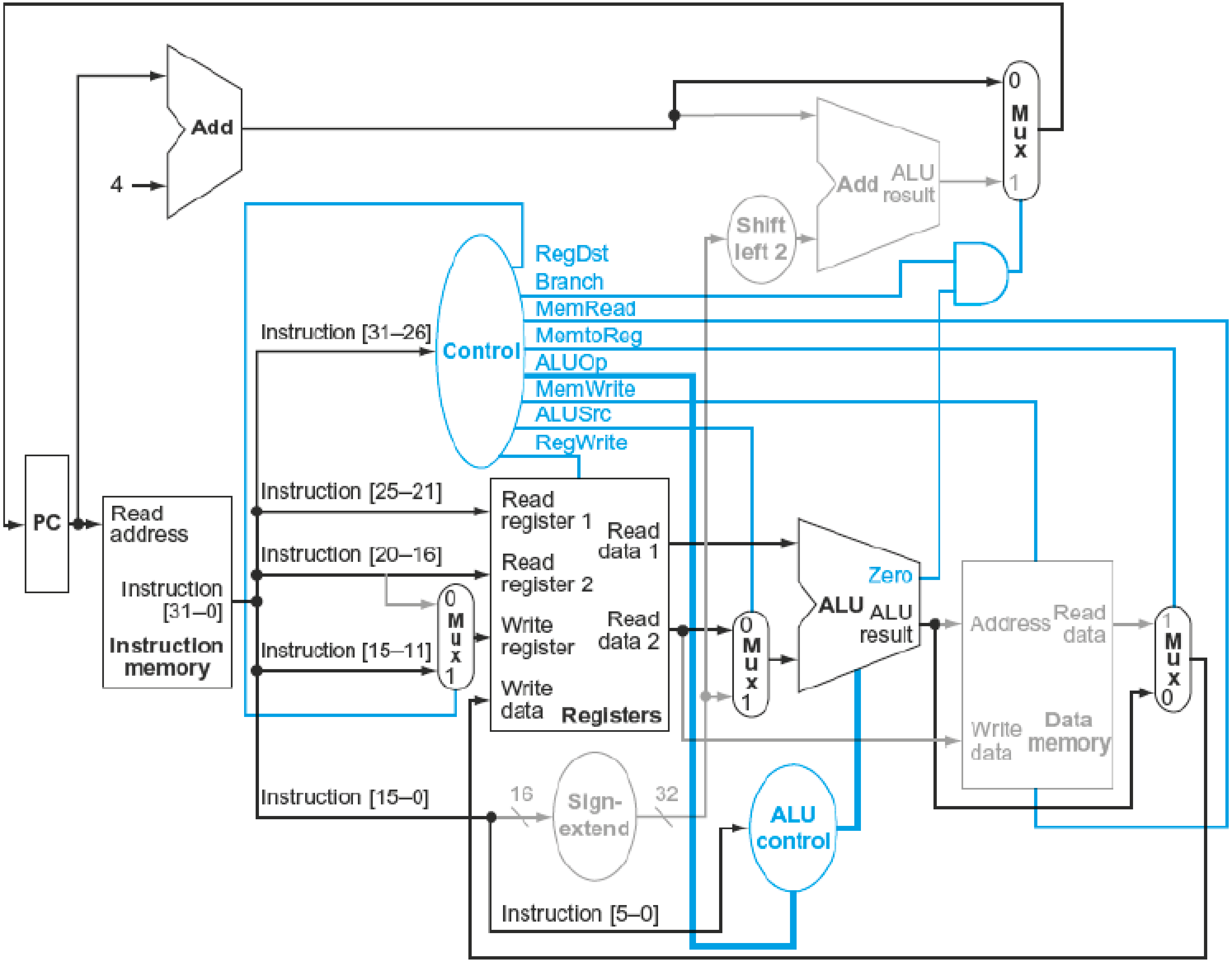
PCSrc cannot be set *directly* from t
opcode: zero test
outcome is require

Datapath with Control II (cont.)

Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUp0
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

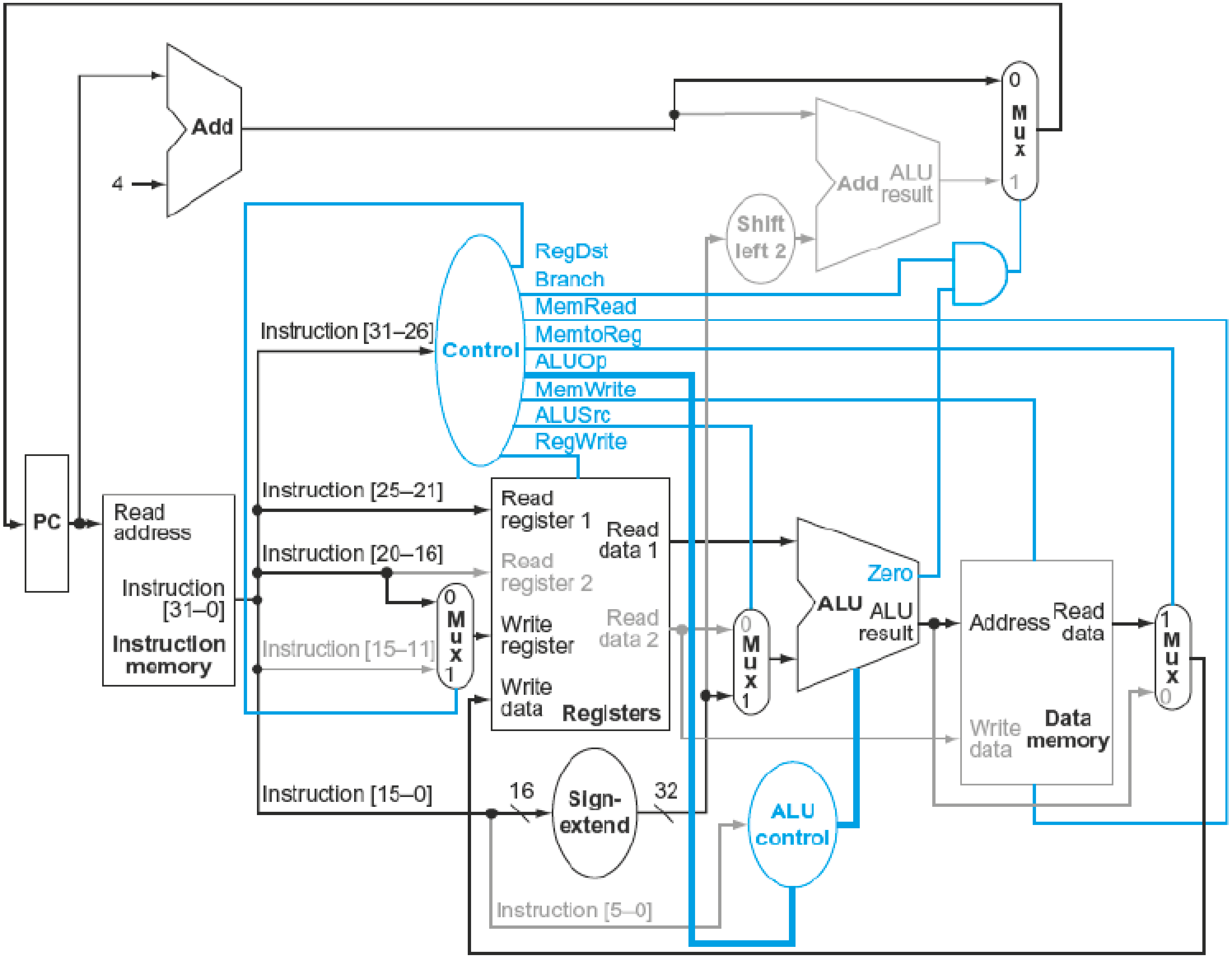
Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0

R-type instruction (add)

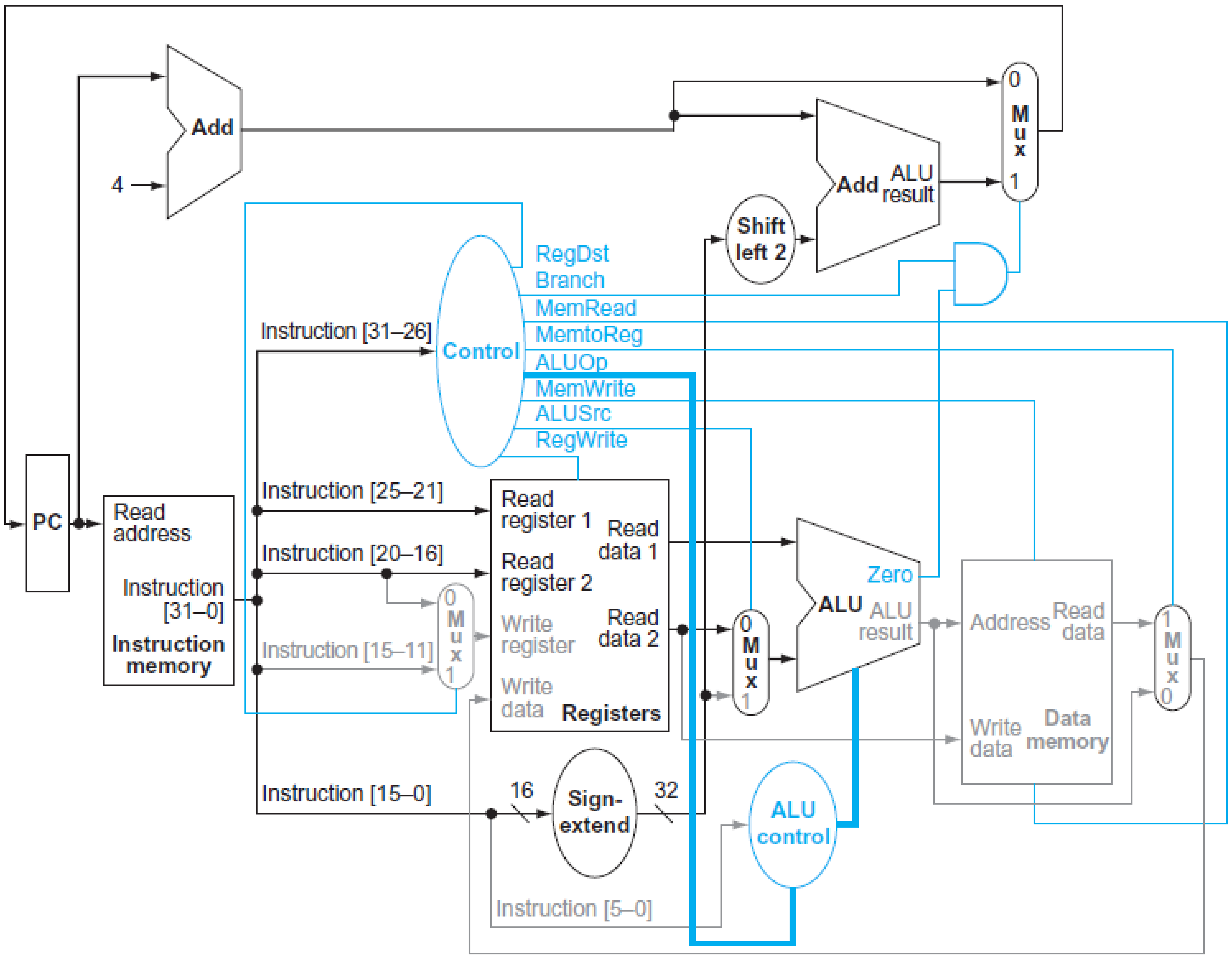


Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
lw	0	1	1	1	1	0	0	0	0

I-type instruction (lw)



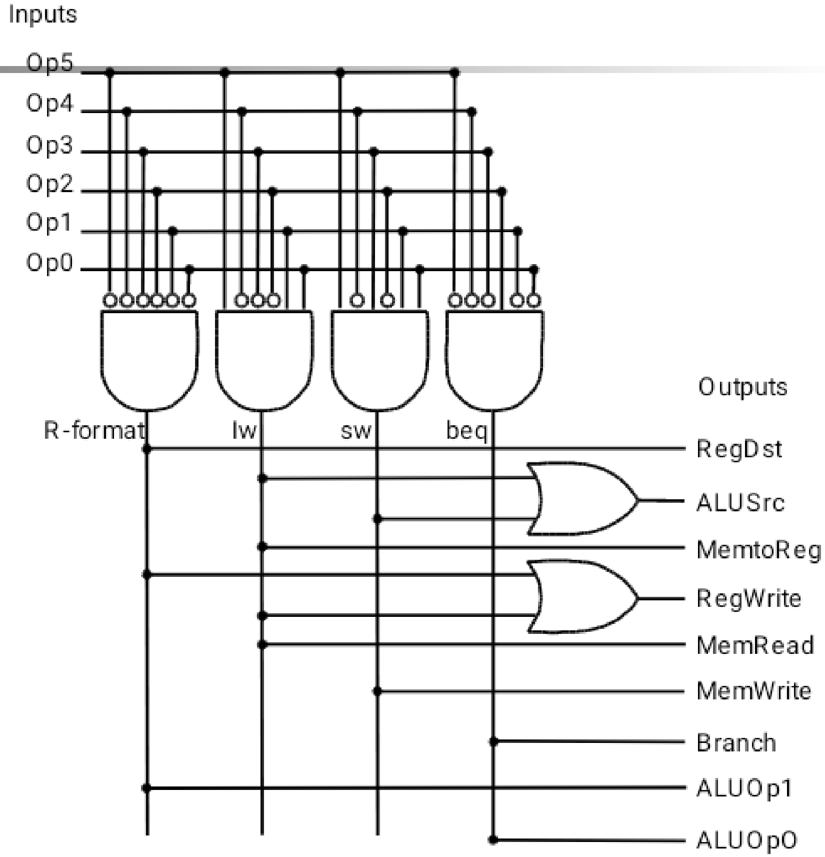
Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
beq	X	0	X	0	0	0	1	0	1



Implementation: Main Control Block

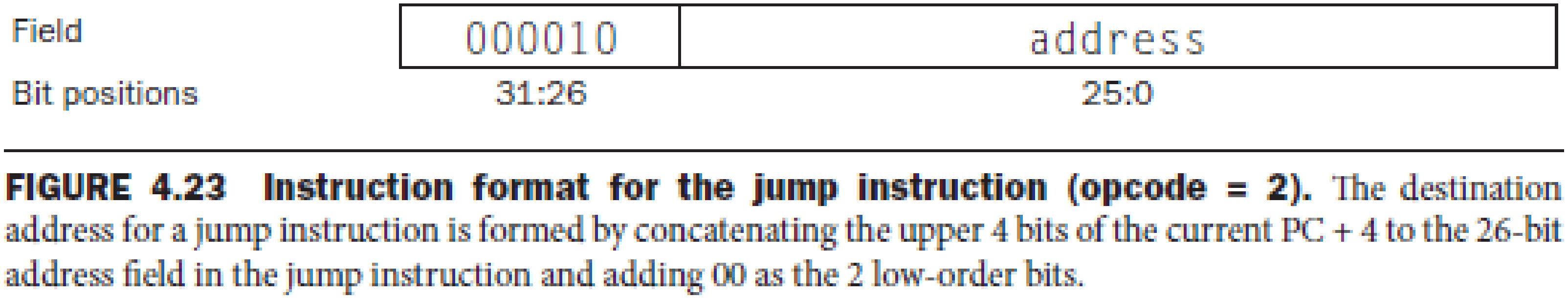
	Signal name	R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	x	x
	ALUSrc	0	1	1	0
	MemtoReg	0	1	x	x
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOP2	0	0	0	1

Truth table for main control signals



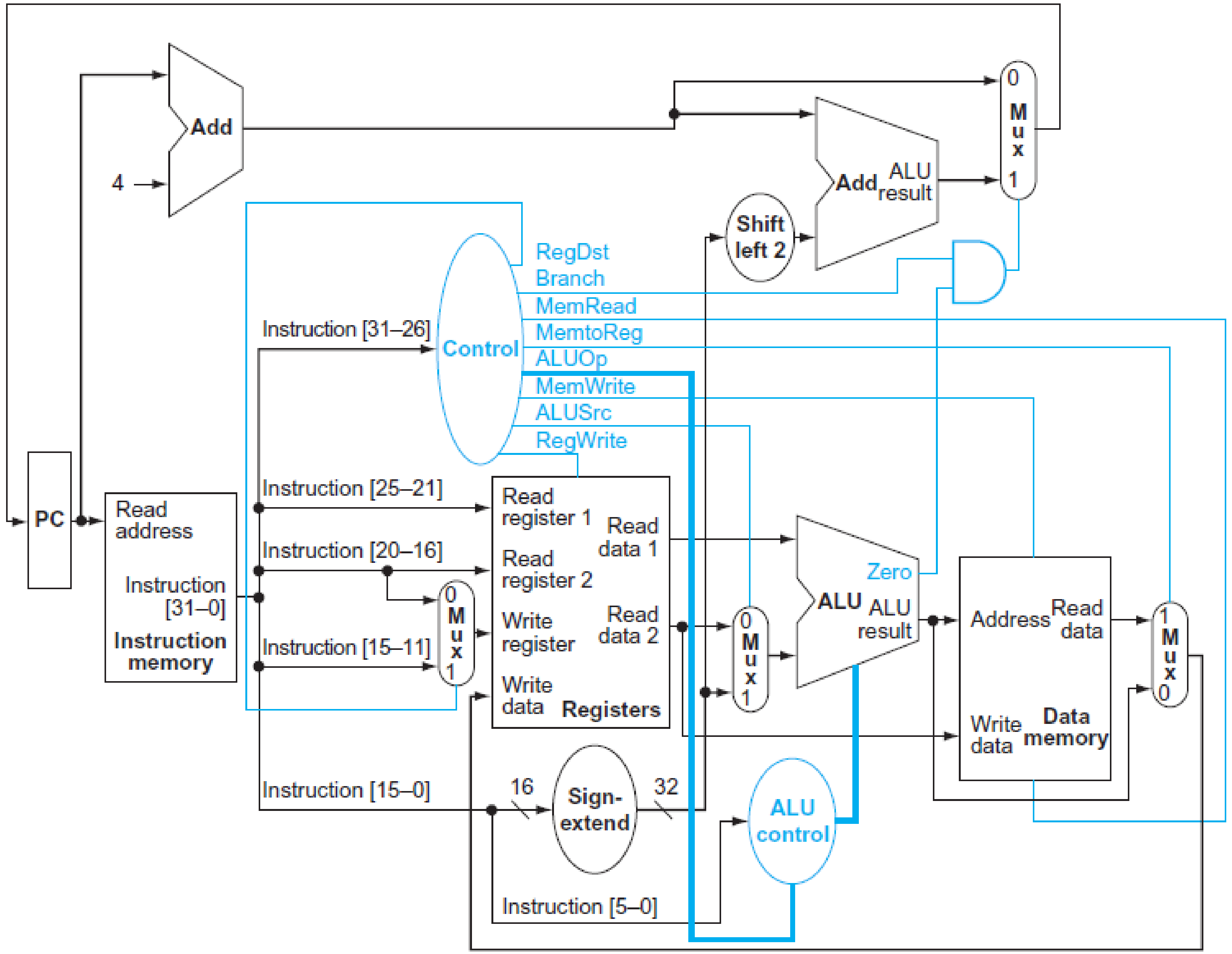
Main control PLA (programmable logic array): principle underlying PLAs is that any logical expression can be written as a sum-of-products

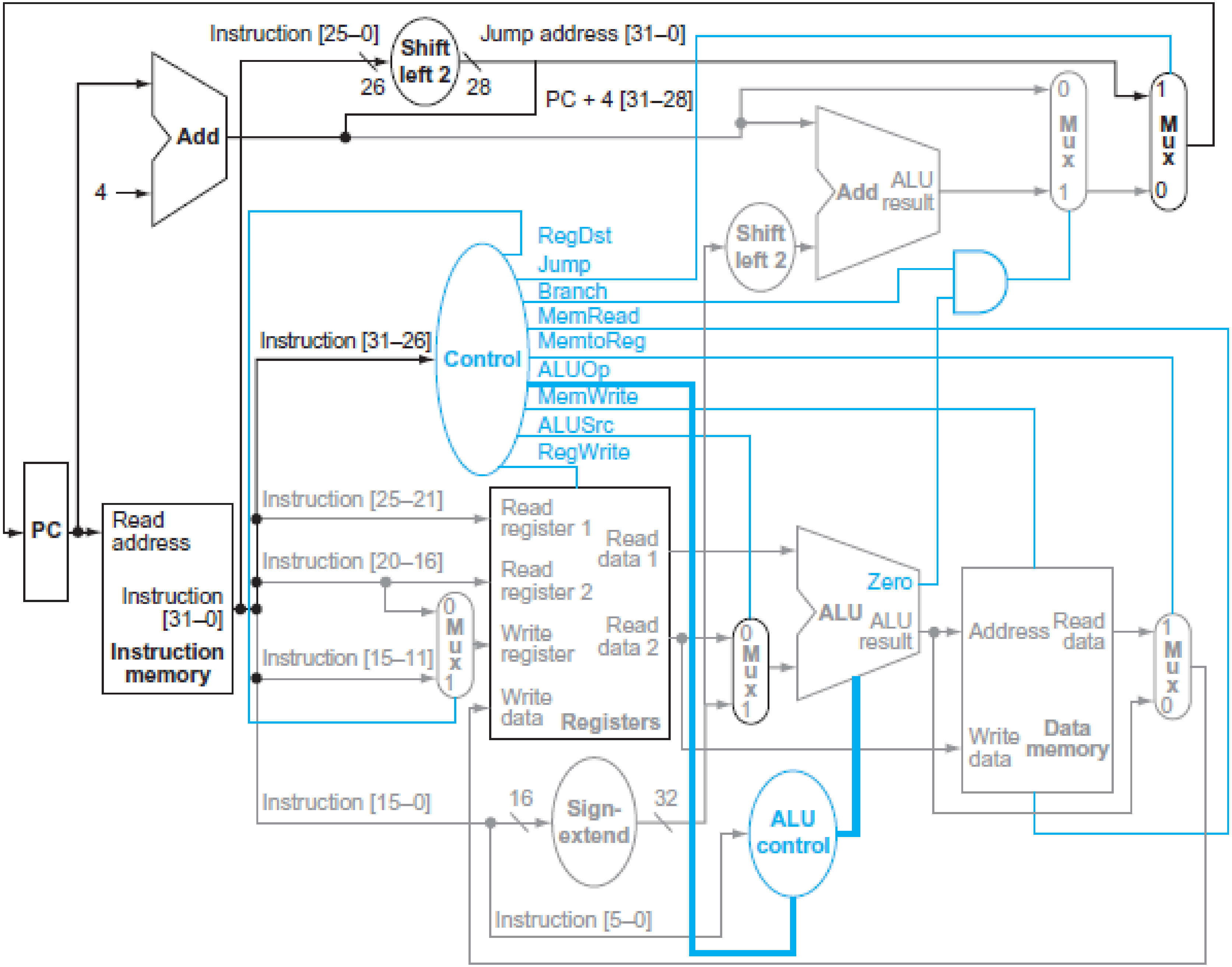
Implementing JUMPs



Field	000010	address
Bit positions	31:26	25:0

FIGURE 4.23 Instruction format for the jump instruction (opcode = 2). The destination address for a jump instruction is formed by concatenating the upper 4 bits of the current PC + 4 to the 26-bit address field in the jump instruction and adding 00 as the 2 low-order bits.


Add
4
ALU
Add result
Shift left 2
0 M u x 1
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
PC
Read address
Instruction [31–0]
Instruction memory
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
0 M u x 1
0 M u x 1
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
1 M u x 0
16
32
Sign-extend
ALU control


Instruction [25–0]
Shift left 2
26
28
Jump address [31–0]
PC + 4 [31–28]
Add
4
Add
ALU result
0
M u x
1
1
M u x
0
Shift left 2
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
0
M u x
1
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
0
M u x
1
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
1
M u x
0
Instruction [15–0]
16
Sign-extend
32
ALU control
Instruction [5–0]