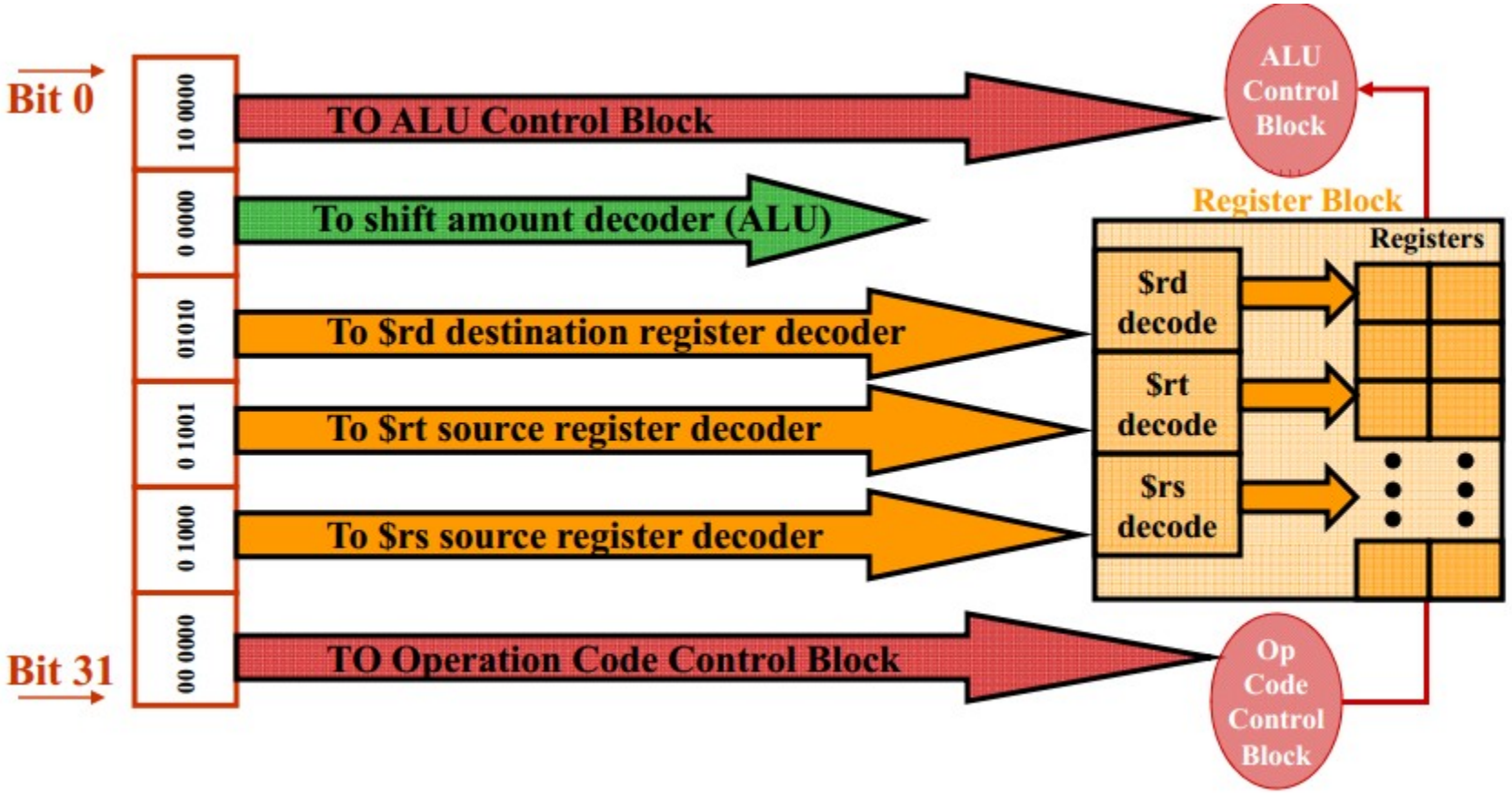
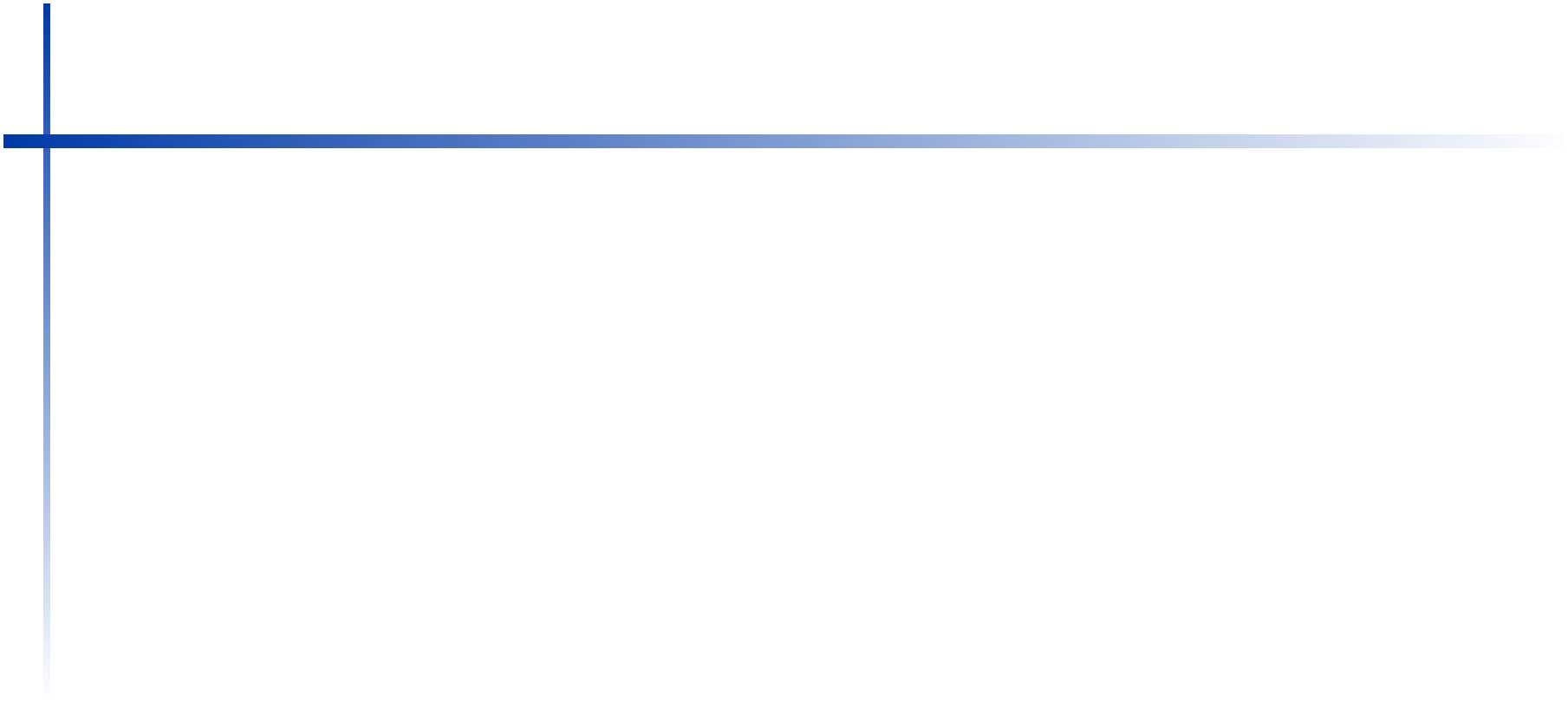
Chapter 4

The Processor

Instruction disposition





Control unit design

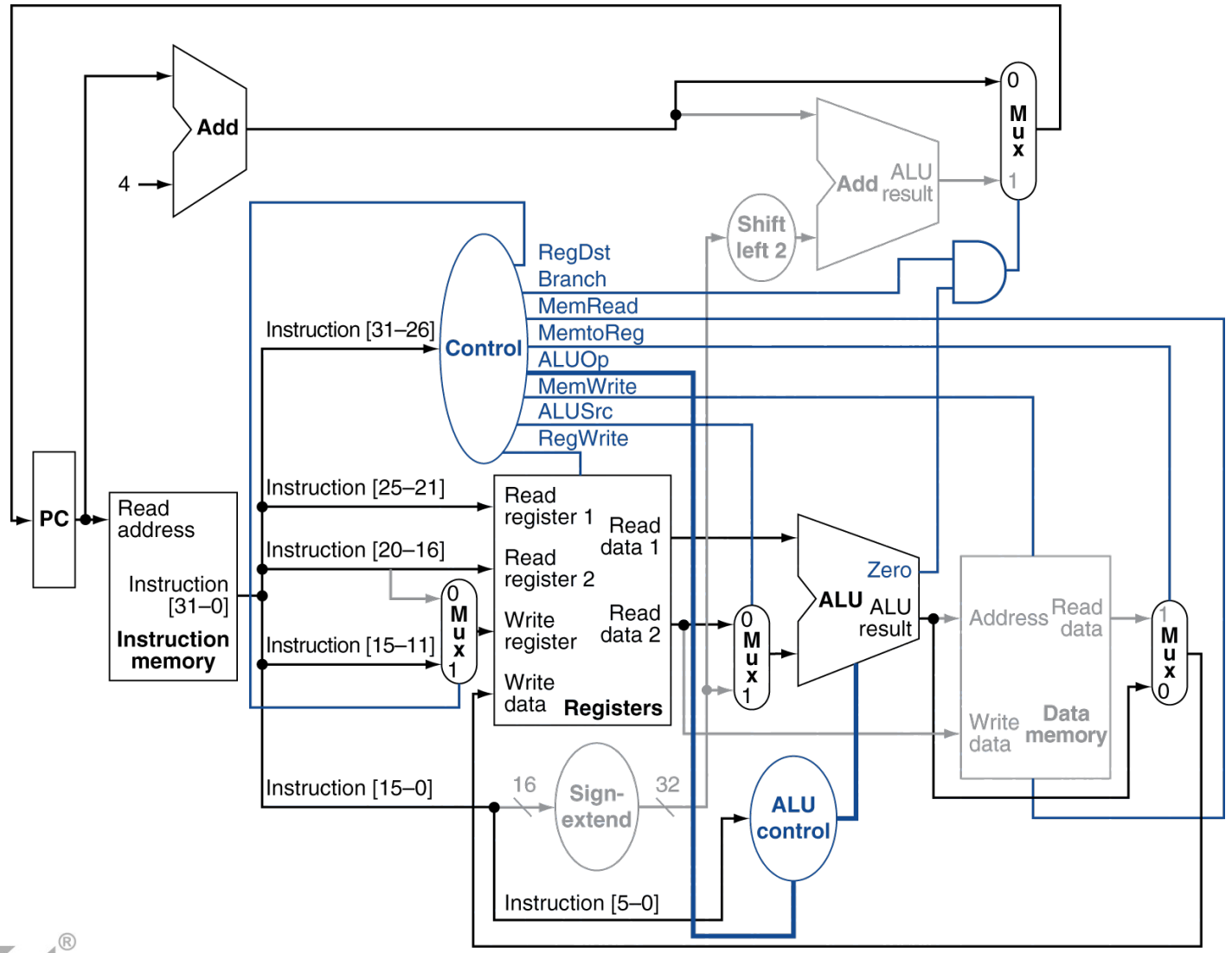
- The control unit must be able to take inputs and generate
  - A write signal for each state element,
  - Selector control for each multiplexor and
  - The ALU control.

ALU Control

- ALU used for
  - Load/Store: F = add
  - Branch: F = subtract
  - R-type: F depends on funct field

ALU control	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set-on-less-than

R-Type Instruction



ALU Control Unit

- Must describe hardware to compute **4-bit ALU control** input
  - given instruction type

    00 = lw, sw
    01 = beq,
    10 = arithmetic

    ALUOp
    computed from instruction typ
  - function code for arithmetic

ALU Control

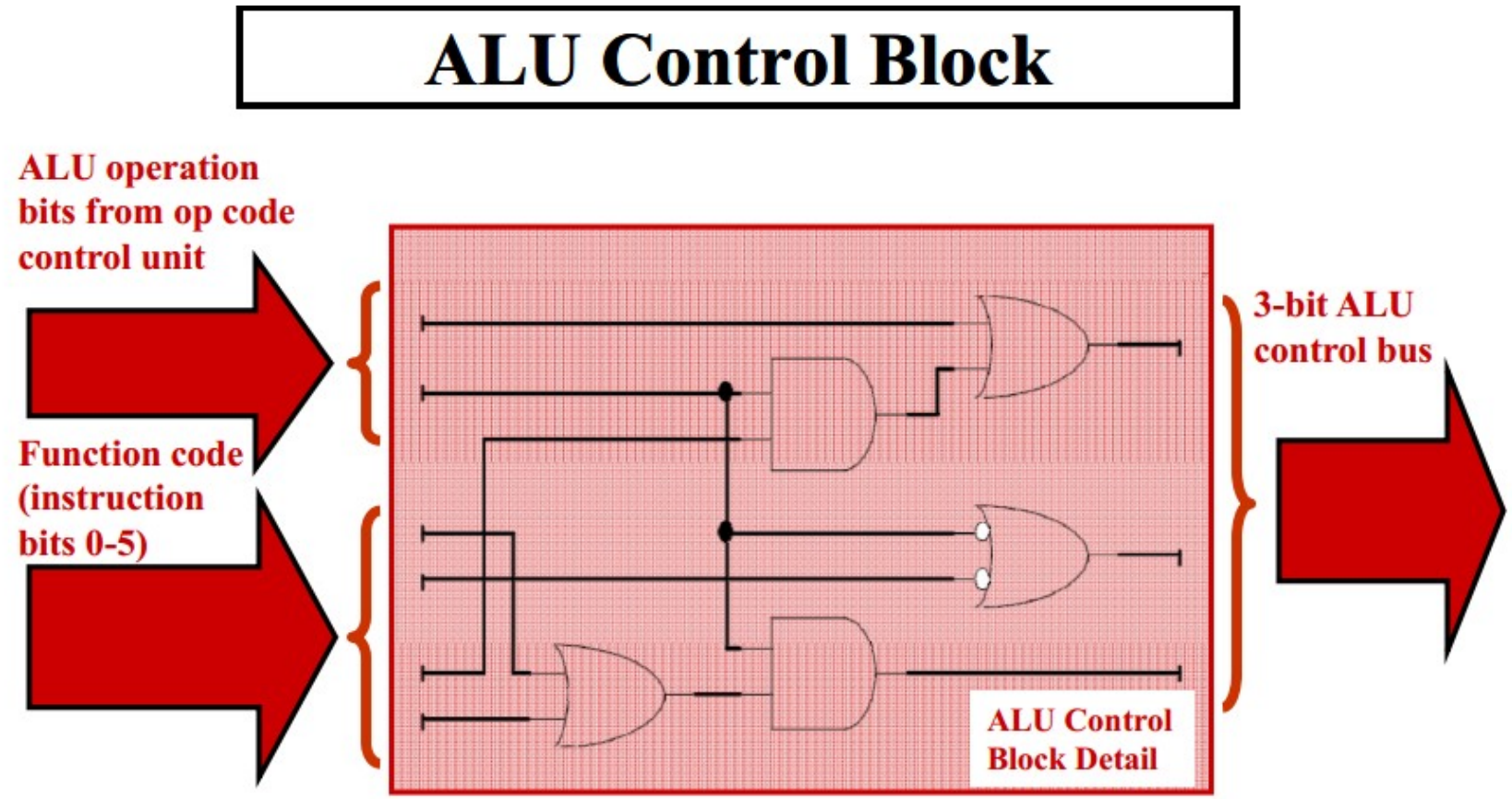
- Assume 2-bit ALUOp derived from opcode
  - Combinational logic derives ALU control

Opcode	ALUOp	Operation	funct	ALU function	ALU control
lw	00	load word	XXXXXX	add	0010
sw	00	store word	XXXXXX	add	0010
Beq	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
		subtract	100010	subtract	0110
		AND	100100	AND	0000
		OR	100101	OR	0001
		set-on-less-than	101010	set-on-less-than	0111

Truth table for ALU control bits

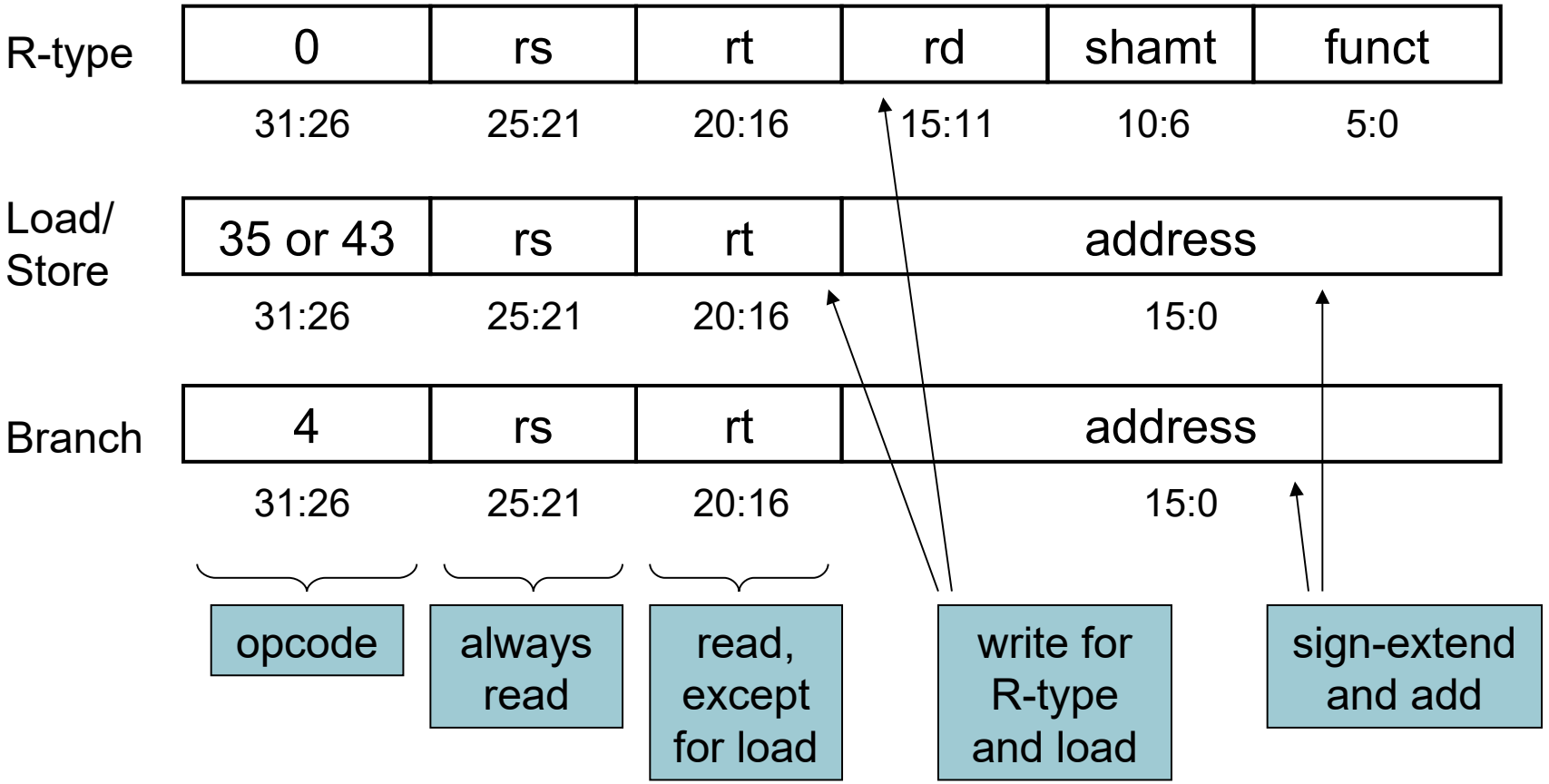
ALUOp		Funct field						Operation
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	0010
X	1	X	X	X	X	X	X	0110
1	X	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	X	X	X	0	1	0	0	0000
1	X	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

Sample ckt for a 3 bit ALU cntrl

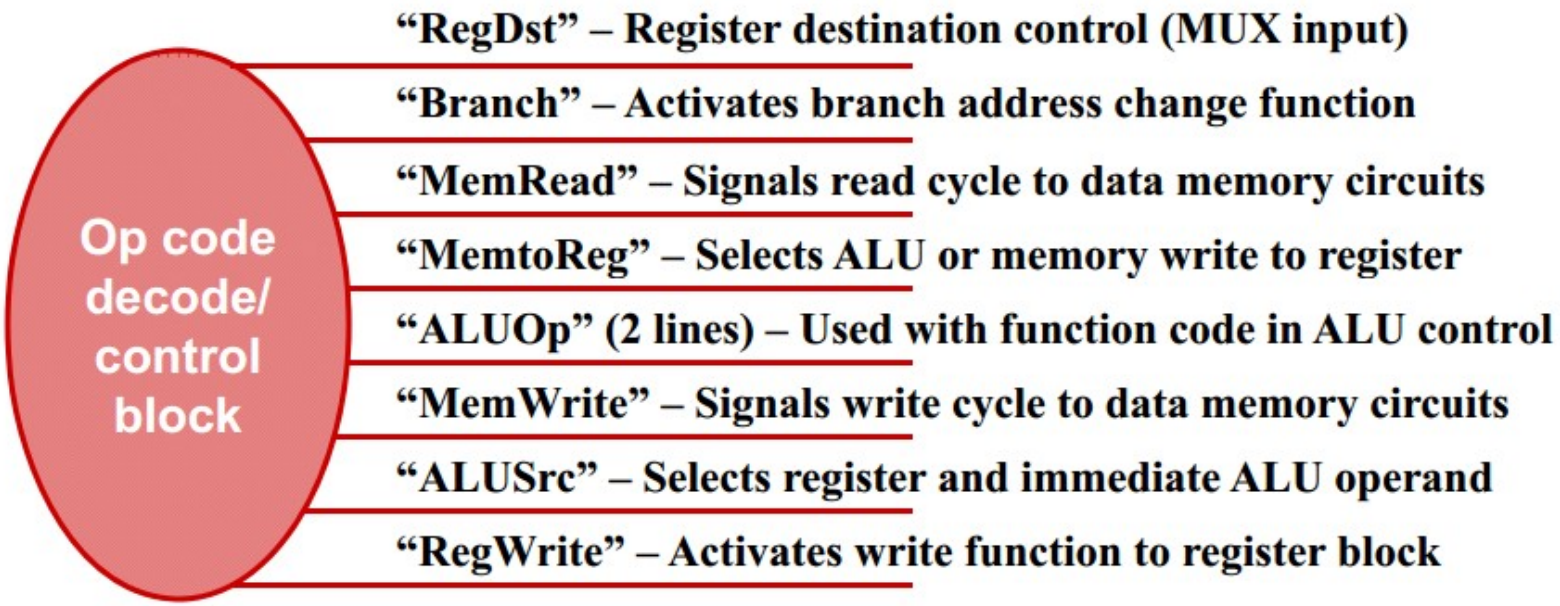


The Main Control Unit

- Control signals derived from instruction



Main control unit



Effect of control signals

Signal Name	When Signal = 1	When Signal = 0
RegDst	Write reg. = $rd (bits 11–15)	Write reg. = $rt (bits 16-20)
Branch	ALU branch compare activated	No branch activated
MemRead	Memory data → write register	No data read from memory
MemtoReg	Memory data → write register	ALU results → write register
ALUOp	NA; lines go to ALU control block	NA; lines go to ALU control block
MemWrite	ALU or register data → memory	No data written to memory
ALUSrc	2nd ALU operand is immediate (sign-extended instr. bits 0-15)	2nd ALU operand is from $rt (instruction bits 16-20)
RegWrite	Memory/ALU data → write reg.	No input to register block

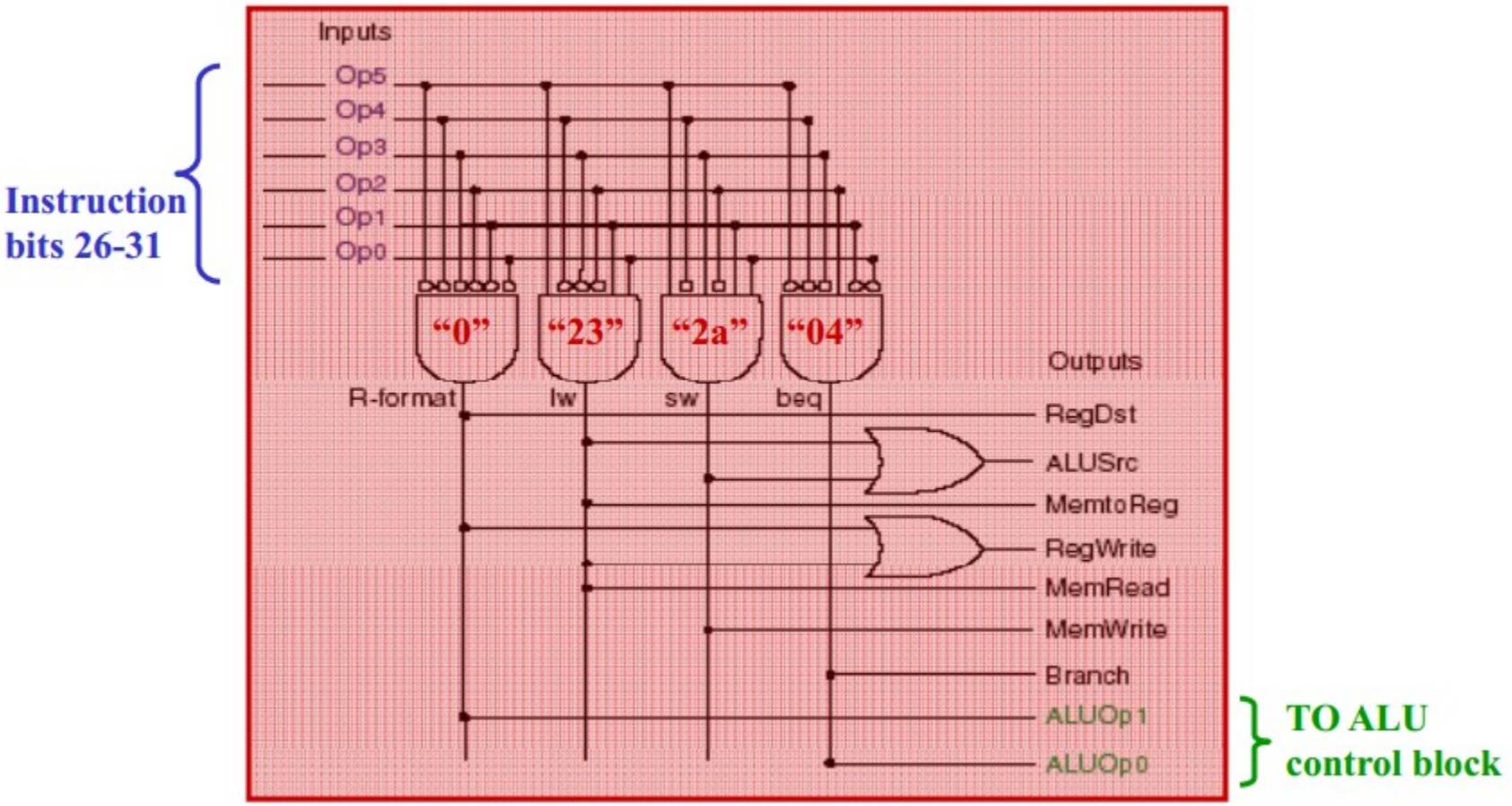
Setting of control signals through opcode fields

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

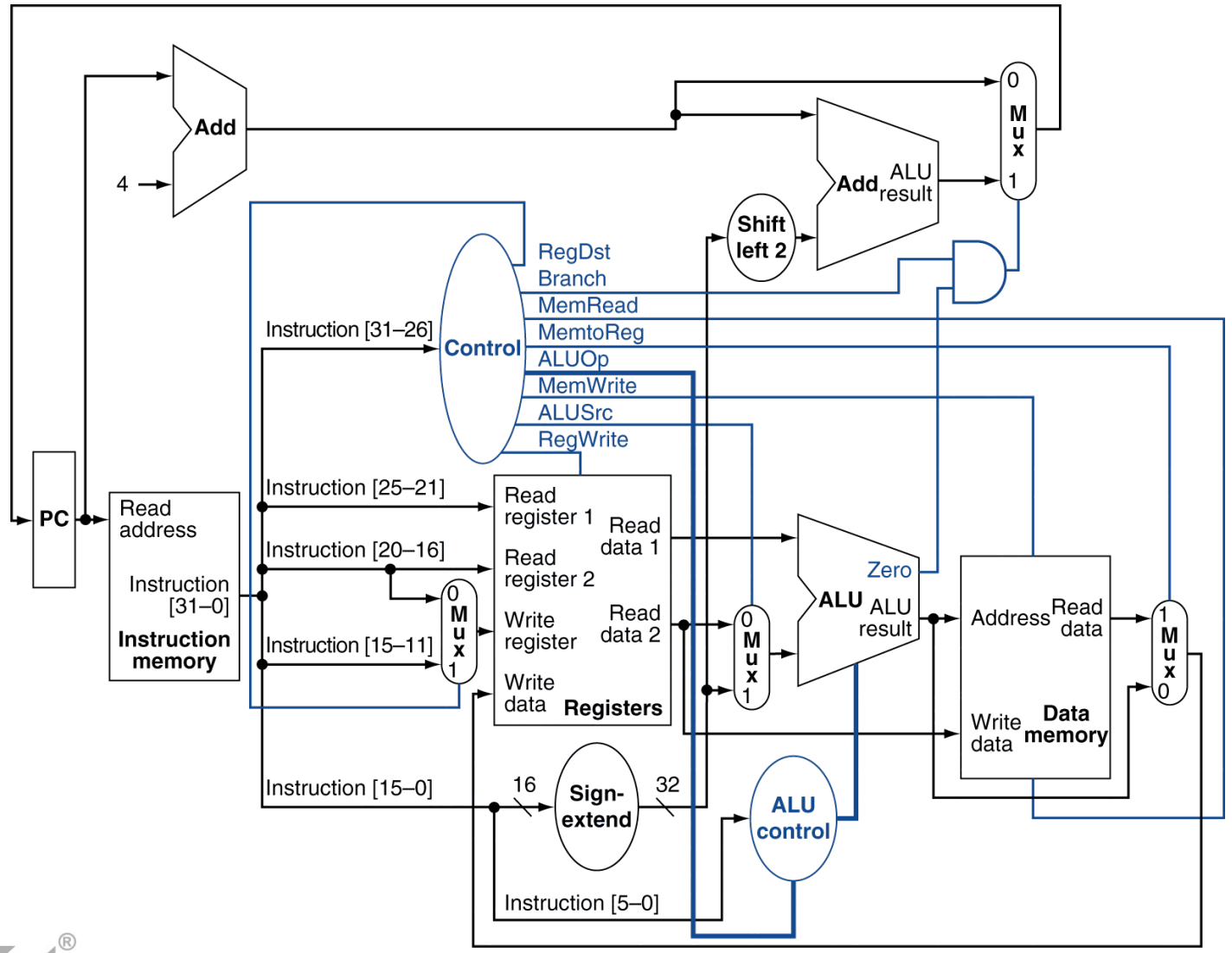
Truth table for control functions

Input or output	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
	ALUOp0	0	0	0	1

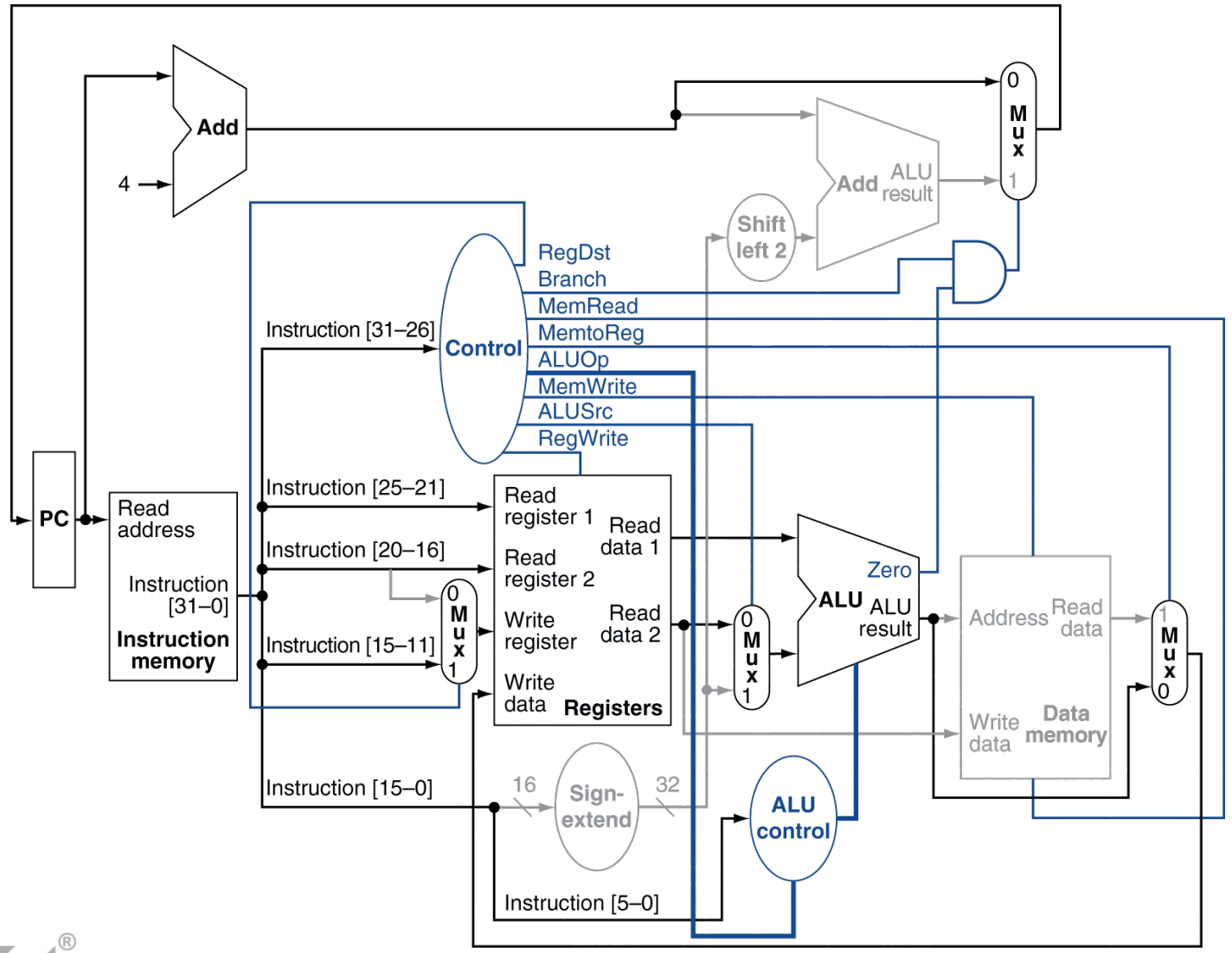
Main control unit

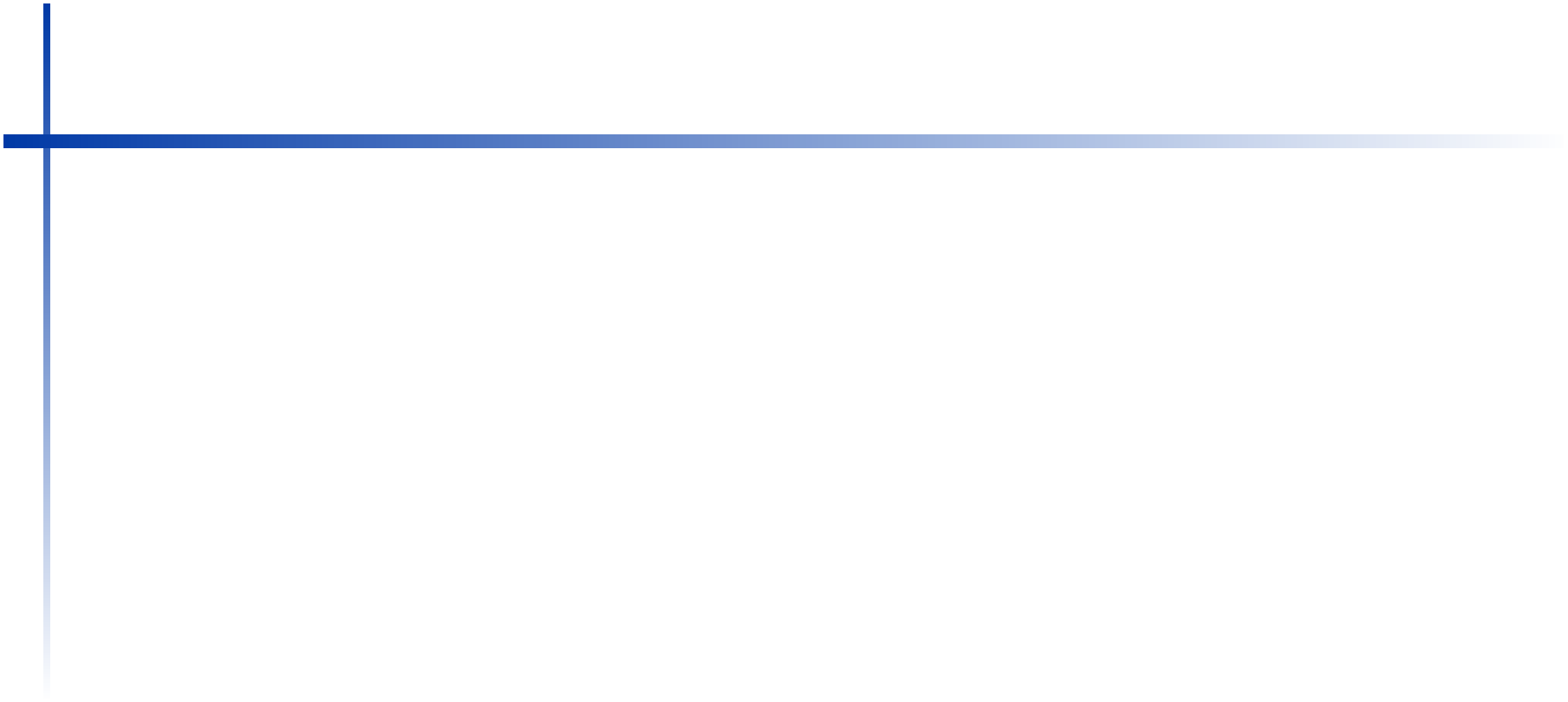


Datapath With Control

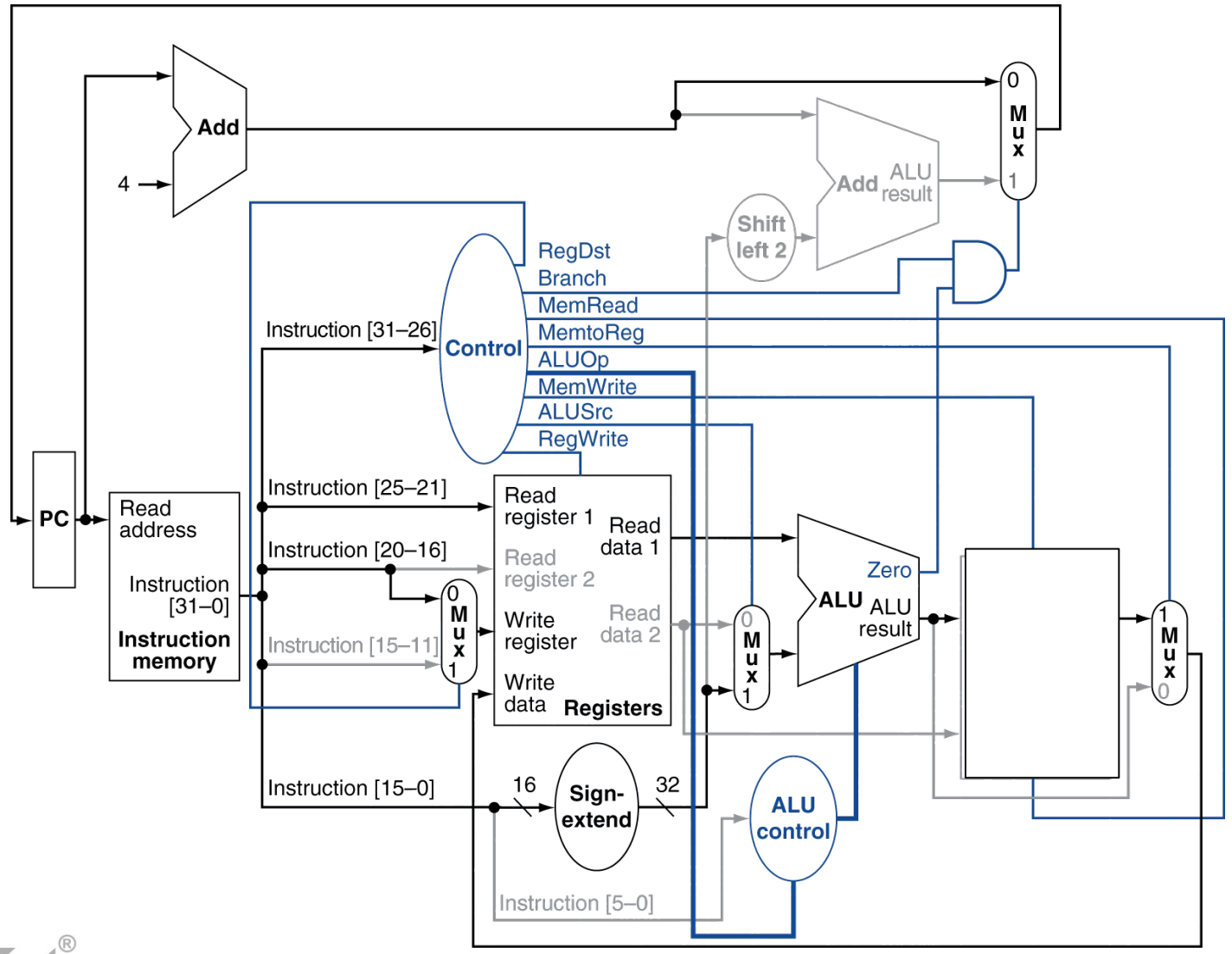


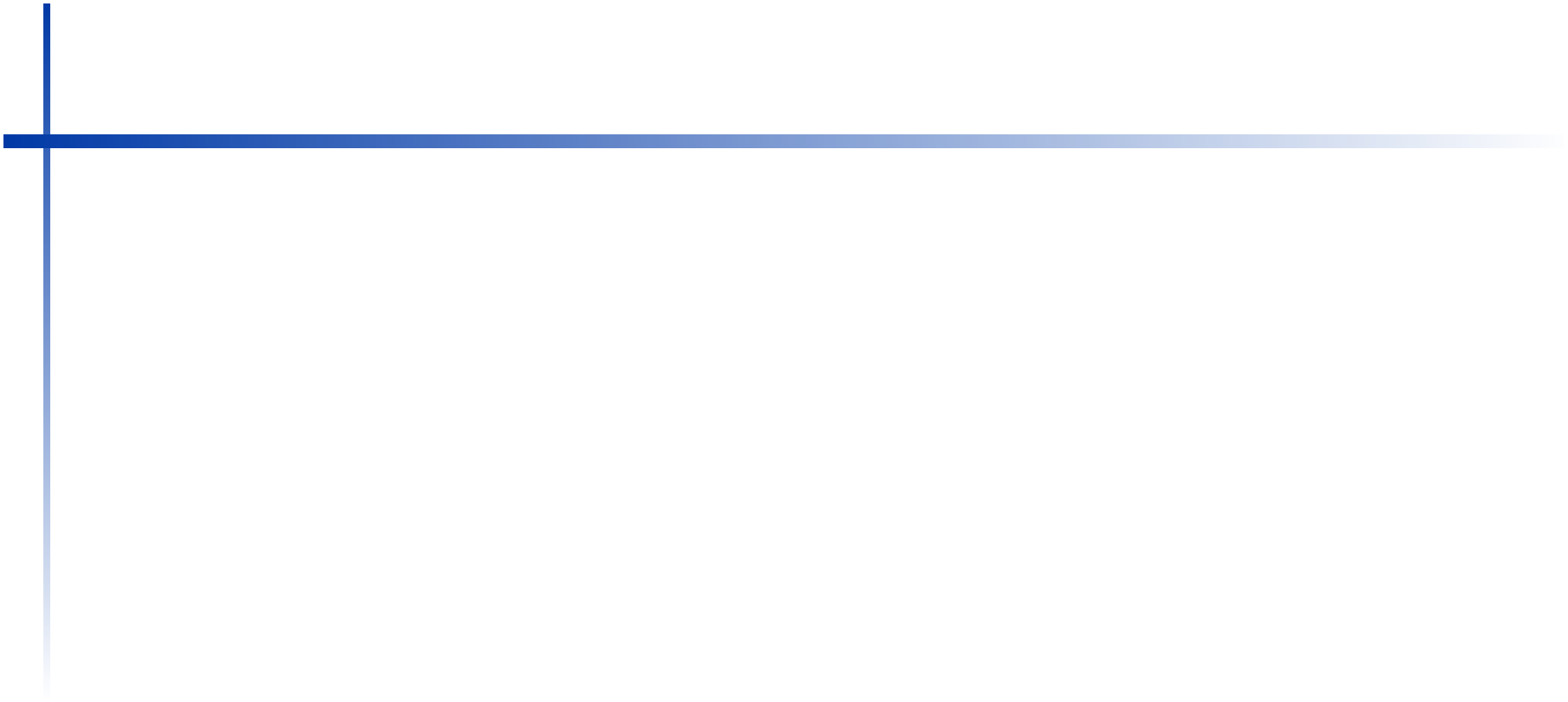
R-Type Instruction





Load Instruction





Branch-on-Equal Instruction