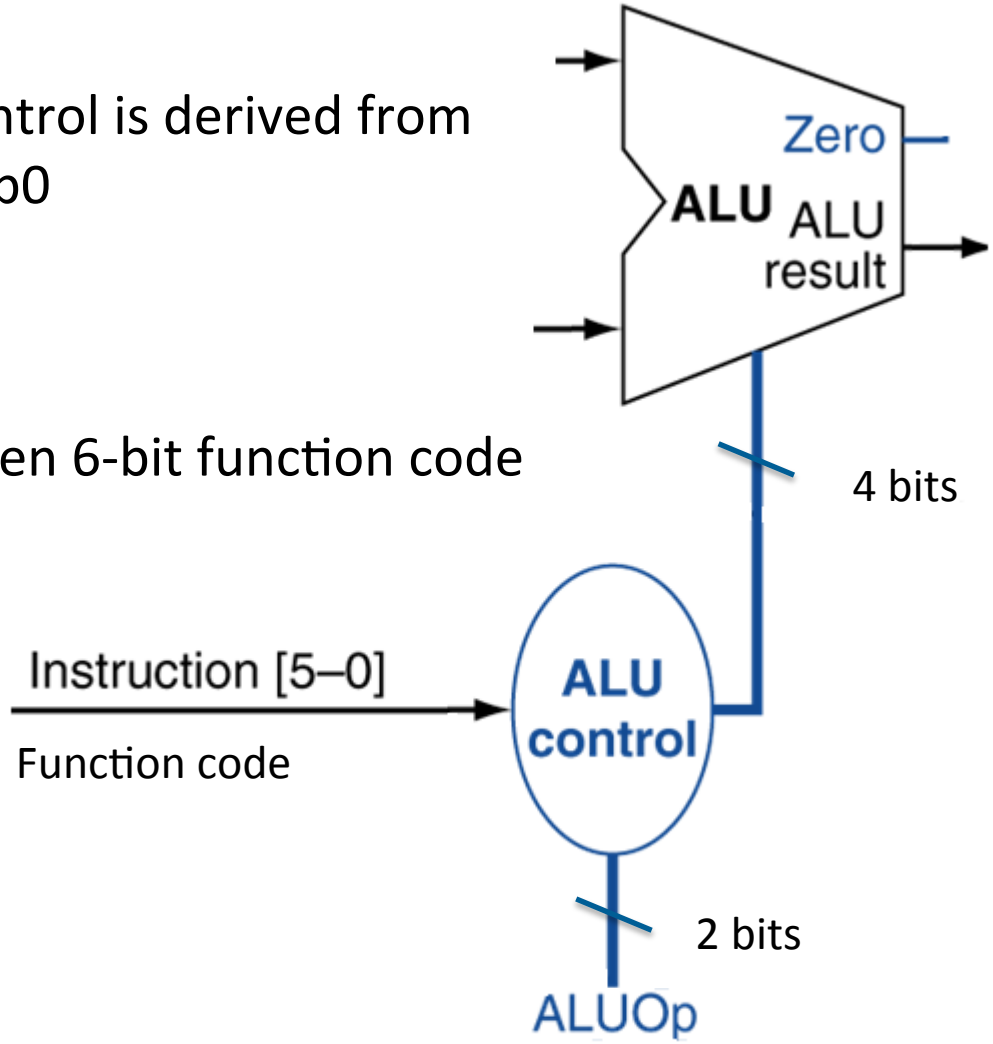
ALU Control

- The ALU takes two input operands
- Generates result as directed by 4-bit control signal
- Sets zero flag if result is 0
- 4-bit control signal derived from opcode & function code

ALU Control

The 4-bit ALU control is derived from ALUOp1 & ALUOp0

If ALUOp = 10, then 6-bit function code is also used



ALU Control

- Instruction type determines ALU operation
  - Load/Store: operation = add
  - Branch: operation = subtract (compare operands)
  - R-type: operation depends on funct field

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

ALU Control

- Combinational logic derives the 2 ALUOp bits from opcode

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

ALU Control

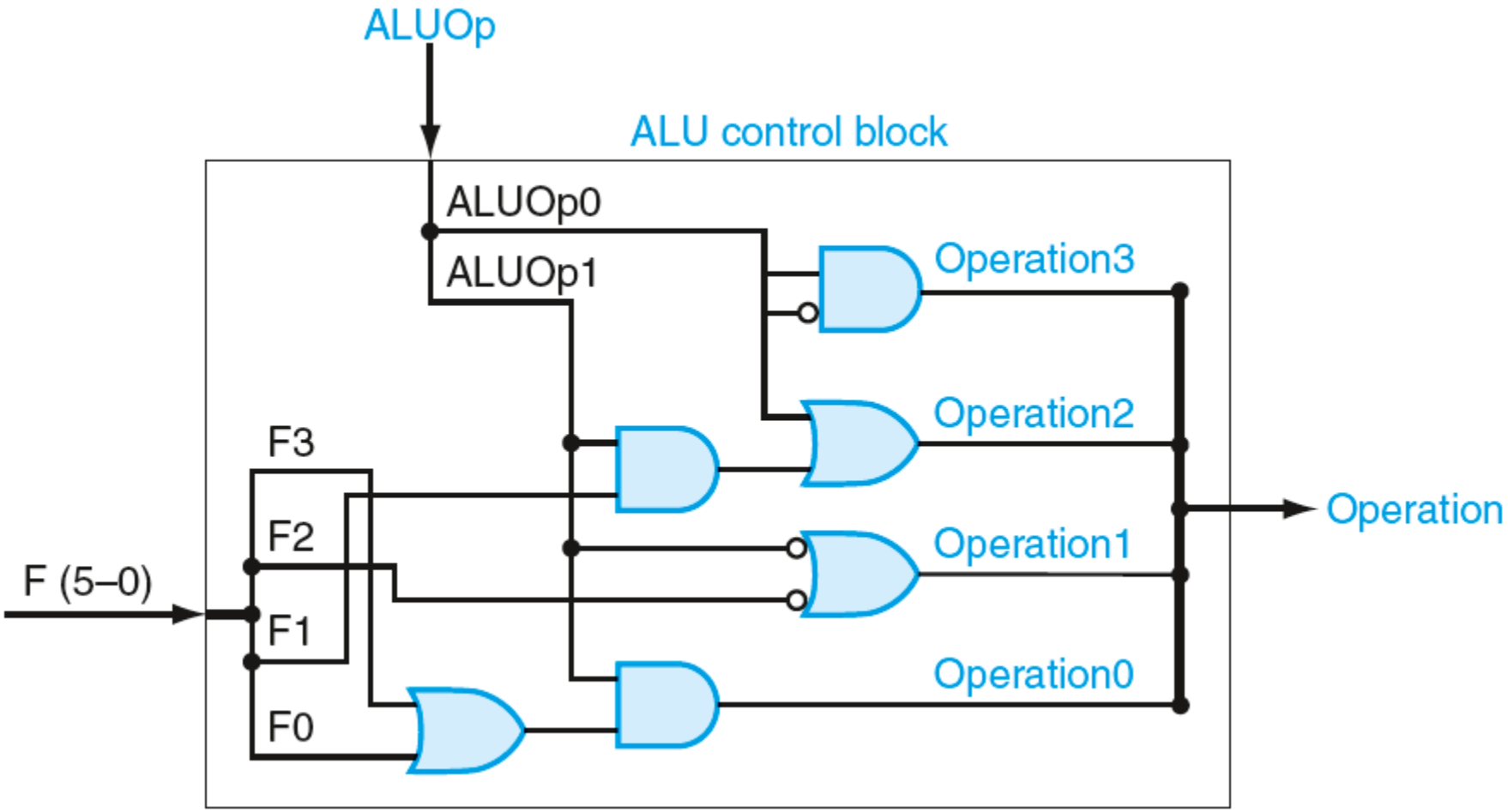
ALUOp		Funct field						Operation
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	0010
X	1	X	X	X	X	X	X	0110
1	X	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	X	X	X	0	1	0	0	0000
1	X	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

- Truth table for the 4 ALU control bits (called Operation)

- Depends on ALUOp and instruction function code field

(X's represent "don't cares")

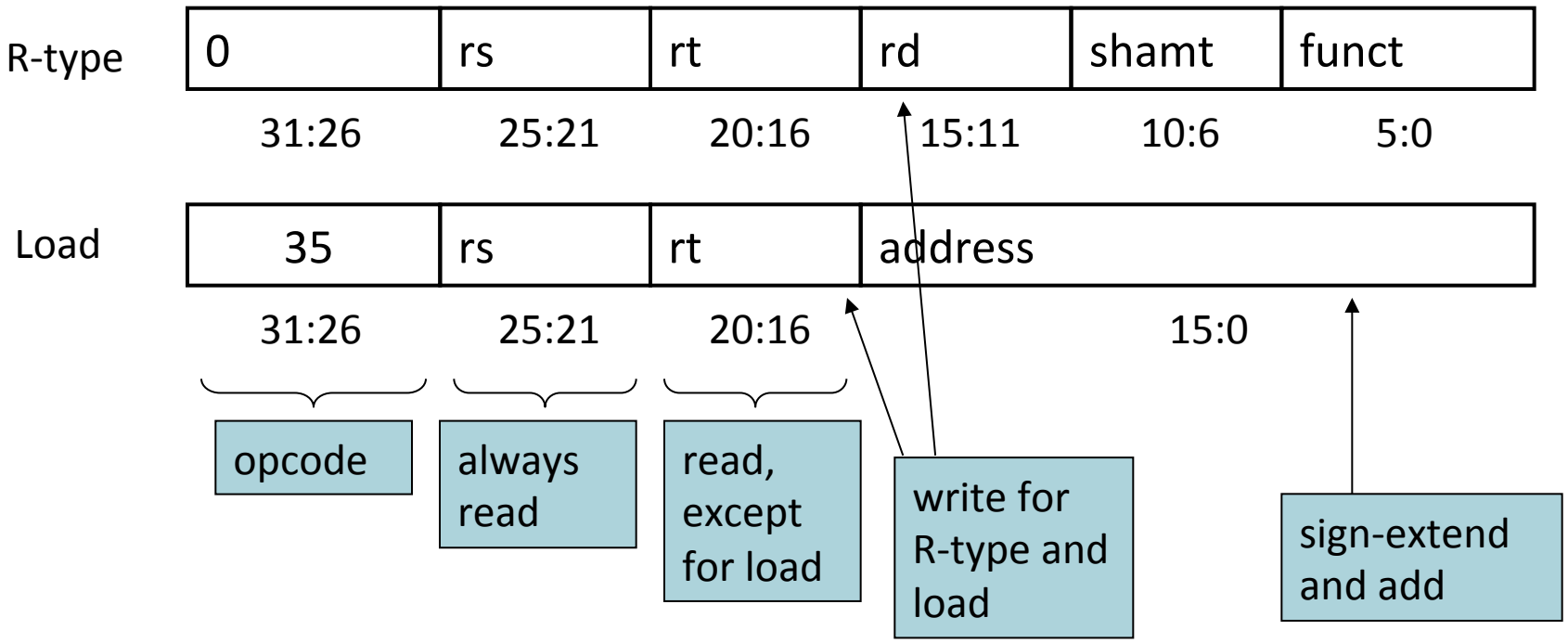
ALU Control



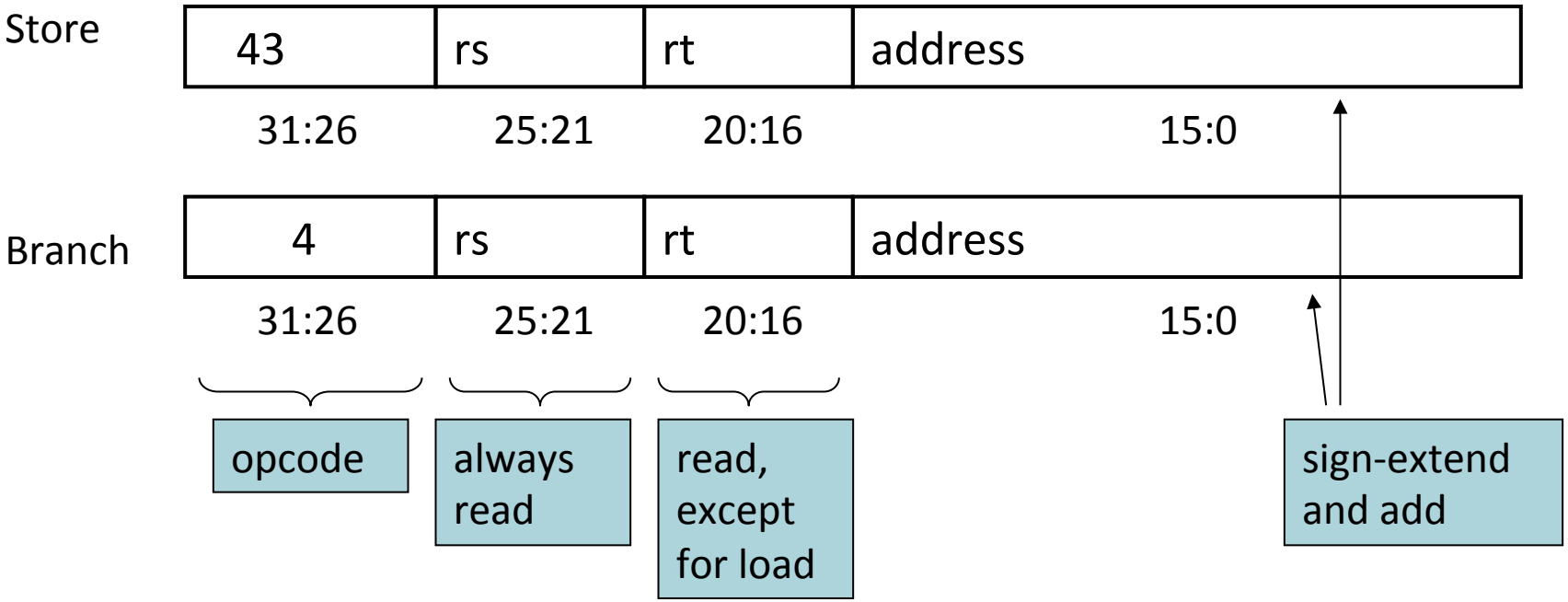
Combinational circuit to generate the 4-bit ALU control signal

Main Control Unit

- Control signals are derived from the instruction



Main Control Unit



Store copies content of rt register into memory, rt not changed

Branch subtracts rs from rt to generate zero flag, rs and rt not changed

Datapath with Control