1. **What does every instruction do in ALU?**
   (1) add: add (with overflow check)
   (2) addu: add
   (3) sub: sub (with overflow check)
   (4) subu: sub
   (5) and: and
   (6) or: or
   (7) xor: xor
   (8) nor: nor
   (9) slt: set less than
   (10) sltu: set less than (unsigned)
   (11) sll: shift left
   (12) srl: shift right
   (13) sra: shift right (arithmetic)
   (14) sllv: shift left
   (15) srlv: shift right
   (16) srav: shift right (arithmetic)
   (17) jr: do nothing.
   (18) addi: add (with overflow check)
   (19) addiu: add
   (20) andi: and
   (21) ori: or
   (22) xori: xor
   (23) lui: left shift
   (24) lw: add
   (25) sw: add
   (26) beq: sub
   (27) bne: sub
   (28) slti: set less than
   (29) sltiu: set less than (unsigned)
   (30) j: do nothing
   (31) jal: do nothing

Summary (ALU Control Output – ALU) [Table 1]

add (with overflow check)	add, addi	0000
add	addu, addiu, lw, sw	0001
sub (with overflow check)	sub	0010
sub	subu, beq, bne	0011
and	and, andi	0100
or	or, ori	0101
xor	xor, xori	0110
nor	nor	0111
set less than	slt, slti	1000
set less than (unsigned)	sltu, sltiu	1001
left shift	sll, sllv, lui	1010
right shift	srl, srlv	1011
right shift (arithmetic)	sra, srav	1100
NO USE	-	1101 & 1110
do nothing	j, jal, jr	1111

2. Controller Design: Output lines [Table 2]

Opcode (Explanation)	Reg Dst	ALU Src	Reg Write	Mem To Reg	Mem Read	Mem Write	BEQ	BNE	LUI	Ext Sign	Jal Sign	Jr	ALU Op	Jump
000000 (R-Type except Jr)	1	0	1	0	0	0	0	0	0	0	0	0	1101	0
000000 (Special Jr)	0	0	0	0	0	0	0	0	0	0	0	1	1111	0
001000 (addi)	0	1	1	0	0	0	0	0	0	1	0	0	0000	0
001001 (addiu)	0	1	1	0	0	0	0	0	0	0	0	0	0001	0
001100 (andi)	0	1	1	0	0	0	0	0	0	0	0	0	0100	0
001101 (ori)	0	1	1	0	0	0	0	0	0	0	0	0	0101	0
001110 (xori)	0	1	1	0	0	0	0	0	0	0	0	0	0110	0
001111 (lui)	0	1	1	0	0	0	0	0	1	0	0	0	1010	0
100011 (lw)	0	1	1	1	1	0	0	0	0	1	0	0	0001	0
101011 (sw)	0	1	0	0	0	1	0	0	0	1	0	0	0001	0
000100 (beq)	0	0	0	0	0	0	1	0	0	1	0	0	0011	0
000101 (bne)	0	0	0	0	0	0	0	1	0	1	0	0	0011	0
001010 (slti)	0	1	1	0	0	0	0	0	0	1	0	0	1000	0
001011 (sltiu)	0	1	1	0	0	0	0	0	0	0	0	0	1001	0
000010 (j)	0	0	0	0	0	0	0	0	0	0	0	0	1111	1
000011 (jal)	0	0	0	0	0	0	0	0	0	0	1	0	1111	1

Explanations: ALU Op has the same meanings as ALU Control Output in Table 1.

3. ALU Controller Logic Design (use ALUOp & funct)
(One more signals: ShamtSign (true in sll, srl, sra)

- If ALUOp is NO USE (1101/1110), then R-Type, get the corresponding output;
- If ALUOp is valid, then ALU Control Output = ALU Op.
- Get the **ShamtSign** according to the R-Type instructions.