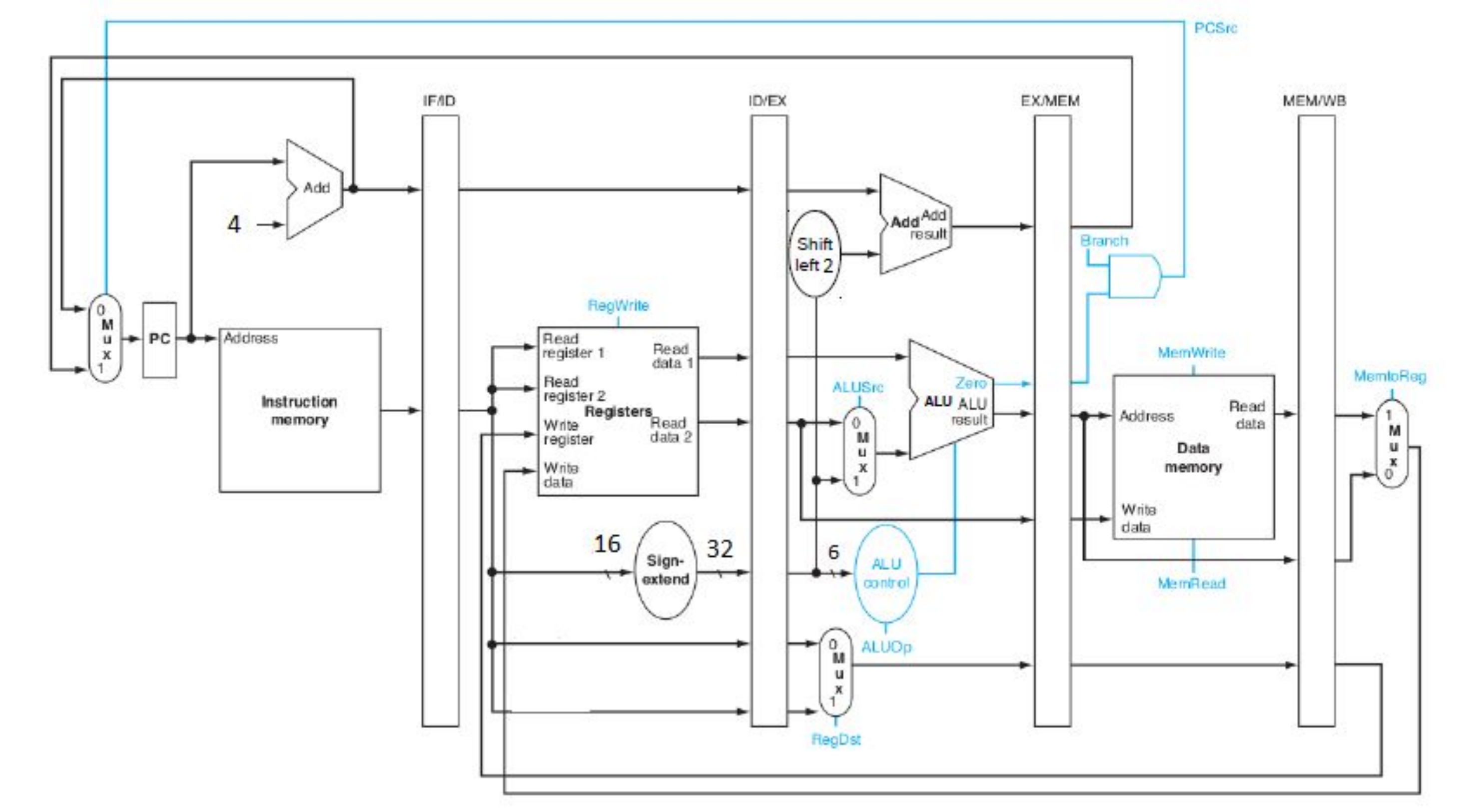
Execute
ID/EX

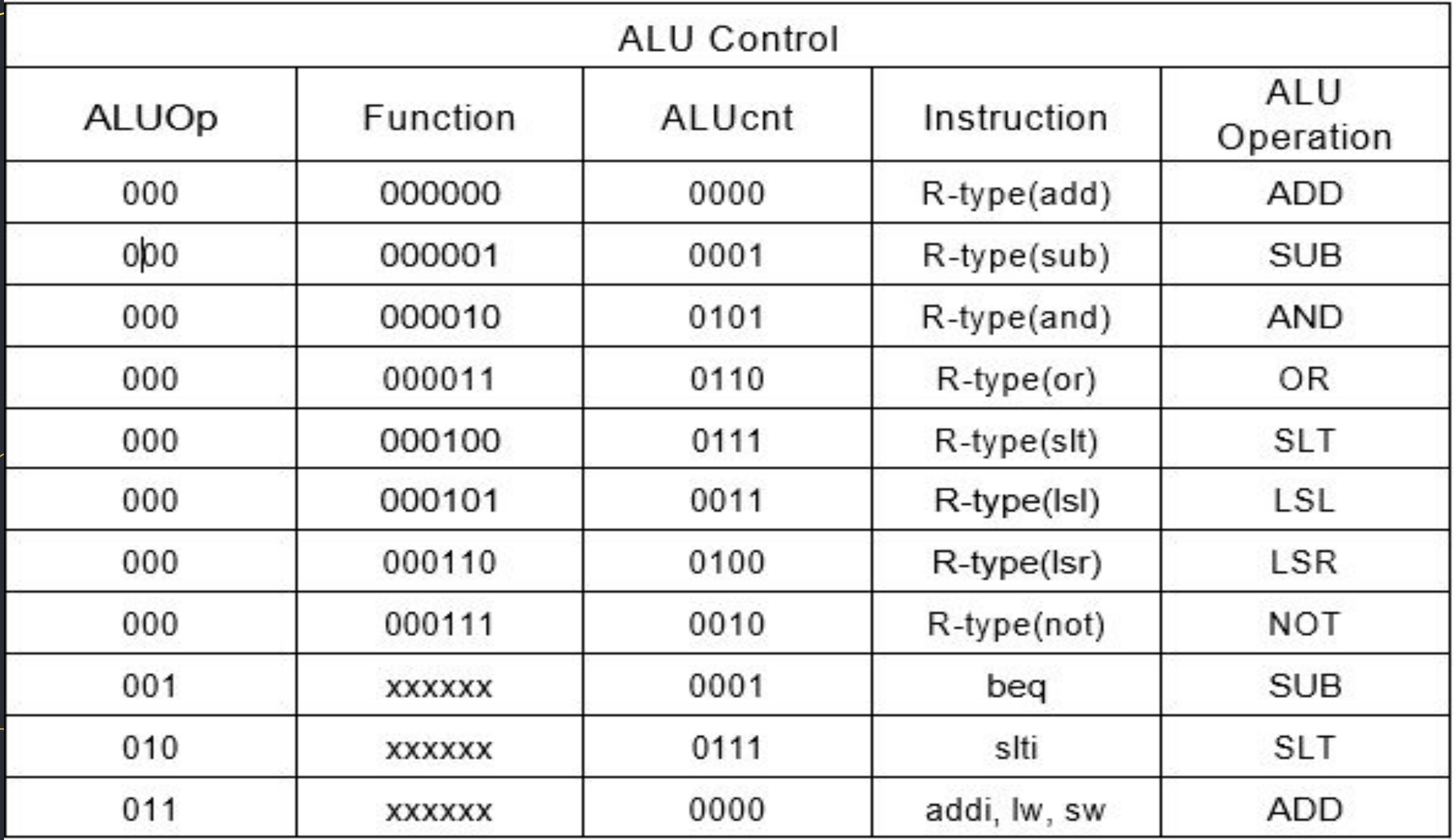
TABLE OF CONTENTS

1. **OVERVIEW OF MIPS PIPELINE**
   Have a quick look on the MIPS pipeline and focus on components we want to implement this week
2. **ALU CONTROL**
   Integrate ALUOp and Funct to produce ALU control signal to identify instructions
3. **IMPLEMENTATION OF ALU**
   Implementing the arithmetic unit using previously implemented modules
4. **INTEGRATING ALL MODULES (EXECUTE & ID/EX COME HERE)**
   Integrate all modules which have been implemented up to now
5. **INPUTS & OUTPUTS**
   Describe inputs and outputs of required modules

1. OVERVIEW OF MIPS PIPELINE



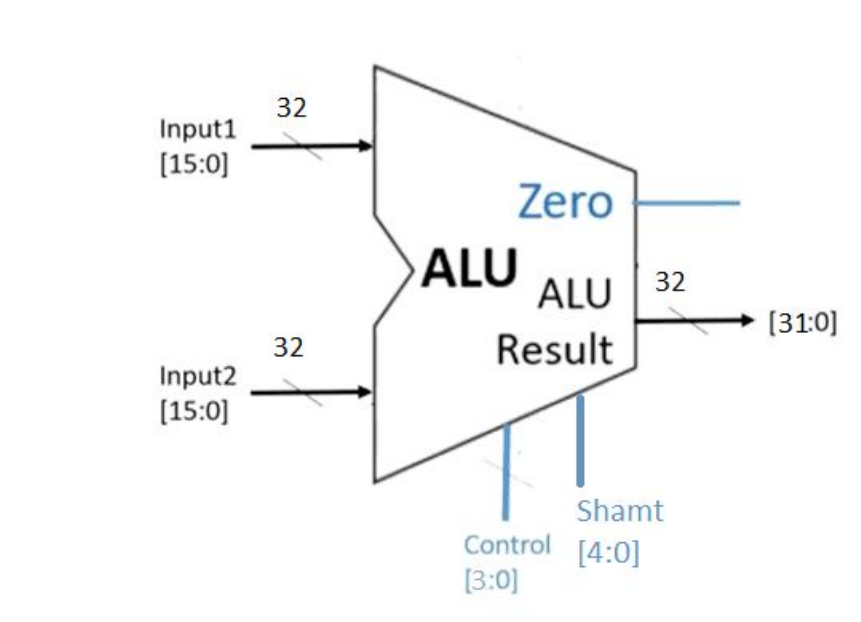
2. ALU CONTROL



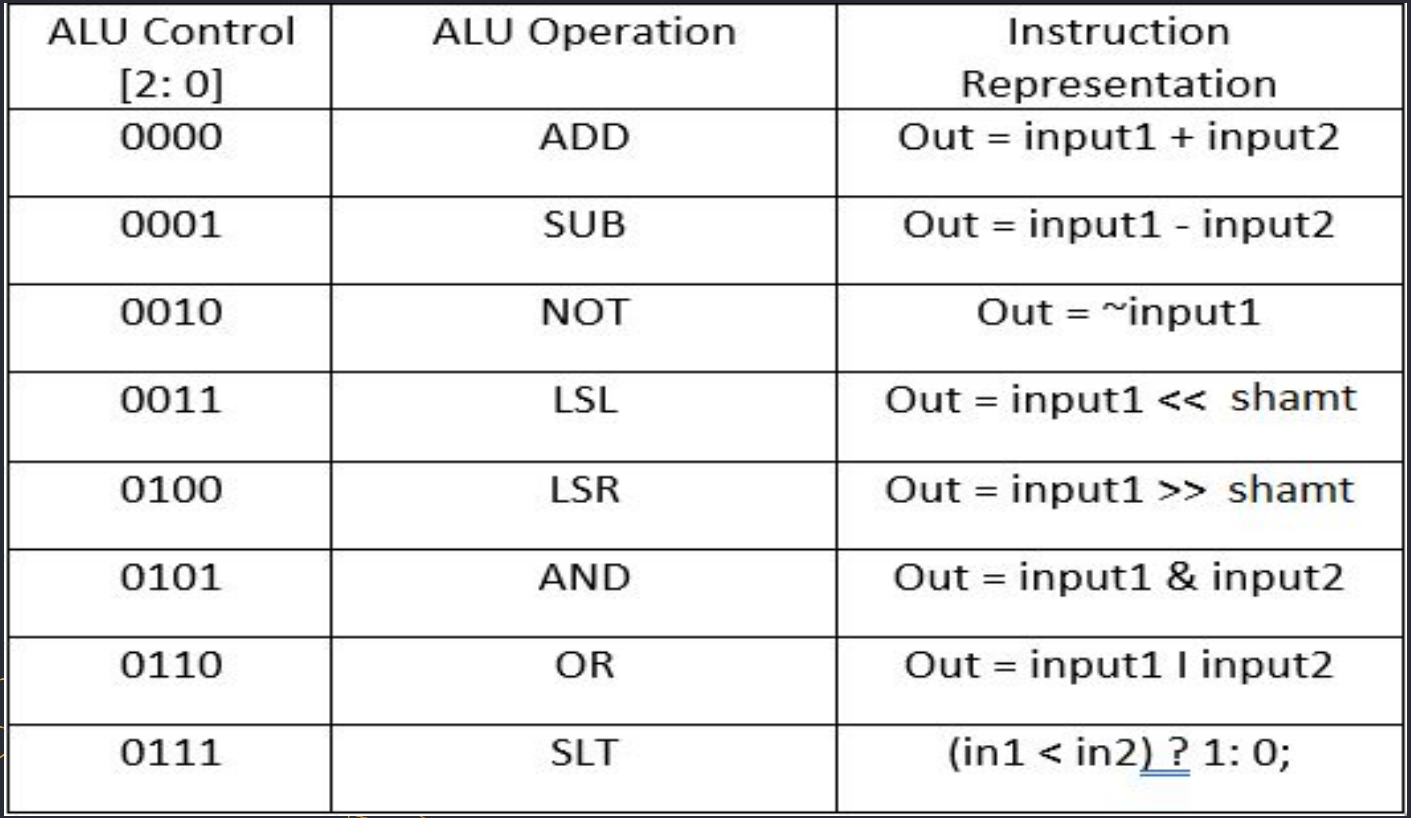
ALU Control				
ALUOp	Function	ALUcnt	Instruction	ALU Operation
000	000000	0000	R-type(add)	ADD
000	000001	0001	R-type(sub)	SUB
000	000010	0101	R-type(and)	AND
000	000011	0110	R-type(or)	OR
000	000100	0111	R-type(slt)	SLT
000	000101	0011	R-type(lsl)	LSL
000	000110	0100	R-type(lsr)	LSR
000	000111	0010	R-type(not)	NOT
001	xxxxxx	0001	beq	SUB
010	xxxxxx	0111	slti	SLT
011	xxxxxx	0000	addi, lw, sw	ADD

3. IMPLEMENTATION OF ALU

- Note that *shamt* can be extracted from the sign extended immediate you had in decode section.
- The 4-bit control signal is the ALU control signal that you get from the previous module (ALU Control).

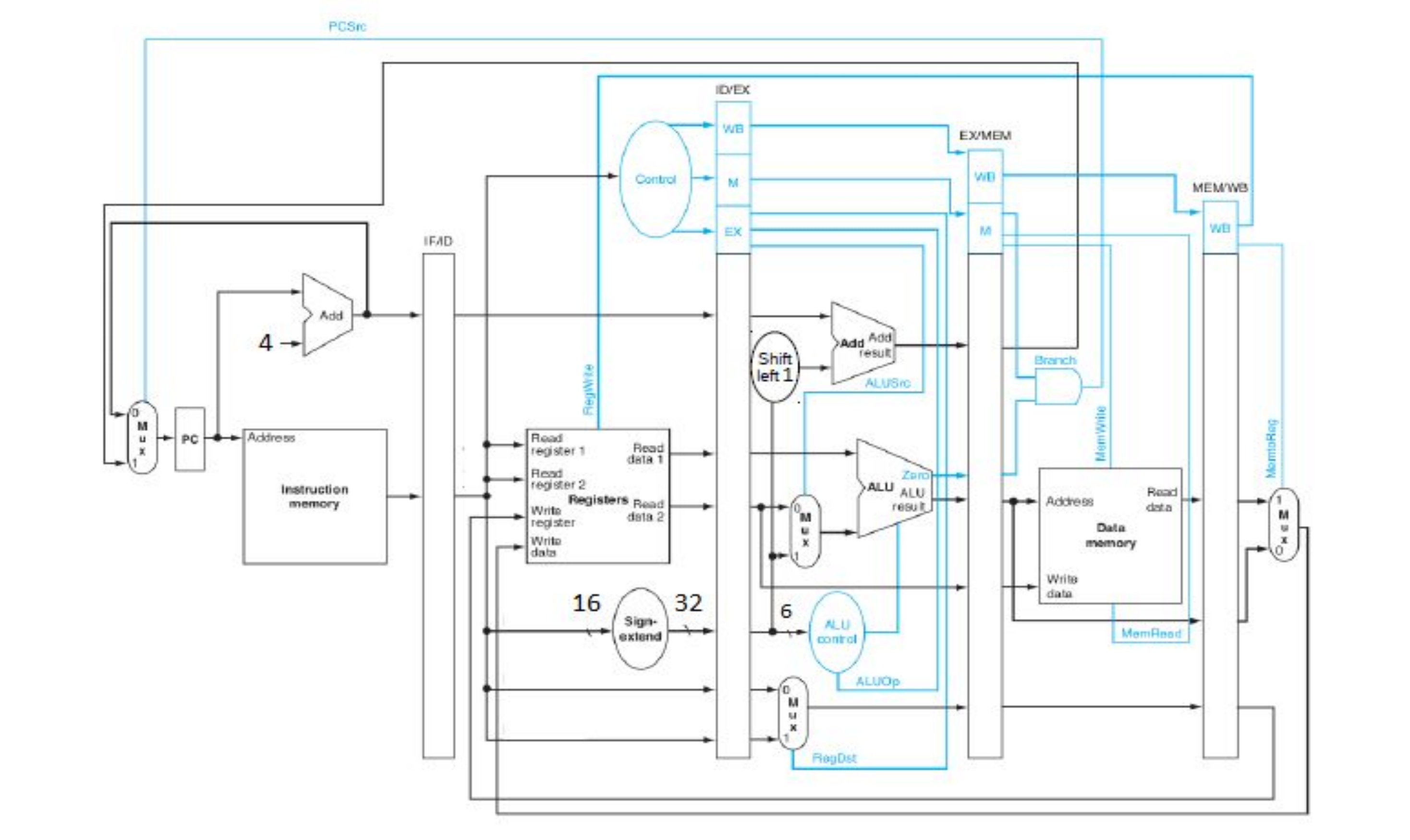


3. IMPLEMENTATION OF ALU (Cont'd)



ALU Control [2: 0]	ALU Operation	Instruction Representation
0000	ADD	Out = input1 + input2
0001	SUB	Out = input1 - input2
0010	NOT	Out = ~input1
0011	LSL	Out = input1 << shamt
0100	LSR	Out = input1 >> shamt
0101	AND	Out = input1 & input2
0110	OR	Out = input1 I input2
0111	SLT	(in1 < in2) ? 1: 0;

4. INTEGRATING ALL MODULES



5. INPUTS & OUTPUTS

Finally, it’s time to instantiate the modules and connect them together. Modules which need to be instantiated and connected are as follows:

ALU Control Unit:

- Inputs: ALUOp (3), funct (6)
- Outputs: ALUCnt (4)

ALU::

- Inputs: ALUCnt (4), input1 (32), input2 (32), shamt (5)
- Outputs: result (32), zero (1)

Execute:

- Inputs: clk (1), ALUReadData1 (32), ALUReadData2 (32), immediate (32), funct (6), ALUOp(3), ALUSrc (1)
- Outputs: ALUResult (32), zero (1) ,AddResult(32),ALUReadData2(32),RdOrRt(5)

5. INPUTS & OUTPUTS (Cont'd)

ID/EX Register:

- Inputs: clk (1), hit (1), readData1 (32), readData2 (32), signExImmediate (32), RegDst(1), ALUSrc (1), MemtoReg (1), RegWrite (1), MemRead(1), MemWrite(1), Branch(1), ALUOp(3), rt(5), rd(5), funct(6), nextPC(32),
- Outputs: readData1Out(32), readData2Out(32), signExImmediateOut(32), RegDstOut(1), ALUSrcOut(1), MemtoRegOut(1), RegWriteOut(1), MemReadOut(1), MemWriteOut(1), BranchOut(1), ALUOpOut(3), rtOut(5), rdOut(5), functOut(6), nextPCOut(32), [redacted]