BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, K.K. BIRLA GOA CAMPUS

II SEMESTER 2011-2012

Advanced Computer Organization (CS C342)

Lab 3

Figure given below shows the execution unit and control circuit design for MIPS architecture with add, sub, and, or, addi, lw, sw and beq instructions.

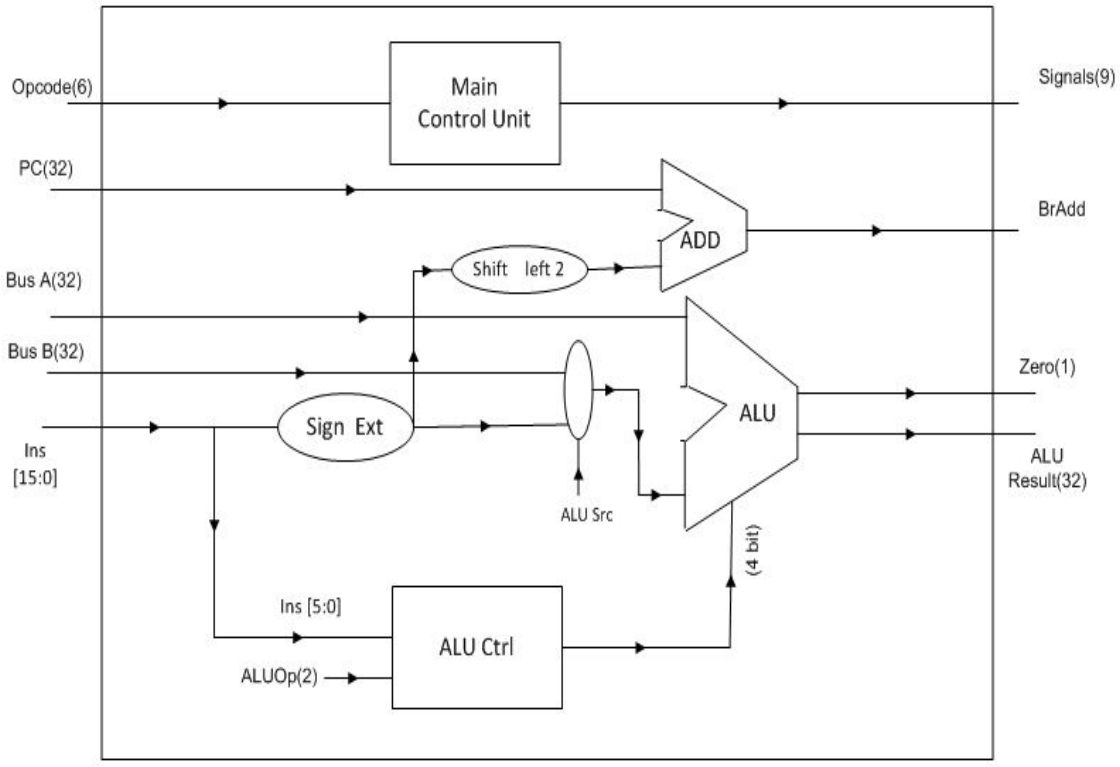

Figure 1

The Opcode and function field (only for R-type) for the instructions are as follows:

Instruction(s)	Opcode
ADD, SUB, AND, OR	000000
ADDI	001000
LW	100011
SW	101011
BEQ	000100

Instruction(s)	Ins[5:0]
ADD	100000
SUB	100010
AND	100100
OR	100101

The 9 bit signals you need to generate for the instructions are 1'b RegDst, 1'b RegWrite, 1'b ALUSrc, 1'b Branch, 1'b MemRd, 1'b MemWr, 1'b MemtoReg and 2'b ALUOp. Values of the signals for each of the instruction is as follows:

Instructions	RegDst	RegWrite	ALUSrc	Branch	MemRd	MemWr	MemtoReg	ALUOp
ADD, SUB, AND, OR	1	1	0	0	0	0	0	10
ADDI	0	1	1	0	0	0	0	00
LW	0	1	1	0	1	0	1	00
SW	0	0	1	0	0	1	0	00
BEQ	0	0	0	1	0	0	0	01

ALUOp	Ins [5:0]	ALU_Ctrl_Out
00	xxxxxx	0010
01	xxxxxx	0110
10	100000	0010
	100010	0110
	100100	0000
	100101	0001

ALU_Ctrl_Out	ALU Operation
0010	ADD
0110	SUBTRACT
0000	AND
0001	OR

Implement the design given in the figure 1 using the following modules:

1. module Main_Ctrl_Unit (Opcode_6b, Signals_9b);
   9'b Signal should be in this order [from MSB to LSB] →RegDst_1b , RegWrite_1b, ALUSrc_1b, Branch_1b, MemRd_1b, MemWr_1b, MemtoReg_1b, ALUOp_2b

2. module ALU_Ctrl_Unit (funct_6b, ALUOp_2b, ALU_Ctrl_Out_4b);

3. module Adder (PC_32b, SExtnShft_32b, BrAdd_32b);

4. module Mux (BusB_32b, SExtn_32, ALUSrc2_32b, ALUSrc_1b);

5. module ALU (BusA_32b, ALUSrc2_32b, ALU_Ctrl_Out_4b, ALUOut_32b, Zero_1b);

6. module Execution_Ctrl_Unit (Opcode_6b, PC_32b, BusA_32b, BusB_32b, Imm_16b, Signals_9b, BrAdd_32b, Zero_1b, ALUOut_32b);