**Lab Assignment 7**

Due: 17 Nov 2020

By: Alex Oswald

EECE 2323

**Verilog Code for pdatapath\_top\_lab7.v:**

1 `timescale 1ns / 1ps

2

3 **module** pdatapath\_top(

4 **input** wire clk,

5 **input** wire rst\_general,

6 **input** wire top\_pb\_clk,

7 **output**[7:0] led

8 );

9

10 wire pb\_clk\_debounced;

11

12 wire [7:0] alu\_1st\_input, alu\_2nd\_input;

13 wire [7:0] alu\_output;

14 wire [2:0] ALUOp;

15 wire alu\_ovf\_flag;

16 wire alu\_take\_branch\_output;

17

18 wire [15:0] instruction;

19 *//instruction fields*

20 wire [3:0] opcode;

21 wire [1:0] rs\_addr;

22 wire [1:0] rt\_addr;

23 wire [1:0] rd\_addr;

24 wire [7:0] immediate;

25

26 *//control signals*

27 wire RegDst;

28 wire RegWrite;

29 wire ALUSrc1;

30 wire ALUSrc2;

31 wire MemWrite;

32 wire MemToReg;

33

34 wire [1:0] regfile\_write\_address;*//destination register address*

35 wire [8:0] regfile\_write\_data;*//result data*

36 wire [8:0] read\_data1;*//source register1 data*

37 wire [8:0] read\_data2;*//source register2 data*

38

39 wire [8:0] alu\_result;

40 wire [7:0] zero\_register;

41 wire [8:0] data\_mem\_out;

42

43 wire [7:0] pc;

44

45 debounce debounce\_clk(

46 .clk\_in(clk),

47 .rst\_in(rst\_general),

48 .sig\_in(top\_pb\_clk),

49 .sig\_debounced\_out(pb\_clk\_debounced)

50 );

51

52

53 *// \*\*\*\*\*\*\*\*\*\*\*\*\* Add the PC logic here \*\*\*\*\*\*\*\*\*\*\*\*\* //*

54 pc\_logic pclog(

55 .clk(pb\_clk\_debounced),

56 .rst(rst\_general),

57 .count(pc));

58

59

60 *//\*\*\*\*\*\*\*\*\*\*Instantiate Your instruction memory here\*\*\*\*\*\*\*\*\*\*//*

61 instr\_mem instructionmem (

62 .a(pc), *// input wire [7 : 0] a*

63 .spo(instruction) *// output wire [15 : 0] spo*

64 );

65

66

67 *//\*\*\*\*\*\*\*\*\*\*Instantiate Your instruction decoder here\*\*\*\*\*\*\*\*\*\*//*

68 decoder decode (

69 .instruction(instruction),

70 .opcode(opcode),

71 .rs\_addr(rs\_addr),

72 .rt\_addr(rt\_addr),

73 .rd\_addr(rd\_addr),

74 .immediate(immediate),

75 .RegDst(RegDst),

76 .RegWrite(RegWrite),

77 .ALUSrc1(ALUSrc1),

78 .ALUSrc2(ALUSrc2),

79 .ALUOp(ALUOp),

80 .MemWrite(MemWrite),

81 .MemToReg(MemToReg)

82 );

83

84

85 *//\*\*\*\*\* random stuff lol \*\*\*\*\*//*

86 **assign** zero\_register = 7'b0;*//ZERO constant*

87 **assign** alu\_result = {alu\_ovf\_flag, alu\_output};

88 **assign** led = alu\_output;

89

90

91 *//\*\*\*\*\*\*\*\*\*\*Instantiate Your alu-regfile here\*\*\*\*\*\*\*\*\*\*//*

92 alu\_regfile ALU\_RegFile (

93 .ReadData1(read\_data1),

94 .ReadData2(read\_data2),

95 .ALUsrc1(ALUSrc1),

96 .ALUsrc2(ALUSrc2),

97 .ALUop(ALUOp),

98 .Instr\_i(immediate),

99 .zero\_register(zero\_register),

100

101 .rst(rst\_general),

102 .clk(pb\_clk\_debounced),

103 .wr\_en(RegWrite),

104 .rd0\_addr(rs\_addr),

105 .rd1\_addr(rt\_addr),

106 .wr\_addr(regfile\_write\_address),

107 .wr\_data(regfile\_write\_data),

108

109 .input\_1(alu\_1st\_input),

110 .input\_2(alu\_2nd\_input),

111 .result(alu\_output),

112 .ovf(alu\_ovf\_flag),

113 .take\_branch(alu\_take\_branch\_output)

114 );

115

116 *//\*\*\*\*\*\*\*\*\*\*Instantiate Your data memory here\*\*\*\*\*\*\*\*\*\*//*

117 data\_memory datamem (

118 .rst(rst\_general),

119 .clk(pb\_clk\_debounced),

120 .write\_enable(MemWrite),

121 .addr(alu\_output),

122 .write\_data(read\_data2),

123 .read\_data(data\_mem\_out)

124 );

125

126 *//\*\*\*\*\*\*\*\*\*\*Mux for regfile\_write\_data\*\*\*\*\*\*\*\*\*\*//*

127 **assign** regfile\_write\_data = MemToReg ? data\_mem\_out : alu\_result;

128

129 *//\*\*\*\*\*\*\*\*\*\*Mux for RegDST\*\*\*\*\*\*\*\*\*\*//*

130 **assign** regfile\_write\_address = RegDst ? rt\_addr : rd\_addr;

131

132 *//\*\*\*\*\*\*\*\*\*\* Instantiate the VIO here \*\*\*\*\*\*\*\*\*\*//*

133 vio\_0 vio (

134 .clk(clk), *// input wire clk*

135

136 .probe\_in0(alu\_output), *// [7:0] alu\_out*

137 .probe\_in1(alu\_ovf\_flag), *// [0:0] alu\_ovf*

138 .probe\_in2(alu\_take\_branch\_output), *// [0:0] take\_branch*

139 .probe\_in3(read\_data1), *// [7:0] ReadData1*

140 .probe\_in4(read\_data2), *// [7:0] ReadData2*

141 .probe\_in5(alu\_1st\_input), *// [7:0] input1*

142 .probe\_in6(alu\_2nd\_input), *// [7:0] input2*

143 .probe\_in7(regfile\_write\_data), *// [8:0] WriteData*

144 .probe\_in8(data\_mem\_out), *// [8:0] DataMemOut*

145 .probe\_in9(opcode), *// [3:0] opcode*

146 .probe\_in10(rs\_addr), *// [1:0] rs\_addr*

147 .probe\_in11(rt\_addr), *// [1:0] rt\_addr*

148 .probe\_in12(rd\_addr), *// [1:0] rd\_addr*

149 .probe\_in13(immediate), *// [7:0] immediate*

150 .probe\_in14(RegDst), *// [0:0] RegDst*

151 .probe\_in15(RegWrite), *// [0:0] RegWrite*

152 .probe\_in16(ALUSrc1), *// [0:0] ALUSrc1*

153 .probe\_in17(ALUSrc2), *// [0:0] ALUSrc2*

154 .probe\_in18(ALUOp), *// [2:0] ALUOp*

155 .probe\_in19(MemWrite), *// [0:0] MemWrite*

156 .probe\_in20(MemToReg), *// [0:0] MemToReg*

157 .probe\_in21(pc), *// [7:0] pc*

158 .probe\_in22(instruction) *// [15:0] instruction*

159 );

160

161 **endmodule**

**VIO Output Visualization:**

Evidence is located in attached **lab7-test.mov** file. The following sequence of instructions was followed via a mycode.coe coefficient file.

clr $0 (0xD000)

clr $1 (0xD140)

clr $2 (0xD280)

clr $3 (0xD3C0)

addi $1, $0, 0x10 (0x3110)

sw $0, 0x4($1) (0x1404)

addi $2, $0, 0x0F (0x320F)

sw $0, 0x5($2) (0x1805)

clr $0 (0xD000)

clr $1 (0xD140)

clr $2 (0xD280)

clr $3 (0xD3C0)

lw $1, 0x5($0) (0x0105)

lw $2, 0x4($0) (0x0204)

add $3, $1, $2 (0x26C0)

sw $0, 0x11($3) (0x1C11)

clr $0 (0xD000)

clr $1 (0xD140)

clr $2 (0xD280)

clr $3 (0xD3C0)

lw $1, 0x5($0) (0x0105)

inv $2, $1 (0x4180)

addi $3, $2, 0x01 (0x3B01)

sw $0, 0x12($3) (0x1C12)

clr $0 (0xD000)

clr $1 (0xD140)

clr $2 (0xD280)

clr $3 (0xD3C0)

lw $1, 0x4($0) (0x0104)

lw $2, 0x12($0) (0x0212)

inv $1, $1 (0x4140)

addi $1, $1, 0x1 (0x3501)

add $3, $1, $2 (0x26C0)

sw $0, 0x13($0) (0x1013)