Digital Logic Design & Computer Architecture Lab report

Second assignment

PREPARED FOR

Miss. Talebpour

PREPARED BY

Mohsen Karbalaei Amini, 98242128

Mohammad Mehdi Parchami, 400243084

March 4, 2023,

12-bit ALU

Implementation (Part a)

We constructed a 12-bit ALU making use of *behavioral* coding style and case statement to define what each op code does. We also used concatenating to generate CarryOut bit.

Here you can see the codes:

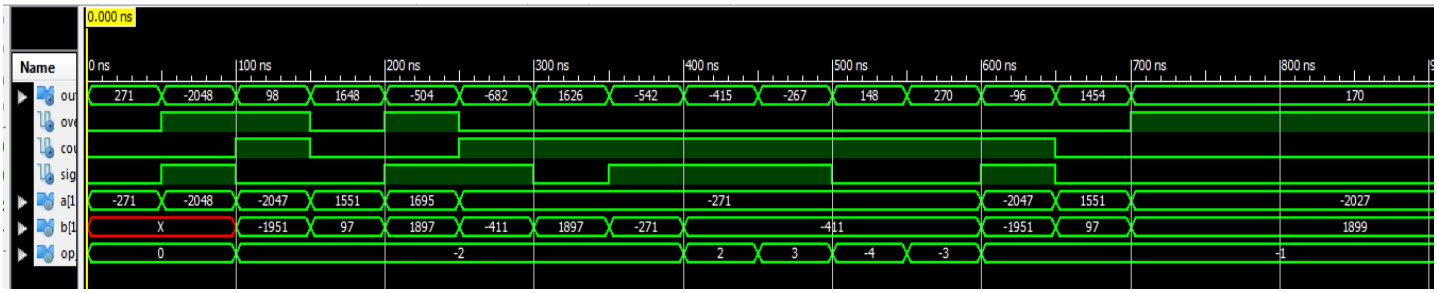
```
module alu(
           input [11:0] A,B,  // Inputs
           input [2:0] OP,// operation Select
           output reg [11:0] Z, // Output
           output reg CarryOut, // Carry Out
           output reg Sign, // sign bit
           output reg OV // Overflow flag
    );

    always @(*) begin
        case(OP)
            3'd0: if (A[11] == 1'b1) begin
                        Z = -A;
                        CarryOut= 1'b0;
                        OV= 1'b0;
                        if (A == 12'b1000_0000_0000)
                                OV = 1'b1;
                  end
                  else  begin
                        CarryOut= 1'b0;
                        OV= 1'b0;
                        Z = A;
                  end
            3'd1: Z = B << 1;
            3'd2: begin Z = A & B; OV = 1'b0; end
            3'd3: begin Z = A | B; OV = 1'b0; end
            3'd4: begin Z = A ^ B; OV = 1'b0; end
            3'd5: begin Z = ~A; OV = 1'b0; end
            3'd6: begin
                    {CarryOut,Z} = ({1'b0,A} + {1'b0,B});
                    OV = (A[11] & B[11] & (~Z[11])) | ((~A[11]) & (~B[11]) & Z[11]);
                  end

            3'd7: begin
                    {CarryOut,Z}= {1'b0,A} - {1'b0,B};
                    OV = (A[11] & (~B[11]) & (~Z[11])) | ((~A[11]) & (B[11]) & Z[11]);
                  end
        endcase
        Sign = Z[11];
    end
endmodule
```

Test Bench (part b)

Code and simulation:

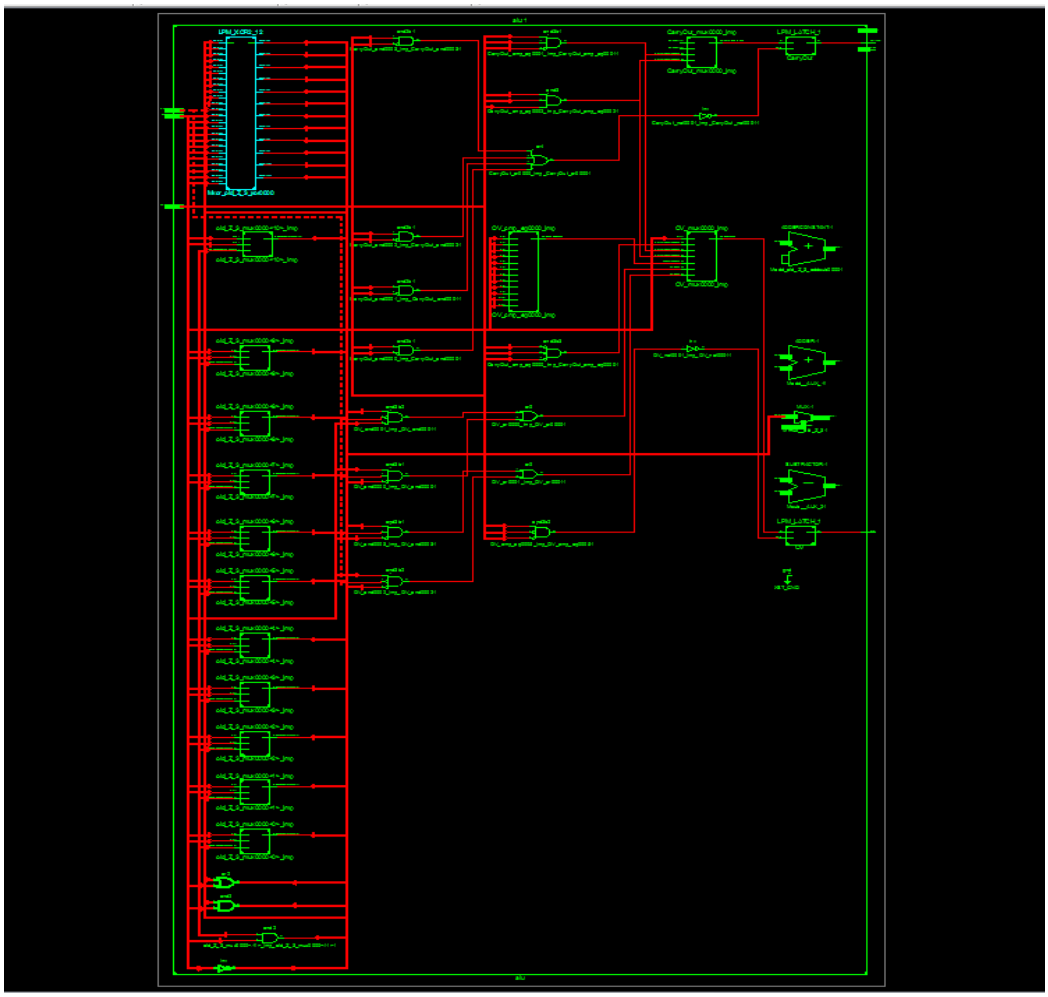
```
26  module alu_tb;
27
28
29          reg [11:0] a, b;
30          reg [2:0] op_select;
31          wire [11:0] out;
32          wire overflow, cout, sign;
33
34          alu alu(a, b, op_select, out, cout, sign, overflow);
35
36          initial begin
37      // abs
38      op_select = 3'd0;
39      a = 12'b1110_1111_0001;
40      #50;
41      // abs
42      op_select = 3'd0;
43      a = 12'b1000_0000_0000;
44      #50;
45      //with overflow negetive numbers
46      op_select = 3'd6;
47      a = 12'b1000_0000_0001;
48      b = 12'b1000_0110_0001;
49      #50;
50      //without overflow posetive numbers
51      op_select = 3'd6;
52      a = 12'b0110_0000_1111;
53      b = 12'b0000_0110_0001;
54      #50;
55      //with overflow posetive numbers
56      op_select = 3'd6;
57      a = 12'b0110_1001_1111;
58      b = 12'b0111_0110_1001;
59      #50;
```

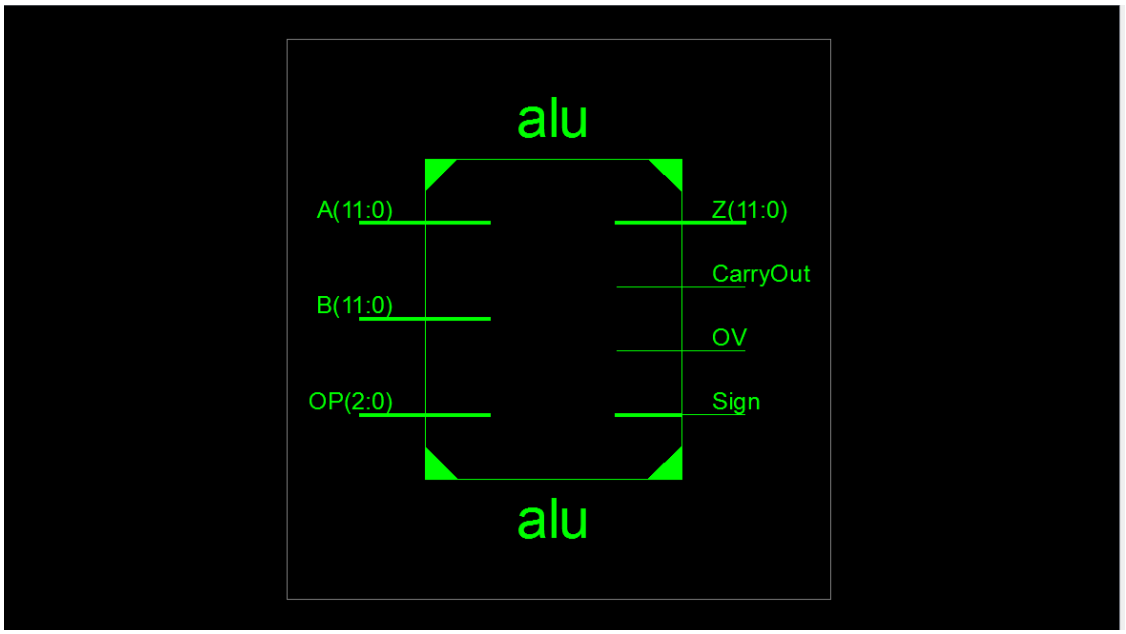
```
60        //without overflow negetive numbers
61        op_select = 3'd6;
62        a = 12'b1110_1111_0001;
63        b = 12'b1110_0110_0101;
64        #50;
65        // postive nad negetive numbers
66        op_select = 3'd6;
67        a = 12'b1110_1111_0001;
68        b = 12'b0111_0110_1001;
69        #50;
70        //left shift
71        b = 12'b1110_1111_0001;
72        #50;
73        // and
74        op_select = 3'd2;
75        a = 12'b1110_1111_0001;
76        b = 12'b1110_0110_0101;
77        #50;
78        //or
79        op_select = 3'd3;
80        #50;
81        //xor
82        op_select = 3'd4;
83        #50;
84        //not
85        op_select = 3'd5;
86        #50;
87        //sub without overflow negetive numbers
88        op_select = 3'd7;
89        a = 12'b1000_0000_0001;
90        b = 12'b1000_0110_0001;
91        #50;
92      //sub without overflow posetive numbers
93      op_select = 3'd7;
94      a = 12'b0110_0000_1111;
95      b = 12'b0000_0110_0001;
96      #50;
97      //sub postive nad negetive numbers with overflow
98      op_select = 3'd7;
99      a = 12'b1000_0001_0101;
100     b = 12'b0111_0110_1011;
101   end
102
103
104   endmodule
```



Synthesize (part c)

Here's the technology and the RTL schematic view.





Critical Path Delay Analysis (part e)

Based on the post-PAR static timing report, the critical path in this Circuit is the way from ***A[1] to Z[3]***, with a delay of 11.780 ns.

Here's a bit summary of the most delayed paths.

Source Pad	Destination Pad	Delay
A<0>	Sign	11.153
A<0>	Z<0>	10.877
A<0>	Z<1>	10.680
A<0>	Z<2>	10.988
A<0>	Z<3>	11.325
A<0>	Z<4>	10.315
...		
A<1>	Z<3>	11.780

Design Summary (part f)

alu Project Status (03/16/2023 - 06:58:14)			
Project File:	dfgdf.xise	**Parser Errors:**	No Errors
Module Name:	alu	**Implementation State:**	Placed and Routed
Target Device:	xc3s100e-5tq144	**•Errors:**	No Errors
Product Version:	ISE 14.7	**•Warnings:**	5 Warnings (0 new)
Design Goal:	Balanced	**•Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	All Constraints Met
Environment:	System Settings	**•Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	106	1,920	5%	
Number of occupied Slices	56	960	5%	
Number of Slices containing only related logic	56	56	100%	
Number of Slices containing unrelated logic	0	56	0%	
Total Number of 4 input LUTs	107	1,920	5%	
Number used as logic	106			
Number used as a route-thru	1			
Number of bonded IOBs	42	108	38%	
IOB Latches	2			
Average Fanout of Non-Clock Nets	2.76			

Performance Summary				[-]
Final Timing Score:	0 (Setup: 0, Hold: 0)	**Pinout Data:**	Pinout Report	
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report	
Timing Constraints:	All Constraints Met			

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Thu Mar 16 06:57:55 2023	0	3 Warnings (0 new)	3 Infos (0 new)
Translation Report	Current	Thu Mar 16 06:58:05 2023	0	0	0
Map Report	Current	Thu Mar 16 06:58:08 2023	0	2 Warnings (0 new)	2 Infos (0 new)
Place and Route Report	Current	Thu Mar 16 06:58:12 2023	0	0	1 Info (0 new)
Power Report					
Post-PAR Static Timing Report	Current	Thu Mar 16 06:58:14 2023	0	0	6 Infos (0 new)
Bitgen Report					