

CA Lab: Homework 9

student: Dimitri Tabatadze

May 26, 2023

Task Description

Immediate Extension Unit

Solution

The module:

```
1 module ALU (  
2     input i,  
3     input [31:0] SrcA, SrcB,  
4     input [3:0] af,  
5     output reg [31:0] Alures,  
6     output Zero, Neg, ovfalu  
7 );  
8     always @(af or i or SrcA or SrcB) begin  
9         case (af)  
10            4'b0000 : Alures = SrcA + SrcB ;  
11            4'b0001 : Alures = SrcA + SrcB ;  
12            4'b0010 : Alures = SrcA - SrcB ;  
13            4'b0011 : Alures = SrcA - SrcB ;  
14            4'b0100 : Alures = SrcA & SrcB ;  
15            4'b0101 : Alures = SrcA | SrcB ;  
16            4'b0110 : Alures = SrcA ^ SrcB ;  
17            4'b0111 : Alures = i ? SrcB << 16 : SrcA ~| SrcB;  
18            4'b1010 : Alures = SrcA < SrcB ;  
19            4'b1011 : Alures = SrcA < SrcB ;  
20        endcase  
21    end  
22    assign Zero = Alures == 0;  
23    assign Neg = (af < 4) & Alures[31];  
24    assign ovfalu = (af == 0 | af == 2) & (Alures[31] ? (SrcA[31] &  
25        SrcB[31]) : (SrcA[31] | SrcB[31]));  
endmodule
```

The testbench:

```
1 module alu_tb();  
2  
3     reg i;  
4     reg [31:0] SrcA;  
5     reg [31:0] SrcB;
```

```

6  reg [3:0] af;
7  wire [31:0] Alures;
8  wire Zero;
9  wire Neg;
10 wire ovfalu;
11
12 ALU UUT (
13     .i(i),
14     .SrcA(SrcA),
15     .SrcB(SrcB),
16     .af(af),
17     .Alures(Alures),
18     .Zero(Zero),
19     .Neg(Neg),
20     .ovfalu(ovfalu)
21 );
22
23 initial begin
24     $dumpfile("alu-tb.vcd");
25     $dumpvars(0, alu_tb);
26     SrcA = 32'd3;
27     SrcB = 32'd4;
28     i = 1;
29     af = 4'b0000;
30     #10;
31     af = 4'b0001;
32     #10;
33     af = 4'b0010;
34     #10;
35     af = 4'b0011;
36     #10;
37     af = 4'b0100;
38     #10;
39     af = 4'b0101;
40     #10;
41     af = 4'b0110;
42     #10;
43     af = 4'b0111;
44     #10;
45     af = 4'b1010;
46     #10;
47     af = 4'b1011;
48     #10;
49     i = 0;
50     af = 4'b0000;
51     #10;
52     af = 4'b0001;
53     #10;
54     af = 4'b0010;
55     #10;
56     af = 4'b0011;
57     #10;
58     af = 4'b0100;
59     #10;
60     af = 4'b0101;
61     #10;
62     af = 4'b0110;

```

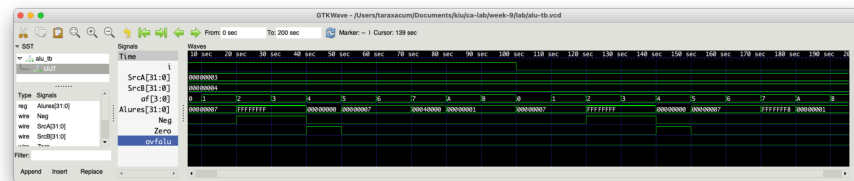
```

63     #10;
64     af = 4'b0111;
65     #10;
66     af = 4'b1010;
67     #10;
68     af = 4'b1011;
69     #10;
70     end
71 endmodule

```

Simulation and Testing

The timing diagrams:



Reference

- me