

```

1  module UnidadedeControleULA (Funct, AluOp, ControleALU, JALR, JR);
2
3
4  input [5:0] Funct;
5  input [2:0] AluOp;
6  output [3:0] ControleALU;
7  output JALR, JR;
8
9  reg[3:0] RegControle;
10 reg RegJALR, RegJR;
11
12 always @(*)
13 begin
14     case (AluOp)
15     3'b000: begin
16         case (Funct)
17         6'b000001: begin // ADD
18             RegControle <= 4'b0010;
19             RegJALR <= 0;
20             RegJR <= 0;
21         end
22         6'b000010: begin // SUB
23             RegControle <= 4'b0011;
24             RegJALR <= 0;
25             RegJR <= 0;
26         end
27         6'b000011: begin // DIV
28             RegControle <= 4'b1001;
29             RegJALR <= 0;
30             RegJR <= 0;
31         end
32         6'b000100: begin // MULT
33             RegControle <= 4'b1000;
34             RegJALR <= 0;
35             RegJR <= 0;
36         end
37         6'b000101: begin // AND
38             RegControle <= 4'b0000;
39             RegJALR <= 0;
40             RegJR <= 0;
41         end
42         6'b000110: begin // OR
43             RegControle <= 4'b0001;
44             RegJALR <= 0;
45             RegJR <= 0;
46         end
47         // 6'b000111: begin // NOT
48         //     RegControle = 4'b0000;
49         // end
50         6'b000111: begin // NOR
51             RegControle <= 4'b1010;
52             RegJALR <= 0;
53             RegJR <= 0;
54         end
55         6'b001000: begin // SLL
56             RegControle <= 4'b1011;
57             RegJALR <= 0;
58             RegJR <= 0;
59         end
60         6'b001001: begin // SRL
61             RegControle <= 4'b1100;
62             RegJALR <= 0;
63             RegJR <= 0;
64         end
65         6'b001010: begin // JR
66             RegControle <= 4'b0010; // ADD ???
67             RegJALR <= 0;
68             RegJR <= 1;
69         end
70         6'b001011: begin // JALR
71             RegControle <= 4'b0000; // AND ???
72             RegJALR <= 1;
73             RegJR <= 0;
74         end
75         6'b001100: begin // SLT
76             RegControle <= 4'b0100;
77             RegJALR <= 0;

```

```
78         RegJR <= 0;
79     end
80     6'b001101: begin // SLET
81         RegControle <= 4'b0111;
82         RegJALR <= 0;
83         RegJR <= 0;
84     end
85     6'b001110: begin // SGT
86         RegControle <= 4'b0101;
87         RegJALR <= 0;
88         RegJR <= 0;
89     end
90     6'b001111: begin // SGET
91         RegControle <= 4'b0110;
92         RegJALR <= 0;
93         RegJR <= 0;
94     end
95 endcase
96 end
97 3b'001: begin // SOMA
98     RegControle <= 4'b0010;
99     RegJALR <= 0;
100    RegJR <= 0;
101 end
102 3b'010: begin // SUB
103     RegControle <= 4'b0011;
104     RegJALR <= 0;
105     RegJR <= 0;
106 end
107 3b'011: begin // AND
108     RegControle <= 4'b0010;
109     RegJALR <= 0;
110     RegJR <= 0;
111 end
112 3b'100: begin // OR
113     RegControle <= 4'b0001;
114     RegJALR <= 0;
115     RegJR <= 0;
116 end
117 3b'101: begin // SLT
118     RegControle <= 4'b0100;
119     RegJALR <= 0;
120     RegJR <= 0;
121 end
122 endcase
123 end
124
125 assign ControleALU = RegControle;
126 assign JALR = RegJALR;
127 assign JR = RegJR;
128
129
130 endmodule
131
```