```
1   /*
2      conditional_ff_logic assesses whether a branching condition has been met, indicating
    instruction status to the control unit.
3      It accepts IR[20:19], referring to the Rb segment of the IR containing the branch
    condition to evaluate.
4      The decoded condition is applied to the current contents of the bus: if CONin is high,
    CON will capt the result of the comparison.
5   */
6
7
8   module conditional_ff_logic  (
9      input [1:0] IR_20_19,
10     input [31:0] BusMuxOut, // Out refers to the perspective of the Bus.
11     input CONin, clk,
12     output reg CON
13     );
14
15     // Assesses whether the current Bus contents are equal to zero (reduction NOR)
16     wire bus_zero = ~| BusMuxOut;
17     // Assesses whether the current Bus contents are negative.
18     wire bus_neg = BusMuxOut[31];
19
20     // Pairing the condition to the bus contents
21     wire CON_D;
22     assign CON_D = (IR_20_19 == 2'b00) && bus_zero || (IR_20_19 == 2'b01) && !bus_zero
23        || (IR_20_19 == 2'b10) && !bus_neg || (IR_20_19 == 2'b11) && bus_neg;
24
25     // Recording the result of the condition in CON if CONin is high.
26     initial CON <= 1'b0;
27     always @(posedge clk) if (CONin) CON <= CON_D;
28
29  endmodule
30
31
32  // Testbench
33  module con_ff_tb();
34     reg [1:0] IR_20_19;
35     reg [31:0] BusMuxOut;
36     reg CONin, clk;
37     wire CON;
38
39     // Design Under Test
40     conditional_ff_logic DUT (IR_20_19, BusMuxOut, CONin, clk, CON);
41
42     // Establishing clock behaviour.
43     parameter clock_period = 20;
44     initial begin
45        clk <= 0;
46        forever #(clock_period/2) clk <= ~clk;
47     end
48
49      initial begin
50          // Branch if 0
51          IR_20_19 <= 2'b00; BusMuxOut <= 32'b0; CONin <= 1'b1; @(posedge clk); @(posedge clk);
52          if (CON !== 1) $display("Failure: IR_20_19=%b, BusMuxOut=%h, Expected CON=1, Got
    CON=%b", IR_20_19, BusMuxOut, CON);
53
54          IR_20_19 <= 2'b00; BusMuxOut <= 32'b1; CONin <= 1'b1; @(posedge clk); @(posedge clk);
55          if (CON !== 0) $display("Failure: IR_20_19=%b, BusMuxOut=%h, Expected CON=0, Got
    CON=%b", IR_20_19, BusMuxOut, CON);
56
57          // Branch if not 0
58          IR_20_19 <= 2'b01; BusMuxOut <= 32'b1; CONin <= 1'b1; @(posedge clk); @(posedge clk);
59          if (CON !== 1) $display("Failure: IR_20_19=%b, BusMuxOut=%h, Expected CON=1, Got
    CON=%b", IR_20_19, BusMuxOut, CON);
60
61          IR_20_19 <= 2'b01; BusMuxOut <= 32'b0; CONin <= 1'b1; @(posedge clk); @(posedge clk);
62          if (CON !== 0) $display("Failure: IR_20_19=%b, BusMuxOut=%h, Expected CON=0, Got
    CON=%b", IR_20_19, BusMuxOut, CON);
63
64          // Branch if pos
65          IR_20_19 <= 2'b10; BusMuxOut <= 32'b1; CONin <= 1'b1; @(posedge clk); @(posedge clk);
66          if (CON !== 1) $display("Failure: IR_20_19=%b, BusMuxOut=%h, Expected CON=1, Got
    CON=%b", IR_20_19, BusMuxOut, CON);
67
68          IR_20_19 <= 2'b10; BusMuxOut <= 32'hFFFF0000; CONin <= 1'b1; @(posedge clk); @(
```

```
    posedge clk);
69          if (CON !== 0) $display("Failure: IR_20_19=%b, BusMuxOut=%h, Expected CON=0, Got
    CON=%b", IR_20_19, BusMuxOut, CON);
70
71          // Branch if neg
72          IR_20_19 <= 2'b11; BusMuxOut <= 32'hFFFF0000; CONin <= 1'b1; @(posedge clk); @(
    posedge clk);
73          if (CON !== 1) $display("Failure: IR_20_19=%b, BusMuxOut=%h, Expected CON=1, Got
    CON=%b", IR_20_19, BusMuxOut, CON);
74
75          IR_20_19 <= 2'b11; BusMuxOut <= 32'b1; CONin <= 1'b1; @(posedge clk); @(posedge clk);
76          if (CON !== 0) $display("Failure: IR_20_19=%b, BusMuxOut=%h, Expected CON=0, Got
    CON=%b", IR_20_19, BusMuxOut, CON);
77
78          $display("Testbench completed successfully" );
79          @(posedge clk)
80          $stop;
81      end
82  endmodule
83
```