```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity contador6Bits is
5     port ( clk_c6B,ld_c6B,clr_c6B: in std_logic;
6            out_c6B: out std_logic_vector(5 downto 0));
7   end contador6Bits ;
8   architecture ckt of contador6Bits is
9      component ffjk is
10        port (clk ,J ,K ,P ,C: in std_logic;
11              q: out std_logic);
12     end component;
13
14     signal q0_out,q1_out,q2_out,q3_out,q4_out,q5_out, clr_reverso: std_logic;
15     signal resp_and: std_logic_vector(4 downto 0);
16
17     begin
18
19        clr_reverso <= not clr_c6B;
20        Q0: ffjk port map(
21             clk => clk_c6B,
22             J => ld_c6B,
23             K => ld_c6B,
24             P => '1',
25             C => clr_reverso,
26             q => q0_out);
27
28        resp_and(0) <= q0_out and ld_c6B;
29
30        Q1: ffjk port map(
31             clk => clk_c6B,
32             J => resp_and(0),
33             K => resp_and(0),
34             P => '1',
35             C => clr_reverso,
36             q => q1_out);
37
38        resp_and(1) <= q1_out and resp_and(0);
39
40        Q2: ffjk port map(
41             clk => clk_c6B,
42             J => resp_and(1),
43             K => resp_and(1),
44             P => '1',
45             C => clr_reverso,
46             q => q2_out);
47
48        resp_and(2) <= q2_out and resp_and(1);
49
50        Q3: ffjk port map(
51             clk => clk_c6B,
52             J => resp_and(2),
53             K => resp_and(2),
54             P => '1',
55             C => clr_reverso,
56             q => q3_out);
57
58        resp_and(3) <= q3_out and resp_and(2);
59
60        Q4: ffjk port map(
61             clk => clk_c6B,
62             J => resp_and(3),
63             K => resp_and(3),
64             P => '1',
65             C => clr_reverso,
66             q => q4_out);
```

```
67
68          resp_and(4) <= q4_out and resp_and(3);
69
70          Q5: ffjk port map(
71               clk => clk_c6B,
72               J => resp_and(4),
73               K => resp_and(4),
74               P => '1',
75               C => clr_reverso,
76               q => q5_out);
77
78          out_c6B(0) <= q0_out;
79          out_c6B(1) <= q1_out;
80          out_c6B(2) <= q2_out;
81          out_c6B(3) <= q3_out;
82          out_c6B(4) <= q4_out;
83          out_c6B(5) <= q5_out;
84
85
86    end ckt ;
```

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   ENTITY ffjk IS
5     port ( clk ,J ,K ,P ,C : in std_logic;
6              q : out std_logic );
7   END ffjk ;
8   ARCHITECTURE ckt OF ffjk IS
9   SIGNAL qS : std_logic;
10  BEGIN
11    PROCESS ( clk ,P ,C )
12    BEGIN
13      IF P = '0' THEN qS <= '1';
14      ELSIF C ='0' THEN qS <= '0';
15      ELSIF clk = '1' AND clk ' EVENT THEN
16        IF J = '1' AND K = '1' THEN qS <= NOT qS ;
17        ELSIF J = '1' AND K = '0' THEN qS <= '1';
18        ELSIF J = '0' AND K = '1' THEN qS <= '0';
19        END IF;
20      END IF;
21    END PROCESS ;
22  q <= qS ;
23  END ckt ;
```