```
1  ----------------------------------------------------------------------------
2  -- Company: DAMAGE Inc.
3  -- Engineer: MYXATA ;)
4  --
5  -- Create Date:    14:01:20 08/24/2007
6  -- Design Name: Rotary Encoder
7  -- Module Name:    rot_enc - Behavioral
8  -- Project Name: Rotary Encoder
9  -- Target Devices: Any CPLD (xilinx preffered)
10 -- Tool versions: whatever
11 -- Description: simple rotary encoder
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 --
18 ----------------------------------------------------------------------------
19 library IEEE;
20 use IEEE.STD_LOGIC_1164.ALL;
21 use IEEE.STD_LOGIC_ARITH.ALL;
22 use IEEE.STD_LOGIC_UNSIGNED.ALL;
23
24 entity rot_enc is
25     Port ( IN_A : in  STD_LOGIC;
26            IN_B : in  STD_LOGIC;
27            CLK : in  STD_LOGIC;--Podberi si podhodiasht CLK
28            ROT_DIR : out  STD_LOGIC;
29            ROT_IRQ : out  STD_LOGIC);
30 end rot_enc;
31
32 architecture Behavioral of rot_enc is
33 signal rotary_in : std_logic_vector(1 downto 0);
34 signal rot_fil_A : std_logic;
35 signal rot_fil_B : std_logic;
36 signal rot_fil_A_delayed: std_logic;
37 signal rot_direction : std_logic;
38 begin
39 --====================ROTARY FILTER=======================================--
40 -- za da se izbegne slu4aino srabotwane ot kratkovremenni impulsi
41 --========================================================================--
42 rotary_filter: process(clk)
43    begin
44       if clk'event and clk='1' then
45          rotary_in <= IN_A & IN_B;
46          case rotary_in is
47             when "00" => rot_fil_A <= '0'; rot_fil_B <= rot_fil_B;
48             when "01" => rot_fil_A <= rot_fil_A; rot_fil_B <= '0';
49             when "10" => rot_fil_A <= rot_fil_A; rot_fil_B <= '1';
50             when "11" => rot_fil_A <= '1'; rot_fil_B <= rot_fil_B;
51             when others => rot_fil_A <= rot_fil_A;rot_fil_B <= rot_fil_B;
52             end case;
53       end if;
54 end process rotary_filter;
55 --====================DIRECTION FINDER====================================--
56 -- posokata ROT_DIR se 4ete pri ROT_IRQ
57 --========================================================================--
58
59 direction_finder: process(clk)
60    begin
61       if clk'event and clk='1' then
62          rot_fil_A_delayed <= rot_fil_A;
63          if rot_fil_A ='1' and rot_fil_A_delayed='0' then
64             ROT_IRQ <= '1';
65             rot_direction <= rot_fil_B;
66          else
67             ROT_IRQ <= '0';
68             rot_direction <= rot_direction;
69          end if;
```

```
70              end if;
71          end process direction_finder;
72          ROT_DIR <= rot_direction;
73          end Behavioral;
74
75
```