```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity examen is
 port (
 clk, clr, C: in std_logic;
 display: out std_logic_vector(6 downto 0));
attribute pin_numbers of examen: entity is
 "display(0):18 display(1):15 display(2):20 display(3):23 display(4):19 display(5):17 display(6):14";
end entity;
architecture arq_examen of examen is
 signal num:std_logic_vector (2 downto 0);
 constant DIG1:std_logic_vector(6 downto 0):= "1001111";
 constant DIG2:std_logic_vector(6 downto 0):= "0010010";
 constant DIG3:std_logic_vector(6 downto 0):= "0000110";
 constant DIG4:std_logic_vector(6 downto 0):= "1001100";
 constant DIG5:std_logic_vector(6 downto 0):= "0100100";
 constant DIG6:std_logic_vector(6 downto 0):= "0100000";
begin
 process(clk, clr, C)
 begin
  if (clr = '0') then
   num <= "000"; display <= "1111111";
  elsif (clk'event and clk='1') then
   case C is
    when '0' =>
     case num is
      when "000" => num <= "001"; display <= DIG2;
      when "001" => num <= "010"; display <= DIG3;
      when "010" => num <= "011"; display <= DIG4;
      when "011" => num <= "100"; display <= DIG5;
      when "100" => num <= "101"; display <= DIG6;
      when "101" => num <= "000"; display <= DIG1;
      when others => num <= "000"; display <= DIG1;
     end case;
    when '1'=>
     case num is
      when "000" => num <= num; display <= DIG1;
      when "001" => num <= num; display <= DIG2;
      when "010" => num <= num; display <= DIG3;
      when "011" => num <= num; display <= DIG4;
      when "100" => num <= num; display <= DIG5;
      when "101" => num <= num; display <= DIG6;
      when others => num <= num; display <= DIG1;
     end case;
    when others =>
     num <= "000"; display <= "1001111";
    end case;
   end if;
  end process;
end arq_examen;
```