```vhdl
-- Contador BCD y modulo 816


library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;


entity cnt_BCD_816 is
port(
      clk,nrst: in std_logic;
      rst   : buffer std_logic

);

end entity;

architecture rtl of cnt_BCD_816 is

      signal cnt_uni: std_logic_vector(3 downto 0);
      signal cnt_dec: std_logic_vector(3 downto 0);
      signal cnt_cen: std_logic_vector(3 downto 0);
      signal cout_u: std_logic;
      signal cout_d: std_logic;

begin

-- Unidades
process(clk,nrst)
begin
   if nrst = '0' then
      cnt_uni <= (others => '0');
   elsif clk'event and clk = '1' then
      if rst = '1' then
         cnt_uni <= (others => '0');
      else
         if cnt_uni = 9 then
            cnt_uni <= (others => '0');
         else
            cnt_uni <= cnt_uni + 1;
         end if;
      end if;
   end if;
end process;

cout_u <= '1' when cnt_uni = 9 else '0';

-- Decenas

process(clk,nrst)
begin
   if nrst = '0' then
      cnt_dec <= (others => '0');
   elsif clk'event and clk = '1' then
      if rst = '1' then
         cnt_dec <= (others => '0');
      elsif cout_u = '1' then
```

```
         if cnt_dec = 9 then
            cnt_dec <= (others => '0');
         else
            cnt_dec <= cnt_dec + 1;
         end if;
      end if;
   end if;
end process;

cout_d <= '1' when cnt_dec = 9 and cout_u = '1' else '0';

-- Centenas

process(clk,nrst)
begin
   if nrst = '0' then
      cnt_cen <= (others => '0');
   elsif clk'event and clk = '1' then
      if rst = '1' then
         cnt_cen <= (others => '0');
      elsif cout_d = '1' then
         if cnt_cen = 9 then
            cnt_cen <= (others => '0');
         else
            cnt_cen <= cnt_cen + 1;
         end if;
      end if;
   end if;
end process;

rst <= '1' when cnt_uni = 5 and cnt_dec = 1 and cnt_cen = 8 else '0';

end rtl;
```

```
-- Test-bench del contador BCD con ena y modulo 816


library ieee;
use ieee.std_logic_1164.all;

entity tb_cnt_BCD_816 is
end entity;

architecture test of tb_cnt_BCD_816 is

      signal clk, nrst: std_logic;
      signal ena: std_logic;
      signal rst: std_logic;

      constant T_CLK : time := 1 us;
      signal final: boolean := false;

begin

dut: entity Work.cnt_BCD_816(rtl)
```

```
port map(
      clk => clk,
      nrst => nrst,
      rst => rst
      );

process
begin

      clk <= '0';
      wait for T_CLK/2;
      clk <= '1';
      wait for T_CLK/2;
        if final then
           wait;
        end if;
end process;

process
begin
      -- Inicializacion asincrona
      nrst <= '0';
      wait until clk'event and clk = '1';
      wait until clk'event and clk = '1';


      nrst <= '1';
      wait until clk'event and clk = '1';
      wait until clk'event and clk = '1';
      wait for 100*T_CLK;
      wait until clk'event and clk = '1';


      wait until clk'event and clk = '1';
      wait for 1000*T_CLK;
      wait until clk'event and clk = '1';


      final <= true;
      wait;

end process;
end test;
```