BCD Counter Code:-

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity count is

Port ( clk : in STD\_LOGIC;

clk\_div: inout std\_logic;

rst : in STD\_LOGIC;

an : out std\_logic\_vector(3 downto 0);

seg : out std\_logic\_vector(6 downto 0);

q : inout STD\_LOGIC\_VECTOR (3 downto 0));

end count;

architecture Behavioral of count is

signal count : std\_logic\_vector( 25 downto 0):= (others => '0');

begin

an<="0001";

process(clk\_div,rst)

variable temp: std\_logic\_vector(3 downto 0):="0000";

begin

if (rst='1') then

temp:= "0000";

elsif (clk\_div'event and clk\_div='1') then

temp:= temp+'1';

if( temp > "1001") then

temp := "0000";

end if;

end if;

q<=temp;

end process;

process(clk)

begin

if (clk'event and clk='1') then

count <= count + '1';

end if;

end process;

clk\_div<= count(1);

--clk\_div<= count(1);

process(q)

begin

case q is

when "0000" => seg <= "1000000"; -- 0

when "0001" => seg <= "1001111"; -- 1

when "0010" => seg <= "0100100"; -- 2

when "0011" => seg <= "0110000"; -- 3

when "0100" => seg <= "0011001"; -- 4

when "0101" => seg <= "0010010"; -- 5

when "0110" => seg <= "0000010"; -- 6

when "0111" => seg <= "1111000"; -- 7

when "1000" => seg <= "0000000"; -- 8

when "1001" => seg <= "0010000"; -- 9

when others => seg <= "1111111";

end case;

end process;

end Behavioral;

BCD\_Counter\_TB –

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity COUNTER\_TB is

-- Port ( );

end COUNTER\_TB;

architecture Behavioral of COUNTER\_TB is

COMPONENT count is

Port ( clk : in STD\_LOGIC;

clk\_div: inout std\_logic;

rst : in STD\_LOGIC;

q : inout STD\_LOGIC\_VECTOR (3 downto 0));

end component;

signal clk,clk\_div: std\_logic;

signal rst:std\_logic;

signal q: std\_logic\_vector(3 downto 0);

begin

u1: count port map(clk,clk\_div,rst,q);

process

begin

clk<='1';

wait for 100 ns;

clk<='0';

wait for 100 ns;

end process;

rst<='1', '0' after 300 ns ;

end Behavioral;