RTL code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mod8_counter is
   Port ( CLK,RST : in  STD_LOGIC;
          COUNT : inout  STD_LOGIC_VECTOR (2 downto 0));
end mod8_counter;

architecture Behavioral of mod8_counter is

begin
process (CLK,RST)
begin

if (RST = '1')then
COUNT <= "000";
elsif(rising_edge(CLK))then
COUNT <= COUNT+1;

end if;
end process;
end Behavioral;
```

TB:

```
ENTITY mod8_tb IS
END mod8_tb;

ARCHITECTURE behavior OF mod8_tb IS

   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT mod8_counter
   PORT(
      CLK : IN  std_logic;
      RST : IN  std_logic;
      COUNT : INOUT  std_logic_vector(2 downto 0)
```

```
    );
 END COMPONENT;


  --Inputs
  signal CLK : std_logic := '0';
  signal RST : std_logic := '0';

            --BiDirs
  signal COUNT : std_logic_vector(2 downto 0);

  -- Clock period definitions
  constant CLK_period : time := 10 ns;

BEGIN

            -- Instantiate the Unit Under Test (UUT)
  uut: mod8_counter PORT MAP (
       CLK => CLK,
       RST => RST,
       COUNT => COUNT
      );

  -- Clock process definitions
  CLK_process :process
  begin
                        CLK <= '0';
                        wait for CLK_period/2;
                        CLK <= '1';
                        wait for CLK_period/2;
  end process;


  -- Stimulus process
  stim_proc: process
  begin
    -- hold reset state for 100 ns.
   RST <= '1';
              wait for 50 ns;
   RST <= '0';
    -- insert stimulus here

    wait;
  end process;

END;
```