```
-- This goes in the ‘gray_counter_defs.vhd’ file
-- Package of definitions for gray code counter
package gray_counter_defs is
   type state_type is (S0, S1, S2, S3);
end gray_counter_defs;



-- Gray code counter with synchronous reset and
-- count enable. This is the ‘gray_counter.vhd’
-- file

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use work.gray_counter_defs.all;

entity gray_counter is
    Port ( clock : in std_logic;
           reset : in std_logic;
          enable : in std_logic;
           count : out std_logic_vector (1 downto 0));
end gray_counter;

architecture Behavioral of gray_counter is
   type state_type is (S0, S1, S2, S3);
   signal state: state_type;
begin
   process (clock, reset)
      begin
         if rising_edge(clock) then
            if reset = '1' then
               state <= S0;
            elsif enable = '1' then
               case state is
                  when S0 => state <= S1;
                  when S1 => state <= S2;
                  when S2 => state <= S3;
                  when others =>    state <= S0;
               end case;
            end if;
         end if;
   end process;

   --  signal assignments for outputs
   count <= "00" when state = S0 else
            "01" when state = S1 else
            "11" when state = S2 else
            "10";  -- when state = S3
end Behavioral;
```