MOD N Simple code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_arith.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity modn is
   Generic(N:integer:=6);
   Port ( clk : in  STD_LOGIC;
          rst : in  STD_LOGIC;
          Q : out integer range 0 to N-1);
end modn;

architecture Behavioral of modn is
signal temp:integer range 0 to N-1;
begin
process(clk,rst)
begin
if clk'event and clk='1' then
if(rst='1' OR temp>=N-1) then
temp <=0;
else
```

```
temp<=temp+1;
end if;
end if;
Q <=temp;
end process;
end Behavioral;
```

MOD N testbench code

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY modtb IS
END modtb;

ARCHITECTURE behavior OF modtb IS

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

   COMPONENT modn
   PORT(
        clk : IN  std_logic;
        rst : IN  std_logic;
        Q : OUT  integer range 0 to 5
       );
   END COMPONENT;


  --Inputs
```

```
  signal clk : std_logic := '0';
  signal rst : std_logic := '0';

        --Outputs
  signal Q : integer range 0 to 5;

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

BEGIN

        -- Instantiate the Unit Under Test (UUT)
  uut: modn PORT MAP (
       clk => clk,
       rst => rst,
       Q => Q
      );

  -- 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.
    wait for 100 ns;
    rst<='1';
    wait for 100 ns;
                    rst<='0';
    wait for clk_period*10;
    -- insert stimulus here
    wait;
  end process;
END;
```