library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity mod25 is

Port ( rst : in STD\_LOGIC;

pr : in STD\_LOGIC;

clk : in STD\_LOGIC;

dir : in STD\_LOGIC;

Q : out STD\_LOGIC\_VECTOR (4 downto 0));

end mod25;

architecture mod25\_arch of mod25 is

signal Qtemp : STD\_LOGIC\_VECTOR (4 downto 0) := "00000";

begin

process(rst,pr,clk,dir)

begin

if rst ='1' then

Qtemp <= (OTHERS =>'0');

elsif pr='1' then

Qtemp <= (OTHERS =>'1');

elsif falling\_edge(clk) then

if dir = '1' then

if Qtemp < 24 then

Qtemp <= Qtemp + 1;

Else

Qtemp <= "00000";

end if;

else

if Qtemp > 7 then

Qtemp <= Qtemp - 1;

else

Qtemp <= "11111";

end if;

end if;

end if;

end process;

Q<=Qtemp;

end mod25\_arch;

# 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;