**Piso main code**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity piso is

Port ( clk : in STD\_LOGIC;

slow\_clk: inOUT STD\_LOGIC;

load : in STD\_LOGIC;

in3 : in STD\_LOGIC\_VECTOR (3 downto 0);

out3 : out STD\_LOGIC);

end piso;

architecture Behavioral of piso is

signal s3 : STD\_LOGIC;

signal temp : STD\_LOGIC\_VECTOR(3 DOWNTO 0);

signal counter: std\_logic\_vector(1 downto 0):=(others =>'0');

begin

process(slow\_clk,in3,load,s3)

begin

if(load='1') THEN

temp(3 DOWNTO 0) <= in3(3 DOWNTO 0);

elsif(slow\_clk'EVENT and slow\_clk='1') THEN

temp(3 DOWNTO 1) <= temp(2 DOWNTO 0);

s3 <= temp(3);

end if;

end process;

out3 <= s3;

process(clk)

begin

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

counter <=counter+'1';

end if;

end process;

slow\_clk <= counter(1);

end Behavioral;

**Piso\_tb**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity piso\_tb is

-- Port ( );

end piso\_tb;

architecture Behavioral of piso\_tb is

component piso

Port ( clk : in STD\_LOGIC;

slow\_clk: inout STD\_LOGIC;

load : in STD\_LOGIC;

in3 : in STD\_LOGIC\_VECTOR (3 downto 0);

out3 : out STD\_LOGIC);

end component;

signal clk, load, out3,slow\_clk: std\_logic;

signal in3 : STD\_LOGIC\_VECTOR (3 downto 0);

begin

u1: piso port map ( clk,slow\_clk,load,in3,out3);

process

begin

clk<= '0';

wait for 10 ns;

clk<= '1';

wait for 10 ns ;

end process;

Process

begin

load<= '1' ;

in3<= "1100";

wait for 20 ns;

load<= '0';

wait for 500 ns ;

end process;

end Behavioral;

**Output:**

