library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity Generator is

port(clk,reset: in std\_logic;

Data: out std\_logic\_vector(3 downto 0));

end Generator;

architecture Gen of generator is

component D\_FF is

port(d,reset,clk: in std\_logic;

q: out std\_logic);

end component;

component nxor\_gate is

port ( a,b: in std\_logic;

c: out std\_logic);

end component;

signal Out3, Out2, Out1, Out0, Out1NXOROut0: std\_logic;

begin

NXOR1: NXOR\_Gate port map(Out1, Out0, Out1NXOROut0);

Bist3: D\_FF port map (out1NXOrout0,reset,clk,Out3);

Bist2: D\_FF port map(Out3, reset, clk, Out2);

Bist1: D\_FF port map(Out2, reset, clk, Out1);

Bist0: D\_FF port map(Out1, reset, clk, Out0);

Data<=Out3&Out2&Out1&Out0;

end Gen;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity nxor\_gate is

port ( a,b: in std\_logic;

c: out std\_logic);

end nxor\_gate;

architecture nxor1 of nxor\_gate is

begin

c<= a xnor b;

end nxor1;

library ieee;

use ieee.std\_logic\_1164.all;

entity D\_FF is

port(d,reset,clk: in std\_logic;

q: out std\_logic);

end D\_FF;

architecture D of D\_FF is

begin

process(clk, reset)

begin

if reset='1' then

q<= '0';

elsif clk='1' and clk'event then

q<=d;

end if;

end process;

end D;