library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity FIFO is

port(Inp: in bit\_vector(3 downto 0);

Outp: out bit\_vector(3 downto 0);

clk,reset: in bit);

end FIFO;

architecture FIFO1 of FIFO is

component FIFO\_cell is

port(clk,reset,SerialIn: in bit;

dataOut: out bit);

end component;

begin

Cell1: FIFO\_Cell port map(clk,reset,Inp(3),Outp(3));

Cell2: FIFO\_Cell port map(clk,reset,Inp(2),Outp(2));

Cell3: FIFO\_Cell port map(clk,reset,Inp(1),Outp(1));

Cell4: FIFO\_Cell port map(clk,reset,Inp(0),Outp(0));

end FIFO1;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity FIFO\_cell is

port(clk,reset,SerialIn: in bit;

dataOut: out bit);

end FIFO\_cell;

architecture FIFO1 of FIFO\_cell is

begin

process(clk,reset,serialIn)

variable state: bit\_vector(3 downto 0):="0000";

begin

if reset = '1' then

state:="0000";

elsif clk='1' and clk'event then

state := state srl 1;

state(3):= serialIn;

dataOut<=state(0);

end if;

end process;

end FIFO1;