library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity reg is

port(clk,reset: in std\_logic;

mode: in std\_logic\_vector(1 downto 0);

d: in std\_logic\_vector(3 downto 0);

serialIn: in std\_logic;

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

end reg;

architecture REGISTER1 of reg is

component MUX\_D is

port(reset,clk:in std\_logic;

i3,i2,i1,i0: in std\_logic;

sel: in std\_logic\_vector(1 downto 0);

Q: out std\_logic);

end component;

signal intQ: std\_logic\_vector(3 downto 0);

begin

first: MUX\_D port map(reset,clk,intQ(3),d(3),serialIn,intQ(2),mode,intQ(3));

second: MUX\_D port map(reset,clk,intQ(2),d(2),intQ(3),intQ(1),mode,intQ(2));

third: MUX\_D port map(reset,clk,intQ(1),d(1),intQ(2),intQ(0),mode,intQ(1));

fourth: MUX\_D port map(reset,clk,intQ(0),d(0),intQ(1),intQ(3),mode,intQ(0));

q<=intQ;

end REGISTER1;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity D\_FF is

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

q: out std\_logic);

end D\_FF;

architecture D of D\_FF is

begin

process(set,reset,clk)

begin

if(reset='1') then

q<='0';

elsif clk='1' and clk'event and set='1' then

q<='1';

elsif clk='1' and clk'event then

q<=d;

end if;

end process;

end D;

library ieee;

use ieee.std\_logic\_1164.all;

use std\_logic\_unsigned.all;

entity mux is

port(s: in std\_logic\_vector(1 downto 0);

i: in std\_logic\_vector(3 downto 0);

y: out std\_logic);

end entity mux;

architecture a of mux is

begin

y<=i(conv\_integer(s));

end;

library ieee;

use ieee.std\_logic\_1164.all;

use std\_logic\_unsigned.all;

entity MUX\_D is

port(reset,clk:in std\_logic;

i3,i2,i1,i0: in std\_logic;

sel: in std\_logic\_vector(1 downto 0);

Q: out std\_logic);

end MUX\_D;

architecture MUXD of MUX\_D is

component mux is

port(s: in std\_logic\_vector(1 downto 0);

i: in std\_logic\_vector(3 downto 0);

y: out std\_logic);

end component;

component D\_FF is

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

q: out std\_logic);

end component;

signal int,set: std\_logic;

signal i: std\_logic\_vector(3 downto 0);

begin

i<=i3&i2&i1&i0;

choose: mux port map(sel,i,int);

delay: d\_ff port map(set,reset,clk,int,q);

end MUXD;