library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity reg3 is

port( clk : in std\_logic;

deop\_alu\_in3 : in std\_logic\_vector(3 downto 0);

deop\_alu\_out3 : out std\_logic\_vector(3 downto 0);

reg\_alu\_in3 : in std\_logic\_vector(7 downto 0);

reg\_alu\_out3 : out std\_logic\_vector(7 downto 0);

mux2\_dmem\_in : in std\_logic\_vector(7 downto 0);

mux2\_dmem\_out : out std\_logic\_vector(7 downto 0);

alu\_mux3\_in : in std\_logic\_vector(7 downto 0);

alu\_mux3\_out : out std\_logic\_vector(7 downto 0);

mux3\_selectline\_in : in std\_logic;

mux3\_selectline\_out : out std\_logic;

mux3\_reg\_in : in std\_logic\_vector(7 downto 0);

mux3\_reg\_out : out std\_logic\_vector(7 downto 0)

);

end reg3;

architecture behv of reg3 is

begin

deop\_alu\_sig: process(clk, deop\_alu\_in3)

begin

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

deop\_alu\_out3 <= deop\_alu\_in3;

end if;

end process;

reg\_alu\_sig: process(clk, reg\_alu\_in3)

begin

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

reg\_alu\_out3 <= reg\_alu\_in3;

end if;

end process;

mux2\_dmem\_sig: process(clk, mux2\_dmem\_in)

begin

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

mux2\_dmem\_out<= mux2\_dmem\_in;

end if;

end process;

alu\_mux3\_sig: process(clk , alu\_mux3\_in)

begin

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

alu\_mux3\_out <= alu\_mux3\_in;

end if;

end process;

mux3\_selectline\_sig: process(clk , mux3\_selectline\_in)

begin

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

mux3\_selectline\_out <= mux3\_selectline\_in;

end if;

end process;

mux3\_reg\_sig: process(clk, mux3\_reg\_in)

begin

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

mux3\_reg\_out <= mux3\_reg\_in;

end if;

end process;

end behv;