library IEEE;

use IEEE.std\_logic\_1164.all;

use IEEE.std\_logic\_unsigned.all;

entity writeback is

PORT

(

writeback\_reg\_alu : IN std\_logic\_vector(7 DOWNTO 0); --memd from Rx (register bank)

writeback\_mux2\_dmem : IN std\_logic\_vector(7 DOWNTO 0);--dAddress from Mux

writeback\_deop\_alu: in std\_logic\_vector(3 downto 0); -- opcode coming in

-- q internal signal

writeback\_clk\_sig : IN std\_logic;

--PC : IN std\_logic\_vector ( DATA\_WIDTH - 1 DOWNTO 0); -- PC COUNTER INPUT

writeback\_alu\_mux3: in std\_logic\_vector(7 downto 0);

-- B is q internal signal

writeback\_mux3\_selectline: in std\_logic;

writeback\_mux3\_reg: out std\_logic\_vector( 7 downto 0)

);

end entity;

architecture top of writeback is

-----------------------------------------COMPONENTS----------------------------------------------

component datamem IS

PORT

(

input : IN std\_logic\_vector(7 DOWNTO 0); --memd from Rx (register bank)

address : IN std\_logic\_vector(7 DOWNTO 0);--dAddress from Mux

aluop: in std\_logic\_vector(3 downto 0); -- opcode coming in

q : OUT std\_logic\_vector(7 DOWNTO 0);-- output

clk : IN std\_logic

);

END component datamem;

component mux is

port(

a: in std\_logic\_vector(7 downto 0);

b: in std\_logic\_vector(7 downto 0);

sel: in std\_logic;

output: out std\_logic\_vector(7 downto 0)

) ;

end component mux;

-------------------------------------SIGNALS---------------------------------------------------

signal writeback\_dmem\_mux3: std\_logic\_vector(7 downto 0);

begin

datamem1: datamem port map(writeback\_reg\_alu,writeback\_mux2\_dmem,writeback\_deop\_alu,writeback\_dmem\_mux3,writeback\_clk\_sig);

mux1: mux port map(writeback\_alu\_mux3,writeback\_dmem\_mux3,writeback\_mux3\_selectline, writeback\_mux3\_reg);

end top;