LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.numeric\_std.ALL;

ENTITY datamem IS

GENERIC

(

ADDRESS\_WIDTH : integer := 8;

DATA\_WIDTH : integer := 8

);

PORT

(

input : IN std\_logic\_vector(DATA\_WIDTH - 1 DOWNTO 0); --memd from Rx (register bank)

address : IN std\_logic\_vector(ADDRESS\_WIDTH - 1 DOWNTO 0);--dAddress from Mux

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

q : OUT std\_logic\_vector(DATA\_WIDTH - 1 DOWNTO 0);-- output

clk : IN std\_logic

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

);

END datamem;

ARCHITECTURE logic OF datamem IS

TYPE dmem IS ARRAY(0 TO 2 \*\* ADDRESS\_WIDTH - 1) OF std\_logic\_vector(DATA\_WIDTH - 1 DOWNTO 0);

SIGNAL dmem\_block : dmem := (OTHERS => "00000000");

BEGIN

PROCESS (input,address,aluop )

BEGIN

IF (aluop= "1001" or aluop="1011") then

dmem\_block(to\_integer(unsigned(address))) <= input; -- store instructions to data memory

--y<=1 after 200ps;

else

q<="00000000";

END IF;

if (aluop = "1000" or aluop = "1010") THEN

q <= dmem\_block(to\_integer(unsigned(address))) ; -- load instructions to output

-- y<=2 after 200ps ;

else

q<="00000000";

END IF;

END PROCESS;

END logic;