```
module DRAM(
    input clk,
    input [3:0]write,
    input [63:0] address ,
    input [63:0] data_in ,
    output reg [63:0] data_out);
    reg [15:0] MEM [65535:0];
    initial begin
        MEM[0]=16'd5; //mat_a_base
        MEM[1]=16'd32; //mat_b_base
        MEM[2]=16'd57; //mat_c_base

        MEM[3]=16'd5; //i_ref
        MEM[4]=16'd5; //j_ref

        //[13, 17, 1, 5, 13]
        MEM[5]=16'd13;
        MEM[6]=16'd17;
        MEM[7]=16'd1;
        MEM[8]=16'd5;
        MEM[9]=16'd13;
        //[5, 20, 18, 14, 5]
        MEM[10]=16'd5;
        MEM[11]=16'd20;
        MEM[12]=16'd18;
        MEM[13]=16'd14;
        MEM[14]=16'd5;
        //[2, 2, 10, 1, 17]
        MEM[15]=16'd2;
        MEM[16]=16'd2;
        MEM[17]=16'd10;
        MEM[18]=16'd1;
        MEM[19]=16'd17;
        //[0, 10, 13, 5, 3]
        MEM[20]=16'd0;
        MEM[21]=16'd10;
        MEM[22]=16'd13;
        MEM[23]=16'd5;
        MEM[24]=16'd3;
        //[12, 8, 6, 16, 5]
        MEM[25]=16'd12;
        MEM[26]=16'd8;
        MEM[27]=16'd6;
        MEM[28]=16'd16;
        MEM[29]=16'd5;

        MEM[30]=16'd5; //j_ref
        MEM[31]=16'd5; //k_ref

        //[8, 18, 18, 13, 13]
        MEM[32]=16'd8;
        MEM[33]=16'd18;
        MEM[34]=16'd18;
        MEM[35]=16'd13;
```

```
        MEM[36]=16'd13;
        //[20, 15, 7, 4, 9]
        MEM[37]=16'd20;
        MEM[38]=16'd15;
        MEM[39]=16'd7;
        MEM[40]=16'd4;
        MEM[41]=16'd9;
        //[19, 10, 18, 10, 8]
        MEM[42]=16'd19;
        MEM[43]=16'd10;
        MEM[44]=16'd18;
        MEM[45]=16'd10;
        MEM[46]=16'd8;
        //[3, 9, 10, 12, 10]
        MEM[47]=16'd3;
        MEM[48]=16'd9;
        MEM[49]=16'd10;
        MEM[50]=16'd12;
        MEM[51]=16'd10;
        //[3, 9, 20, 19, 15]
        MEM[52]=16'd3;
        MEM[53]=16'd9;
        MEM[54]=16'd20;
        MEM[55]=16'd19;
        MEM[56]=16'd15;

    end
        always @(posedge clk)
            begin
                if (write[0:0] == 1)
                begin
                    MEM[address[15:0]] <= data_in[15:0];
                    $display("address0:- %d, mem_data_in0:-
%d",address[15:0],data_in[15:0]);
                end
                if (write[1:1] == 1)
                begin
                    MEM[address[31:16]] <= data_in[31:16];
                    $display("address1:- %d, mem_data_in1:-
%d",address[31:16],data_in[31:16]);
                end
                if (write[2:2] == 1)
                begin
                    MEM[address[47:32]] <= data_in[47:32];
                    $display("address2:- %d, mem_data_in2:-
%d",address[47:32],data_in[47:32]);
                end
                if (write[3:3] == 1)
                begin
                    MEM[address[63:48]] <= data_in[63:48];
                    $display("address3:- %d, mem_data_in3:-
%d",address[63:48],data_in[63:48]);
                end
                data_out[15:0] <= MEM[address[15:0]];
                data_out[31:16] <= MEM[address[31:16]];
```

```
106                 data_out[47:32] <= MEM[address[47:32]];
107                 data_out[63:48] <= MEM[address[63:48]];
108         end
109 endmodule
```