```
module mdr(
    input clk,  //clock
    input rst,  //reset
    input mdr_ena,  //mdr enable
    input [1:0] mdr_sel,  //data
select
    input [15:0] reg_in1,   //signal
from register
    input [15:0] reg_in2,
    input [15:0] mem_in,    //signal
from memory
    output reg [15:0] mem_out,
//output to data bus
    output reg [15:0] reg_out
//output to instruction register
    );

    parameter regin1=2'b01,
              regin2=2'b10,
              memin=2'b11;
    always @ (posedge clk or negedge
rst)
begin
        if(!rst)
        begin
            reg_out=16'h0000;
            mem_out=16'h0000;
        end
    else if(mdr_ena)
        begin
            case(mdr_sel)
            regin1:
            begin
                reg_out=reg_in1;
                mem_out=16'hz;
            end
            regin2:
            begin
                reg_out=reg_in2;
                mem_out=16'hz;
            end
            memin:
            begin
                reg_out=16'hz;
                mem_out=mem_in;
            end
            default:
            begin
                reg_out=16'hz;
                mem_out=16'hz;
            end
            endcase
        end
    end
endmodule
```