```
module mar(
    input clk,  //clock signal, "fetch" in clock.v
    input rst,  //reset signal
    input mar_ena,  //whether enable memory address register or not
    input [1:0] mar_sel,//select the address source
    input [15:0] ir_addr1,  //data from register
    input [15:0] ir_addr2,  //data from register
    input [15:0] pc_addr,  //data from PC
    input [15:0] sp_addr,  //data from sp
    output reg [15:0] mar_addr  //memory address
    );
    parameter pc=2'b00,//from PC
              dr=2'b01,//from reg_out1[DR]
              sr=2'b10,//from reg_out2[SR]
              sp=2'b11;//from sp

    always @ (posedge clk)
    begin
        if(!rst)
        begin
            mar_addr=16'h0000;  //reset the address
        end
        else if(mar_ena)
        begin
            case(mar_sel)
            pc:
            mar_addr=pc_addr;
            dr:
            mar_addr=ir_addr1;
            sr:
            mar_addr=ir_addr2;
            sp:
            mar_addr=sp_addr;
            endcase
        end
    end
endmodule
```