1. Design the switch device that communicated with the processor using the bussing system that constitutes address bus (abus) and data bus (dbus) and write enable (we). The busses and write enable are provided below. You don’t have to instantiate the SW device module. You can ignore the debouncing issue of SW. Implement the design in Verilog.

```verilog
wire [(DBITS-1):0] abus;
tri [(DBITS-1):0] dbus;
wire we;

// In the processor
assign abus = memaddr_M;
assign we = wrmem_M;
assign dbus = wrmem_M ? wmemval_M : {DBITS{1'bz}};

module SwDevice (clk, reset, abus, dbus, we, sw);
    parameter DBITS = 32;
    parameter SW_BITS = 10;
    parameter SW_BASE = 32'hF00000008;
    parameter CTRL_BASE = 32'hF00000018;

    inputclk, reset, we,
    input [DBITS-1 :0] abus;
    inout [DBITS-1 :0] dbus;
    input [SWBITS-1 :0] sw;
    reg [SWBITS-1 :0] oldSw;
    reg [DBITS-1 :0] ctrl;

    wire swEn = (abus == SW_BASE);
    wire ctrlEn = (abus == CTRL_BASE);
    wire rdSw = (swEn && !we);
    wire rdCtrl = (ctrlEn && !we);
    wire wrSw = (swEn && we);
    wire wrCtrl = (ctrlEn && we);

    assign dbus = rdSw ? {{(DBITS - SWBITS){1'b0}}, oldSw} :
                      rdCtrl ? ctrl :
                      {DBITS{1'bz}};

    always @(posedge clk) begin
        if (reset) begin
            oldSw <= sw;
            ctrl <= 0;
        end else if (rdSw) begin
            ctrl[0] <= 0;
            ctrl[2] <= 0;
        end else if (rdCtrl) begin
            ctrl[8] <= 0;
        end else if (wrSw) begin
            oldSw <= sw;
            ctrl[0] <= 0;
            ctrl[2] <= 0;
        end else if (wrCtrl) begin
            oldSw <= sw;
            ctrl[1] <= 0;
            ctrl[2] <= 0;
    end
endmodule
```
end else if (wrCtrl) begin
    if (dbus[0] == 0) begin
        ctrl[0] <= 1'b0;
    end
    if (dbus[2] == 0) begin
        ctrl[2] <= 1'b0;
    end
end else if (oldSw != sw) begin
    if (ctrl[0]) begin
        ctrl[2] <= 1;
    end else begin
        ctrl[0] <= 1;
    end
    oldSw <= sw;
end
endmodule
2. List the necessary changes in the processor for supporting interrupts. Enumerate the extra registers and instructions.

**Registers**
1. PCS (Processor Control Status)
2. IRA (Interrupt Handler Address)
3. IRA (Interrupt Return Address)
4. IDN (Interrupt Device Number)

**Instructions:**
1. RETI (Return from interrupt)
2. RSR (Read system register)
3. WSR (Write system register)