Course Name: Computer Architecture	**Course Code:** CMPE-421L
Assignment Type: Lab	**Dated:** 6th November2023
Semester: 7th	**Session:** 2020
Lab/Project/Assignment #: 8	**CLOs to be covered: CLO 2**
Lab Title: CSR Support	**Teacher Name:** Engr. Afeef Obaid

Lab Evaluation

CLO 2	Understand the basics of RISC-V architecture, its assembly & design of basic Datapath components of a single cycle RISC-V processor.					
Levels (Marks)	**Level1**	**Level2**	**Level3**	**Level4**	**Level5**	**Level6**
(10)						
					Total	**/10**

Rubrics for Current Lab Evaluation

Scale	**Marks**	**Level**	**Rubric**
Excellent	**9-10**	L1	Submitted all lab tasks, BONUS task, have good understanding.
Very Good	**7-8**	L2	Submitted the lab tasks but have good understanding
Good	**5-6**	L3	Submitted the lab tasks but have weak understanding.
Basic	**3-4**	L4	Submitted the lab tasks but have no understanding.
Barely Acceptable	**1-2**	L5	Submitted only one lab task.
Not Acceptable	**0**	L6	Did not attempt

Lab # 8

Lab Goals

By reading this manual, students will be able to:

- Understand CSR Register File
- Understand how to implementing the read and write operations in CSR.

Equipment Required

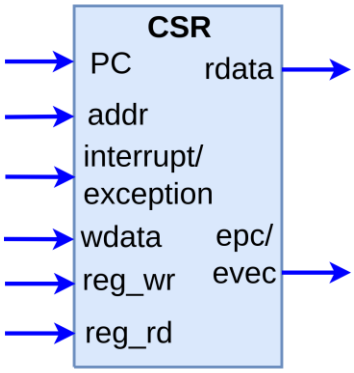
- Computer system with ModelSim or Xcelium, installed on it.

CSR Support

In this lab, we are going to support privileged architecture in our single cycle processor. For this purpose, we are going to partially support the machine mode of the RISC-V specification in our processor. This will involve adding support for some new instructions in our datapath. We are also going to create a new register file which is going to contain the machine mode CSR registers which can be accessed by these new instructions.

CSR Register File

First, we are going to create a new registerfile which will contain our CSR registers. For this lab, we are only going to implement mip, mie, mstatus, mcause, mtvec and mepc in our register file. These registers have their 12-bit address defined as part of the RISC-V specification. The register file will have the address of the CSR register, the value of PC during the Memory-Writeback stage, the CSR register write control, the CSR register read control, the interrupt/exception pins and the CSR register write data at its input. The register file will have the CSR read data and the exception PC at its output.



CSR Register File.

Above Figure illustrates the implementation of the CSR Register file read and write operations. In the illustrated code we can see that the read and write operation will depend on the 12-bit address of the CSR registers. For checking the values address of the CSR registers refer to Table 2.5 of the RISC-V privileged specifications manual.

```
// CSR read operation
always_comb begin
   csr_rdata  = '0;
   if(exe2csr_ctrl.csr_reg_rd) begin
       case (exe2csr_data.csr_addr)
           CSR_ADDR_MSTATUS        : csr_rdata    = csr_mstatus_ff;
           CSR_ADDR_MIE            : csr_rdata    = csr_mie_ff;
           ...
           CSR_ADDR_MEPC           : csr_rdata    = csr_mepc_ff;
       endcase // exu2csr_data.csr_addr
```

```
    end
end
// CSR write operation
always_comb begin
    csr_mstatus_wr_flag         = 1'b0;
    csr_mie_wr_flag             = 1'b0;
    ...
    csr_mepc_wr_flag            = 1'b0;
    if (exe2csr_ctrl.csr_wr_req) begin
        case (exe2csr_data.csr_addr)
            CSR_ADDR_MSTATUS        : csr_mstatus_wr_flag  = 1'b1;
            CSR_ADDR_MIE            : csr_mie_wr_flag      = 1'b1;
            ...
            CSR_ADDR_MEPC           : csr_mepc_wr_flag     = 1'b1;
        endcase // exu2csr_data.csr_addr
    end // exe2csr_ctrl.csr_wr_req
end

// Update the mip (machine interrupt pending) CSR
always_ff @(negedge rst_n, posedge clk) begin
    if (~rst_n) begin
        csr_mip_ff <= {`XLEN{1'b0}};
    end else if (csr_mip_wr_flag) begin
        csr_mip_ff <= csr_wdata;
    end
end
...
// Update the mtvec CSR
always_ff @(negedge rst_n, posedge clk) begin
    if (~rst_n) begin
        csr_mtvec_ff <= {`XLEN{1'b0}};
    end else if (csr_mtvec_wr_flag) begin
        csr_mtvec_ff <= csr_wdata;
    end
end
```

CSR Register File Read and Write operations.

Tasks

- Implement the CSR register file while complying with the specifications manual and simulate it to check the read and write operation.