University Of Engineering and Technology, Lahore
Computer Engineering Department

Course Name: Computer Architecture	**Course Code:** CMPE-421L
Assignment Type: Lab	**Dated:** 13th November2023
Semester: 7th	**Session:** 2020
Lab/Project/Assignment #: 9	**CLOs to be covered: CLO 2**
Lab Title: Interrupt Handling in CSR	**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 # 9

Lab Goals

By reading this manual, students will be able to:

- Understand the Interrupt in CSR
- Understand how to implementing MRET operations in CSR.

Equipment Required

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

Implementing the CSR Instructions

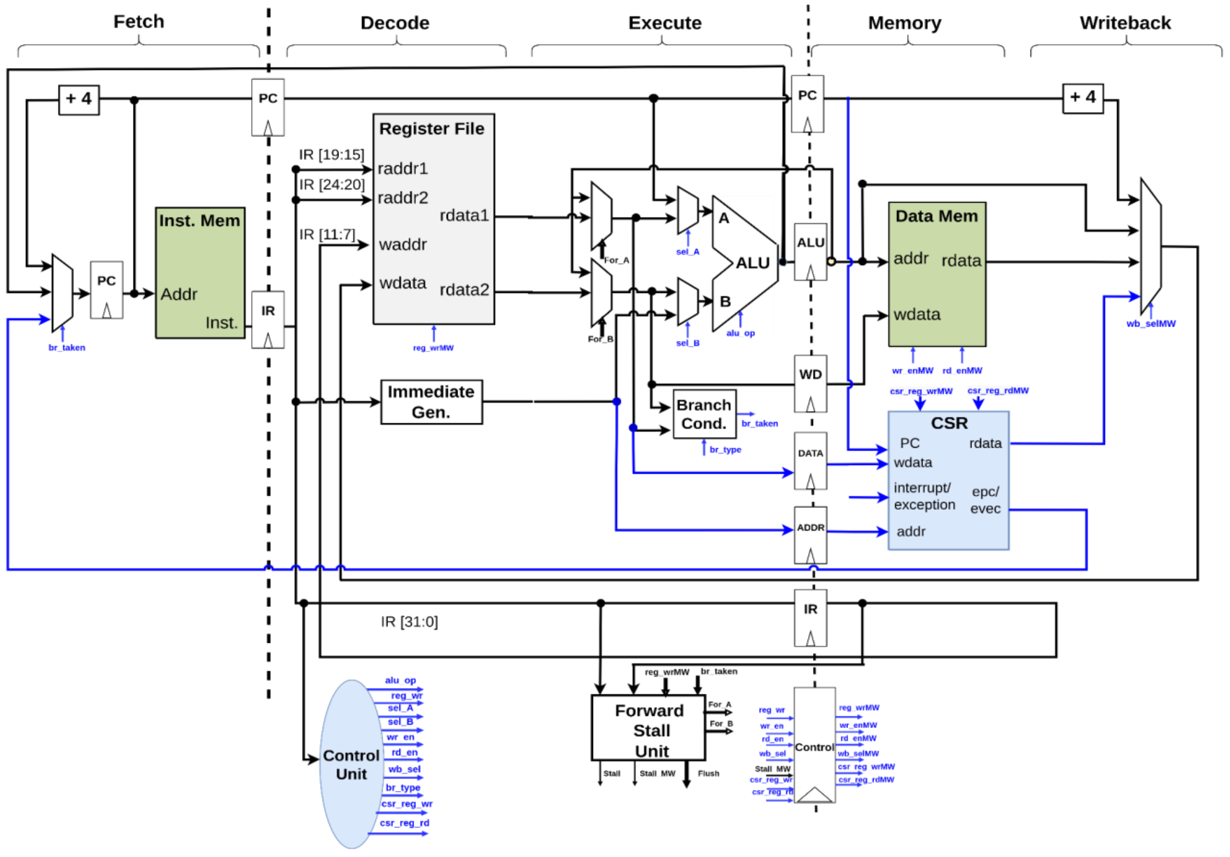
In this lab, we are going to implement the CSRRW and mret instructions in our pipeline. The CSRRW is the CSR Read Write instruction which reads the old value of the CSR register and saves it in the destination register. Then the value of the source register is written to the CSR register.

31 20	15	12	7	0
csr	rs1	func3	rd	opcode

CSRRW Instruction Format.

Implementation of CSRRW

For implementing these instructions, we will be required to integrate the CSR register file in our datapath. We will also modify our controller for adding two new control signals for the read and write operation of the CSR register file. The address of the CSR register file will be given by the immediate value generator and the data to be written to the CSR register will be given by the output of the rs1 forwarding mux. The read data output of the CSR register file will be connected to the writeback mux.
For Single Cycle Processor: https://shorturl.at/gmGW5



Integration of CSRRW instruction in the Datapath.

Tasks

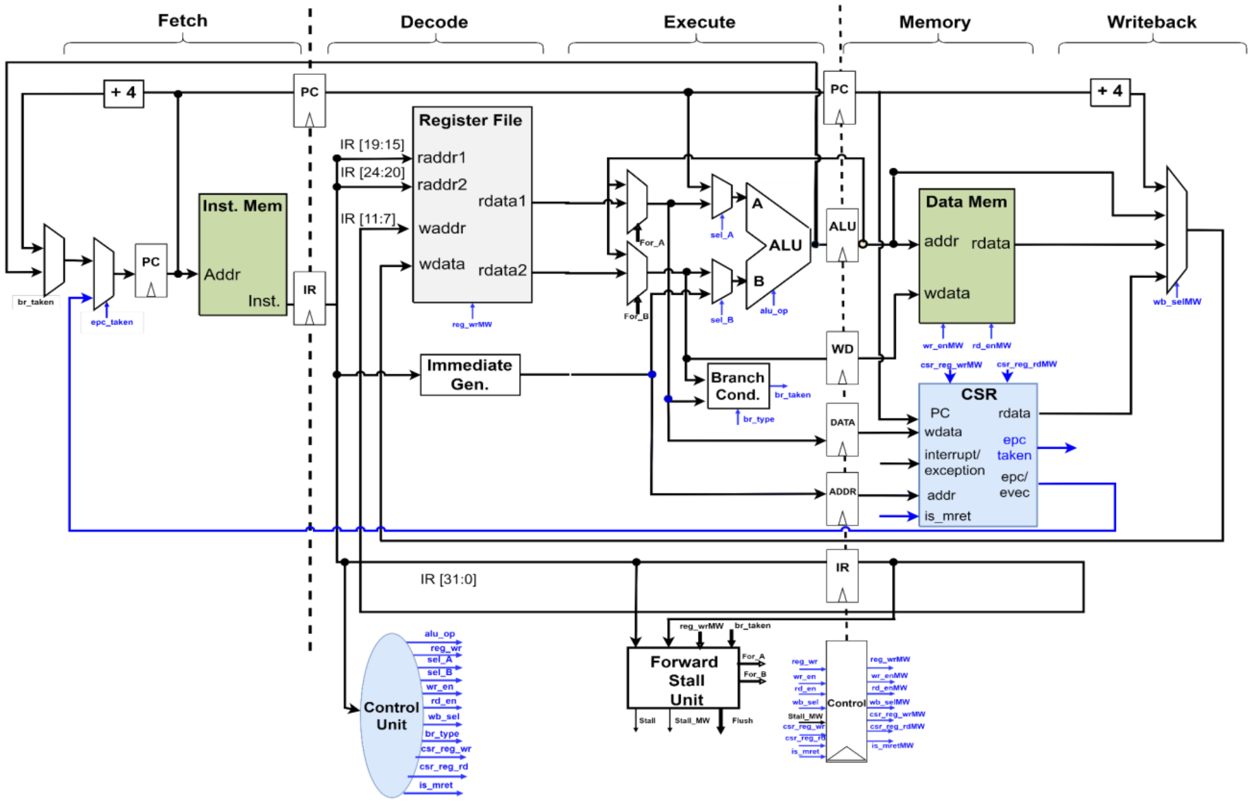
- Integrate the CSR register file and implement the CSRRW instruction in your processor.
- Write a simple assembly code to test whether the CSR instruction is working correctly or not. A sample code has been provided below

```
li x10,1
li x12,10
csrrw x11,mtvec,x10
csrrw x13,mtvec,x12
```

Sample CSR Instruction Test assembly

Implementation of MRET

Now we will also modify our controller for adding a new multiplexor and new control signals for detecting the mret instruction. The CSR register file receives this control signal from the pipeline register. In case the mret instruction is received by the CSR register file the value of mepc register will be loaded in the epc/evec output of the register file and epc_taken flag of the mux will be asserted.
For Single Cycle Processor: https://shorturl.at/gmGW5



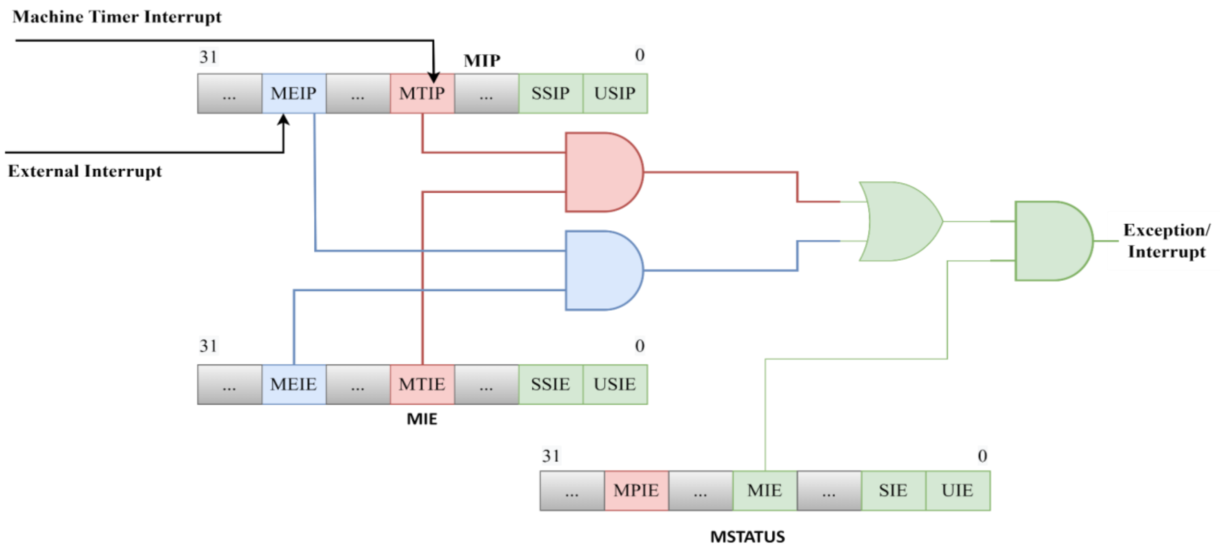
Integration of MRET instruction in the Datapath.

Tasks

- Implement the MRET instruction in your processor.

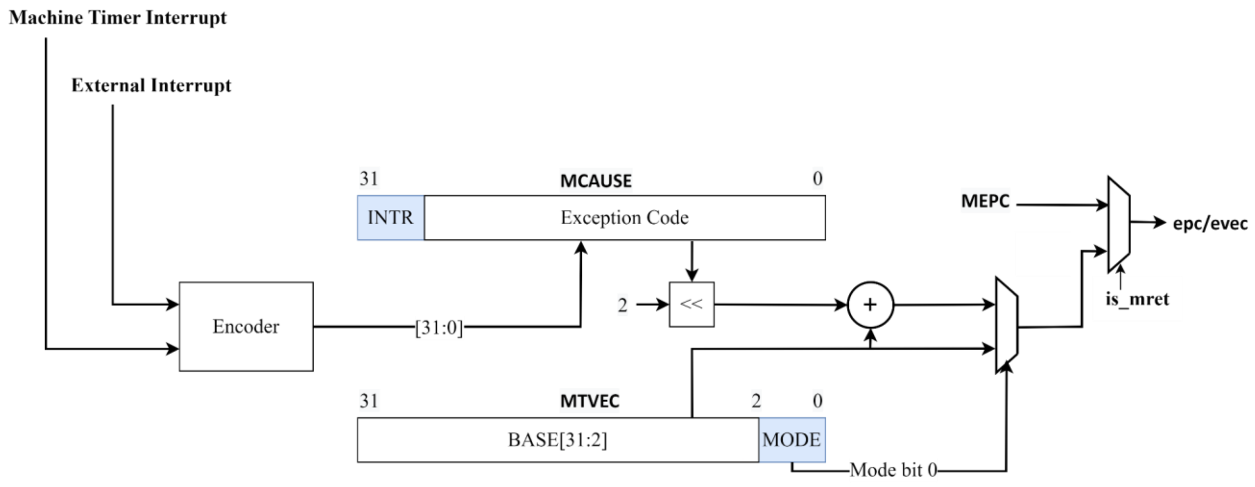
Interrupt Handling

Whenever an interrupt arrives, the interrupt will be registered at the positive edge of the clock in the corresponding bit of the **`mip`** register based on the source of the interrupt. If the corresponding bit of the **`mie`** register is high and the MIE bit of the **`mstatus`** register is high then we say that the Exception/interrupt has occurred and the value of PC in Memory-Writeback stage will be saved in the **`mepc`** register.



Interrupt Handling in CSR Register File.

When the Exception/interrupt has occurred **`epc_taken`** flag in datapath will be asserted and the cause of the interrupt will be saved in the **`mcause`** register by the hardware. Based on the mode of the **`mtvec`** register, the **`epc/evec`** address will be calculated as the ***`base_address`*** and ***`base_address+cause*4`*** in non-vector mode and vector mode respectively. Please note that the **`mie, mstatus and mtvec`** register will be configured by software using ***`csrrw`*** instruction.



Address Calculation in CSR Register File.

Tasks

- Modify the CSR register file for handling single interrupt. Generate the interrupts using testbenches.
- Write a simple assembly code to test whether the processor is handling interrupt. Consult the RISC-V specifications for configuring the CSR registers. A sample code is shown in Listing 9.3.

```
j main
j interrupt_handler

main:
    li x11,(calculate your own value)
    li x12,(calculate your own value)
    li x13,(calculate your own value)
    csrrw x0,mie,x11
    csrrw x0,mstatus,x12
    csrrw x0,mtvec,x13

loop:
    addi x14,x14,1
    j loop

interrupt_handler:
    csrrw x0,mie,x0
    li x2,0xFFFFFFFF
    xor x3,x3,x2
    csrrw x0,mie,x11
    mret
```

Sample Interrupt Handling Test assembly