Task 1

Implement and configure a function and a trap handler to switch modes

a. Pre-requisites

- mstatus specific fields, mepc, mtval, mtvec, mret, mcause, ecall

b. Implement the following in your test:

1. Function:
   - Implement a function which will take an argument (0 or 1) and then jump to relevant mode (0 for supervisor and 1 for user). (This function is only called in machine mode)
2. Trap handler:
   - Implement a trap handler function for mcause (SUPERVISOR_ECALL and USER_ECALL) and jump to 1 higher mode e.g if there is USER_ECALL then after handling the trap it should return in Supervisor mode.

c. Write assembly test:

- Your test should start in M mode (How can you start your test in M-mode?)
- Setup your trap-handler and the function as explained in point b.
- Try switching to S mode
- Verify that you have switched to the required mode(Figure out How to verify this)
- Try switching to U mode. (figure out how you can do this using standard way) Note: You are currently in S mode.
- Now switch to M mode and exit the test. (always exit your test in M mode)

➢ Solution:

Github Link:

https://github.com/BilalAli10x/Riscv_Arch_Test/tree/main/BilalAli_Task1

Test Description:

The objective of this test is to implement and verify privilege mode switching in RISC-V using Machine (M), Supervisor (S), and User (U) modes. The test focuses on:

- Implementing a function that switches privilege modes based on an argument **(0 for Supervisor, 1 for User)**, callable only **from M-mode.**
- Implementing a Machine-mode trap handler that handles **ECALLs** originating from **User** and **Supervisor** modes.
- Verifying correct transitions between M -> S -> U and back to M using architectural CSRs and standard return instructions.

The test is executed on the Spike ISS, starting in Machine mode and always exiting in Machine mode, as required.

Test Flow Overview

Flow:

Start in M
M -> U **(via function)**
U -> ecall -> M -> **return** S

S -> ecall -> M -> **return** M
M -> S **(via function)**
S -> U **(standard way using sret)**
U -> ecall -> M -> **return** S
S -> ecall -> M -> **return** M
Exit in M

Description:

1. Start execution in Machine mode **(M-mode)**.
2. Configure the Machine trap vector **(mtvec)** to point to a custom trap handler.
3. Use a mode-switching function to transition:
    - **M -> U** (via **mret** and **mstatus.MPP** configuration).
4. Trigger ecall from **U-mode**:
    - Trap is handled **in M-mode**.
    - Control is **returned to Supervisor mode (U -> S)**.
5. Trigger ecall **from S-mode**:
    - Trap is handled **in M-mode**.
    - Control is returned to **Machine mode (S -> M).**
6. Again switch:
    - **M -> S** using the **mode-switch function.**
    - **S -> U** using the standard **sret mechanism.**
7. Trigger **ecall** from **U-mode** and **S-mode** again to verify correct upward transitions.
8. Exit the test in Machine mode and **signal pass/fail** via the **tohost** interface.

Mode Switching Function

A function named **switch_mode** is implemented and is only called from **Machine mode**. It takes an argument in register a0:

- a0 = 0 -> Switch to **Supervisor mode**
- a0 = 1 -> Switch to **User mode**

The function works by:

- Clearing the **MPP** field in the **mstatus CSR**.
- Setting **MPP** to the **target privilege mode**.
- Writing the return address to **mepc**.
- Executing **mret** to complete the mode transition.

Switching from Supervisor to User Mode

While in Supervisor mode, switching to User mode is performed using the standard architectural method:

- Clear the **SPP** bit in **sstatus** to indicate the next lower privilege (**User mode**).
- Write the return address to **sepc**.
- Execute **sret** to enter User mode.

Trap Handler Implementation

A Machine-mode trap handler is implemented to handle ECALL exceptions based on the mcause CSR:

- **USER_ECALL (mcause = 8)**
  - The trap handler advances **mepc** to skip the ECALL instruction.
  - Sets **mstatus.MPP = 01 (Supervisor mode)**.
  - Returns using **mret**, transitioning from **U -> S.**
- **SUPERVISOR_ECALL (mcause = 9)**
  - The trap handler advances **mepc**.
  - Sets **mstatus.MPP = 11 (Machine mode).**
  - Returns using **mret**, transitioning from **S -> M.**

Verification Method

Machine Mode (M-mode):

M-mode is verified by accessing the **mstatus CSR**, which is only accessible in Machine mode. Successful access confirms execution in M-mode.

Supervisor Mode (S-mode):

S-mode is verified by issuing an ecall.

- An ecall from Supervisor mode generates **mcause = 9**.
- The trap is handled in Machine mode and returns correctly.
- If execution were still in M-mode, the ECALL would be unexpected and jump to the **failure path**.

Successful handling of mcause = 9 confirms Supervisor mode.

User Mode (U-mode):

User mode is verified using ECALL transitions.

- An ecall from User mode generates mcause = 8 and returns to Supervisor mode.

- A second ecall from Supervisor mode generates mcause = 9 and returns to Machine mode.

Successful execution of both ECALLs without failure confirms correct User mode entry.

Failure Detection:
ECALLs are only allowed from User and Supervisor modes. Any ECALL from Machine mode or unexpected mcause results in test failure.

Actual Output

- The test runs successfully on the Spike ISS.
- All ECALLs are correctly trapped and handled.
- Privilege transitions occur as expected.
- The test exits in Machine mode and writes a success value to the tohost address.

Spike log output confirms correct mode transitions without any unexpected traps or failures.Here is snapshot of spike.log:

```
core   0: >>>>  verify_m_mode
core   0: 0x80000038 (0x300022f3) csrr    t0, mstatus
core   0: 3 0x80000038 (0x300022f3) x5  0x000000a8
core   0: 0x8000003c (0x0fc0006f) j       pc + 0xfc
core   0: 3 0x8000003c (0x0fc0006f)
core   0: 0x80000138 (0x00100193) li      gp, 1
core   0: 3 0x80000138 (0x00100193) x3  0x00000001
core   0: 0x8000013c (0x00001297) auipc   t0, 0x1
core   0: 3 0x8000013c (0x00001297) x5  0x8000113c
core   0: 0x80000140 (0xec32a223) sw      gp, -316(t0)
core   0: 3 0x80000140 (0xec32a223) mem 0x80001000 0x00000001
```

Reference to Specification

- RISC-V Privileged Architecture Specification
  - **Section 3.1.6:** Machine Status Registers **(mstatus, MPP)**
  - **Section 4.1.1:** Supervisor Status Register **(sstatus, SPP)**
  - **Section 3.1.15: Machine Cause Register** (ECALL exception causes (mcause values 8 and 9))
  - Trap return instructions: mret and sret

Answers to Task Questions

- How does the test start in M-mode?
  Answer:
  Spike resets the hart in Machine mode by default.

- How is Supervisor mode verified?
  Answer:
  By issuing an ECALL and observing correct handling of mcause = 9.

- How is User mode entered and verified?
  Answer:
  User mode is entered using sret after clearing SPP, and verified via mcause = 8 on ECALL.

- How does the test exit?
  Answer:
  The test always exits in Machine mode and signals completion via the tohost interface.