Task 2

- Write an assembly test for PMP
- a. Pre-requisites
- i. Read RISCV privileged architecture spec for PMP
- b. Assembly Test
- · Enhance your trap handler from task 1 to handler load/store/execute exceptions
- · Configure two 4KB PMP regions one using TOR method and other using NAPOT method.
- · Grant read permission to NAPOT and execute permission to TOR region
- · Jump to S mode.
- · You will get an Instruction access fault exception after entering into lower modes. Why is that ? (Try to resolve it yourself)
- · Now (after resolving v), do read, write and execute from both the pmp region. You should get appropriate Access faults
- · Jump to machine mode and now apply above (point iii) mentioned pmp restrictions for machine mode too. What do you need to do for this ?
- · Now again read, write and execute from both the pmp region. You should get appropriate Access faults

➢ Solution:

Github Link:

https://github.com/BilalAli10x/Riscv_Arch_Test/tree/main/BilalAli_Task2

Test Description

This test is verifying correct implementation of Physical Memory Protection (PMP) in RISC-V.

This test is:

- Configuring two 4KB PMP regions
- Using TOR method for first region
- Using NAPOT method for second region
- Granting execute permission to TOR region
- Granting read permission to NAPOT region
- Jumping to Supervisor mode
- Generating and handling access faults
- Applying PMP restrictions to Machine mode
- Verifying lock bit functionality

This test is demonstrating privilege-based memory isolation using PMP.

Implementation:

1. Configuring PMP Regions

Configuring Region 0 (TOR – Execute Only)

Defining range:

[_tor_start , _tor_end)

Setting:

- L = 0 A = 01 (TOR) X = 1 W = 0 R = 0
- Binary: 00001100 Hex: 0x0C

Configuring Region 1 (NAPOT – Read Only, 4KB)

Calculating NAPOT encoding:

pmpaddr = (Base >> 2) | ((4096 / 8) - 1)

= (Base >> 2) | (0x1FF)

Setting:

- L = 0 A = 11 (NAPOT) X = 0 W = 0 R = 1
- Binary: 00011001 Hex: 0x19

Writing pmpcfg0

Layout:

[pmp3cfg][pmp2cfg][pmp1cfg][pmp0cfg]

↑ ↑

NAPOT TOR

Writing:

0x00190C00

2. Jumping to Supervisor Mode

Switching privilege level using:

- Clearing MPP bits in mstatus
- Setting MPP = 01 (Supervisor)
- Writing mepc
- Executing mret

After entering S-mode, an Instruction Access Fault is occurring.

3. Explaining Instruction Access Fault

The fault is occurring because of:

Default Deny model for S and U modes.

When an address is not matching any PMP entry:

- S-mode → Access denied
- U-mode → Access denied
- M-mode → Access allowed

At the time of switching mode, the current PC is not matching any executable PMP region.

Therefore instruction fetch is generating:

mcause = 1 (Instruction Access Fault)

4. Resolving Instruction Access Fault

Resolving this by modifying PMP configuration inside trap handler.

Granting R/W/X permission to Default Deny Region region: 0x0F
Returning using mret and continuing execution.
This is demonstrating whitelist security model of PMP.

Test Flow

Phase 1 – Testing in Supervisor Mode

Testing TOR Region (Execute Only)

Operation	Expected Result
Load	Load Access Fault (5)
Store	Store Access Fault (7)
Execute	Allowed

Testing NAPOT Region (Read Only)

Operation	Expected Result
Load	Allowed
Store	Store Access Fault (7)
Execute	Instruction Access Fault (1)

Trap handler is:

- Incrementing mepc
- Returning using mret
- Continuing execution

Phase 2 – Returning to Machine Mode

Triggering:

- ecall from Supervisor
- Generating:
- mcause = 9

Setting:

- MPP = 11 (Machine)
- Returning using mret.

Phase 3 – Applying PMP Restrictions to Machine Mode

By default:

- PMP is not restricting Machine mode.

- Machine mode is following Default Allow model.

Applying restrictions by:

Setting Lock bit (L = 1) in both PMP entries.

Updating configuration:

- Region 0: 0x0C → 0x8C
- Region 1: 0x19 → 0x99

Masking:

0x00998C00

Setting L-bit is:

- Making configuration immutable
- Applying PMP rules to Machine mode

Phase 4 – Testing in Machine Mode (Locked)

Now Machine mode is behaving same as Supervisor mode.

Testing TOR Region (Execute Only)

Operation	Expected Result
Load	Load Access Fault (5)
Store	Store Access Fault (7)
Execute	Allowed

Testing NAPOT Region (Read Only)

Operation	Expected Result
Load	Allowed
Store	Store Access Fault (7)
Execute	Instruction Access Fault (1)

This is verifying that Lock bit is correctly enforcing PMP in Machine mode.

Expected Output

Program is ending with:

gp = 1 → PASS

gp = 2 → FAIL

Spike log is showing:

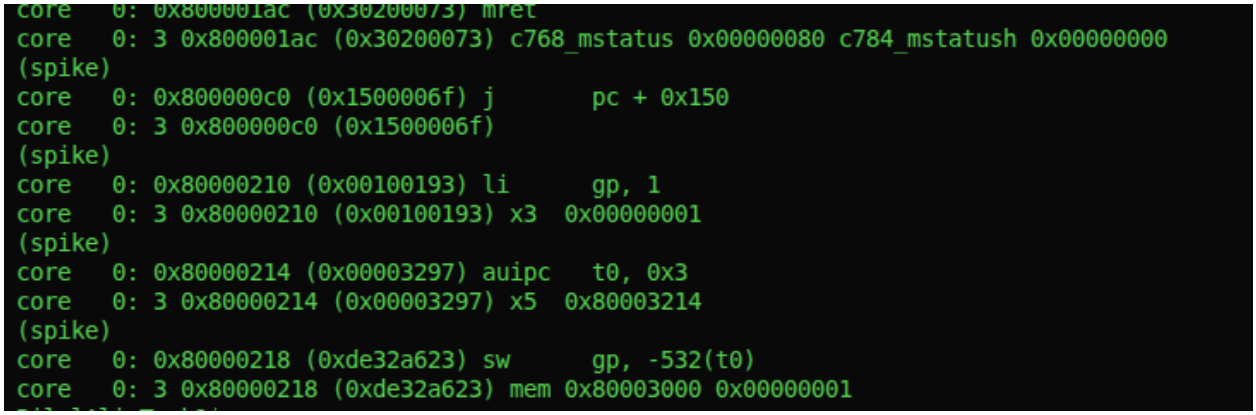
- Instruction Access Fault

- Load Access Fault
- Store Access Fault
- ECALL from Supervisor

After locking:

- **Load/Store faults also occurring in Machine mode**

Spike Log Snapshot Example:



Reference to Specification

RISC-V Privileged Architecture Specification

- **Section 3.1.6** - mstatus, MPP
- **Section 4.1.1** - sstatus, SPP
- **Section 3.1.15** - mcause
    - ECALL from U = 8
    - ECALL from S = 9
- **Section 3.7:PMP Section** - Address matching and L-bit behavior