RISC V ARCh TEST
TASK07

Task Description : First Set the PMP region then set MML bit, try to add execution permission to only the mode region or the shared region

Q: Add a rule with executable privileges that is either M-mode-only or a locked Shared-Region?? What happens and why.

Answer: Adding a rule with executable privileges that is either M-mode-only or a locked Shared-Region is not possible, and such pmpcfg writes are ignored, leaving pmpcfg unchanged. This restriction can be temporarily lifted, e.g., during the boot process, by setting mseccfg.RLB.

Sol: I declared 4 PMP regio,ns each of b : at the start of the code set RLB bit

1:0x80000000 to 0x80001000 set by NAPOT with locked and Execution Permissions as MLL bit set, permissions interpretation as, only machine mode to execute the code And all other combinations get access fault.

```
core   0: 0x8000004c (0x3b031073) csrw    pmpaddr0, t1
core   0: 3 0x8000004c (0x3b031073) c944_pmpaddr0 0x200001ff
```

2: 0x80001000 to 0x80002000set by NAPOT region with only Write permissions as we set MML This region becomes a shared data region ,read and write by M-mode and only read by S-mode will be allowed, other combinations to get access faults.

```
core   0: 0x80000060 (0x3b131073) csrw    pmpaddr1, t1
core   0: 3 0x80000060 (0x3b131073) c945_pmpaddr1 0x200005ff
```

3: 0x80002000 -0x80003000 by NAPOT region locked and with only read permissions As we set the MML bit, this region read only by M-mode and it give access fault for all combinations.

```
core   0: 0x80000074 (0x3b231073) csrw    pmpaddr2, t1
core   0: 3 0x80000074 (0x3b231073) c946_pmpaddr2 0x200009ff
```

4: 0x80003000 -0x80004000 by NAPOT region locked with only Execution permission As MLL set M mode get only Execute permission and all other combinations get access fault

```
core   0: 0x80000088 (0x3b331073) csrw    pmpaddr3, t1
core   0: 3 0x80000088 (0x3b331073) c947_pmpaddr3 0x20000dff
```

RISC V ARCh TEST
TASK07

5: Set The MML bit and Jump to region which have only read permission
Access fault expected

```
core   0: 0x800000a4 (0x75d0106f) j       pc + 0x1f5c
core   0: 3 0x800000a4 (0x75d0106f)
core   0: exception trap_instruction_access_fault, epc 0x80002000
```

6: We set mccge with Set MML and RLB ,so in trap handler we can update the permissions Read only will be updated as executed only by mode

```
core   0: 3 0x80003004 (0xa9c50513) x10 0x9c9a1a9c
core   0: 0x80003008 (0x3a051073) csrw    pmpcfg0, a0
core   0: 3 0x80003008 (0x3a051073) c928_pmpcfg0 0x9c9a1a9c
```

7: Return from trape and execute the instruction that caused the exception as the Permissions are updated and then jump test pass lable

```
core   0: 0x8000300c (0x30200073) mret
core   0: 3 0x8000300c (0x30200073) c768_mstatus 0x00000080
core   0: 0x80002000 (0x02800513) li      a0, 40
core   0: 3 0x80002000 (0x02800513) x10 0x00000028
core   0: 0x80002004 (0x8bcfe06f) j       pc - 0x1f44
```